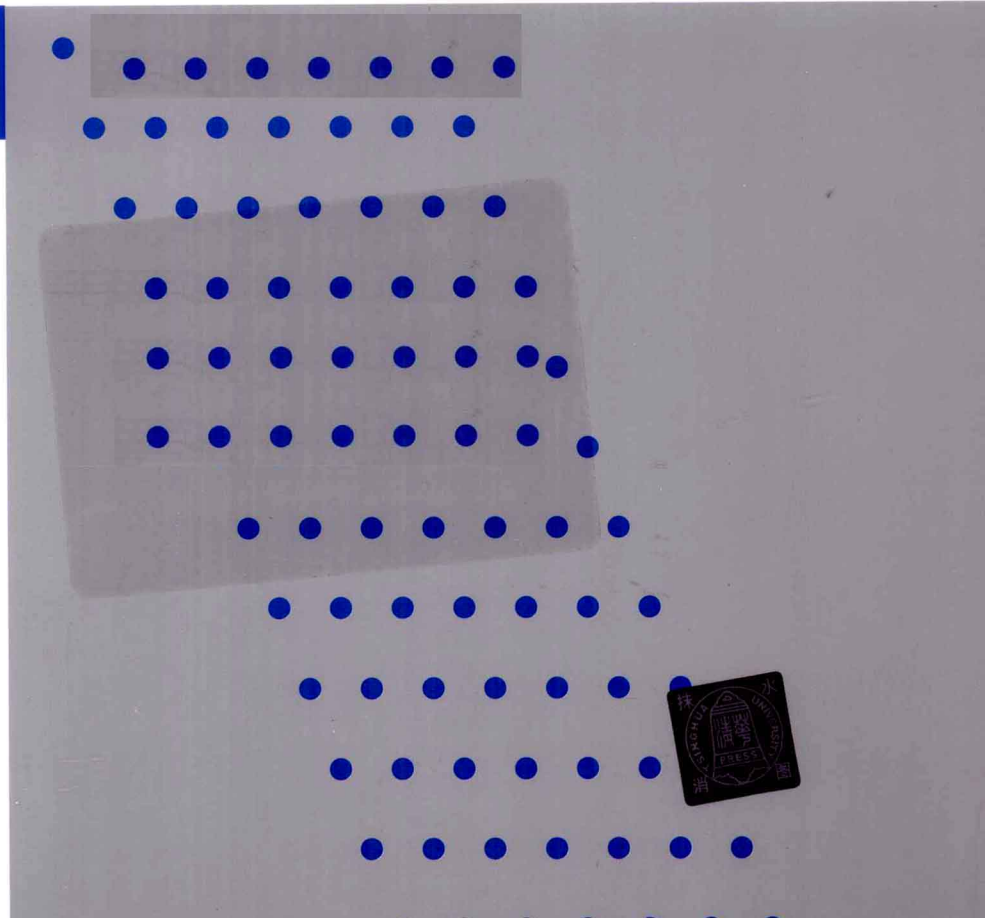
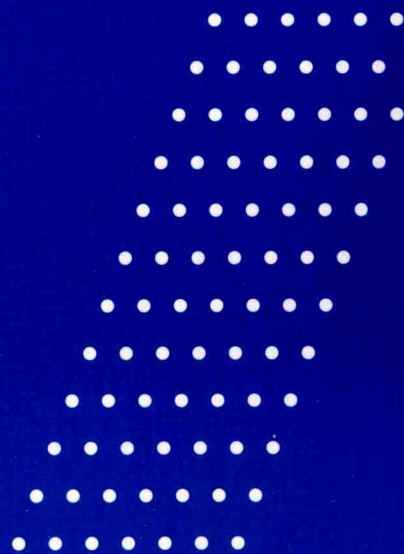


宁杨 周毓林 编著

# 嵌入式系统 基础及应用



21世纪高等学校嵌入式系统专业规划教材

宁杨 周毓林 编著

# 嵌入式系统 基础及应用

清华大学出版社  
北京

## 内 容 简 介

本书从基础知识着手,比较全面地介绍嵌入式系统的基础知识、常用外设接口及嵌入式操作系统的实现。书中所有示例均使用 MDK 在 STM32 系列 MCU 上实现,读者不必另外购买开发板即可完成实验,所有的实验结果均是可见、可操作、可改的。全书共 10 章,基础部分,在讲解 Cortex-M3 处理器结构的基础上,详细介绍了 Cortex-M3 处理器的编程模型、总线架构、存储结构、异常处理机制等。应用部分,结合 STM32 处理器介绍了所有接口及其结构、特点和功能,并给出了所有相应特殊功能寄存器的功能;每章最后一节均有独立的习题以及供读者思考的问题。

本书可作为高等学校的专业课教材使用,同时也可以作为嵌入式应用开发工程师技术人员 ARM 最新处理器核 Cortex-M3 的编程入门指导书,又可以作为使用 MDK 进行 STM32 处理器开发的参考书。

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

嵌入式系统基础及应用/宁杨,周毓林编著. —北京:清华大学出版社,2012.1  
(21 世纪高等学校嵌入式系统专业规划教材)

