

新工科建设之路·计算机类专业规划教材

嵌入式 系统导论

● 钱晓捷 程楠 编著



新工科建设之路·计算机类专业规划教材

嵌入式系统导论

钱晓捷 程楠 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书基于 ARM Cortex-M3 处理器的 STM32 微控制器,使用 MDK-ARM 开发软件和 C 语言,结合 STM32 驱动程序库和 STM32 开发板,面向底层应用编程,介绍嵌入式计算机系统的工作原理和应用技术,包括嵌入式系统组成、Cortex-M3 处理器编程结构、常用指令、STM32 微控制器通用输入/输出端口(GPIO)、外部中断接口(EXTI)、串行接口(USART)、直接存储器传输接口(DMA)、定时器接口(SysTick、IWDG、WWDG、TIMx、RTC)和模拟接口(ADC、DAC)等。

本书可以作为高等院校计算机、电子、通信及自动控制等专业“嵌入式系统导论”或“嵌入式系统基础”的教材或参考书,也适合嵌入式系统应用开发人员和希望学习嵌入式系统的普通读者和技术人员参考。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

嵌入式系统导论 / 钱晓捷, 程楠编著. —北京: 电子工业出版社, 2017.7

ISBN 978-7-121-31594-7

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

中国版本图书馆 CIP 数据核字 (2017) 第 116748 号

策划编辑: 章海涛

责任编辑: 章海涛 特约编辑: 徐 堃

印 刷: 三河市双峰印刷装订有限公司

装 订: 三河市双峰印刷装订有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 12.75 字数: 330 千字

版 次: 2017 年 7 月第 1 版

印 次: 2017 年 7 月第 1 次印刷

定 价: 34.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888, 88258888。

质量投诉请发邮件至 zllts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式: 192910558 (QQ 群)。

前 言

嵌入式系统是一个快速发展的领域，又是一个知识覆盖面广、技术范围宽的交叉学科。本书面向底层开发的初学者，侧重软件编程，是一本相对初级的“入门”教材。本书是在作者多年教学基础上整理、总结而成的，教材内容结合教学体会，既适合教师进行教学，又利于学生自主学习，具有一定特色。

在内容上，本书基于 ARM 公司 Cortex-M3 处理器的 STM32 微控制器，使用主流的 MDK-ARM 开发软件和高级语言（C 语言），结合 STM32 驱动程序库和 STM32 开发板，介绍嵌入式系统的工作原理和应用技术。在结构上，本书不是照搬产品参考手册，也有别于数据手册的面面俱到，而是从学习者角度重新编排，做到有详有略，既有对技术、原理的补充说明，又有对程序代码的分析和解释。相较于大量的中英文资料，尤其是参考手册、用户指南之类的官方文档，本书没有烦琐的寄存器细节和堆砌的固件库函数列表，而是进行知识重组、内容提炼，并尝试在正文中提供一些阅读帮助，还通过大量习题让读者巩固所学；另外，设计开放性题目，引导读者阅读相关文档、深入学习。

全书在前 2 章提供必要的基础知识，后续章节以完成示例项目为目标（案例导向），介绍相关技术，分析程序流程，详解核心代码，突出实践和提供指导。希望读者完成每个项目，“学中做、做中学”，即所谓“DIY (Do It Yourself)”和“LBD (Learning By Doing)”。

本书面向信息技术类专业的普通本科（专科）学生或自学者，考虑初学者的实际知识水平，努力以清晰的结构，浅显的语言，循序渐进、由浅入深，结合示例项目，展开 STM32 微控制器及其基本外设接口的编程应用。读者应熟悉 C 语言编程，最好具有计算机组成原理或者微机原理的先修知识。另外，课程虽然涉及硬件接口，但本书重点讲述软件编程，配合开发环境的软件模拟，可以不需购买开发板，也可以购买价格低廉的 STM32 开发板。

本书由郑州大学钱晓捷组织，钱晓捷和程楠共同编写。钱晓捷老师编写了前 7 章以及第 8 章和第 9 章前 2 节的初稿；程楠老师编写剩余章节的初稿，并验证了所有示例项目。全书由钱晓捷老师统稿。本书编写和试用过程中，得到张青、姚俊婷等老师的帮助，在此表示感谢。

