

德国嵌入式专家亚历山大·胡瓦特先生
精心指导开发，引入项目驱动的一体化教程

嵌入式应用技术

基于STM32固件库编程

编著 刘志远 黄远民 易铭 樊亚妮 梁利滨



STM32

嵌入式应用技术

——基于 STM32 固件库编程

编著 刘志远 黄远民 易 铭
樊亚妮 梁利滨

苏州大学出版社

图书在版编目(CIP)数据

嵌入式应用技术: 基于 STM32 固件库编程/ 刘志远
等编著. —苏州: 苏州大学出版社, 2019. 6
ISBN 978-7-5672-2826-9

I. ①嵌… II. ①刘… III. ①微处理器—系统设计
IV. ①TP332. 021

中国版本图书馆 CIP 数据核字(2019)第 107802 号

QIANRUSHI YINGYONG JISHU

嵌入式应用技术

——基于 STM32 固件库编程

刘志远 黄远民 易 铭 樊亚妮 梁利滨 编著

责任编辑 周建兰

苏州大学出版社出版发行

(地址: 苏州市十梓街 1 号 邮编: 215006)

苏州工业园区美柯乐制版印务有限责任公司印装

(地址: 苏州工业园区东兴路 7-1 号 邮编: 215021)

开本 787mm × 1 092mm 1/16 印张 19.25 字数 457 千

2019 年 6 月第 1 版 2019 年 6 月第 1 次印刷

ISBN 978-7-5672-2826-9 定价: 49.50 元

苏州大学版图书若有印装错误, 本社负责调换

苏州大学出版社营销部 电话: 0512-67481020

苏州大学出版社网址 <http://www.sudapress.com>

苏州大学出版社邮箱 sdcbs@suda.edu.cn

Vorwort

序

Eingebettete Systeme sind aus unserem Leben nicht mehr wegzudenken. Sie begleiten uns unterwegs im Auto, im Zug und im Flugzeug. Unsere Wohnungen werden durch sie smart. Sie können uns sogar dabei helfen gesünder zu leben. Bei der Arbeit, beim Lernen, in der Freizeit, kaum ein Lebensbereich für den es nicht smarte Dinge gibt. Was sich in fast allen smarten Dingen versteckt sind kleine Computer. Wir nennen diese Mikrocontroller. Wie versteckt diese Mikrocontroller sein können merken wir kaum. Zum Beispiel sind in einem modernen Auto mehr als 100 solcher Mikrocontroller versteckt. Wir haben also nicht einen Boardcomputer der das Auto smart macht, sondern ein Netzwerk von vielen Mikrocontrollern und Steuergeräten. Die Tendenz zum Einsatz von Mikrocontrollern ist nach wie vor steigend. Nächstes Jahr sind in den neuen Modellen vielleicht schon mehr als 200 Mikrocontroller pro Fahrzeug verbaut. Die Anzahl solcher Systeme weltweit ist unvorstellbar groß. Momentan sind weit mehr als 80% aller Computer eingebettete System. Genau in diesem Moment, jetzt, laufen fast zehnmal mehr Mikrocontroller als Notebook- oder Desktopprozessoren. Ich verrate hier kein Geheimnis, wenn ich sage, dass diese Mikrocontroller von irgendjemandem programmiert werden müssen.

嵌入式系统已经成为我们生活中不可或缺的一部分。在汽车、火车和飞机上,它们伴随我们左右。我们的住宅也因有嵌入式系统而变得智能。它们甚至可以帮助我们过上更健康的生活。在工作、学习中,在空闲时间,我们生活中几乎没有一个领域不涉及智能。几乎所有智能产品中都隐藏着微型计算机,我们称之为微控制器。平时我们并未注意到这些微控制器隐藏在哪里。例如,汽车中隐藏着 100 多个这样的微控制器。也就是说,我们没有使用行车电脑来使汽车变得智能化,而是利用一个由许多微控制器和控制设备组成的网络。微控制器的应用越来越普遍。明年,也许每辆新车型中会安装超过 200 个微控制器。嵌入式系统在全球范围内的数量是不可想象的。目前,超过 80% 的计算机都是嵌入式系统。当下正在运行的嵌入式处理器的数量是 PC 处理器数量的十倍。众所周知,这些微控制器需要有人来编程。

Es gibt neben der zunehmenden Verbreitung eingebetteter Systeme ein weiteres Phänomen mit dem man sich auseinandersetzen muss. Die Leistungsfähigkeit aller Prozessoren auch der Mikrocontroller verdoppelt sich etwa alle zwei Jahre. Dieses Phänomen ist bekannt als Moor's Law. Es beschreibt das exponentielle Wachstum der Leistungsfähigkeit von Mikroelektronik. Obwohl schon oft das Ende von Moor's Law verkündet wurde ist dieser Trend ungebrochen und in

den Laboren der Entwickler werden die nächsten Generationen von Systemen mit doppelt so viel Speicher, doppelt so schneller Geschwindigkeit, mit noch mehr Rechenkernen, halb so groß und mit der Hälfte an Strombedarf getestet.