ISBN 978-7-302-26766-9

I. ①嵌… II. ①宁… ②周… III. ①微处理器—系统设计—高等学校—教材 IV. ①TP332

中国版本图书馆 CIP 数据核字(2011)第 185716 号

责任编辑:高买花 薛 阳

责任校对:李建庄

责任印制:何 芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954,jsjic@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015,zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京密云胶印厂

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185×260 印 张:25.5 字 数:635 千字

版 次:2012 年 1 月第 1 版 印 次:2012 年 1 月第 1 次印刷

印 数:1~3000

定 价:39.50 元

# 前 言

本书从基础知识着手,比较全面地介绍了嵌入式系统的基础知识、常用外设接口及嵌入式操作系统的实现。为了方便讲解,书中所有示例均使用 MDK 在 STM32 系列 MCU 上实现。以 STM32 为例的原因,主要是考虑 STM32 接口齐全,在消费电子领域应用非常广泛,是目前比较主流的 MCU;并且考虑读者操作方便,其中的实验均在 MDK 的仿真环境下实现,读者不必另外购买开发板即可完成实验,所有的实验结果均是可见、可操作、可改的。

全书共 10 章,主要分为两个部分讲解。基础部分,在讲解 Cortex-M3 处理器结构的基础上,详细介绍了 Cortex-M3 处理器的编程模型、总线架构、存储结构、异常处理机制等。应用部分,结合 STM32 处理器介绍了所有接口及其结构、特点和功能,并给出了所有相应特殊功能寄存器的功能;每章最后一节均有独立的习题以及供读者思考的问题。同时本书提供了 11 个独立的实验供教学者参考,可以全做也可以只做其中的一部分,每个实验均覆盖了书中所讲述的理论知识,以达到理论与实践并用的效果。

本书可作为高等学校的专业课教材使用。书中每个实验花费的时间最长不超过 90 分钟。书中的实验覆盖了书中所讲述的理论知识,达到理论与实践并用的效果。同时本书也可以作为嵌入式应用开发工程师技术人员 ARM 最新处理器核 Cortex-M3 的编程入门指导书,又可以作为使用 MDK 进行 STM32 处理器开发的参考书,还可以作为 STM32 处理器的开发参考手册。

作 者

2011 年 12 月

# 目 录

<b>第 1 章 嵌入式系统概述</b> .....	1
1.1 嵌入式系统简介 .....	1
1.1.1 嵌入式系统的历史.....	1
1.1.2 嵌入式系统的定义.....	2
1.1.3 其他相关联的概念.....	3
1.2 嵌入式系统结构与应用领域 .....	4
1.2.1 单片嵌入式系统.....	4
1.2.2 单片机的发展历史.....	4
1.2.3 单片机的发展趋势.....	5
1.2.4 单片机嵌入式系统的结构.....	8
1.2.5 单片机嵌入式系统的应用领域.....	9
1.3 习题.....	10
<b>第 2 章 嵌入式系统的开发与设计</b> .....	11
2.1 嵌入式系统的应用模式.....	11
2.1.1 嵌入式处理器 .....	11
2.1.2 嵌入式操作系统 .....	13
2.2 基于单片机的嵌入式系统开发.....	15
2.2.1 嵌入式系统的特点 .....	15
2.2.2 基于单片机的嵌入式系统开发过程 .....	16
2.2.3 基于单片机的嵌入式系统开发模式 .....	17
2.2.4 嵌入式系统开发工具 .....	18
2.3 基于单片机的嵌入式系统设计.....	20
2.3.1 嵌入式系统设计的基本指导思想 .....	20
2.3.2 基于单片机的嵌入式系统设计基本原则与依据 .....	20
2.3.3 设计开发时的注意事项 .....	21
2.3.4 嵌入式系统设计方法(从单片机到单片系统) .....	23
2.4 习题.....	26
<b>第 3 章 开发环境</b> .....	27
3.1 Keil $\mu$ Vision3 介绍 .....	27
3.2 Keil $\mu$ Vision3 的功能 .....	27
3.3 Keil $\mu$ Vision3 的安装与配置 .....	28

3.3.1	Keil $\mu$ Vision3 系统要求 .....	28
3.3.2	Keil $\mu$ Vision3 安装 .....	28
3.3.3	Keil $\mu$ Vision3 目录结构 .....	28
3.3.4	注册与帮助 .....	31
3.4	开发过程 .....	32
3.4.1	创建工程并选择处理器 .....	32
3.4.2	配置处理器启动代码 .....	32
3.4.3	配置硬件选项 .....	34
3.4.4	创建源文件及文件组 .....	35
3.4.5	编译链接工程 .....	40
3.4.6	调试程序 .....	41
3.4.7	建立 HEX 文件 .....	43
3.4.8	下载 HEX 文件 .....	43
3.5	习题 .....	44
<b>第 4 章</b>	<b>基于 ARM Cortex-M3 的嵌入式处理器</b> .....	<b>45</b>
4.1	ARM 及 ARM 架构的背景 .....	45
4.2	ARM 的架构版本 .....	46
4.3	ARM Cortex-M3 处理器初探 .....	47
4.4	Cortex-M3 处理器内核 vs. 基于 Cortex-M3 的 MCU .....	49
4.5	Thumb-2 指令集体系结构 .....	52
4.6	习题 .....	53
<b>第 5 章</b>	<b>Cortex-M3 处理器寄存器及指令系统</b> .....	<b>54</b>
5.1	寄存器组 .....	54
5.1.1	寄存器 .....	54
5.1.2	Cortex-M3 的寄存器 .....	54
5.2	处理器操作模式与特权级别 .....	55
5.3	ARM 汇编语言基础 .....	56
5.3.1	ARM 汇编伪指令 .....	56
5.3.2	ARM 的汇编语言结构 .....	59
5.4	特殊功能寄存器组 .....	60
5.4.1	程序状态寄存器(PSRs 或 PSR) .....	60
5.4.2	PRIMASK、FAULTMASK 和 BASEPRI .....	61
5.4.3	控制寄存器(CONTROL) .....	61
5.5	异常和中断 .....	62
5.6	嵌套向量中断控制器 .....	63
5.7	存储器保护单元 .....	63
5.8	堆栈区的操作 .....	64