由于编者水平有限，本书难免会有疏漏和不当之处，欢迎广大师生和读者指正（iexjqian@zzu.edu.cn，iencheng@zzu.edu.cn）。

本书为读者提供相关教学资源（含电子课件），有需要者，请登录 <http://www.hxedu.com.cn>，注册之后进行下载。

作 者

本书教学要求

全书共分 10 章，各章的教学要求如下。

目 录	教学要求
第 1 章 嵌入式系统设计基础	熟悉嵌入式系统的概念、特点和组成，了解嵌入式系统的开发模式、流程以及配套的软件、硬件
第 2 章 ARM Cortex-M3 处理器	了解 ARM 处理器发展，熟悉 Cortex-M3 处理器的结构、寄存器和存储器地址空间
第 3 章 Thumb 指令系统	理解 ARM 指令集和 Thumb 指令集；通过语句格式、程序结构、变量定义和常量表达，了解 ARM 统一汇编语言 UAL 的特点；通过数据寻址、常用指令，了解 Thumb 指令系统的特色，进而理解 STM32 启动代码及其作用；通过简单汇编语言程序的开发，掌握 MDK-ARM 集成开发工具的使用
第 4 章 STM32 微控制器	了解 STM32 系列微控制器及其系统结构，熟悉 CMSIS 和 STM32 库及其作用，掌握 C 语言在嵌入式系统开发的应用特点，掌握复位和时钟控制单元（RCC）的作用
第 5 章 STM32 的通用 I/O 端口	在了解通用 I/O 端口（GPIO）功能、结构、寄存器的基础上，结合 LED 控制输出和按键查询输入两个 GPIO 示例项目，掌握使用 MDK 进行项目创建、STM32 库函数使用、GPIO 应用程序编写、模拟运行和硬件仿真的整个开发过程，最后总结基于 STM32 库开发的一般规则
第 6 章 CM3 异常和 STM32 中断	了解 Cortex-M3 的异常和中断控制器（NVIC）；结合 STM32 外部中断（EXTI），实现按键中断的示例项目；掌握中断控制器的初始化配置和外设的中断配置，以及中断服务程序的编写
第 7 章 STM32 的串行通信接口	在了解异步串行通信协议基础上，熟悉 STM32 的串行接口（USART）功能，结合 C 语言标准输入/输出函数的重定向示例项目，掌握 USART 接口的应用
第 8 章 STM32 的 DMA 接口	在理解直接存储器传输（DMA）的作用的基础上，熟悉 STM32 支持的 DMA 功能，结合 USART 接口的 DMA 传输示例项目，掌握 DMA 初始化配置和应用编程
第 9 章 STM32 的定时器接口	以定时为主线，熟悉系统时钟（SysTick）、STM32 看门狗（IWDG 和 WWDG）、STM32 基本定时器（TIMx）和实时时钟（RTC）的功能结构；各结合一个定时相关的示例项目，掌握它们的编程应用
第 10 章 STM32 的模拟接口	熟悉 STM32 模拟/数字转换器（ADC）和数字/模拟转换器（DAC）的结构及特性；结合数据采集和电压输出示例项目，掌握它们的编程应用

目 录

第 1 章 嵌入式系统设计基础	1
1.1 嵌入式系统概述	1
1.1.1 什么是嵌入式系统	1
1.1.2 嵌入式系统的技术特点	2
1.1.3 嵌入式系统的组成	2
1.2 嵌入式系统的开发	4
1.2.1 嵌入式系统的开发模式	4
1.2.2 嵌入式系统开发需要的软件、硬件	5
1.2.3 软件开发流程	6
习题 1	7
第 2 章 ARM Cortex-M3 处理器	9
2.1 ARM 处理器	9
2.2 Cortex-M3 处理器结构	10
2.3 寄存器	13
2.4 存储器组织	15
习题 2	18
第 3 章 Thumb 指令系统	20
3.1 ARM 指令集和 Thumb 指令集	20
3.2 统一汇编语言	21
3.2.1 汇编语言的语句格式	21
3.2.2 汇编语言的程序结构	22
3.2.3 存储器空间分配指示符	24
3.2.4 常量表达	24
3.3 数据寻址	25
3.3.1 寄存器寻址	26
3.3.2 存储器寻址	27
3.4 常用指令	28
3.4.1 处理器指令格式	28
3.4.2 数据传送指令	29
3.4.3 数据处理指令	31
3.4.4 分支跳转指令	32