除了嵌入式系统日益普及之外,还有另一个需要解决的问题。包括微控制器在内的所有处理器的性能大约每两年翻一番。这种现象被称为摩尔定律。它描述了微电子性能呈指数级增长的趋势。虽然摩尔定律已经多次宣布失效,但这种趋势未被中断:开发人员在实验室中测试的下一代系统的内存是原来的两倍,速度是原来的两倍,内核数量也在不断增加,而处理器的大小只有原来的一半,耗电量也只有原来的一半。

Was bedeutet das alles? Es bedeutet, dass die Fähigkeit Mikrocontroller zu programmieren zunehmend gefragt sein wird. Es bedeutet, dass die Lösungen die programmiert werden müssen zunehmend komplexer werden. Es bedeutet, dass nur durch lebenslanges Lernen die sich die immer weiter entwickelnde Technologie beherrscht werden kann. Und es bedeutet, dass embedded Systems nie langweilig werden. Wenn Sie heute mit kleinen 8 Bit Controllern und der Programmiersprache C beginnen werden Sie sehr bald 32 Bit Controller programmieren und sich mit objektorientierten Programmiersprachen und vielleicht auch mit der grafischen Programmierung in der Unified Modeling Language UML auseinandersetzen. Mit jedem neuen Schritt werden nicht nur die Systeme die Sie entwickeln immer smarter, sondern auch Sie.

这一切意味着什么?这意味着具有微控制器编程能力的人将越来越受欢迎,也意味着解决问题的方案会变得越来越复杂,只有通过终身学习才能掌握日新月异的技术。嵌入式系统永远不会令人枯燥乏味。如果你从今天开始学习小型8位控制器和C语言,你很快就会学习32位控制器和面向对象的编程语言,甚至可能学习统一建模语言UML中的图形编程。一步一步,不仅您开发的系统变得更聪明,而且您也需要一起变得更聪明。

Das vorliegende Lehrbuch wird Sie Schritt für Schritt in diese spannende Technologie einführen. Für mich war ein großes Vergnügen, dass ich an diesem Projekt mitwirken konnte.

本教材将由浅入深地向您介绍有趣的嵌入式系统技术。我很高兴参与了佛山职业技术学院“嵌入式应用技术”课程开发的项目。

Ich wünsche viel Erfolg und vor allem Freude beim Lernen.

希望您能享受学习的乐趣,并学有所获!



Alexander Huwaldt
亚历山大·胡瓦特

前言

第四次工业革命已经来临,物联网、智能传感器等技术的发展均离不开嵌入式系统。本书以嵌入式应用技术为基础,在德国嵌入式专家亚历山大·胡瓦特的指导下,选取 STM32F407 作为嵌入式学习的平台,通过 STM32F4 的固件库编程来展开嵌入式应用技术的学习。

本书在内容编排上,从 STM32 的基础应用出发,以项目为载体,选取典型应用作为项目案例,逐步深入,符合学生的认知规律。在项目的实践过程中,从固件库的应用出发,便于学生掌握 STM32 系列的固件库编程。

建议本书的学时数为 72 学时。本书采用的开发板是野火电子技术有限公司的 STM32F407 开发板,选取的应用项目也是基于此开发板开发的。

对嵌入式应用,要学习的内容很多。学生除了要学习课堂上的内容外,还应该学习如何修改开发板上的例程,以实现其他功能。如果使用其他开发板,只需稍微修改项目中的例程即可。

本书由刘志远、黄远民、易铭、樊亚妮、梁利滨编著,其中项目 6、项目 8、项目 9、项目 10 由佛山职业技术学院的刘志远编写,项目 2、项目 3 由佛山职业技术学院的黄远民编写,项目 1、项目 7 由佛山职业技术学院的易铭编写,项目 4、项目 5 由广东第二师范学院的樊亚妮编写,“认识 STM32”由广东恺鹏勒信息科技有限公司的梁利滨编写并负责本书的校稿。在编写过程中,德国嵌入式专家亚历山大·胡瓦特提供了宝贵的经验与意见。书中编写内容参考了意法半导体 STM32F4 的参考资料与外设芯片的数据手册。在教材编写与项目的例程开发上,得到了东莞野火电子技术有限公司的技术支持,在此致以诚挚的谢意。

由于编者水平有限,书中难免存在疏漏与不当之处,敬请读者批评指正。

另外,本书配有丰富的网络教学资源,读者可以扫描书中二维码观看微课,也可以在佛山职业技术学院的嵌入式应用技术课程资源网站观看教学视频,下载项目例程、数据手册、芯片手册、工具软件等资源。

