

随书附赠 Freescale S12 工具包

# 单片机嵌入式应用的 在线开发方法

邵贝贝 著



清华大学出版社



随书附赠 Freescale S12 工具包

# 单片机嵌入式应用的 在线开发方法

邵贝贝 著

清华大学出版社

北京

## 内 容 简 介

本书是讲述单片机在线开发方法的一本专著，也是清华大学研究生精品课“嵌入式实时系统与单片机应用”的教材。

本书介绍如何从最小硬件系统设计开始，建立开发单片机应用系统的基本硬件条件，而不必使用仿真器等单片机开发工具；如何写单片机监控程序建立单片机应用软件调试的基本环境；如何用 C 语言编写应用程序；如何利用商用软件 CodeWarrior 的教学版本，或使用自由软件 Linux 下的 GCC 交叉 C 编译器，将嵌入式实时多任务操作系统  $\mu\text{C}/\text{OS-II}$  移植到单片机上，实现基于 RTOS 的开发环境。

本书以 16 位单片机 MC9S12 为例，给出监控程序源代码，还提供了单片机各 I/O 模块的设备驱动程序。这种在线开发方法同样适用于 8 位、32 位单片机。

随书所附光盘中有免费软件开发工具和大量原文资料。

本书为造就 IT 高手指路，也供开发单片机应用系统的工程技术人员参考。

版权所有，翻印必究。举报电话：010-62782989 13901104297 13801310933

本书封面贴有清华大学出版社激光防伪标签，无标签者不得销售。

### 图书在版编目 (CIP) 数据

单片机嵌入式应用的在线开发方法/邵贝贝著. —北京：清华大学出版社，2004.10  
ISBN 7-302-09658-9

I. 单… II. 邵… III. 单片微型计算机—系统开发 IV. TP368.1

中国版本图书馆 CIP 数据核字 (2004) 第 100766 号

出 版 者：清华大学出版社

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

社 总 机：010-62770175

地 址：北京清华大学学研大厦

邮 编：100084

客户服务：010-62776969

组稿编辑：张 龙

文稿编辑：汪汉友

封面设计：华 森

印 刷 者：北京国马印刷厂

装 订 者：三河市新茂装订有限责任公司

发 行 者：新华书店总店北京发行所

开 本：185×260 印张：27.5 字数：650 千字

版 次：2004 年 10 月第 1 版 2004 年 10 月第 1 次印刷

书 号：ISBN 7-302-09658-9/TP·6692

印 数：1~5000

定 价：39.00 元 (含光盘)

---

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系调换。联系电话：(010) 62770175-3103 或 (010) 62795704

# 前 言

本书是讲述单片机开发方法的一本专著，讲的是单片机的在线开发方法，这种方法是我们在多年来开发单片机应用的主要方法——它强调的是在线开发和不使用仿真器。这本书也是清华大学研究生精品课“嵌入式实时系统与单片机应用”教材的单片机部分，实时操作系统部分采用《嵌入式实时操作系统  $\mu\text{C}/\text{OS-II}$ （第2版）》一书。因为最初的  $\mu\text{C}/\text{OS}$  就是为 Motorola 的 MC68HC11 单片机写的，故本书采用和其源码兼容的替代产品——S12 系列单片机，来实现  $\mu\text{C}/\text{OS-II}$  的运行。这本书也是为嵌入式应用开发工程师写的，由于 16 位单片机比 8 位机要复杂很多，所以对那些急于开发产品而又不熟悉 16 位单片机的工程师，采用我们提供的开发工具，尽快把目标系统开发环境搭建起来，边开发、边学习，是多快好省的办法。本教材也可供其他工科院校电类本科生学习使用。