3.5	STM32 启动代码	33
3.6	开发工具 MDK	37
3.6.1	MDK 安装	37
3.6.2	MDK 目录结构	38
3.6.3	创建应用程序	39
3.6.4	汇编语言程序的开发	42
	习题 3	44
第 4 章	STM32 微控制器	46
4.1	STM32 微控制器结构	46
4.1.1	STM32 系列微控制器	46
4.1.2	STM32 系统结构	48
4.1.3	STM32 存储结构	49
4.2	STM32 微控制器开发	51
4.2.1	Cortex 微控制器软件接口标准 CMSIS	51
4.2.2	STM32 驱动程序库	53
4.2.3	C 语言应用	54
4.3	复位与时钟控制 (RCC)	57
	习题 4	62
第 5 章	STM32 的通用 I/O 端口	65
5.1	GPIO 的结构和功能	65
5.2	GPIO 寄存器	66
5.2.1	GPIO 寄存器的功能	67
5.2.2	GPIO 寄存器的应用	68
5.3	GPIO 输出应用示例: LED 灯的亮灭控制	70
5.3.1	项目创建和选项配置	70
5.3.2	应用程序分析	72
5.3.3	应用程序编写	76
5.3.4	程序模拟运行	79
5.3.5	程序硬件仿真	81
5.4	GPIO 输入应用示例: 查询按键状态	83
5.4.1	程序分析和编写	84
5.4.2	程序调试和运行	86
5.5	STM32 库编程总结	88
5.5.1	基于 STM32 库的开发过程	88
5.5.2	使用 STM32 库的一般规则	88
5.5.3	对比直接对寄存器编程	91
	习题 5	93

第 6 章	CM3 异常和 STM32 中断	96
6.1	Cortex-M3 的异常	96
6.2	STM32 的中断应用	99
6.2.1	NVIC 初始化配置	100
6.2.2	外部中断 EXTI	101
6.2.3	GPIO 引脚的中断配置	104
6.2.4	芯片外设的中断配置	104
6.3	EXTI 应用示例: 按键中断	105
6.3.1	主程序流程	105
6.3.2	中断初始化配置	106
6.3.3	中断应用程序编写	108
	习题 6	110
第 7 章	STM32 的串行通信接口	112
7.1	串行异步通信	112
7.1.1	串行异步通信字符格式	112
7.1.2	串行异步通信接口	113
7.2	通用同步/异步接收/发送器	114
7.2.1	STM32 的 USART 功能	115
7.2.2	STM32 的 USART 应用	116
7.3	USART 应用示例: 实现 C 语言标准输入/输出函数	118
7.3.1	USART 初始化配置	118
7.3.2	输入/输出函数的重定向	120
7.3.3	信息交互应用程序	122
7.3.4	USART 接口的中断应用	124
	习题 7	128
第 8 章	STM32 的 DMA 接口	130
8.1	DMA 控制器	130
8.1.1	DMA 传输过程	130
8.1.2	STM32 的 DMA 功能	131
8.1.3	STM32 的 DMA 应用	132
8.2	DMA 应用示例: USART 接口的 DMA 传输	135
8.2.1	DMA 初始化配置	135
8.2.2	DMA 传输应用程序编写	137
8.3	DMA、USART 和 GPIO 的综合应用	139
8.3.1	综合应用的项目分析	139
8.3.2	综合应用的编程	140
	习题 8	143

第 9 章 STM32 的定时器接口	145
9.1 系统时钟 (SysTick)	145
9.1.1 系统嘀嗒定时器	145
9.1.2 SysTick 应用示例: 精确定时	148
9.2 STM32 看门狗	150
9.2.1 独立看门狗	150
9.2.2 IWDG 应用示例: IWDG 复位	153
9.2.3 窗口看门狗	154
9.2.4 WWDG 应用示例: 适时“喂狗”	157
9.3 STM32 定时器	159
9.3.1 基本定时器	159
9.3.2 基本定时器应用示例: 周期性定时中断	162
9.4 STM32 实时时钟	165
9.4.1 RTC 结构及特性	166
9.4.2 RTC 应用示例: 闹钟	168
习题 9	171
第 10 章 STM32 的模拟接口	173
10.1 STM32 的 ADC 接口	173
10.1.1 ADC 结构及特性	173
10.1.2 ADC 的转换模式	178
10.1.3 STM32 的 ADC 函数	179
10.1.4 ADC 应用示例: 数据采集	181
10.2 STM32 的 DAC 接口	185
10.2.1 DAC 结构及特性	185
10.2.2 STM32 的 DAC 函数	188
10.2.3 DAC 应用示例: 输出模拟电压	190
习题 10	192
参考文献	194

第 1 章 嵌入式系统设计基础

进入 21 世纪, 嵌入式系统 (Embedded System) 逐渐流行。嵌入式系统是嵌入式计算机系统的简称, 是相对于通用计算机系统而言的一类专用计算机系统。嵌入式系统无处不在, 广泛应用于科学研究、工程设计、军事领域, 以及人们日常生活的方方面面。各种嵌入式设备 (系统) 在数量上远远超过通用计算机系统。

