

前 言

意法半导体(STMicroelectronics, ST)公司是世界上最大的半导体公司之一,在很多领域居世界领先水平。其中,STM32 系列微控制器凭借其产品多样化、性价比高、开发简单易学等优点,成为应用最广的 ARM 器件之一。STM32F1 系列属于中低端的 32 位 ARM 微控制器,该系列芯片由 ST 公司出品,其内核是 Cortex-M3。该系列芯片按片内 Flash 的大小可分为三大类:小容量(16KB 和 32KB)、中容量(64KB 和 128KB)和大容量(256KB、384KB 和 512KB)。芯片集成 TIM、CAN、ADC、SPI、I2C、USB、UART 等多种片上资源,适合测控系统的开发。

为了使读者更加快速并深入地掌握 ARM 微控制器的设计与开发,本书以具有代表性的 ST 公司产品中的 STM32F103ZET6 为例,并结合最新的库开发方式,全面系统地介绍 ARM 微控制器的原理和应用。本书首先从 STM32F103ZET6 的硬件基础入手,分析 STM32 内核架构的组成;接着讲述 STM32 最小应用系统,以及 STM32 的几种常用接口电路及引脚配置等;然后介绍基于 Keil 软件的开发流程,给出片内常用模块原理及使用方法;最后以两轮自平衡小车控制系统设计为例详细讲述 STM32 应用系统的软硬件设计方法。

全书共 9 章。第 1 章为绪论,简要介绍 STM32 处理器的特点、应用领域以及学习方法。第 2 章讲述 STM32 内核架构,包括内部功能结构、寄存器组及 CPU 运行模式、存储器和存储空间、独特的 Thumb-2 指令集、中断和异常、调试支持等。第 3 章为 STM32 硬件基础,内容涵盖 STM32 最小应用系统设计、串行数据通信接口电路设计、人机接口及显示电路设计、STM32 引脚配置、STM32 芯片的命名规则和选型问题。第 4 章为软件开发基础,主要以如何点亮一个 LED 为例介绍 STM32 软件开发流程、集成开发环境及其应用。第 5 章为库函数及应用,主要包括库函数简介、库函数应用、LED 闪烁例程程序剖析。第 6 章为常用外设模块及其应用开发,包括通用数字输入/输出模块、NVIC 与中断控制、显示与键盘模块、模/数转换模块、DMA 控制器模块。第 7 章为时间模块及其应用开发,主要讲述 SysTick 定时器、实时时钟、基本通用定时器模块、定时器脉冲宽度调制、定时器互补死区等。第 8 章为通信外设模块及其应用开发,包括串行通信模块、SPI 通信模块、I2C 通信模块、局域网控制器、可变静态存储控制器。第 9 章为 ARM 在自平衡小车中的应用,系统介绍两轮自平衡小车控制系统、两轮自平衡小车控制系统硬件设计、两轮自平衡小车控制系统软件设计。一般章节均配有内容概述和实验例程,并在最后附有习题与思考题,以便读者实践。

本书内容系统翔实,章节安排结构合理,讲解通俗易懂,便于读者循序渐进地推进 STM32 的学习与开发,其由浅入深的安排尤其适合在校学生和科研机构开发人员学习使用。希望读者在学习过程中尝试更多的练习与实践,这样会更快地领悟和吸收书中的知识。

本书由周翟和主编,姚睿、陆熊参编,其中第 3、4 章由陆熊编写,第 5、6 章由姚睿编写并参与配套实验的软硬件开发与校稿,其余章节由周翟和编写并统稿。研究生胡佳佳、沈

超、赵庆涛、虞波、陈如意、吕建新、王锋、崔培林、张倩云、钟雨露、刘硕绮、马静敏等做了大量辅助性工作。

本书列入了南京航空航天大学“十四五”规划教材。在本书编写过程中参考了大量的中外文献和资料，在此一并致谢。

由于编者水平有限，书中难免存在不当之处，恳请读者批评指正。

编 者

2026 年 2 月

目 录

第 1 章 绪论	1
1.1 概述	1
1.2 典型 STM32 处理器简介	1
1.3 STM32 系列 MCU 的优点	2
1.4 STM32 应用领域	4
1.5 STM32 学习方法	5
习题与思考题	5
第 2 章 STM32 内核架构	6
2.1 概述	6
2.2 内部功能结构	7
2.2.1 总体结构及功能模块概述	7
2.2.2 总线结构	8
2.2.3 流水线	9
2.3 寄存器组及 CPU 运行模式	10
2.3.1 寄存器组	10
2.3.2 CPU 运行模式	12
2.4 存储器和存储空间	13
2.4.1 存储空间映射	13
2.4.2 存储器的各种访问属性	14
2.4.3 位带技术	15
2.5 独特的 Thumb-2 指令集	16
2.5.1 指令集开发	17
2.5.2 Thumb-2 指令集体系结构	17
2.6 中断和异常	18
2.6.1 NVIC 概览	18
2.6.2 中断的进入与退出	18
2.6.3 中断嵌套	19
2.6.4 尾链技术	19
2.6.5 晚到异常	19
2.7 调试支持	19
习题与思考题	20
第 3 章 STM32 硬件基础	21
3.1 概述	21
3.2 STM32 芯片的引脚选型和配置	21