目 录

认识 STM32	1
0.1 什么是嵌入式系统	1
0.2 ARM	2
0.3 STM32	3
0.4 如何选择 STM32	4
0.5 STM32 开发板	5
0.6 STM32 的内核与片上外设	6
0.7 存储器映射	7
项目 1 开发环境的建立	12
1.1 Keil 的配置	12
1.2 开发板的准备	14
1.3 连接开发板	15
1.4 编写并下载第一个程序	15
1.5 工程文件	23
1.6 小 结	26
项目 2 点亮 LED	27
2.1 STM32F407 的 GPIO	27
2.2 GPIO 的工作模式	30
2.3 配置 GPIO 的寄存器分布	31
2.4 GPIO 选择工作模式	37

2.5	输出低电平	38
2.6	打开 RCC 时钟	38
2.7	编写程序	39
2.8	进一步模块化程序	41
2.9	小 结	49
项目 3	初识 STM32 固件库	50
3.1	获取 STM32 固件库	50
3.2	使用固件库建立一个工程模板	55
3.3	使用固件库模板编程点亮 LED	63
3.4	进一步使用固件库使 LED 闪烁	73
3.5	固件库帮助文件简介	74
3.6	小 结	79
项目 4	使用按键控制 LED	80
4.1	GPIO 作为输入的设置	80
4.2	使用固件库函数操作按键输入	81
4.3	建立按键控制 LED 的工程	82
4.4	小 结	87
项目 5	外部中断的使用	88
5.1	STM32F4××的中断系统	88
5.2	外部中断 EXTI	97
5.3	使用外部中断点亮 LED	101
5.4	小 结	113
项目 6	SysTick 与定时器	114
6.1	SysTick 定时器	114
6.2	STM32 的基本定时器	126
6.3	小 结	136
项目 7	USART 的使用	137
7.1	STM32F4××的串行口	137
7.2	STM32 的 USART	139
7.3	简单 UART 通信实验	156

7.4 实现远程控制 LED	165
7.5 完善串口驱动 usart1.c	167
7.6 小 结	168
项目 8 ADC 的使用	169
8.1 STM32F4 的 ADC	169
8.2 ADC 的固件库	175
8.3 独立模式单通道采集实验	179
8.4 独立模式下混合通道采集实验	185
8.5 小 结	193
项目 9 SPI 接口应用	194
9.1 SPI 接口	194
9.2 STM32 的 SPI	198
9.3 SPI 接口的 Flash	202
9.4 STM32F407 读写 W25Q128	207
9.5 小 结	238
项目 10 I2C 总线的应用	239
10.1 I2C 协议	239
10.2 STM32 的 I2C	242
10.3 使用 I2C 的 EEPROM	246
10.4 I2C 的固件库读写 EEPROM	248
10.5 小 结	276
结束语	277
附 录	284

认识 STM32

0.1 什么是嵌入式系统

嵌入式系统是以应用为中心,软硬件可裁减的,对功能、可靠性、成本、体积、功耗等有严格要求的专用计算机系统。嵌入式系统从概念上比较难以理解,但在生活中处处可见,比如手机、平板、电视等电子产品中都有嵌入式系统。可以简单地认为,嵌入式系统是为各种电子设备设计的专用的微型计算机系统。所谓嵌入式,重点在于裁剪,可以灵活地设计、定制。除上述电子产品之外,嵌入式技术已经应用到各个领域,如航空航天、汽车、仪器仪表、自动化设备、家用电器等。

嵌入式系统从单板机开始,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)阶段,即单片微型计算机阶段,出现在 20 世纪 70 年代。典型的产品有 Intel 的 8031、Zilog 的 Z80 等,如图 0-1 所示,即 MCS-51 系统,在单芯片上集成了 CPU、I/O、存储器、总线等,使用计算机语言进行编程控制。SCM 与通用计算机具有完全不同的思路,它把计算机的几大模块都集成于芯片中。它的设计重点在于各种设备的控制与实现。这个时期的芯片还需要大量的外围扩展芯片来实现各种控制,用一块电路板来完成一个微机控制系统,因此也称为单板机。尽管其在性能上与通用性上与计算机相差甚远,但在家电、仪器仪表、工业控制上受到了广泛的应用,掀起了一股数字化的热潮。SCM 的出现,奠定了嵌入式系统发展的方向。