本章主要介绍嵌入式系统的定义、特点、组成和开发, 为学习嵌入式系统设计奠定基础。

1.1 嵌入式系统概述

嵌入式系统的发展是在微处理器 (Microprocessor) 问世后, 源于单片机 (Single Chip Microcomputer, SCM)。单片机是指通常用于控制领域的微处理器芯片, 其内部除中央处理器 (CPU) 外, 还集成了计算机的其他一些主要部件, 如只读存储器 (ROM) 和随机读写存储器 (RAM)、定时器、并行接口、串行接口, 有的芯片集成了模拟/数字 (A/D)、数字/模拟 (D/A) 转换电路等。换句话说, 一块芯片几乎就是一台计算机, 只要配上少量的外部电路和设备, 就可以构成具体的应用系统。单片机是国内习惯的名称, 国际上多称为微控制器 (Micro Controller) 或嵌入式控制器 (Embedded Controller)。

微控制器的发展初期 (1976 至 1978 年) 以英特尔 (Intel) 公司的 8 位 MCS-48 系列为代表。1978 年以后, 微控制器进入普及阶段, 以 8 位为主, 最著名的是英特尔公司的 8 位 MCS-51 系列, 还有爱特梅尔 (Atmel) 公司的 8 位 AVR 系列、Microchip Technology 公司的 PIC 系列。1982 年以后, 出现了高性能的 16 位、32 位微控制器, 如英特尔公司的 16 位 MCS-96/98 系列、基于 ARM (Advanced RISC Machine) 处理器核心的 32 位微控制器。

通常, 单片机主要是指 8 位微控制器。至今, 兼容 8 位 51 系列的单片机仍然在各种产品中广泛应用。因此, 国内称为“单片机”的课程、教材等多以 51 系列为教学内容。16 位微控制器由于性价比不高, 几乎被 32 位微控制器替代。32 位微控制器多基于 ARM 处理器。ARM 处理器采用精简指令集 RISC (Reduced Instruction Set Computer) 结构, 具有耗电少、成本低、性能高的特点, 因此广泛应用于 32 位嵌入式系统。所以, 通常所说的嵌入式系统主要是指基于 ARM 核心的微控制器构成的计算机系统。在微控制器领域, ARM 处理器传统上使用 ARM7 和 ARM9 (在高性能应用场合使用 ARM11), 而目前应用 Cortex-M 系列, 最基本、最主要的是 Cortex-M3 (本书的教学内容)。更高性能的 ARM 处理器有 Cortex-M4 等, 性能略低但功耗更低的有 Cortex-M0/M1 等。

1.1.1 什么是嵌入式系统

简单地说, 嵌入式系统是将计算机的软件、硬件嵌入机电设备所构成的专用计算机系统。这个说法虽然简单, 但是体现了嵌入式系统的 3 个基本特点: 嵌入性、专用性、计算机系统。

美国电气和电子工程师协会 (Institute of Electrical and Electronics Engineers, IEEE) 给出的定义是: 嵌入式系统是“用于控制、监视或辅助设备、机器或装置操作的仪器”(devices used to control, monitor, or assist the operation of equipment, machinery or plants)。国内一般采用的较完整的定义是: 嵌入式系统是以应用为中心, 以计算机技术为基础, 软件、硬件可剪裁, 以适用于应用系统对功能、可靠性、成本、体积、功耗等要求严格的专用计算机系统。这些定义表明, 嵌入式系统主要体现了 3 方面的含义: ① 嵌入式系统与具体应用系统紧密结合, 具有很强的专用性; ② 嵌入式系统融合先进的计算机技术、电子技术以及各应用领域的具体技术; ③ 嵌入式系统必须根据应用需求对软件、硬件进行高效设计, 量体裁衣, 剔除冗余, 在满足功能性和可靠性的基础上降低产品成本和能源损耗。

基于以上含义, 在现代社会, 人们随时携带的智能移动终端 (智能手机、平板电脑等) 就是高性能的嵌入式系统。实际上, 人们的日常生活、工作中到处存在各种各样的嵌入式系统, 如控制系统、智能仪器、家用电器、网络通信设备、医疗设备等。