5.8.1 堆栈的基本操作 .....	64
5.8.2 Cortex-M3 堆栈的实现 .....	64
5.8.3 Cortex-M3 的双堆栈机制 .....	65
5.9 习题 .....	66
<b>第 6 章 STM32 架构简介 .....</b>	<b>67</b>
6.1 STM32 系列微控制器简介 .....	67
6.2 STM32 微控制器的分类 .....	67
6.2.1 STM32F101xx 系列微控制器 .....	67
6.2.2 STM32F103xx 系列微控制器 .....	73
6.3 STM32 微控制器的主要优点 .....	81
6.4 STM32 微控制器的应用 .....	83
6.5 习题 .....	83
<b>第 7 章 嵌入式处理器外设 .....</b>	<b>84</b>
7.1 存储器和总线结构 .....	84
7.1.1 系统结构 .....	84
7.1.2 存储器结构 .....	85
7.1.3 嵌入式 SRAM .....	85
7.1.4 位段 .....	85
7.1.5 嵌入式闪存存储器(Flash) .....	88
7.1.6 启动配置 .....	89
7.2 电源控制 .....	89
7.2.1 电源供应 .....	89
7.2.2 电源供应管理 .....	91
7.2.3 低功耗模式 .....	92
7.2.4 电源控制寄存器 .....	96
7.3 复位和时钟控制 .....	98
7.3.1 复位 .....	98
7.3.2 时钟 .....	99
7.3.3 RCC 寄存器描述 .....	103
7.4 通用 I/O 和复用 I/O(GPIO 和 AFIO) .....	119
7.4.1 GPIO 介绍 .....	119
7.4.2 STM32 下的 GPIO 描述 .....	119
7.4.3 GPIO 寄存器描述 .....	122
7.4.4 复用功能 I/O 和调试配置(AFIO) .....	128
7.4.5 AFIO 寄存器描述 .....	132
7.4.6 实验: 通过使用 Keil $\mu$ Vision3 在 STM32 模拟器下实现 GPIO 的 跑马灯程序 .....	137

7.5	外部中断和事件 .....	143
7.5.1	什么是中断 .....	143
7.5.2	STM32F10x 外部中断/事件功能描述 .....	147
7.5.3	EXTI 相关功能寄存器 .....	149
7.5.4	实验: 通过使用 Keil $\mu$ Vision3 在 STM32 模拟器下实现按键处理 .....	151
7.6	通用定时器(TIMx) .....	157
7.6.1	简介 .....	157
7.6.2	主要特性 .....	157
7.6.3	功能描述 .....	158
7.6.4	通用定时器(TIMx)寄存器描述 .....	172
7.6.5	实验: 通过使用 Keil $\mu$ Vision3 在 STM32 模拟器下对定时器的实现 .....	188
7.7	实时时钟 .....	194
7.7.1	简介 .....	194
7.7.2	主要特性 .....	194
7.7.3	功能描述 .....	195
7.7.4	RTC 寄存器描述 .....	198
7.7.5	实验: 通过使用 Keil $\mu$ Vision3 在 STM32 模拟器下对定时器下实现仿真钟表输出 .....	202
7.8	习题 .....	213
<b>第 8 章</b>	<b>STM32 外部通信接口 .....</b>	<b>214</b>
8.1	串行通信接口 .....	214
8.1.1	串行通信接口(UART)介绍 .....	214
8.1.2	STM32 下的 USART 操作 .....	216
8.1.3	STM32 下的 UART 寄存器描述 .....	234
8.1.4	实验: 通过使用 Keil $\mu$ Vision3 在 STM32 模拟器下实现 UART 的字符输入输出 .....	243
8.2	SPI .....	247
8.2.1	SPI 介绍 .....	247
8.2.2	STM32 下的 SPI 操作 .....	250
8.2.3	STM32 下的 SPI 寄存器介绍 .....	258
8.2.4	实验: 通过使用 Keil $\mu$ Vision3 在 STM32 模拟器下实现 SPI 的输出 .....	263
8.3	I <sup>2</sup> C .....	269
8.3.1	I <sup>2</sup> C 介绍 .....	269
8.3.2	STM32 下的 I <sup>2</sup> C 操作 .....	273
8.3.3	STM32 下的 I <sup>2</sup> C 寄存器介绍 .....	283
8.3.4	实验: 通过使用 Keil $\mu$ Vision3 在 STM32 模拟器下实现 I <sup>2</sup> C 口 .....	