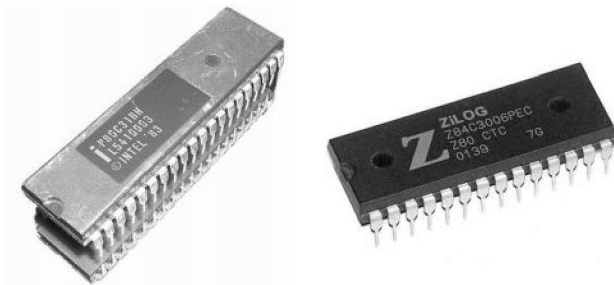


图 0-1 Intel 的 8031 与 Zilog 的 Z80

(2) MCU (Micro Controller Unit) 阶段,即微控制器阶段。为了满足不断扩展的各种外围电路与接口电路,各个厂家不断增强单片机的 CPU 运算能力,加入 I/O 的第二功能(内置各种总线接口、ADC、PWM 等),增加存储器。相对于 SCM,MCU 的性能得到了大大的提升。在这个阶段,MCU 厂家百花齐放,比如国外的 Philips、Atmel 等公司,国内的 STC(宏晶科技)公司,而 Intel 则淡出单片机领域,专心发展通用计算机领域。

(3) SoC (System on Chip) 阶段,即片上系统阶段。单片机技术发展的初衷是寻求应用系统在芯片上的最大化集成,随着微电子技术、IC 设计、EDA 工具的发展,SoC 将数字信号处理器、RISC (Reduced Instruction Set Computer) 处理器、存储器、I/O、半定制电路集中到单芯片中,使得具有高性能运算能力的嵌入式系统成为可能。目前我们所说的嵌入式系统基本都是基于 SoC。



0.2 ARM

提到嵌入式就不得不提起 ARM 了。ARM (Advanced RISC Machines) 是微处理器行业的一家知名企业,设计了大量高性能、廉价、低功耗的 RISC 处理器,以及相关技术和软件。ARM 本身并不生产芯片,而是将其技术授权给世界上许多著名的半导体、软件和 OEM 厂商,如苹果、高通、华为等。ARM 负责提供一系列内核、体系扩展、微处理器和系统芯片方案。由于所有产品均采用一个通用的软件体系,所以相同的软件可在所有产品中运行,比如华为的海思麒麟和高通的骁龙是各自设计的 SoC,但都使用了 ARM 内核,因此都能完全兼容安卓系统。

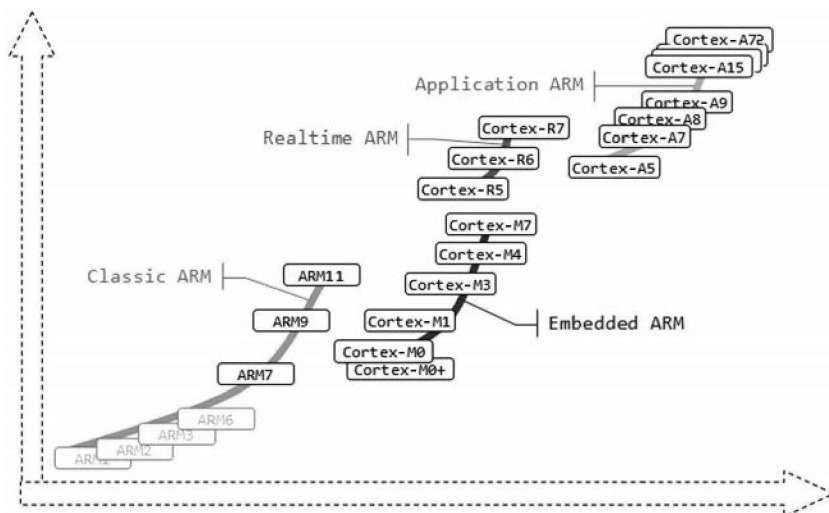


图 0-2 ARM 产品分布图

ARM 设计的处理器类型很多,涵盖了从低端、低功耗到高性能、人工智能与机器学习等的各种处理器,早期的有 ARM7、ARM9、ARM11,最新的有 Cortex-M、Cortex-R、Cortex-A

等。ARM 产品分布图如图 0-2 所示。其中 Cortex 系列的 M 代表了低功耗,R 代表高性能与实时性,A 代表最高性能,对操作系统进行优化。Cortex 系列的涵盖面非常广,尤其是 M 系列,各个授权的半导体厂家不断地降低成本与售价,它已经延伸到传统 MCU 的领域。总的来说,ARM 是嵌入式领域的领跑者。

0.3 STM32

STM32 是意法半导体 (SGS-THOMSON Microelectronics, 简称 STMicroelectronics) 公司 (以下简称“ST 公司”)生产的系列微处理器,包含了基于 ARM Cortex-M0、Cortex-M0+、Cortex-M3、Cortex-M4 及 Cortex-M7 内核并具备丰富外设选择的 32 位微处理器,目前提供 12 大产品线 (F0、F1、F2、F3、F4、F7、H2、L0、L1、L4、L4+、WB),超过 800 个型号,如图 0-3 所示。STM32 产品广泛应用于工业控制、消费电子、物联网、通信设备、医疗服务、安防监控等应用领域,其优异的性能进一步推动了生活和产业智能化的发展。