本书第 1 章简单介绍令人眼花缭乱的各类单片机，目的在于强调单片机应用是个性化的，用户针对不同的应用，一定要选择最合适的单片机。第 2 章以 16 位单片机为例，给出在线开发单片机的一般方法；从设计单片机最小硬件系统开始，实现人与单片机通过串行口对话，尽快让单片机“活”起来。第 3 章和第 4 章介绍单片机汇编语言和如何写一个简单的监控程序，从而实现单片机的在线调试。第 5 章讲如何在单片机上建立 C 语言程序的运行环境，C 语言和汇编语言是怎么接口的。第 6 章讲如何将实时操作系统内核  $\mu\text{C}/\text{OS-II}$  移植到单片机上。第 7 章介绍如何用自由软件 GCC 在 PC 上交叉编译 C 程序，生成单片机的应用代码，从学习使用 Linux 下的 GCC 入门，意在指出一条通向 IT 行业高手之路。第 8 章讲解商用软件集成开发环境的使用，用的是网上提供的免费教学版本，这一章还介绍了 S12 单片机的直接在线调试方法，即后台调试法(Background Debug Mode)。第 9 章简单介绍单片机相关的 I/O 模块，给出基本驱动程序。第 10 章通过一个实例，给出单片机模糊控制原理的入门知识，因为模糊控制指令也是 S12 单片机特有的，故特别加了这一章。第 11 章讲述单片机系统硬件设计应注意的电磁兼容问题。

在附录 A 中，还给出相关开发工具的使用方法，并将自主开发的监控程序源代码在附录 B 中给出，这个代码已经在开发板上检验、运行过，读者可以直接拿来使用。附录 C 和附录 D 是为便于用户开发中查阅的指令表。

更详尽的原文手册和 C 的交叉编译软件如 GCC 软件等也都附在光盘上。另外，光盘上还有大量从网上下载的应用范例。本书所附光盘给出的工具与资料如下：

1. 可以从网站上选择到的有关 S12 单片机的原文资料；
2. Metrowerks 公司提供的 CodeWarrior 特殊版本 C 交叉编译软件和我们的工程范例；
3. 用于 S12 的自由软件：GNU 提供的 GCC 交叉编译器软件；
4. 可直接下载到目标板上的 S19 格式的 S12 单片机监控程序。

本教材中用到的单片机实验开发环境可以自己做，并全部使用自由软件，也可以全都购买；或买一些、自己做一些。

感谢 Motorola 半导体对我校教学与科研的长期支持，特别是提供了 S12BDM 调试工具的制造技术。感谢我的研究生刘永毅、龚光华在研制 MC9S12DP256/DG128 开发板、BDM 调试工具方面做出的贡献。本书中对  $\mu\text{C}/\text{OS-II}$  的移植是我的研究生邵森龙完成的，目标板设计范例是研究生陈萌萌提供的。另外，他还提供了本书的第 7 章和第 9 章。

本书写成时，正逢 Motorola 半导体部从原公司分离出来成立飞思卡尔 (Freescale) 独立上市公司，以求更好的发展。无论将来采用何种命名法，单片机技术是继承和发展的。

需求在增长，技术在发展，社会对人才的需求是无止境的，谨求本教材的思路能得到同行们的认可，不当之处请读者指正。关于 S12 单片机开发工具和今后的发展，请访问我们的网站：[www.tsinghua-mot.com](http://www.tsinghua-mot.com)。

邵贝贝  
于清华大学

# 目 录

引言.....	1
<b>第 1 章 色彩斑斓的单片机世界.....</b>	<b>6</b>
1.1 单片机世界.....	6
1.2 开发单片机应用不再需要仿真器.....	8
1.3 Motorola (Freescale) 单片机.....	8
1.4 MC68HC08 系列单片机.....	10
1.4.1 Nitron 系列单片机.....	10
1.4.2 面向低端产品的 8 位单片机.....	10
1.4.3 通用型 8 位单片机.....	10
1.4.4 带 LCD (LED) 驱动接口的 8 位单片机.....	11
1.4.5 用于电动机控制的 8 位单片机.....	12
1.4.6 带 CAN 总线接口的 8 位单片机.....	12
1.4.7 支持 LIN 总线的 8 位单片机.....	12
1.4.8 带 USB 接口的 8 位单片机.....	13
1.4.9 用于彩色液晶监视器控制的 8 位单片机.....	13
1.4.10 带无线通信功能的 8 位单片机.....	14
1.5 MCS08 系列 8 位单片机.....	14
1.6 MC68HC11 系列单片机.....	14
1.7 MC68HC12 系列单片机.....	17
1.8 MC9S12 系列单片机.....	18
1.8.1 MC9S12A 系列和 B 系列 16 位单片机.....	18
1.8.2 带 CAN 总线的 MC9S12D 系列 16 位单片机.....	19
1.8.3 MC9S12DP256 单片机.....	20
1.8.4 带液晶驱动的 16 位单片机系列.....	22
1.8.5 低供电电压的 16 位单片机.....	23
1.8.6 带 USB 接口的 16 位单片机.....	24
1.8.7 带以太网接口的 16 位单片机.....	24
1.9 MC68HC16 系列单片机.....	24
1.10 Motorola (Freescale) 的 32 位单片机.....	26
1.10.1 以 68K、CPU32 为 CPU 的 32 位单片机.....	26
1.10.2 以 ColdFire 为 CPU 的 32 位单片机.....	27
1.10.3 用于控制的以 PowerPC 为 CPU 的 32 位单片机.....	28