3.2.1	STM32 芯片的命名规则和选型	21
3.2.2	STM32 芯片的引脚配置	22
3.3	STM32 最小应用系统设计	25
3.3.1	电源电路设计	25
3.3.2	复位电路设计	27
3.3.3	时钟电路设计	27
3.3.4	JTAG 接口电路设计	28
3.3.5	外部存储器扩展	28
3.4	外设板硬件资源设计	30
3.4.1	串行通信接口	31
3.4.2	串行外设接口	32
3.4.3	USB 接口	33
3.4.4	CAN 总线控制器接口	33
3.4.5	键盘接口电路	34
3.4.6	数码管显示电路	35
3.4.7	LCD 及其接口电路	36
3.4.8	步进电机	38
3.4.9	SD 卡接口	38
3.4.10	其他外设资源	39
	习题与思考题	41
第 4 章	软件开发基础	42
4.1	概述	42
4.2	集成开发环境及其应用	42
4.2.1	开发工具简介	42
4.2.2	固件库简介	43
4.2.3	Keil MDK 的安装	44
4.2.4	J-LINK 驱动安装	45
4.3	STM32 应用程序开发调试实例	46
4.3.1	STM32 工程建立	46
4.3.2	LED 闪烁代码编写及分析	56
4.3.3	配置 J-LINK 硬件调试	57
4.3.4	STM32 程序下载	57
	习题与思考题	59
第 5 章	库函数及应用	60
5.1	概述	60
5.2	库函数简介	61
5.3	库函数应用实例	62
5.4	深入剖析 LED 闪烁例程	65
5.4.1	LED 闪烁例程说明	65

5.4.2	LED 闪烁例程分析与实验	65
5.4.3	程序分析	68
	习题与思考题	72
第 6 章	常用外设模块及其应用开发	73
6.1	概述	73
6.2	通用数字输入/输出模块	73
6.2.1	GPIO 模块结构与工作原理	73
6.2.2	GPIO 模块寄存器说明	75
6.2.3	实验部分	78
6.3	NVIC 与中断控制	83
6.3.1	STM32 中断优先级	83
6.3.2	中断的使能与除能	84
6.3.3	中断的挂起与清除	85
6.3.4	中断建立全过程的演示	86
6.3.5	实验部分	86
6.4	显示与键盘模块	91
6.4.1	数码管显示	91
6.4.2	LCD 显示与键盘模块	96
6.5	模/数转换模块	101
6.5.1	ADC 模块结构及其工作原理	103
6.5.2	ADC 模块寄存器	103
6.5.3	实验部分	108
6.6	DMA 控制器模块	114
6.6.1	DMA 模块结构及其工作原理	115
6.6.2	DMA 模块寄存器	116
6.6.3	实验部分	117
	习题与思考题	122
第 7 章	时间模块及其应用开发	123
7.1	概述	123
7.2	时钟系统	123
7.2.1	时钟源	123
7.2.2	时钟树	124
7.3	SysTick 定时器	126
7.3.1	SysTick 定时器时钟	126
7.3.2	SysTick 定时器工作分析	127
7.3.3	SysTick 控制寄存器	127
7.3.4	实验部分	129
7.4	实时时钟	132
7.4.1	RTC 模块结构	132

7.4.2	UNIX 时间戳	134
7.4.3	实验部分	134
7.5	基本与通用定时器模块	144
7.5.1	基本定时器功能与工作原理	144
7.5.2	通用定时器功能与工作原理	145
7.5.3	TIMx 寄存器	147
7.5.4	实验部分	151
7.6	定时器脉冲宽度调制	154
7.6.1	定时器脉冲宽度调制输出工作原理	154
7.6.2	PWM 相关寄存器	155
7.6.3	实验部分	159
7.7	高级定时器	165
7.7.1	高级定时器工作原理	165
7.7.2	刹车和死区寄存器(TIMx_BDTR)	167
7.7.3	实验部分	168
	习题与思考题	171
第 8 章	通信外设模块及其应用开发	172
8.1	概述	172
8.2	USART 通信模块	172
8.2.1	串行通信模块的结构与工作原理	172
8.2.2	串行通信模块的寄存器	174
8.2.3	实验部分	176
8.3	SPI 通信模块	182
8.3.1	SPI 通信模块的结构与工作原理	182
8.3.2	SPI 通信模块的寄存器	184
8.3.3	实验部分	189
8.4	I2C 通信模块	197
8.4.1	STM32 的 I2C 通信模块简介	197
8.4.2	I2C 协议简介	198
8.4.3	I2C 通信模块的寄存器	199
8.4.4	实验部分	205
8.5	局域网控制器	213
8.5.1	CAN 协议简介	213
8.5.2	STM32 的 CAN 特性及架构	216
8.5.3	实验部分	217
8.6	可变静态存储控制器 FSMC	226
8.6.1	FSMC 模块的结构与工作原理	226
8.6.2	FSMC 的寄存器	227
8.6.3	实验部分	229

习题与思考题	233
第 9 章 ARM 在自平衡小车中的应用	234
9.1 两轮自平衡小车控制系统	234
9.1.1 控制原理	234
9.1.2 控制系统组成结构	235
9.2 两轮自平衡小车控制系统硬件设计	236
9.2.1 控制系统硬件结构	236
9.2.2 控制系统硬件原理	237
9.2.3 引脚配置说明	248
9.3 两轮自平衡小车控制系统软件设计	249
9.3.1 软件总体设计	249
9.3.2 部分模块软件设计	250
9.3.3 姿态融合算法软件设计	255
习题与思考题	259
参考文献	260