的输出 .....	293
8.4 DMA 控制器 .....	303
8.4.1 DMA 介绍 .....	303
8.4.2 STM32 下的 DMA 操作 .....	305
8.4.3 STM32 下的 DMA 寄存器 .....	309
8.4.4 实验：通过使用 Keil $\mu$ Vision3 在 STM32 模拟器下实现 DMA 口的控制 .....	313
8.5 A/D 转换 .....	320
8.5.1 A/D 转换介绍 .....	320
8.5.2 STM32 下的 A/D 转换器操作 .....	323
8.5.3 STM32 下的 A/D 转换寄存器描述 .....	335
8.5.4 实验：通过使用 Keil $\mu$ Vision3 在 STM32 模拟器下实现 A/D 转换器的控制 .....	346
8.6 习题 .....	352
<b>第 9 章 前后台操作系统介绍 .....</b>	<b>354</b>
9.1 前后台操作系统的特征 .....	354
9.1.1 前后台操作系统的前台与后台 .....	354
9.1.2 前后台操作系统的任务特征 .....	355
9.1.3 前后台操作系统的实时特征 .....	357
9.2 实现一个前后台操作系统 .....	360
9.2.1 分析前后台操作系统 .....	360
9.2.2 将消息队列应用到前后台操作系统中 .....	362
9.2.3 变量的互斥条件 .....	364
9.3 实验：实现一个基于消息驱动的前后台操作系统 .....	366
9.3.1 实验目的 .....	366
9.3.2 实验假设 .....	366
9.3.3 准备工作 .....	366
9.3.4 实验步骤 .....	366
9.4 习题 .....	370
<b>第 10 章 基于多任务的嵌入式操作系统介绍 .....</b>	<b>371</b>
10.1 多任务操作系统 .....	371
10.2 实现一个多任务操作系统 .....	374
10.2.1 前期准备 .....	374
10.2.2 系统启动 .....	375
10.2.3 实现基本调度定义 .....	376
10.2.4 实现任务 .....	378
10.2.5 调度进阶 .....	384

10.3	实验：分析一个简单的多任务操作系统内核的系统利用率	388
10.3.1	实验目的	388
10.3.2	准备工作	388
10.3.3	实验步骤	388
10.4	习题	393
<b>参考文献</b>		<b>394</b>

# 第 1 章 嵌入式系统概述

## 1.1 嵌入式系统简介

嵌入式系统本身是一个相对模糊的定义。目前嵌入式系统(Embedded System)应用的普遍性和适用性及其目标几乎是无处不在,无所不能,而恰恰由于这种范围的扩大,使得“嵌入式系统”更加难于明确定义。

举个简单例子:PC 键盘是否是一个嵌入式产品?答案肯定是“是”。另外,一个汽车上的控制系统,如娱乐系统是嵌入式系统吗?当然也是。而工业控制则是嵌入式系统技术的最主要的一个典型应用领域。那到底什么是嵌入式系统?莫非嵌入式系统只是一个难以定义的抽象概念?

### 1.1.1 嵌入式系统的历史

虽然嵌入式系统是近几年才风靡起来的,但是这个概念并非最近才出现。从 20 世纪 70 年代单片机的出现到今天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近 30 年的发展历史。

作为一个系统,往往是在硬件和软件交替发展的双螺旋的支撑下逐渐趋于稳定和成熟,嵌入式系统也不例外。

嵌入式系统的出现最初是基于单片机的。20 世纪 70 年代单片机的出现,使得汽车、家电、工业机器、通信装置以及成千上万种产品可以通过内嵌电子装置来获得更佳的使用性能:更容易使用、更快、更便宜。这些装置已经初步具备了嵌入式的应用特点,但是这时的应用只是使用 8 位的芯片,执行一些单线程的程序,还谈不上“系统”的概念。

**注意:**最早的单片机是 Intel 公司的 8048,它出现在 1976 年。Motorola 同时推出了 68HC05,Zilog 公司推出了 Z80 系列,这些早期的单片机均含有 256 字节的 RAM、4KB 的 ROM、4 个 8 位并口、1 个全双工串行口、两个 16 位定时器。之后在 20 世纪 80 年代初,Intel 又进一步完善了 8048,在它的基础上研制成功了 8051,这在单片机的历史上是值得纪念的一页。迄今为止,51 系列的单片机仍然是最为成功的单片机芯片,在各种产品中有着非常广泛的应用。

从 20 世纪 80 年代早期开始,嵌入式系统的程序员开始用商业级的“操作系统”编写嵌入式应用软件,这使得可以获取更短的开发周期,更低的开发资金和更高的开发效率,至此“嵌入式系统”真正出现了。确切地说,这个时候的操作系统是一个实时核,这个实时核包含许多传统操作系统的特征,包括任务管理、任务间通信、同步与相互排斥、中断支持、内存管理等功能。其中比较著名的有 Ready System 公司的 VRTX、Integrated System Incorporation(ISI)的 PSOS 和 IMG 的 VxWorks、QNX 公司的 QNX 等。这些嵌入式操作系统都具有嵌入式的典型特点:它们均采用占先式的调度,响应的时间很短,任务执行的