1.10.4	用于通信的以 PowerPC 为 CPU 的 32 位单片机 .....	29
1.10.5	Motorola (Freescale) 的 ARM 系列 32 位单片机 .....	30
1.11	Motorola (Freescale) 的 DSP 型单片机 .....	31
1.11.1	普通 16 位 DSP 型单片机 .....	31
1.11.2	使用增强型内核 DSP56800E 的 16 位 DSP 型单片机 .....	35
1.11.3	StarCore 系列 DSP 型 16 位单片机 .....	36
1.11.4	24 位、32 位的 DSP 型单片机 .....	36
<b>第 2 章</b>	<b>单片机基本系统的硬件设计 .....</b>	<b>39</b>
2.1	16 位单片机 .....	39
2.1.1	MC9S12 系列单片机 .....	40
2.1.2	MC9S12DP256 单片机 .....	40
2.2	初识单片机最小硬件系统 .....	43
2.2.1	16 位单片机 MC9S12 的最小系统 .....	45
2.2.2	8 位单片机 MC68HC08GP32 的最小系统 .....	45
2.2.3	32 位单片机 MC68332 的最小系统 .....	46
2.2.4	监控程序 .....	47
2.2.5	体验机器码 .....	50
2.3	异步串行通信 .....	52
2.3.1	串行通信协议 RS-232 标准 .....	52
2.3.2	ASCII 码 .....	53
2.3.3	串行数据格式 .....	54
2.3.4	RS-232-C 电缆的连接方法 .....	55
2.3.5	通信速率 .....	55
2.4	MC9S12 单片机系统的硬件设计 .....	56
2.4.1	时钟电路 .....	56
2.4.2	串行口的 RS-232 驱动电路 .....	59
2.4.3	电源电路 .....	59
2.4.4	复位电路 .....	60
2.4.5	BDM 接口 .....	60
2.4.6	单片机并行口及驱动能力 .....	60
2.4.7	调试显示 .....	61
2.5	运行模式 .....	61
2.5.1	单片运行模式 .....	61
2.5.2	扩展运行模式 .....	61
2.6	BDM 调试器及硬件设计 .....	62
2.6.1	BDM 调试器 .....	62
2.6.2	BDM 调试器硬件设计 .....	63

<b>第3章 用汇编语言编程</b> .....	65
3.1 CPU 的内部寄存器结构 .....	65
3.1.1 16位单片机 HC11/12 的 CPU 内部结构 .....	65
3.1.2 8位单片机 HC08 的 CPU 内部结构 .....	66
3.1.3 32位单片机 68K/ColdFire 的 CPU 内部结构 .....	67
3.2 内存空间分配 .....	68
3.2.1 基本内存空间分配 .....	68
3.2.2 内存空间的扩展 .....	69
3.3 汇编指令集 .....	71
3.4 指令按功能分类 .....	71
3.4.1 数据传送指令 .....	72
3.4.2 算术与逻辑运算指令 .....	73
3.4.3 程序控制指令 .....	77
3.5 CPU12 的模糊逻辑指令 .....	79
3.5.1 模糊化指令 .....	79
3.5.2 模糊推理指令 REV 和 REVW .....	80
3.5.3 反模糊化指令 WAV .....	80
3.6 指令按寻址方式分类 .....	81
3.6.1 隐含寻址 .....	81
3.6.2 立即数寻址 .....	81
3.6.3 直接寻址 .....	81
3.6.4 扩展寻址 .....	82
3.6.5 变址寻址 .....	82
3.6.6 带自动加、减 5 位偏移量的间接寻址 .....	83
3.6.7 相对寻址 .....	83
3.7 汇编指令表 .....	83
3.8 指令的机器码组织 .....	85
3.9 用汇编语言编写程序 .....	86
3.9.1 汇编程序的格式 .....	86
3.9.2 汇编管理指令 .....	86
3.10 汇编语言程序设计举例 .....	88
3.10.1 检查 SCI 输入端口状态 .....	88
3.10.2 输入一个字符 .....	88
3.10.3 输出一个字符 .....	89
3.10.4 输出空格 .....	89
3.10.5 显示字符串 .....	90
3.10.6 输入并显示字符 .....	90
3.11 码的转换类子程序 .....	90