1.1.2 嵌入式系统的技术特点

较之功能强大、置于桌面的通用计算机系统, 多隐藏于机电设备内部的嵌入式系统显然有所不同, 因此在研发和应用中的软件、硬件呈现出独有的特点。

1. 硬件的特点

为了能够嵌入具体的设备, 嵌入式系统的硬件电路高度集成, 体积较小。同时, 在保证功能性、实时性、可靠性等基础上, 硬件电路还需要具有低成本、低功耗等特性。这使得嵌入式系统本身无法支持自身的开发, 所以开发环境受限, 系统调试复杂。

由于开发的应用程序常需要直接控制硬件电路, 因此要求嵌入式系统开发人员具有一定的计算机硬件工作原理方面的知识。

2. 软件的特点

嵌入式系统的软件包括系统软件 (主要是嵌入式操作系统) 和应用软件。一些功能简单的嵌入式系统可以没有操作系统支持, 或者只是功能单一的监控程序。相对于通用计算机系统, 嵌入式系统的系统软件和应用软件往往紧密结合成一个有机的整体, 并呈现如下特点。

- ❖ 实时性和可靠性: 系统能够随时响应各种应用事件, 并及时处理完成。为应对复杂的现场情况, 系统需要具有抗干扰能力、自我恢复等能力。
- ❖ 软件剪裁和固化: 软件程序能够结合具体应用适当剪裁、组合, 达到最优, 并保存在半导体存储器芯片 (不是保存在外部磁盘上) 中, 使得系统开机后可以直接运行。
- ❖ 代码高效: 虽然存储器容量不断增加, 但减少程序二进制代码仍然非常重要。精练的代码不仅节省了存储空间, 还提升了执行性能, 同时提高了系统的实时性、可靠性, 并能够降低功耗。

1.1.3 嵌入式系统的组成

嵌入式系统主要由嵌入式处理器、外围硬件设备、嵌入式操作系统 (可选) 和用户应用程序组成, 如图 1-1 所示。

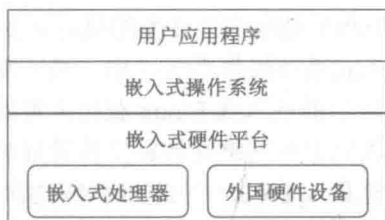


图 1-1 嵌入式系统组成结构

1. 嵌入式处理器

嵌入式处理器是指作为运算和控制中心的中央处理器 CPU。随着嵌入式系统的发展，作为硬件核心部件的嵌入式处理器呈现出多样性，有如下芯片类型。

① 微控制器 (Micro Controller Unit, MCU): 也称为单片机，一般以某种处理器为核心，芯片内部集成存储器、输入/输出 (I/O) 接口等必要电路，具有体积小、功耗少、成本低等特点，是目前嵌入式系统的主流产品，如大量基于 ARM 处理器的微控制器。

② 嵌入式微处理器 (Embedded Microprocessor Unit, MPU): 源于通用微处理器，但按照嵌入式应用需求专门设计。在功能上与标准通用微处理器基本一样，但在降低功耗、提高抗干扰能力等方面做了改进，如 AM186/88、386EX、PowerPC、MIPS 处理器等。

③ 数字信号处理器 (Digital Signal Processor, DSP): 一种对系统结构和指令系统进行特殊设计的处理器。其内部集成有高速乘法器，能够执行快速乘法和加法运算，更适合数字信号的高速处理。DSP 芯片自 1979 年英特尔公司开发 Intel 2920 以后，经历了多代发展，其中美国德州仪器 TI (Texas Instruments) 公司的 TMS320 各代产品具有代表性，还有摩托罗拉公司的 DSP56000 系列。

④ 片上系统 (System on Chip, SoC): 随着集成电路制作工艺和电子设计自动化技术迅速发展，可以针对具体应用，把整个电子系统全部集成于一块半导体芯片中，构成片上系统。SoC 技术把处理器、存储器和各种外设作为器件库，以使用户根据应用需求自行开发、设计应用系统，实现了真正的“量体裁衣”。

2. 外围硬件设备

嵌入式系统的硬件除了处理器核心，生产厂商还应根据需求，配套常用接口电路、外围器件以及具体外设。这里的设备 (Device) 不仅包括通常所说的输入/输出设备 (外设, Peripheral)，还包括各种接口电路 (Interface) 器件等。常用的接口电路如下所述。