时间可以确定；系统内核很小，具有可裁减、可扩充和可移植性，可以移植到各种处理器上；较强的实时和可靠性，适合嵌入式应用。这些嵌入式实时多任务操作系统的出现，使得应用开发人员得以从小范围的开发中解放出来，同时也促使嵌入式有了更为广阔的应用空间。

20 世纪 90 年代以后，随着对实时性要求的提高，软件规模不断上升，实时核逐渐发展为实时多任务操作系统(RTOS)，并作为一种软件平台逐步成为目前国际嵌入式系统的主流。这时更多的公司看到了嵌入式系统的广阔发展前景，开始大力发展自己的嵌入式操作系统。除了以上几家老牌公司的产品以外，还出现了 Palm OS、WinCE、嵌入式 Linux、Lynx、Nucleux，以及国内的 Hopen、Delta Os 等嵌入式操作系统。随着嵌入式技术的发展前景日益广阔，相信会有更多的嵌入式操作系统软件出现。

### 1.1.2 嵌入式系统的定义

根据 IEEE(电气和电子工程师协会)的定义，嵌入式系统是“控制、监视或者辅助装置、机器和设备运行的装置”(devices used to control, monitor, or assist the operation of equipment, machinery or plants)。从中可以看出嵌入式系统是软件和硬件的综合体，还可以涵盖机械等附属装置。目前国内一个普遍被认同的定义是：以应用为中心、以计算机技术为基础、软件硬件可裁减、适应应用系统对功能、可靠性、成本、体积、功耗严格要求的专用计算机系统。这个定义可从几方面来理解：

- 嵌入式系统是面向用户、面向产品、面向应用的，它必须与具体应用相结合才会具有生命力、才更具有优势。因此可以这样理解上述“三个面向”的含义，即嵌入式系统是与应用紧密结合的，它具有很强的专用性，必须结合实际系统需求进行合理的裁减利用。
- 嵌入式系统是将先进的计算机技术、半导体技术和电子技术和各个行业的具体应用相结合后的产物，这一点就决定了它必然是一个技术密集、资金密集、高度分散、不断创新的知识集成系统。所以，介入嵌入式系统行业，必须有一个正确的定位。例如 Palm 之所以在 PDA 领域占有 70% 以上的市场，就是因为其立足于个人电子消费品，着重发展图形界面和多任务管理；而风河的 VxWorks 之所以在火星车上得以应用，则是因为其高实时性和高可靠性。
- 嵌入式系统必须根据应用需求对软硬件进行裁减，满足应用系统对功能、可靠性、成本、体积等的要求。所以，如果能建立相对通用的软硬件基础，然后在其上开发出适应各种需要的系统，是一个比较好的发展模式。目前的嵌入式系统的核心往往是一个只有几个千字节到几十千字节的微内核，需要根据实际的使用进行功能扩展或者裁减，但是由于微内核的存在，使得这种扩展能够非常顺利地进行。

实际上，嵌入式系统本身是一个外延极广的名词，凡是与产品结合在一起的具有嵌入式特点的系统都可以叫嵌入式系统，而且有时很难以给它下一个准确的定义。现在人们在谈到嵌入式系统时，某种程度上指近些年比较热门的具有操作系统的嵌入式系统，本文在进行分析和展望时，也沿用这一观点。

一般而言，嵌入式系统的构架可以分成 4 个部分：处理器、存储器、输入输出(I/O)和软件(由于多数嵌入式设备的应用软件和操作系统都是紧密结合的，在这里对其不加以区分，

这也是嵌入式系统和 Windows 系统的最大区别)。

### 1.1.3 其他相关联的概念

嵌入式系统中有许多非常重要的概念。

#### 1. 嵌入式处理器

嵌入式处理器是嵌入式系统的核心,是控制、辅助系统运行的硬件单元。范围极其广阔,从最初的 4 位处理器,到目前仍在大规模应用的 8 位单片机,到最新的受到广泛青睐的 32 位、64 位嵌入式 CPU。

#### 2. 实时操作系统

实时操作系统是嵌入式系统目前最主要的组成部分。根据操作系统的工作特性,实时是指物理进程的真实时间。实时操作系统具有实时性、能从硬件方面支持实时控制系统工作的操作系统。其中实时性是第一要求,需要调度一切可利用的资源完成实时控制任务,其次才着眼于提高计算机系统的使用效率,其重要特点是要满足对时间的限制和要求。

#### 3. 分时操作系统

对于分时操作系统,软件的执行在时间上的要求并不严格,时间上的错误一般不会造成灾难性的后果。目前分时系统的强项在于多任务的管理,而实时操作系统的重要特点是具有系统的可确定性,即系统能对运行情况的最好和最坏等情况做出精确的估计。

#### 4. 多任务操作系统

系统支持多任务管理和任务间的同步和通信,传统的单片机系统和 DOS 系统等对多任务支持的功能很弱,而目前的 Windows 是典型的多任务操作系统。在嵌入式应用领域中,多任务是一个普遍的要求。