3.11.1	输入一个十六进制数.....	90
3.11.2	输入一个字节.....	91
3.11.3	输入两个字节的十六进制数.....	91
3.11.4	输出两个字节的十六进制数.....	92
3.12	汇编语言编程技巧.....	92
3.13	用汇编语言写 BDM 调试命令.....	94
3.13.1	后台调试模式.....	94
3.13.2	进入 BDM 模式.....	95
3.13.3	BDM 通信协议.....	96
3.13.4	BDM 命令.....	98
3.13.5	BDM 的 ROM.....	99
<b>第 4 章</b>	<b>建立单片机运行环境和写监控程序.....</b>	<b>100</b>
4.1	建立单片机运行环境.....	100
4.2	堆栈指针初始化.....	100
4.3	时钟初始化.....	101
4.3.1	找出与时钟相关的寄存器.....	101
4.3.2	初始化时钟系统.....	104
4.4	串行口初始化.....	105
4.4.1	与串行口有关的寄存器.....	105
4.4.2	初始化串行口.....	108
4.5	监控程序.....	109
4.6	人机对话.....	111
4.7	命令字与跳转表.....	114
4.8	建立 CPU 在内存中的固定映像.....	115
4.8.1	CPU 响应中断后的栈结构.....	115
4.8.2	CPU 寄存器的固定映像.....	116
4.9	执行程序.....	117
4.10	显示与修改内存.....	118
4.10.1	显示内存.....	118
4.10.2	修改内存.....	119
4.11	显示和修改 CPU 寄存器.....	120
4.12	向 RAM 下载程序.....	122
4.12.1	数据文件格式.....	123
4.12.2	下载程序.....	124
4.13	Flash 的擦除与写入.....	126
4.13.1	与 Flash 有关的寄存器.....	126
4.13.2	Flash 擦除与写入的步骤.....	127

4.14	设置断点.....	134
4.14.1	在 RAM 程序中设置断点.....	134
4.14.2	在 Flash 程序中设置断点.....	138
4.15	中断向量表管理.....	139
4.16	系统调用表.....	142
4.17	帮助信息.....	143
<b>第 5 章</b>	<b>用 C 语言开发应用程序.....</b>	<b>144</b>
5.1	C 语言是开发单片机应用软件的有力工具.....	144
5.2	开发嵌入式应用的 C 编译器的特点.....	145
5.3	交叉编译和 C 语言程序运行环境的建立.....	147
5.3.1	应用程序的构成与模块化程序结构.....	150
5.3.2	全局变量与局部变量.....	151
5.3.3	函数的结构与函数间参数的传递.....	152
5.3.4	C 语言中的 I/O 语句.....	154
5.3.5	程序模块的框架与组织.....	155
5.3.6	程序的链接与定位.....	156
5.4	交叉 C 编译器及用 C 语言扩展监控程序.....	157
5.4.1	EEPROM.....	157
5.4.2	EEPROM 擦除和编程步骤.....	158
5.4.3	EEPROM 编程命令字及其含义.....	159
5.4.4	EEPROM 的写保护区设定.....	159
5.5	嵌入式应用中的 I/O.....	164
5.5.1	关于“Hello, World”.....	164
5.5.2	自己写 printf() 函数.....	164
<b>第 6 章</b>	<b>使用嵌入式实时操作系统.....</b>	<b>175</b>
6.1	嵌入式实时操作系统.....	175
6.1.1	嵌入式实时操作系统简介.....	175
6.1.2	嵌入式实时操作系统 $\mu$ C/OS-II.....	175
6.2	移植 $\mu$ C/OS-II.....	177
6.2.1	重新定义内核的大小和功能.....	178
6.2.2	OS_CPU.H.....	181
6.3	编写与硬件相关的代码.....	183
6.3.1	中断服务子程序 OSTickISR().....	184
6.3.2	任务堆栈初始化函数 OSTaskStkInit().....	185
6.3.3	让优先级最高的就绪态任务开始运行 OSStartHighRdy().....	187
6.3.4	任务级任务切换函数 OSCtxSw().....	187
6.3.5	中断级任务切换函数 OSIntCtxSw().....	188