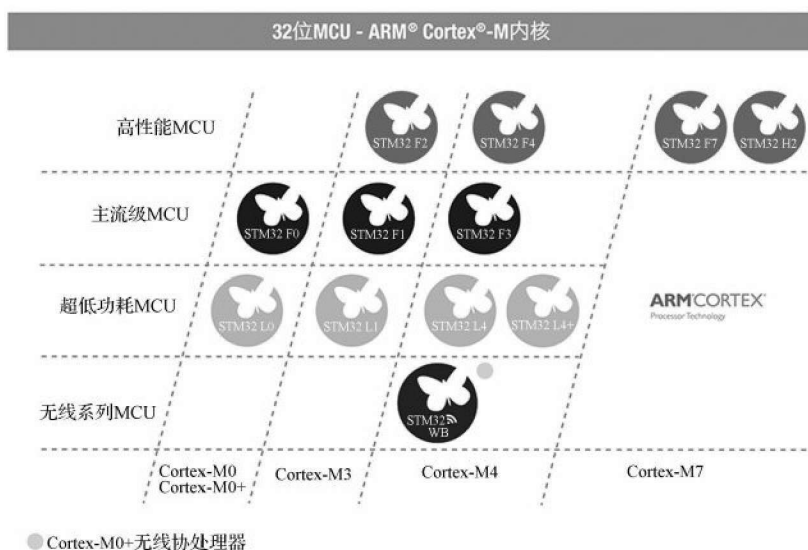


图 0-3 STM32 系列产品分布图

比较常见的 STM32F103 系列属于 F1 系列,基于 Cortex-M3 内核,属于低端产品,价格低廉,在性能上可直接与传统 MCU 进行竞争。另一个常见的系列为 STM32F407 系列,属于 F4 系列,基于 Cortex-M4 内核,性能较强,接口丰富,属于中端产品,有较强的运算处理能力。由于内核均采用了 Cortex,所以它们的编程方法基本都是一样的。学会了 F1 系列的编程,同样也就学会了 F4 系列的编程,反之亦然。

0.4 如何选择 STM32

STM32 系列产品丰富。当我们设计产品时应根据产品性能、引脚数量、使用环境、成本等的需求选择相应的微处理器。STM32/8 部分芯片特性如表 0-1 所示。图 0-4 是 STM32/8 官方的详细命名规定。以 STM32F103RCT6 为例说明,它是 STM32 的 103 基础型,有 64 个引脚、256KB 的 Flash,采用 QFP 封装,工作温度为 $-40^{\circ}\text{C} \sim 85^{\circ}\text{C}$,如表 0-2 表示。由于产品十分丰富,选型应多参考官方数据手册。

表 0-1 STM32/8 部分芯片特性

CPU 位数	内核	系列	描述
32	Cortex-M0	STM32-F0	入门级
		STM32-L0	低功耗
	Cortex-M3	STM32-F1	基础型,主频 72MHz
		STM32-F2	高性能
		STM32-L1	低功耗
	Cortex-M4	STM32-F3	高性能,主频 180MHz
		STM32-F4	高性能,主频 180MHz
STM32-L4		低功耗	
8	超级版 6502	STM32-F7	高性能
		STM8S	标准系列
		STM8AF	标准系列的汽车应用设计
		STM8AL	低功耗的汽车应用设计
		STM8L	低功耗