#### 5. 实时操作系统中的重要概念

系统响应时间(System Response Time): 系统发出处理要求到系统给出应答信号的时间。

任务换道时间(Context-Switching Time): 任务之间切换使用的时间。

中断延迟(Interrupt Latency): 计算机接收到中断信号到操作系统做出响应,并完成换道转入中断服务程序的时间。

#### 6. 实时操作系统的工作状态

实时系统中的任务有运行(Executing)、就绪(Ready)、挂起(Suspended)和冬眠(Dormant) 4 种状态。

运行: 获得 CPU 控制权。

就绪: 进入任务等待队列,通过调度转为运行状态。

挂起: 任务发生阻塞,移出任务等待队列,等待系统实时事件的发生而唤醒,从而转为就绪或运行。

冬眠: 由于任务完成或错误等原因被清除的任务,也可以认为是系统中不存在的任务。

任何时刻系统中只能有一个任务处在运行状态,各任务按级别通过时间片分别获得对 CPU 的访问权。

## 1.2 嵌入式系统结构与应用领域

### 1.2.1 单片嵌入式系统

以单片机为核心的单片嵌入式系统大部分应用于专业性极强的工业控制系统中。其主要特点是结构和功能相对单一,存储容量较小,计算能力和效率比较低,简单的用户接口。由于这种嵌入式系统功能专一可靠、价格便宜,因此在工业控制、电子智能仪器设备等领域有着广泛的应用。

单片机作为单片嵌入式系统的核心控制部件,从体系结构到指令系统都是按照嵌入式系统的应用特点专门设计的,它能最好地满足面向控制对象、应用系统的嵌入、现场的可靠运行和优良的控制功能要求。因此,单片嵌入式应用是发展最快、品种最多、数量最大的嵌入式系统,在其基本体系结构上,可衍生出能满足各种不同应用系统要求的系统和产品。用户可根据应用系统的各种不同要求和功能,选择最佳型号的单片机。

作为一个典型的嵌入式系统——单片嵌入式系统,在我国大规模应用已有几十年的历史。它不但是在中、小型工控领域、智能仪器仪表、家用电器、电子通信设备和电子系统中最重要的工具和最普遍的应用手段,同时正是由于单片嵌入式系统的广泛应用和不断发展,也大大推动了嵌入式系统技术的快速发展。因此对于电子、通信、工业控制、智能仪器仪表等相关专业的学生来讲,深入学习和掌握单片嵌入式系统的原理与应用,不仅能对自己所学的基础知识进行检验,而且能够培养和锻炼自己的问题分析、综合应用和动手实践的能力,掌握真正的专业技能和应用技术。同时,深入学习和掌握单片嵌入式系统的原理与应用,也为更好地掌握其他嵌入式系统打下重要的基础,这个特点尤其表现在硬件设计方面。

### 1.2.2 单片机的发展历史

单片机诞生于 20 世纪 70 年代末,经历了 SCM、MCU、SoC 三大阶段。

#### 1. SCM(Single Chip Microcomputer,单片微型计算机)阶段

该阶段主要是寻求最佳的单片形态嵌入式系统的最佳体系结构。“创新模式”获得成功,奠定了 SCM 与通用计算机完全不同的发展道路。在开创嵌入式系统独立发展的道路上,Intel 公司功不可没。

#### 2. MCU(Micro Controller Unit,微控制器)阶段

该阶段主要的技术发展方向是不断扩展满足嵌入式应用时,对象系统要求的各种外围电路与接口电路,突显其对象的智能化控制能力。它所涉及的领域都与对象系统相关,因此,发展 MCU 的重任不可避免地落在电气、电子技术厂家。从这一角度来看,Intel 逐渐淡出 MCU 的发展也有其客观因素。在发展 MCU 方面,最著名的厂家当数 Philips 公司。

Philips 公司以其在嵌入式应用方面的巨大优势,将 MCS-51 从单片微型计算机迅速发展 to 微控制器。因此,当回顾嵌入式系统发展的道路时,不应忘记 Intel 和 Philips 的历史功绩。

#### 3. 单片机是嵌入式系统的独立发展之路

向 MCU 阶段发展的重要因素,就是寻求应用系统在芯片上的最大化解决方案;因此,

专用单片机的发展自然形成了 SoC(片上系统)化趋势。随着微电子技术、IC 设计、EDA 工具的发展,基于 SoC 的单片机应用系统设计会有较大的发展。因此,对单片机的理解可以从单片微型计算机、单片微控制器延伸到单片应用系统。

### 1.2.3 单片机的发展趋势

计算机系统的发展已明显地朝三个方向发展,这三个方向就是巨型化,单片化,网络化。解决复杂系统计算和高速数据处理的仍然是巨型计算机,故而,巨型计算机目前正在朝高速及处理能力的方向努力。单片机在出现时,Intel 公司就为其单片机取名为嵌入式微控制器(Embedded Microcontroller)。单片机的最明显的优势,就是可以嵌入到各种仪器、设备中。这一点是巨型计算机和网络不可能做到的。