① 系统基本电路: 提供嵌入式系统运行必需的时钟电路、复位电路、供电电路以及基本的存储器 (Flash ROM、SRAM)。

② 基本接口电路: 如通用 I/O 端口、通信接口、定时电路、模拟/数字转换电路。

③ 常用外设支持电路: 如 CAN 总线、USB 接口、存储卡接口、以太网接口等。

3. 嵌入式操作系统

嵌入式系统经历了无操作系统、简单操作系统 (监控程序) 和实时操作系统阶段。对于工作简单、任务单一的嵌入式系统，可能并不需要操作系统；而功能强大、任务复杂的嵌入式系统通常需要利用操作系统。嵌入式操作系统一方面为应用程序提供底层硬件驱动的支持，另一方面可以减少开发工作量。对工作任务有严格时间要求的嵌入式系统，需要实时操作系统

(Real-Time Operation System, RTOS) 调度多个任务的执行, 并满足实时性。

20 世纪 70 年代后期出现嵌入式系统操作系统以来, 经过发展, 面向不同应用, 形成了多种嵌入式操作系统。例如, 广泛应用的嵌入式 Linux 获得大量的硬件支持, 具有源码开放、内核稳定、软件丰富等优势, 并提供完善的网络支持和文件管理机制等。智能手机和平板电脑主要应用 Google 公司的 Android 系统和 Apple 公司的 iOS 系统。实时操作系统常用免费的 uC/OS-II (现在已发展为 uC/OS-III) 或者商业化的 VxWorks 等。

4. 用户应用程序

用户应用程序是按照具体应用项目需求而开发的应用软件, 是开发人员的主要工作。嵌入式系统开发人员可以笼统地分成嵌入式硬件工程师和嵌入式软件程序员。

从事硬件设计的硬件工程师需要完成器件选择、PCB 板设计等工作, 通常使用硬件描述语言 (Hardware Description Language, HDL) 进行电子设计。为硬件电路设计软件接口的硬件工程师 (或软件程序员) 需要了解低层硬件电路, 熟悉启动程序 (Bootloader) 和设备驱动程序, 通常使用 C/C++ 语言和汇编语言进行软件编程。

嵌入式软件程序员又分为系统程序员和应用程序员。系统程序员的主要工作涉及嵌入式操作系统的剪裁和移植、驱动程序编写和移植等, 通常使用 C/C++ 语言和汇编语言。应用程序员的主要任务是基于嵌入式操作系统面向高层应用, 使用 C++、Java 等进行面向对象技术开发。

本书作为“嵌入式系统导论”课程的教材, 主要介绍基本的入门知识, 涉及的内容面向从事底层硬件设计的软件程序员。

1.2 嵌入式系统的开发

类似常规的工程设计方法, 嵌入式系统的开发分成 3 个阶段: 分析、设计和实现。需求分析结束, 开发人员通常面临一个纠结的问题, 即硬件平台和软件平台的选择。硬件平台主要指处理器, 软件平台主要指操作系统、编程语言和集成开发环境等。其中, 处理器的选择是最重要的。C 语言简洁、高效, 具有广泛的库函数支持, 目前是嵌入式系统开发最主要的编程语言。

1.2.1 嵌入式系统的开发模式

由于嵌入式系统往往软、硬件资源有限, 无法直接支持开发, 通常需要通用微型机 (个人计算机, 即 PC) 的支持, 被称为宿主机 (Host); 待开发的嵌入式系统称为目标机 (Target)。所以, 嵌入式系统一般采用宿主机-目标机开发模式 (如图 1-2 所示), 以便利用宿主机上丰富的软件、硬件资源以及良好的开发环境和调试工具, 来开发目标机上的软件。

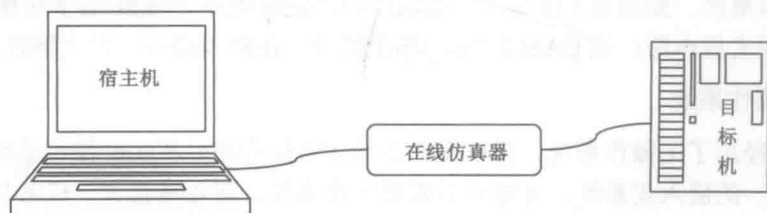


图 1-2 宿主机-目标机开发模式

宿主机-目标机组成的开发平台中, 宿主机建立完整的开发环境, 交叉编译产生目标机的

可执行代码，然后通过在线仿真器、串行接口、网络等方式下载到目标机运行。这称为交叉开发（Cross Development）。其中，交叉编译是指宿主机的开发软件将源程序编译生成目标机的机器代码，而不是运行于宿主机本身的可执行代码。