第 1 章 绪 论

1.1 概 述

STM32 系列是为高性能、低成本、低功耗的嵌入式应用专门设计的 ARM Cortex-M3 (以下简称 CM3) 内核, 按性能不同又分成两个不同的系列: STM32F103 增强型系列和 STM32F101 基本型系列。增强型系列的时钟频率达到 72MHz, 是同类产品中性能最高的产品; 基本型系列的时钟频率为 36MHz, 以 16 位产品的价格得到比 16 位产品大幅提升的性能, 是 16 位产品用户的最佳选择。两个系列都内置 32~128KB 的闪存, 不同的是 SRAM 的最大容量和外设接口的组合不同。时钟频率为 72MHz 时, 从闪存执行代码, STM32 的功耗为 36mA, 是市场上 32 位功耗最低的产品, 相当于 0.5mA/MHz。

STM32 已经发布了超过 75 个不同的型号。这些型号分成四组: 中央处理器 (central processing unit, CPU) 时钟频率高达 36MHz 的基本型、CPU 时钟频率高达 72MHz 的增强型、增加通用串行总线 (universal serial bus, USB) device 外设并且运行在时钟频率为 48MHz 的 USB 型, 以及互联型。互联型增加的先进通信外设包括以太网 MAC 和 USB HOST/OTG 控制器。所有型号 STM32 的引脚和软件兼容, 并且提供容量高达 512KB 的 Flash ROM 和 64KB 的 SRAM。由最初的版本到现在, STM32 已经可以扩展到更大的 RAM 和 Flash ROM, 以及具有更加复杂外设的设备。

STM32 也是一款低功耗、高性能的微控制器。它可以以 2V 的供电电压运行在 72MHz 效率处, 所有功能运行起来仅消耗 36mA。结合 Cortex-M3 的低功耗模式, STM32 的待机功耗仅为 2 μ A。同时, 一个内部 8MHz RC 振荡器的存在, 使该芯片能够在外部振荡器处于启动的情况下, 迅速退出低功耗模式。这种快速进入和退出低功耗模式的优点进一步降低了整体功耗。

1.2 典型 STM32 处理器简介

STM32 系列处理器目前分为两个系列, 如图 1.1 所示。STM32F101 是基本型系列, 工作在 36MHz 频率处; STM32F103 是增强型系列, 工作在 72MHz 频率处, 带有更多片内 RAM 和丰富的外设。两个系列的产品拥有相同的片内 Flash 选项, 在软件和引脚封装方面可兼容。

基本型系列是 STM32 处理器的入门产品, 其价格仅相当于 16 位的 MCU, 却拥有 32 位 MCU 的性能, 其外设的配置能提供优秀的控制和连接能力。增强型系列产品则将 32 位 MCU 的性能和功效引向一个新的级别, 内含的 Cortex-M3 内核工作在 72MHz 频率处, 能实现高端的运算, 且其外设的配置可以带来极好的控制和连接能力。

STM32 全系列处理器拥有脚对脚、外设及软件的高度兼容性。这给其应用带来了全方位的灵活性, 可以在不必修改原始框架及软件的前提下, 将应用升级到需要更多的存储空间,

或精简到使用更少的存储空间，或改用不同的封装规格。

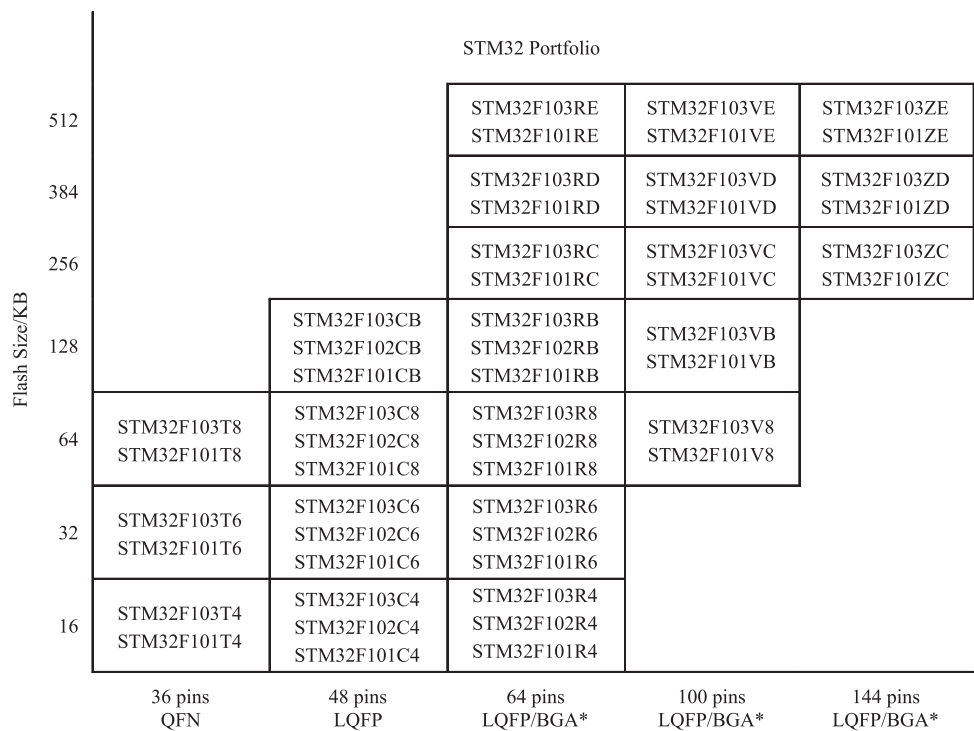


图 1.1 STM32 系列处理器的分类

对于使用同一平台进行多个项目的开发，STM32 是一种非常好的选择。因为在 STM32 全系列产品中，既有适合仅需少量的存储空间和引脚的，也有满足需要更多的存储空间和引脚的；既有适合高性能应用的，又有满足低功耗要求的；既有适合低成本简单应用的，也有满足高端复杂应用的。STM32 产品全系列兼容，这个优点使项目之间的代码重用和代码移植变得非常方便。

1.3 STM32 系列 MCU 的优点

1. 先进的内核结构

STM32 系列使用了 ARM 最新的、具有先进架构的 Cortex-M3 内核。Cortex-M3 是一个 32 位的处理器内核，采用哈佛结构，拥有独立的指令总线 and 数据总线，可以让取指与数据访问并行。

2. 优秀的功耗控制

高性能并非意味着更加耗电，STM32 处理器经过特殊处理，针对市场上主要的 3 种能耗需求进行了优化。

(1) 在运行模式下，使用高效率的动态耗电机制，当代码在 Flash 中以 72MHz 全速运行时，如果外部时钟开启，处理器仅消耗 27mA 的电流。

(2) 在待机状态下保持极低的电能消耗，典型的耗电值仅为 $2\mu\text{A}$ 。

(3) 在使用电池供电时，提供 $2.0\sim 3.6\text{V}$ 的低电压工作能力。

STM32 处理器具有 3 种低功耗模式和灵活的时钟控制机制，用户可以根据自己所需的耗电性能要求进行合理的优化。STM32 还内嵌了实时时钟 (RTC)，它既可由 32kHz 外部晶体提供频率基准，也可由内部 RC 电路提供频率基准。RTC 有其单独的供电电路，内置的开关使其既可使用外部纽扣电池供电，又可由主电源供电。在 3.3V 的供电电压下，其典型的消耗电流仅为 $1.4\mu\text{A}$ 。另外，RTC 中还包含用于数据备份的 20B RAM。

STM32 处理器从停机模式唤醒通常只需要不到 $7\mu\text{s}$ 的时间，而从待机或复位状态启动通常只需 $55\mu\text{s}$ 就可以进入运行状态。

3. 性能优越而且功能创新的片上外设

STM32 处理器片上外设的优势来源于双 APB 总线结构，其中有一个高速 APB (速度可达 CPU 的运行频率)，使连接到该总线上的外设能以更高的速度运行。

(1) USB 接口可达到 12Mbit/s ;

(2) USART 接口高达 4.5Mbit/s ;

(3) SPI 接口可达 18Mbit/s ;

(4) I2C 接口频率可达 400kHz ;

(5) GPIO 的最大翻转频率为 18MHz ;

(6) PWM 定时器最高可使用 72MHz 时钟输入。

针对 MCU 应用中最常见的电机控制，STM32 对片上外设进行了一些功能创新。STM32 增强型系列处理器内嵌了非常适合三相无刷电机控制的定时器和 ADC，其高级 PWM 定时器具有以下功能：

(1) 6 路 PWM 输出；

(2) 产生带死区时间的 PWM 信号；

(3) 边沿对齐和中心对称波形；

(4) 紧急故障停机、可与两路 ADC 及其他定时器同步；

(5) 可编程防范机制可用于防止对寄存器的非法写入；

(6) 编码器输入接口；

(7) 霍尔传感器接口；

(8) 完整的向量控制环。

以上专门的外围电路与高性能 Cortex-M3 内核相结合，可将完整的向量控制环软件执行时间缩短为 $21\mu\text{s}$ (无传感器模式、三相永磁同步电机 (Permanent Magnet Synchronous Motor, PMSM))。当电流采样频率为 10kHz 时，CPU 的工作负载低于 25% ，这样，处理器还可以执行电机控制之外的其他任务。

4. 高度的集成整合

STM32 处理器最大限度地实现集成，尽可能地减少对外部器件的要求。STM32 封装如图 1.2 所示。

(1) 内嵌电源监控器，带有上电复位、低电压检测、掉电检测、自带时钟的看门狗定时器。

(2) 一个主晶振可以驱动整个系统。低成本的 $4\sim 6\text{MHz}$ 晶振即可驱动 CPU、USB 以及所有外设；内嵌 PLL 可产生多种频率；可以为内部实时时钟选择 32kHz 的晶振。

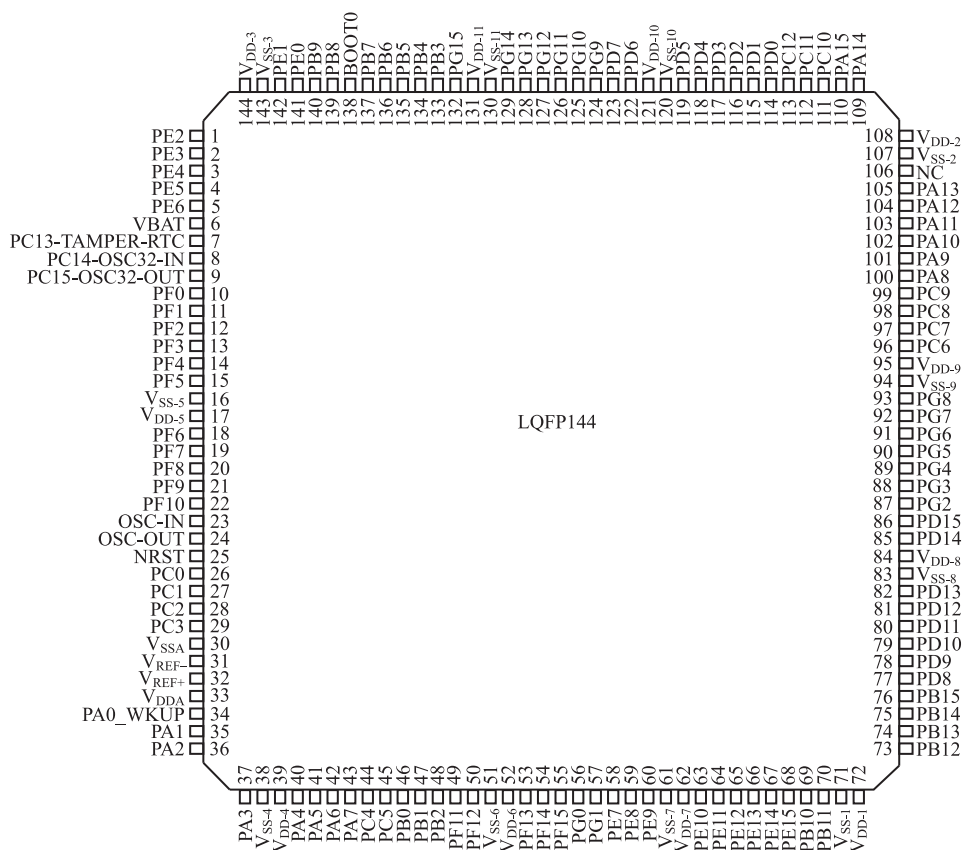


图 1.2 STM32 系列 144 引脚 LQFP 封装顶视图

(3) 内嵌精确的 8MHz RC 振荡电路，可用作主时钟源，还有针对 RTC 或看门狗的低频率 RC 电路。

(4) LQFP100 封装芯片的最小系统只需 7 个滤波电容作为外围器件。

(5) 易于开发。STM32 系列处理器易于开发，可使产品快速进入市场。

1.4 STM32 应用领域

纵观 MCU 市场，主流的构架方式可以分为两类：自主知识产权构架和基于 ARM 构架的 MCU。自主知识产权构架主要包括英特尔的 X86 构架、飞思卡尔的 Power PC、Microchip 的 PIC 系列、Atmel 的 AVR 系列等，基于 ARM 构架的 MCU 由众多开发基于 ARM 构架的厂商组成，包括 ST、NXP、NEC 以及收购了 Luminary 的 TI 等。

2007 年 6 月，ST 公司发布第一款基于 Cortex-M3 并内嵌 32~128KB 闪存的 STM32 微控制器系列产品。它的应用领域十分广泛，一种结构可以覆盖低功耗、高性能和低成本等多种产品需求。在销售终端(银行读卡机、收银机等)、身份识别设备(公路自动收费系统、安全和生物特征识别等)、工业自动化(现场数据采集器、电表等)、消费类电子(计算机外设、游戏手柄等)、建筑安全防护/消防/供热通风与空气调节(Heating, Ventilation Air Conditioning,

HVAC) (报警系统、控制面板等)、医疗领域(心脏监控、便携式测试仪器等)、通信领域(3G 基站监控、光纤接入控制等)、家电(电动自行车、洗衣机等)、仪表(电子秤、电表等)中都可以看到 STM32。由此可见，STM32 应用于市面上一半以上的电子产品。

1.5 STM32 学习方法

学习 STM32 控制器，一般可以遵循如图 1.3 所示的学习流程。首先了解其结构框架和性能概述，掌握其特点；其次要学习其硬件基础，包括 CPU 结构、存储器配置及最小系统支持模块等；接下来要掌握其软件开发方法，熟悉其仿真软件和开发环境，掌握其寄存器组；然后，学习自己需要用到的片内外设资源的工作原理和编程方法；最后根据 STM32 应用系统设计方法，进行软、硬件设计和系统调试。

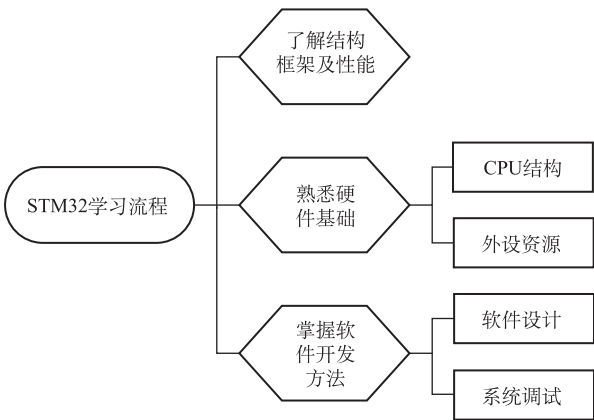


图 1.3 STM32 学习流程

在学习 STM32 的过程中应该多实践、多写程序、多看代码、多问问题，这也是学习微处理器的通用方法。

习题与思考题

- 1.1 STM32 采用哪种内核？该内核有何特点？
- 1.2 STM32 系列如何分类？
- 1.3 简述 STM32 系列 MCU 的优点。
- 1.4 简述哈佛结构的特点。
- 1.5 简述 STM32 处理器的发展历史和发展趋势。
- 1.6 简述 STM32 处理器的典型应用领域。
- 1.7 目前市场上的主要 ARM 厂商有哪些？
- 1.8 简述 STM32 的主要性能。

第 2 章 STM32 内核架构

2.1 概 述

STM32 是基于 Cortex 内核的微处理器。目前，Cortex 内核已经成为 ARM 公司最新一代嵌入式处理的核心，Cortex 处理器具有一个完整的处理核心，包括 Cortex CPU 和围绕在其周围的一系列系统设备。Cortex-M3 是一个 32 位处理器内核，体现在内部数据路径、寄存器、存储器接口都是 32 位的。内核采用哈佛结构，独立的指令总线和数据总线可以让取指和数据访问并行处理。同时为了适应比较复杂的应用情况，需提供更多的存储系统功能，为此 Cortex-M3 提供了一个可选的 MPU。而 Cortex-M3 处理器相对于早期的 ARM 处理器的一个关键性进步在于，它为开发人员提供了一个标准的既快速又具备绝对性的中断系统结构。此外，Cortex-M3 内部还具有很多调试组件，可用于在硬件水平上支持调试操作，如指令断点、数据观察点等。图 2.1 为 Cortex-M3 的简化视图。

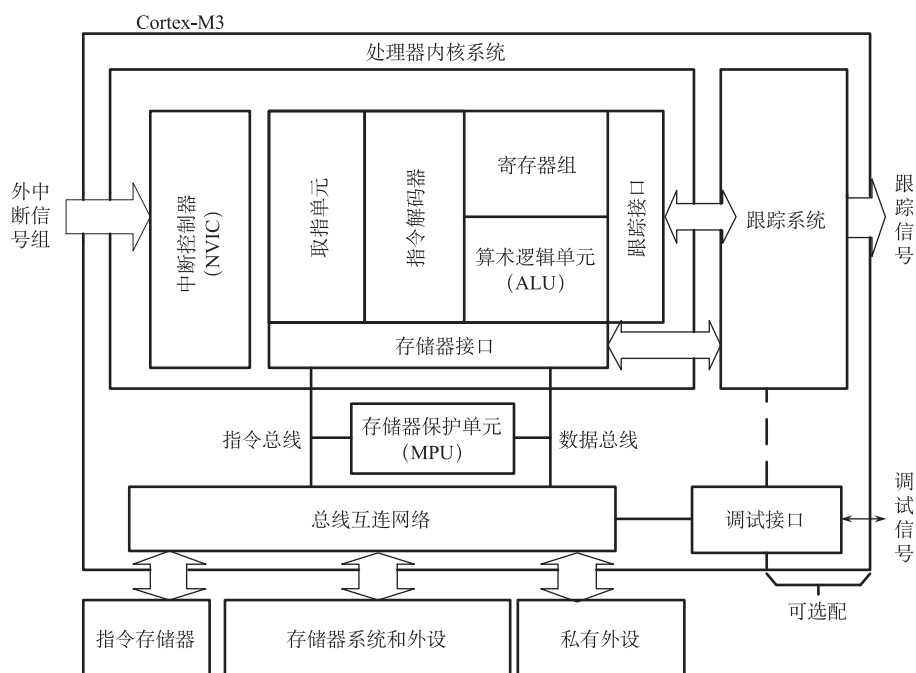


图 2.1 Cortex-M3 的简化视图

本章将从内核的总体结构、CPU、存储器、指令集、中断系统以及调试支持等方面简要介绍 STM32 的内核架构，让读者对内核架构有基本的了解。

2.2 内部功能结构

2.2.1 总体结构及功能模块概述

STM32 具有丰富的片内资源和强大的功能，其总体结构如图 2.2 所示。其片内含 CPU、各种类型的存储器及外设，它们均挂载在总线(包括程序总线、数据总线、DMA 总线等)上。

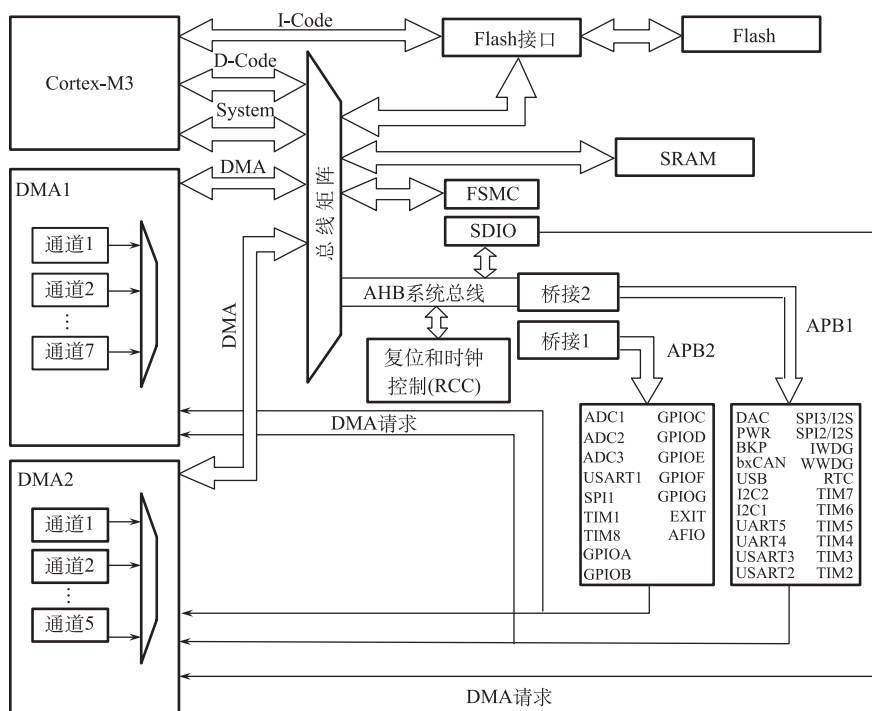


图 2.2 STM32F103ZET6 总体结构

1. CPU

STM32F103ZET6 的 CPU 具有 32 位 Cortex-M3 架构，支持多达 240 个外部中断，内嵌了嵌套向量中断控制器 (Nested Vector Interrupt Controller, NVIC)，还可以选择配上一个存储器保护单元 (MPU)。

2. 片内存储器

STM32F103ZET6 包括 512KB (256K×16 位) 的片内 Flash、64KB (32K×16 位) 的 SRAM。Flash 用于存放用户程序代码或者数据表，可以通过 JTAG 接口烧写或者擦除，CPU 对其访问需要等待时间。SRAM 每个周期只能访问一次，但是 CPU 对其访问时无须等待。

3. 片内外设

STM32F103ZET6 的片内外设主要包括 4 个通用 16 位带有输入捕获、输出比较和 PWM 输出的定时器，2 个 16 位基本定时器，2 个可以死区输出的高级定时器，2 个看门狗定时器，以及 SysTick 定时器。STM32 强大的定时器可以体现 STM32 的强大性能和在电机控制方面的特殊优势。除此之外，还有 3 个串行外设结构 (SPI/ I2C)、一个现场总线通信接口 (CAN)、

2 个内部集成电路模块(I2C)和 112 个通用输入/输出(GPIO)引脚。另外，还包括 NVIC 中断管理、3 个 ADC、1 个 DAC、1 个 SDIO。

4. 片内总线

STM32F103ZET6 内部有若干个总线接口，使得它的内核(Cortex-M3)能够同时读取地址和访问内存。这些总线包括：

- (1)指令存储区总线(两条)；
- (2)系统总线；
- (3)私有外设总线。

两条指令存储区总线负责对代码存储区的访问，分别是 I-Code 总线和 D-Code 总线。I-Code 总线用于取指，D-Code 总线用于查表等操作，它们按最佳执行速度进行优化。

系统总线用于访问内存和外设，覆盖的区域包括 SRAM、片上外设、片外 RAM、片外扩展，以及系统级存储区的部分空间。

私有外设总线负责一部分私有外设的访问，主要是访问调试组件，它们也在系统级存储区。

2.2.2 总线结构

总线结构是计算机体系结构中最基本的结构，其性能很大程度上决定了 CPU 的性能。为了提高处理速度，可通过提高 CPU 时钟频率以提高响应速度，还可加宽数据总线宽度以进行更高位数的复杂运算。此外，还有一种最佳方案就是采用并行机制，即采用多组总线，执行流水线。STM32 的主要总线结构如图 2.3 所示。

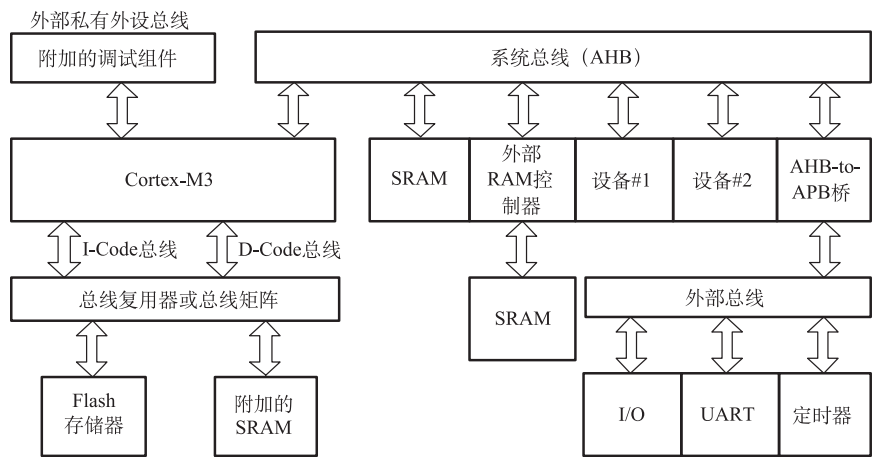


图 2.3 STM32 的主要总线结构

通常情况下，芯片厂商都会钩住(hook up)所有送往寄存器和外设的总线信号，并且在少数情况下，芯片厂商会把总线连接到总线桥上，并且允许外部总线系统连接到芯片上。基于 Cortex-M3 内核的 STM32 微控制器的总线接口是基于 AHB-LITE 和 APB 协议的，它们的规格在 AMBA(Advanced Microcontroller Bus Architecture)片上总线架构规格书(第 4 版)中给出。

1. I-Code 总线

I-Code 总线是一条基于 AHB-LITE 总线协议的 32 位总线,负责在 0x0000_0000~0x1FFF_FFFF 的取指操作。取指以字的长度执行,即使是对于 16 位指令也是如此。因此,CPU 内核可以一次取出两条 16 位 Thumb 指令。

2. D-Code 总线

D-Code 总线也是一条基于 AHB-LITE 总线协议的 32 位总线,负责在 0x0000_0000~0x1FFF_FFFF 的数据访问操作。尽管 Cortex-M3 支持非对齐访问,但是绝对不会看到该总线上任何非对齐的地址,这是因为处理器的总线接口会把非对齐的数据传送都转换成对齐的数据传送。因此,连接到 D-Code 总线上的任何设备只需支持 AHB-LITE 的对齐访问,不需要支持非对齐访问。

3. 系统总线

系统总线也是一条基于 AHB-LITE 总线协议的 32 位总线,负责在 0x2000_0000~0xDFFF_FFFF 和 0xE010_0000~0xFFFF_FFFF 的所有数据传送,取指和数据访问都要算上。和 D-Code 总线一样,所有的数据传送都是对齐的。

4. 外部私有外设总线

这是一条基于 APB 总线协议的 32 位总线。此总线负责 0xE004_0000~0xE00F_FFFF 的私有外设访问。但是,由于此 APB 存储空间的一部分已经被 TPIU、ETM 以及 ROM 表使用,只留下 0xE004_2000~E00F_F000 这个区间用于配接附加的(私有)外设。

5. 调试访问端口总线

调试访问端口总线接口是一条基于增强型 APB 规格的 32 位总线,它专用于挂接调试接口,如 SWJ-DP 和 SW-DP。

2.2.3 流水线

基于 Cortex-M3 内核的 STM32 微处理器使用一个 3 级流水线,流水线的 3 个级别分别是取指、解码和执行,如图 2.4 所示。

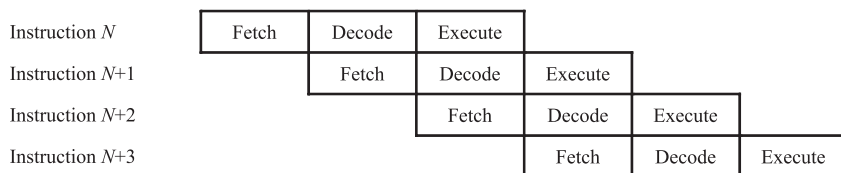


图 2.4 STM32 微处理器的流水线

当看到图 2.4 时,很多读者会认为是 4 级流水线结构,理由是总线接口在访问内存时的行为。但是这一级是在处理器的外部,故处理器自身还是只有 3 级流水线。

当运行的指令大多数是 16 位时,处理器每隔一个周期会做一次取指。这是因为 Cortex-M3 有时可以一次取出两条指令(一次能取 32 位),所以在第 1 条 16 位指令取来时,也顺带着把第 2 条 16 位指令取来了。此时总线接口就可以先休息一个周期再取指。或者如果缓冲区是满的,总线接口就空闲下来了。有些指令执行需要多个周期,在这期间流水线就会暂停。

当执行到跳转指令时，需要清洗流水线，处理器会从跳转目的地重新取指。为了改善这种情况，Cortex-M3 支持一定数量的 ARM v7-M 新指令，以避免很多微型跳转。

由于流水线的存在，以及出于对 Thumb 代码兼容的考虑，读取 PC 时，会返回当前指令地址+4 的值。无论执行的是 16 位指令还是 32 位指令，这个偏移量总是 4，从而保证了 Thumb 和 Thumb-2 之间的一致性。

在处理器内核的预取单元中也有一个指令缓冲区，它允许后续的指令在执行前先在里面排队，也能在执行未对齐的 32 位指令时，避免流水线断流。

2.3 寄存器组及 CPU 运行模式

寄存器组作为 CPU 内部重要的数据存储资源，用于暂存数据，是汇编程序员能直接使用的硬件资源之一。而 CPU 运行模式分为处理器模式和线程模式，同时通过引入特权级和用户级，能够在硬件水平上限制某些不受信任的或者还没有调试好的程序，以保护涉及要害的寄存器，提高了系统的可靠性。本节将简要介绍寄存器组及 CPU 的运行模式。

2.3.1 寄存器组

STM32F103ZET6 处理器拥有 R0~R15 以及一些特殊功能的寄存器，如图 2.5 所示。R0~R12 是最“通用目的”的，但是绝大多数的 16 位指令只能使用 R0~R7(低组寄存器)，而 32 位的 Thumb-2 指令则可以访问所有通用目的寄存器。特殊功能寄存器有预定义的功能，而且必须通过专用的指令来访问。

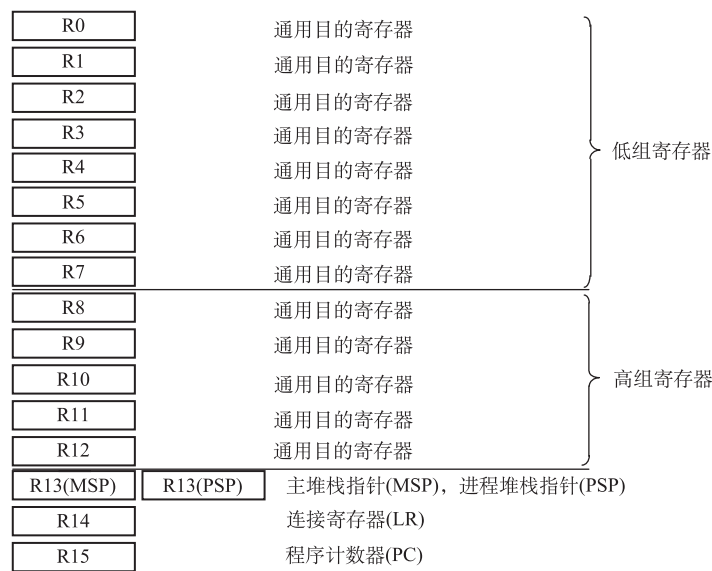


图 2.5 寄存器组

1. 通用目的寄存器 R0~R7

R0~R7 也称为低组寄存器，所有指令都能访问它们。它们的字长全部是 32 位，复位后的初值是不可预料的。