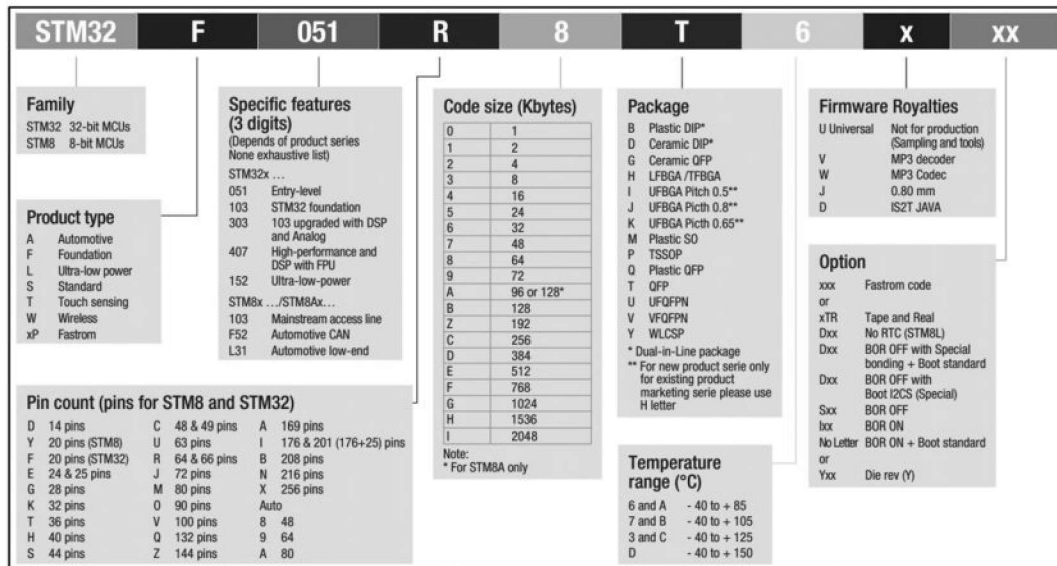


图 0-4 STM32/8 系列命名含义

表 0-2 STM32F103RCT6 命名说明

-	STM32	F	103	R	C	T	6
家族	STM32 表示 32bit 的 MCU						
产品类型	F 表示基础型						
具体特性	基础型						
引脚数目	R 表示 64Pin。其他常用的有: C 表示 48Pin, V 表示 100Pin, Z 表示 144Pin, B 表示 208Pin, N 表示 216Pin						
Flash 大小	C 表示 256KB, 其他常用的有: B 表示 128KB, E 表示 512KB, I 表示 2048KB						
封装	T 表示 QFP 封装, 这是最常用的封装方式						
温度	6 表示温度等级为 A, 工作温度为 $-40^{\circ}\text{C} \sim 85^{\circ}\text{C}$						

0.5 STM32 开发板

和学习单片机一样,所有的程序都需要下载到 MCU 进行验证。学习单片机时可以使用 Protues 软件仿真,也可以使用单片机开发板进行实体下载验证。嵌入式学习也一样。目前支持软件仿真的 ARM 产品很少。即使是最新版的 Protues 8.7,对 ARM 产品的支持也非常有限,因此我们必须使用开发板进行学习。STM32 的开发板很多,尤其是 F1 与 F4 系列,有最小系统,也有扩展齐全、接口丰富的开发板。图 0-5 所示的 F103 系列开发板,价格从十几元到几百元不等。用户可以根据实际情况选择,建议选择配备常用接口,比如串行口、SWD/JTAG 接口(JLINK)、液晶、按键、LED 等,如果预算足够,可以选择自带外设的开发板,这样使用起来更方便些。

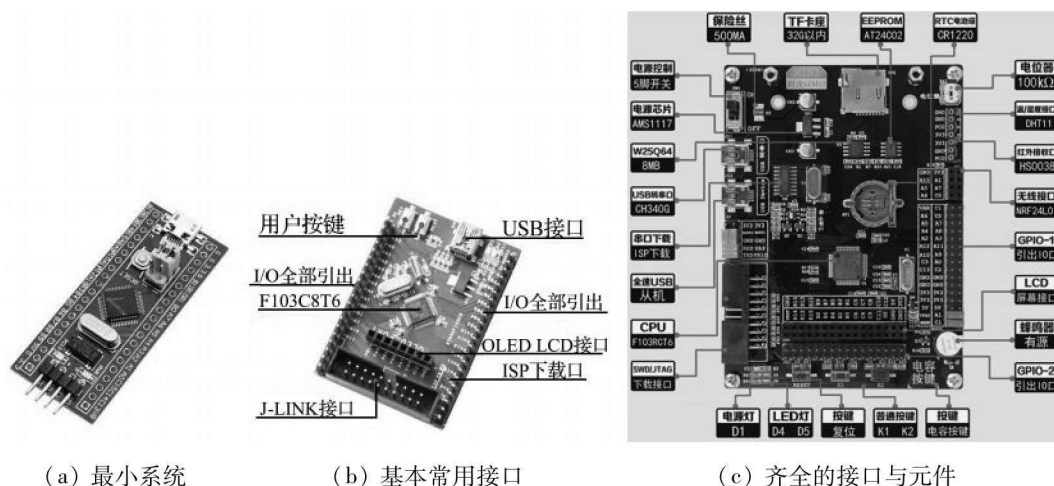


图 0-5 各类 STM32F103 开发板

市场上尽管有各种开发板,但是硬件内核都是一样的,因此不影响我们学习。需要了

解的是,STM32 可以通过计算机串行口使用 ISP 方式下载程序。图 0-5(a)所示的最小系统只有 ISP 接口,这一点与 STC 的单片机一样,比较烦琐,速度也比较慢。STM32 还可以采用 SWD/JTAG 方式下载,速度比 ISP 要快很多,而且可以使用开发软件的同步仿真调试功能,非常适合我们学习 STM32。对于初学者而言,不建议使用 ISP 方式,而建议采用 SWD/JTAG 方式,购买开发板时应选择带 SWD/JTAG 接口的,如图 0-5(b)和图 0-5(c)所示。往往 SWD 与 JTAG 接口都集成到一个接口上,各个厂家都采用统一的引脚封装,以提高兼容性。使用者不论是采用 SWD 还是采用 JTAG,都可以直接插上去使用。但是使用 SWD/JTAG 时必须另外购买仿真器。这里介绍几种常见的 SWD 仿真器。