在目标机运行交叉开发的可执行代码时，常需要调试。在宿主机的软件集成开发环境中，可以先利用模拟器（Simulator）进行软件模拟，再连接在线仿真器（In-Circuit Emulator, ICE）进行硬件仿真，实现目标代码的运行和调试。也就是说，调试程序运行于宿主机，而被调试程序运行于目标机，两者通过在线仿真器或者串口、网络进行通信。调试程序可以控制被调试程序，查看和修改目标机的寄存器、主存单元，并且进行断点和单步调试等操作，即远程调试（Remote Debug）。

1.2.2 嵌入式系统开发需要的软件、硬件

除了宿主机外，为方便嵌入式系统开发，许多公司提供多种软件、硬件。

1. 开发工具套件

嵌入式系统的软件开发工具主要使用交叉开发集成环境（Integrated Development Environment, IDE），大多数开发工具套件包含编译程序、汇编程序、连接程序、闪存编程器、调试器、模拟器以及文件转换等工具。以 Cortex-M 微控制器的嵌入式开发为例，有十家公司销售基于 Cortex-M 微控制器的 C 语言编译器套件。其中既有开源免费工具，也有价格低廉的工具，还有高端商业软件包，如源自 Keil 公司的 ARM 微控制器开发工具集（MDK-ARM）、ARM 公司的 DS-5（Development Studio 5）、IAR 公司的嵌入式工作工具（IAR for ARM）、UNIX/Linux 平台的 GNU 编译器集合（GCC）等。

2. 开发板

开发板（Demo Board）是用于嵌入式系统开发的电路板，由微控制器、外扩存储器、常用 I/O 接口和简单的外部设备等组成，如图 1-3 所示。开发板可以由嵌入式系统开发人员根据应用需求自己设计制作，故也称为目标板。为便于研制目标系统和产品推广，许多公司提供基于特定微控制器的开发板。一些半导体厂商也会提供廉价的开发板或评估板（Evaluation Board）用于产品测试，软件开发公司（如 Keil 公司）也推出自己的评估板。

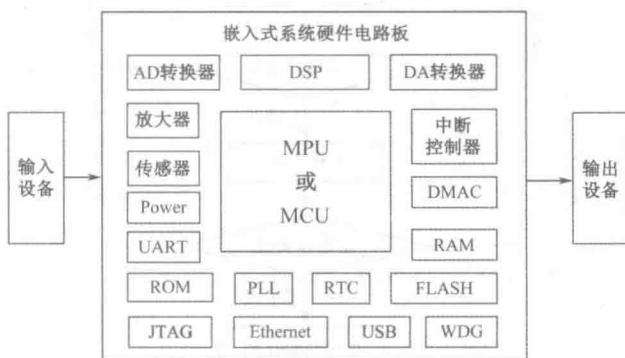


图 1-3 嵌入式系统开发板的硬件组成

实际开发过程中，也许还需要连接于开发板的附加硬件，如外部 LCD 显示模块、通信接

口适配器等，可能还会用到逻辑分析仪/示波器、信号发生器等硬件实验工具。

开始学习 Cortex-M 微控制器时，开发板并不是必需的。有些开发工具套件包括指令集模拟器，Keil MDK-ARM 甚至支持部分 Cortex-M 微控制器的设备级模拟。

3. 在线仿真器

为了给开发板下载程序、进行目标系统调试，常需要一个在线仿真器，即调试适配器。它通常一端连接 PC 的 USB 接口，一端连接开发板。嵌入式系统的软件开发工具公司都有自己的调试适配器产品，如 Keil 公司的 ULINK 系列、Segger 公司的 J-LINK 等。多数开发工具套件也支持第三方调试仿真器。

4. 设备驱动程序

为了便于微控制器软件开发人员开展工作，微控制器厂商通常提供设备的基本驱动程序，包括寄存器定义、外设访问函数的头文件和源程序代码。这些驱动程序代码和应用示例程序放在厂商的网站，供免费下载。开发人员可以将这些文件添加到自己的软件项目中，通过函数调用，方便地访问外设功能和存取外设寄存器；也可以参考驱动程序和示例代码，编写自己的应用程序；还可以修改驱动程序代码，优化自己的应用程序。

另外，除了设备驱动程序，微控制器厂商还会提供微控制器的用户手册、应用程序说明书、常见问题及解答、在线讨论组等丰富的网络资源。

1.2.3 软件开发流程

根据使用的开发工具套件不同，软件开发流程会有差异，但主要步骤大致相同。对于使用宿主机的集成化开发环境，软件开发流程一般包括创建项目、添加文件、编译连接、下载调试等步骤，如图 1-4 所示。

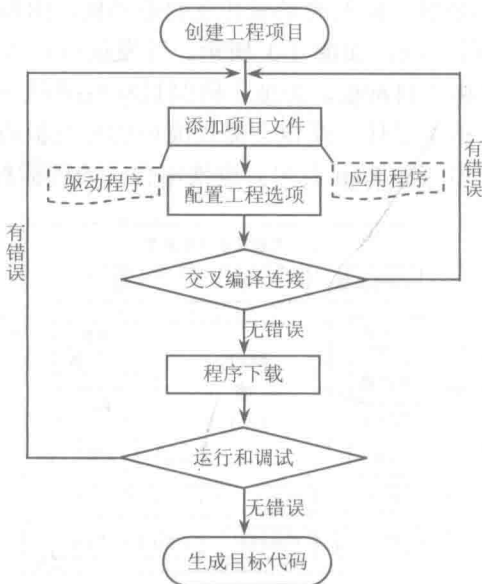


图 1-4 软件开发流程（一般步骤）

① 创建工程项目：在配置硬件设备和安装软件开发工具后，就可以开始创建工程项目，

通常需要根据项目文件的存储位置及目标处理器。

② 添加项目文件：开发人员需要创建源程序文件，编写应用程序代码，并添加到工程项目中；还将使用设备驱动程序的文件，包括启动代码、头文件和一些外设控制函数，甚至中间件（Middleware）等。这些文件也需要添加到项目中。

③ 配置工程选项：源于硬件设备的多样性和软件工具的复杂性，工程项目提供了不少选项，需要开发人员配置，如输出文件类型和位置、编译选项和优化类型等，还要根据选用的开发板和在线仿真器，配置代码调试和下载选项等。

④ 交叉编译连接：利用开发软件工具对项目的多个文件分别编译，生成相应的目标文件，然后连接生成最终的可执行文件映像，以下载到目标设备的文件格式保存。如果编译连接有错误，返回修改；如果没有错误，先进行软件模拟运行和调试，再下载到开发板运行和调试。

⑤ 程序下载：目前，绝大多数微控制器都使用闪存（Flash Memory）保存程序。创建可执行文件映像后，需要使用在线仿真器（或串行接口、网络），将其下载到微控制器的闪存中，实现闪存的编程；还可以将可执行文件下载到SRAM中运行。

⑥ 运行和调试：程序下载后，可以启动运行，看是否正常工作。如果有问题，连接在线仿真器，借助软件开发工具的调试环境进行断点和单步调试，观察程序操作的详细过程。如果应用程序运行有错误，返回修改。

习题 1

1-1 单项或多项选择题（选择一个或多个符合要求的选项）。

(1) 嵌入式系统的英文名称是（ ）。

- A. Microprocessor B. Embedded Controller
C. Micro Controller D. Embedded System

(2) 嵌入式系统的特点是（ ）。

- A. 兼具自我开发能力 B. 软件、硬件可剪裁
C. 体积大，功耗也大 D. 系统通用性好

(3) 随着嵌入式系统的发展，作为嵌入式处理器的中央处理器有（ ）类型。

- A. MCU B. MPU C. DSP D. SoC

(4) 嵌入式操作系统有多种，包括（ ）。

- A. MS-DOS B. uC/OS-III C. Android D. Unix

(5) 进行嵌入式系统开发，通常采用的模式是（ ）。

- A. 仅使用目标机（开发板）开发模式 B. 宿主机-目标机开发模式
C. 基于宿主机的软件模拟开发模式 D. 通过网络实施的远程开发模式

1-2 什么是嵌入式系统？嵌入式系统有哪些主要技术特点？

1-3 简单区分几种嵌入式处理器：微控制器 MCU（单片机 SCM）、嵌入式微处理器 MPU、数字信号处理器 DSP 和片上系统 SoC。

1-4 嵌入式系统的开发模式是什么？什么是交叉编译和远程调试？

1-5 简述嵌入式系统的软件开发流程。

1-6 嵌入式系统发展迅猛，有些内容无法也不需详细介绍。而且，当你阅读本书的时候，有些内容会有所改变，有些技术会有所改进。所以，作为教学内容的延伸，你可以选择某些方面（不限

于本章内容，可以是后续章节内容)深入学习，并提交总结报告(或论文)。例如:

- (1) 通过某个产品，对比嵌入式系统与通用 PC 系统的主要差异。
- (2) 常用嵌入式操作系统有哪些?各有什么特点?
- (3) 简介基于 STM32 微控制器的开发板和配套的在线仿真器。
- (4) ARM 公司的 Cortex-M 系列处理器的最新发展。
- (5) ST 公司的 STM32 系列微控制器的最新发展。

.....