#### 1. 单片机的技术发展

单片机的技术进步反映在内部结构、功率消耗、外部电压等级以及制造工艺上。在这几方面,较为典型地说明了单片机的水平。在目前,用户对单片机的需要越来越多,但是,要求也越来越高。下面分别就这 4 个方面说明单片机的技术进步状况。

##### (1) 内部结构的进步。

① 多功能集成化。单片机在内部已集成了越来越多的部件,这些部件包括一般常用的电路,例如:定时器、比较器、A/D 转换器、D/A 转换器、串行通信接口、Watchdog 电路、LCD 控制器、USB 控制器、以太网控制器等。

有的单片机为了构成控制网络或形成局部网,内部含有局部网络控制模块 CAN。例如,Infineon 公司的 C505C, C515C, C167CR, C167CS-32FM, 81C90; Motorola 公司的 68HC08AZ 系列等; ST 公司的 STM32 等,因此,这类单片机十分容易构成网络。特别是在控制较为复杂的系统时,构成一个控制网络十分有用。

为了能在变频控制中方便使用单片机,形成最具经济效益的嵌入式控制系统。有的单片机内部设置了专门用于变频控制的脉宽调制控制电路,这些单片机有 Fujitsu 公司的 MB89850 系列、MB89860 系列; Motorola 公司的 MC68HC08MR16、MR24 等。在这些单片机中,脉宽调制电路有 6 个通道输出,可产生三相脉宽调制交流电压,并且内部含死区控制等功能。

② 采用 RISC 体系结构。早期的单片机大多采用 CISC 结构体系,指令复杂,指令代码、周期数不统一;指令运行很难实现流水线操作,大大阻碍了运行速度的提高。如 MCS-51 系列单片机,当外部时钟为 12MHz 时,其单周期指令运行速度也仅为 1MIPS。采用 RISC 体系结构和精简指令后,单片机的指令绝大部分成为单周期指令,而通过增加程序存储器的宽度(如从 8 位增加到 16 位),实现了一个地址单元存放一条指令。在这种体系结构中,很容易实现并行流水线操作,大大提高了指令运行速度。目前一些 RISC 结构的单片机,如美国 ATMEL 公司的 AVR 系列单片机已实现了一个时钟周期执行一条指令。与 MCS-51 相比,在相同的 12MHz 外部时钟下,单周期指令运行速度可达 12MIPS。一方面可获得很高的指令运行速度,另一方面,在相同的运行速度下,可大大降低时钟频率,有利于获得良好的电磁兼容效果。

③ 片内存储器的改进与发展。目前新型的单片机一般在片内集成两种类型的存储器:随机读写存储器 SRAM,作为临时数据存储器存放工作数据用;只读存储器 ROM,作为程

序存储器存放系统控制程序和固定不变的数据。片内存储器的改进与发展的方向是扩大容量、ROM 数据的易写和保密等。

片内存储容量的增加。新型的单片机一般在片内集成的 SRAM 为 128B~1KB,ROM 的容量一般为 4~8KB。为了适应网络、音视频等高端产品的需要,高档的单片机在片内集成了更大容量的 RAM 和 ROM 存储器。如 ATMEEL 公司的 ATmega16,片内的 SRAM 为 1KB,FlashROM 为 16KB。而该系列的高端产品 ATmega256,片内集成了 8KB 的 SRAM,256KB 的 FlashROM 和 4KB 的 EEPROM。

片内程序存储器由 EPROM 型向 FlashROM 发展。早期的单片机在片内往往没有程序存储器或片内集成 EPROM 型的程序存储器。将程序存储器集成在单片机内可以大大提高单片机的抗干扰性能、提高程序的保密性、减少硬件设计的复杂性和空间等许多优点,因此片内集成程序存储器已成为新型单片机的标准方式。但由于 EPROM 需要使用 12V 高电压编程写入、紫外线光照擦除、重写入次数有限等缺点,给使用带来了不便。新型的单片机则采用 FlashROM 以及 MaskROM、OTPROM 作为片内的程序存储器。FlashROM 在通常电压(如 5V/3V)下就可以实现编程写入和擦除操作,重写次数在 10 000 次以上,并可实现在线编程写入 ISP 技术的优点,为使用带来了极大的方便。采用 MaskROM 的微控制器称为掩模芯片,它是在芯片制造过程中就将程序“写入”了,并永远不能改写。采用 OTPROM 的微控制器,其芯片出厂时片内的程序存储器是“空的”,它允许用户将自己编写好的程序一次性地编程写入,之后便再也无法修改了。后两种类型的单片机适合于大批量产品生产的使用,而前两种类型的微控制器则适合产品的设计开发、批量生产以及学习培训的应用。

④ 多核处理方式的采用,现在有的单片机已采用所谓的三核(TrCore)结构。这是一种建立在系统级芯片(System on a chip)概念上的结构。这种单片机由三个核组成:一个是微控制器和 DSP 核,一个是数据和程序存储器核,最后一个外围专用集成电路(ASIC)。这种单片机的最大特点在于把 DSP 和微控制器同时做在一个片上。虽然从结构定义上讲,DSP 是单片机的一种类型,但其作用主要反映在高速计算和特殊处理如快速傅里叶变换等上面。把它和传统单片机结合集成大大提高了单片机的功能。这是目前单片机最大的进步之一。这种单片机最典型的有 Infineon 公司的 TC10GP; Hitachi 公司的 SH7410,SH7612 等。这些单片机都是高档单片机,MCU 都是 32 位的,而 DSP 采用 16 或 32 位结构,工作频率一般在 60MHz 以上。