(1) J-LINK。J-LINK 是德国 SEGGER 公司推出的基于 JTAG 的仿真器,如图 0-6(a)所示。它完成了从软件到硬件的转换工作。J-LINK 是一个通用的开发工具,可以用于 Keil、IAR、ADS 等平台,支持大部分 ARM,兼容性好。

(2) U-LINK。U-LINK 是 ARM/Keil 公司推出的仿真器。U-LINK/U-LINK2 可以配合 Keil 软件实现仿真功能,增加了串行调试(SWD)支持,如图 0-6(b)所示。U-LINK 支持大部分 ARM 产品。但要注意的是,U-LINK 是 Keil 公司开发的仿真器,专用于 Keil 平台上,在 ADS、IAR 平台上不能使用。

(3) ST-LINK。ST-LINK 是专门针对 ST 公司 STM8 和 STM32 系列芯片的仿真器,如图 0-6(c)所示。ST-LINK/V2 使用的是 SWIM 标准接口和 SWD/JTAG 标准接口。同样地,ST-LINK 只能用于 ST 公司的微处理器,但能用于 Keil、IAR、ADS 等平台上。



图 0-6 三种基于 SWD/JTAG 接口的仿真器

本教材采用的开发平台是 Keil MDK 5.24 与 STM32F407,因此,读者采购以上三种仿真器中的任意一种都可以进行仿真与程序下载。

0.6 STM32 的内核与片上外设

STM32 的内核采用的是 Cortex-M4 内核,由 ARM 公司授权,剩下的存储器与片上外设,比如 GPIO、USART、I2C 等全部由 ST 公司设计,因此,无论是 F1 系列还是 F4 系列,都是由 ST 公司生产的 ARM 内核的 SoC,其框架图如图 0-7 所示。

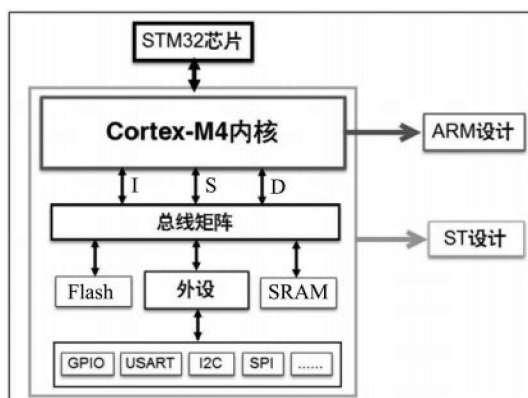


图 0-7 STM32 内部框架图

STM32 的内核、存储器与片上外设通过一个总线矩阵进行管理访问，主系统由 32 位多层 AHB(高级高性能总线)总线矩阵构成，比如 STM32F407 中主控总线有 8 条(S0 ~ S7)，被控总线有 7 条(M0 ~ M6)，如图 0-8 所示。总线之间交叉的时候若有圆圈，则表示可以通信；若没有圆圈，则表示不可以通信。借助总线矩阵，可以实现主控总线到被控总线的访问，这样使得系统即使在多个高速外设同时运行期间，也可以实现并发访问和高效运行。

