感谢武汉力源对STM32F0参考手册的翻译

RM0091
参考手册

STM32F05xxx 先进的 ARM 核 32 位微控制器

简介

本参考手册向应用程序开发人员提供关于如何使用 STM32F05xxx 微控制器的内存和外设所涉及的全部信息。

STM32F05xxx 是一个由不同存储容量、封装和外设配备的微控制器组成的微控制器家族。

关于型号信息、尺寸和器件的电气特性等详细数据请参考对应的数据手册（datasheet）。

关于 ARM CORTEX ™ -M0 内核的相关信息，请参考 Cortex-M0 技术参考手册。

表 1. 适用产品

类型	型号
微控制器	STM32F051x4, STM32F051x6 , STM32F051x8

相关文件

- Cortex-M0 技术参考手册，可以得自：*http://infocenter.arm.com/help/topic/com.arm.doc.ddi0432c/ DDI0432C_cortex_m0_r0p0_trm.pdf*
- STM32F05xxx 数据手册（datasheets） 可得自最近的 ST 销售部门。

目录

表

图

1 文中的约定

1.1 寄存器描述中使用的缩写列表

寄存器的描述中使用了以下的缩写：

缩写	对应的英文	缩写含义
rw	read/write	软件能读写这些位 (或指定位)
r	read-only	软件只读这些位 (或指定位)
w	write-only	软件只写这些位 (或指定位)，
rc_w1	read/clear	软件可读该位，可通过对该位写 1 时清除该位。对该位写 0 时，该位值无变化。
rc_w0	read/clear	软件可读该位，可通过对该位写 0 时清除该位。对该位写 1 时，该位值无变化。
rc_r	read/clear by read	软件可读该位，读后该位自动清为 0。对该位写 0 时，该位值无变化。
rs	read/set	软件可读写该位。但其与 rw 有区别，一般设置该位为 1 时启动某种硬件动作，当完成硬件动作后该位会被硬件自动清 0。
rt_w	read-only write trigger	软件可以读此位；对该位写 0 或 1 触发一个事件但对此位数值没有影响。
t	Toggle	软件只能对该位写 1 来翻转此该位，写 0 对该位无影响。
Res	Reserved	保留位，必须保持默认值不变

1.2 有关术语

本节给出本文档涉及到的部分缩写词的一个简洁定义：

- SWD: 为 Serial Wire Debug 的首字母缩写。其是 Cortex-M0 内核集成的一个调试口，是基于 SWD 协议的 2 线调试接口。有关更详细的内容可参考 Cortex-M0 technical reference manual。
- Word: 字，32 位长的数据或指令长度。
- Half word: 半字，16 位长的数据或指令长度。
- Byte: 字节，8 位数据长度。
- IAP： in-application programming 的首字母缩写。直译为在应用编程，即用户程序可以更新自身的程序，从而达到应用升级。
- ICP： in-circuit programming 的首字母缩写。直译为在电路编程，即用户可通过应用板上的 JTAG 口或 SWD 口对 MCU 的 FLASH 进行编程。
- Option bytes: 选项字节，保存在 Flash 中的 MCU 配置字节。
- OBL: option byte loader 的首字母缩写，选项字节装载器。
- AHB: advanced high-performance bus 的首字母缩写，直译为先进高性能总线。

1.3 可用的外设

有关该类 MCU 可用的外设及外设数量，请查阅相关器件的数据手册。

2 系统及存储器概述

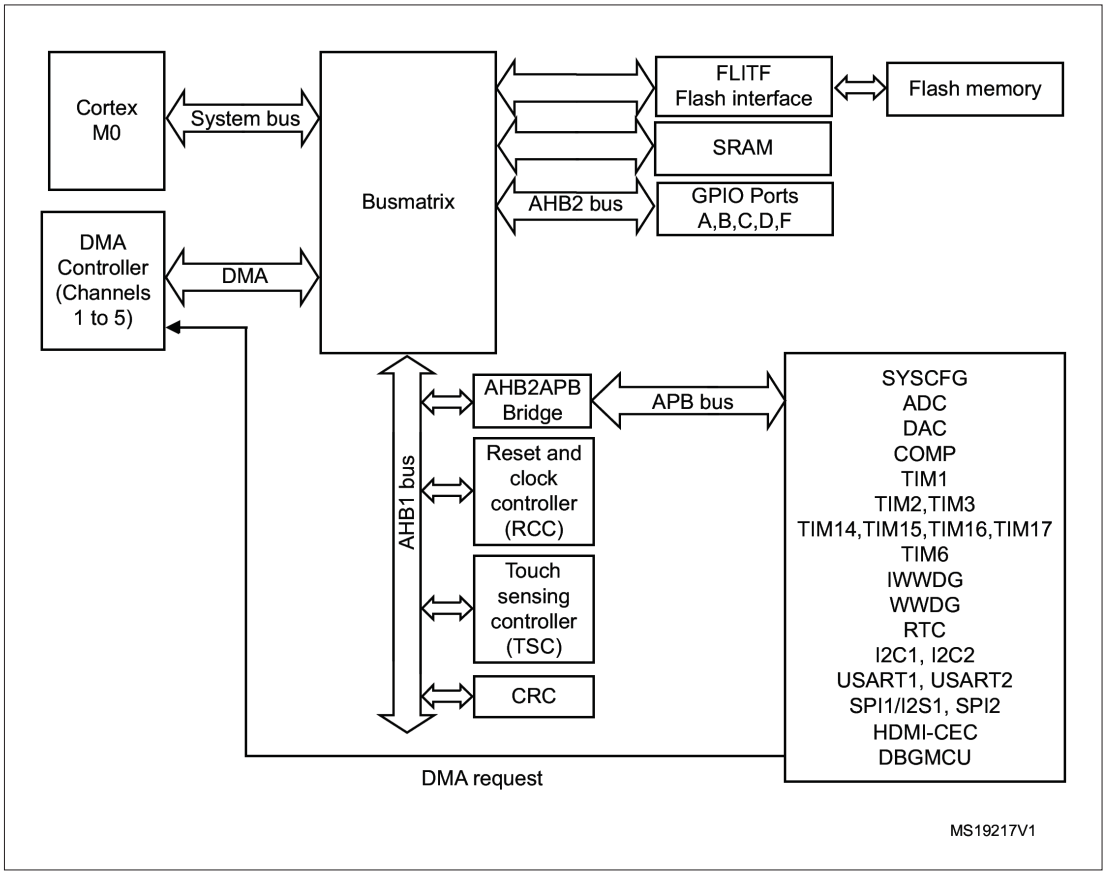
2.1 系统架构

系统主要由以下几个模块组成 :

- 二个主模块 :
  - Cortex-M0 内核及先进高性能总线 (AHB bus)
  - 通用 DMA (GP-DMA -- general-purpose DMA)
- 四个从模块 :
  - 内部 SRAM
  - 内部闪存存储器
  - AHB 到 APB 的桥 , 所有的外设都挂在 APB 总线上
  - 专门用于连接 GPIO 口的 AHB2

内部由一个多层 AHB 互联的系统总线结构图如图 1 所示：(淡黄色部分为主模块，淡绿色部分为从模块)

图 1. 系统架构图



系统部线
此总线连接 Cortex-M0 内核的系统总线到总线矩阵，总线矩阵来协调内核和 DMA 间的总线访问控制。

DMA 总线
此总线将 DMA 的 AHB 主控接口与总线矩阵相联，总线矩阵协调 CPU 和 DMA 到 SRAM、闪存和外设的访问控制。

总线矩阵 (BusMatrix)
总线矩阵管理着内核系统总线与 DMA 总线的访问仲裁，总线矩阵由 2 个主模块总线 (CPU AHB、系统总线、DMA 总线) 及四个从模块总线 (FLIFT、SRAM、AHB2GPIO 和 AHB2APB 桥) 组成。
AHB 外设通过总线矩阵与系统总线相连，充许 DMA 访问。

AHB 到 APB 桥（AHB2APB bridges － APB)
AHB 到 APB 桥在 AHB 与 APB 总线间提供同步连接。
有关连接到桥的不同外设的地址映射请参考 表 2.2.2。

在每次复位之后，所有的外设时钟都关闭 (除了 SRAM 及 FLIFT 外)。在用一个外设前，你必须打开相应的 RCC_AHBENR、RCC_APB2ENR 或 RCC_APB1ENR 寄存器中时钟使能位。

注：　当对 APB 寄存器进行 8 位或者 16 位访问时，该访问会被自动转换成 32 位的访问：桥会自动将 16 位或者 8 位的数据扩展以配合 32 位的宽度。

2.2 存储器组织

2.2.1 介绍

程序存储器，数据存储器，寄存器及 I/O 口统一编址，其线性地址空间达到 4G。

数据字节以小端格式存放在存储器中，一个字里的最低地址字节被认为是该字的最低有效字节，而最高地址字节是最高有效字节。

寻址空间分成 8 块，每块 512MB。

其他所有没有分配给片上存储器和外设的存储器空间都是保留的地址空间。详细请参考存储器映像和寄存器编址章节和外设章节。

2.2.2 存储器映像和寄存器编址

表 1 给出了 STM32F051xx 器件的存储器映像和寄存器编址。

表 1. STM32F05x 存储器映像和外设寄存器编址

总线	编址范围	大小	外设	外设寄存器映象
	0xE000 0000 - 0xE00F FFFF	1MB	Cortex M0 内部外设	
	0x4800 1800 - 0x5FFF FFFF	~384 MB	Reserved	
AHB2	0x4800 1400 - 0x4800 17FF	1KB	GPIOF	参见 8.4.11 节
	0x4800 1000 - 0x4800 13FF	1KB	Reserved	
	0x4800 0C00 - 0x4800 0FFF	1KB	GPIOD	参见 8.4.11 节
	0x4800 0800 - 0x4800 0BFF	1KB	GPIOC	参见 8.4.11 节
	0x4800 0400 - 0x4800 07FF	1KB	GPIOB	参见 8.4.11 节
	0x4800 0000 - 0x4800 03FF	1KB	GPIOA	参见 8.4.11 节
	0x4002 4400 - 0x47FF FFFF	~128 MB	Reserved	

表 1(续). STM32F05x 存储器映像和外设寄存器编址

总线	编址范围	大小	外设	外设寄存器映象
AHB1	0x4002 4000 - 0x4002 43FF	1 KB	TSC	参见 26.6.11 节
	0x4002 3400 - 0x4002 3FFF	3 KB	Reserved	
	0x4002 3000 - 0x4002 33FF	1 KB	CRC	参见 5.4.5 节
	0x4002 2400 - 0x4002 2FFF	3 KB	Reserved	
	0x4002 2000 - 0x4002 23FF	1 KB	FLASH 接口	参见 Flash 寄存器映象节
	0x4002 1400 - 0x4002 1FFF	3 KB	Reserved	
	0x4002 1000 - 0x4002 13FF	1 KB	RCC	参见 7.4.15 节
	0x4002 0400 - 0x4002 0FFF	3 KB	Reserved	
	0x4002 0000 - 0x4002 03FF	1 KB	DMA	参见 10.4.7 节
	0x4001 8000 - 0x4001 FFFF	32 KB	Reserved	
APB	0x4001 5C00 - 0x4001 7FFF	9 KB	Reserved	
	0x4001 5800 - 0x4001 5BFF	1 KB	DBGMCU	
	0x4001 4C00 - 0x4001 57FF	3 KB	Reserved	
	0x4001 4800 - 0x4001 4BFF	1 KB	TIM17	参见 18.6.16 节
	0x4001 4400 - 0x4001 47FF	1 KB	TIM16	参见 18.6.16 节
	0x4001 4000 - 0x4001 43FF	1 KB	TIM15	参见 18.5.18 节
	0x4001 3C00 - 0x4001 3FFF	1 KB	Reserved	
	0x4001 3800 - 0x4001 3FF	1 KB	SART1	参见 24.7.12 节
	0x4001 3400 - 0x4001 37FF	1 KB	Reserved	
	0x4001 3000 - 0x4001 33FF	1 KB	SPI1/I2S1	参见 25.7.10 节
	0x4001 2C00 - 0x4001 2FFF	1 KB	TIM1	参见 15.4.21 节
	0x4001 2800 - 0x4001 2BFF	1 KB	Reserved	
	0x4001 2400 - 0x4001 27FF	1 KB	ADC	参见 12.12.11 节
	0x4001 0800 - 0x4001 23FF	7 KB	Reserved	
	0x4001 0400 - 0x4001 07FF	1 KB	EXTI	参见 11.3.7 节
	0x4001 0000 - 0x4001 03FF	1 KB	SYSCFG + COMP	参见 9.1.7 节
	0x4000 8000 - 0x4000 FFFF	32 KB	Reserved	

表 1(续). STM32F05x 存储器映像和外设寄存器编址

总线	编址范围	大小	外设	外设寄存器映象
APB	0x4000 7C00 - 0x4000 7FFF	1 KB	Reserved	
	0x4000 7800 - 0x4000 7BFF	1 KB	CEC	参见 27.5.7 节
	0x4000 7400 - 0x4000 77FF	1 KB	DAC	参见 13.4.9 节
	0x4000 7000 - 0x4000 73FF	1 KB	PWR	参见 6.4.3 节
	0x4000 5C00 - 0x4000 6FFF	5 KB	Reserved	
	0x4000 5800 - 0x4000 5BFF	1 KB	I2C2	参见 23.8 节
	0x4000 5400 - 0x4000 57FF	1 KB	I2C1	参见 23.8 节
	0x4000 4800 - 0x4000 53FF	3 KB	Reserved	
	0x4000 4400 - 0x4000 47FF	1 KB	USART2	参见 24.7.12 节
	0x4000 3C00 - 0x4000 43FF	2 KB	Reserved	
	0x4000 3800 - 0x4000 3BFF	1 KB	SPI2	参见 25.7.10 节
	0x4000 3400 - 0x4000 37FF	1 KB	Reserved	
	0x4000 3000 - 0x4000 33FF	1 KB	IWWDG	参见 20.4.6 节
	0x4000 2C00 - 0x4000 2FFF	1 KB	WWDG	参见 21.6.4 节
	0x4000 2800 - 0x4000 2BFF	1 KB	RTC	参见 22.6.17 节
	0x4000 2400 - 0x4000 27FF	1 KB	Reserved	
	0x4000 2000 - 0x4000 23FF	1 KB	TIM14	参见 17.4.12 节
	0x4000 1400 - 0x4000 1FFF	3 KB	Reserved	
	0x4000 1000 - 0x4000 13FF	1 KB	TI6	参见 19.4.9 节
	0x4000 0800 - 0x4000 0FFF	2 KB	Reserved	
	0x4000 0400 - 0x4000 07FF	1 KB	TIM3	参见 16.4.19 节
	0x4000 0000 - 0x4000 03FF	1 KB	TIM2	参见 16.4.19 节
	0x2000 2000 - 3FFF FFFF	~512 MB	Reserved	
	0x2000 0000 - 0x2000 1FFF	8 KB	SRAM	
	0x1FFF FC00 - 0x1FFF FFFF	1 KB	Reserved	
	0x1FFF F800 - 0x1FFF FBFF	1 KB	Option bytes	
	0x1FFF EC00 - 0x1FFF F7FF	3 KB	System memory	
	0x0801 0000 - 0x1FFF EBFF	~384 MB	Reserved	
	0x0800 0000 - 0x0800 FFFF	64 KB	Main Flash memory	

表 1(续). STM32F05x 存储器映像和外设寄存器编址

总线	编址范围	大小	外设	外设寄存器映象
	0x0001 0000 - 0x07FF FFFF	128 MB	Reserved	
	0x0000 000 - 0x0000 FFFF	64 KB	主闪存存储器，系统存储器或是 SRAM 有赖于 BOOT 的配置	

2.3 内置的 SRAM

STM32F051xx 内置 8K 字节的静态 SRAM. 它可以以字节 (8 位)、半字 (16 位) 或字 (32 位) 进行访问。该类存储器 CPU 及 DMA 都可用最快的系统时钟且不插入任何等待进行访问。

校验检测 Parity check

数据总线宽度为 36 位，其中 32 位为数据，4 位用于每字节的奇偶校验位。051 系列 MCU 以此来增强数据存储的鲁棒性，来适应欧洲 B 类及 SIL 规范的数据安全要求。

奇偶校验位是计算和存储时写扩 SRAM 的。当读这些数据时，MCU 会自动校验数据。如果一位失败，则会产生 NMI 中断。当 SYSCFG_CFGR2 中的 SRAM_PARITY_LOCK 控制位和 SRAM_PEF 都有效时，其还可链接到 TIM1 的 BRK_IN 的输入。

2.4 闪存存储器概述

闪存 存储器有两个不同存储区域：

- 主闪存在储块，它包括应用程序和用户数据区 (若需要时)
- 信息块，其包含两个部分：
  - 选项字节 (Option bytes) 一 内含硬件及存储保护用户配置选项。
  - 系统内存 (System memory) 一其包含 boot loader 代码。参见内置闪存 存储器章节。

闪存接口基于 AHB 协议执行指令和数据存取。其预取缓冲的功能可加速 CPU 执行代码的速度。

2.5 启动配置 (Boot configuration)

在 STM32F051xx 中，可通过 BOOT0 及 BOOT1 脚的配置选择三种不同的启动模式，如下表所示。

表 2. 启动模式

启动模式选择		启动模式	说明
BOOT1	BOOT0		
x	0	主闪存存储器	主闪存存储器选为启动区域
1	1	系统存储器	系统存储器选为启动区域
0	1	内置 SRAM	内置 SRAM 选为启动区域

器件复位后，在 SYSCLK 的第 4 个上升沿锁存 BOOT0 和 BOOT1 的引脚值，用户可通过设置 BOOT1 和 BOOT0 来选择启动模式。

从待机模式唤醒时，CPU 会得新采样 BOOT0 及 BOOT1 的引脚值，因此在有待机应用的场合需要保持启动模式的设置。在启动延迟之后，CPU 从地址 0x0000 0000 获取堆栈顶的地址，并从启动存储器的 0x0000 0004 指示的地址开始执行代码。

根据选定的启动模式，主闪存存储器，系统存储器或 SRAM 按照以下的说明访问：

- 从主闪存存储器启动 : 主闪存存储器被映射到启动存储空间 (0x0000 0000), 但仍然能从原有的地址空间 (0x800 0000) 访问 . 即闪存存储器的内容可从两个地址开始访问，0x0000 0000 或 0x800 0000。
- 从系统存储器启动 : 系统存储器被映射到启动空间 (0x0000 0000)，但仍然能够在它原有的地址空间 (0x1FFF EC00) 访问。
- 从内置的 SRAM 启动 : SRAM 映射到启动空间 (0x0000 0000)，但其仍然能够在它原有的地址 空间 (0x2000 0000) 访问

内嵌的自举程序

内嵌的自举程序存放在系统存储器，由 ST 在生产时写入。该程序可以通过 USART1 对闪存进行重新编程，有关细节请参见 AN2606。

3 嵌入式闪存

3.1 闪存主要特性

- 高达 64K 字节闪存存储器
- 存储器结构：
  - 主闪存模块：16K 字（16K×32 位）
  - 信息模块：1K 字（1K×32 位）

闪存接口的特性为：

- 带预取缓冲器的读接口 (每字为 2×64 位)
- 选择字节加载器
- 闪存编程 / 擦除操作
- 访问 / 写保护
- 低功耗模式

3.2 闪存功能描述

3.2.1 闪存结构

闪存空间由 32 位宽的存储单元组成，既可以存代码又可以存数据。主闪存块按 64 页（每页 1K 字节）或 16 扇区（每扇区 4K 字节）分块，以扇区为单位设置写保护（参见存储保护相关内容）。

表 3. Flash 模块结构

Flash 区	Flash 存储器地址	大小 (字节)	名称	描述
主存储块	0x0800 0000 - 0x0800 03FF	1K	页 0	扇区 0
	0x0800 0400 - 0x0800 07FF	1K	页 1	
	0x0800 0800 - 0x0800 0BFF	1K	页 2	
	0x0800 0C00 - 0x0800 0FFF	1K	页 3	
	……	……	……	
	0x0800 7000 - 0x0800 73FF	1K	页 60	扇区 15
	0x0800 7400 - 0x0800 77FF	1K	页 61	
	0x0800 7800 - 0x0800 7BFF	1K	页 62	
	0x0800 7C00 - 0x0800 7FFF	1K	页 63	

表 3(续). Flash 模块结构

Flash 区	Flash 存储器地址	大小 (字节)	名称	描述
信息块	0x1FFF E000 - 0x1FFF F7FF	3K		系统存储器
	0x1FFF F800 - 0x1FFF F80F	4		选择字节
闪存接口寄存器	0x4002 2000 - 0x4002 2003	4		FLASH_ACR
	0x4002 2004 - 0x4002 2007	4		FALSH_KEYR
	0x4002 2008 - 0x4002 200B	4		FLASH_OPTKEYR
	0x4002 200C - 0x4002 200F	4		FLASH_SR
	0x4002 2010 - 0x4002 2013	4		FLASH_CR
	0x4002 2014 - 0x4002 2017	4		FLASH_AR
	0x4002 2018 - 0x4002 201B	4		保留
	0x4002 201C - 0x4002 201F	4		FLASH_OBR
	0x4002 2020 - 0x4002 2023	4		FLASH_WRPR

3.2.2 读保护

嵌入式 Flash 模块可以像普遍存储空间一样直接寻址访问。任何对 Flash 模块内容的读操作都须经过专门的判断过程。

取指令和取数据都是通过 AHB 总线读取访问，能够按照 Flash 访问控制寄存器（Flash_ACR）中得选项所指定的方式执行：

- 取指：预取值缓冲区使能后可提高 CPU 运行速度
- 潜伏期：等待位的个数，保证正确的读取。

取指

Cortex-M0 通过 AHB 总线取指。预取指模块的功效在于提高取指效率。

预取缓冲区

预取指缓冲区分 3 块，每块 8 个字节，其中的内容与 Flash 相同，能够完全替代一次同样大小的读取访问。预取缓冲区的工作使得更快速的 CPU 执行成为可能，因为 CPU 取一个字指令的同时下一个字的指令内容页已经在预取缓冲区中，这意味着如果代码是按照 64 位对齐的前提下，可以达到 2 倍的取指加速。

预取控制器

预取控制器会根据预取缓冲区的可用空间来把握访问 Flash 的时机。当预取缓冲区中存在至少一块可用空间时，预取控制器会发起一次读取请求。复位后，预取指缓冲区的默认状态是打开的。只有在 SYSCLK 低于 24MHz，并且 AHB 时钟没有经过任何分频的条件下（SYSCLK 必须等于 HCLK）才可以开 / 关预取指缓冲区。通常情况下，预取指缓冲区在初始化过程中就已经决定好开关状态了，而当时 MCU 运行在内部 8MHz 的 RC 振荡器下。

注： *当 AHB 时钟的预分频器不等于 1 时，预取指缓冲区必须打开访问潜伏期。*

访问潜伏期

为了保护对 Flash 的正确读取，必须在 Flash 访问控制寄存器中的 LATENCY[2:0] 中指定预取指控制器的速度比，这个数值等于每次访问 Flash 后到下次访问之间所需插入的等待周期的个数。复位后，这个值默认为零，也就是没有插入等待周期的状态。

3.2.3 Flash 写和擦除操作

STM32F051xx 的嵌入式闪存支持在线编程以及在应用编程。

ICP 是指使用 SWD 或 Boot loader 的方法在线改变 Flash 的内容，将用户代码烧录到单片机中。ICP 提供了一种简单高效的方法，免除了烧写芯片时的芯片装夹等问题。

与 ICP 方法不同的是，IAP（在应用编程）能够使用 MCU 支持的任何通信接口 (I/Os,USB,CAN,UART,I^2C,SPI，等等) 下载程序或者数据。IAP 允许用户在运行程序的过程中重写应用程序，前提是一部分应用程序必须预先用 ICP 的方法烧写进去。

烧写和擦除操作在整个产品工作电压范围内都可以完成。该操作有下列 7 个寄存器取舍完成：

- 关键字寄存器（FLASH_KEYR）
- 选项字节关键字寄存器 (FLASH_OPRKEYR)
- Flash 控制寄存器 (FLASH_CR)
- Flash 状态寄存器 (FLASH_SR)
- Flash 地址寄存器 (FLASH_AR)
- 选项字节寄存器 (FLASH_OBR)
- 写保护寄存器 (FLASH_WRPR)

只要 CPU 不去访问 Flash 空间，进行中的 Flash 写操作不会妨碍 CPU 的运行。也就是说，在对 Flash 进行写 / 擦除操作的同时，任何对 Flash 的访问都会令总线停顿，直到写 / 擦除操作完成后才会继续执行，这意味着在写 / 擦除 Flash 的同时不可以对它取指和访问数据。

在对 Flash 空间做写 / 擦除操作时，内部 RC 振荡器 (HSI) 必须处于开启状态。

对 Flash 空间的解锁

复位后，Flash 存储器默认是受保护状态的，这样可以防范意外的擦除动作。FLASH_CR 寄存器不允许被改写，除非执行一串针对 FLASH_KEYR 寄存器的解锁操作才能开启对 FLASH_CR 的访问权限。这串操作由下面 2 个写操作构成：

- 写关键字 1=0x45670123
- 写关键字 2=0xCDEF89AB

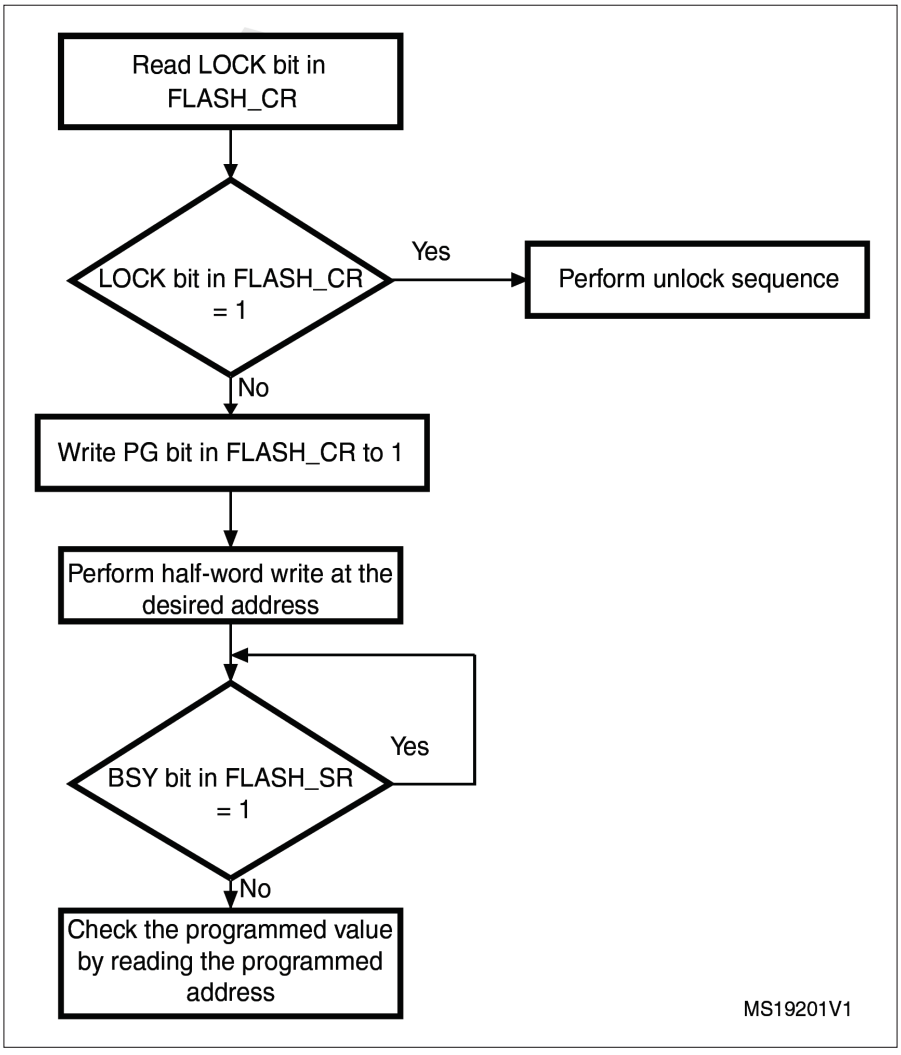
任何错误的顺序将会锁死 FLASH_CR 直至下次复位。

当发生关键字错误时，会由总线错误引发一次硬件错误中断。如果 KEY1 出错就会立即中断，或如果 KEY1 正确但 KEY2 错误时就会在 KEY2 错的那个时候引发中断。

主闪存编程

主闪存一次可以编程 16 位。当 FLASH_CR 中的 PG 位为 1 时，直接对相应的地址写一个半字（16 位），就是一次编程操作。如果试图写别的长度而不是半字，将引起硬件错误中断。

图 2. 编程流程



Flash 存储器接口会预读一下待编程字节后是否为全 1，如果不是，那么编程操作会自动取消，并且在 FLASH_SR 寄存器的 PGERR 位上提示编程错误告警。如果被编程的内容为全零，则会例外，这时会正确编程并且不告警。

如果待编程地址所对应的 FLASH_WRPR 中的写保护位有效，同样也不会有编程动作，同样也会产生编程错误告警。编程动作结束后，FLASH_SR 寄存器中得 EOP 位会给出提示。

主 Flash 存储器标准模式下的编程过程如下：

- 检查 FLASH_SR 中的 BSY 位，以确认上次编程操作已经结束
- 置 FLASH_CR 寄存器中的 PG 位
- 以半字为单位向目标地址写入数据
- 等待 FLASH_SR 寄存器中的 BSY 归零
- 读数据以校验

注： *当 FLASH_SR 中得 BSY 位为 1 的时候，这些寄存器不能写。*

Flash 存储器擦除

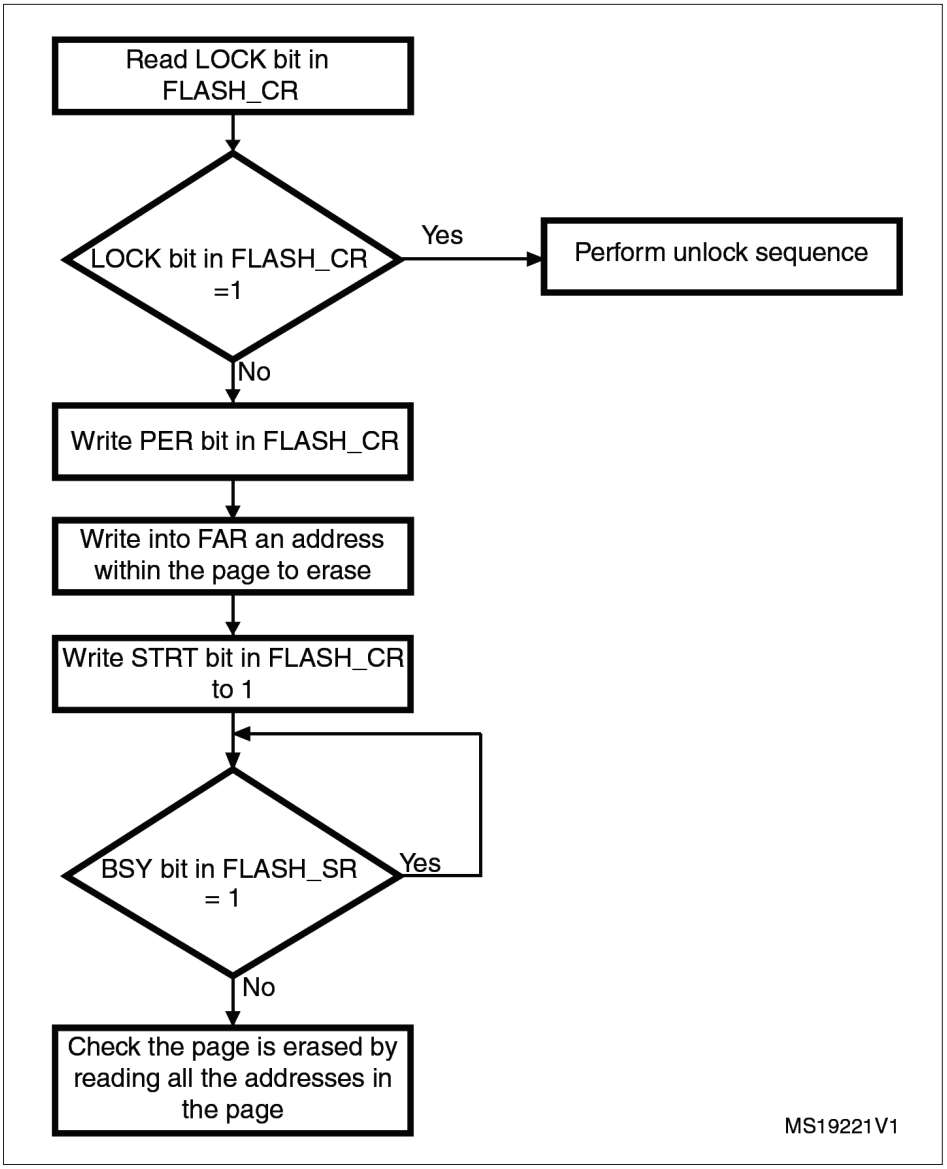
Flash 存储器可以按页为单位擦除，也可以整片擦除。

页擦除

擦除页的步骤如下：

- 检查 FLASH_SR 中的 BSY 位，以确认上次编程操作已经结束
- 置 FLASH_CR 寄存器中得 PER 位为 1
- 写 FLASH_AR 寄存器以选择待擦除的页
- 置 FLASH_CR 寄存器中的 STRT 位为 1
- 等待 FLASH_SR 中的 BSY 归零
- 读取已擦除页以校验

图 3. Flash 寄存器页擦除流程

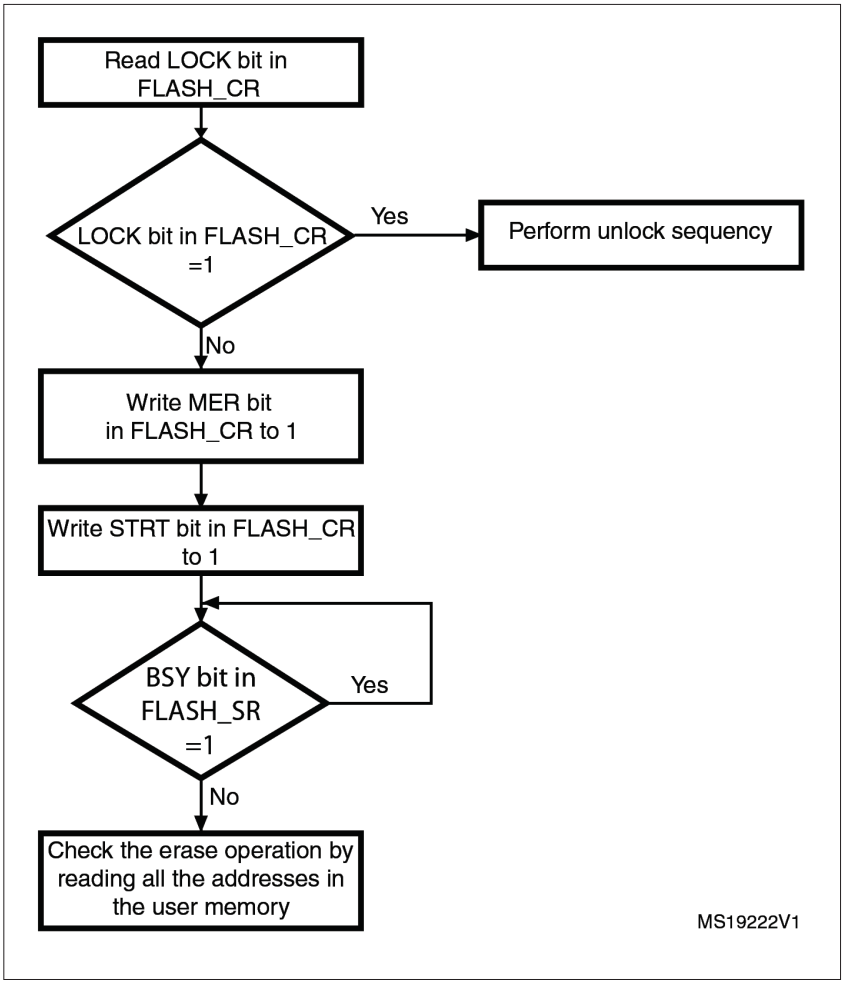


整片擦除

可以用整片擦除命令一次擦除整个 Flash 用户区，但信息块不会受这个命令影响，具体步骤如下：

- 检查 FLASH_SR 中的 BSY 位，以确认上次编程操作已经结束
- 置 FLASH_CR 寄存器中的 MER 位为 1
- 置 FLASH_CR 寄存器中的 STRT 位为 1
- 等待 BSY 位归零
- 读取全部页并校验

图 4. Flash 寄存器整片擦除流程



选项字节编程

选项字节的编程与常规用户地址不同，总共就是 4 个字节（2 个写保护，1 个读保护，1 个硬件配置）。解除 Flash 访问限制后，还需要针对 FLASH_OPTKEYR 寄存器完成关键字写入操作。完成该操作后，FLASH_CR 寄存器中的 OPTWRE 位会被置 1，然后就可以先置位 FLASH_CR 中的 OPTPG 位，再按半字单位写目标地址。同样是会自动检查选项字节是否为 1，否则相关操作会被取消并且在 FLASH_SR 中的 WRPRTERR 位提示错误。编程操作结束后，会由 FLASH_SR 寄存器的 EOP 位给出提示。

在编程操作开始前，LSB 值会自动补到 MSB，这会保证选项字节的值总是对的。

步骤如下：

- 检查 FLASH_SR 寄存器中的 BSY 位，以确保上次编程结束
- 解锁 FLASH_CR 寄存器中的 OPTWRE 位
- 置 FLASH_CR 寄存器中的 OPTPG 位为 1
- 写数据（半字）到目标地址
- 等待 BSY 位归零
- 读取并校验

当读保护选项字节由保护状态被改成非保护状态时，会自动引发一次整片擦除。如果用户只想改写其他的字节，则不会引发整片擦除，这个机制用于保护 Flash 的内容。

擦除过程

选项字节的擦除过程如下：

- 检查 FLASH_SR 寄存器中的 BSY 位，以确保上次编程结束
- 解锁 FLASH_CR 寄存器中的 OPTWRE 位
- 置 FLASH_CR 寄存器中的 OPTER 位为 1
- 置 FLASH_CR 寄存器中的 STRT 位为 1
- 等待 BSY 位归零
- 读取并校验

3.3 存储保护

可以防范用户区 Flash 区的代码被不可信的代码读出，也可以防范在程序跑飞的时候对 Flash 的意外擦除，写保护的最小单位是一个扇区（4 页）。

3.3.1 读保护

将选项字节中的 RDP 字节置位，然后重新复位，读保护就被激活了。

注： *如果读保护被置位的时候仍然通过 SWD 连着调试器，可以用 POR（上电复位）代替系统复位。*

存在 level0（无保护）到 level2（最大限度）保护三个保护级别，最大限度保护后将不再允许改变 Flash 内容。参见表 5：保护状态和保护级别及运行模式对照表。

Flash 存储器的保护级别和 RDP 选项字节及其补数内容的对应关系，见表 4。

表 4. Flash 读保护状态

RDP 字节值	RDP 补码值	读保护级别
0xxAA	0x55	Level 0
任意值 除 0xAA 和 0xCC 外	任意值（不要求互补） 除 0x55 和 0x33 外	Level 1（默认）
0xCC	0x33	Level 2

系统存储区不受读保护字节的影响，但该区域不允许编程和擦除操作。

Level 0：无保护

针对主 Flash 区域的读写和擦除操作都被允许，选项字节也全都可以操作。

Level 1：读保护

这是 RDP 选项字节被擦除之后的默认保护级别。对应的 RDP 值为除 0xAA 和 0xCC 以外的任意值或者其补数不正确。

- 用户模式：在用户模式下执行的代码允许对主 Flash 和选项字节做全部操作。
- Debug，boot RAM 和 boot loader 模式：在调试模式下或运行在 boot RAM 和 boot loader 状态下，主 Flash 区和备份寄存器均不允许访问。在该状态下，任何简单的读访问都会引起总线错误并引发硬件错误中断。主 Flash 区同时也禁止写和擦除操作，以防范恶意程序修改代码，任何尝试改写的操作都会引起 FLASH_SR 中的 PGERR 标志置位。当 RPD 字节的内容有 0xAA 改为 Level 0 的级别会先执行整片擦除操作，并且备份寄存器的值也会被复位。

Level 2：无 debug

在这个级别上，Level 1 的保护功能肯定都是有的，除此之外，CortexM0 的调试能力也被禁止了，所以 SWD 调试口从 RAM 启动，以及从系统区启动等功能也都没有了。

在用启执行模式下，允许对主 Flash 区做全部操作，相反对于选项字节却只能读取和写入而并不能做擦除。

此外，RDP 字节不能再改写，因此，level 2 这个保护级别永远不能清除掉，是一个不可恢复性的操作。当试图改写 RDP 字节时，FLASH_SR 寄存器中的保护错误标志 WRPRTERR 会被置位并引发一个中断。

注： *1、在复位条件下，调试功能不能用*

2、意法半导体也不能对打开了 Level 2 保护功能的器件做分析处理。

表 5. 保护状态和保护级别及运行模式对照表

区域	保护级别	用户代码执行			调试 / 从 RAM 启动 / 从系统区域启动		
		读	写	擦除	读	写	擦除
主 Flash 区域	1	Yes	Yes	Yes	No	No	No(3)
	2	Yes	Yes	Yes	N/A(1)	N/A(1)	N/A(1)
系统区域 (2)	1	Yes	No	No	Yes	No	No
	2	Yes	No	No	NA(1)	N/A(1)	N/A(1)
选项字节	1	Yes	Yes(3)	Yes	Yes	Yes(3)	Yes
	2	Yes	Yes(4)	No	N/A(1)	N/A(1)	N/A(1)
备份寄存器	1	Yes	Yes	N/A	No	No	Yes
	2	Yes	Yes	N/A	N/A(1)	N/A(1)	N/A(1)

1. 当 Level 2 保护级别使能，调试口、从 RAM 启动和从系统启动都被禁止
2. 系统区是唯一在任何情况下可读的区域
3. 当 RDP 被改成不保护时，主 Flash 会被擦除
4. 所有的选项字节中，除 RDP 字节外都能被再次编程

改变读保护级别

改变 RDP 的值到其他值（除 0xCC 以外）就可以轻松的从 Level 0 级迁移到 Level 1 级别。将 RDP 写成 0xCC，就可以直接进入 Level 2 级别。相反的，绕开整片擦除动作而进入 Level 0 级别是不可能的。当 RDP 变成 0xAA 的时候，整片擦除动作已经启动了。

注：当整片擦除命令执行时，备份寄存器（RTC_BKPxR in the RTC）也被复位。位了确保保护级别生效，选项字节必须通过 Flash 控制寄存器中的 FORCE_OPTLOAD 位强制重新加载。

3.3.2 写保护

写保护以一个扇区为单位（4 页）来控制，配置选项字节中的 WRP[1:0] 位，然后通过 FLASH_CR 寄存器中的 FORCE_OPTLOAD 位强制重新加载选项字节就可以使能这个保护。如果试图写入或擦除一个受保护的扇区，会引起 FLASH_SR 中的 WRPRTERR 标志位被置位。

写保护的解除

解除写保护有 2 个应用例子可以提供：

- 例 1：在解除写保护后禁止读保护
  — 使用 FLASH_CR 中的 OPTER 位擦除整个选项字节区域
  — 向 RDP 写入 0xAA 从而解除所有保护，这自然会引起整片擦除
  — 设置 FLASH_CR 中的 FORCE_POTLOAD 位，引起选项字节 (和新 WRP[1:0] 位) 重新加载写保护解除
- 例 2：在解除写保护后，读保护仍然有效，这种办法对使用用户 boot loader 进行在应用编程时很有用。
  — 使用 FLASH_CR 中的 OPTER 位擦除整个选项字节区域
  — 设置 FLASH_CR 中的 FORCE_POTLOAD 位，引起选项字节 (和新 WRP[1:0] 位) 重新加载写保护解除

3.3.3 选项字节的写保护

选项字节默认是写保护的并且任何时候都可读。为了对选项字节进行写 / 擦除操作，必须对 POTKEYR 顺序写入关键字（类似解锁）。正确的关键字会引起 FLASH_CR 中的 POTWRE 置位，表明解锁成功。同样，通过对该位清零，能够再度禁止对选项字节的写操作。

3.4 Flash 中断

表 6. Flash 中断请求

中断事件	事件标志	使能控制位
操作结束	EOP	EOPIE
写保护错误	WRPRTERR	ERRIE
编程错误	PGERR	ERRIE

3.5 Flash 寄存器描述

Flash 寄存器必须按 32 位的方式访问（半字节访问是不允许的）

3.5.1 Flash 访问控制寄存器(FLASH_ACR)

地址偏移：0x00

复位值：0x0000 0030

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRFTBS	PRFTBE	Res.	LATENCY[2:0]		
										r	rw		rw	rw	rw

位 31:6 保留，必须保留复位值

位 5 PRFTBS：预取缓冲区状态

该位提供预取缓冲区的状态

0：预取缓冲区被禁止

1：预取缓冲区被使能

位 4 PRFTBS：预取缓冲区使能

0：禁止预取指

1：使能预取指

位 3 保留，必须保留复位值

位 1:0 LATENCY[2:0]：潜伏期

本位预设 SYSCLK 周期和 Flash 访问时间的比率关系

000：零等待位，适用于 0 < SYSCLK ≤ 24 MHz

001：1 个等待位适用于 24 MHz < SYSCLK ≤ 48 MHz

3.5.2 Flash 关键字寄存器

地址偏移：0x04

复位值：xxxx xxxx

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FKEYR[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FKEYR[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w		

注： *该位全部只写，如果读之会返回 0*

位 31:0 FKEYR：关键字

该位用于输入关键字以解锁 Flash。

3.5.3 Flash 选项关键字寄存器 (FLASH_OPTKEYR)

地址偏移 : 0x08

复位值 : xxxx xxxx

所有寄存器位全部只写，如果读之会返回 0。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OPTKEYR[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPTKEYR[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits 31:0 OPTKEYR: 选项字节关键字

该位用于输入关键字以解锁 OPTWRE.

3.5.4 Flash 状态寄存器 (FLASH_SR)

地址偏移 : 0x0C

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EOP	WRPRT ERR	Res.	PG ERR	Res.	BSY
										rw	rw		rw		r

Bits 31:6 保留，必须保留复位值

Bit 5 EOP: 操作结束

当 Flash 操作（写 / 擦除）完成时由硬件置位

写 1 后可清零

注：用 EOP 可以判断是否每次操作都顺利完成

Bit 4 WRPRTERR: 写保护错误标志

当出现对写保护区域的写操作时被硬件置位

写 1 后可清零

Bit 3 保留，必须保留复位值

Bit 2 PGERR: 写入错误标志

当被编程区域的状态不为 '0xFFFF' 的情况下，执行写入操作时被硬件置位

写 1 后可清零

注：在写操作之前 FLASH_CR 寄存器中的 STRT 位应该先被清零

Bit 1 保留，必须保留复位值

Bit 0 BSY: 忙标志

该位标明 Flash 操作处于过程中。当开始 Flash 操作的时候被硬件置位，当操作结束时或发生错误时被硬件清零。

3.5.5 Flash 控制寄存器 (FLASH_CR)

地址偏移 : 0x10

复位值 : 0x0000 0080

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	FORCE_ OPTLOAD	EOPIE	Res.	ERRIE	OPTWR E	Res.	LOCK	STRT	OPTER	OPT PG	Res.	MER	PER	PG
		rw	rw		rw	rw		rw	rw	rw	rw		rw	rw	rw

Bits 31:14 保留，必须保留复位值

Bit 13 FORCE_OPTLOAD: 选项字节强制更新

当被写为 1 时，该位强制选项字节的重加载，该操作会引起系统复位。

0: 无效

1: 有效

Bit 12 EOPIE: 操作结束中断使能

该位使能操作结束中断，使得 FLASH_SR 中的 EOP 位变成 1 的时候产生中断请求

0: 中断禁止

1: 中断使能

Bit 11 保留，必须保留复位值

Bit 10 ERRIE: 错误中断使能

该位使能操作错误中断，使得 FLASH_SR 中的 PGERR/WRPRTERR 位变成 1 的时候产生中断请求。

0: 中断禁止

1: 中断使能

Bit 9 OPTWRE: 选项字节写使能

该位为 1 时，选项字节即允许改写。对 FLASH_OPTKEYR 寄存器写入正确的关键字序列就可以将它置 1。

该位可软件清零。

Bit 8 保留，必须保留复位值

Bit 7 LOCK: 锁定 Flash 标志

只能写 1。当该位为 1 时，表明 Flash 为锁定状态。该位可以解锁关键时序来清零，当发现解锁不成功时，该位就一直为 1 了，除非下次复位重新操作。

Bit 6 STRT: 启动

该位会触发一个擦除操作，仅由软件置 1，仅会在 BSY 被清零时清零。

Bit 5 OPTER: 选项字节擦除

选项字节擦除时选择

Bit 4 OPTPG: 选项字节写入

选项字节写入时选择

Bit 3 保留，必须保留复位值

Bit 2 MER: 整片擦除
整片擦除时选择
Bit 1 PER: 页擦除
页擦除时选择
Bit 0 PG: 写入
Flash 写入时选择

3.5.6 Flash 地址寄存器 (FLASH_AR)

地址偏移 : 0x14
复位值 : 0x0000 0000
本寄存器由硬件根据当前和上次操作的地址更新。对于页擦除操作，该寄存器该由软件来更新以便瞄准要擦除的页。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FAR[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

15	14	13	12	11	10	9	8	6	5	4	3	2	1	0	
FAR[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits 31:0 FAR: Flash 地址
当 PG 位被选中时，选择待写入的地址，或当 PER 位被选中时，选择待擦除的页。
注：当 FLASH_SR 中的 BSY 位是 1 时，对这个寄存器的写访问将被阻塞。

3.5.7 选项字节寄存器 (FLASH_OBR)

地址偏移 : 0x1C

复位值 : 0x03FF FFF2

本寄存器的复位值取决于选项字节的写入值，OPTERR 位的复位值取决于复位时选项字节加载环节中比较选项字节及其补码的结果。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data1								Data0								Res.	Res.	VDDA_MONITOR	BOOT1	Res.	nRST_STDBY	nRST_STOP	WDG_SW	Res.	Res.	Res.	Res.	Res.	LEVEL2_PROT	LEVEL1_PROT	OPTERR
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r			r	r		r	r	r						r	r	r

Bits 31:24 Data1

Bits 23:16 Data0

Bits 15:8 User option bytes :

Bit 15 : reserved

Bit 14 : reserved

Bit 13 : VDDA_MONITOR

Bit 12 : BOOT1

Bit 11 : reserved

Bit 10 : nRST_STDBY

Bit 9 : nRST_STOP

Bit 8 : WDG_SW

Bits 7:3 保留，必须保留复位值

Bit 2 LEVEL2_PROT: Level 2 保护状态

当置位时，表明当前处于 Level 2 保护状态

Bit 1 LEVEL1_PROT: Level 1 保护状态

当置位时，表明当前处于 Level 1 保护状态 (reset value=1).

Bit 0 OPTERR: 选项字节错误

当置位时，表明加载选项字节发现互补关系不成立。

3.5.8 写保护寄存器 (FLASH_WRPR)

地址偏移 : 0x20

复位值 : 0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WRP[15:0]															

Bits 31:0 WRP: 写保护

本寄存器包含从选项字节中加载到的写保护状态

3.6 Flash 寄存器镜像

表 7. Flash 接口 - 寄存器镜像和复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	**FLASH_ACR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRFTBS	PRFTBE	HLFCYA	LATENCY [2:0]		
	Reset value																											1	1	0	0	0	0
0x004	**FLASH_KEYR**	FKEY1:0]																															
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
0x008	**FLASH_OPTKEYR**	OPTKEYR[31:0]																															
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x00C	**FLASH_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EOP	WRPRTERR	Res.	PGERR	ERLYBSY	BSY
	Reset value																											0	0		0	0	0
0x010	**FLASH_CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FORCE_OPTLOAD	EOPIE	Res.	ERRIE	OPTWRE	Res.	LOCK	STRT	OPTER	OPTPG	Res.	MER	PER	PG
	Reset value																			0	0		0	0		1	0	0	0		0	0	0
0x014	**FLASH_AR**	FAR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x01C	**FLASH_OBR**	Data1								Data0								Res.	Res.	VDDA_MONITOR	BOOT1	Res.	nRST_STDBY	nRST_STOP	WDG_SW	Res.	Res.	Res.	Res.	Res.	LEVEL2_PROT	LEVEL1_PROT	OPTERR
	Reset value							1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
0x020	**FLASH_WRPR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP[15:0]															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

绝对地址参见 2.2.2 小结中的寄存器边界

4 选项字节描述

共有 4 个选项字节，这些选项由终端用户的应用需求配置。配置示例：看门狗是由硬件还是软件启动的模式配置。

一个 32 位字分离出如下表的选项字节：

表 8. 选项字节格式

31-24	23-16	15 -8	7-0
选项字节 1 补码	选项字节 1	选项字节 0 补码	选项字节 0

选项字节在信息块结构如表 9 所示。

选项字节可以读表 9 中所列的地址读取或从寄存器 FLASH_OBR 中读取。

注： *系统复位后，装入新编程的选项字节（用户，读 / 写保护）。*

表 9. 选项字节结构

地址	[31:24]	[23:16]	[15:8]	[7:0]
0x1FFF F800	nUSER	USER	nRDP	RDP
0x1FFF F804	nData1	Data1	nData0	Data0
0x1FFF F808	nWRP1	WRP1	nWRP0	WRP0

表 10. 选项字节描述

闪存存储器地址	选项字节
0x1FFF F800	位 [31:24] nUSER 位 [23:16] USER: 用户选项字节 (存于 FLASH_OBR[15:8]) 该字节用于配置如下特性： – 选择看门狗事件 : 硬件还是软件。 – 当进入停止模式时复位事件。 event when entering Stop mode. – 当进入待机模式时复位事件。 位 23 : 保留 位 22 : 保留 . 值必须为 1. 位 21 : VDDA_MONITOR 0 : V_{DDA} 电源电压监控失能 1 : V_{DDA} 电源电压监控使能 位 20 : BOOT1 连同 BOOT0 引脚 , 选择从主闪存存储器，SRAM 或系统内存启动。参见 节 2.5: 启动配置 位 19 : 保留 位 18: nRST_STDBY 0: 当进入待机模式产生复位 . 1: 不产生复位 位 17: nRST_STOP 0: 当进入停机模式产生复位 1: 不产生复位 位 16: WDG_SW 0: 硬件看门狗 1: 软件看门狗 位 [15:8]: nRDP 位 [7:0]: RDP: 读保护选项字节 该字节的值决定闪存存储器保护级别 0xAA : 水平 0 0xXX (除了 0xAA 和 0xCC) : 水平 1 0xCC : 水平 2 *注 : 保护级别水平 1 和 2 分别保存在 FLASH_OBR 闪存选项寄存器 (LEVEL1_PROT 和 LEVEL2_PROT) 中 . 详见* 节 3.2.2: 读保护。
0x1FFF F804	Datax: 两字节的用户数据 . 这些地址可由选项字节编程过程进行编程。 位 [31:24]: nData1 位 [23:16]: Data1 (存于 FLASH_OBR[25:18]) 位 [15:8]: nData0 位 [7:0]: Data0 (存于 FLASH_OBR[17:10])

表 10. 选项字节描述 (续)

闪存存储器地址	选项字节
0x1FFF F808	WRPx: 闪存存储器写保护选项字节 位 [31:24]: nWRP1 位 [23:16]: WRP1 (存于 FLASH_WRPR[15:8]) 位 [15:8]: nWRP0 位 [7:0]: WRP0 (存于 FLASH_WRPR[7:0]) 1: 写保护使能 0: 写保护失能 详见 节 3.3.2: 写保护 .

每次系统复位后，选项字节装载器 (OBL) 读信息块数据并且存储这些数据到相应的选项字节寄存器 (FLASH_OBR) 和闪存保护寄存 器 (FLASH_WRPR) 中。每个选项字节都有其值的补码数据同样存放在信息块中，其目的用于校验选项字节的正确性。当选项字节装载后，CPU 会检查选项字节的正确性，若选项字节与其补码比较不一致时，会产生选项字节校验错 (OPTERR) 信息 。当比较错产生后， CPU 会强制相应的选项字节值变为 0xFF。当选项字节与其补码都为 0XFF（擦除状态）时，CPU 就不会比较其与补码的差异。

5 CRC 计算单元

5.1 简介

CRC 计算单元可以用来按照既定的多项式算法，依据输入数据快速算出循环冗余校验的结果码。

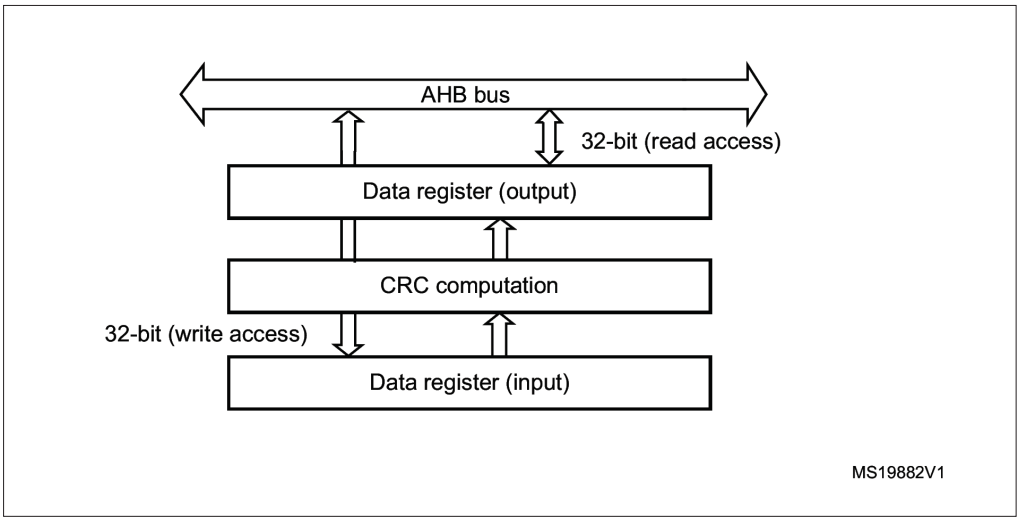
在很多应用中，通常使用循环冗余校验的技术来检查数据传输或存储的完整性。 在功能安全标准范围内，这提供了校验 Flash 存储可靠性的技术手段。 CRC 计算单元可随时计算软件签名，使得可以在通讯和存储的时候就地完成签名比较。

5.2 CRC 主要功能

- 采用的 CRC-32（与以太网标准相同）多项式 0x4C11DB7
  . X32 + X26 + X23 + X22 + X16 + X12 + X11 + X10 +X8 + X7 + X5 + X4 + X2+ X +1
- 操作 8，16，32 位数据
- CRC 初值可预置
- 单 输入 / 输出 32 位数据寄存器
- 配有输入缓冲区可以在总线停顿的时候不影响实时计算。
- 每次 CRC 计算在 4 个 AHB 时钟周期（HCLK）内完成
- 配有通用目的的 8 位寄存器（可被用来当作临时存储）
- 用来提供 I/O 数据的可逆性选项。

5.3 CRC 功能描述

图 5. CRC 计算单元框图



CRC 计算单元中包含一个 32 位可读 / 写的数据寄存器（CRC_DR）。 它用来输入新的计算数据（写操作），也用来输出上次计算的结果（读操作）。

每次对该寄存器的写操作都会使得 CRC 计算单元将所写入的数据和上次计算得到的数据进行综合计算，并得到一个新的计算结果。 CRC 计算单元根据写入数据的格式不同来决定是整字计算还是一个字节一个字节计算。

CRC_DR 寄存器可以按字访问，也可以按找右对齐的半字或者右对齐的字节的方式访问。
根据数据宽度不同的计算持续时间：

- 32 位需要 4 个 AHB 时钟周期
- 16 位需要 2 个 AHB 时钟周期
- 8 位需要 1 个 AHB 时钟周期

一个输入缓冲区，使得可以立即启动计算而不需要等待 4 个 HCLK 周期。
数据宽度可以动态调整到针对待计算数据块所合适的最小的计算写入次数。 例如，一个针对 5 个字节的计算可以改成先计算一个字，然后再计算一个字节的形式。

输入数据高低位可以颠倒，以适应各种不同的大小端体系。 数据颠倒操作可以按找 8 位、16 位和 32 位进行，这个功能由 CRC_CR 寄存器中的 REV_IN[1:0] 位来选择设置。

例如 : 输入数据 0x1A2B3C4D 用于 CRC 计算 :
按字节颠倒就是 :0x58D43CB2
按半字颠倒就是 0xD458B23C
按整字颠倒就是 0xB23CD458
输入数据也可以通过设置 CRC_CR 寄存器中的 REV_OUT 位来自动颠倒。 该操作是按位进行的： 例如，输出数据 0x11223344 转换过来就是 0x22CC4488

可以通过 CRC_CR 寄存器中的 RESET 控制位将 CRC 计算器的初值初始化为一个特定的值，这个值可以由程序指定，默认为 0xFFFFFFFF.

这个初始值可以通过 CRC_INIT 寄存器来指定。 在初始化时自动更新到 CRC_DR 寄存器中。

而 CRC_IDR 寄存器则用来保存 CRC 计算的临时结果。 这个寄存器不会受 CRC_CR 寄存器中的 RESET 位影响。

5.4 CRC 寄存器

5.4.1 数据寄存器 (CRC_DR)

地址偏移 : 0x00

复位值 : 0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DR[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DR[15:0]															
rw															

Bits 31:0 DR[31:0]: 数据寄存器位

该寄存器用于接收待计算的新数据，直接将其写入即可。 读取该寄存器得到的则是上次 CRC 计算的结果。

如果数据不足 32 位，则读写到的正确数据只针对有意义的位。

5.4.2 独立数据寄存器 (CRC_IDR)

地址偏移 : 0x04

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDR[7:0]							
								rw							

Bits 31:8 保留 , 必须保持 0

Bits 7:0 IDR[7:0]: 通用目的的 8 位数据寄存器位

这些位可以当作一个字节的临时存储。

该寄存器不受 CRC_CR 寄存器中的 RESET 位所引发的复位动作的影响。

5.4.3 控制寄存器 (CRC_CR)

地址偏移 : 0x8

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REV_OUT	REV_IN[1:0]		Res.	Res.	Res.	Res.	RESET
								rw	rw	rw					rs

Bits 31:5 保留 , 必须保持 0

Bit 7 REV_OUT: 翻转输出数据

该位控制输出数据的翻转

0: 不翻转

1: 翻转

Bits 6:5 REV_IN[1:0]: 翻转输入数据

该位控制输入数据的翻转

00: 不翻转

01: 按字节为单位翻转

10: 按半字为单位翻转

11: 按字为单位翻转 ,

Bits 4:3 保留 , 必须保持 0

Bits 2:1 保留 , 必须保持 0

Bit 0 RESET: 复位控制

该位用来复位整个 CRC 计算单元，并将 CRC_INIT 寄存器中的值更新到当前计算状态中，由软件置位，由硬件清零。

5.4.4 CRC 初值寄存器 (CRC_INIT)

地址偏移 : 0x10

复位值 : 0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CRC_INIT[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRC_INI[15:0]															
rw															

Bits 31:0 CRC_INIT: CRC 初值预置

该寄存器用来设置 CRC 的初值。

5.4.5 CRC 寄存器镜像

表 11. CRC 寄存器镜像和复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	CRC_DR	Data register																															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x04	CRC_IDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Re	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Independent data register							
	Reset value																									0	0	0	0	0	0	0	0
0x08	CRC_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REV_OUT	REV_IN		Res.	Res.	Res.	Res.	RESET
	Reset value																									0	0	0					0
0x10	CRC_INIT	CRC initial value																															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		

寄存器边界地址请参见 *35 页的章节 2.2.2*

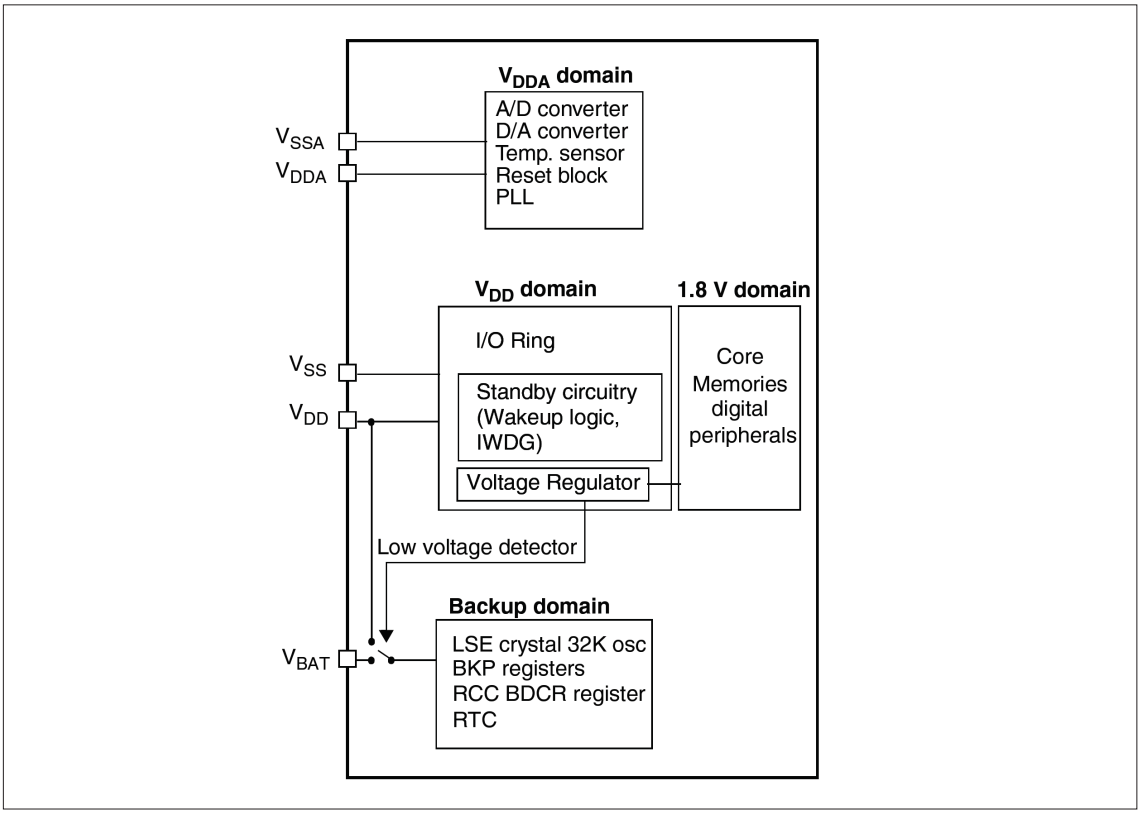
6 电源控制 (PWR)

6.1 电源

STM32051 系列器件的工作电压为 1.65 ～ 3.6 V 。内置电压调节器提供给内部 1.8V 的数字电路电源。

当主电源 VDD 掉电后，器件的实时时钟和备份域寄存器可由 VBAT 供电。

图 6. 电源供电框图



6.1.1 独立的 A/D 和 D/A 转换器供电和参考电压

为了提高转换精度，ADC 和 DAC 用独立的电源供电，单独供电的目的是过滤和屏蔽来自电路板的噪声。

- ADC 和 DAC 的电源引脚是 VDDA。
- 独立的电源地 VSSA.。

VDDA 供电 / 参考电压可以大等于 VDD. 电压。

当器件用单电源供电方式，则 VDDA 可连接到 VDD, 通过外部滤波电路以确保无噪声的 VDDA/ 参考电压。

当 VDDA 与 VDD 不同时，VDDA 必须高于或等于 VDD 的供电电压。在通电或断电的过程中，为了也确保这一条件的成立可在 VDD 和 VDDA 之间串接一个肖特基二极管。

6.1.2 电池备份域

为了保持备份域寄存器的内容及保证 RTC 时钟的工作，当 VDD 掉电后，VBAT 脚可由可选的电池供电或由其它形式的电源供电。

由 VBAT 引脚供电为 RTC 模块、LSE 振荡器和 PC13、PC14 及 PC15 的 IO 口提供了能源。当器件主电源掉电后，其为 RTC 的正常运行提供了能源保障。在复位模块中的电源掉电复位电路为 VBAT 为电源的正确切换提供了保证。

注意：
在 VDD 上升阶段 (tRSTTEMPO) 或者检测到掉电复位 (PDR) 之后，VBAT 和 VDD 之间的电源开关仍会保持连接到 VBAT。在 VDD 上升阶段，如果 VDD 在小于 tRSTTEMPO 的时间内达到稳定状态 (关于 tRSTTEMPO 可参考数据手册中的相关部分)，且 VDD > VBAT + 0.6V 时，电流可能通过 VDD 和 VBAT 之间的内部二极管注入到 VBAT。如果与 VBAT 连接的电源或者电池不能承受这样的注入电流，强烈建议在外部 VBAT 和电源之间连接一个低压降二极管。

如果在应用中没有外部电池，建议 VBAT 在外部通过一个 100nF 的陶瓷电容与 VDD 相连，更多细节参阅 AN2586

当备份区域由 VDD(内部模拟开关连到 VDD) 供电时，下述功能可用：

- PC14 和 PC15 可以作为 GPIO 口或 LSE 引脚
- PC13 可作为 GPIO 口 , TAMPER 引脚 , RTC 校准时钟 , RTC 闹钟或秒脉冲输出 (详见 *22.6.6 节的 RTC 备份域寄存器 (RTC_BKPxR)*

注： 因为模拟开关只能通过少量的电流 (3mA)，在输出模式下使用 PC13 至 PC15 的 I/O 口功能是有限制的：速度必须限制在 2MHz 以下，最大负载为 30pF，而且这些 I/O 口绝对不能当作电流源 (如驱动 LED)

当备份域由 VBAT 供电时 (VDD 消失后模拟开关连到 VBAT)，可以使用下述功能：

- PC14 和 PC15 只能用于 LSE 引脚
- PC13 可以作为 TAMPER 引脚、RTC 闹钟或秒脉冲输出 (*参见 22.6.13 节：RTC 时钟校准寄存器 (RTC_CALR)*)

6.1.3 电压调节器

器件复位后电压调节器总是打开着的，其根据应用模式有三种不同的工作模式。

- 运行模式：调节器以全功耗模式为域 (内核，内存和数字外设) 提供 1.8V 电源。
- 停止模式：调节器以低功耗模式为保持寄存器及 SRAM 数据部分域提供 1.8V 的电源。
- 待机模式：调节器断电，除了待机电路及备份域电路外，寄存器和 SRAM 的内容全部丢失。

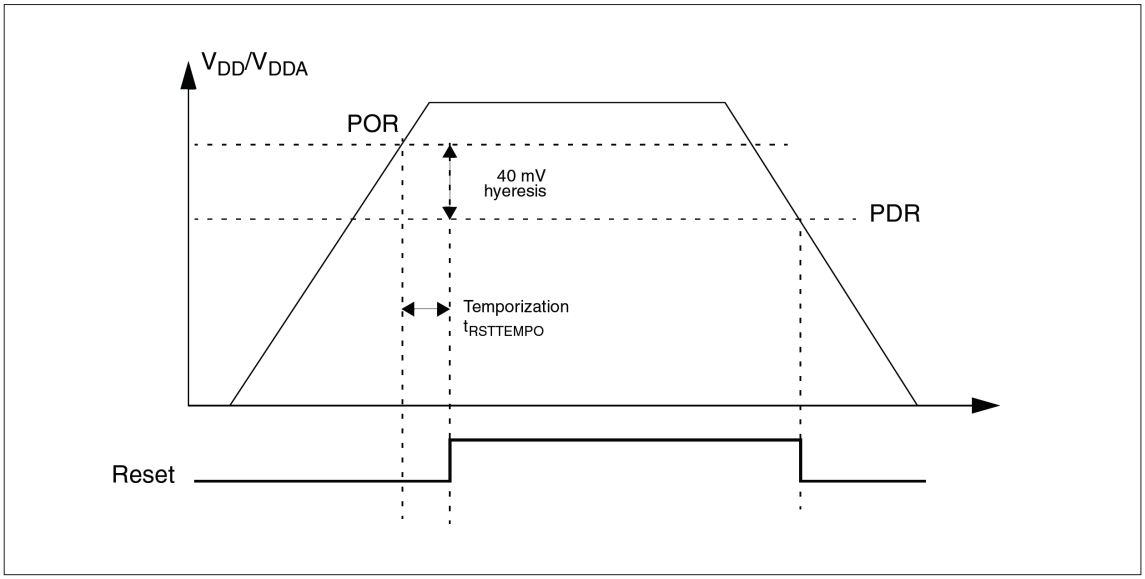
6.2 电源管理器

6.2.1 上电复位 (POR)/ 掉电复位 (PDR)

STM32F051xx 内置恰当处理电源升至或降至 1.65V 所对应的 POR 及 PDR 电路。

当 VDD/VDDA 的电压低于一个规定门限时，器件维持复位状态，而无需外部复位电路。有关上电与掉电复位的电压门限请参考数据手册的电气特性章节内容。

图 7. 上电复位 / 掉电复位波形图



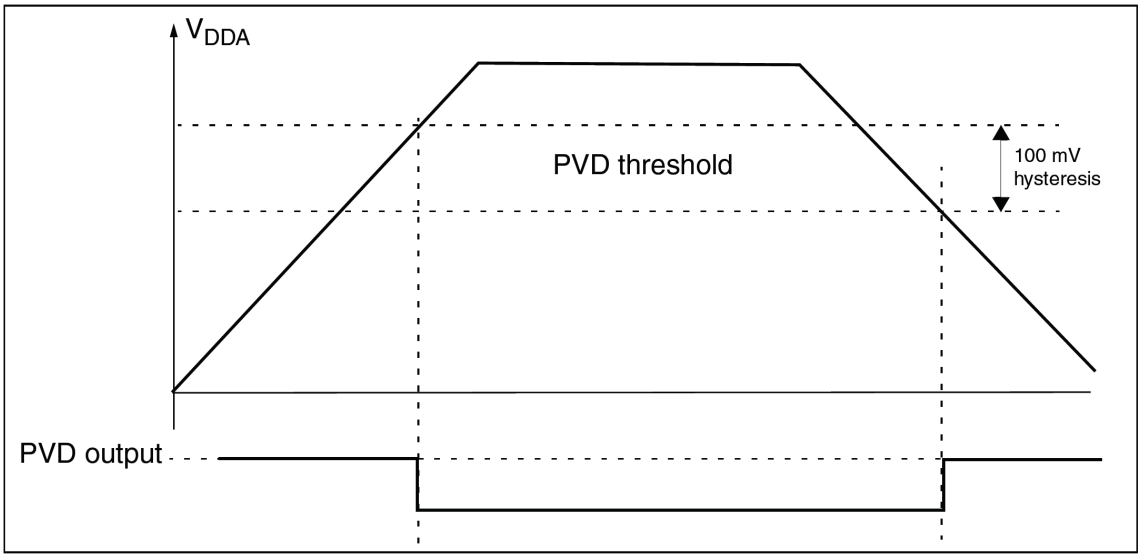
6.2.2 可编程电压检测器 (PVD)

用户可用 PVD 设置的电压阀值去监视比较 VDDA 电源供电情况，PVD 的设置参考电源控制寄存器 (PWR_CR) 中 PLS[2:0] 位。

通过设置 PVDE 位来使能 PVD。

电源控制 / 状态寄存器 (PWR_CSR) 中的 PVDO 标志用来表明 VDD 是高于还是低于 PVD 的电压阀值。该事件在内部连接到外部中断的第 16 线，如果该中断在外部中断寄存器中是使能的，该事件就会产生中断。当 VDD 下降到 PVD 阀值以下和 (或) 当 VDD 上升到 PVD 阀值之上时，根据外部中断第 16 线的上升 / 下降边沿触发设置，就会产生 PVD 中断。例如，这一特性可用于用于执行紧急关闭任务。

图 8. PVD 阀值



6.3 低功耗模式

缺省情况下，当在系统复位或电源复位之后微控制器工作于运行模式。当 CPU 不需要继续运行时，可以利用多种低功耗模式来节省功耗，例如等待某个外部事件时。用户需要根据最低电源消耗、最短启动时间和可用的唤醒源等条件，选定一个最佳的低功耗模式。

STM32F051xx 系列有三种低功耗模式：The device features three low-power modes:

- 睡眠模式 (Sleep mode) (CPU 时钟关闭 , 所有外设包括内核外设如 NVIC, SysTick, 等仍在运行)
- 停止模式 (Stop mode) (所有时钟都停止)
- 待机模式 (Standby mode) (1.8V 供电域断电)

此外，在运行模式下，可以用以下方法降低功耗：

- 降低系统时钟频率
- 关闭未用外设的时钟

表 12. 低功好模式一览

模式	进入	唤醒	对 1.8V 区域时钟的影响	对 VDD 区域时钟的影响	电压调节器
睡 眠 (Sleep) (Sleep now or Sleep-on - exit)	WFI	任一中断	CPU 时钟关闭，对其它时钟及模拟时钟无影响	无	开
	WFE	唤醒事件			
停机 (Stop)	PDDS 和 LPDS 位 + SLEEPDEEP 位 + WFI 或 WFE	任一外部中断 (在 EXTI 寄存器中设置) 指定通信口接收事件 (CEC, USART, I2C)	所有 1.8V 区域时钟关闭	HSI 和 HSE 振荡器关闭	长启或牌低功耗模式 (依据 (电源控制寄存器 *(PWR_CR) 的设置*)
待机 (Standby)	PDDS 位 + SLEEPDEEP 位 + WFI 或 WFE	唤醒引脚上升沿 , RTC 闹钟 , NRST 脚的外部复位 ,IWDG 复位			关

6.3.1 降低系统时钟频率

在运行模式中的系统时钟 (SYSCLK, HCLK, PCLK) 可通过可编程预分频寄存器降速。在进入睡眠模式前这些分频器也可用来降低外设时钟。.

有关细节请参考 *7.4.2 节：时钟配置寄存器 (RCC_CFGR).*

6.3.2 外设时钟控制

在运行模式下，可随时通过停止对外设和存储器提供时钟 (HCLK 和 PCLK) 来减少功耗。

为了在睡眠模式下减少更多功耗，可在执行 WFI 或 WFE 的指令前关闭外设时钟。

外设时钟由 AHB 外设时钟使能寄存器 (RCC_AHBENR) 来控制。

6.3.3 睡眠模式 (Sleep mode)

进入睡眠模式

执行 WFI (等待中断) 或 WFE (等待事件) 指令可让 STM32F051xx 进入睡眠模式。依据 Cortex-M0 系统控制寄存器中 SLEEPONEXIT 位，有两种有效选项可用于选择睡眠模式进入机制 :

- Sleep-now: 当 SLEEPONEXIT 位清 0 时，只要执行 WFI 或 WFE 指令 MCU 就立即进入睡眠模式。
- Sleep-on-exit: 当 SLEEPONEXIT 位置 1 时，MCU 从最低优先级的中断服务程序中退出时，再进入睡眠模式。

在睡眠模式下，所有的 I/O 口线都保持着与运行模式一样的状态。有关进入睡眠模式的细节请参考*表 13* 和 *表 14* 内容。

退出睡眠模式

如果是执行 WFI 指令时入了睡眠模式，任意一个由 NVIC 识别的外设中断都可以唤醒 MCU 退出睡眠模式。

如果是执行 WFE 指令时进入了睡眠模式，当任一事件发生时，MCU 退出睡眠模式。唤醒事件可通过以下方式产生：

- 在外设控制寄存器中使能一个中断，但不在相应的 NVIC 中使能，并且在 Cortex-M0 系统控制寄存器中使能 SEVONPEND 位。当 MCU 从 WFE 中唤醒后，外设的中断挂起位和外设的 NVIC 中断通道挂起位 (在 NVIC 中断清除挂起寄存器中) 必须被清除。
- 配置一个外部或内部 EXTI 线做为事件模式。当 CPU 从 WFE 唤醒后，因为与事件线对应的挂起位未被设置，不必清除外设的中断挂起位或外设的 NVIC 中断通道挂起位。

因没有在中断的进入或退出上浪费时间，所以该模式唤醒所需时间最短。有关退出睡眠模式的更多细节请参考*表 13* 和 *表 14*。

表 13. Sleep-now 模式

Sleep-now 模式	说明
进入	在以下条件下执行 WFI (等待中断) 或 WFE (等待事件) 指令： – SLEEPDEEP = 0 和 – SLEEPONEXIT = 0 参考 Cortex-M0 系统控制寄存器。
退出	当执行 WFI 指令进入睡眠模式：中断：参考*表 26: 中断向量表* 当执行 WFE 指令进入睡眠模式：唤醒事件：参考 *11.2.3 节：唤醒事件管理*
唤醒延时	无

表 14. Sleep-on-exit 模式

Sleep-on-exit 模式	说明
进入	在以下条件下执行了 WFI 指令： – SLEEPDEEP = 0 和 – SLEEPONEXIT = 1 参考 Cortex-M0 系统控制寄存器。
退出	中断：参考*表 26：中断向量表*
唤醒延时	无

6.3.4 停机模式 (Stop mode)

停止模式是在 Cortex-M0 的深睡眠模式基础上结合了外设的时钟控制机制，在停止模式下电压调节器可运行在正常或低功耗模式。此时在 1.8V 供电区域的的所有时钟都被停止，PLL、HSI 和 HSE RC 振荡器的功能被禁止，SRAM 和寄存器内容被保留下来。

在停止模式下，所有的 I/O 引脚都保持它们在运行模式时的状态。

进入停止模式

关于如何进入停止模式，评见*表 15*。

在停止模式下，通过设置电源控制寄存器 (PWR_CR) 的 LPDS 位使内部调节器进入低功耗模式，能够降低更多的功耗。

如果正在进行闪存编程，须等到对存储器的访问完成，系统才进入停止模式。

如果正在进行对 APB 的访问，须等到对 APB 访问完成，系统才进入停止模式。

在停止模式中 , 可通过对独立的控制位进行编程来选择如下功能：

- 独立看门沟 (IWDG): 独立看门狗可由写看门狗键寄存器或硬件选项配置来启动。一旦启动了独立看门狗，看门狗会一直开启除非进行系统复位。详见 *20.3 节：IWWDG 功能描述－独立看门狗 (IWWDG)*。

- 实时时钟 (RTC)：通过备份域控制寄存器 (RCC_BDCR) 的 RTCEN 位来设置。
- 内部低速 RC 振荡器 (LSI RC)：通过控制 / 状态寄存器 (RCC_CSR) 的 LSION 位来设置。
- 外部 32.768 kHz 振荡器 (LSE OSC)：通过备份域控制寄存器 (RCC_BDCR) 的 LSEON 位设置。

在停止模式下，如果在进入该模式前 ADC 和 DAC 没有被关闭，那么这些外设仍然消耗电流。可通过设置寄存器 ADC_CR2 的 ADON 位和寄存器 DAC_CR 的 ENx 位为 0 来关闭这 2 个外设。

退出停止模式

有关如何退出停止模式，详见*表 15*。

当由一个中断或事件唤醒 MCU 退出停止模式时，HSI RC 振荡器被选为系统时钟。

当电压调节器处于低功耗模式下，系统从停止模式退出时，将会有一段额外的启动延时。如果在停止模式期间保持内部调节器开启，则退出启动时间会缩短，但相应的功耗会增加。

表 15. 停止模式

停止模式	说明
进入	当在以下条件下执行了 WFI (等待中断) 或 WFE (等待事件) 指令： – SLEEPDEEP = 1 (Cortex-M0 系统控制寄存器) – PDDS = 0 (电源控制寄存器 (PWR_CR)) – 通过设置 LPDS(PWR_CR) 位选择电压调节器模式 注：为了进入停止模式，所有的外部中断请求挂起位 (在挂起寄存器 (*EXTI_PR) 中*) 和 RTC 闹钟标志必须清除。否则，系统会忽略 WFI 或 WFE 批令程序继续运行。
退出	如果是执行了 WFI 批令进入了停止模式： – 任一外部中断线配置为中断模式 (在 NVIC 中必须使能相应的 EXTI 中断向量). – 一些特定的通信外设 (CEC, USART, I2C) 中断，当编程为唤醒模式 (该外设必须配置为唤醒模式且在 NVIC 中相应的中断向量必须使能)。 参考*表 26：中断向量表* 如果是执行了 WFE 指令进入了停止模式：任一外部中断线配置为事件模式 . 参考 *11.2.3 节：唤醒事件管理*
唤醒延时	HSI RC 唤醒时间 + 调压器从低功耗模式唤醒的时间

6.3.5 待机模式

待机模式可实现系统的最低功耗。该模式是在 Cortex-M0 深睡眠模式时关闭电压调节器。整个 1.8V 供电区域被断电。PLL、HSI 和 HSE 振荡器也被断电。SRAM 和寄存器内容丢失。只有备份的寄存器和待机电路维持供电 (见 *图 6*).

进入待机模式

有关如何进入待机模式详见*表 16* 。

可以通过设置独立的控制位，选择以下待机模式的功能：

- 独立看门沟 (IWDG): 独立看门狗可由写看门狗键寄存器或硬件选项配置来启动。一旦启动了独立看门狗，看门狗会一直开启除非进行系统复位。详见*20.3 节：IWWDG 功能描述－独立看门狗 (IWWDG)*。
- 实时时钟 (RTC)：通过备份域控制寄存器 (RCC_BDCR) 的 RTCEN 位来设置。
- 内部低速 RC 振荡器 (LSI RC)：通过控制 / 状态寄存器 (RCC_CSR) 的 LSION 位来设置。
- 外部 32.768 kHz 振荡器 (LSE OSC)：通过备份域控制寄存器 (RCC_BDCR) 的 LSEON 位设置。

退出待机模式

当有 NRST 引脚上的外部复位信号，IWDG 复位，WKUP 引脚上的上升沿或 RTC 闹钟事件 (见*图 193：RTC 模块图*) 发生时微控制器退出待机模式。当退出待机模式时除了电源控制 / 状态寄存器 (PWR_CSR) 外所有寄存器复位。

从待机模式唤醒后的，程序执行等同于复位后的执行 (采样启动模式引脚、读取复位向量等)。电源控制 / 状态寄存器 (PWR_CSR) 的 SBF 状态标志指示内核由待机状态退出。

有关如何退出待机模式 参考*表 16*。

表 16. 待机模式

待机模式	说明
进入	在以下条件下执行 WFI(等待中断) 或 WFE(等待事件) 指令： – SLEEPDEEP = 1 (Cortex-M0 系统控制寄存器) – PDDS = 1 (电源控制寄存器 PWR_CR) – WUF = 0 (电源控制 / 状态寄存器 PWR_CSR)
退出	WKUP 引脚上升沿，RTC 闹钟事件，NRST 引脚上的外部中断信号，独立看门狗复位。
唤醒延时	与复位相同

待机模式中的 I/O 状态

在待机模式下，除了以下引脚外，所有的 I/O 口线处于高阻态。

- 复位引脚 (始终有效)
- 当配置为防侵入或校准输出时的 TAMPER 引脚
- 使能的唤醒 (WKUP) 引脚

调试模式

默认情况下，如果在进行调试微处理器时，使微处理器进入停止或待机模式，将失去调试连接。这是因为 Cortex-M0 的内核失去了时钟。

然而，通过设置 DBGMCU_CR 寄存器中的某些配置位，可以在使用低功耗模式下调试软件。

6.3.6 低功耗模式下的自动唤醒

RTC 可以在不需要依赖外部中断的情况下唤醒低功耗模式下的微控制器 (自动唤醒模式)。RTC 提供一个可编程的时间基数，用于周期性从停止或待机模式下唤醒。通过对备份区域控制寄存器 (RCC_BDCR) 的 RTCSEL[1:0] 位的编程，三个 RTC 时钟源中的二个时钟源可以选作实现此功能。

- 低功耗 32.768kHz 外部晶振 (LSE)
  该时钟源提供了一个低功耗且精确的时间基准。(在典型情形下消耗小于 1 μ A)
- 低功耗内部 RC 振荡器 (LSI RC)
  使用该时钟源，节省了一个 32.768kHz 晶振的成本。但是 RC 振荡器将少许增加电源消耗。

为了用 RTC 闹钟事件将系统从停止模式下唤醒，必须进行如下操作：

- 配置外部中断线 17 为上升沿触发。
- 配置 RTC 使其可产生 RTC 闹钟事件。

如果要从待机模式中唤醒，不必配置外部中断线 17。

6.4 电源控制寄存器

可以用半字 (16 位) 或字 (32 位) 的方式操作这些外设寄存器。

6.4.1 电源控制寄存器 (PWR_CR)

偏移地址：0x00

复位值：0x0000 0000 (从待机模式唤醒时清除)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	Res	DBP	PLS[2:0]			PVDE	CSBF	CWUF	PDDS	LPDS
							rw	rw	rw	rw	rw	rc_w1	rc_w1	rw	rw

位 31:9 保留 , 必须保持为复位值。

位 8 DBP：取消备份域的写保护

在复位后，RTC 和后备寄存器处于被保护状态以防意外写入。设置这位允许写入这些寄存器。

0：禁止写入 RTC 和后备寄存器

1：允许写入 RTC 和后备寄存器 注：

注：如果 RTC 的时钟是 HSE/128，该位必须保持为’1’。

位 7:5 PLS[2:0]: PVD 电平选择 .

软件设置这些位用于选择可编程电压检测器的电压阀值。

000: 2.2V

001: 2.3V

010: 2.4V

011: 2.5V

100: 2.6V

101: 2.7V

110: 2.8V

111: 2.9V

注 :

1. 详细说明见数据手册的电气特性部分。
2. 当 PVD_LOCK 使能 (B 类安全保护) PLS[2:0] 这些位就再也不能改写了。

位 4 PVDE: 电源电压检测使能 .

该位用软件设置或清除。

0: PVD 失能

1: PVD 使能

位 3 CSBF: 清除待机标志

该位始终读出为 0。

0: 无效

1: 清除 SBF 待机标志 (写).

位 2 CWUF: 清除唤醒标志
该位始终读出为 0。
0: 无效
1: 清除 WUF 唤醒标志，置 1 后 2 个系统时钟周期清除 WUF(写)

位 1 PDDS: 掉电深睡眠
该位用软件设置或清除，与 LPDS 位协同工作。
0: 当 CPU 进入深睡眠时进入停机模式，调压器的状态由 LPDS 位控制。
1: 当 CPU 进入深睡眠时进入待机模式。

位 0 LPDS: 深睡眠下的低功耗
该位用软件设置或清除，与 PDDS 位协同工作。
0: 在停机模式下电压调节器开启
1: 在停机模式下电压调节器处于低功耗模式

6.4.2 电源控制 / 状态寄存器 (PWR_CSR)

偏移地址： 0x04

复位值：0x0000 0000 (从待机模式下唤醒时该值不被清除)

与标准的 APB 读相比，读此寄存器需要额外的 APB 周期

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	EWUP2	EWUP1	Res	Res	Res	Res	Res	PVDO	SBF	WUF
						rw	rw						r	r	r

位 31:10 保留 , 必须保持为复位值。

位 9 EWUP2: 使能 WKUP2 引脚

该位用软件设置或清除

0: WKUP2 引脚作为通用 IO 口。WKUP2 引脚上的事件不能将 CPU 从待机模式唤醒。

1: WKUP2 引脚用于将 CPU 从待机模式唤醒，WKUP2 引脚被强置为输入下拉的配置 (WKUP2 引脚上的上升沿将系统从待机模式唤醒)

注：在系统复位时清除这一位。

位 8 EWUP1: 使能 WKUP1 引脚

该位用软件设置或清除

0: WKUP1 引脚作为通用 IO 口。WKUP1 引脚上的事件不能将 CPU 从待机模式唤醒。

1: WKUP1 引脚用于将 CPU 从待机模式唤醒，WKUP1 引脚被强置为输入下拉的配置 (WKUP1 引脚上的上升沿将系统从待机模式唤醒)

注：在系统复位时清除这一位。

位 7:3 保留 , 必须保持为复位值。

位 2 PVDO: PVD 输出

该位只能由硬件设置和清除。它仅 PVD 使能 (PVDE=1) 的情况下有效。

0: VDD/VDDA 高于由 PLS[2:0] 选定的 PVD 阀值。

1: VDD/VDDA 低于由 PLS[2:0] 选定的 PVD 阀值。

注：

1. CPU 处于待机模式时 PVD 停止工作。因此，待机模式后或复位后，直到设置 PVDE 位之前，该位为 0。
2. 一旦 PVD 使能且配置相应的 PCR 寄存器位，PVDO 可用于生产由外部中断控制器控制的中断。
3. 一旦 PVD_LOCK 使能 (B 类安全保护), PVDO 就不能被失效。

位 1 SBF: 待机标志

该位由硬件设置，并只能由 POR/PDR(上电 / 掉电复位) 或设置电源控制寄存器 (PWR_CR) 的 CSBF 位清除。

0：系统不在待机模式

1：系统进入待机模式

位 0 WUF: 唤醒标志

该位由硬件设置，并只能由 POR/PDR(上电 / 掉电复位) 或设置电源控制寄存器 (PWR_CR) 的 CWUF 位清除。

0: 没有唤醒事件发生

1: 从 WKUP 或 RTC 闹钟产生一个唤醒事件。

注：当 WKUP 引脚已经是高电平时，在 (通过设置 EWUP 位) 使能 WKUP 引脚时，会检测到一个额外唤醒的事件。

6.4.3 PWR register map

下表列出所有 PWR 寄存器。

表 17. PWR 寄存器地址映射及复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	**PWR_CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBP	PLS[2:0]			PVDE	CSBF	CWUF	PDDS	LPDS
	Reset value																								0	0	0	0	0	0	0	0	0
0x004	**PWR_CSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EWUP2	EWUP1	Res.	Res.	Res.	Res.	Res.	PVDO	SBF	WUF
	Reset value																							0	0						0	0	0

有关寄存器的起始地址参见 *2.2.2 节相关内容*。

7 复位和时钟控制 (RCC)

7.1 复位

有三种复位：系统复位、电源复位和备份域复位。

7.1.1 System reset

系统复位将复位除时钟控制寄存器 CSR 中的复位标志和备份域中的寄存器以外的所有寄存器为它们的复位数值 (见图 6)。

当以下事件中的一件发生时，产生一个系统复位：

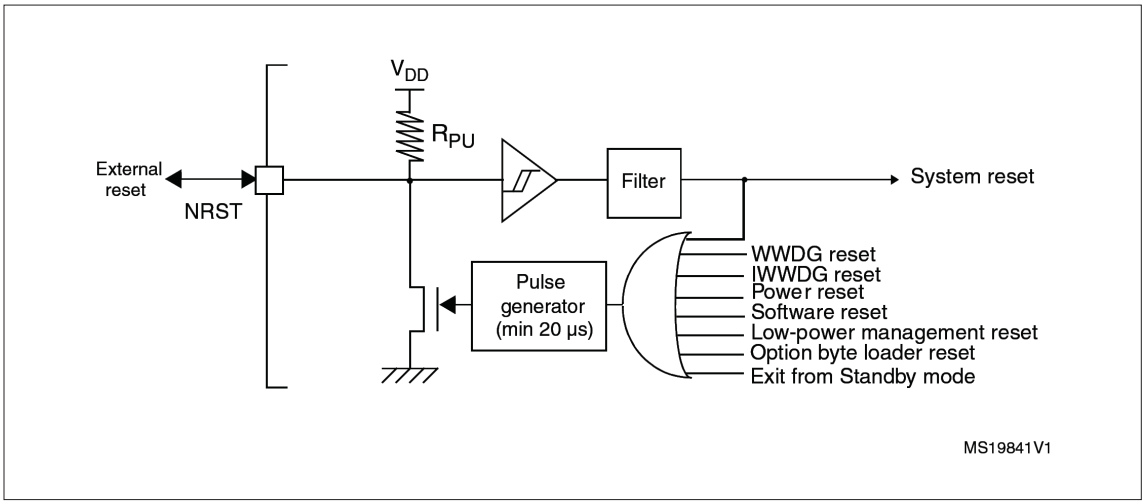
1. NRST 引脚上的低电平 (外部复位)
2. 窗口看门狗事件 (WWDG 复位)
3. 独立看门狗事件 (IWWDG 复位)
4. 软件复位 (SW 复位)
5. 低功耗管理复位
6. 选项字节装载器复位

可通过查看 RCC_CSR(7.4.10 节) 控制状态寄存器中的复位状态标志位识别复位事件来源。

图 9 中的复位源将最终作用于 NRST 引脚，并在一定的延时时段内保持低电平。复位入口地址被固定在 0x0000_0004 处。

内部的复位信号会在 NRST 引脚上输出，脉冲发生器保证每一个 (外部或内部) 复位源都能有至少 20 µ s 的脉冲延时；当 NRST 引脚被拉低产生外部复位时，它将产生复位脉冲。

图 9. 复位电路简化图



软件复位

通过将 Cortex-M0 应用中断和复位控制寄存器中的 SYSRESETREQ 位置 1，可实现软件复位。请参考 Cortex-M0 技术参考手册获得进一步信息。

低功耗管理复位

在以下两种情况下可产生低功耗管理复位：

1. 在进入待机模式时产生低功耗管理复位：
   通过将用户选择字节中的 nRST_STDBY 位置 1 将使能该复位。这时，即使执行了进入待机模式的过程，系统将被复位而不是进入待机模式。
2. 在进入停止模式时产生低功耗管理复位：
   通过将用户选择字节中的 nRST_STOP 位置 1 将使能该复位。这时，即使执行了进入停机模式的过程，系统将被复位而不是进入停机模式。

关于用户选择字节的进一步信息参考第 4 章节。

选项字节装载器复位

在设置 FORCE_OBL(FLASH_CR 寄存器中) 为 1 的情况下，当软件读取选项字时，发出选项字节装载器复位。

7.1.2 电源复位

当以下事件中之一发生时，产生电源复位：

1. 上电 / 掉电复位 (POR/PDR 复位)
2. 从待机模式中返回

电源复位将复位除了备份域外的所有寄存器 (见图 6)

7.1.3 备份域复位

备份域拥有两个专门的复位，它们只影响备份域 (见图 6)。

当以下事件中之一发生时，产生备份域复位：

1. 软件复位，由备份域控制寄存器 (RCC_BDCR) 的 BDRST 位触发。
2. 当 VDD 和 VBAT 都掉电的情况下，VDD 或 VBAT 上电。

7.2 时钟

有三种不同的时钟源可被用来驱动系统时钟：

- 内部高速 (HSI)8MHz RC 振荡器时钟
- 外部高速 (HSE) 振荡器时钟
- PLL 时钟

该类器件还有以下附加的时钟源：

- 40 kHz 低速内部 RC 振荡器 (LSI RC)，用于驱动独立看门狗和用于自动从停机或待机模式下唤醒的 RTC 时钟。
- 低速用于驱动实时时钟 (RTCCLK) 的 32.768 kHz 低速的外部晶体 (LSE 晶体)。
- 专门用于 ADC 的 14 MHz 高速内部 RC 振荡器 (HSI14)。

每种时钟源都可以单独的打开或关断，当它们不用时，可以关断它们来降低功耗。

有多个分频器可用于配置 AHB 和 APB 时钟域，AHB 和 APB 域的最大时钟频率为 48MHz。

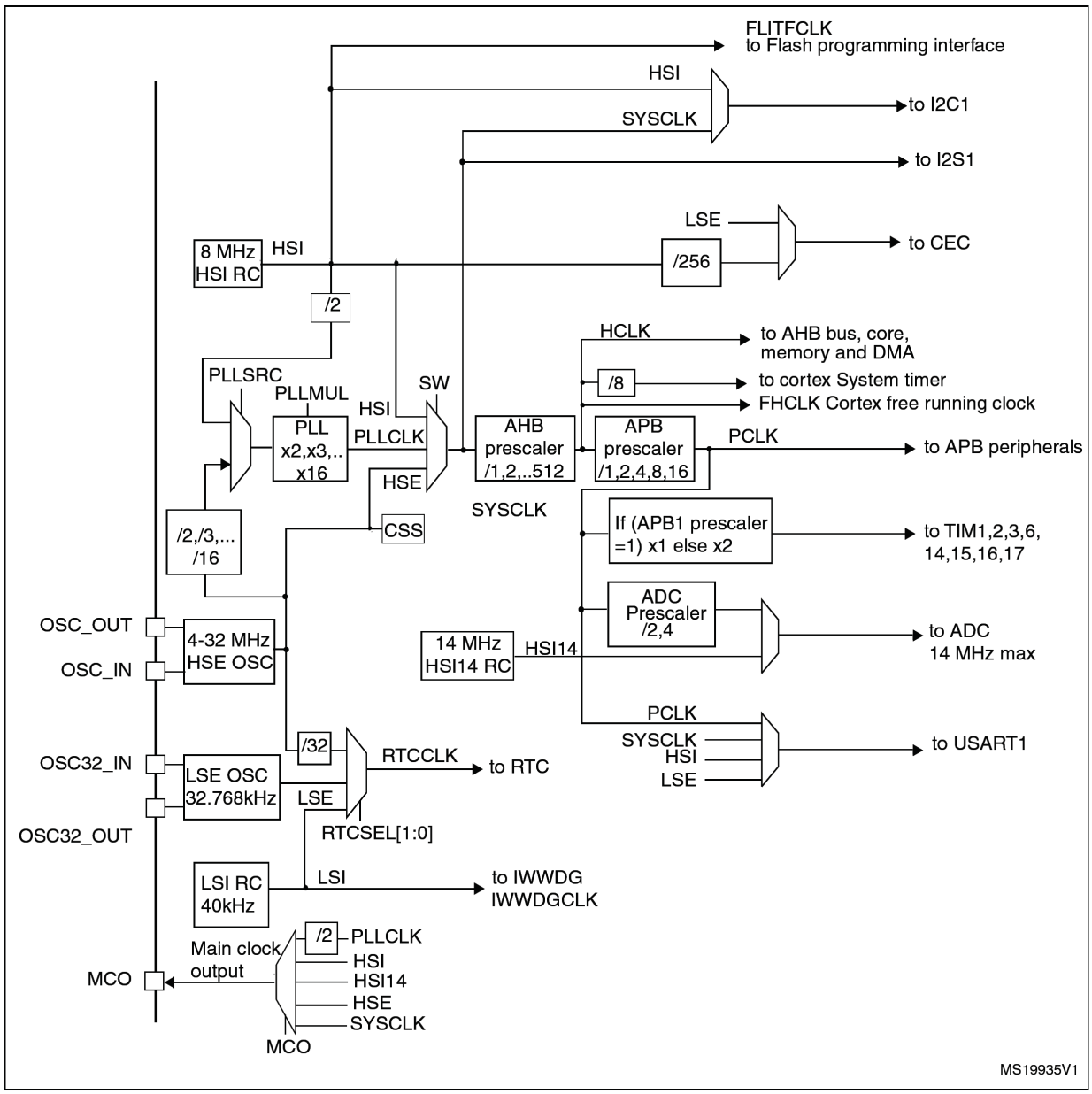
Cortex 系统定时器由 AHB 时钟驱动，其可由 AHB/8 或 AHB 时钟频率直接驱动 (通过 Cortex Systick 配置位来配置)。

所有的外设时钟由其所在的总线时钟 (HCLK 或 PCLK) 驱动，以下几个除外：

- 闪存编程接口时钟 (FLITFCLK) 总是由 HSI 时钟驱动。
- 选项字节装载器时钟也由 HIS 时钟驱动。
- ADC 时钟由下列之一时钟得到 (由软件选择)：
  - 专门的 HSI14 时钟，运行在最大的采样率。
  - APB (PCLK) 时钟除 2 或除 4
- USART1 的时钟为下列的时钟源之一 (由软件选择)：
  - 系统时钟
  - HSI 时钟
  - LSE 时钟
  - APB 时钟 (PCLK)
- I2C1 的时钟为下列的时钟源之一 (由软件选择)：
  - 系统时钟
  - HSI 时钟
- CEC 时钟来自于 HSI/244 或者 LSE。
- I2S1 时钟为系统时钟。
- RTC 时钟来自于 LSE、LSI 或 HSE/32。
- IWWDG 时钟永远来自 LSI。

RCC 通过 AHB 时钟 (HCLK)8 分频后作为 Cortex 系统定时器 (SysTick) 的外部时钟。通过对 SysTick 控制与状态寄存器的设置，可选择上述时钟或 Cortex(HCLK) 时钟作为 SysTick 时钟。

图 10. 时钟树



1. 有关内部和外部时钟源的细节，请参考相关器件数据手册的电气特性章节。

定时器时钟频率分配由硬件按以下 2 种情况自动设置：

1. 如果相应的 APB 预分频系数是 1，定时器的时钟频率与所在 APB 总线频率一致。
2. 否则，定时器的时钟频率被设为与其相连的 APB 总线频率的 2 倍。

FCLK 是 Cortex-M0 的自由运行时钟。详见 ARM 的 Cortex ™ -M0 r0p0 技术参考手册 (TRM)

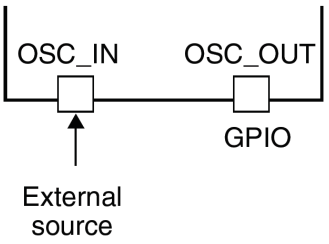
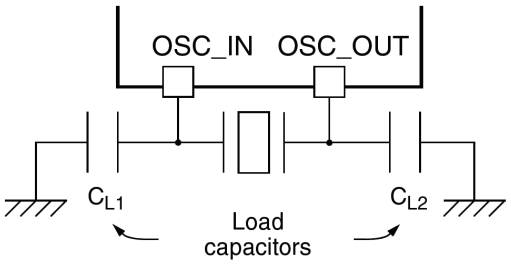
7.2.1 HSE clock

高速外部时钟信号 (HSE) 由以下两种时钟源产生：

- HSE 外部晶体 / 陶瓷谐振器
- HSE 用户外部时钟

为了减少时钟输出的失真和缩短启动稳定时间，晶体 / 陶瓷谐振器和负载电容器必须尽可能地靠近振荡器引脚。负载电容值必须根据所选择的振荡器来调整。

图 11. HSE/ LSE 时钟源

Clock source	Hardware configuration
External clock	
Crystal/Ceramic resonators	

外部晶体 / 陶瓷谐振 (HSE 晶体)

4 到 32 MHz 外部振荡器可为系统提供非常精确的主时钟。

相关的硬件配置如图 11，进一步信息可参考数据手册的电气特性部分。

在时钟控制寄存器 (RCC_CR) 中的 HSERDY 位用来指示高速外部振荡器是否稳定。在启动时，直到这一位被硬件置 1，该时钟才可使用。如果在时钟中断寄存器 (RCC_CIR) 中允许产生中断，将会产生相应中断。

HSE 晶体可以通过设置时钟控制寄存器 (RCC_CR) 中的 HSEON 位被启动和关闭。

外部时钟源 (HSE 旁路)

在这个模式里，必须提供外部时钟。它的频率最高可达 32MHz。用户可通过设置在时钟控制寄存器 (RCC_CR) 中的 HSEBYP 和 HSEON 位来选择这一模式。占空比为 40% ～ 60% 外部时钟信号 (方波、正弦波或三角波) 必须连到 SOC_IN 引脚，同时保证 OSC_OUT 不做为 IO 口使用。见图 11。

7.2.2 HSI clock

HSI 时钟信号由内部 8MHz 的 RC 振荡器产生，可直接作为系统时钟或在 2 分频后作为 PLL 输入。

HSI RC 振荡器能够在不需要任何外部器件的条件下提供系统时钟。它的启动时间比 HSE 晶体振荡器短。然而，即使在校准之后它的时钟频率精度仍较差 (相比于晶体振荡器或陶瓷谐振器)。

校准

制造工艺决定了不同芯片的 RC 振荡器频率会不同，这就是为什么每个芯片的 HSI 时钟频率在出厂前已经被 ST 校准到 1%(25° C) 的原因。

系统复位时，工厂校准值被装载到时钟控制寄存器的 HSICAL[7:0] 位中 (HSICAL 在时钟控制寄存器 RCC_CR 中)。

如果用户的应用基于不同的电压或环境温度，这将会影响 RC 振荡器的精度。可以通过时钟控制寄存器 (RCC_CR) 里的 HSITRIM[4:0] 位来调整 HSI 频率。

时钟控制寄存器 (RCC_CR) 中的 HSIRDY 位用来指示 HSI RC 振荡器是否稳定。在时钟启动过程中，直到这一位被硬件置 1，HSI RC 输出时钟才可以被使用。

SI RC 可由时钟控制寄存器 (RCC_CR) 中的 HSION 位来启动和关闭。

如果 HSE 晶体振荡器失效，HSI 时钟会被作为备用时钟源。参考 7.2.7 节时钟安全系统 (CSS)。

7.2.3 PLL

内部 PLL 可用 HSI 或 HSE 倍频得到，参考图 10 和时钟控制寄存器 (RCC_CR)。

PLL 配置 (选择输入时钟，倍频因子) 须在使能 PLL 前配置好，一旦 PLL 使能，PLL 使用到的这些参数就不能被改变。

改变 PLL 配置过程如下：

1. 设置 PLLON=0，禁用 PLL。
2. 等待 PLLRDY 清 0，PLLRDY 清 0 时才表明 PLL 已经完全停止。
3. 改变 PLL 所需参数。
4. 设置 PLLON=1 重新使能 PLL。

当使能时钟中断寄存器 (RCC_CIR) 的相应位，当 PLL 时钟就绪时会产生一个中断。
PLL 输出频率设置的范围是 16-48 MHz.

7.2.4 LSE 时钟

LSE 晶体是一个 32.768kHz 的低速外部晶体或陶瓷谐振器。它为实时时钟或者其他定时功能提供一个低功耗且精确的时钟源。

晶体振荡器的开和关可用备份域控制寄存器 (RCC_BDCR) 中的 LSEON 位来控制。RCC_BDCR 中的 LSEDRV[1:0] 位用来控制选对 LSE 的驱动能力，其值是系统的鲁棒性、短启动时钟及低功耗折中选择。

备份域寄存器 (RCC_BDCR) 中的 LSERDY 位指示 LSE 晶体振荡是否稳定。在该位被硬件置为 1 之前，LSE 的时钟信号都不被使用。如果在时钟中断寄存器 (RCC_CIR) 里被允许，可产生中断申请。

外部时钟源 (LSE 旁路)

在这个模式里必须提供一个 32.768kHz 频率的外部时钟源。你可以通过设置在备份域控制寄存器 (RCC_BDCR) 里的 LSEBYP 和 LSEON 位来选择这个模式。具有 50% 占空比的外部时钟信号 (方波、正弦波或三角波) 必须连到 OSC32_IN 引脚，同时保证 OSC32_OUT 引脚不作为 IO 口使用，见图 11。

7.2.5 LSI 时钟

LSI RC 做为一个低功耗时钟源，它可以在停机和待机模式下保持运行，为独立看门狗和 RTC 提供时钟。LSI 时钟频率大约为 40kHz(在 30kHz 和 60kHz 之间)。进一步信息请参考数据手册中有关电气特性部分。

LSI RC 振荡器可由时钟控制状态寄存器 (RCC_CSR) 中的 LSION 位来控制开或关。

在时钟控制 / 状态寄存器 (RCC_CSR) 里的 LSIRDY 位指示低速内部振荡器是否稳定。在这个位被硬件设置为 1 之前，LSI 时钟都不能被使用。如果在时钟中断寄存器 (RCC_CIR) 里被允许，将产生 LSI 中断申请。

7.2.6 系统时钟 (SYSCLK) 选择

有三种时钟源可用于驱动系统时钟 (SYSCLK):

- HSI 振荡器
- HSE 振荡器
- PLL

系统复位后，HSI 振荡器被选为系统时钟。当时钟源被直接或通过 PLL 间接作为系统时钟时，它将不能被停止。

时钟的切换只有在目标时钟源可用的情况下才能进行 。假如系统选择了未准备好的时钟源做为当前系统时钟，那么只有在目标时钟源准备好之后才真正执行切换时钟源的操作。时钟控制寄存器 (RCC_CR) 指示当前系统时钟采用哪个时钟源做为系统时钟。

7.2.7 时钟安系统 (CSS)

时钟安全系统可以通过软件被激活。一旦其被激活，时钟监测器将在 HSE 振荡器启动延迟后被使能，并在 HSE 时钟关闭后关闭。

如果 HSE 时钟发生故障，HSE 振荡器被自动关闭，时钟失效事件将被送到高级定时器 (TIM1 和 TIM8) 的刹车输入，并产生时钟安全中断 CSSI，允许软件完成系统的补救处理。此 CSSI 中断连接到 Cortex-M0 的 NMI 中断 (不可屏蔽中断)。

注意： 一旦 CSS 被激活，并且 HSE 时钟出现故障，CSS 中断就产生，并且 NMI 也自动产生。NMI 将被不断执行，直到 CSS 中断挂起位被清除。因此，在 NMI 的处理程序中必须通过设置时钟中断寄存器 (RCC_CIR) 里的 CSSC 位来清除 CSS 中断。

如果 HSE 振荡器被直接或间接地作为系统时钟，(间接的意思是：它被作为 PLL 输入时钟或通过 PLL2，并且 PLL 时钟被作为系统时钟)，时钟故障将导致系统时钟自动切换到 HSI 振荡器，同时外部 HSE 振荡器被关闭。在时钟失效时，如果 HSE 振荡器时钟 (直接的或通过 PLL2) 是作为 PLL 的输入时钟，PLL 也将被关闭。

7.2.8 ADC 时钟

ADC 时钟可从专门的 14 MHz RC 振荡器 (HSI14) 或 PCLK /2(或 /4) 得到。. 当 ADC 时钟源于 PCLK 时，其 ADC 时钟为 PCLK 时钟的反相信号。14MHzr 的 HSI RC 振荡器可以软件配置成由 ADC 接口控制的打开 / 关闭 (自动关) 模式或者常开模式。当 APB 时钟被选为内核时钟时，14MHz HSI RC 振荡器不能被 ADC 接口打开。

7.2.9 RTC 时钟

通过设置备份域控制寄存器 (RCC_BDCR) 里的 RTCSEL[1:0] 位，RTCCLK 时钟源可以由 HSE/32、LSE 或 LSI 时钟提供。除非备份域复位，此选择不能被改变。系统必须按 PCLK 的频率须快于或等于 RTCCLK 的频率的方式配置才能正确操作 RTC。

LSE 时钟属于备份域的，但 HSE 和 LSI 时钟不属于备份域时钟，因此：

- 若 LSE 被选为 RTC 时钟：
  - 只要维持 VBAT 正常供电，即使 VDD 掉电，RTC 仍会继续工作。
- 若 LSI 被 选为 RTC 时钟：
  - 当 VDD 掉电时，RTC 处于不定的状态。
- 若 HSE/32 被选为 RTC 时钟：
  - 当 VDD 掉电或内部电压调压器 (1.8V 域的供电切断) 掉电时，RTC 处于不定的状态。

7.2.10 看门狗时钟

如果独立看门狗已经由硬件选项或软件启动，LSI 振荡器将被强制在打开状态，并且不能被关闭。在 LSI 振荡器稳定后，时钟供应给 IWWDG。

7.2.11 时钟输出

微控制器允许输出时钟信号到外部 MCO 引脚。对应 MCO 的 GPIO 口须配置为备用的功能模式。有如下的 5 种信号可选为 MCO 时钟输出：

- HSI14
- SYSCLK
- HSI
- HSE
- PLL / 2

MCO 时钟的选择由时钟配置寄存器 (RCC_CFGR) 的 MCO[2:0] 位决定。

7.3 低功耗模式

APB 外设时钟和 DMA 时钟可用软件禁止。

睡眠模式停止 CPU 时钟。在 CPU 睡眠中存储器接口时钟 (Flash 和 RAM 接口) 可被停止。当连接到 APB 所有外设的时钟禁止后，当进入睡眠期间 AHB 到 APB 桥时钟由 CPU 的硬件关闭。

CPU 进入停止模式时停止 V18 域、PLL、HSI、HSI14 和 HSE 振荡器的时钟。

HDMI CEC, USART1 和 I2C1 即使在 MCU 进入停止模式 下仍有能力打开 HIS 振荡器 (假如 HIS 被选为这些外设的时钟).

在 LSE 振荡器已使能的情况下，HDMI CEC 和 USART1 当在系统进入停止模式下也可由 LSE 振荡器驱动 (假如 LSE 被选为这些外设时钟)。但是这些外设没有打开 LSE 振荡器的能力。

CPU 进入待机模式时停止 V18 域、PLL、HSI、HSI14 和 HSE 振荡器的时钟。

当设置 DBGMCU_CR 寄存器中的 DBG_STOP 或 DBG_STANDBY 位，那么 CPU 在相应的深度睡眠模式下也可以具有调试功能。

当系统由中断 (停止模式) 或复位 (待机模式) 唤醒后，HSI 振荡器被选为系统时钟 (不管进入停止模式或待机模式前选用的是何种时钟)。

假如当前正在进行闪存编程，只有在闪存编程全部完成之后才会进入深度睡眠模式 (深度睡眠延后)。若当前正在使用 APB 域，那么只有全部完成 APB 域的操作后才进入深度睡眠模式。

7.4 RCC 寄存器

请参考第 1.1 章节中有关寄存器描述中用到的缩写

7.4.1 时钟控制寄存器 (RCC_CR)

偏移地址： 0x00

复位值： 0x0000 XX83 X 表示未定义

访问：无等待状态，字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	PLL RDY	PLLON	Res	Res	Res	Res	CSS ON	HSE BYP	HSE RDY	HSE ON
						r	rw					rw	rw	r	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSICAL[7:0]								HSITRIM[4:0]					Res	HSI RDY	HSION
r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw		r	rw

位 31:26 保留 , 必须保持为复位值。

位 25 PLLRDY: PLL 时钟就绪标志

硬件置 1 表明 PLL 被锁定。

0: PLL 未锁定

1: PLL 锁定

位 24 PLLON: PLL 使能

由软件设置或清零。

当进入停止或待机模式时由硬件清零。当 PLL 时钟正做为系统时钟或被选用将做为系统时钟时，该位不能被清零。

0: PLL 关闭

1: PLL 使能

位 23:20 保留 , 必须保持为复位值。

位 19 CSSON: 时钟安全系统使能

同软件设置或清零。当 CSSON=1，HSE 振荡器就绪时，时钟检测器由硬件打开，当检测到 HSE 时钟错误时清零。

0: 时钟检测器关闭

1: 时钟检测器打开 (假如 HSE 就绪，时钟检测器打开。若未就绪则关闭)。

位 18 HSEBYP: 外部高速时钟旁路

由软件设置和清零。外部时钟必须用 HSEON=1 打开， HSEBYP 位只能在 HSE 振荡器关闭的情况下使用。

0: HSE 晶体振荡器无旁路

1: HSE 晶体振荡器旁路

位 17 HSERDY: HSE 时钟就绪标志
由硬件设置，表明 HSE 振荡器是否稳定。当 HSEON 清零后，该位需要 6 个 HSE 振荡器周期才清零。
0: HSE 振荡器未就绪
1: HSE 振荡器就绪

位 16 HSEON: HSE 时钟使能
由软件设置和清零。
当进入停止或待机模式后由硬件清除并停止 HSE 振荡器。当直接或简接使用 HSE 时钟时，该位不能被清零。
0: HSE 振荡器关闭
1: HSE 振荡器开启

位 15:8 HSICAL[7:0]: HSI 时钟校准
在启动时这些位被自动初始化为出厂值校准值。

位 7:3 HSITRIM[4:0]: HSI 时钟调整
这些位在 HSICAL[7:0] 的基础上，让用户可以输入一个调整数值，根据电压和温度的变化调整内部 HSI RC 振荡器的频率。 默认数值为 16，可以把 HSI 调整到 8MHz±1%；每步 HSICAL 的变化调整约 40kHz。

位 2 保留 , 必须保持为复位值。

位 1 HSIRDY: HSI 时钟就绪标志
由硬件置 1 来指示内部 HSI 振荡器已经稳定。在 HSION 位清零后，HSIRDY 位需要 6 个 HSI 振荡器周期清零。.
0: HSI 振荡器未就绪
1: HSI 振荡器就绪

位 0 HSION: HSI 时钟使能
由软件设置和清零。
当从待机和停止模式返回或用作系统时钟的 HSE 振荡器发生故障时，该位由硬件置 1 来启动 HSI 振荡器。当 HSI 振荡器被直接或间接地用作或被选择将要作为系统时钟时，该位不能被清零。
0: HSI 振荡器关闭
1: HSI 振荡器开启

7.4.2 时钟配置寄存器 (RCC_CFGR)

偏移地址 : 0x04

复位值 : 0x0000 0000

访问 : 0 到 2 个等待周期 , 字 , 半字和字节访问

只有当访问发生在时钟切换时，才会插入 1 或 2 个等待周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	MCO[2:0]			Res	Res	PLLMUL[3:0]				PLL XTPRE	PLL SRC
					rw	rw	rw			rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	ADCPRE	Res	Res	Res	PPRE[2:0]			HPRE[3:0]				SWS[1:0]		SW[1:0]	
	rw				rw	rw	rw	rw	rw	rw	rw	r	r	rw	rw

位 31:27 保留 , 必须保持为复位值。

位 26:24 MCO: 微控制器时钟输出 (Microcontroller clock output)

由软件置 1 或清零。

000: 时钟 输出禁止，MCO 引脚上没有时钟输出

001: 保留

010: 保留

011: HSI14

100: 系统时钟 (SYSCLK)

101: HSI 时钟

110: HSE 时钟

111: PLL / 2

注： 在启动和切换 MCO 时钟源时会被切断。

位 23:22 保留 , 必须保持为复位值。

位 21:18 PLLMUL: PLL 倍频系数 (PLL multiplication factor)
由软件设置来确定 PLL 倍频系数。只有在 PLL 关闭的情况下才可被写入。
注意： PLL 输出最大频率不能超过 48 MHz.
0000: PLL 输入时钟的 2 倍频
0001: PLL 输入时钟的 3 倍频
0010: PLL 输入时钟的 4 倍频
0011: PLL 输入时钟的 5 倍频
0100: PLL 输入时钟的 6 倍频
0101: PLL 输入时钟的 7 倍频
0110: PLL 输入时钟的 8 倍频
0111: PLL 输入时钟的 9 倍频
1000: PLL 输入时钟的 10 倍频
1001: PLL 输入时钟的 11 倍频
1010: PLL 输入时钟的 12 倍频
1011: PLL 输入时钟的 13 倍频
1100: PLL 输入时钟的 14 倍频
1101: PLL 输入时钟的 15 倍频
1110: PLL 输入时钟的 16 倍频
1111: PLL 输入时钟的 16 倍频

位 17 PLLXTPRE: HSE 分频器作为 PLL 输入 (HSE divider for PLL input clock)
由软件置 1 或清 0 来分频 HSE 后作为 PLL 输入时钟。只能在关闭 PLL 时才能写入此位。
注： 该位与时钟配置寄存器 2(RCC_CFGR2) 中的 PREDIV 最低位是一样的 (为了兼容其他 STM32 产品)
0: HSE 不分频
1: HSE 2 分频

位 16 PLLSRC: PLL 输入时钟源 (PLL entry clock source)
由软件置’1’或清’0’来选择 PLL 输入时钟源。只能在关闭 PLL 时才能写入此位。
0：HIS/2 作为 PLL 输入时钟
1：HSE/PREDIV 作为 PLL 输入时钟 (参考 7.4.12 章节的时钟配置寄存器 2)

位 15 保留 , 必须保持为复位值。

位 14 ADCPRE: ADC 预分频 (ADC prescaler)
由软件设置和清零来选择 ADC 的时钟频率。
0: PCLK / 2
1: PCLK / 4

位 13:11 保留 , 必须保持为复位值。

位 10:8 PPRE: PCLK 预分频 (PCLK prescaler)
由软件设置和清零来控制 APB 时钟 (PCLK) 的除数因子。
0xx: HCLK 不分频
100: HCLK / 2
101: HCLK / 4
110: HCLK / 8
111: HCLK / 16

位 7:4 HPRE: HCLK 预分频 (HCLK prescaler)
由软件设置和清零来控制 AHB 时钟的除数因子。
0xxx: SYSCLK 不分频
1000: SYSCLK / 2
1001: SYSCLK / 4
1010: SYSCLK / 8
1011: SYSCLK / 16
1100: SYSCLK / 64
1101: SYSCLK / 128
1110: SYSCLK / 256
1111: SYSCLK / 512
注：当 AHB 时钟的预分频系数大于 1 时，必须开启预取缓冲器。详见读操作章节。

位 3:2 SWS: 系统时钟切换状态 (System clock switch status)
由硬件置 1 或清 0 来指示哪一个时钟源被作为系统时钟。
00：HSI 作为系统时钟
01：HSE 作为系统时钟
10：PLL 输出作为系统时钟
11：不可用。

位 1:0 SW: 系统时钟切换 (System clock switch)
由软件置 1 或清 0 来选择系统时钟源。
在从停止或待机模式中返回时或直接或间接作为系统时钟的 HSE 出现故障时，由硬件强制选择 HSI 作为系统时钟 (如果时钟安全系统已经启动)
00：HSI 作为系统时钟
01：HSE 作为系统时钟
10：PLL 输出作为系统时钟
11：不可用

7.4.3 时钟中断寄存器 (RCC_CIR)

偏移地址 : 0x08

复位值 : 0x0000 0000

访问 : 无等待周期 , 字 , 半字 和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	CSSC	Res	HSI14 RDYC	PLL RDYC	HSE RDYC	HSI RDYC	LSE RDYC	LSI RDYC
								w		w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	HSI14 RDYIE	PLL RDYIE	HSE RDYIE	HSI RDYIE	LSE RDYIE	LSI RDYIE	CSSF	Res	HSI14 RDYF	PLL RDYF	HSE RDYF	HSI RDYF	LSE RDYF	LSI RDYF
		rw	rw	rw	rw	rw	rw	r		r	r	r	r	r	r

位 31:24 保留 , 必须保持为复位值。

位 23 CSSC: 清除时钟安全系统中断 (Clock security system interrupt clear)
由软件置 1 来清除安全系统中断标志位 CSSF。
0: 无作用
1: 清 CSSF 标志

位 22 保留 , 必须保持为复位值。.

位 21 HSI14RDYC: HSI 14 MHz 就绪中断清除
由软件置 1 来清除 HSI14RDYF 标志 .
0: 无作用
1: 清 HSI14RDYF 标志

位 20 PLLRDYC: PLL 就绪中断清除
由软件置 1 来清除 PLLRDYF 标志 .
0: 无作用
1: 清 PLLRDYF 标志

位 19 HSERDYC: HSE 就绪中断清除
由软件置 1 来清除 HSERDYF 标志 .
0: 无作用
1: 清 HSERDYF 标志

位 18 HSIRDYC: HSI 就绪中断清除
由软件置 1 来清除 HSIRDYF 标志 .
0: 无作用
1: 清 HSIRDYF 标志

位 17 LSERDYC: LSE 就绪中断清除
由软件置 1 来清除 LSERDYF 标志 .
0: 无作用
1: 清 LSERDYF 标志

位 16 LSIRDYC: LSI ready interrupt clear
由软件置 1 来清除 LSIRDYF 标志
0: 无作用
1: 清除 LSIRDYF 标志

位 15:14 保留 , 必须保持为复位值。

位 13 HSI14RDYIE: HSI14 就绪中断使能
由软件设置和清除来使能 / 关闭由 HSI14 振荡器引发的就绪中断
0: HSI14 就绪中断关闭
1: HSI14 就绪中断使能

位 12 PLLRDYIE: PLL 就绪中断使能
由软件设置和清除来使能 / 关闭由 PLL 锁定引发的中断 .
0: PLL 锁定中断关闭
1: PLL 锁定中断使能

位 11 HSERDYIE: HSE 就绪中断使能
由软件设置和清除来使能 / 关闭由 HSE 振荡器引发的就绪中断
0: HSE 就绪中断关闭
1: HSE 就绪中断使能

位 10 HSIRDYIE: HSI 就绪中断使能
由软件设置和清除来使能 / 关闭由 HSI 振荡器引发的就绪中断
0: HSI 就绪中断关闭
1: HSI 就绪中断使能

位 9 LSERDYIE: LSE 就绪中断使能
由软件设置和清除来使能 / 关闭由 LSE 振荡器引发的就绪中断 .
0: LSE 就绪中断关闭
1: LSE 就绪中断使能

位 8 LSIRDYIE: LSI 就绪中断使能
由软件设置和清除来使能 / 关闭由 LSI 振荡器引发的就绪中断 .
0: LSI 就绪中断关闭
1: LSI 就绪中断使能

位 7 CSSF: 时钟安全系统中断标志
当系统检测到 HSE 振荡器错误时由硬件置位该位。
由软件对 CSSC 置 1 时清除该位。
0: 无由 HSE 错误引发的时钟安全系统中断
1: 有由 HSE 错误引发的时钟安全系统中断

位 6 保留 , 必须保持为复位值。

位 5 HSI14RDYF: HSI14 就绪中断标志
当 HSI14 时钟趋稳定、HSI14RDYDIE=1 且 HSI14ON =1((RCC_CFGR2) 时由硬件对该位置 1。当 HSI14ON =0, 无论 HSI14 是否已稳定该位都一直为 0。
软件置 HSI14RDYC=1 时该位清除。
0: 无由 HSI14 振荡器引发的时钟就绪中断
1: 有由 HSI14 振荡器引发的时钟就绪中断

位 4 PLLRDYF: PLL ready interrupt flag
当 PLL 时钟就绪且 PLLRDYDIE=1 时由硬件对该位置 1。
软件置 PLLRDYC=1 时该位清除。
0: 无由 PLL 时钟引发的时钟就绪中断
1: 有由 PLL 时钟引发的时钟就绪中断

位 3 HSERDYF: HSE 就绪中断标志
当 HSE 时钟就绪且 HSERDYDIE=1 时由硬件对该位置 1。
软件置 HSERDYC=1 时该位清除。
0: 无由 HSE 振荡器引发的时钟就绪中断
1: 有由 HSE 振荡器引发的时钟就绪中断

位 2 HSIRDYF: HSI 就绪中断标志
当 HSI 时钟趋稳定、HSIRDYDIE=1 且 HSION=1(RCC_CR) 时由硬件对该位置 1。
当 HSION=0, 无论 HSI 是否已稳定该位都一直为 0。
软件置 HSIRDYC=1 时该位清除。
0: 无由 HSI 振荡器引发的时钟就绪中断
1: 有由 HSI 振荡器引发的时钟就绪中断

位 1 LSERDYF: LSE 就绪中断标志
当 LSE 时钟就绪且 LSERDYDIE=1 时由硬件对该位置 1。
软件置 LSERDYC=1 时该位清除。
0: 无由 LSE 振荡器引发的时钟就绪中断
1: 有由 LSE 振荡器引发的时钟就绪中断

位 0 LSIRDYF: LSI 就绪中断标志
当 LSI 时钟就绪且 LSIRDYDIE=1 时由硬件对该位置 1。
软件置 LSIRDYC=1 时该位清除。
0: 无由 LSI 振荡器引发的时钟就绪中断
1: 有由 LSI 振荡器引发的时钟就绪中断

7.4.4 APB2 外设复位寄存器 (RCC_APB2RSTR)

偏移地址 : 0x0C
复位值 : 0x00000 0000
访问：无等待周期，字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	DBG MCU RST	Res	Res	Res	TIM17 RST	TIM16 RST	TIM15 RST
									rw				rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	USART1 RST	Res	SPI1 RST	TIM1 RST	Res	ADC RST	Res	Res	Res	Res	Res	Res	Res	Res	SYS CFG RST
	rw		rw	rw		rw									rw

位 31:23 保留 , 必须保持为复位值。

位 22 DBGMCURST: 调试 MCU 复位
由软件置 1 或清 0.
0: 无作用
1: 复位调试 MCU
位 21:19 保留，必须保持为复位值。
位 18 TIM17RST: TIM17 定时器复位
由软件置 1 或清 0.
0: 无作用
1: 复位 TIM17 定时器
位 17 TIM16RST: TIM16 定时器复位
由软件置 1 或清 0.
0: 无作用
1: 复位 TIM16 定时器
位 16 TIM15RST: TIM15 定时器复位
由软件置 1 或清 0.
0: 无作用
1: 复位 TIM15 定时器 timer
位 15 保留，必须保持为复位值。
位 14 USART1RST: USART1 复位
由软件置 1 或清 0.
0: 无作用
1: 复位 USART1
位 13 保留，必须保持为复位值。
位 12 SPI1RST: SPI1 复位
由软件置 1 或清 0.
0: 无作用
1: 复位 SPI1
位 11 TIM1RST: TIM1 定时器复位
由软件置 1 或清 0.
0: 无作用
1: 复位 TIM1 定时器
位 10 保留，必须保持为复位值。
位 9 ADCRST: ADC 接口复位
由软件置 1 或清 0.
0: 无作用
1: 复位 ADC 接口
位 8:1 保留，必须保持为复位值。
位 0 SYSCFGRST: SYSCFG and COMP 复位
由软件置 1 或清 0.
0: 无作用
1: 复位 SYSCFG 和 COMP

7.4.5 APB1 外设复位寄存器 (RCC_APB1RSTR)

偏移地址 : 0x10

复位值 : 0x0000 0000

访问：无等待周期，字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	CECRST	DACRST	PWRRST	Res	Res	Res	Res	Res	I2C2RST	I2C1RST	Res	Res	Res	USART2RST	Res
	rw	rw	rw						rw	rw				rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	SPI2RST	Res	Res	WWDGRST	Res	Res	TIM14RST	Res	Res	Res	TIM6RST	Res	Res	TIM3RST	TIM2RST
	rw			rw			rw				rw			rw	rw

位 31 保留 , 必须保持为复位值。

位 30 CECRST HDMI CEC 复位
由软件置 1 或清 0.
0: 无作用
1: 复位 HDMI CEC

位 29 DACRST: DAC 接口复位
由软件置 1 或清 0.
0: 无作用
1: 复位 DAC 接口

位 28 PWRRST: 电源接口复位
由软件置 1 或清 0.
0: 无作用
1: 复位电源接口

位 27:23 保留 , 必须保持为复位值。

位 22 I2C2RST: I2C2 复位
由软件置 1 或清 0.
0: 无作用
1: 复位 I2C2

位 21 I2C1RST: I2C1 复位
由软件置 1 或清 0.
0: 无作用
1: 复位 I2C1

位 20:18 保留 .

位 17 USART2RST: USART2 复位
由软件置 1 或清 0.
0: 无作用
1: 复位 USART2

位 16:15 保留 , 必须保持为复位值。

位 14 SPI2RST: SPI2 复位
由软件置 1 或清 0.
0: 无作用
1: 复位 SPI2
位 13:12 保留 , 必须保持为复位值。
位 11 WWDGRST: 窗口看门狗复位
由软件置 1 或清 0.
0: 无作用
1: 复位窗口看门够
位 10:9 保留 , 必须保持为复位值。
位 8 TIM14RST: TIM14 定时器复位
由软件置 1 或清 0.
0: 无作用
1: 复位 TIM14
位 7:5 保留 , 必须保持为复位值。
位 4 TIM6RST: TIM6 定时器复位
由软件置 1 或清 0.
0: 无作用
1: 复位 TIM6
位 3:2 保留 , 必须保持为复位值。
位 1 TIM3RST: TIM3 定时器复位
由软件置 1 或清 0.
0: 无作用
1: 复位 t TIM3
位 0 TIM2RST: TIM2 定时器复位
由软件置 1 或清 0.
0: 无作用
1: 复位 TIM2

7.4.6 AHB 外部时钟使能寄存器 (RCC_AHBENR)

偏移地址 : 0x14
复位值 : 0x0000 0014
访问：无等待周期，字，半字和字节访问

注： 当外设时钟没有启用时，软件不能读出外设寄存器的数值，返回的数值始终是 0x0。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	TSCEN	Res	IOPFEN	Res	IOPDEN	IOPCEN	IOPBEN	IOPAEN	Res
							rw		rw		rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	Res	Res	Res	CRCEN	Res	FLITFEN	Res	SRAMEN	Res	DMAEN
									rw		rw		rw		rw

位 31:25 保留，必须保持为复位值。

位 24 TSCEN: 触摸传感控制器时钟使能
由软件置 1 或清 0.
0: TSC 时钟关闭
1: TSC 时钟开启

位 23 保留，必须保持为复位值。

位 22 IOPFEN: GPIOF 时钟使能
由软件置 1 或清 0.
0: GPIOF 时钟关闭
1: GPIOF 时钟开启

位 21 保留，必须保持为复位值。

位 20 IOPDEN: GPIOD 时钟使能
由软件置 1 或清 0.
0: GPIOD 时钟关闭
1: GPIOD 时钟开启

位 19 IOPCEN: GPIOC 时钟使能
由软件置 1 或清 0.
0: GPIOC 时钟关闭
1: GPIOC 时钟开启

位 18 IOPBEN: GPIOB 时钟使能
由软件置 1 或清 0.
0: GPIOB 时钟关闭
1: GPIOB 时钟开启

位 17 IOPAEN: GPIOA 时钟使能
由软件置 1 或清 0.
0: GPIOA 时钟关闭
1: GPIOA 时钟开启

位 16:7 保留，必须保持为复位值。

位 6 CRCEN: CRC 时钟使能
由软件置 1 或清 0.
0: CRC 时钟关闭
1: CRC 时钟开启

位 5 保留，必须保持为复位值。

位 4 FLITFEN: FLITF 时钟使能
由软件置 1 或清 0 来关闭 / 开启在睡眠模式下的 FLITF 时钟
0: 在睡眠模式下 FLITF 时钟关闭
1: 在睡眠模式下 FLITF 时钟开启

位 3 保留，必须保持为复位值。

位 2 SRAMEN: SRAM 接口时钟使能
由软件置 1 或清 0 来关闭 / 开启在睡眠模式下的 SRAM 时钟
0: 在睡眠模式下 SRAM 接口时钟关闭
1: 在睡眠模式下 SRAM 接口时钟开启

位 1 保留，必须保持为复位值。

位 0 DMAEN: DMA 时钟使能
由软件置 1 或清 0.
0: DMA 时钟关闭
1: DMA 时钟开启

7.4.7 APB2 外设时钟使能寄存器 (RCC_APB2ENR)

偏移地址 : 0x18
复位值 : 0x0000 0000
访问 : 字，半字和字节访问
无等待周期，除了出现先前的 APB 访问未完成的情况下必须插入等待直至先前的 APB 外设访问完成。

注： 当外设时钟没有启用时，软件不能读出外设寄存器的数值，返回的数值始终是 0x0。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	DBGMCUEN	Res	Res	Res	TIM17EN	TIM16EN	TIM15EN
									rw				rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	USART1EN	Res	SPI1EN	TIM1EN	Res	ADCEN	Res	Res	Res	Res	Res	Res	Res	Res	SYSCFGEN
	rw		rw	rw		rw									rw

位 31:23 保留 , 必须保持为复位值。
位 22 DBGMCUEN MCU 调试模块时钟使能
由软件置 1 或清 0.
0: MCU 调试模块时钟关闭
1: MCU 调试模块时钟开启
位 21:19 保留 , 必须保持为复位值。
位 18 TIM17EN: TIM17 定时器时钟使能
由软件置 1 或清 0.
0: TIM17 定时器时钟关闭
1: TIM17 定时器时钟开启
位 17 TIM16EN: TIM16 定时器时钟使能
由软件置 1 或清 0.
0: TIM16 定时器时钟关闭
1: TIM16 定时器时钟开启
位 16 TIM15EN: TIM15 定时器时钟使能
由软件置 1 或清 0.
0: TIM15 定时器时钟关闭
1: TIM15 定时器时钟开启
位 15 保留 , 必须保持为复位值。

位 14 USART1EN: USART1 时钟使能
由软件置 1 或清 0.
0: USART1 时钟关闭
1: USART1 时钟开启
位 13 保留 , 必须保持为复位值。
位 12 SPI1EN: SPI1 时钟使能
由软件置 1 或清 0.
0: SPI1 时钟关闭
1: SPI1 时钟开启
位 11 TIM1EN: TIM1 定时器时钟使能
由软件置 1 或清 0.
0: TIM1 定时器时钟关闭
1: TIM1 定时器时钟开启
位 10 保留 , 必须保持为复位值。
位 9 ADCEN: ADC 接口时钟使能
由软件置 1 或清 0.
0: ADC 接口时钟关闭
1: ADC 接口时钟开启
位 8:1 保留 , 必须保持为复位值。
位 0 SYSCFGEN: SYSCFG 时钟使能
由软件置 1 或清 0.
0: SYSCFG 时钟关闭
1: SYSCFG 时钟开启

7.4.8 APB1 外设时钟使能寄存器 (RCC_APB1ENR)

偏移地址 : 0x1C
复位值 : 0x0000 0000
访问 : 字，半字和字节访问
无等待周期，除了出现先前的 APB1 访问未完成的情况下必须插入等待直至先前的 APB1 外设 访问完成。

注： 当外设时钟没有启用时，软件不能读出外设寄存器的数值，返回的数值始终是 0x0。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	CEC EN	DAC EN	PWR EN	Res	Res	Res	Res	Res	I2C2 EN	I2C1 EN	Res	Res	Res	USART 2EN	Res
	rw	rw	rw						rw	rw				rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	SPI2 EN	Res	Res	WWD GEN	Res	Res	TIM14 EN	Res	Res	Res	TIM6 EN	Res	Res	TIM3 EN	TIM2 EN
	rw			rw			rw				rw			rw	rw

位 31 保留，必须保持为复位值。
位 30 CECEN: HDMI CEC 接口时钟使能
由软件置 1 或清 0.
0: HDMI CEC 时钟关闭
1: HDMI CEC 时钟开启
位 29 DACEN: DAC 接口时钟使能
由软件置 1 或清 0.
0: DAC 接口时钟关闭
1: DAC 接口时钟开启
位 28 PWREN: Power 接口时钟使能
由软件置 1 或清 0.
0: Power 接口时钟关闭
1: Power 接口时钟开启
位 27:24 保留，必须保持为复位值。
位 22 I2C2EN: I2C2 时钟使能
由软件置 1 或清 0.
0: I2C2 时钟关闭
1: I2C2 时钟开启
位 21 I2C1EN: I2C1 时钟使能
由软件置 1 或清 0.
0: I2C1 时钟关闭
1: I2C1 时钟开启
位 20:18 保留，必须保持为复位值。
位 17 USART2EN: USART2 时钟使能
由软件置 1 或清 0.
0: USART2 时钟关闭
1: USART2 时钟开启
位 16:15 保留，必须保持为复位值。
位 14 SPI2EN: SPI2 时钟使能
由软件置 1 或清 0.
0: SPI2 时钟关闭
1: SPI2 时钟开启
位 13:12 保留，必须保持为复位值。
位 11 WWDGEN: 看门狗时钟使能
由软件置 1 或清 0.
0: 看门狗时钟关闭
1: 看门狗时钟开启
位 10:9 保留，必须保持为复位值。
位 8 TIM14EN: TIM14 定时器时钟使能
由软件置 1 或清 0.
0: TIM14 定时器时钟关闭
1: TIM14 定时器时钟开启
位 7:5 保留，必须保持为复位值。

位 4 TIM6EN: TIM6 定时器时钟使能
由软件置 1 或清 0.
0: TIM6 时钟关闭
1: TIM6 时钟开启

位 3:2 保留 , 必须保持为复位值。

位 1 TIM3EN: TIM3 定时器时钟使能
由软件置 1 或清 0.
0: TIM3 时钟关闭
1: TIM3 时钟开启

位 0 TIM2EN: TIM2 定时器时钟使能
由软件置 1 或清 0.
0: TIM2 时钟关闭
1: TIM2 时钟开启

7.4.9 备份域控制寄存器 (RCC_BDCR)

偏移地址 : 0x20

复位值 : 0x0000 0000, 由备份域复位电路复位

访问：0 到 3 个等待周期，字、半字和字节访问

当连续对该寄存器进行访问时，将插入等待状态。

注：　备份域控制寄存器(RCC_BDCR)的LSEON, LSEBYP, RTCSEL 和RTCEN位处于备份域。因此，这些位在复位后处于写保护状态。只有在电源控制寄存器 (PWR_CR) 中的 DBP 位置为 1 后才能对这些位进行改动。进一步信息请参考 6.1.2 节。这些位仅在备份域复位 (见 7.1.3 节的备份域复位) 后才清 0，任何内部或外部复位都不会影响这些位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	BDRST
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTC EN	Res	Res	Res	Res	Res	RTCSEL[1:0]		Res	Res	Res	LSEDRV[1:0]		LSE BYP	LSE RDY	LSEON
rw						rw	rw				rw	rw	rw	r	rw

位 31:17 保留 , 必须保持为复位值。

位 16 BDRST: 备份域软件复位

由软件置 1 或清 0.

0: 复位未激活

1: 复位整个备份域

位 15 RTCEN: RTC 时钟使能

由软件置 1 或清 0.

0: RTC 时钟关闭

1: RTC 时钟开启

位 14:10 保留 , 必须保持为复位值。

位 9:8 RTCSEL[1:0]: RTC 时钟源选择

由软件设置来选择 RTC 时钟源。一旦 RTC 时钟源被选定，这些位值不能被改变，除非备份域被复位。可通过设置 BDRST 位来复位备份域。

00: 无时钟

01: LSE 振荡器作为 RTC 时钟

10: LSI 振荡器作为 RTC 时钟

11: HSE / 32 作为 RTC 时钟

位 7:5 保留 , 必须保持为复位值。

位 3:3 LSEDRV LSE 振荡器驱动能力

由软件设置或清除 LSE 振荡器驱动能力设置。当复位备份域时，会重装载缺省值。

00: ‘晶体模式’ 弱驱动能力

01: ‘晶体模式’ 中低驱动能力

10: ‘晶体模式’ 中高驱动能力

11: ‘晶体模式’ 强驱动 (复位后的缺省值)

注：振荡器是晶体模式不为旁路模式。

位 2 LSEBYP: LSE 振荡器旁路

由软件设置和清除，该位仅在外部 32KHz 振荡器关闭的情冲下写值。

0: LSE 振荡器未被旁路

1: LSE 振荡器被旁路

位 1 LSERDY: LSE 振荡器就绪

由硬件置 1 或清 0 来指示是否外部 32kHz 振荡器就绪。在 LSEON 被清零后，该位需要 6 个外部低速振荡器的周期才被清零。

0: LSE 振荡器未就绪

1: LSE 振荡器就绪

位 0 LSEON: LSE 振荡器使能

由软件置 1 或清 0.

0: LSE 振荡器关闭

1: LSE 振荡器开启

7.4.10 控制 / 状态寄存器 (RCC_CSR)

偏移地址 : 0x24

复位值 : 0x0C00 0000, 除复位标志外由系统复位复位，复位标志只能由电源复位清除

访问：0 到 3 等待周期，字、半字和字节访问

当连续对该寄存器进行访问时，将插入等待状态

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPWR RSTF	WWDG RSTF	IW WDG RSTF	SFT RSTF	POR RSTF	PIN RSTF	OB LRSTF	RMVF	Res	Res	Res	Res	Res	Res	Res	Res
rw	rw	rw	rw	rw	rw	rw	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	LSI RDY	LSION
														r	rw

位 31 LPWRRSTF: 低功耗复位标志

在低功耗管理复位发生时由硬件置 1。

由软件通过写 RMVF 位清除该位。

0：无低功耗管理复位发生

1：发生低功耗管理复位

关于低功耗管理复位的详细信息，请参考低功耗管理复位章节

位 30 WWDGRSTF: 窗口看门狗复位标志

在窗口看门狗复位发生时由硬件置 1。

由软件通过写 RMVF 位清除该位。

0：无窗口看门狗复位发生

1：发生窗口看门狗复位

位 29 IWWDGRSTF: 独立看门狗复位标志

在独立看门狗复位发生时由硬件置 1。

由软件通过写 RMVF 位清除该位。

0: 无看门狗复位发生

1: 发生看门狗复位

位 28 SFTRSTF: 软件复位标志

在软件复位发生时由硬件置 1。

由软件通过写 RMVF 位清除。

0：无软件复位发生

1：发生软件复位

位 27 PORRSTF: 上电 / 掉电复位标志

在 NRST 引脚复位发生时由硬件置 1。

由软件通过写 RMVF 位清除。

0：无 NRST 引脚复位发生

1：发生 NRST 引脚复位

位 26 PINRSTF: NRST 引脚复位标志
在 NRST 引脚复位发生时由硬件置 1。
由软件通过写 RMVF 位清除。
0：无 NRST 引脚复位发生
1：发生 NRST 引脚复位

位 25 OBLRSTF: 选项字节装载器复位标志
在选项字节装载器装载选项字节发生时由硬件置 1。
由软件通过写 RMVF 位清除。
0: 无选项字节装载器复位发生
1: 发生了选项字节装载器复位

位 24 RMVF: 清除复位标志
由软件置 1 来清除复位标志。
0: 无作用
1: 清除复位标志

位 23:2 保留 , 必须保持为复位值。

位 1 LSIRDY: LSI 振荡器就绪
由硬件置 1 或清 0 来指示内部 LSI 振荡器是否就绪。在 LSION 清零后，3 个 LSI 振荡器的周期后 LSIRDY 被清零。
0: LSI 振荡器未就绪
1: LSI 振荡器就绪

位 0 LSION: LSI 振荡器使能
由软件置 1 或清 0.
0: LSI 振荡器关闭
1: LSI 振荡器开启

7.4.11 AHB 外设复位寄存器 (RCC_AHBRSTR)

偏移地址 : 0x28

复位值 : 0x0000 0000

访问：无等待周期，字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	TSC RST	Res	IOPF RST	Res	IOPD RST	IOPC RST	IOPB RST	IOPA RST	Res
							rw		rw		rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res

位 31:25 保留 , 必须保持为复位值。

位 24 TSCRST: 触摸传感控制器复位
由软件置 1 或清 0.
0: 无作用
1: 复位触摸传感控制器

位 23 保留 , 必须保持为复位值。

位 22 IOPFRST: GPIOF 口复位
由软件置 1 或清 0.
0: 无作用
1: 复位 GPIOF 口

位 21 保留 , 必须保持为复位值。

位 20 IOPDRST: GPIOD 口复位
由软件置 1 或清 0.
0: 无作用
1: 复位 GPIOD 口

位 19 IOPCRST: GPIOC 口复位
由软件置 1 或清 0.
0: 无作用
1: 复位 GPIOC 口

位 18 IOPBRST: GPIOB 口复位
由软件置 1 或清 0.
0: 无作用
1: 复位 GPIOB 口

位 17 IOPARST: GPIOA 口复位
由软件置 1 或清 0.
0: 无作用
1: 复位 GPIOA 口

位 16:0 保留 , 必须保持为复位值。

7.4.12 时钟配置寄存器 2 (RCC_CFGR2)

偏移地址 : 0x2C

复位值 : 0x0000 0000

访问：无等待周期，字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PREDIV[3:0]			
												rw	rw	rw	rw

位 31:4 保留 , 必须保持为复位值。

位 3:0 PREDIV[3:0] PREDIV 分频因子

这些位用于设置或清除 PREDIV 分频因子。这些位仅能在 PLL 关闭些改写。

注：：位 0 与 RCC_CFGR 的位 17SH M , 修改 RCC_CFGR 的位 17 同时改变这里的位 0 (用于兼容其他 STM32 产品)

0000: HSE 作为 PLL 的输入，不分频

0001: HSE 作为 PLL 的输入 2 分频

0010: HSE 作为 PLL 的输入 3 分频

0011: HSE 作为 PLL 的输入 4 分频

0100: HSE 作为 PLL 的输入 5 分频

0101: HSE 作为 PLL 的输入 6 分频

0110: HSE 作为 PLL 的输入 7 分频

0111: HSE 作为 PLL 的输入 8 分频

1000: HSE 作为 PLL 的输入 9 分频

1001: HSE 作为 PLL 的输入 10 分频

1010: HSE 作为 PLL 的输入 11 分频

1011: HSE 作为 PLL 的输入 12 分频

1100: HSE 作为 PLL 的输入 13 分频

1101: HSE 作为 PLL 的输入 14 分频

1110: HSE 作为 PLL 的输入 15 分频

1111: HSE 作为 PLL 的输入 16 分频

7.4.13 时钟配置寄存器 3 (RCC_CFGR3)

Address: 0x30

复位值 : 0x0000 0000

访问：无等待周期，字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	Res	ADC SW	Res	CEC SW	Res	I2C1 SW	Res	Res	USART1SW[1:0]	
							rw		rw		rw			rw	rw

位 31:9 保留 , 必须保持为复位值。

位 8 ADCSW: ADC 时钟源选择

由软件设置和清 0 来选择 ADC 的时钟源。

0: HSI14 时钟被选为 ADC 时钟 (缺省)

1: PLCK 2 分频或 4 分频被选为 ADC 时钟

注：当 HSI14 被选为 ADC 的时钟源，则 HSI14 振荡器不能关闭 (RCC_CR2 中的 HSI14DIS=0)。

位 7 保留 , 必须保持为复位值。

位 6 CECSW: HDMI CEC 时钟源选择

由软件设置和清 0 来选择 CEC 的时钟源。

0: HSI / 244 被选为 CEC 的时钟源 (缺省)

1: LSE 时钟被选为 CEC 的时钟源

位 5 保留 , 必须保持为复位值。

位 4 I2C1SW: I2C1 时钟源选择

由软件设置和清 0 来选择 I2C1 的时钟源。

0: HSI 时钟被选为 I2C1 时钟源 (缺省)

1: 系统时钟 (SYSCLK) 被选为 I2C1 的时钟源

位 3:2 保留 , 必须保持为复位值。

位 1:0 USART1SW[1:0]: USART1 时钟源选择

由软件设置和清 0 来选择 USART1 的时钟源

00: PCLK 被选为 USART1 的时钟源 (缺省)

01: 系统时钟 (SYSCLK) 被选为 USART1 的时钟源

10: LSE 时钟被选为 USART1 的时钟源

11: HSI 时钟被选为 USART1 的时钟源

7.4.14 时钟控制寄存器 2 (RCC_CR2)

偏移地址 : 0x34

复位值 : 0x0000 XX80, X 未定义。

访问：无等待周期，字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSI14CAL[7:0]								HSI14TRIM[4:0]					HSI14 DIS	HSI14 RDY	HSI14 ON
r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	r	rw

位 31:16 保留 , 必须保持为复位值。

位 15:8 HSI14CAL[7:0]: HSI14 时钟校准

启动时这些位会被自动初始化为出厂校准参数

位 7:3 HSI14TRIM[4:0]: HSI14 时钟调整

这些位在 HSI14CAL[7:0] 的基础上，让用户可以输入一个调整数值，根据电压和温度的变化调整内部 HSI14 RC 振荡器的频率。 默认数值为 16，可以把 HSI14 调整到 14MHz ±1%；每步 HSI14CAL 的变化调整约 50kHz(Fhsi14trim)。

位 2 HSI14DIS ADC HSI14 时钟请求禁止

由软件置 1 或清 0.

当置 1 时阻止 ADC 接口打开 HSI14 振荡器。

0: ADC 接口能打开 HSI14 振荡器

1: ADC 接口不能打开 HSI14 振荡器

位 1 HSI14RDY: HSI14 时钟就绪标志

由硬件置 1 来指示内部 HSI14 振荡器已经稳定。在 HSI14ON 位清零后，HSI14RDY 位需要 6 个 HSI14 振荡器周期清零。.

0: HSI14 振荡器未就绪

1: HSI14 振荡器就绪

位 0 HSI14ON: HSI14 时钟使能

由软件置 1 或清 0.

0: HSI14 振荡器关闭

1: HSI14 振荡器开启

7.4.15 RCC 寄存器映象

下表给出 RCC 寄存器映象和复位值。

表 18. RCC 寄存器映象和复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	RCC_CR	Res.	Res.	Res.	Res.	Res.	Res.	PLL RDY	PLL ON	Res.	Res.	Res.	Res.	CSSON	HSEBYP	HSERDY	HSEON	HSICAL[7:0]								HSITRIM[4:0]					Res.	HSIRDY	HSION
	Reset value							0	0					0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0		1	1
0x04	RCC_CFGR	Res.	Res.	Res.	Res.	Res.	Res.	MCO [2:0]		Res.	Res.	PLLMUL[3:0]				PLLXTPRE	PLLSRC	Res.	ADC PRE	Res.	Res.	Res.	PPRE [2:0]			HPRE[3:0]				SWS [1:0]		SW [1:0]	
	Reset value							0	0			0	0	0	0	0	0		0				0	0	0	0	0	0	0	0	0	0	0
0x08	RCC_CIR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSSC	Res.	HSI14 RDYC	PLLRDYC	HSERDYC	HSIRDYC	LSERDYC	LSIRDYC	Res.	Res.	HSI14 RDYIE	PLLRDYIE	HSERDYIE	HSIRDYIE	LSERDYIE	LSIRDYIE	CSSF	Res.	HSI14 RDYF	PLLRDYF	HSERDYF	HSIRDYF	LSERDYF	LSIRDYF
	Reset value									0		0	0	0	0	0	0			0	0	0	0	0	0	0		0	0	0	0	0	0
0x0C	RCC_APB2RSTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBGMCURST	Res.	Res.	Res.	TIM17RST	TIM16RST	TIM15RST	Res.	USART1RST	Res.	SPI1RST	TIM1RST	Res.	ADCRST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSCFGRST
	Reset value										0				0	0	0		0		0	0	0	0									0
0x010	RCC_APB1RSTR	Res.	CECRST	DACRST	PWRRST	Res.	Res.	Res.	Res.	Res.	I2C2RST	I2C1RST				USART2RST	Res.	Res.	SPI2RST	Res.	Res.	WWDGRST	Res.	Res.	TIM14RST	Res.	Res.	Res.	TM6RST	Res.		TIM3RST	TIM2RST
	Reset value		0	0	0						0	0				0			0			0			0				0			0	0
0x14	RCC_AHBENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSCEN	Res.	IOPFEN	Res.	IOPDEN	IOPCEN	IOPBEN	IOPAEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRCEN	Res.	FLITFEN	Res.	SRAMEN	Res.	DMAEN
	Reset value								0		0		0	0	0	0											0		1		1		0
0x18	RCC_APB2ENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBGMCUEN	Res.	Res.	Res.	TIM17 EN	TIM16 EN	TIM15 EN	Res.	USART1EN	Res.	SPI1EN	TIM1EN	Res.	ADC1N	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSCFGEN
	Reset value										0				0	0	0		0		0	0		0									0
0x1C	RCC_APB1ENR	Res.	CECEN	DACEN	PWREN	Res.	Res.	Res.	Res.	Res.	I2C2EN	I2C1EN	Res.	Res.	Res.	USART2EN	Res.	Res.	SPI2EN	Res.	Res.	WWDGEN	Res.	Res.	TIM14EN	Res.	Res.	Res.	TIM6EN	Res.	Res.	TIM3EN	TIM2EN
	Reset value		0	0	0						0	0				0			0			0			0				0			0	0
0x20	RCC_BDCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BDRST	RTCEN	Res.	Res.	Res.	Res.	Res.	RTC SEL [1:0]		Res.	Res.	Res.	LSE DRV [1:0]		LSEBYP	LSERDY	LSEON
	Reset value																0	0						0	0				0	0	0	0	0
0x24	RCC_CSR	LPWRSTF	WWDGRSTF	IWWDGRSTF	SFTRSTF	PORRSTF	PINRSTF	OBLRSTF	RMVF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSIRDY	LSION
	Reset value	0	0	0	0	1	1	0	0																							0	0
0x28	RCC_AHBRSTR	Res	Res	Res	Res	Res	Res	Res	TSC RST	Res	IOPF RST	Res	IOPD RST	IOPC RST	IOPB RST	IOPA RST	Res	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value								0		0		0	0	0	0																	
0x2C	RCC_CFGR2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREDIV[3:0]			
	Reset value																													0	0	0	0

有关寄存器起始地址参见 2.2.2 节。

8 通用 I/O (GPIO)

8.1 GPIO 介绍

每个通用 I/O 口都有 4 个 32 位配置寄存器 (GPIOx_MODER, GPIOx_OTYPER, GPIOx_OSPEEDR 和 GPIOx_PUPDR)，2 个 32 位数据寄存器 (GPIOx_IDR and GPIOx_ODR) 和 1 个 32 位置位 / 复位寄存器 (GPIOx_BSRR)。口 A 和口 B 还含有 1 个 32 位锁定寄存器 (GPIOx_LCKR) 和 2 个 32 位替代功能寄存器 (GPIOx_AFRH and GPIOx_AFRL).

8.2 GPIO 主要特性

- 输出状态：带有上拉或下拉的推挽输出或开漏输出
- 从数据寄存器 (GPIOx_ODR) 或外设 (复用功能输出) 输出数据
- 可选的每个 I/O 口的速度
- 输入状态：浮空、上拉 / 下拉、模拟输入
- 从数据寄存器 (GPIOIDR) 或外设输入数据 (复用功能输出)
- 位置位 / 复位寄存器 (GPIOx_RR) 为对 GPIOx_ODR 寄存器提供位访问能力
- 口 A 或口的锁定机制 (GPIOx_LCKR) 配置
- 模拟功能
- 可选的口 A 和口 B 复用功能
- 每两个时钟周期快速切换口线值能力
- 允许 GPIO 口和外设引脚的高灵活性复用

8.3 GPIO 功能描述

根据数据手册中列出的每个 I/O 端口的特定硬件特征， GPIO 端口的每个位可以由软件分别配置成多种模式：

- 浮空输入
- 上拉输入
- 下拉输入
- 模拟输入
- 具有上拉或下拉能力的开漏输出
- 具有上拉或下拉能力的推挽输出
- 复用功能且具有上拉或下拉能力的推挽输出
- 复用功能且具有上拉或下拉能力的开漏输出

每个 I/O 端口位可以自由编程，然而 I/0 端口寄存器可按 32 位字，半字或字节访问。GPIOx_BSRR 寄存器允许对任何 GPIO 寄存器进行位读 / 改写访问。这种情况下，在读和更改访问之间产生 IRQ 时也不会发生危险。

图 12 和图 13 分别给出了一个标准 IO 端口位和 5V 容忍的 IO 端口位的基本结构
表 20 给出了端口位的可能配置

图 12. 标准 I/O 端口位的基本结构

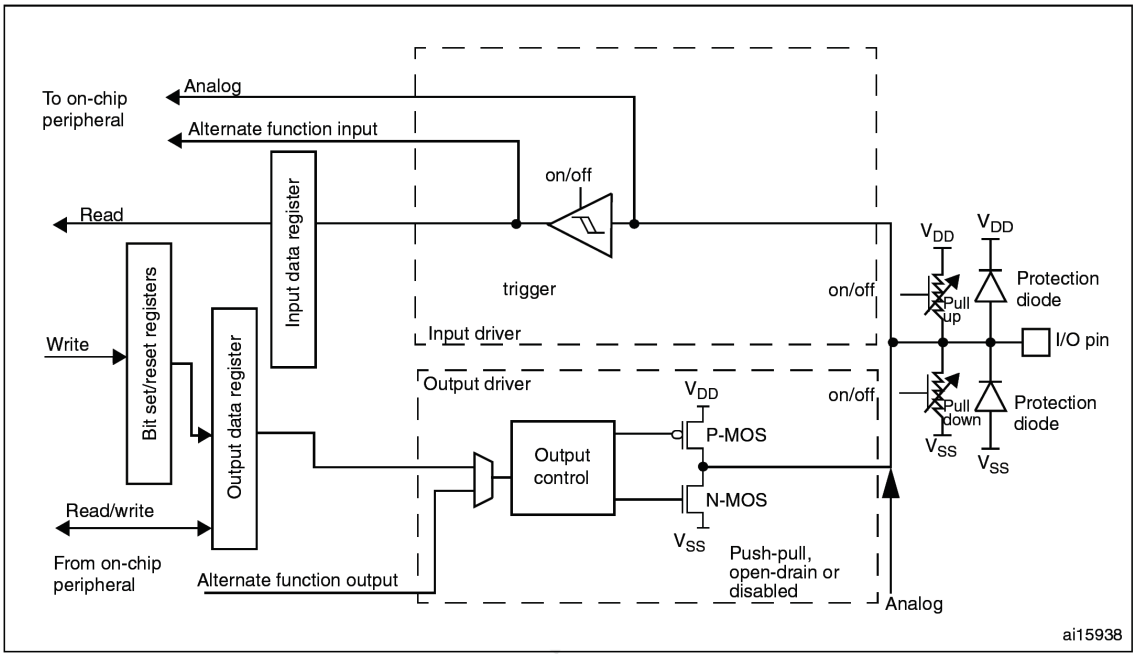
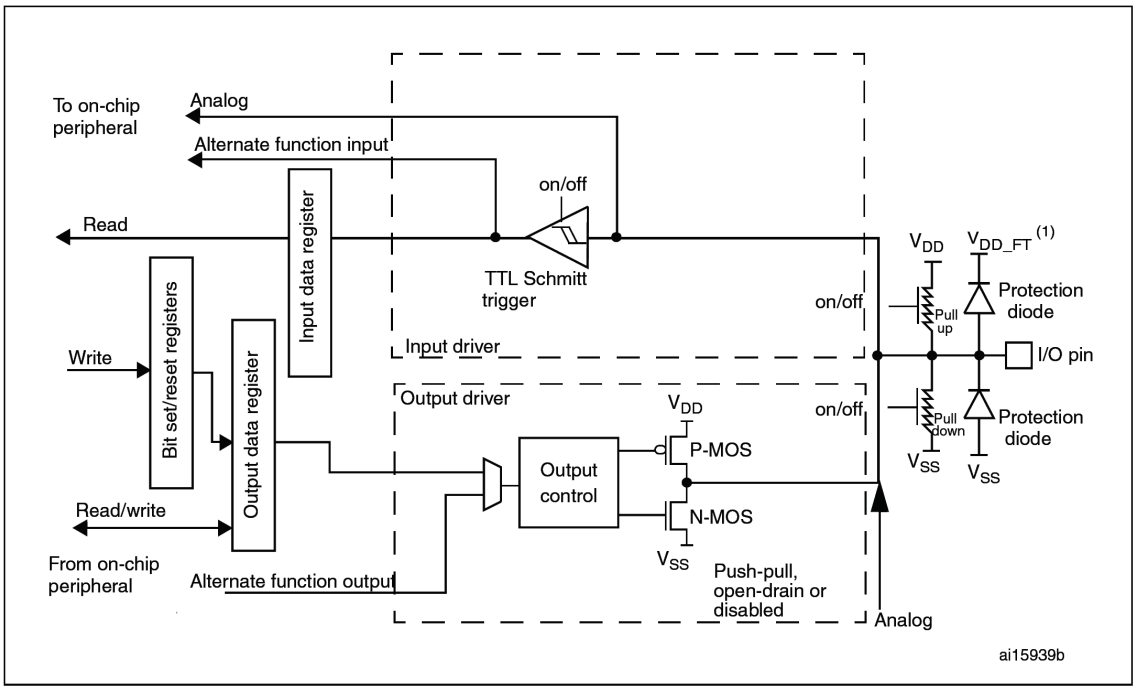


图 13. 5V 容忍 I/O 端口位的基本结构



1. VDD_FT 对 5 伏容忍 I/O 脚是特殊的，它与 VDD 不同 .

表 19. 端口位配置表 (1)

MODER(i) [1:0]	OTYPER(i)	OSPEEDR(i) [B:A]		PUPDR(i) [1:0]		I/O 配置	
01	0	SPEED [B:A]		0	0	GP 输出	PP
	0			0	1	GP 输出	PP + PU
	0			1	0	GP 输出	PP + PD
	0			1	1	保留	
	1			0	0	GP 输出	OD
	1			0	1	GP 输出	OD + PU
	1			1	0	GP 输出	OD + PD
	1			1	1	保留 (GP 输出 OD)	
10	0	SPEED [B:A]		0	0	AF	PP
	0			0	1	AF	PP + PU
	0			1	0	AF	PP + PD
	0			1	1	保留	
	1			0	0	AF	OD
	1			0	1	AF	OD + PU
	1			1	0	AF	OD + PD
	1			1	1	保留	
00	x	x	x	0	0	输入	浮空
	x	x	x	0	1	输入	PU
	x	x	x	1	0	输入	PD
	x	x	x	1	1	保留 (浮空输入)	
11	x	x	x	0	0	输入 / 输出	模拟
	x	x	x	0	1	保留	
	x	x	x	1	0		
	x	x	x	1	1		

1. GP = 通用 , PP = 推挽输出 , PU = 上拉 , PD = 下拉 , OD = 开漏 , AF = 复用功能 .

8.3.1 通用 I/O (GPIO)

复位期间和刚复位后，复用功能未开启且所有的 I/O 端口被配置为浮空输入模式。

复位后，调试引脚被置为复用功能的上拉 / 下拉模式：

- PA14: SWCLK 置于下拉模式
- PA13: SWDAT 置于上拉模式

当作为输出配置时，写到输出数据寄存器 (GPIOx_ODR) 的值输出到相应的引脚上。可以以推挽模式或开漏模式 (仅低电平被驱动，高电平表现为高阻) 使用输出驱动器。

输入数据寄存器 (GPIOx_IDR) 在每个 AHB 时钟周期捕捉 I/O 引脚上的数据。

所有 GPIO 引脚都有一个内部弱上拉和弱下拉电阻，它们被激活或断开有赖于 GPIOx_PUPDR 寄存器的值。

8.3.2 I/O 引脚的复用功能和重映射

器件 I/O 口线通过多路复用器连接到内嵌的外设 / 模块。微观上，同一时刻仅允许外设的复用功能一个引脚连接到一个 I/O 口线上。因此，同一口线上不能有冲突的外设引脚分配。

每个 I/O 引脚有一个多达 16 个复用功能输入 (从 AF0 到 AF15) 的多路复用器，其可通过配置 GPIOx_AFRL 寄存器 (从引脚 0 到引脚 7) 和 GPIOx_AFRH 寄存器 (从引脚 8 到引脚 15) 来实现。

- 复位后，所有的 I/O 口都连接到复用功能 0
- 有关每个引脚的具体复用功能在器件数据手册有详尽描述

除了这种灵活的 I/O 复用结构，每个外设还有复用功能映射到不同的 I/O 引脚上，这种方法用于在小封装器件上优化更多的可用外设。

为了使用一个给定的 I/O 口配置，你必须按如下的原则执行：

- 调试功能： 每个器件复位后，这些引脚立即配置为复用功能用来支持调用。
- GPIO: 在 GPIOx_MODER 寄存器中配置所需的 I/O 口为输出，输入或模拟输入。
- 外设的复用功能：
  - 连接 I/O 到所需的 AFx ，AFx 定义在 GPIOx_AFRL 或 GPIOx_AFRH 寄存器中
  - 通过对 GPIOx_OTYPER, GPIOx_PUPDR 和 GPIOx_OSPEEDER 寄存器来配置相应引脚的上拉 / 下拉和输出速度
  - 在 GPIOx_MODER 寄存器中配置所需的 I/O 口线为复用功能
- 附加功能：
  - 对于 ADC 和 DAC, 在 GPIOx_MODER 寄存器中配置所需的 I/O 口线为模拟方式并在 ADC 或 DAC 寄存器中配置所需的功能。
  - 对于附加功能如 RTC、WKUPx 和振荡器，在关联的 RTC、PWR 和 RCC 寄存器配置相应所需的功能。

有关详细的 I/O 口线复用功能映射，请参考器件数据手册中的 “复用功能映射” 表。

8.3.3 I/O 端口控制寄存器

每个 GPIO 口都有 4 个 32 位的控制寄存器 (GPIOx_MODER, GPIOx_OTYPER, GPIOx_OSPEEDR, GPIOx_PUPDR) 用来配置多达 16 I/O 口线。GPIOx_MODER 寄存器用来选择 I/O 模式 (如输入，输出、复用或模拟)。GPIOx_OTYPER 和 GPIOx_OSPEEDR 寄存器用来选择输出类型 (如推挽或开漏) 和速度。GPIOx_PUPDR 寄存器用来选择上拉 / 下拉方式。

8.3.4 I/O 端口数据寄存器

每个 GPIO 口有两个 16 位数据寄存器：输入和输出数据寄存器 (GPIOx_IDR 和 GPIOx_ODR). GPIOx_ODR 用于存储输出数据 , 其可进行读 / 写访问。从 I/O 口线的输入数据存放在 (GPIOx_IDR) 寄存器中，该寄存器为只读寄存器。

有关数据寄存器的详细说明参见 8.4.5 章节：GPIO 口输入数据寄存器 (GPIOx_IDR) (x = A..D,F) 和 8.4.6 章节： GPIO 口输出数据寄存器 (GPIOx_ODR) (x = A..D,F)。

8.3.5 I/O 数据位处理

位置位复位寄存器 (GPIOx_BSRR) 是一个 32 位寄存器，其允许应用对输出数据寄存器 (GPIOx_ODR) 的每个位进行置位和复位操作。位置位和复位寄存器的有效数据宽度是 GPIOx_ODR 有效数据宽度的两倍。

对于 GPIOx_ODR 中的每位，在 GPIOx_BSRR 中有两位与之对应： BS(i) 和 BR(i)。当对位 BS(i) 写 1 时则设置相应的 ODR(i) 位。当对 BR(i) 写 1 时，则复位相应的 ODR(i) 位。

对 GPIOx_BSRR 中的任意位写 0 都不会影响 GPIOx_ODR 寄存器的值。若对 GPIOx_BSRR 的 BS(i) 和 BR(i) 同时置 1 ，那么其置位操作具有优先权 (即对相应位做置位操作)。

用 GPIOx_BSRR 寄存器来改变 GPIOx_ODR 的相应位， GPIOx_ODR 位也可直接从这个寄存器进行访问。GPIOx_BSRR 寄存器提供对 GPIOx_ODR 寄存器原子位操作处理机制。

GPIOx_ODR 用 GPIOx_BSRR 置位或复位的访问机制不需要软件去关闭中断来访问 GPIOx_ODR：在一个 AHB 写访问周期改变 1 位或多位数据是可能的。

8.3.6 GPIO 锁定机制

用一个特定对 GPIOx_LCKR 寄存器的写序列来冻结端口 A 和端口 B 的控制寄存器是可行的。冻结的寄存器有：GPIOx_MODER, GPIOx_OTYPER, GPIOx_OSPEEDR, GPIOx_PUPDR, GPIOx_AFRL 和 GPIOx_AFRH.

为了写 GPIOx_LCKR 寄存器，须发出一个特定的写 / 读序列。当正确的锁定序列作用于这个寄存器的位 16 时， LCKR[15:0] 的值用来锁定 I/O 口的配置 (在写序列期间要保持 LCKR[15:0] 值不变)。

当锁定序列已经作用于一个端口位，该端口位的值再也不能改变直到下一次复位。每个 GPIOx_LCKR 位冻结控制寄存器中 (GPIOx_MODER, GPIOx_OTYPER, GPIOx_OSPEEDR, GPIOx_PUPDR, GPIOx_AFRL 和 GPIOx_AFRH) 的相应位。

锁定序列 (参见 8.4.8 章节：GPIO 端口配置锁定寄存器 (GPIOx_LCKR) (x = A..B)) 只能用字 (32-位长) 访问 GPIOx_LCKR 寄存器，基于 GPIOx_LCKR 位 16 必须与 [15:0] 位同时设置的事实。

有关详情请参考 8.4.8 章节： GPIO 端口配置锁定寄存器 (GPIOx_LCKR) (x = A..B) 的 LCKR 寄存器描述

8.3.7 I/O 复用功能输入 / 输出

选择每个端口线的有效复用功能之一是由两个寄存器来决定的。你可根据你应用的需求用这两寄存器连接复用功能模块到其他引脚。

这就表明每个 GPIO 口线可能做为多个外设的口线，用 GPIOx_AFRL 和 GPIOx_AFRH 复用功能寄存器来配置这些外设口线。

欲知哪个功能被复用到哪些 GPIO 引脚 , 参考相应器件的数据手册。

8.3.8 外部中断 / 唤醒线

所有端口都有外部中断能力。为了用做外部中断口线，端口线必须配置为输入模式， 有关细节参考 11.2 章节：外部中断和事件控制器 (EXTI) 和 11.2.3 章节：唤醒事件管理内容。

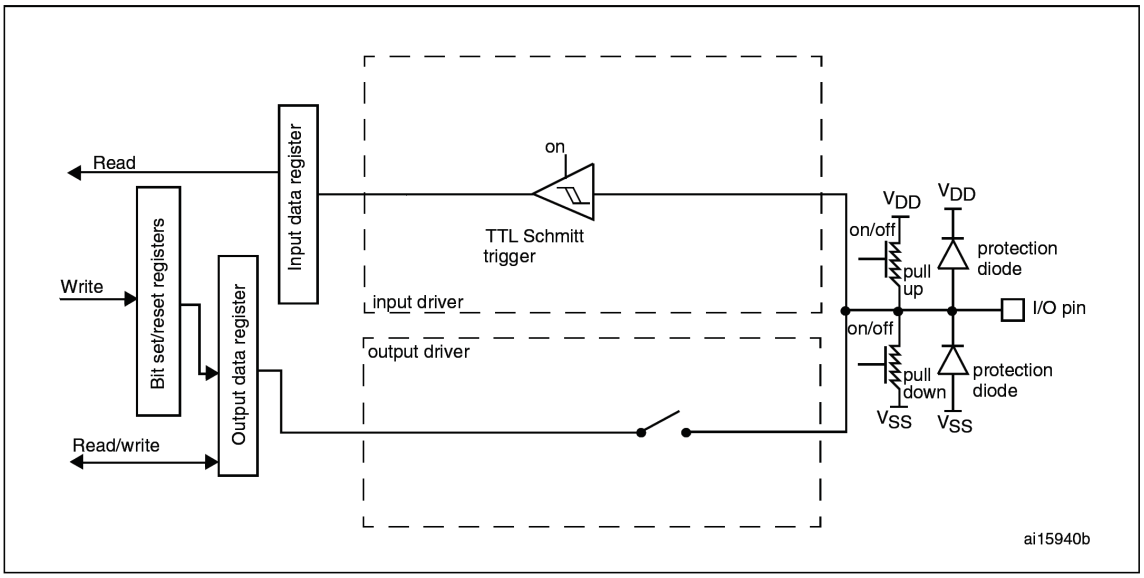
8.3.9 输入配置

当 I/O 口编程配置为输入时：

- 该输出缓冲区禁用
- 施密特触发器输入激活
- 由 GPIOx_PUPDR 寄存器的值来激活上拉和下拉电阻
- 在每个 AHB 时钟周期 I/O 引脚上的数据被采样进入输入数据寄存器。
- 用对输入数据寄存器的读访问来获取 I/O 口状态

图 14 给出 I/O 端口位的输入配置 .

图 14. 浮空输入 / 上拉 / 下拉配置



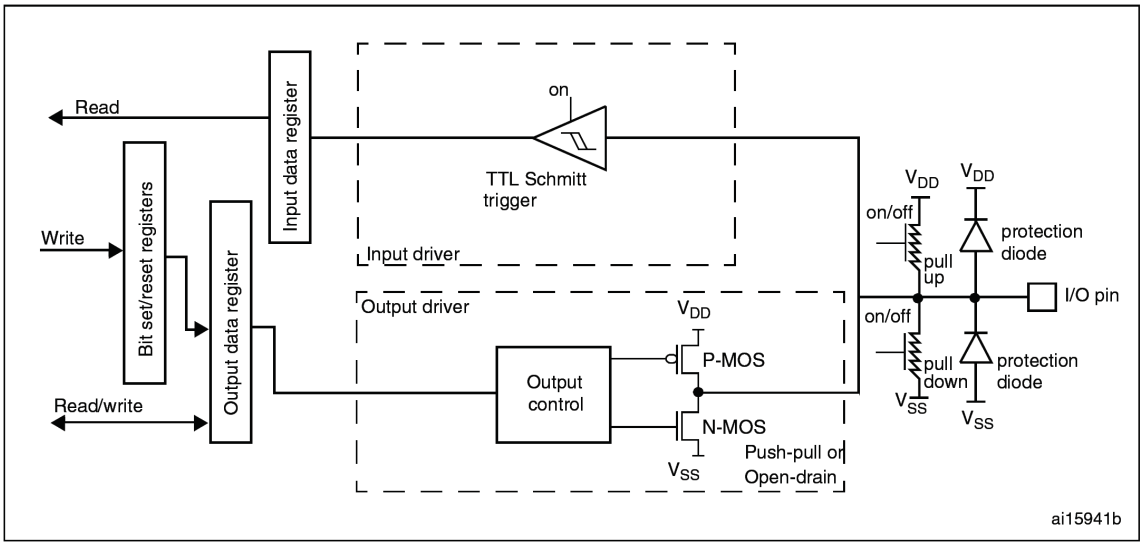
8.3.10 输出配置

当 I/O 口配置为输出时：

- 输出缓冲开启：
  - 开漏模式：输出寄存器上的’0’激活 N-MOS，而输出寄存器上的’1’将端口置于高阻状态 (P-MOS 从不被激活)。
  - 推挽模式：输出寄存器上的’0’激活 N-MOS，而输出寄存器上的’1’将激活 P-MOS。
- 施密特触发输入被激活
- 弱上拉和弱下拉电阻是否激活取决于 GPIOx_PUPDR 寄存器的值
- 在每个 AHB 时钟周期 I/O 引脚上的数据被采样进入输入数据寄存器
- 用对输入数据寄存器的读访问来获取 I/O 口状态
- 用对输出寄存器的读访问来获取最后写进该寄存器的值

图 15 给出了 I/O 端口位的输出配置。

图 15. 输出配置



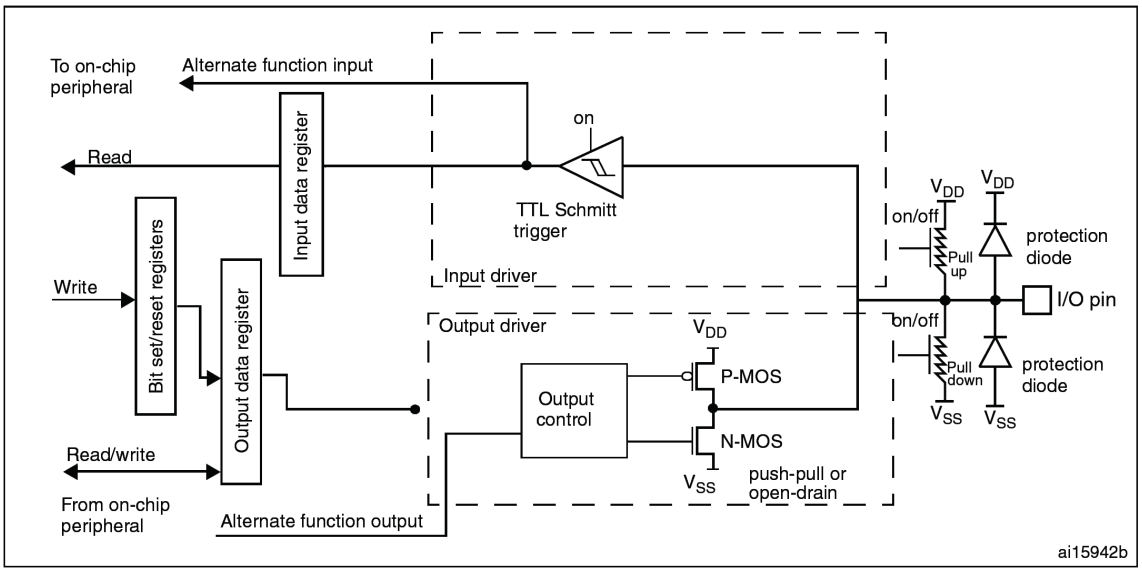
8.3.11 复用功能配置

当 I/O 端口被配置为复用功能时：

- 在开漏或推挽模式下输出缓冲器可被配置
- 外设的信号驱动输出缓冲器
- 施密特触发输入被激活
- 弱上拉和弱下拉电阻是否激活取决于 GPIOx_PUPDR 寄存器的值
- 在每个 AHB 时钟周期 I/O 引脚上的数据被采样进入输入数据寄存器
- 用对输入数据寄存器的读访问来获取 I/O 口状态

图 16 给出了 I/O 端口位的复用功能配置。

图 16. 复用功能配置



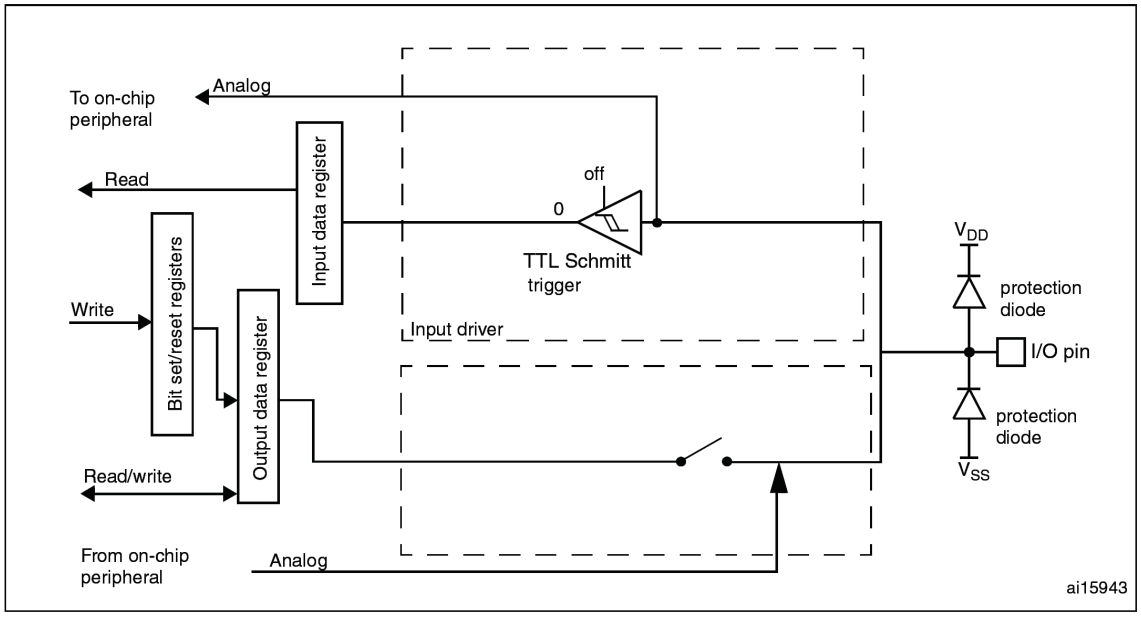
8.3.12 模拟配置

当 I/O 端口编程为模拟配置时：

- 输出缓冲器关闭
- 禁止施密特触发输入，实现了每个模拟 I/O 引脚上的零消耗。施密特触发输出值被强置为’0’
- 弱上拉和下拉电阻被禁止
- 读取输入数据寄存器时数值为 0

图 17 给出了 I/O 端口位的高阻抗模拟输入配置。

图 17. 高阻抗模拟配置



8.3.13 HSE 或 LSE 引脚用作 GPIO

当 HSE 或 LSE 振荡器关断时 (复位后的缺省状态) , 相关振荡器引脚可以用做普通的 GPIO 口。当 HSE 或 LSE 振荡器开启 (在 RCC_CSR 寄存器设置 HSEON 或 LSEON 位来开启) 振荡器控制其相关引脚且相关引脚的 GPIO 配置无效。当振荡器配置为用户外部时钟方式，仅使用 OSC_IN 或 OSC32_IN 引脚做为时钟办理， OSC_OUT 或 OSC32_OUT 引脚可仍然配置为正常的 GPIO 引脚。

8.3.14 备份域供电下 GPIO 引脚的使用

当 VCORE 域断电 (当器件进入待机模式) 时，PC13/PC14/PC15 GPIO 功能失去。在这种情况下，若这些 GPIO 配置为不被 RTC 配置旁路，这些引脚被设为模拟输入模式。有关 I/O 由 RTC 控制的详情，参见 22.3 章节： RTC 功能描述

8.4 GPIO 寄存器

该章节给出了 GPIO 寄存器的详细说明。

有关 GPIO 寄存器位，寄存器地址偏移和复位值，参见表 20。

这些外设寄存器可以字、半字或字节的方式写。

8.4.1 GPIO 端口模式寄存器 (GPIOx_MODER) (x = A..D,F)

偏移地址： 0x00

复位值：

- 0x2800 0000 端口 A
- 0x0000 0000 其他口

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 2y+1:2y MODERy[1:0]: 端口 x 配置位 (y = 0..15)

这些位可由软件写来配置 I/O 口模式。

00: 输入模式 (复位状态)

01: 通用输出模式

10: 复用功能模式

11: 模拟模式

8.4.2 GPIO 端口输出类型寄存器 (GPIOx_OTYPER) (x = A..D,F)

偏移地址：0x04

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持为复位值。

位 15:0 OTy[1:0]: 端口 x 的配置位 (y = 0..15)

这些位可由软件写来配置 I/O 口的输出类型。

0: 推挽输出 (复位状态)

1: 开漏输出

8.4.3 GPIO 口输出速度寄存器 (GPIOx_OSPEEDR) (x = A..D,F)

偏移地址：0x08

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSPEEDR15[1:0]		OSPEEDR14[1:0]		OSPEEDR13[1:0]		OSPEEDR12[1:0]		OSPEEDR11[1:0]		OSPEEDR10[1:0]		OSPEEDR9[1:0]		OSPEEDR8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSPEEDR7[1:0]		OSPEEDR6[1:0]		OSPEEDR5[1:0]		OSPEEDR4[1:0]		OSPEEDR3[1:0]		OSPEEDR2[1:0]		OSPEEDR1[1:0]		OSPEEDR0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 2y+1:2y OSPEEDRy[1:0]: 端口 x 配置位 (y = 0..15)

这些位要由软件写来配置 I/O 口的速度。

x0: 低速

01: 中速

11: 高速

注： 参考数据手册有关在各种速度下的频率规范及供电和负载条件信息。

8.4.4 GPIO 口上拉 / 下拉寄存器 (GPIOx_PUPDR) (x = A..D,F)

偏移地址：0x0C

复位值：

- 0x2400 0000 端口 A
- 0x0000 0000 其它端口

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUPDR15[1:0]		PUPDR14[1:0]		PUPDR13[1:0]		PUPDR12[1:0]		PUPDR11[1:0]		PUPDR10[1:0]		PUPDR9[1:0]		PUPDR8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUPDR7[1:0]		PUPDR6[1:0]		PUPDR5[1:0]		PUPDR4[1:0]		PUPDR3[1:0]		PUPDR2[1:0]		PUPDR1[1:0]		PUPDR0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 2y+1:2y PUPDRy[1:0]: 端口 x 配置位 (y = 0..15)

这些位由软件写来配置 I/O 口的上拉或下拉。

00: 无上拉和下拉

01: 上拉

10: 下拉

11: 保留

8.4.5 GPIO 端口输入数据寄存器 (GPIOx_IDR) (x = A..D,F)

偏移地址：0x10

复位值： 0x0000 XXXX (X 表明不定)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持为复位值。

位 15:0 IDR[15:0]: 端口输入数据

这些位只读。它们包含相应 I/O 口的输入值。

8.4.6 GPIO 端口输出数据寄存器 (GPIOx_ODR) (x = A..D,F)

偏移地址：0x14

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持为复位值。

位 15:0 ODR[15:0]: 端口输出数据

这些位可由软件读写。

注： 对于原子位的设置 / 清除，可单独对 GPIOx_BSRR(x = A..D,F) 寄存器操作来实现。

8.4.7 GPIO 端口置位 / 复位寄存器 (GPIOx_BSRR) (x = A..D,F)

偏移地址：0x18

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:16 BRy: 端口 x 复位位 y(y = 0..15)

这些位只写。读这些位时返回 0x0000 数值。

0: 对相应的 ODRx 位无影响

1: 复位相应的 ODRx 位

注： 若 BSx 和 BRx 同时设置，BSx 有优先权。

位 15:0 BSy: 端口 x 设置位 y (y= 0..15)

这些位只写。读这些位时返回 0x0000 数值。

0: 对相应的 ODRx 位无影响

1: 置位相应的 ODRx 位

8.4.8 GPIO 端口配置锁定寄存器 (GPIOx_LCKR) (x = A..B)

当一个正确的写序列应用于位 16(LCKK) 时，这个寄存器是用来锁定端口位的配置。位 [15:0] 的值用于锁定相 GPIO 位的配置。在写序列期间， LCKR[15:0] 的值不会改变。当锁序列已经应用到端口位，端口位的值再不会被改变直到下一次复位的到来。

注：　一个特定的写序列用于写 GPIOx_LCKR 寄存器。在这个锁序列期间，仅能字 (32 位) 访问该寄存器。

每个锁定位冻结一个指定的配置寄存器 (控制和复用功能寄存器).

偏移地址：0x1C
复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LCKK
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持为复位值。

位 16　LCKK: 锁定键
该位可随时读取。它仅能由锁键写序列来改写。
0: 端口配置锁定键不激活
1: 端口配置锁定键激活。GPIOx_LCKR 寄存器锁定直到一个 MCU 复位产生。

锁定键写序列：
写 LCKR[16] = ‘1’ + LCKR[15:0]
写 LCKR[16] = ‘0’ + LCKR[15:0]
定 LCKR[16] = ‘1’ + LCKR[15:0]
读 LCKR
读 LCKR[16] = ‘1’ (这个读操作可选，但其为确认锁定是否激活)
注：在锁写键写序列期间， LCK[15:0] 值必须不变。
在锁定写序中出现任何错误都会中止锁定操作。
在第次 1 锁定序列在端品的任意位的操作后，任何读 LCKK 位时将返回‘1’直到下次 CPU r 复位。

位 15:0　LCKy: 端口 x 锁写位 y (y= 0..15)
这些位可读 / 写，但仅 LCKK 为‘0’时写。
0: 端口配置未锁定
1: 端口配置锁定

8.4.9 GPIO 复用功能低位寄存器 (GPIOx_AFRL) (x = A..B)

偏移地址：0x20

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFRL7[3:0]				AFRL6[3:0]				AFRL5[3:0]				AFRL4[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFRL3[3:0]				AFRL2[3:0]				AFRL1[3:0]				AFRL0[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 AFRLy: 端口 X 引脚 y 的复用功能选择 (y = 0..7)

这些位可由软件写来配置复用功能 I/O 口。

AFRLy 选择 :

0000: AF0
0001: AF1
0010: AF2
0011: AF3
0100: AF4 (仅限端口 A)
0101: AF5 (仅限端口 A)
0110: AF6 (仅限端口 A)
0111: AF7 (仅限端口 A)
1000: 保留
1001: 保留
1010: 保留
1011: 保留
1100: 保留
1101: 保留
1110: 保留
1111: 保留

8.4.10 GPIO 复用功能高位寄存器 (GPIOx_AFRH) (x = A..B)

偏移地址：0x24

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFRH15[3:0]				AFRH14[3:0]				AFRH13[3:0]				AFRH12[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFRH11[3:0]				AFRH10[3:0]				AFRH9[3:0]				AFRH8[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 AFRHy: 端口 x 引脚 y 的复用功能选择 (y = 8..15)

这些位可由软件写来配置复用功能 I/O 口。

AFRHy 选择 :

0000: AF0
0001: AF1
0010: AF2
0011: AF3
0100: AF4(仅限端口 A)
0101: AF5(仅限端口 A)
0110: AF6(仅限端口 A)
0111: AF7(仅限端口 A)
1000: 保留
1001: 保留
1010: 保留
1011: 保留
1100: 保留
1101: 保留
1110: 保留
1111: 保留

8.4.11 端口位复位寄存器 (GPIOx_BRR) (x=A..G)

偏移地址：0x28

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:16 保留

位 15:0 BRy: 端口 x 复位位 y (y= 0 .. 15)

这些位为只写位。读这些位的返回值为 0x0000

0: 对相应的 ODRx 位无影响

1: 复位相应的 ODRx 位

8.4.12 GPIO 寄存器映像

下表给出 GPIO 寄存器映像和复位值。

表 20. GPIO 寄存器映像及复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	GPIOA_MODER	MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]		MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
	Reset value	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00	GPIOx_MODER (where x = B..D,F)	MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]		MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	GPIOx_OTYPER (where x = A..D,F)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	GPIOx_OSPEEDER (where x = A..D,F)	OSPEEDR15[1:0]		OSPEEDR14[1:0]		OSPEEDR13[1:0]		OSPEEDR12[1:0]		OSPEEDR11[1:0]		OSPEEDR10[1:0]		OSPEEDR9[1:0]		OSPEEDR8[1:0]		OSPEEDR7[1:0]		OSPEEDR6[1:0]		OSPEEDR5[1:0]		OSPEEDR4[1:0]		OSPEEDR3[1:0]		OSPEEDR2[1:0]		OSPEEDR1[1:0]		OSPEEDR0[1:0]	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	GPIOA_PUPDR	PUPDR15[1:0]		PUPDR14[1:0]		PUPDR13[1:0]		PUPDR12[1:0]		PUPDR11[1:0]		PUPDR10[1:0]		PUPDR9[1:0]		PUPDR8[1:0]		PUPDR7[1:0]		PUPDR6[1:0]		PUPDR5[1:0]		PUPDR4[1:0]		PUPDR3[1:0]		PUPDR2[1:0]		PUPDR1[1:0]		PUPDR0[1:0]	
	Reset value	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 20. GPIO 寄存器映像及复位值 (续)

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0C	GPIOx_PUPDR (where x = B..D,F)	PUPDR15[1:0]		PUPDR14[1:0]		PUPDR13[1:0]		PUPDR12[1:0]		PUPDR11[1:0]		PUPDR10[1:0]		PUPDR9[1:0]		PUPDR8[1:0]		PUPDR7[1:0]		PUPDR6[1:0]		PUPDR5[1:0]		PUPDR4[1:0]		PUPDR3[1:0]		PUPDR2[1:0]		PUPDR1[1:0]		PUPDR0[1:0]	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	GPIOx_IDR (where x = A..D,F)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
	Reset value																	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x14	GPIOx_ODR (where x = A..D,F)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	GPIOx_BSRR (where x = A..D,F)	BRr5	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0	BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	GPIOx_LCKR (where x = A..B)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LCKK	LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	GPIOx_AFRL (where x = A..B)	AFRL7[3:0]				AFRL6[3:0]				AFRL5[3:0]				AFRL4[3:0]				AFRL3[3:0]				AFRL2[3:0]				AFRL1[3:0]				AFRL0[3:0]	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	GPIOx_AFRH (where x = A..B)	AFRH15[3:0]				AFRH14[3:0]				AFRH13[3:0]				AFRH12[3:0]				AFRH11[3:0]				AFRH10[3:0]				AFRH9[3:0]				AFRH8[3:0]	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	GPIOx_BRR (where x = A..D,F)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

有关寄存器起始地址参见 2.2.2 节。

9 系统配置控制器 (SYSCFG)

该器件具有一组配置寄存器。系统配置控制器的主要用途如下：

- 在部分 IO 口上启用或禁用 I2C 超快模式 (Fast Mode Plus)。
- 重映射部分从 TIM16 和 TIM17，USART1 和 ADC 的 DMA 触发源到其它不同的 DMA 通道上。
- 重映射存储器到代码起始区域。
- 管理连接到 GPIO 口的外部中断。
- 管理系统的可靠性特性。

9.1 SYSCFG 寄存器

9.1.1 SYSCFG 配置寄存器 1 (SYSCFG_CFGR1)

该寄存器专门用于配置内存映射。
两位用来配置访问地址为 0x0000 0000 的存储类型。
这些位可由软件设置来用于选择存储的物理映射。
复位后，这些位通过 BOOT0 和 BOOT1 的引脚配置值来选择相应值。

偏移地址 : 0x00
复位值 : 0x0000 000X (X 的值由 BOOT0 和 BOOT1 存储选择引脚决定)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	I2C_PB9_FM+	I2C_PB8_FM+	I2C_PB7_FM+	I2C_PB6_FM+
												rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	TIM17_DMA_RMP	TIM16_DMA_RMP	USART1_RX_DMA_RMP	USART1_TX_DMA_RMP	ADC_DMA_RMP	Res.	Res.	Res.	Res.	Res.	Res.	MEM_MODE	
			rw	rw	rw	rw	rw							rw	rw

位 31:20 保留 , 必须保持为复位值。

位 19:16 I2C_PBx_FM+: 超快模式 (FM+) 驱动能力激活位
由软件设置和清 0 这些位。每位分别为 PB6, PB7, PB8, 和 PB9 口线开启 I2C 超快模式 (FM+)。
0: PBx 引脚设置为标准模式
1: PBx 引脚配置为 I2C 超快模式 (FM+), 且 I2C 速度控制被旁路 (被忽略)。

位 15:13 保留 , 必须保持为复位值。

位 12 TIM17_DMA_RMP: TIM17 DMA 请求重映射位
由软件设置和清除该位。它控制着 TIM17 DMA 通道请求的重映射。
0: 无重映射 (TIM17_CH1 和 TIM17_UP DMA 请求映射在 DMA 通道 1 上)
1: 重映射 (TIM17_CH1 和 TIM17_UP DMA 请求映射在 DMA 能通道 2 上)

位 11 TIM16_DMA_RMP: TIM16 DMA 请求重映射位
由软件设置和清除该位。它控制着 TIM16 DMA 通道请求的重映射。
0: 无重映射 (TIM16_CH1 和 TIM16_UP DMA 请求映射在 DMA 通道 3 上)
1: 重映射 (TIM16_CH1 和 TIM16_UP DMA 请求映射在 DMA 能通道 4 上)

位 10 USART1_RX_DMA_RMP: USART1_RX DMA 请求重映射位
由软件设置和清除该位。它控制着 USART1_ RX DMA 通道请求的重映射
0: 无重映射 (USART1_ RX 请求映射在 DMA 通道 3 上)
1: 重映射 (USART1_ RX 请求映射在 DMA 通道 5 上)

位 9 USART1_TX_DMA_RMP: USART1_TX DMA 请求重映射位
由软件设置和清除该位。它控制着 USART1_TX DMA 通道请求的重映射
0: 无重映射 (USART1_TX 请求映射在 DMA 通道 2 上)
1: 重映射 (USART1_TX 请求映射在 DMA 通道 4 上)

位 8 ADC_DMA_RMP: ADC DMA 请求重映射位
由软件设置和清除该位。它控制着 ADC DMA 通道请求的重映射
0: 无重映射 (ADC DMA 请求映射在 DMA 通道 1 上)
1: 重映射 (ADC DMA 请求映射在 DMA 通道 2 上)

位 7:2 保留 , 必须保持为复位值。

位 1:0 MEM_MODE: 存储映射选择位
由软件设置和清除这些位。它控制存储器内部映射到地址 0x0000 0000。当复位后这些位值由 BOOT0 和 BOOT1 的引脚配置值决定。
x0: 主闪存存储器映射到 0x0000 0000
01: 系统闪存映射到 0x0000 0000
11: 嵌入式 RAM 映射到 0x0000 0000

9.1.2 SYSCFG 外部中断配置寄存器 1 (SYSCFG_EXTICR1)

偏移地址 : 0x08
复位值 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI3[3:0]				EXTI2[3:0]				EXTI1[3:0]				EXTI0[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留 , 必须保持为复位值。

位 15:0 EXTIx[3:0]: EXTI x 配置位 (x = 0 到 3)

这些位由软件进行改写来选择 EXTIx 的外部中断源。

x000: PA[x] 引脚

x001: PB[x] 引脚

x010: PC[x] 引脚

x011: PD[x] 引脚

x100: 保留

x101: PF[x] 引脚

其它配置：保留

注 : 上述提及的部分 I/O 口线在小封装的器件中可能不可用。

9.1.3 SYSCFG 外部中断配置寄存器 2 (SYSCFG_EXTICR2)

偏移地址 : 0x0C

复位值 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI7[3:0]				EXTI6[3:0]				EXTI5[3:0]				EXTI4[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留 , 必须保持为复位值。

位 15:0 EXTIx[3:0]: EXTI x 配置位 (x = 4 到 7)

这些位由软件进行改写来选择 EXTIx 的外部中断源。

x000: PA[x] 引脚

x001: PB[x] 引脚

x010: PC[x] 引脚

x011: 保留

x100: 保留

x101: PF[x] 引脚

其它配置：保留

注 : 上述提及的部分 I/O 口线在小封装的器件中可能不可用。

9.1.4 SYSCFG 外部中断配置寄存器 3 (SYSCFG_EXTICR3)

偏移地址 : 0x10

复位值 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI11[3:0]				EXTI10[3:0]				EXTI9[3:0]				EXTI8[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留 , 必须保持为复位值。

位 15:0 EXTIx[3:0]: EXTI x 配置位 (x = 8 到 11)

这些位由软件进行改写来选择 EXTIx 的外部中断源。

x000: PA[x] 引脚

x001: PB[x] 引脚

x010: PC[x] 引脚

其他配置：保留

注 : 上述提及的部分 I/O 口线在小封装的器件中可能不可用。

9.1.5 SYSCFG 外部 中断配置寄存器 4 (SYSCFG_EXTICR4)

偏移地址 : 0x14

复位值 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI15[3:0]				EXTI14[3:0]				EXTI13[3:0]				EXTI12[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留 , 必须保持为复位值。

位 15:0 EXTIx[3:0]: EXTI x 配置位 (x = 12 到 15)

这些位由软件进行改写来选择 EXTIx 的外部中断源。

x000: PA[x] 引脚

x001: PB[x] 引脚

x010: PC[x] 引脚

其他配置：保留

注 : 上述提及的部分 I/O 口线在小封装的器件中可能不可用。

9.1.6 SYSCFG 配置寄存器 2 (SYSCFG_CFGR2)

偏移地址 : 0x18

复位值 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	SRAM_PEF	Res.	Res.	Res.	Res.	Res.	PVD_LOCK	SRAM_PARITY_LOCK	LOCUP_LOCK
							rc_w1						rw	rw	rw

位 31:9 保留 , 必须保持为复位值。

位 8 SRAM_PEF: SRAM 校验标志

当 SRAM 校验错被检测到时硬件自动设置该位。对该位写 1 时清该位。

0: 无 SRAM 校验错

1: 检测到 SRAM 校验错

位 7:3 保留 , 必须保持为复位值。

位 2 PVD_LOCK: PVD 锁定使能位

该位由软件设置，由系统复位清除。 可用于使能并锁定 PVD 连接到 TIM1/15/16/17 的刹车 (Break) 输入 , 锁定 PVDE 和 PLS[2:0] (PWR_CR 寄存器)

0: PVD 中断信号断开与 TIM1/15/16/17 刹车 (Break) 输入的连接。PVDE 和 PLS[2:0] 位可被应用编程。

1: PVD 中断信号连接到 TIM1/15/16/17 刹车 (Break) 输入 , PVDE 和 PLS[2:0] 位为只读。

位 1 SRAM_PARITY_LOCK: SRAM 校验锁定位

该位由软件设置，由系统复位清除。可用于锁定 SRAM 校验错信号连接到 TIM1/15/16/17 的刹车 (Break) 输入。

0: SRAM 校验错信号断开与 TIM1/15/16/17 刹车 (Break) 输入的连接

1: SRAM 校验错信号连接到 TIM1/15/16/17 刹车 (Break) 输入

位 0 LOCKUP_LOCK: Cortex-M0 LOCKUP 位使能位

该位由软件设置，由系统复位清除。它可用于使能并锁定连接 Cortex-M0 LOCKUP (硬件故障) 输出到 TIM1/15/16/17 的刹车 (Break) 输入。

0: Cortex-M0 LOCKUP 输出断开与 TIM1/15/16/17 刹车 (Break) 输入的连接

1: Cortex-M0 LOCKUP 输出连接到 TIM1/15/16/17 刹车 (Break) 输入

9.1.7 SYSCFG 寄存器映射

下表给出 SYSCFG 寄存器地址映射和复位值。

表 21. SYSCFG 寄存器映射及复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	SYSCFG_CFGR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	I2C_PB9_FM+	I2C_PB8_FM+	I2C_PB7_FM+	I2C_PB6_FM+	Res.	Res.	Res.	TIM17_DMA_RMP	TIM16_DMA_RMP	USART1_RX_DMA_RMP	USART1_TX_DMA_RMP	ADC_DMA_RMP	Res.	Res.	Res.	Res.	Res.	Res.	MEM_MODE	
	Reset value													rw	rw	rw	rw				rw	rw	rw	rw	rw							X	X
0x08	SYSCFG_EXTICR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI3[3:0]				EXTI2[3:0]				EXTI1[3:0]				EXTI0[3:0]			
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	SYSCFG_EXTICR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI7[3:0]				EXTI6[3:0]				EXTI5[3:0]				EXTI4[3:0]			
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	SYSCFG_EXTICR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI11[3:0]				EXTI10[3:0]				EXTI9[3:0]				EXTI8[3:0]			
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 21. SYSCFG 寄存器映射及复位值（续）

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x14	SYSCFG_EXTICR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI15[3:0]				EXTI14[3:0]				EXTI13[3:0]				EXTI12[3:0]			
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	SYSCFG_CFGR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SRAM_PEF	Res.	Res.	Res.	Res.	Res.	PVD_LOCK	SRAM_PARITY_LOCK	LOCUP_LOCK
	Reset value																								0						0	0	0

参考 2.2.2 章节的寄存器起始地址

10 DMA 控制器 (DMA)

10.1 DMA 简介

直接存储器存取 (DMA) 用来提供在外设和存储器之间或者存储器和存储器之间的高速数据传输。无须 CPU 干预，数据都可以通过 DMA 进行快速地传输。这就为其他操作保留了 CPU 资源。

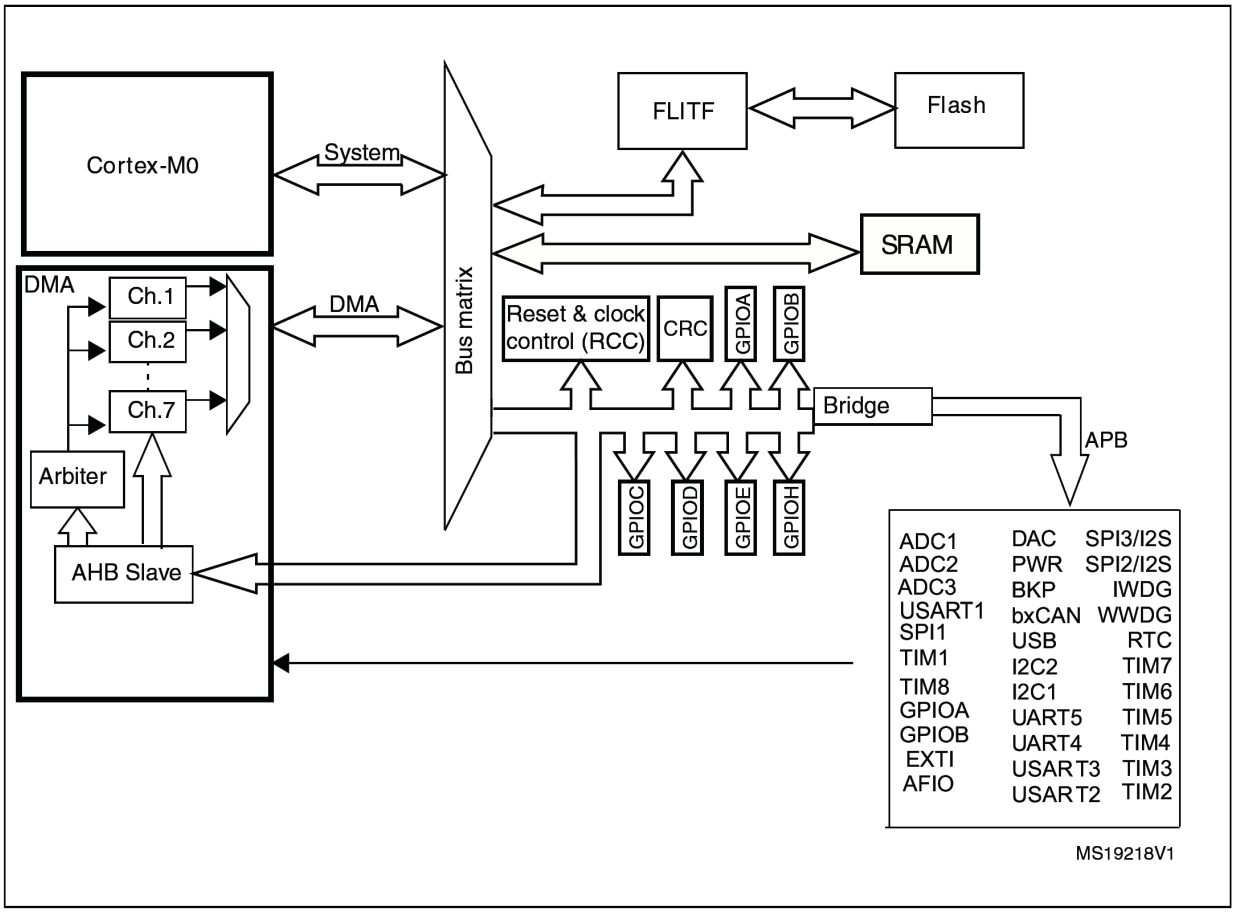
两个 DMA 控制器共有 5 个通道，每个通道专门用来管理来自于一个或多个外设对存储器访问的请求。还有一个仲裁器来协调各个 DMA 请求的优先权。

10.2 DMA 主要特性

- 5 个独立的可配置通道 (请求)
- 每个通道都直接连接专用的硬件 DMA 请求，每个通道都同样支持软件触发。这些配置通过软件来完成。
- 在同一个 DMA 模块上，多个请求间的优先权可以通过软件编程设置 (共有四级：很高、高、中等和低)，优先权设置相等时由硬件决定 (请求 1 优先于请求 2，依此类推) 。
- 独立数据源和目标数据区的传输宽度 (字节、半字、全字)，模拟打包和拆包的过程。源和目标地址必须按数据传输宽度对齐
- 支持循环的缓冲器管理
- 每个通道都有 3 个事件标志 (DMA 半传输、DMA 传输完成和 DMA 传输出错)，这 3 个事件标志逻辑或成为一个单独的中断请求。
- 存储器和存储器间的传输
- 外设到存储器和存储器到外设，外设到外设间的传输
- 闪存、SRAM、APB 和 AHB 外设均可作为访问的源和目标
- 可编程的数据传输数目：最大为 65536

DMA 模块图如下图所示：

图 18. STM32F05xx 的 DMA 框图



10.3 DMA 功能描述

DMA 控制器和 Cortex-M0 内核共享系统数据总线，执行直接存储器数据传输。当 CPU 和 DMA 同时访问相同的目标 (RAM 或外设) 时，DMA 请求会暂停 CPU 访问系统总线达若干个周期，总线仲裁器执行循环调度，以确保 CPU 至少可以得到一半的系统总线带宽 (存储器和外设)。

10.3.1 DMA 处理

在发生一个事件后，外设向 DMA 控制器发送一个请求信号。DMA 控制器根据通道的优先权处理请求。当 DMA 控制器开始访问发出请求的外设时，DMA 控制器立即发送给它一个应答信号。当从 DMA 控制器得到应答信号时，外设立即释放它的请求。一旦外设释放了这个请求，DMA 控制器同时撤销应答信号。如果有更多的请求时，外设可以启动下一个周期。

总之 , 每次的 DMA 传输由 3 组操作组成：

- 从外设数据寄存器或者从当前外设 / 存储器地址寄存器指示的存储器地址取数据，第一次传输时的开始地址是 DMA_CPARx 或 DMA_CMARx 寄存器指定的外设基地址或存储器单元。
- 从外设数据寄存器或者从当前外设 / 存储器地址寄存器指示的存储器地址取数据，第一次传输时的开始地址是 DMA_CPARx 或 DMA_CMARx 寄存器指定的外设基地址或存储器单元。
- 对 DMA_CNDTRx 寄存器执行一次递减操作，DMA_CNDTRx 寄存器存放着未完成的 DMA 操作的计数。

10.3.2 仲载器

仲裁器根据优先级管理着通道的请求和启动外设 / 存储器的访问

优先级管理分两个方面：

- 软件：可通过 DMA_CCRx 寄存器配置每个通道的优先级，优先级分 4 个等级：
  - 最高优先级
  - 高优先级
  - 中等优先级
  - 低优先级
- 硬件：如果 2 个请求有相同的软件优先级，则较低编号的通道比较高编号的通道有较高的优先权。举个例子，通道 2 优先于通道 4。

10.3.3 DMA 通道

每个通道都可以在外设寄存器固定地址和存储器地址之间执行 DMA 传输。DMA 传输的数据量是可编程的，最大达到 65535。每次传输之后相应的计数寄存器都做一次递减操作，直到计数为 0 。

可编程的传输数据量

外设和存储器的传输数据量可以通过 DMA_CCRx 寄存器中的 PSIZE 和 MSIZE 位编程。

增量指针

通过设置 DMA_CCRx 寄存器中的 PINC 和 MINC 标志位，外设和存储器的指针在每次传输后可以有选择地完成自动增量。当设置为增量模式时，下一个要传输的地址将是前一个地址加上增量值，增量值取决与所选的数据宽度为 1、2 或 4。第一个传输的地址是存放在 DMA_CPARx /DMA_CMARx 寄存器中地址。在传输过程中，这些寄存器保持它们初始的数值，软件不能改变和读出当前正在传输的地址 (它在内部的当前外设 / 存储器地址寄存器中)。

当通道配置为非循环传输模式时，传输结束后 (即传输计数已递减到 0) 将不再产生 DMA 请求。为了在 DMA_CNDTRx 寄存器中重新写入传输数目，需要先关闭相应的 DMA 通道。

注： 假如一个 DMA 通道关闭，DMA 寄存器的值未复位。DMA 通道寄存器 (DMA_CCRx, DMA_CPARx 和 DMA_CMARx) 在通道配置期间会保持着初始值。

在循环模式下，最后一次传输结束时，DMA_CNDTRx 寄存器自动重新装载为初始编程的计数值。当前内部地址寄存器从 DMA_CPARx/DMA_CMARx 寄存器中装载基地址值。

通道配置过程

下面给出的是 DMA 通道 x (x 代表通道号) 的配置步骤。

1. 在 DMA_CPARx 寄存器中设置外设寄存器地址。发生外设数据传输时，这个地址将是数据传输的源或目的地址。
2. 在 DMA_CMARx 寄存器中设置存储器地址。发生外设数据传输时，传输的数据将从这个地址读出或写入。
3. 在 DMA_CNDTRx 寄存器中写入需要传输的数据量，每次 DMA 传输后，该寄存器值递减。
4. 在 DMA_CCRx 的 PL[1:0] 位中配置通道的优先级。
5. 在 DMA_CCRx 中配置数据的传输方向、循环模式、外设和存储器的增量模式、外设和存储器的数据宽度和传输一半产生中断或传输完成产生中断的设置。
6. 设置 DMA_CCRx 寄存器中的 ENABLE 位来启动该通道。

一旦启动了 DMA 通道，它既可响应连接到该道道上的外设的 DMA 请求。

当传输一半的数据后，半传输标志 (HTIF) 被置 1 ，当设置了允许半传输中断位 (HTIE) 时，将产生一个半传输完成的中断请求。当传输完成时，传输完成标志 (TCIF) 被置 1 ，当设置了允许传输完成中断位 (TCIE) 时，将产生一个传输完成的中断请求。

循环模式

循环模式用于处理循环缓冲区和连续的数据传输 (如 ADC 的扫描模式)。在 DMA_CCRx 寄存器中的 CIRC 位用于开启这一功能。当启动了循环模式，一组的数据传输完成时，计数寄存器将会自动地被恢复成配置该通道时设置的初值，DMA 操作将会继续进行。

存储器到存储器模式

DMA 通道的操作可以在没有外设请求的情况下进行，这种操作就是存储器到存储器模式。.

当设置了 DMA_CCRx 寄存器中的 MEM2MEM 位之后，在软件设置了 DMA_CCRx 寄存器中的 EN 位启动 DMA 通道时，DMA 传输将马上开始。当 DMA_CNDTRx 寄存器变为 0 时，DMA 传输结束。存储器到存储器模式不能与循环模式同时使用。

10.3.4 可编程的数据宽度，数据对齐方式和数据大小端

当 PSIZE 和 MSIZE 不相同时，DMA 模块按照表 22 进行数据对齐。

表 22. 可编程的数据传输宽度和大小端操作 (当 PINC = MINC = 1)

Source port width	Destination port width	Number of data items to transfer (NDT)	Source content: address / data	Transfer operations	Destination content: address / data
8	8	4	@0x0 / B0 @0x1 / B1 @0x2 / B2 @0x3 / B3	1: READ B0[7:0] @0x0 then WRITE B0[7:0] @0x0 2: READ B1[7:0] @0x1 then WRITE B1[7:0] @0x1 3: READ B2[7:0] @0x2 then WRITE B2[7:0] @0x2 4: READ B3[7:0] @0x3 then WRITE B3[7:0] @0x3	@0x0 / B0 @0x1 / B1 @0x2 / B2 @0x3 / B3
8	16	4	@0x0 / B0 @0x1 / B1 @0x2 / B2 @0x3 / B3	1: READ B0[7:0] @0x0 then WRITE 00B0[15:0] @0x0 2: READ B1[7:0] @0x1 then WRITE 00B1[15:0] @0x2 3: READ B3[7:0] @0x2 then WRITE 00B2[15:0] @0x4 4: READ B4[7:0] @0x3 then WRITE 00B3[15:0] @0x6	@0x0 / 00B0 @0x2 / 00B1 @0x4 / 00B2 @0x6 / 00B3
8	32	4	@0x0 / B0 @0x1 / B1 @0x2 / B2 @0x3 / B3	1: READ B0[7:0] @0x0 then WRITE 000000B0[31:0] @0x0 2: READ B1[7:0] @0x1 then WRITE 000000B1[31:0] @0x4 3: READ B3[7:0] @0x2 then WRITE 000000B2[31:0] @0x8 4: READ B4[7:0] @0x3 then WRITE 000000B3[31:0] @0xC	@0x0 / 000000B0 @0x4 / 000000B1 @0x8 / 000000B2 @0xC / 000000B3
16	8	4	@0x0 / B1B0 @0x2 / B3B2 @0x4 / B5B4 @0x6 / B7B6	1: READ B1B0[15:0] @0x0 then WRITE B0[7:0] @0x0 2: READ B3B2[15:0] @0x2 then WRITE B2[7:0] @0x1 3: READ B5B4[15:0] @0x4 then WRITE B4[7:0] @0x2 4: READ B7B6[15:0] @0x6 then WRITE B6[7:0] @0x3	@0x0 / B0 @0x1 / B2 @0x2 / B4 @0x3 / B6
16	16	4	@0x0 / B1B0 @0x2 / B3B2 @0x4 / B5B4 @0x6 / B7B6	1: READ B1B0[15:0] @0x0 then WRITE B1B0[15:0] @0x0 2: READ B3B2[15:0] @0x2 then WRITE B3B2[15:0] @0x2 3: READ B5B4[15:0] @0x4 then WRITE B5B4[15:0] @0x4 4: READ B7B6[15:0] @0x6 then WRITE B7B6[15:0] @0x6	@0x0 / B1B0 @0x2 / B3B2 @0x4 / B5B4 @0x6 / B7B6
16	32	4	@0x0 / B1B0 @0x2 / B3B2 @0x4 / B5B4 @0x6 / B7B6	1: READ B1B0[15:0] @0x0 then WRITE 0000B1B0[31:0] @0x0 2: READ B3B2[15:0] @0x2 then WRITE 0000B3B2[31:0] @0x4 3: READ B5B4[15:0] @0x4 then WRITE 0000B5B4[31:0] @0x8 4: READ B7B6[15:0] @0x6 then WRITE 0000B7B6[31:0] @0xC	@0x0 / 0000B1B0 @0x4 / 0000B3B2 @0x8 / 0000B5B4 @0xC / 0000B7B6
32	8	4	@0x0 / B3B2B1B0 @0x4 / B7B6B5B4 @0x8 / BBBAB9B8 @0xC / BFBEBDBC	1: READ B3B2B1B0[31:0] @0x0 then WRITE B0[7:0] @0x0 2: READ B7B6B5B4[31:0] @0x4 then WRITE B4[7:0] @0x1 3: READ BBBAB9B8[31:0] @0x8 then WRITE B8[7:0] @0x2 4: READ BFBEBDBC[31:0] @0xC then WRITE BC[7:0] @0x3	@0x0 / B0 @0x1 / B4 @0x2 / B8 @0x3 / BC
32	16	4	@0x0 / B3B2B1B0 @0x4 / B7B6B5B4 @0x8 / BBBAB9B8 @0xC / BFBEBDBC	1: READ B3B2B1B0[31:0] @0x0 then WRITE B1B0[7:0] @0x0 2: READ B7B6B5B4[31:0] @0x4 then WRITE B5B4[7:0] @0x1 3: READ BBBAB9B8[31:0] @0x8 then WRITE B9B8[7:0] @0x2 4: READ BFBEBDBC[31:0] @0xC then WRITE BDBC[7:0] @0x3	@0x0 / B1B0 @0x2 / B5B4 @0x4 / B9B8 @0x6 / BDBC
32	32	4	@0x0 / B3B2B1B0 @0x4 / B7B6B5B4 @0x8 / BBBAB9B8 @0xC / BFBEBDBC	1: READ B3B2B1B0[31:0] @0x0 then WRITE B3B2B1B0[31:0] @0x0 2: READ B7B6B5B4[31:0] @0x4 then WRITE B7B6B5B4[31:0] @0x4 3: READ BBBAB9B8[31:0] @0x8 then WRITE BBBAB9B8[31:0] @0x8 4: READ BFBEBDBC[31:0] @0xC then WRITE BFBEBDBC[31:0] @0xC	@0x0 / B3B2B1B0 @0x4 / B7B6B5B4 @0x8 / BBBAB9B8 @0xC / BFBEBDBC

寻址一个不支持字节或半字写的 AHB 外设

当 DMA 模块开始一个 AHB 的字节或半字写操作时，数据将在 HWDATA[31:0] 总线中未使用的部分重复。因此，如果 DMA 以字节或半字写入不支持字节或半字写操作的 AHB 设备时 (即 HSIZE 不适于该模块)

不会发生错误，DMA 将按照下面两个例子写入 32 位 HWDATA 数据：

- 当 HSIZE= 半字时，写入半字‘0xABCD’，DMA 将设置 HWDATA 总线为‘0xABCDABCD’。
- 当 HSIZE= 字节时，写入字节‘0xAB’，DMA 将设置 HWDATA 总线为‘0xABABABAB’。

假定 AHB/APB 桥是一个 AHB 的 32 位从设备，它不处理 HSIZE 参数，它将按照下述方式把任何 AHB 上的字节或半字按 32 位传送到 APB 上：

- 一个 AHB 上对地址 0x0(或 0x1、0x2 或 0x3) 的写字节数据’0xB0’操作，将转换到 APB 上对地址 0x0 的写字数据’0xB0B0B0B0’操作。
- 一个 AHB 上对地址 0x0(或 0x2) 的写半字数据’0xB1B0’操作，将转换到 APB 上对地址 0x0 的写字数据’0xB1B0B1B0’操作。

例如，如果要写入 APB 后备寄存器 (与 32 位地址对齐的 16 位寄存器)，需要配置存储器数据源宽度 (MSIZE) 为’16 位’，外设目标数据宽度 (PSIZE) 为’32 位’。

10.3.5 错误管理

读写一个保留的地址区域，将会产生 DMA 传输错误。当在 DMA 读写操作时发生 DMA 传输错误时，硬件会自动地清除发生错误的通道所对应的通道配置寄存器 (DMA_CCRx) 的 EN 位，该通道操作被停止。此时，在 DMA_IFR 寄存器中对应该通道的传输错误中断标志位 (TEIF) 将被置位，如果在 DMA_CCRx 寄存器中设置了传输错误中断允许位，则将产生中断。

10.3.6 中断

每个 DMA 通道都可以在 DMA 传输过半、传输完成和传输错误时产生中断。为应用的灵活性考虑，通过设置寄存器的不同位来打开这些中断

表 23. DMA 中断请求

中断事件	事件标志位	使能控制位
传输过半	HTIF	HTIE
传输完成	TCIF	TCIE
传输错误	TEIF	TEIE

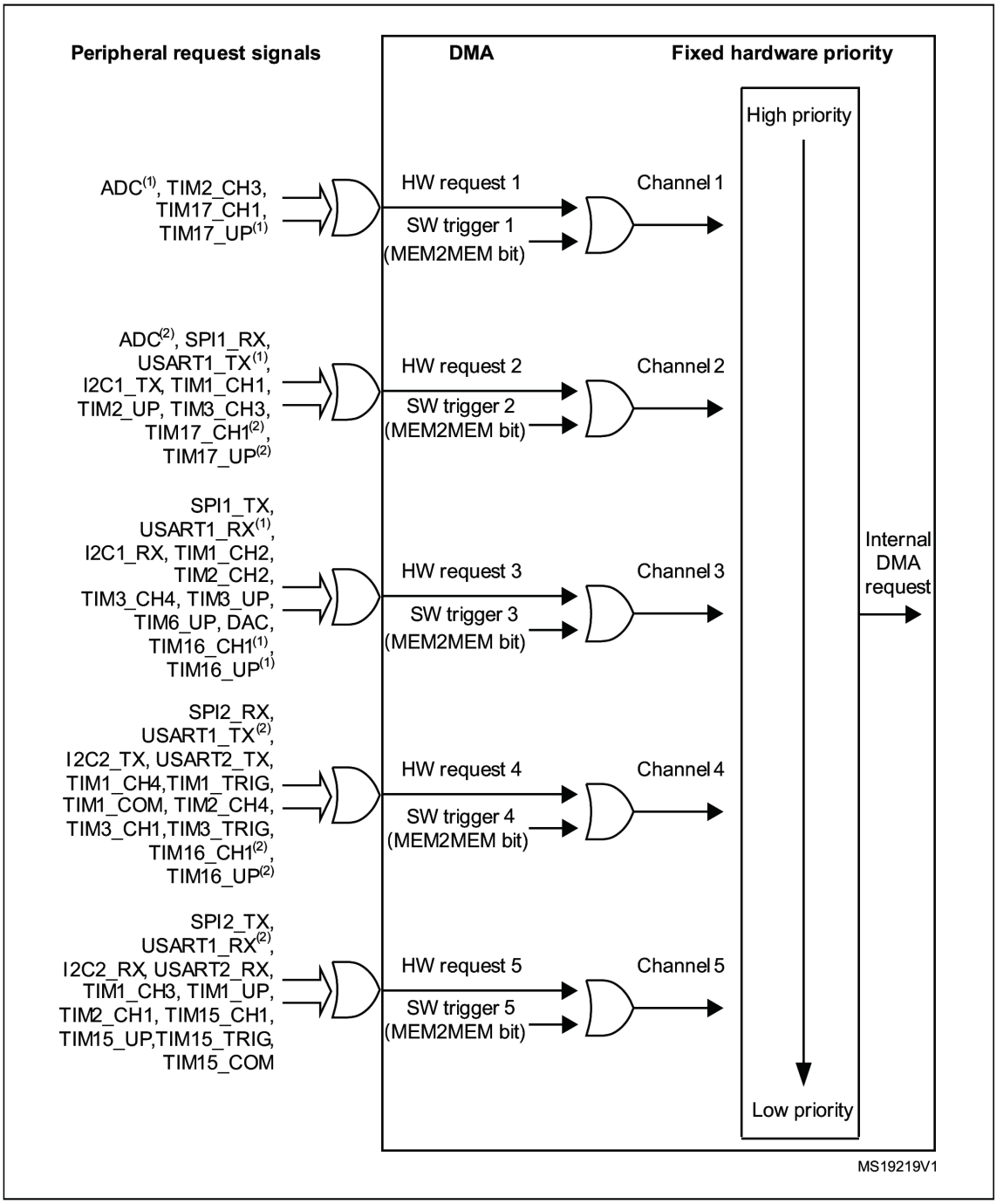
10.3.7 DMA 请求映象

DMA 控制器

外设 (TIMx, ADC , DAC, SPI, I2C, 和 USARTx) 的硬件请求简单地进行逻辑或运算后进入 DMA。这就意味着同一刻只允许一个 DMA 请求进入 DMA 控制器。参见图 19：DAM 请求映象。

外设的 DMA 请求，可以通过设置相应外设寄存器中的控制位，被独立地开启或关闭。

图 19. DMA 请求映象



1. 只有在 SYSCFG_CFGR1 寄存器相应的重映象位清 0 时该 DMA 请求映象到这个 DAM 通道。详情请参见 9.1.1 章节 SYSCFG 配置寄存器 1 (SYSCFG_CFGR1)
2. 只有在 SYSCFG_CFGR1 寄存器相应的重映象位置位时该 DMA 请求映象到这个 DMA 通道。详情请参见 9.1.1 章节 SYSCFG 配置寄存器 1 (SYSCFG_CFGR1)

表 24 各个通道的 DMA 请求列表

表 24. 各个通道的 DMA 请求一览表

Peripherals	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
ADC	ADC(1)	ADC(2)			
SPI		SPI1_RX	SP1_TX	SPI2_RX	SPI2_TX
USART		USART1_TX(1)	USART1_RX(1)	USART1_TX(2) USART2_TX	USART1_RX(2) USART2_RX
I2C		I2C1_TX	I2C1_RX	I2C2_TX	I2C2_RX
TIM1		TIM1_CH1	TIM1_CH2	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_CH3 TIM1_UP
TIM2	TIM2_CH3	TIM2_UP	TIM2_CH2	TIM2_CH4	TIM2_CH1
TIM3		TIM3_CH3	TIM3_CH4 TIM3_UP	TIM3_CH1 TIM3_TRIG	
TIM6 / DAC			TIM6_UP DAC		
TIM15					TIM15_CH1 TIM15_UP TIM15_TRIG TIM15_COM
TIM16			TIM16_CH1(1) TIM16_UP(1)	TIM16_CH1(2) TIM16_UP(2)	
TIM17	TIM17_CH1(1) TIM17_UP(1)	TIM17_CH1(2) TIM17_UP(2)			

1. 只有在 SYSCFG_CFGR1 寄存器相应的重映象位清 0 时该 DMA 请求映象到这个 DAM 通道。详情请参见 9.1.1 章节 SYSCFG 配置寄存器 1 (SYSCFG_CFGR1)
2. 只有在 SYSCFG_CFGR1 寄存器相应的重映象位置位时该 DMA 请求映象到这个 DMA 通道。详情请参见 9.1.1 章节 SYSCFG 配置寄存器 1 (SYSCFG_CFGR1)

10.4 DMA 寄存器

有关寄存器描述中用到的缩写，请参见 1.1 章节

外设寄存器可用字节 (8 位)、半字 (16 位)、字 (32 位) 访问。

10.4.1 DMA 中断状态寄存器 (DMA_ISR)

偏移地址： 0x00

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TEIF5	HTIF5	TCIF5	GIF5
												r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEIF4	HTIF4	TCIF4	GIF4	TEIF3	HTIF3	TCIF3	GIF3	TEIF2	HTIF2	TCIF2	GIF2	TEIF1	HTIF1	TCIF1	GIF1
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 保留，必须保持为复位值

位 19, 15, 11, 7, 3 TEIFx: 通道 x (x = 1 ..5) 传输错误标志

该位由硬件设置。由软件对 DMA_IFCR 寄存器的相应位写 1 清除。

0: 在通道 x 上无传输错误 (TE)

1: 在通道 x 上发生了传输错误 (TE)

位 18, 14, 10, 6, 2 HTIFx: 通道 x(x = 1 ..5) 传输一半标志

该位由硬件设置。由软件对 DMA_IFCR 寄存器的相应位写 1 清除。

0: 在通道 x 上无传输一半 (HT) 的事件发生

1: 在通道 x 上发生了传输一半 (HT) 的事件

位 17, 13, 9, 5, 1 TCIFx: 通道 x (x = 1 ..5) 传输完成标志

该位由硬件设置。由软件对 DMA_IFCR 寄存器的相应位写 1 清除。

0: 在通道 x 上无传输完成 (TC) 的事件发生

1: 在通道 x 上发生了传输 (TC) 的事件

位 16, 12, 8, 4, 0 GIFx: 通道 x(x = 1 ..5) 全局中断标志

该位由硬件设置。由软件对 DMA_IFCR 寄存器的相应位写 1 清除。

0: 在通道 x 上无 TE、HT 或 TC 事件发生

1: 在通道 x 上发生了 TE、HT 或 TC 事件

10.4.2 DMA 中断标志清除寄存器 (DMA_IFCR)

偏移地址 : 0x04

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTEIF5	CHTIF5	CTCIF5	CGIF5
												w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTEIF4	CHTIF4	CTCIF4	CGIF4	CTEIF3	CHTIF3	CTCIF3	CGIF3	CTEIF2	CHTIF2	CTCIF2	CGIF2	CTEIF1	CHTIF1	CTCIF1	CGIF1
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:28 保留，必须保持为复位值

位 19, 15, 11, 7, 3 CTEIFx: 通道 x (x = 1 ..5) 传输错误清除，该位由软件置 1 清除。
0: 无效
1: 在 DMA_ISR 寄存器中清除相应的 TEIF 标志

位 18, 14, 10, 6, 2 CHTIFx: 通道 x (x = 1 ..5) 传输一半标志清除，该位由软件置 1 清除。
0: 无效
1: 在 DMA_ISR 寄存器中清除相应的 HTIF 标志

位 17, 13, 9, 5, 1 CTCIFx: 通道 x (x = 1 ..5) 传输完成标志清除，该位由软件置 1 清除。
0: 无效
1: 在 DMA_ISR 寄存器中清除相应的 TCIF 标志

位 16, 12, 8, 4, 0 CGIFx: 通道 x (x = 1 ..5) 全局中断标志清除，该位由软件置 1 清除。
0: 无效
1: 在 DMA_ISR 寄存器中清除 GIF、TEIF、HTIF 和 TCIF 标志

10.4.3 DMA 通道 x 配置寄存器 (DMA_CCRx) (x = 1..5)

偏移地址 : 0x08 + 0d20 × (通道号 – 1)

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:15 保留，必须保持为复位值

位 14 MEM2MEM: 存储器到存储器模式

该位由软件置 1 清除。

0: 非存储器到存储器模式

1: 启动存储器到存储器模式

位 13:12 PL[1:0]: 通道优先级

这些位由软件设置和清除。

00: 低

01: 中

10: 高

11: 最高

位 11:10 MSIZE[1:0]: 存储器数据宽度

这些位由软件设置和清除。

00: 8 位

01: 16 位

10: 32 位

11: 保留

位 9:8 PSIZE[1:0]: 外设数据宽度

这些位由软件设置和清除。

00: 8 位

01: 16 位

10: 32 位

11: 保留

位 7 MINC: 存储器地址增量模式

该位由软件置 1 清除。

0: 不执行存储器地址增量操作

1: 执行存储器地址增量操作

位 6 PINC: 外设地址增量模式

该位由软件置 1 清除。

0: 不执行外设地址增量模式操作

1: 执行外设地址增量模式操作

位 5 CIRC: 循环模式

该位由软件置 1 清除。

0: 不执行循环模式操作

1: 执行循环模式操作

位 4 DIR: 数据传输方向
该位由软件置 1 清除。
0: 从外设读
1: 从存储器读

位 3 TEIE: 传输错误中断使能
该位由软件置 1 清除。
0: 禁止 TE 中断
1: 允许 TE 中断

位 2 HTIE: 传输一半中断使能
该位由软件置 1 清除。
0: 禁止 HT 中断
1: 允许 HT 中断

位 1 TCIE: 传输完成中断使能
该位由软件置 1 清除。
0: 禁止 TC 中断
1: 允许 TC 中断

位 0 EN: 通道使能
该位由软件置 1 清除。
0: 通道不工作
1: 开启通道

10.4.4 DMA 通道 x 传输数量寄存器 (DMA_CNDTRx) (x = 1..5)

偏移地址 : 0x0C + 0d20 × (通道号 – 1)
复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	1	1	10	9	8	7	6	5	4	3	2	1	0
NDT															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持为复位值

位 15:0 NDT[15:0]: 传输数据的数量
传输数据的数量 (0 到 65535)。这个寄存器只能在通道不工作 (DMA_CCRx 的 EN=0) 时写入。一旦允许通道工作，该寄存器变为只读，其值为剩余待传输数据字节数目。寄存器内容在每次 DMA 传输后递减。
数据传输结束后，寄存器的内容或者变为 0; 或者当该通道配置为自动重加载模式时，寄存器的内容将被自动重新加载为之前配置时的数值。
当寄存器的内容为 0 时，无论通道是否开启，都不会发生任何数据传输。

10.4.5 DMA 通道 x 外设地址寄存器 (DMA_CPARx) (x = 1..5)

偏移地址 : 0x10 + 0d20 × (通道号 – 1)

复位值 : 0x0000 0000

当通道工作时道 (DMA_CCRx 的 EN=1) 不能写该寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PA																															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 PA[31:0]: 外设地址

外设数据寄存器的基地址，作为数据传输的源或目标。

当 PSIZE=’01’(16 位)，不使用 PA[0] 位。操作自动地与半字地址对齐。

当 PSIZE=’10’(32 位)，不使用 PA[1:0] 位。操作自动地与字地址对齐。

10.4.6 DMA 通道 x 存储器地址寄存器 (DMA_CMARx) (x = 1..5)

偏移地址 : 0x14 + 0d20 × (通道号 – 1)

复位值 : 0x0000 0000

当通道工作时道 (DMA_CCRx 的 EN=1) 不能写该寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MA																															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 MA[31:0]: 存储器地址

存储器地址作为数据传输的源或目标。

当 MSIZE =’01’(16 位)，不使用 MA[0] 位。操作自动地与半字地址对齐。

当 MSIZE =’10’(32 位)，不使用 MA[1:0] 位。操作自动地与字地址对齐。

10.4.7 DMA 寄存器映像

下表给出 DMA 寄存器映像及其复位值 .

表 25. DMA 寄存器映像及复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	**DMA_ISR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TEIF5	HTIF5	TCIF5	GIF5	TEIF4	HTIF4	TCIF4	GIF4	TEIF3	HTIF3	TCIF3	GIF3	TEIF2	HTIF2	TCIF2	GIF2	TEIF1	HTIF1	TCIF1	GIF1
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x004	**DMA_IFCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTEIF5	CHTIF5	CTCIF5	CGIF5	CTEIF4	CHTIF4	CTCIF4	CGIF4	CTEIF3	CHTIF3	CTCIF3	CGIF3	CTEIF2	CHTIF2	CTCIF2	CGIF2	CTEIF1	CHTIF1	CTCIF1	CGIF1
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x008	**DMA_CCR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL [1:0]		M SIZE [1:0]		PSIZE [1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C	**DMA_CNDTR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDT[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x010	**DMA_CPAR1**	PA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x014	**DMA_CMAR1**	MA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x01C	**DMA_CCR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL [1:0]		M SIZE [1:0]		PSIZE [1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x020	**DMA_CNDTR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDT[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x024	**DMA_CPAR2**	PA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x028	**DMA_CMAR2**	MA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x030	**DMA_CCR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL [1:0]		M SIZE [1:0]		PSIZE [1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x034	**DMA_CNDTR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDT[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x038	**DMA_CPAR3**	PA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x03C	**DMA_CMAR3**	MA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x044	**DMA_CCR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL [1:0]		M SIZE [1:0]		PSIZE [1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x048	**DMA_CNDTR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDT[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 25. DMA 寄存器映像及复位值 (续)

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x04C	**DMA_CPAR4**	PA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x050	**DMA_CMAR4**	MA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x054	Reserved																																
0x058	**DMA_CCR5**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL [1:0]		M SIZE [1:0]		PSIZE [1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x05C	**DMA_CNDTR5**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDT[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x060	**DMA_CPAR5**	PA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x064	**DMA_CMAR5**	MA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

有关寄存器起始参见 2.2.2 章节

11 中断和事件

11.1 嵌套向量中断控制器 (NVIC)

11.1.1 NVIC 主要特性

- 32 个可屏蔽中断通道 (不包含 16 个 Cortex-M0 的中断线)
- 4 个可编程的优先级 (使用了 2 位的中断优先级)
- 低延时的异常和中断处理
- 电源管理控制
- 系统控制寄存器的实现

嵌套向量中断控制器 (NVIC) 和处理器核的接口紧密相连，可以实现低延迟的中断处理和高效地处理晚到的中断。

嵌套向量中断控制器管理着包括内核异常等中断。更多关于异常和 NVIC 编程的说明请参考《STM32F051xx Cortex-M0 编程手册》。

11.1.2 SysTick 校准值寄存器

SysTick 校准值设置为 6000， 当 SysTick 时钟设置为 6 MHz (fHCLK/8 的最大值) 时，会产生 1ms 时间基准。

11.1.3 中断和异常向量

表 26 STM32F51xx 器件向量表

表 26. 向量表

位置	优先级	优先级类型	名称	说明	地址
	-	-	-	保留 (Reserved)	0x0000 0000
	-3	固定	Reset	复位 (Reset)	0x0000 0004
	-2	固定	NMI	不可屏蔽中断。RCC 时钟安全系统 (CSS) 联接到 NMI 向量	0x0000 0008
	-1	固定	HardFault	所有类型的错误 (fault)	0x0000 000C
	3	可设置	SVCall	通用 SWI 指令调用的系统服务	0x0000 002C
	5	可设置	PendSV	可挂起的系统服务	0x0000 0038
	6	可设置	SysTick	系统嘀嗒定时器	0x0000 003C
0		可设置	WWDG	窗口看门狗中断	0x0000 0040
1		可设置	PVD	连接到 EXTI 线的可编程电压检测 (PVD) 中断	0x0000 0044
2		可设置	RTC	RTC 全局中断	0x0000 0048

表 26. 向量表 (续)

位置	优先级	优先级类型	名称	说明	地址
3		可设置	FLASH	Flash 全局中断	0x0000 004C
4		可设置	RCC	RCC 全局中断	0x0000 0050
5		可设置	EXTI0_1	EXTI 线 [1:0] 中断	0x0000 0054
6		可设置	EXTI2_3	EXTI 线 [3:2] 中断	0x0000 0058
7		可设置	EXTI4_15	EXTI 线 15 和 EXTI 线 4 中断	0x0000 005C
8		可设置	TSC	触摸传感中断	0x0000 0060
9		可设置	DMA_CH1	DMA 通道 1 中断	0x0000 0064
10		可设置	DMA_CH2_3	DMA 通道 2 和 3 中断	0x0000 0068
11		可设置	DMA_CH4_5	DMA 通道 4 和 5 中断	0x0000 006C
12		可设置	ADC_COMP	ADC 和比较器 1 和 2 中断	0x0000 0070
13		可设置	TIM1_BRK_UP_TRG_COM	TIM1 刹车、更新、触发、和通信中断	0x0000 0074
14		可设置	TIM1_CC	TIM1 捕获比较中断	0x0000 0078
15		可设置	TIM2	TIM2 全局中断	0x0000 007C
16		可设置	TIM3	TIM3 全局中断	0x0000 0080
17		可设置	TIM6_DAC	TIM6 全局中断和 DAC 欠载中断	0x0000 0084
18		可设置	Reserved		0x0000 0088
19		可设置	TIM14	TIM14 全局中断	0x0000 008C
20		可设置	TIM15	TIM15 全局中断	0x0000 0090
21		可设置	TIM16	TIM16 全局中断	0x0000 0094
22		可设置	TIM17	TIM17 全局中断	0x0000 0098
23		可设置	I2C1	I2C1 全局中断	0x0000 009C
24		可设置	I2C2	I2C2 全局中断	0x0000 00A0
25		可设置	SPI1	SPI1 全局中断	0x0000 00A4
26		可设置	SPI2	SPI2 全局中断	0x0000 00A8
27		可设置	USART1	USART1 全局中断	0x0000 00AC
28		可设置	USART2	USART2 全局中断	0x0000 00B0
29		可设置	reserved		0x0000 00B4
30			CEC	CEC 全局中断	0x0000 00B8
31		可设置	reserved		0x0000 00BC

11.2 外部中断和事件控制器 (EXTI)

外部中断和事件控制器 (EXTI) 管理外部和内部异步事件 / 中断，并生成相应的事件请求到 CPU/ 中断控制器和到电源管理的唤醒请求。

EXTI 允许管理多达 28 个外部 / 内部事件线 (21 个外部事件线及 7 个内部事件线)。

每个外部中断线可以独立选择触发沿，而内部总为上升沿触发。一个中断可以一直让其挂起：如果发生了外部中断或事件， 状态寄存器示例并指示其中断源；一个事件通常是一个简单的脉冲，用于触发核的唤醒 (如 Cortex-M0 的 RXEV 引脚)。对于内部中断，挂起状态由核 (IP) 产生，所以不需要特定标志。每条输入线可单独屏蔽其产生中断或事件。另外，内部线只有在停止模式下采样。控制器允许软件写特定的寄存器仿真触发相应的线产生事件和中断。

11.2.1 主要特性

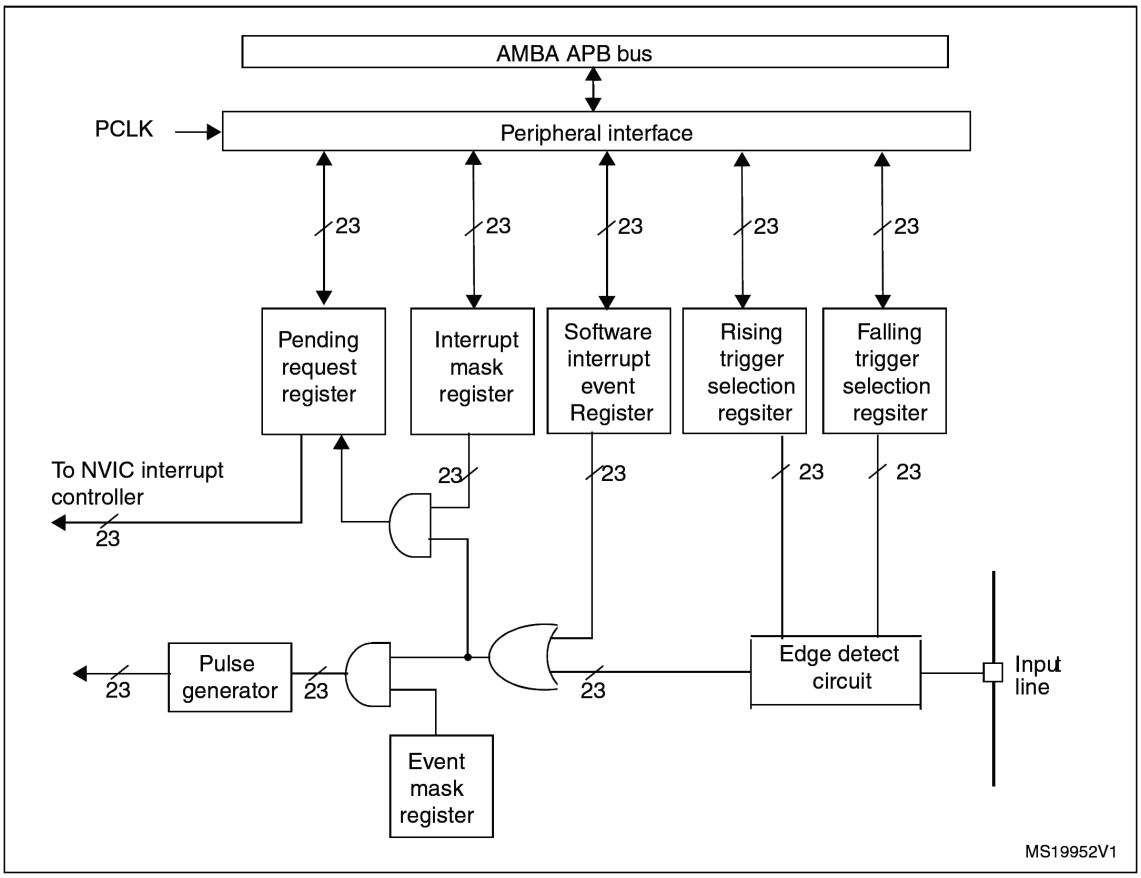
EXTI 主要特性如下：

- 支持产生多达 28 个事件 / 中断请求 ;
- 作为外部或内部事件请求的每一线都可独立配置；
- 每个事件 / 中断线都有独立的屏蔽；
- 当系统不处于停机 (STOP) 模式时自动禁止内部各线；
- 独立触发外部事件 / 中断线 ;
- 每个外部中断线都有专用的状态位；
- 仿真所有的外部事件请求。

11.2.2 框图

外部中断 / 事件框图如图 20 所示 .

图 20. 外部中断 / 事件框图



11.2.3 唤醒事件管理

STM32F05x 可以处理外部和内部事件来唤醒内核 (WFE)。 唤醒事件可由下述配置产生：

- 在外设控制寄存器中使能一个外部中断但不在 NVIC 中使能，同时使能 Cortex-M0 系统控制寄存器中的 SEVONPEND 位。当 MCU 从 WFE 恢复后，需要清除相应外设的中断挂起位和外设 NVIC 中断通道挂起位 (在 NVIC 中断清除挂起寄存器中)。
- 配置一个外部或内部 EXTI 线为事件模式，当 CPU 从 WFE 恢复后，因为对应事件线的挂起位没有被置位，不必清除相应外设的中断挂起位或 NVIC 中断通道挂起位。

11.2.4 异步的内部中断

部分通信外设 (UART，I2C， CEC) 在系统处于运行模式和在处于允许唤醒的停止模式中有能力产生事件。

为了实现这些功能，外设必须按要求产生同步 (到系统时钟，如 APB 时钟) 和异步事件。

11.2.5 功能说明

要产生中断，必须先配置好并使能中断线。根据需要的边沿检测设置 2 个触发寄存器，同时在中断屏蔽寄存器的相应位写’1’允许中断请求。当外部中断线上发生了期待的边沿时，将产生一个中断请求，对应的挂起位也随之被置’1’。在挂起寄存器的对应位写’1’，将清除该中断请求。

对于内部中断线，触发沿都为上升沿，中断屏蔽寄存器相应值使能这些中断，但内部中断线没有相应的挂起位。

如果需要产生事件，必须先配置好并使能事件线。根据需要的边沿检测通过设置 2 个触发寄存器，同时在事件屏蔽寄存器的相应位写’1’允许事件请求。当事件线上发生了期待的边沿时，将产生一个事件请求脉冲，对应的挂起位不被置’1’。

对于外部中断线，一个中断 / 事件请求也可由软件对相应的软件中断 / 事件寄存器位写’1’来产生。

注：　关联到内部线的中断或事件仅在系统处于停止模式下才能被触发。当系统运行时，不会产生该类似的中断 / 事件。

硬件中断选择

根据下列过程可配置 1 条线路做为中断源：

- 在 EXTI_IMR 寄存器中配置所选中断线的屏蔽位
- 在 (EXTI_RTSR 和 EXTI_FTSR) 中配置所选中断线的触发选择位
- 配置对应到外部中断控制器 (EXTI) 的 NVIC 中断通道的使能和屏蔽位，使得中断线中的请求可以被正确地响应。

硬件事件选择

根据下列过程可配置 1 条线路做为事件源：

- 在 EXTI_EMR 寄存器中配置所选事件的屏蔽位
- 在 (EXTI_RTSR 和 EXTI_FTSR) 中配置所选事件线的触发选择位

软件中断 / 事件的选择

任何外部中断线可被配置为软件中断 / 事件线。以下过程产生一个软件中断。

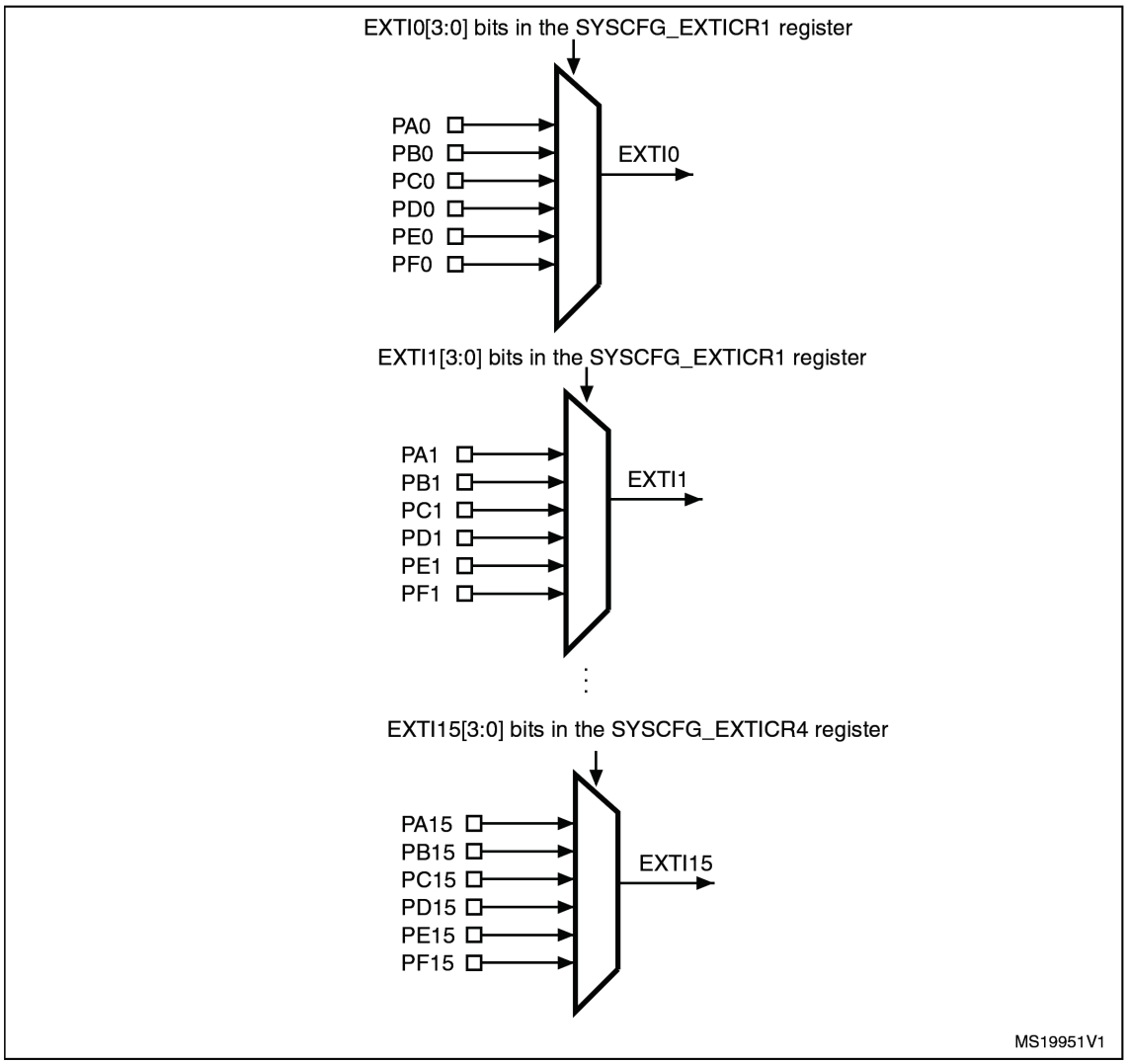
- 在 (EXTI_IMR, EXTI_EMR) 中配置相应的屏蔽位
- 在软件中断寄存器 (EXTI_SWIER) 设置相应的请求位

11.2.6 外部和内部中断 / 事件线路映像

在 STM32F05x 中，共有最多 28 中断 / 事件线可用： 7 线内部 (包含一条保留) 和 21 线外部。

GPIO 口连接到 16 个外部中断 / 事件线如下图 :

图 21. 外部中断 / 事件 GPIO 映像



余下线路连接如下：

- EXTI 16 线连接到 PVD 输出
- EXTI 17 线连接到 RTC Alarm 事件
- EXTI 18 线保留 (内部保持为低电平)
- EXTI 19 线连接到 RTC 窜改和时间戳
- EXTI 20 线保留 (内部保持为低电平)
- EXTI 21 线连接到比较器 1 的输出
- EXTI 22 线连接到比较器 2 的输出
- EXTI 23 线连接到 I2C1 唤醒
- EXTI 24 线保留 (内部保持为低电平)
- EXTI 25 线连接到 USART1 的唤醒
- EXTI 26 线保留 (内部保持为低电平)
- EXTI 27 线连接到 CEC 唤醒 .

注： EXTI 18, 20, 23, 24, 25, 26 和 27 为内部 I 线

11.3 EXTI 寄存器

有关寄存器的缩写参考 1.1 章节

该外设寄存器只能用 32 位字宽访问。

11.3.1 中断屏蔽寄存器 (EXTI_IMR)

偏移地址 : 0x00

复位值 : 0x0F94 0000 (请看以下注释)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	MR27	MR26	MR25	MR24	MR23	MR22	MR21	MR20	MR19	MR18	MR17	MR16
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR15	MR14	MR13	MR12	MR11	MR10	MR9	MR8	MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持为复位值 (0).

位 27:0 MRx: 外部 / 内部线 x 的中断屏蔽位

0: 屏蔽来自线 x 上的中断请求

1: 开放来自线 x 上的中断请求

注： 内部线 (18, 20, 23, 24, 25, 26 和 27) 的复位值 设为“1”，为了缺省情况下打开这些中断。

11.3.2 事件屏蔽寄存器 (EXTI_EMR)

偏移地址 : 0x04

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	MR27	MR26	MR25	MR24	MR23	MR22	MR21	MR20	MR19	MR18	MR17	MR16
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR15	MR14	MR13	MR12	MR11	MR10	MR9	MR8	MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持为复位值 (0).

位 27:0 MRx: 外部 / 内部线 x 的事件屏蔽位

0: 屏蔽来自线 x 上的事件请求

1: 开放来自线 x 上的事件请求

11.3.3 上升沿 触发选择寄存器 (EXTI_RTSR)

偏移地址 : 0x08

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TR19	Res.	TR17	TR16
												rw		rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TR15	TR14	TR13	TR12	TR11	TR10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:20 保留，必须保持为复位值。

位 19 TR19: 线 19 上的上升沿触发事件配置位

0: 禁止输入线 19 上的上升沿触发 (中断和事件)

1: 禁止输入线 19 上的上升沿触发 (中断和事件)

位 18 保留，必须保持为复位值。

位 17:0 TRx: 线 x 上的上升沿触发事件配置位 (x = 17 到 0)

0: 禁止输入线 x 上的上升沿触发 (中断和事件)

1: 禁止输入线 x 上的上升沿触发 (中断和事件).

注： 外部唤醒线是边沿触发的，这些线上不能出现毛刺信号。在写 EXTI_RTSR 寄存器时，在外部中断线上的上升沿信号不能被识别，挂起位也不会被置位。在同一中断线上，可以同时设置上升沿和下降沿触发。即任一边沿都可触发中断。

11.3.4 下降沿触发选择寄存器 (EXTI_FTSR)

偏移地址 : 0x0C

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TR19	Res.	TR17	TR16
												rw		rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TR15	TR14	TR13	TR12	TR11	TR10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:20 保留，必须保持为复位值。

位 19 TR19: 线 19 上的下降沿触发事件配置位

0: 禁止输入线 19 上的下降沿触发 (中断和事件)

1: 禁止输入线 19 上的下降沿触发 (中断和事件)

位 18 保留，必须保持为复位值。

位 17:0 TRx: 线 x 上的上升沿触发事件配置位 (x = 17 到 0)

0: 禁止输入线 x 上的下降沿触发 (中断和事件)

1: 禁止输入线 x 上的下降沿触发 (中断和事件).

注： 外部唤醒线是边沿触发的，这些线上不能出现毛刺信号。在写 EXTI_RTSR 寄存器时，在外部中断线上的上升沿信号不能被识别，挂起位也不会被置位。在同一中断线上，可以同时设置上升沿和下降沿触发。即任一边沿都可触发中断。

.

11.3.5 软件中断事件寄存器 (EXTI_SWIER)

偏移地址 : 0x10

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWIER 19	Res.	SWIER 17	SWIER 16
												rw		rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIER 15	SWIER 14	SWIER 13	SWIER 12	SWIER 11	SWIER 10	SWIER 9	SWIER 8	SWIER 7	SWIER 6	SWIER 5	SWIER 4	SWIER 3	SWIER 2	SWIER 1	SWIER 0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:20 保留，必须保持为复位值。

位 19 SWIER19: 19 线的软中断位

当该位为’0’时，写’1’将设置 EXTI_PR 中 RP19 的挂起位。如果在 EXTI_IMR 和 EXTI_EMR 中允许产生该中断，则此时将产生一个中断。

通过清除 EXTI_PR 的对应位 (写入’1’)，可以清除该位为’0’。

位 18 保留，必须保持为复位值。

位 17:0 SWIERx: 线 x 上的软件中断 (x = 17 to 0)

当该位为’0’时，写’1’将设置 EXTI_PR 中相应的挂起位。如果在 EXTI_IMR 和 EXTI_EMR 中允许产生该中断，则此时将产生一个中断。

通过清除 EXTI_PR 的对应位 (写入’1’)，可以清除该位为’0’。

11.3.6 挂起寄存器 (EXTI_PR)

偏移地址 : 0x14

复位值 : 不定

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PR19	Res.	PR17	PR16
												rc_w1		rc_w1	rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PR15	PR14	PR13	PR12	PR11	PR10	PR9	PR8	PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:20 保留，必须保持为复位值。

位 19 PR19: 线 19 挂起位
0: 没有发生触发请求
1: 发生了选择的触发请求
当在外部中断线上发生了选择的边沿事件，该位被置’1’。在该位中写入’1’可以清除它，也可以通过改变边沿检测的极性清除。

位 18 保留，必须保持为复位值。

位 19:0 PRx: 线 x 挂起位 (x = 17 到 0)
0: 没有发生触发请求
1: 发生了选择的触发请求
当在外部中断线上发生了选择的边沿事件，该位被置’1’。在该位中写入’1’可以清除它，也可以通过改变边沿检测的极性清除。

11.3.7 EXTI 寄存器映像

下表给出 EXTI 寄存器映像及复位值 .