6.3.6	相关接口函数.....	190
6.4	产生时钟节拍中断.....	191
6.5	制作用户自己的项目.....	192
6.5.1	main.h.....	192
6.5.2	main.c.....	194
6.5.3	TaskStart.c.....	195
6.5.4	task1.c 和 task2.c.....	196
6.5.5	hardware.c.....	197
6.5.6	userlib.c.....	198
6.5.7	链接与程序定位.....	198
6.6	估算 $\mu$ C/OS-II 占用的 RAM 资源.....	200
6.6.1	$\mu$ C/OS-II 中的全局变量.....	201
6.6.2	任务控制块.....	202
6.6.3	事件控制块.....	204
6.6.4	任务堆栈.....	205
6.6.5	估算内核占用 RAM 空间举例.....	207
6.7	多任务下的设备驱动.....	208
6.7.1	重新认识异步串行口.....	208
6.7.2	SCI 的中断.....	209
6.7.3	用中断方式接收.....	209
6.7.4	用中断方式发送.....	210
<b>第 7 章</b>	<b>使用 GCC 交叉编译器开发 HC/S12 单片机.....</b>	<b>213</b>
7.1	使用免费的 GCC 交叉编译器.....	213
7.2	用 GCC 开发 HC/S12 系列单片机.....	214
7.2.1	安装“GCC for HCS12”.....	216
7.2.2	使用 GNU 针对 MC68HC11/MC68HC12 的开发环境.....	218
7.2.3	使用范例程序库进行交叉编译.....	221
7.3	如何编写 makefile 文件.....	223
7.3.1	简单 makefile 的书写规则.....	224
7.3.2	make 命令的使用.....	225
7.4	GCC for HCS12 编译器.....	226
7.4.1	内嵌汇编语言.....	229
7.4.2	陷阱、软中断和中断.....	229
7.4.3	填写中断向量表.....	230
7.4.4	支持页面 Flash ROM.....	231
7.4.5	编译参数.....	232
7.4.6	预处理参数.....	234

7.4.7	汇编参数.....	234
7.4.8	链接参数.....	235
7.4.9	链接器.....	235
7.5	hello world 工程应用范例.....	239
7.5.1	源文件描述.....	239
7.5.2	链接地址描述.....	241
7.5.3	编译并运行.....	241
<b>第 8 章</b>	<b>单片机软件开发工具.....</b>	<b>243</b>
8.1	商用软件开发工具“CodeWarrior for HCS12”.....	243
8.2	安装 CodeWarrior.....	243
8.3	建立工程.....	244
8.3.1	使用 C 编译器.....	246
8.3.2	使用汇编器.....	246
8.3.3	增加新程序模块.....	248
8.4	编写应用程序 main.c.....	248
8.4.1	main.c.....	248
8.4.2	定义装载地址.....	250
8.4.3	利用 make 命令编译.....	251
8.5	在目标板上运行程序.....	252
8.5.1	向目标机下载程序.....	252
8.5.2	运行程序.....	253
8.6	建立自己的 C 程序运行环境.....	255
8.7	编译基于 $\mu$ C/OS-II 的应用程序.....	255
<b>第 9 章</b>	<b>实验系统与 I/O 模块.....</b>	<b>261</b>
9.1	通用 I/O 接口.....	262
9.2	MC9S12DP256 的片内总线接口.....	264
9.2.1	同步串行接口.....	264
9.2.2	I <sup>2</sup> C 总线接口.....	272
9.2.3	CAN 总线接口.....	276
9.2.4	Motorola S12 系列单片机的 MSCAN 模块.....	281
9.2.5	CAN 总线实验原理及通信程序设计.....	283
9.3	增强型定时器.....	285
9.3.1	输入捕捉/输出比较.....	286
9.3.2	输入捕捉/输出比较通道.....	287
9.3.3	8 位脉冲累加器.....	287
9.3.4	模数计数器.....	288
9.4	PWM 模块.....	288