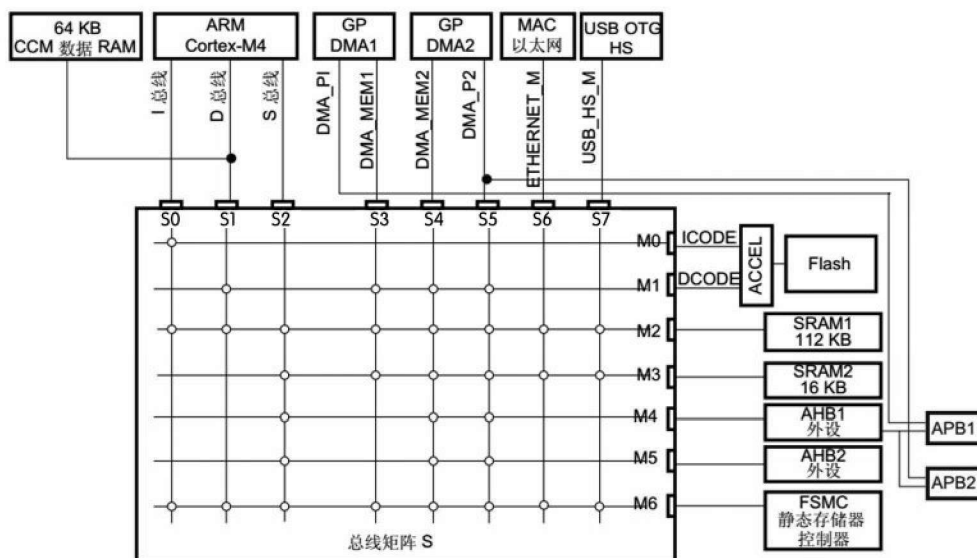


图 0-8 STM32F407 的总线矩阵

0.7 存储器映射

计算机对所有设备的管理都是通过地址进行操作的。每个设备都具有自己唯一的地址，但每一个设备只是一个元件，本身不具备地址信息，它们的地址由芯片厂商分配。给存

存储器分配地址的过程,称为存储器映射。

如图 0-9 所示,被控总线连接的是 Flash、RAM 和片上外设。这些存储器与外设的功能部件共同排列在一个 4GB 的地址空间内。如何分配这些设备与 4GB 的地址空间呢?在这 4GB 的地址空间中,ARM 已经粗略地平均分成了 8 个块,每个块为 512MB,每个块也都规定了用途,如图 0-9 所示。当然,并不是所有地址都会用完,那些未分配给片上存储器和外设的所有存储区域均被视为“预留区”。

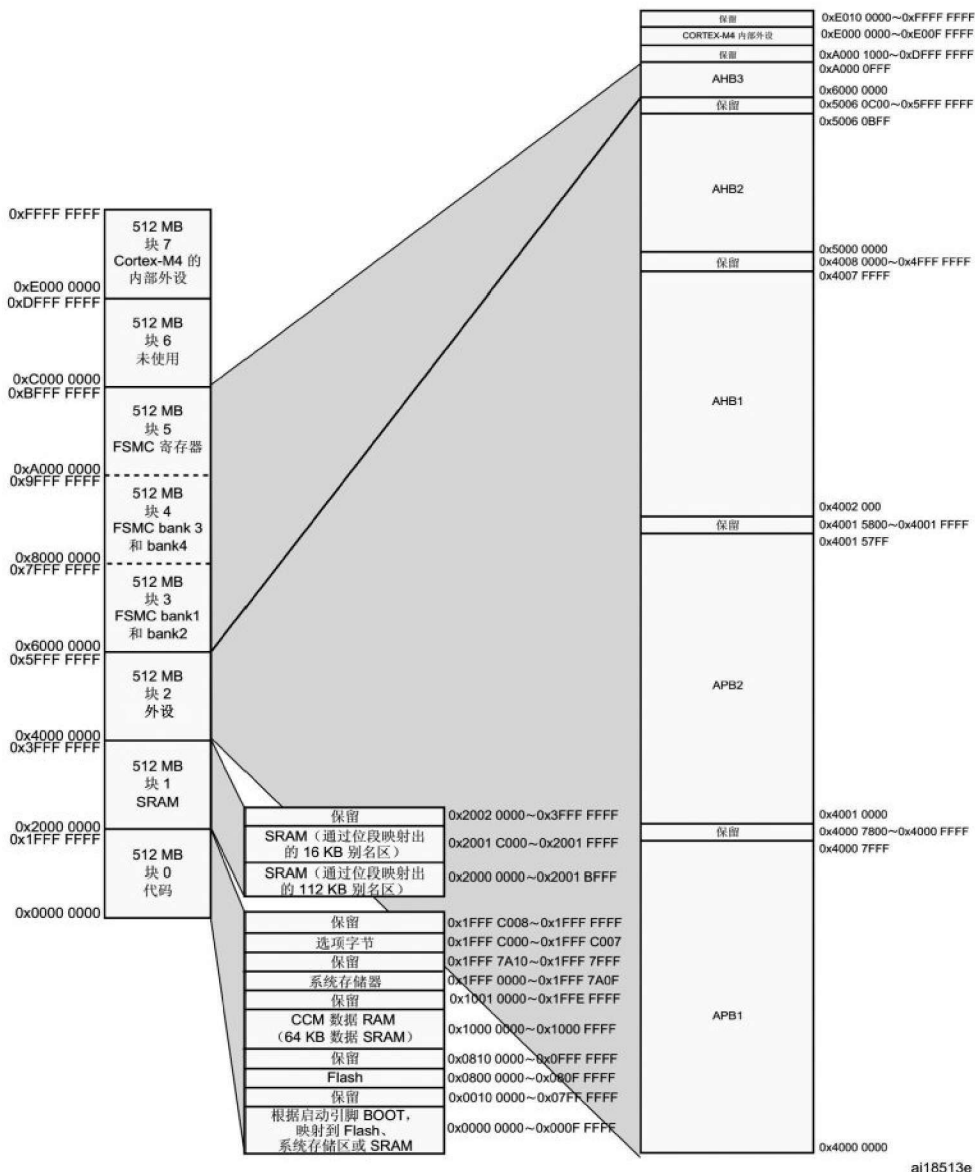


图 0-9 STM32F4 的寄存器映射分布

这里简要地把 8 个块映射的内容列成表 0-3。在这 8 个块里面有 3 个块是最重要的: