

ARM Cortex-M3 系统设计与实现 ——STM32基础篇

郭书军 王玉花 / 编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

ARM Cortex-M3 系统设计与实现

——STM32 基础篇

郭书军 王玉花 编著



电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书以 STM32 系列 32 位 Flash MCU 为例, 以“一切从简单开始”为宗旨, 以直接操作寄存器为设计方法, 介绍了 ARM Cortex-M3 系统的设计与实现。

全书共 10 章: 第 1 章简单介绍 STM32 MCU 和 SysTick 的结构, 第 2 章以一个简单的嵌入式系统设计为例详细介绍 SysTick 和 GPIO 的应用设计, 第 3、4、7 章分别介绍 USART、SPI 和 I²C 等串行接口的结构和设计实例, 第 5、6 章分别介绍 TIM 和 ADC 的结构和设计实例, 第 8、9 章分别介绍 NVIC 和 DMA 的结构和设计实例, 第 10 章以实时钟系统设计为例介绍 STM32 MCU 系统的设计与实现。

本书所有设计程序均为原创, 并经过两年 4 轮学生实验的改进, 内容简单易懂, 特别适合初学者学习参考, 也可以作为嵌入式系统设计教材, 供电子、通信和自动化等相关专业的师生使用, 书后附有实验指导, 更方便实验教学。

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

ARM Cortex-M3 系统设计与实现. STM32 基础篇 / 郭书军, 王玉花编著. —北京: 电子工业出版社, 2014.1

ISBN 978-7-121-21883-5

I. ①A… II. ①郭… ②王… III. ①微处理器—系统设计 IV. ①TP332

中国版本图书馆 CIP 数据核字 (2013) 第 271620 号

责任编辑: 赵 娜 特约编辑: 王 纲

印 刷: 北京季峰印刷有限公司

装 订: 北京季峰印刷有限公司

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: 720×1 000 1/16 印张: 10.75 字数: 241 千字

印 次: 2014 年 1 月第 1 次印刷

定 价: 29.00 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。

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

服务热线: (010) 88258888。

反侵权盗版声明

电子工业出版社依法对本作品享有专有出版权。任何未经权利人书面许可，复制、销售或通过信息网络传播本作品的行为；歪曲、篡改、剽窃本作品的行为，均违反《中华人民共和国著作权法》，其行为人应承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。

为了维护市场秩序，保护权利人的合法权益，我社将依法查处和打击侵权盗版的单位和个人。欢迎社会各界人士积极举报侵权盗版行为，本社将奖励举报有功人员，并保证举报人的信息不被泄露。

举报电话：(010) 88254396; (010) 88258888

传 真：(010) 88254397

E-mail: dbqq@phei.com.cn

通信地址：北京市万寿路 173 信箱

电子工业出版社总编办公室

邮 编：100036

前 言

ARM Cortex-M3 处理器树立了全球微控制器的标准, 40 多个 ARM 合作伙伴已获得这些处理器的使用许可, 其中包括 STMicroelectronics、NXP Semiconductors、Texas Instruments 和 Toshiba 等领先供应商。通过采用标准处理器, ARM 合作伙伴可制造出具有统一架构的设备, 同时能够专注于各自差异化的设计, 帮助开发人员满足不同的嵌入式应用需要。

STM32 系列 32 位 Flash MCU 是 STMicroelectronics 开发的基于 ARM Cortex-M3 系列的 MCU, 包括一系列 32 位产品, 具有高性能、实时功能、数字信号处理、低功耗与低电压操作特性, 同时还保持了集成度高和易于开发的特点。

本书以 STM32 系列 32 位 Flash MCU 为例, 以“一切从简单开始”为宗旨, 以直接操作寄存器为设计方法, 介绍 ARM Cortex-M3 系统的设计与实现。

全书分为 10 章, 从一个简单的嵌入式系统设计开始, 依次介绍 SysTick 和 GPIO, USART、SPI 和 I²C 等串行接口, TIM 和 ADC 等片内设备, 以及 NVIC 和 DMA 的结构和设计实例, 最后以实时钟系统设计为例介绍 STM32 MCU 系统的设计与实现。

第 1 章简单介绍 STM32 MCU 和 SysTick 的结构, 第 2 章在简单介绍 GPIO 结构的基础上, 以一个简单的嵌入式系统设计为例详细介绍 SysTick 和 GPIO 的应用设计。简单系统以自制简单实验系统为硬件平台, 软件设计采用直接操作寄存器和使用库函数两种方法, 在直接操作寄存器软件设计中详细地介绍了用 Keil 新建工程、新建添加 C 语言源文件、生成目标程序文件、调试和运行目标程序的方法和步骤, 在使用库函数软件设计中对两种方法进行了比较。最后介绍了用 GPIO 控制 LCD 的程序设计和实现方法。

第 3、4、7 章分别介绍 USART、SPI 和 I²C 等串行接口的结构和设计实例。USART 是最常用的串行接口, 设计实例实现了 MCU 与计算机的串行通信。SPI 的编程操作和 USART 相似, 设计实例实现了 SPI 的环回和用 SPI 控制 LCD。I²C 的编程操作相对复杂一些, 设计实例实现了通过 I²C 读写 2 线串行 EEPROM。

第 5、6 章分别介绍 TIM 和 ADC 的结构和设计实例, TIM 设计实例实现了

1s 定时程序设计、矩形波输出程序设计和矩形波测量程序设计等，ADC 设计实例用 ADC 规则通道实现了外部输入模拟信号的模数转换和用 ADC 注入通道实现内部温度传感器的温度测量等。

第 8、9 章分别介绍 NVIC 和 DMA 的结构和设计实例。中断和 DMA 是高效的数据传送控制方式，对前面介绍的接口和设备数据传送查询方式稍做修改即可实现中断功能，再结合 DMA 可以实现数据的批量传送。

第 10 章以实时钟系统设计为例介绍 STM32 MCU 系统的设计与实现，包括系统结构、程序设计和程序实现等。

本书所有设计程序均为原创，并在自制硬件和 Keil 4.12 环境下测试通过。

由于编者水平所限，书中难免会有不妥之处，敬请广大读者批评指正。编者联系方式 (E-mail): cortex_m3@126.com。

编者

2013 年 10 月

目 录

第 1 章	STM32 MCU 简介	1
1.1	STM32 MCU 结构	1
1.2	STM32 MCU 存储器映像	3
1.3	STM32 MCU 系统时钟树	4
1.4	Cortex-M3 简介	9
第 2 章	通用并行接口 GPIO	12
2.1	GPIO 结构及寄存器说明	12
2.2	GPIO 设计实例	14
2.2.1	直接操作寄存器软件设计	14
2.2.2	使用库函数范例程序 (V2.0.1)	30
2.2.3	用 GPIO 控制 LCD	37
第 3 章	通用同步/异步收发器接口 USART	44
3.1	UART 简介	44
3.2	USART 结构及寄存器说明	46
3.3	USART 设计实例	50
3.3.1	USART 基本功能程序设计	50
3.3.2	与计算机通信程序设计	51
3.3.3	用 printf()实现通信程序设计	56
第 4 章	串行设备接口 SPI	58
4.1	SPI 结构及寄存器说明	58
4.2	SPI 设计实例	61
4.2.1	SPI 基本功能程序设计	62
4.2.2	SPI 环回程序设计	63
4.2.3	用 SPI 控制 LCD	65
第 5 章	定时器 TIM	68
5.1	TIM 结构及寄存器说明	68
5.2	TIM 设计实例	76

5.2.1	1s 定时程序设计	76
5.2.2	矩形波输出程序设计	78
5.2.3	矩形波测量程序设计	80
第 6 章	模/数转换器 ADC	82
6.1	ADC 结构及寄存器说明	82
6.2	ADC 设计实例	89
6.2.1	用 ADC1 规则通道实现外部输入模拟信号的模数转换	89
6.2.2	用 ADC1 注入通道实现内部温度传感器的温度测量	91
第 7 章	内部集成电路总线接口 I²C	94
7.1	I ² C 结构及寄存器说明	94
7.2	I ² C 设计实例	98
7.2.1	24C32A 简介	99
7.2.2	24C32A 程序设计	99
7.2.3	24C32A 程序应用	101
第 8 章	嵌套向量中断控制器 NVIC	104
8.1	NVIC 简介	104
8.2	EXTI 中断	108
8.3	USART 中断	111
8.4	TIM 中断	112
8.5	ADC 中断	113
第 9 章	直接存储器存取 DMA	115
9.1	DMA 简介	115
9.2	USART 的 DMA 操作	118
9.3	TIM 的 DMA 操作	119
9.4	ADC 的 DMA 操作	120
第 10 章	实时钟系统设计	122
10.1	系统结构	122
10.2	程序设计	124
10.2.1	实时钟程序设计	124
10.2.2	按键和显示程序设计	126
10.2.3	系统程序设计	132
10.3	程序实现	133

附录 A 开发板简介	136
附录 B 调试器简介	144
附录 C ASCII 码表	146
附录 D C 语言运算符	147
附录 E 实验指导	148
实验 1 设计入门	148
实验 2 LCD 实验	149
实验 3 USART 实验	150
实验 4 SPI 实验	152
实验 5 TIM 实验	154
实验 6 ADC 实验	156
实验 7 I ² C 实验	158
实验 8 中断实验	159
实验 9 DMA 实验	160
实验 10 实时钟实验	161
参考文献	162

STM32 系列 32 位 Flash 微控制器基于 ARM Cortex-M 系列处理器，旨在为 MCU 用户提供新的开发自由度。STM32 MCU 包括一系列 32 位产品，具有高性能、实时功能、数字信号处理、低功耗与低电压操作特性，同时还保持了集成度高和易于开发的特点。品种齐全的 STM32 产品基于行业标准内核，提供了大量工具和软件选项，使该系列产品成为小型项目和完整平台的理想选择。

作为一个主流的微控制器系列，STM32 满足了工业、医疗和消费电子市场的各种应用需求。凭借这个产品系列，ST 在全球的 ARM Cortex-M 微控制器中处于领先地位，同时树立了嵌入式应用的里程碑。该系列最大化地集成了高性能与低功耗、低电压工作特性，在可以接受的价格范围内提供简单的架构和易用的工具。

STM32 MCU 系列包含五个产品线，它们之间引脚、外设和软件相互兼容。

- 基本型系列 STM32F101: 36MHz 最高主频，具有高达 1MB 的片上闪存。
- USB 基本型系列 STM32F102: 48MHz 最高主频，具有全速 USB 模块。
- 增强型系列 STM32F103: 72MHz 最高主频，具有高达 1MB 的片上闪存，集成电机控制、USB 和 CAN 模块。
- 互连型系列 STM32F105/107: 72MHz 最高主频，具有以太网 MAC、CAN 以及 USB 2.0 OTG 功能。

本书以增强型系列 STM32F103 为核心，介绍 STM32 MCU 的设计基础。

1.1 STM32 MCU 结构

STM32 MCU 由控制单元、从属单元和总线矩阵三大部分组成，控制单元和从属单元通过总线矩阵相连接，如图 1.1 所示。

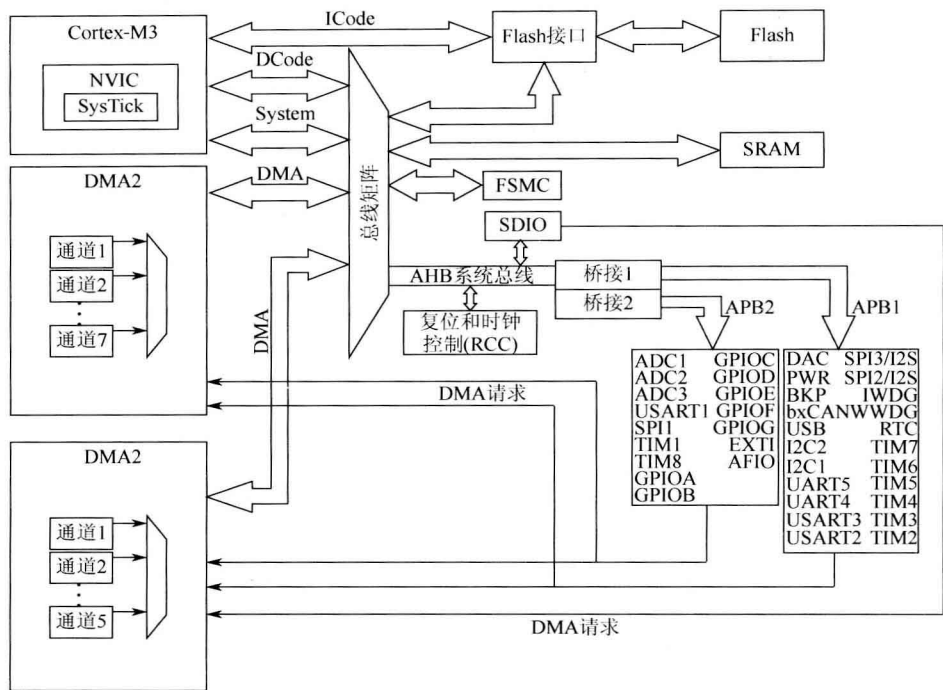


图 1.1 STM32 MCU 结构

控制单元包括 Cortex-M3 内核和两个 DMA 控制器（DMA1 和 DMA2）。其中 Cortex-M3 内核通过指令总线 ICode 从 Flash 中读指令，通过数据总线 DCode 与存储器交换数据，通过系统总线 System（设备总线）、高性能系统总线 AHB 和高级设备总线 APB 与设备交换数据。

从属单元包括存储器（Flash 和 SRAM 等）和设备（连接片外设备的接口和片内设备）。其中设备通过 AHB-APB 桥接器和总线矩阵与控制单元相连接，与 APB1 相连的是低速设备（最高频率为 36MHz），与 APB2 相连的是高速设备（最高频率为 72MHz）。

连接片外设备的接口有并行接口和串行接口两种，并行接口即通用 I/O 接口 GPIO，串行接口有通用同步/异步收发器接口 USART、串行设备接口 SPI、内部集成电路总线接口 I²C、通用串行总线接口 USB 和控制器局域网网络接口 CAN 等。

片内设备有定时器 TIM、模数转换器 ADC 和数模转换器 DAC 等，其中定时器包括高级控制定时器 TIM1/8、通用定时器 TIM2-5、基本定时器 TIM6/7、实时钟 RTC、独立看门狗 IWDG 和窗口看门狗 WWDG 等。

系统复位后，除 Flash 接口和 SRAM 时钟开启外，所有设备都被关闭，使用前必须设置时钟使能寄存器（RCC_APBENR）开启设备时钟。

1.2 STM32 MCU 存储器映象

STM32 MCU 的程序存储器、数据存储器 and 输入 / 输出端口寄存器被组织在同一个 4GB 的线性地址空间内，存储器映象如表 1.1 所示。

表 1.1 STM32 MCU 存储器映象表

地 址 范 围		设 备 名 称	备 注
0xE000 0000 – 0xE00F FFFF (1MB)		内核设备	
	0xE000E100-0xE000E4EF	NVIC（嵌套矢量中断控制）	详见表 8.2
	0xE000E010-0xE000E01F	SysTick（系统滴答定时器）	详见表 1.5
0x4000 0000 – 0x5FFF FFFF (512MB)		片上设备	
AHB	0x5000 0000 – 0x5003 FFFF	USB OTG 全速	
	0x4002 8000 – 0x4002 9FFF	以太网	
	0x4002 3000 – 0x4002 33FF	CRC	
	0x4002 2000 – 0x4002 23FF	Flash 接口	
	0x4002 1000 – 0x4002 13FF	RCC（复位和时钟控制）	详见表 1.2
	0x4002 0400 – 0x4002 07FF	DMA2	
	0x4002 0000 – 0x4002 03FF	DMA1	详见表 9.2
	0x4001 8000 – 0x4001 83FF	SDIO	
APB2	0x4001 3C00 - 0x4001 3FFF	ADC3	
	0x4001 3800 - 0x4001 3BFF	USART1	详见表 3.3
	0x4001 3400 - 0x4001 37FF	TIM8 定时器	
	0x4001 3000 - 0x4001 33FF	SPI1	详见表 4.2
	0x4001 2C00 - 0x4001 2FFF	TIM1 定时器	详见表 5.2
	0x4001 2800 - 0x4001 2BFF	ADC2	详见表 6.2
	0x4001 2400 - 0x4001 27FF	ADC1	详见表 6.2
	0x4001 2000 - 0x4001 23FF	GPIO 端口 G	
	0x4001 1C00 - 0x4001 1FFF	GPIO 端口 F	
	0x4001 1800 - 0x4001 1BFF	GPIO 端口 E	
	0x4001 1400 - 0x4001 17FF	GPIO 端口 D	
	0x4001 1000 - 0x4001 13FF	GPIO 端口 C	详见表 2.1
	0x4001 0C00 - 0x4001 0FFF	GPIO 端口 B	详见表 2.1
	0x4001 0800 - 0x4001 0BFF	GPIO 端口 A	详见表 2.1
	0x4001 0400 - 0x4001 07FF	EXTI	详见表 8.6
	0x4001 0000 - 0x4001 03FF	AFIO	

续表

地 址 范 围		设 备 名 称	备 注
APB1	0x4000 7400 - 0x4000 77FF	DAC	
	0x4000 7000 - 0x4000 73FF	PWR（电源控制）	
	0x4000 6C00 - 0x4000 6FFF	BKP（后备寄存器）	
	0x4000 6800 - 0x4000 6BFF	bxCAN2	
	0x4000 6400 - 0x4000 67FF	bxCAN1	
	0x4000 6000 - 0x4000 63FF	USB/CAN 共享的 512 字节 SRAM	
	0x4000 5C00 - 0x4000 5FFF	USB 全速设备寄存器	
	0x4000 5800 - 0x4000 5BFF	I ² C2	详见表 7.2
	0x4000 5400 - 0x4000 57FF	I ² C1	详见表 7.2
	0x4000 5000 - 0x4000 53FF	UART5	
	0x4000 4C00 - 0x4000 4FFF	UART4	
	0x4000 4800 - 0x4000 4BFF	USART3	详见表 3.3
	0x4000 4400 - 0x4000 47FF	USART2	详见表 3.3
	0x4000 4000 - 0x4000 43FF	保留	
	0x4000 3C00 - 0x4000 3FFF	SPI3/I2S3	
	0x4000 3800 - 0x4000 3BFF	SPI2/I2S2	
	0x4000 3400 - 0x4000 37FF	保留	
	0x4000 3000 - 0x4000 33FF	IWDG（独立看门狗）	
	0x4000 2C00 - 0x4000 2FFF	WWDG（窗口看门狗）	
	0x4000 2800 - 0x4000 2BFF	RTC	
	0x4000 1400 - 0x4000 17FF	TIM7 定时器	
	0x4000 1000 - 0x4000 13FF	TIM6 定时器	
	0x4000 0C00 - 0x4000 0FFF	TIM5 定时器	
	0x4000 0800 - 0x4000 0BFF	TIM4 定时器	
	0x4000 0400 - 0x4000 07FF	TIM3 定时器	
	0x4000 0000 - 0x4000 03FF	TIM2 定时器	详见表 5.2
	0x2000 0000 - 0x3FFF FFFF (512MB)		SRAM
	0x0000 0000 - 0x1FFF FFFF (512MB)		Flash
	0x1FFF F800 - 0x1FFF F80F	选择字节	
	0x1FFF F000 - 0x1FFF F7FF	系统存储器	
	0x0800 0000 - 0x0801 FFFF	主存储器	

1.3 STM32 MCU 系统时钟树

STM32 MCU 系统时钟树由系统时钟源、系统时钟 SYSCLK 和设备时钟等部分组成，如图 1.2 所示。

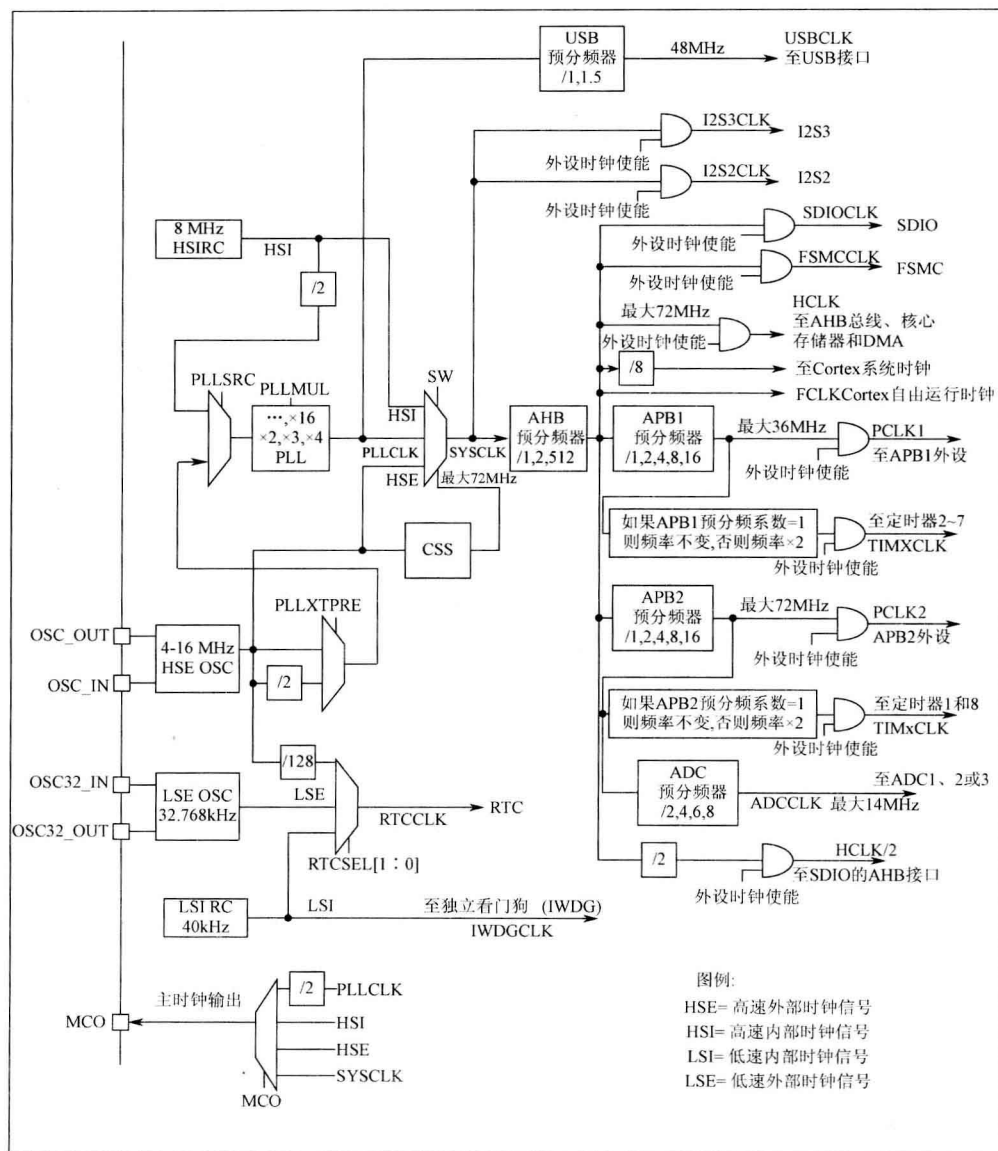


图 1.2 STM32 MCU 系统时钟树

系统时钟源有 4 个：高速外部时钟 HSE（4-16MHz）、低速外部时钟 LSE（32.768kHz）、高速内部时钟 HSI（8MHz）和低速内部时钟 LSI（40kHz）。其中外部时钟用晶体振荡器 OSC 实现，内部时钟用 RC 振荡器实现。

系统时钟 SYSCLK（最大 72MHz）可以是 HSE 或 HSI，也可以是 HSE 或 HSI 通过锁相环 2-16 倍频后的锁相环时钟 PLLCLK。系统复位后的系统时钟为 HSI，这就意味着即使没有 HSE 系统也能正常工作，只是 HSI 的精度没有 HSE 高。

SYSCLK 经 AHB 预分频器分频后得到 AHB 总线时钟 HCLK（最大 72MHz），HCLK 经 APB1/APB2 预分频器分频后得到 APB1/APB2 总线时钟 PCLK1（最大 36MHz）和 PCLK2（最大 72MHz），PCLK1 和 PCLK2 分别为相连的设备提供设备时钟。

系统时钟树中的时钟选择、预分频值和外设时钟使能等都可以通过对复位和时钟控制（RCC）寄存器编程来实现，复位和时钟控制（RCC）寄存器如表 1.2 所示（RCC 的基地址是 0x4002 1000）。

表 1.2 复位和时钟控制（RCC）寄存器

偏移地址	名称	类型	复位值	说明
0x00	RCC_CR	读/写	0x0000 0083	时钟控制寄存器（HSIRDY=1，HSION=1）
0x04	RCC_CFGR	读/写	0x0000 0000	时钟配置寄存器（SYSCLK=HSI，AHB、APB1 和 APB2 均不分频，即频率均为 8MHz，定时器时钟频率也为 8MHz，ADC 时钟为 APB2/2，即频率为 4MHz）
0x08	RCC_CIR	读/写	0x0000 0000	时钟中断寄存器（禁止所有中断）
0x0C	RCC_APB2RSTR	读/写	0x0000 0000	APB2 设备复位寄存器
0x10	RCC_APB1RSTR	读/写	0x0000 0000	APB1 设备复位寄存器
0x14	RCC_AHBENR	读/写	0x0000 0014	AHB 设备时钟使能寄存器（开启 Flash 接口和 SRAM 时钟）
0x18	RCC_APB2ENR	读/写	0x0000 0000	APB2 设备时钟使能寄存器（关闭所有 APB2 设备时钟，详见表 1.3）
0x1C	RCC_APB1ENR	读/写	0x0000 0000	APB1 设备时钟使能寄存器（关闭所有 APB1 设备时钟，详见表 1.4）
0x20	RCC_BDCR	读/写	0x0000 0000	备份域控制寄存器
0x24	RCC_CSR	读/写	0x0C00 0000	控制状态寄存器（上电复位，NRST 引脚复位）

常用的 APB2 设备时钟使能寄存器（RCC_APB2ENR）如表 1.3 所示。

表 1.3 APB2 设备时钟使能寄存器 (RCC_APB2ENR)

位	名 称	类 型	复 位 值	说 明
0	AFIOEN	读/写	0	AFIO 时钟使能: 0-关闭时钟, 1-开启时钟
2	GPIOAEN	读/写	0	GPIOA 时钟使能: 0-关闭时钟, 1-开启时钟
3	GPIOBEN	读/写	0	GPIOB 时钟使能: 0-关闭时钟, 1-开启时钟
4	GPIOCEN	读/写	0	GPIOC 时钟使能: 0-关闭时钟, 1-开启时钟
5	GPIODEN	读/写	0	GPIOD 时钟使能: 0-关闭时钟, 1-开启时钟
6	GPIOEEN	读/写	0	GPIOE 时钟使能: 0-关闭时钟, 1-开启时钟
7	GPIOFEN	读/写	0	GPIOF 时钟使能: 0-关闭时钟, 1-开启时钟
8	GPIOGEN	读/写	0	GPIOG 时钟使能: 0-关闭时钟, 1-开启时钟
9	ADC1EN	读/写	0	ADC1 时钟使能: 0-关闭时钟, 1-开启时钟
10	ADC2EN	读/写	0	ADC2 时钟使能: 0-关闭时钟, 1-开启时钟
11	TIM1EN	读/写	0	TIM1 定时器时钟使能: 0-关闭时钟, 1-开启时钟
12	SPI1EN	读/写	0	SPI1 时钟使能: 0-关闭时钟, 1-开启时钟
13	TIM8EN	读/写	0	TIM8 定时器时钟使能: 0-关闭时钟, 1-开启时钟
14	USART1EN	读/写	0	USART1 时钟使能: 0-关闭时钟, 1-开启时钟
15	ADC3EN	读/写	0	ADC3 时钟使能: 0-关闭时钟, 1-开启时钟

APB1 设备时钟使能寄存器 (RCC_APB1ENR) 如表 1.4 所示。

表 1.4 APB1 设备时钟使能寄存器 (RCC_APB1ENR)

位	名 称	类 型	复 位 值	说 明
0	TIM2EN	读/写	0	TIM2 时钟使能: 0-关闭时钟, 1-开启时钟
1	TIM3EN	读/写	0	TIM3 时钟使能: 0-关闭时钟, 1-开启时钟
2	TIM4EN	读/写	0	TIM4 时钟使能: 0-关闭时钟, 1-开启时钟
3	TIM5EN	读/写	0	TIM5 时钟使能: 0-关闭时钟, 1-开启时钟
4	TIM6EN	读/写	0	TIM6 时钟使能: 0-关闭时钟, 1-开启时钟
5	TIM7EN	读/写	0	TIM7 时钟使能: 0-关闭时钟, 1-开启时钟
11	WWDGEN	读/写	0	WWDG 时钟使能: 0-关闭时钟, 1-开启时钟
14	SPI2EN	读/写	0	SPI2 时钟使能: 0-关闭时钟, 1-开启时钟
15	SPI3EN	读/写	0	SPI3 时钟使能: 0-关闭时钟, 1-开启时钟
17	USART2EN	读/写	0	USART2 时钟使能: 0-关闭时钟, 1-开启时钟
18	USART3EN	读/写	0	USART3 时钟使能: 0-关闭时钟, 1-开启时钟
19	UART4EN	读/写	0	UART4 时钟使能: 0-关闭时钟, 1-开启时钟
20	UART5EN	读/写	0	UART5 时钟使能: 0-关闭时钟, 1-开启时钟
21	I2C1EN	读/写	0	I2C1 时钟使能: 0-关闭时钟, 1-开启时钟
22	I2C2EN	读/写	0	I2C2 时钟使能: 0-关闭时钟, 1-开启时钟

续表

位	名 称	类 型	复 位 值	说 明
23	USBEN	读/写	0	USB 时钟使能：0-关闭时钟，1-开启时钟
25	CANEN	读/写	0	CAN 时钟使能：0-关闭时钟，1-开启时钟
27	BKPEN	读/写	0	BKP 时钟使能：0-关闭时钟，1-开启时钟
28	PWREN	读/写	0	PWR 时钟使能：0-关闭时钟，1-开启时钟
29	DACEN	读/写	0	DAC 时钟使能：0-关闭时钟，1-开启时钟

在 Keil 的调试界面选择“Peripherals”（设备）菜单下的“Power, Reset and Clock Control”（电源、复位和时钟控制）子菜单可以打开“Power, Reset and Clock Control (PRCC)”对话框，如图 1.3 所示，其中包含复位和时钟控制寄存器的复位值。

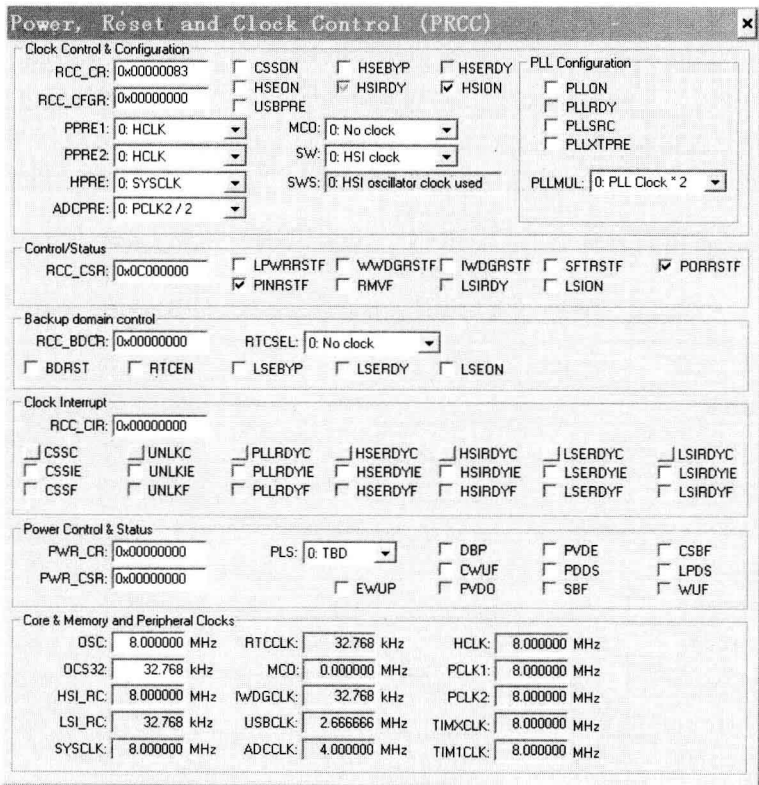


图 1.3 电源、复位和时钟控制对话框

选择“Peripherals”（设备）菜单下的“APB Bridges”（APB 桥）子菜单可以打开“APB Bridge 1”（APB 桥 1）和“APB Bridge 2”（APB 桥 2）对话框，如图 1.4 所示，其中分别包含 APB1 和 APB2 设备时钟使能寄存器的复位值。