9.4.1	PWM 模块概述 .....	288
9.4.2	PWM 波用作 D/A 转换接口 .....	288
9.5	A/D 模块 .....	289
9.6	Motorola 16 位单片机 MC9S12DP256/DG128 教学实验系统 .....	291
9.6.1	概述 .....	291
9.6.2	硬件电路介绍 .....	291
<b>第 10 章</b>	<b>单片机模糊控制 .....</b>	<b>297</b>
10.1	模糊控制 .....	297
10.2	模糊控制指令 .....	298
10.2.1	模糊控制专用指令 .....	298
10.2.2	模糊控制相关指令 .....	298
10.3	模糊逻辑的基本概念 .....	298
10.3.1	模糊集合及隶属度函数 .....	298
10.3.2	模糊逻辑与模糊变量 .....	300
10.3.3	模糊推理 .....	301
10.4	模糊控制原理与模糊控制器结构 .....	301
10.4.1	模糊化 .....	303
10.4.2	模糊规则推理 .....	304
10.4.3	反模糊化 .....	307
10.4.4	模糊控制的实现 .....	309
10.4.5	模糊控制器的设计与调整 .....	310
10.5	模糊控制开发软件 .....	313
10.5.1	Motorola 模糊推理机 .....	313
10.5.2	MC68HC11 模糊推理机 .....	313
10.5.3	S12 逻辑推理机程序 .....	320
<b>第 11 章</b>	<b>单片机应用中的电磁兼容问题 .....</b>	<b>323</b>
11.1	电磁兼容基本概念 .....	323
11.2	电磁兼容组织与标准 .....	324
11.3	单片机系统的电磁兼容问题 .....	326
11.4	噪声的来源与传输 .....	327
11.4.1	信号线间交叉干扰 .....	329
11.4.2	来自电源的噪声 .....	330
11.5	印刷电路板 EMC 设计 .....	331
11.5.1	元件的布置 .....	331
11.5.2	印刷电路板接地线的处理 .....	332
11.5.3	多层板设计 .....	333
11.6	常用抗干扰器件 .....	334

11.6.1	去耦电容.....	334
11.6.2	磁性元件.....	335
11.6.3	低通滤波器.....	336
11.6.4	瞬变干扰吸收器件.....	336
11.7	印刷电路板设计中控制噪声的经验.....	337
11.7.1	控制噪声源.....	337
11.7.2	减小噪声的耦合.....	338
11.7.3	减小噪声接收.....	338
11.8	单片机自身的抗干扰措施.....	339
11.8.1	降低外时钟的频率.....	339
11.8.2	时钟监控电路.....	339
11.8.3	打开“看门狗”电路.....	340
11.8.4	电源电压监控.....	340
11.8.5	非法指令中断和剩余程序区处理.....	340
附录 A	MC9S12D 系列单片机开发工具包.....	342
A.1	概述.....	342
A.1.1	HCS12D 系列单片机.....	342
A.1.2	HCS12 开发工具包组件.....	342
A.2	MC9S12DP256/DG128 开发板及与 PC 通信.....	343
A.2.1	MC9S12DP256/DG128 开发板.....	343
A.2.2	开发板上的跳线.....	345
A.2.3	开发板的硬件连接.....	345
A.2.4	PC 的设置.....	346
A.3	监控程序及监控命令详解.....	349
A.3.1	命令详解.....	349
A.3.2	改变波特率.....	353
A.3.3	复位、中断向量表.....	355
A.3.4	用户可以使用的 RAM 空间.....	355
A.3.5	系统调用.....	355
A.4	编译器 CodeWarrior for HCS12 使用方法入门.....	356
A.4.1	建立工程文件.....	356
A.4.2	编写 main.c 程序.....	357
A.4.3	定义存储空间分配.....	359
A.4.4	应用程序的编译.....	360
A.4.5	向开发板下载程序.....	361
A.4.6	运行应用程序.....	362
A.5	BDM 调试器及使用方法.....	363

A.5.1	BDM 调试器使用方法.....	364
A.5.2	设置 BDM 头的时钟频率.....	364
A.5.3	和目标板连接.....	365
A.5.4	Help 命令.....	366
A.5.5	向 Flash 下载程序.....	367
A.5.6	擦除目标板的片内 Flash.....	368
A.6	常用 BDM 调试命令.....	368
A.6.1	汇编和反汇编命令.....	368
A.6.2	控制目标 CPU 运行的命令.....	370
A.6.3	显示和修改 CPU 寄存器的命令.....	370
A.6.4	显示和修改内存.....	370
A.6.5	以 S 格式读出目标代码.....	370
A.6.6	其他 BDM 命令.....	370
附录 B	监控程序源代码.....	372
附录 C	CPU12 汇编指令表.....	406
附录 D	CPU12 指令机器码表.....	420
附录 E	本书所附光盘说明.....	423
参考文献	.....	424

# 引 言

本书是为清华大学“嵌入式实时系统与单片机应用”课程撰写的教材，讲述的是单片机开发方法。这里，笔者希望就单片机教学的历史与未来趋势、课程的定位、科研与教学、授课与实验的关系等提出一些思考和拙见，与同行们讨论。本课程面向非计算机专业的研究生，特别是那些可能在研究生课题中用到单片机的各专业学生。本书讲述的是单片机应用部分，实时操作系统部分则采用《嵌入式实时操作系统  $\mu\text{C}/\text{OS-II}$ （第2版）》<sup>①</sup>。这两部分内容共安排课内 64 学时，其中授课与教师指导下的实验各占约一半比例。按校方要求，这类课程原则上课内、课外学时之比为 1:2，即学生课外要花 128 学时读书和进行无辅导下的实验。

本书又是培养 IT 高手的入门教材，当然也是为嵌入式应用开发的工程师们写的。16 位单片机比 8 位机要复杂很多，对那些仅熟悉 8051 单片机，用惯了仿真器开发方法的工程师，这里介绍的在线开发方法有一定的难度。但笔者相信，一旦掌握了开发环境的搭建过程，就不会再想回到仿真器的老路上去了。如同掌握了嵌入式实时操作系统，就不会再想回到前后台系统的老路上。本教材也可供其他工科院校的电类本科生学习使用，但建议增加一些学时。

由于听我讲课的清华研究生来自不同学科、不同专业，在单片机应用方面水平很不一样。计算机技术的发展是由各行各业需求拉动的，而计算机技术的发展又促进了需求；这里，需求是矛盾的主要方面。由于物理，特别是高能物理对计算机的需求几乎是无限的，故像互联网这样的计算机技术最先源于物理学的需求。工程物理系核电子学专业毕业的一些本科生，单片机应用水平是相当高的，他们在大学 2 年级就有 32 学时的“单片机认识与实践”课程，讲的是简单 8 位单片机的入门知识，当时他们并没有学过数字电路之类的课程。接下来有的同学在进行进一步学习的基础上做一个单片机应用方面的 SRT 项目（Student Research and Training Program），清华大学对 SRT 有专门的经费支持。学生设想一个单片机应用的原理性演示题目，花半年到一年的课余时间，设计一块印刷电路板、焊上元器件，再把软件调出来。优秀作品还可拿到每年校庆的学生科技展上评奖，甚至参加全国大学生“挑战杯”的角逐。这类学生毕业设计中一般会做一个单片机应用方面的课题。经过这样的训练，学生到研究生时可以很快进入科研课题，并且可以帮助我当辅导实验的助教。而个别研究生，特别是外校考入我校的非电类研究生，甚至没有学过微机原理之类课程。

这里只能假设选修这门课的同学都有数字电路课的基础，会用 C 语言写程序。由于水平的参差不齐及学生个人可能安排的课外学时的差异，到课程结束时，每个人到达的水平也是很不一样的，故只能是各按步伐、共同前进。这里，强调的是入门，只要引导学生入

---

<sup>①</sup> Jean J. Labrosse 著. 嵌入式实时操作系统  $\mu\text{C}/\text{OS-II}$ （第2版）. 邵贝贝等译. 北京：北京航空航天大学出版社，2003



门，告诉他们什么是正确的方法，后面的路他们会自己走。

当前，单片机与嵌入式系统应用类课程正处在转型期。过去 20 年中，单片机市场与需求都是以 8 位机为主。最近几年来，16 位、32 位单片机的嵌入式应用呈迅速增长的趋势，一两年后，32 位单片机在产值上会超过 8 位单片机。今后，单片机应用将呈现 8 位、16 位、32 位单片机并举的格局。回忆过去我国单片机教学走过的路，大多数院校都是从 8051 走过来的，故 8051 在国内普及单片机应用方面功不可没，而总结一下由 8051 教学模式带来的副作用将有利于目前的转型。

教学中以一片 8051 配以不同的外围电路去适应各种不同的应用，这就忽略了单片机应用的多样性和单片机应用的个性化定位。单片机姓“单”，世界上的单片机至少有上千种，每种单片机都有各自的定位，强调的是 SOC，即片上系统，只有将整个系统都集成在一个集成电路芯片上，才能做到低成本、高可靠性。只有在不得已的情况下，才扩充额外的接口芯片。故应告诉学生，虽然学的是某种单片机，但针对不同的应用要选择最适宜的单片机。本书第 1 章介绍五彩缤纷的单片机精彩世界，目的在于强调单片机应用是个性化的，有助于扩大学生单片机选型方面的思路；用户针对不同的应用，一定要选择最合适的单片机。然后以 16 位单片机为例，给出在线开发单片机的一般方法，单片机教学要求教会学生一种方法，会用这种单片机，自然也会用其他单片机。

过去 8051 教学引入的另一误区是使用仿真器，如果说过去开发掩膜型单片机使用仿真器属不得已，则如今单片机都采用了可以反复擦写 10 万次以上的 Flash 存储器，实在没有必要使用仿真器。本人在过去 20 年的单片机应用开发中就从来没有用过仿真器，并非没有仿真器可用，只是不主张使用仿真器，特别不主张使用仿真器教学。仿真器使单片机开发变得不直观，高价的仿真器增加了开发成本，也增加了教学的负担。开发者不可能为每一类单片机买一种仿真器，只好以自己有仿真器的那种单片机去适应所有的应用，从而误入歧途。关于单片机的调试方法，笔者始终认为，监控程序法是单片机调试的经典方法。本教材通过讲授如何设计单片机最小硬件系统，如何写监控程序，并辅之以实验的过程，使学生掌握开发单片机的基本方法，从而对单片机有深层次的理解。

教学不同于培训。俗话说，授之以鱼不如授之以渔，教学是授之以渔，授之以解决问题的方法，培养解决问题的能力，从而引导学生入门。通过授课与实验，要求学生能做到三个看懂，看懂原文手册，看懂硬件原理图，看懂别人写的程序。在此基础上学会如何掌握一种自己不熟悉的单片机。而培训的目的是让用户尽快地上手，一周乃至三天后就可以开始写应用程序。培训则是授之以鱼，培训的典型例子是教学生如何使用某单片机的仿真器。

由于对嵌入式系统的需求越来越复杂，集成电路技术发展又非常快，片上存储器容量不断扩大，使得在片上系统的开发中使用嵌入式实时操作系统（RTOS）成为可能。计算机对于 90% 以上的学生都不是他的专业，只是一种工具。同各行各业开发单片机应用的工程师一样，非计算机专业的研究生来自各个领域，他们不可能将主要精力花在研究操作系统上，而是要把主要精力放在所研究的专业上，他们需要的是把计算机专家们研究的成果直接拿来使用。RTOS 可以将一个复杂的应用分解成多个任务，从而简化应用程序的设计；RTOS 可以保证系统的实时性达到或接近理论上可以达到的最好水平。而多任务本身会带