#### (2) 功耗、封装及电源电压的进步。

现在新的单片机的功耗越来越小,特别是很多单片机都设置了多种工作方式,这些工作方式包括等待、暂停、睡眠、空闲、节电等。Philips 公司的单片机 P87LPC762 是一个很典型的例子,在空闲时,其工作电流为 1.5mA,而在节电方式中,其工作电流只有 0.5mA。在工作电流上最令人惊叹的是 TI 公司的单片机 MSP430 系列,它是一个 16 位的系列,有超低功耗工作方式。它的低功耗方式有 LPM1、LPM3、LPM4 三种。当电源为 3V 时,如果工作于 LMP1 方式,即使外围电路处于活动,由于 CPU 不活动,振荡器处于 1~4MHz,这时工作电流只有 50mA。在 LPM3 时,振荡器处于 32kHz,这时工作电流只有 1.3mA。在 LPM4 时,CPU、外围及振荡器 32kHz 都不活动,则工作电流只有 0.1mA。

现在单片机的封装水平已大大提高,随着贴片工艺的出现,单片机也大量采用了各种符



合贴片工艺的封装方式,以减少体积。在这种形势中, Microchip 公司推出的 8 引脚的单片机特别引人注目。这是 PIC12CXXX 系列。它含有 0.5~2KB 程序存储器, 25~128B 数据存储器, 6 个 I/O 端口以及一个定时器, 有的还含 4 道 A/D, 完全可以满足一些低档系统的应用。扩大电源电压范围以及在较低电压下仍然能工作是今天单片机发展的目标之一。目前, 一般单片机都可以在 3.3~5.5V 条件下工作。而一些厂家, 则生产出可以在 2.2~6V 条件下工作的单片机。这些单片机有 Fujitsu 公司的 MB89191~89195, MB89121~125A, MB89130 系列等, 应该说该公司的 F2MC-8L 系列单片机绝大多数都满足 2.2~6V 的工作电压条件。而 TI 公司的 MSP430X11X 系列的工作电压也是低到 2.2V 的。

### (3) 工艺上的进步。

现在的单片机基本上采用 CMOS 技术, 但已经大多数采用了 0.6 $\mu\text{m}$  以上的光刻工艺, 有个别的公司, 如 Motorola 公司则已采用 0.35 $\mu\text{m}$  甚至是 0.25 $\mu\text{m}$  技术。这些技术的进步大大地提高了单片机的内部密度和可靠性。

## 2. 以单片机为核心的嵌入式系统

单片机的另外一个名称就是嵌入式微控制器, 原因在于它可以嵌入到任何微型或小型仪器或设备中。目前, 把单片机嵌入式系统和 Internet 连接已是一种趋势。但是, Internet 一向是一种采用胖服务器, 瘦用户机的技术。这种技术在互联网上存储及访问大量数据是合适的, 但对于控制嵌入式器件就成了“杀鸡用牛刀”了。要实现嵌入式设备和 Internet 连接, 就需要把传统的 Internet 理论和嵌入式设备的实践都颠倒过来。为了使复杂的或简单的嵌入式设备, 例如单片机控制的机床、单片机控制的门锁, 能切实可行地和 Internet 连接, 就要求专门为嵌入式微控制器设备设计网络服务器, 使嵌入式设备可以和 Internet 相连, 并通过标准网络浏览器进行过程控制。

目前, 为了把以单片机为核心的嵌入式系统和 Internet 相连, 已有多家公司在进行这方面的较多研究。这方面较为典型的有 emWare 公司和 TASKING 公司。

emWare 公司提出嵌入式系统入网的方案——EMIT 技术。这个技术包括三个主要部分: 即 emMicro, emGateway 和网络浏览器。其中, emMicro 是嵌入设备中的一个只占内存容量 1KB 的极小的网络服务器; emGateway 作为一个功能较强的用户或服务器, 用于实现对多个嵌入式设备的管理, 还有标准的 Internet 通信接入以及网络浏览器的支持。网络浏览器使用 emObjects 进行显示和嵌入式设备之间的数据传输。

如果嵌入式设备的资源足够, 则 emMicro 和 emGateway 可以同时装入嵌入式设备中, 实现 Internet 的直接接入。否则, 将要求 emGateway 和网络浏览器相互配合。emWare 的 EMIT 软件技术使用标准的 Internet 协议对 8 位和 16 位嵌入式设备进行管理, 但比传统上的开销小得多。

目前, 单片机应用中提出了一个新的问题: 如何使 8 位、16 位单片机控制的产品, 也即嵌入式产品或设备能实现和互联网互连?

TASKING 公司目前已经为解决这个问题提供了途径。为方便用户开发, 该公司已把 emWare 的 EMIT 软件包和有关的软件配套集成, 形成一个集成开发环境。嵌入互联网联盟 (Embed the Internet Consortium, EIT) 正在与其紧密合作, 共同开发嵌入式 Internet 的解决方案。在不久将会有成果公布。