表 27. 外部中断 / 事件控制寄存器映像及复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**EXTI_IMR**	Res.	Res.	Res.	Res.	MR[27:0]																											
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	**EXTI_EMR**	Res.	Res.	Res.	Res.	MR[27:0]																											
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	**EXTI_RTSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TR19	Res.	TR[17:0]																	
	Reset value													0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	**EXTI_FTSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TR19	Res.	TR[17:0]																	
	Reset value													0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	**EXTI_SWIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWIER19	Res.	SWIER[17:0]																	
	Reset value													0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	**EXTI_PR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PR19	Res.	PR[17:0]																	
	Reset value													0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

有关寄存器起始地址参见 2.2.2 章节

12 模拟数字转换器 (ADC)

12.1 介绍

12 位 ADC 是一种逐次逼近型模拟数字转换器。它有多达 19 个通道，可测量 16 个外部和 3 个内部信号源。各通道的 A/D 转换可以单次、连续、扫描或间断模式执行。ADC 的结果可以左对齐或右对齐方式存储在 16 位数据寄存器中。

模拟看门狗允许应用程序检测输入电压是否超出了用户设定的高 / 低阀值。

一个有效低功耗模式实施允许在低频情况下实现低能耗。

12.2 ADC 主要特性

- 高性能
  - 12- 位 , 10- 位 , 8- 位 或 6- 位可配置分辨率。
  - ADC 转换时间 : 1.0 µs @ 12- 位分辨率 (1 MHz), 0.93 µs @10- 位分辨率 , 在更低的转换分辨率下可达到更快的转换时间。
  - 自校准
  - 可编程采样时间
  - 带内嵌数据一致性的数据对齐
  - DMA 支持
- 低功耗
  - 应用为低功耗运行而降低 PLCK 频率的同时仍保持最佳的 ADC 性能。(举例：无论在何种 PCLK 的频率下，保持 1.0 µs 的 ADC 转换时间)
  - 自动延时模式：在应用运行在 PLCK 低速下，防止 ADC 超限。
  - 自动关闭模式：ADC 除了在转换期间工作外，其他时间 ADC 自动断电。这种方式大大降低了 ADC 的功耗。
- 模拟输入通道
  - 从外部 GPIO 口连接的 16 通道模拟输入
  - 1 通道内部温度传感 (VSENSE) 输入
  - 1 通道的内部参考电压 (VREFINT) 输入
  - 1 通道的外部电池 VBAT 供电引脚输入
- 多种启动转换方式：
  - 由软件
  - 由硬件触发 (从 TIM1、TIM2、TIM3 和 TIM15 发出的内部定时器事件)
- 转换模式
  - 可转换单通道或一序列通道。
  - 触发的选定输入单模式转换
  - 持续的选定输入连续模式转换
  - 继续模式 (Discontinuous mode)
- 转换完成后、序列转换完成、模拟看门狗或转换溢出事件都可以产生中断
- 模拟看门狗
- ADC 供电要求：2.4 V 到 3.6 V
- ADC 输入范围： VSSA ≤ VIN ≤ VDDA

图 22 给出 ADC 模块框图。

12.3 ADC 引脚和内部信号

表 28. ADC 内部信号

内部信号名称	信号类型	说明
TIMx_TRG	输入	从定时器来的内部信息
V_{SENSE}	输入	内部温度传感器输出电压
V_{REFINT}	输入	内部参考电压的输出
V_{BAT}	输入	V_{BAT} 引脚输入电压

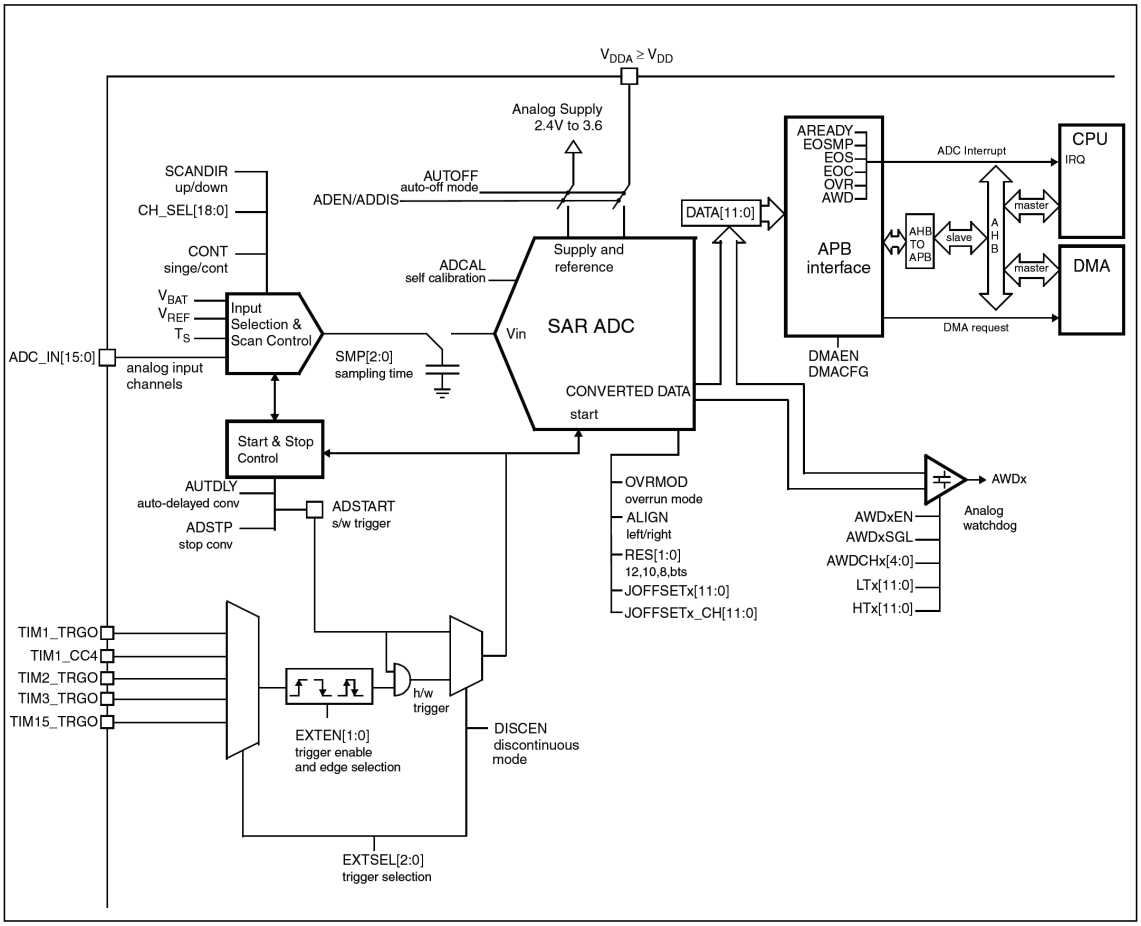
表 29. ADC 引脚

名称	信号类型	注释
V_{DDA}	输入，模拟供电电源	模拟供电电源，ADC 参考电压的正极， $V_{DDA} \geq V_{DD}$
V_{SSA}	输入，模拟电源地	模拟地，$V_{SSA} = V_{SS}$
ADC_IN[15:0]	模拟输入信号	16 路模拟输入

12.4 ADC 功能描述

图 22 给出 ADC 模块图，表 29 给出 ADC 引脚说明。

图 22. ADC 模块图



12.4.1 校准 (ADCAL)

ADC 本身具有校准功能，在校准期间，ADC 计算一个用于 ADC 校准的 7 位校准因子 (ADC 断电后丢失)。在 ADC 校准期间和校准未完成前，应用不能使用 ADC 模块。
在 A/D 转换前应执行校准操作，校准用于消除各芯片 AD 转换的偏移误差。

校准是由软件设置 ADCAL=1 来实现初始化。其只能在 ADC 禁用 (ADEN=0) 时完成初始化。在校准期间，ADCAL 位必须保持为 1。当校准完成后，ADCAL 被硬件清 0 。校准完成后，可从 ADC_DR 寄存器的 6:0 位读出校准因子。

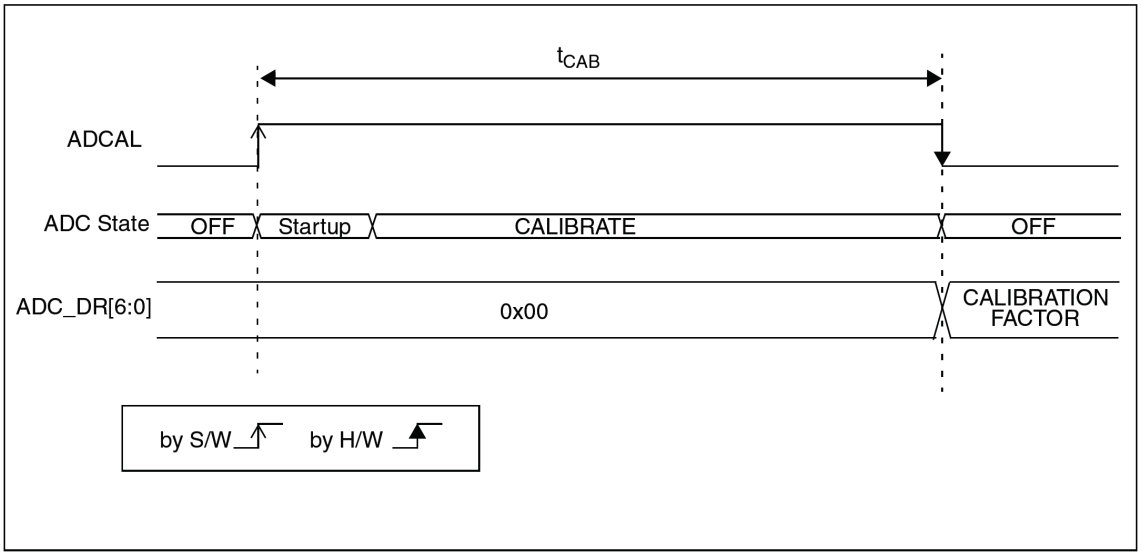
当 ADC 禁用时 (ADEN=0) 校准因子保持原值。然而，若 ADC 长时间禁用，建议在启用 ADC 之前重新做一次 ADC 校准操作。

校准因子在 ADC 每次断电后丢失 (例如当器件从待机模式或由 VBAT 供电模式)。

软件过程：

- 确认 ADEN=0
- 设置 ADCAL=1
- 等待到 ADCAL=0
- 校准因子可从 ADC_DR 寄存器的 6： 0 位中读取。

图 23. ADC 校准



12.4.2 ADC 开关控制 (ADEN, ADDIS, ADRDY)

缺省情况下， 禁用 ADC 模块且处于断电模式 (ADEN=0).
如图 24 所示， 在 ADC 开始精确转换前需要一个稳定时间 tSTAB。

两个控制位用于开启或关闭 ADC：

- 设置 ADEN=1 开启 ADC。当 ADC 模块准备好时 ADRDY 标志置 1。
- 设置 ADDIS=1 来关闭 ADC，并使 ADC 处于断电模式。当 ADC 模块全关断后，硬件自动清除 ADEN 和 ADDIS 位。

ADC 转换也可由设置 SWSTART=1 来启动 (参见 12.5 章节：外部触发和触发极性转换) 或由外部触发事件来触发启动 (若触发启动开启)。

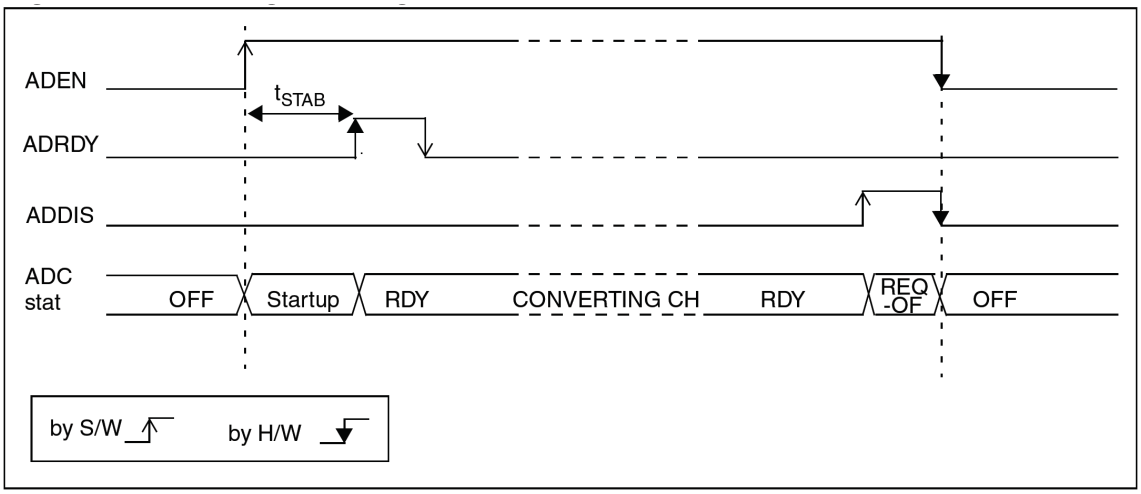
下列是开启 ADC 的过程：

- 在 ADC_CR 寄存器中设置 ADEN=1。.
- 等待直到 ADC_ISR 寄存器中的 ADRDY=1 (当 ADC 启动后 ADRDY 置位)。若 ADRDYIE=1(ADC_IER 寄存器中) 设置 ADC 准备好的中断使能时，也可用中断来处理 ADC 准备好。

以下为禁用 ADC 的过程：

- 检查 ADC_CR 寄存器中的 ADSTART 是否为 0 以确保 ADC 不在转换过程中。若需要，可对 ADC_CR 寄存器中的 ADSTP 置 1 来停止正在进行的 ADC 转换，并等待 ADSTP 被硬件清 0 (清 0 表示转换停止完成)。
- 设置 ADC_CR 寄存器中的 ADDIS=1。
- 若应用要求，则可等待 ADC_CR 寄存器中的 ADEN 位为 0，其表明 ADC 模块完全关闭 (一旦 ADEN=0 时 ADDIS 自动清 0).

图 24. 开启 / 关闭 ADC



注： 在自动关闭模式 (AUTOFF=1) 电源 - 开 / 关阶段不由软件写 ADEN/ADDIS 控制位来执行。然而，也会产生 ADRDY 中断。

12.4.3 ADC 时钟

ADC 具有双时钟域架构，ADC 时钟 (ADC_CLK) 独立于 APB 时钟 (PCLK)。

ADC_CLK 可由两种可能的时钟源产生。ADC_CLK 时钟源可由 RCC 寄存器的选择来产生 (参见 7 章节：复位与时钟控制 (RCC))：

- 选项 1: 专用的 14 MHz 内部振荡器
- 选项 2: PCLK 时钟 /2 或 /4 (最大不能超过 14MHz 的 ADC_CLK)

选项 1 的优势是 ADC 一直具有最佳的 ADC 时钟频率 (14MHz) 无论 HCLK/PCLK 时钟方案是如何选择的。它还提供了 ADC 低功耗自关断模式 (为了节能，ADC 可自动打开 / 关闭内部的 14MHz 振荡器) 的能力。

当 ADC 由定时器触发且应用在触发事件和启动 ADC 期间无任何不确定 (无抖动) 因素的情况下，选项 2 的好处是避免任意时钟域的同步。
在选持选项 1 的情况下，抖动也是诱导同步的触发信号。为了确保没有抖动的存在，寄存器 ADC_CFGR2 中的 JITOFF_D2 或 JITOFF_D4 必须正确配置。这些位有效的去抖动机制有赖于 PLCLK 的分频比。

表 30. 触发和开始转换之间的延迟

ADC 时钟源	JITOFF_D4 位	JITOFF_D2 位	触发事件和开始转换之间的延迟
专用的 14 MHz 时钟	0	0	延迟的不确定性 (抖动)
PCLK / 2	0	1	延迟是确定的 (无抖动) 等于 2.75 个 ADC 时钟周期
PCLK / 4	1	0	延迟是确定的 (无抖动) 等于 2.625 个 ADC 时钟周期

12.4.4 ADC 配置

软件必须在 ADC 禁止 (ADEN 必须为 0) 的情况下改写 ADC_CR 寄存器中的 ADCAL 和 ADEN 位。

软件必须在 ADC 开启且没有关闭请求挂起 (即 ADEN=1 且 ADDIS=0) 的情况下改写 ADC_CR 寄存器中的 ADSTART 和 ADDIS 位。

对于以下的这些位 ADC_IER、ADC_CFGRi、ADC_SMPR、ADC_TR、ADC_CHSELR 和 ADC_CCR 寄存器，软件必须在 ADC 开启 (ADEN = 1) 且无转换期间 (ADSTART = 0) 的情况下才能进行改写。

软件必须在 ADC 开启且无挂起请求 (ADSTART = 1 和 ADDIS = 0) 的情况下改写 ADC_CR 寄存器中的 ADSTP 位。

注：　没有硬件保护机制去防止软件修改以上所述操作规则。假如出现了禁止写发生，ADC 会进入一种不确定状态。为了从这种情况下恢复，ADC 必须被关闭 (ADEN=0 且 ADC_CR 寄存器清 0).

12.4.5 通道选择 (CHSEL, SCANDIR)

共有 19 路复用通道：

- 16 个从 GPIO 引脚引入的模拟输入 (ADC_IN0...ADC_IN15)
- 3 个内部模拟输入 (温度传感、内部参考电压、VBAT 通道)

ADC 可以转换一个单一通道或自动扫描一个序列通道。

被转换的通道序列必须在通道选择寄存器 ADC_CHSELR 中编程选择：每个模拟输入通道有专门的一位选择位 (CHSEL0...CHSEL18).

ADC 扫描的通道顺序由 ADC_CFGR1 中 SCANDIR 位的配置来决定：

- SCANDIR=0: 向前扫描 : 从通道 0 到通道 18.
- SCANDIR=1: 回退扫描：从通道 18 到通道 0

温度传感器 , V_{REFINT} 和 V_{BAT} 内部通道
温度传感连接到 ADC_IN16 通道，内部参考电压 V_{REFINT} 连接到 ADC1_IN17 通道。V_{BAT} 连接到 ADC1_IN18 通道 .

12.4.6 可编程采样时间 (SMP)

在启动数模转换之前，ADC 需要在被测电压源和内嵌采样电容间建立一个直接连接。采样时间必须足够长以便输入电压源对内嵌电容充电到输入电压的水平。

可编程有样时间根据输入电压的输入阻抗来调整转换速度。

ADC 采样输入电压所用的 ADC 时钟个数可用 ADC_SMPR 寄存器中的 SMP[2:0] 位来进行修改。

可编程采样时间对所有通道都通用。如有应用需求，则可用软件改变和适应不同通道间的采样时间。

总转换时间计算如下：

t_{CONV} = 采样时间 + 12.5 x ADC 时钟周期

例如：

当 ADC_CLK = 14 MHz 且采样时间为 1.5 ADC 时钟周期：

t_{CONV} = 1.5 + 12.5 = 14 ADC 时钟周期 = 1 µs

ADC 用设置 EOSMP 标志来表明采样阶段的结束。

12.4.7 单次转换模式 (CONT=0)

单次转换模式下，ADC 执行一次序列转换，转换所有被选的通道。当 ADC_CFGR1 寄存器中的 CONT=0 时，ADC 为单次转换模式。ADC 转换可由下述两种方法启动：

- 在 ADC_CR 寄存器中设置 ADSTART 位
- 硬件触发事件

在序列通道的转换中，每次转换完成后：

- 转换的数据结果存放到 16- 位寄存器 ADC_DR 中。
- EOC (转换结束标志) 标志置位
- 若 EOCIE 位置位则产生一个中断。

通道序列转换完成后：

- EOS (序列结束) 标志置位
- 若 EOSIE 位置位则产生一个中断

转换结束后，ADC 停止直到新的触发事件或 ADSTART 重新置位。

注： *若转换单一通道，则可编程一个长度为 1 的一个转换序列。*

12.4.8 连续转换模式 (CONT=1)

在连续转换模式中，当软件或硬件触发事件产生，ADC 执行一个序列转换。转换所有的通道一次且自动重新开始执行相同的序列转换。当寄存器 ADC_CFGR1 中的 CONT=1 时，ADC 选择为连续转换模式。ADC 转换可由下述两种方法启动：

- 在 ADC_CR 寄存器中设置 ADSTART 位
- 硬件触发事件

在序列通道的转换中，每次转换完成后：

- 转换的数据结果存放到 16- 位寄存器 ADC_DR 中。
- EOC (转换结束标志) 标志置位
- 若 EOCIE 位置位则产生一个中断。

通道序列转换完成后：

- EOS (序列结束) 标志置位
- 若 EOSIE 位置位则产生一个中断

一次序列转换结束后，ADC 立即重新转换相同的序列通道。

注：
1. 若转换单一通道，则可编程一个长度为 1 的一个转换序列。
2. 让 ADC 同时处于断续转换模式和连续转换模式是不可能的事情，在这种情况下 (DISCEN=1，CONT=1)，其表现为连续模式禁止 (即为单次转换模式)

12.4.9 启动转换 (ADSTART)

软件用设置 ADSTART=1 启动 ADC 转换。
当 ADSTART 设置，则转换：

- 当 EXTEN=0x0(软件触发) 时，立即开始
- 当 if EXTEN ≠ 0x0 时，在下一个所选择的活动边沿硬件触发

ADSTART 位也用于说明目前 ADC 转换操作是否正在进行。当 ADC 处于空闲时，该位可重新配置为 ADSTART=0 。

ADSTART 位可由硬件清除。

- 单次转换模式由软件触发 (CONT=0，EXTSEL=0x0)
  - 在序列转换结束后 (EOS=1)
- 在所有的情况下 (CONT=X，EXTSEL=X)
  - 在软件调用并执行 ADSTP 过程后 (有关停止当前转换内容请参见 12.4.11 章节)

注：
1. 在连续模式 (CONT=1) 下，ADSTART 位不能由 EOS 引发的硬件清除，其原因是自动重新开始序列转换。
2. 当硬件触发选择为单次转换模式 (CONT=0 and EXTSEL ≠ 0x00), 则 当 EOS 标志设置后，ADSTART 不会被硬件清 0 。这就避免了需要软件重新设置 ADSTART 位且要确保无硬件触发事件错过。

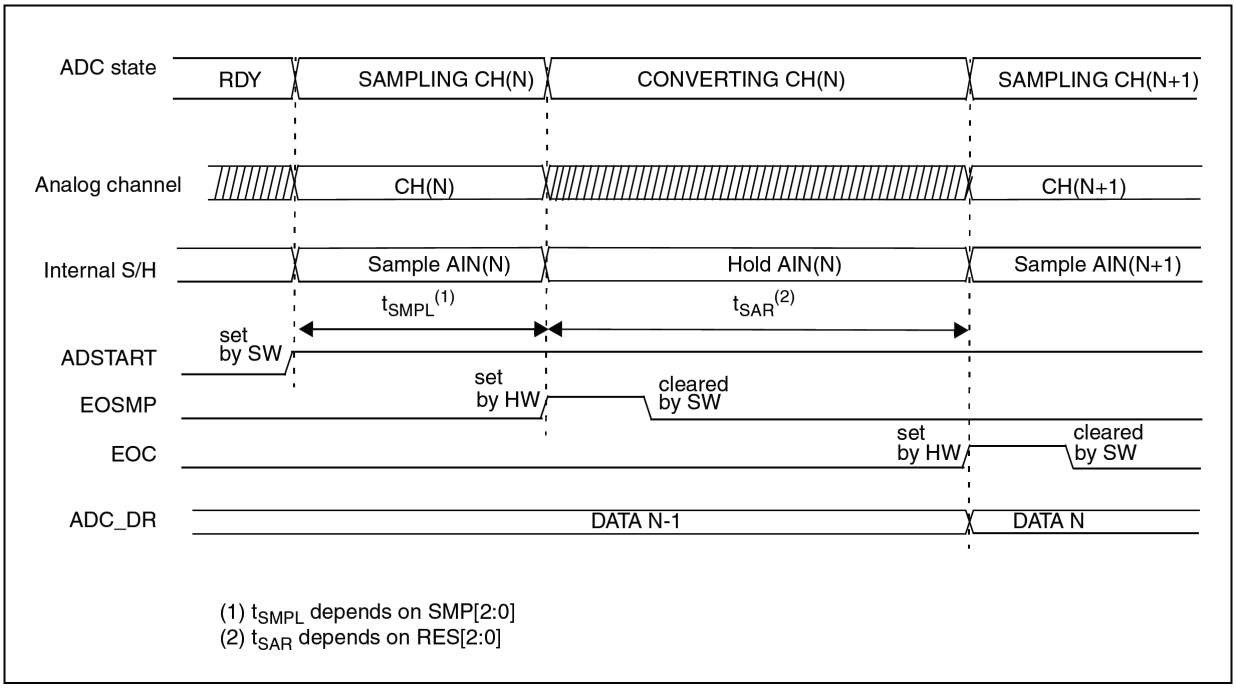
12.4.10 时序

转换所用的时间由启动转换时间和与转换分辨率有关的逐次逼近时间组成。

$t_{ADC} = t_{SMPL} + t_{SAR} = [\ 1.5\ _{|min} + 12.5\ _{|12bit}\ ] \times t_{ADC\_CLK}$

$t_{ADC} = t_{SMPL} + t_{SAR} = 107.1\ ns\ _{|min} + 892.8\ ns\ _{|12bit} = 1\ \mu s\ _{|min}$ (for f_{ADC_CLK} = 14 MHz)

图 25. 模数转换时序



12.4.11 停止当前转换 (ADSTP)

用软件设置 ADC_CR 寄存器中的 ADSTP=1 可以停上当前正在进行的转换。

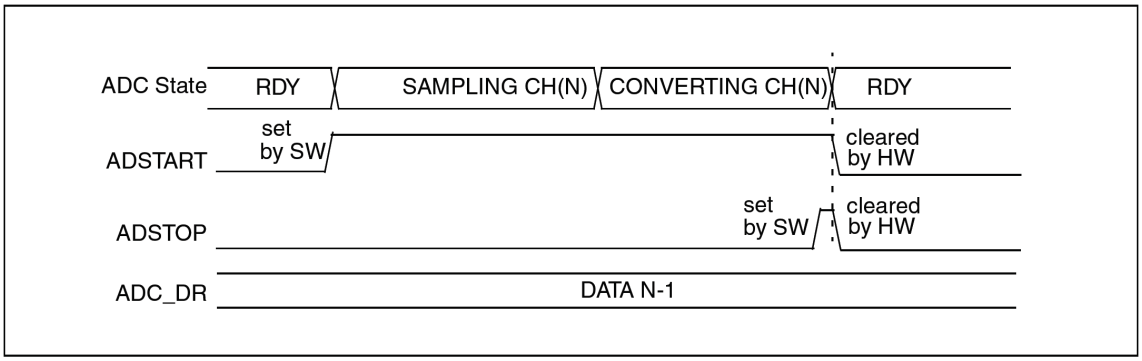
中止会复位 ADC 的操作并让 ADC 进入空闲状态，为下次转换作好准备。

当 ADSTP 由软件设置为 1，任何当前的转换中止且转换结果丢弃 (ADC_DR 寄存器不用当前的转换值进行更新)。

扫描序列也被中止并复位 (即重新启动 ADC 时会用新的序列进行转换)

一旦结束该过程，ADSTP 和 ADSTART 位都由硬件清 0 。

图 26. 停止当前的转换



12.5 转换的外部触发和触发极性 (EXTSEL, EXTEN)

一次转换或一个序列的转换可由软件或外部事件 (例如：定时器捕、输入引脚) 触发。若 EXTEN[1:0] ≠ “0b00”，则外部事件在其所选择的极性上可以用于触发转换。当软件设置 ADSTART=1 时，触发选择有效。

当正在进行 ADC 转换时，任何硬件触发都会被忽略。
当 ADSTART=0 时，任何硬件触发都会忽略。
表 31 给出 EXTEN[1:0] 值与其相对应的极性

表 31. 配置触发极性

源	EXTEN[1:0]
触发检测禁止	00
在上升沿时检测	01
在下降沿时检测	10
在上升沿及下降沿都检测	11

注： *在转换时外部触发极性不能改变。*

EXTSEL[2:0] 控制位用于选择可触发转换的事件。
表 32 给出了规则转换可能的外部触发。
软件源触发事件可由设置 ADC_CR 寄存器中的 ADSTART 位来产生。

表 32. 外部触发

Name	Source	Type	EXTSEL[2:0]
EXT0	TIM1_TRGO	片内定时器产生的内部信号	000
EXT1	TIM1_CC4		001
EXT2	TIM3_TRGO		010
EXT3	TIM1_TRGO		011
EXT4	TIM15_TRGO		100
EXT5	保留		101
EXT6	保留		110
EXT7	保留	外部引脚	111

注： 在转换时外部触发源不能改变。

12.5.1 断续模式 (DISCEN)

该模式由设置 ADC_CFGR1 寄存器中的 DISCEN 位来开启。

在这个模式 (DISCEN=1) 下，需要硬件或软件的触发事件去启动定义在一个序列中的每次转换。相反，DISCEN=0 时，一个硬件或软件的触发事件就可以启动定义在一个序列中的所有转换。

例如：

- DISCEN=1, 需要转换的通道为： 0, 3, 7, 10
  - 1st 触发：通道 0 被转换且一个 EOC 事件产生
  - 2nd 触发：通道 3 被转换且一个 EOC 事件产生
  - 3rd 触发：通道 7 被转换且一个 EOC 事件产生
  - 4th 触发：通道 10 被转换且产生 EOC 和 EOS 事件
  - 5th 触发：通道 0 被转换且一个 EOC 事件产生
  - 6th 触发：通道 3 被转换且一个 EOC 事件产生
  - ...
- DISCEN=0, 需要转换的通道为： 0, 3, 7, 10
  - 1st 触发：整个完整的序列转换：依次为通道 0, 3, 7 和 10。
    每次转换产生一个 EOC 事件，到最后一通道还产生一个 EOS 事件。
  - 任何触发事件都会重新开始完整的序列转换。

注： 让 ADC 同时处于断续转换模式和连续转换模式是不可能的事情，在这种情况下 (DISCEN=1, CONT=1)，其表现为连续模式禁止。

12.5.2 可编程转换分辨率 (RES) – 快速转换模式

用降低转换分辨率来获取更快的转换时间 (tSAR) 是可行的。转换分辨率可通过设置 ADC_CFGR1 寄存器中的 RES[1:0] 来配置为 12, 10, 8, 或 6 位模式。当应用不需要高精度数据时，可用低的转换分辨率来加快转换时间。

转换结果也是 12 位宽度且低位补 0 。
低分辨率模式减少逐次逼近的转换时间，如下表所示：

表 33. tSAR 与转换分辨率有关的转换时间

RES[1:0] 位	t_{SAR} (ADC 时钟周期)	t_{SAR} (ns) @ f_{ADC} = 14 MHz	$t_{SMPL\ (min)}$ (ADC 时钟 周期)	t_{ADC}(ADC 时钟周期) (用是小的 . t_{SMPL})	t_{ADC} (µs) @ f_{ADC} = 14 MHz
12	12.5	893 ns	1.5	14	1000 ns
10	11.5	821 ns	1.5	13	928 ns
8	9.5	678 ns	1.5	11	785 ns
6	7.5	535 ns	1.5	9	643 ns

12.5.3 转换结束 , 采样阶段结束 (EOC, EOSMP 标志)

ADC 通知应用每次转换结束 (EOC) 事件。

一旦在 ADC_DR 寄存器中的一个转换数据有效后，ADC 在 ADC_ISR 寄存器中设置 EOC 标志表明转换完成。当 ADC_IER 中的 EOCIE 置为 1 时，则会产生一个 EOC 中断。EOC 标志由软件写 1 清除或读 ADC_DR 寄存器来清除。

ADC 同样在 ADC_ISR 寄存器中给出采样阶段结束标志 EOSMP。EOSMP 标志可写 1 请除。当在 ADC_IER 寄存器中的 EOSMPIE 置为 1 后，则会产生一个 EOSMP 中断。

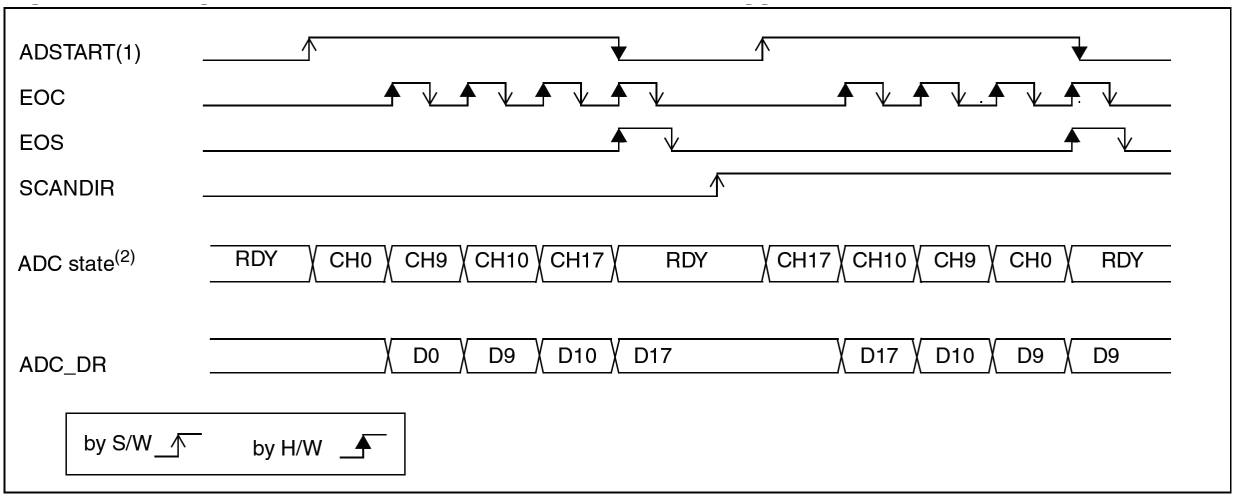
12.5.4 序列转换结束 (EOS 标志)

ADC 通知应用每次序列转换结束 (EOS) 事件。

一旦一个转换序列的最后一个通道转换数据有效后， ADC 在 ADC_ISR 寄存器中设置 EOS 标志。当 ADC_IER 中的 EOSIE 位置 1 时，则会产生 EOS 中断。EOS 标志由软件写 1 清 0 。

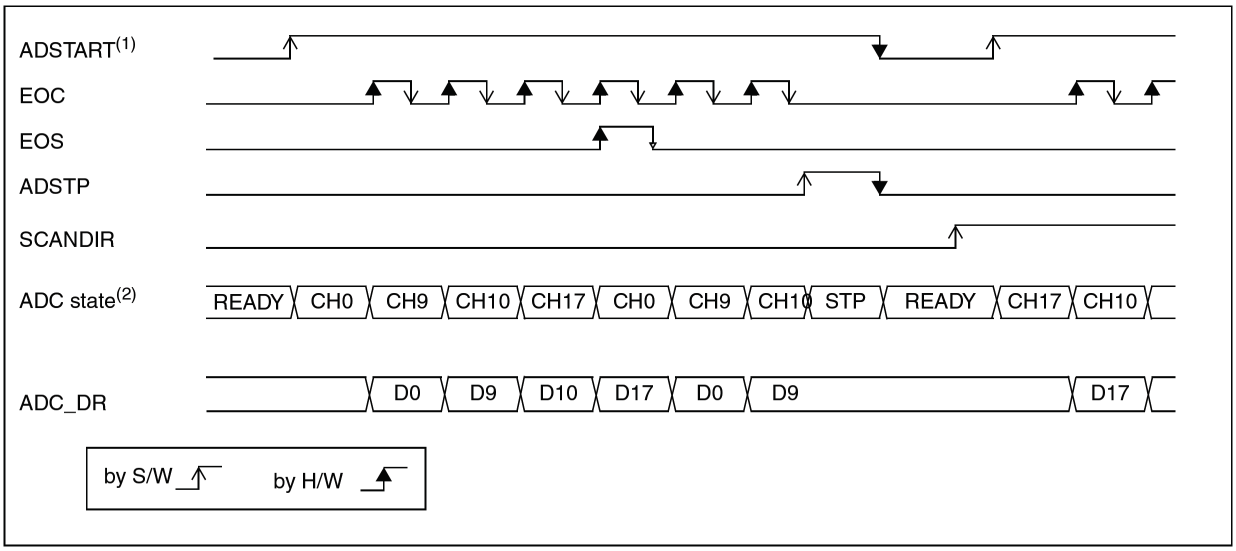
12.5.5 示例时序图 (单次 / 连续 模式 , 硬件 / 软件触发)

图 27. 一个序列的单次转换，软件触发



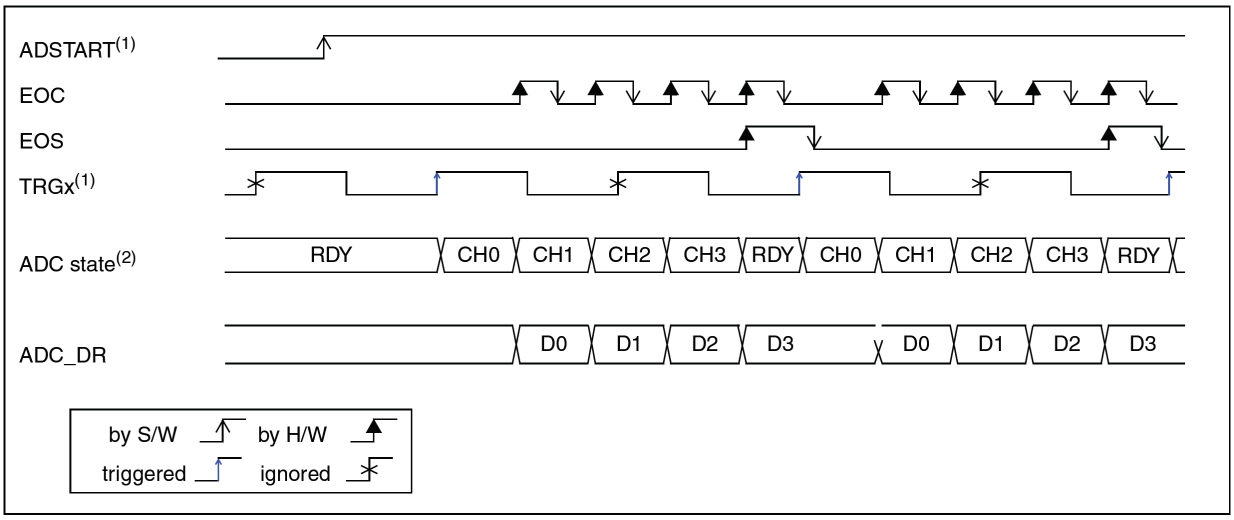
1. EXTEN=0x0, CONT=0
2. CHSEL=0x20601, AUTDLY=0, AUTOFF=0

图 28. 一个序列的连续转换，软件触发



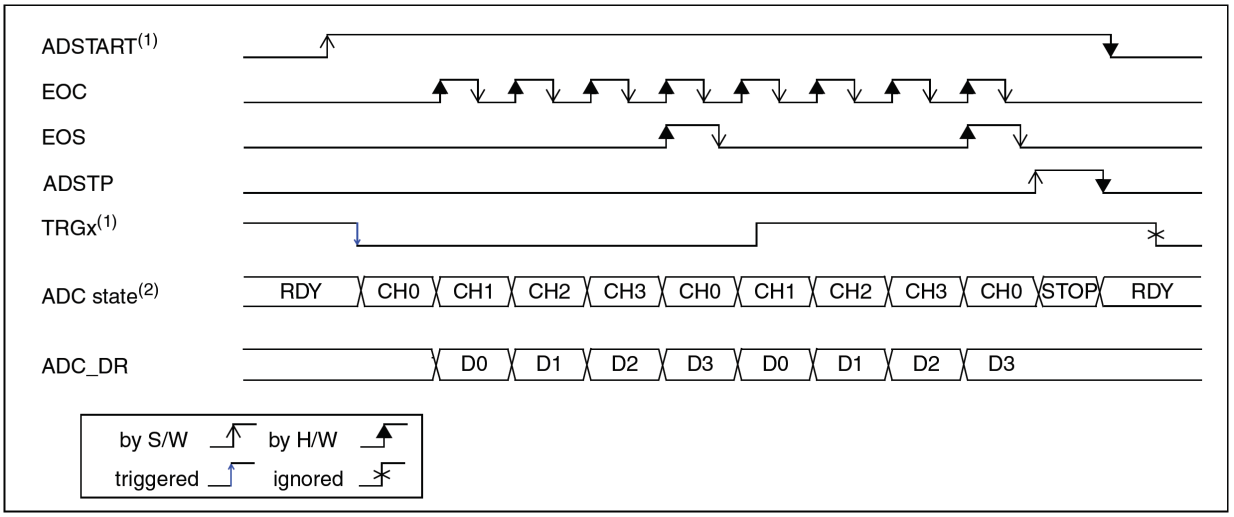
1. EXTEN=0x0, CONT=1,
2. CHSEL=0x20601, AUTDLY=0, AUTOFF=0

图 29. 一个序列的单次转换，硬件触发



1. EXTSEL=TRGx, EXTEN=0x1 (上升沿), CONT=0
2. CHSEL=0xF, SCANDIR=0, AUTDLY=0, AUTOFF=0

图 30. 一个序列的连续转换，硬件触发



1. EXTSEL=TRGx, EXTEN=0x2 (下降沿), CONT=1
2. CHSEL=0xF, SCANDIR=0, AUTDLY=0, AUTOFF=0

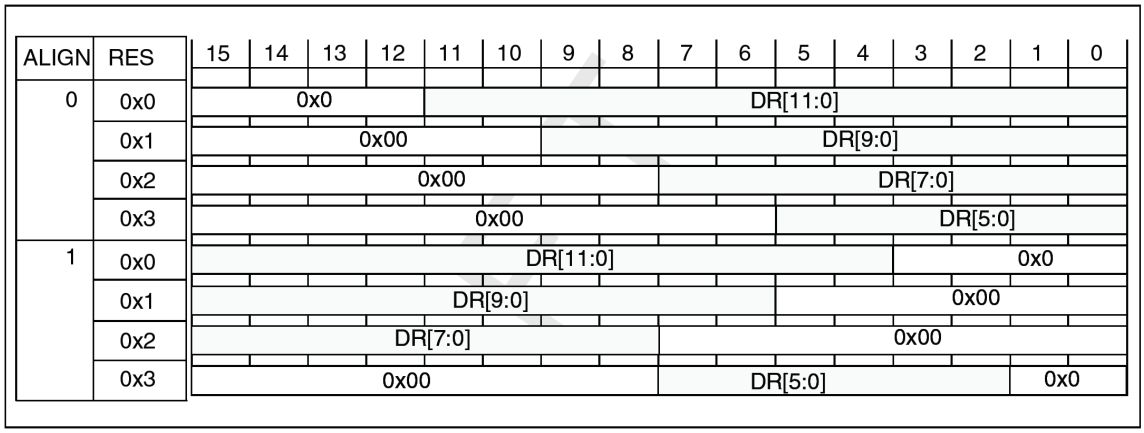
12.6 数据对齐

12.6.1 数据寄存器与数据对齐 (ADC_DR, ALIGN)

在每次转换结束 (当 EOC 事件产生时)，转换的结果数据被存放到 16 位宽 ADC_DR 数据寄存器中。

ADC_DR 数据格式与所配置的数据对齐和转换分辨率有关。The ALIGN bit in the ADC_CFGR1 寄存器中的 ALIGN 位用于选择数据存储的对齐方式，数据可选为右对齐 (ALIGN=0) 或左对齐 (ALIGN=1)。 如图 31 所示：

图 31. 数据对齐与分辨率



12.6.2 ADC 过冲 (OVR, OVRMOD)

ADC 过冲标志 (OVR) 是指一个缓冲区过冲事件，当转换好的数据未被 CPU 或 DMA 及时读取时，另一个转换数据已经有效时，就发生了 ADC 过冲。

若 EOC 还为’1’的情况下，这时一个新的转换已经完成，那么 CPU 就会在 ADC_ISR 寄存器中的 OVR 标志被置位，表明 ADC 过冲。当 ADC_IER 寄存器中的 OVRIE 置位时，产生一个 ADC 过冲中断。

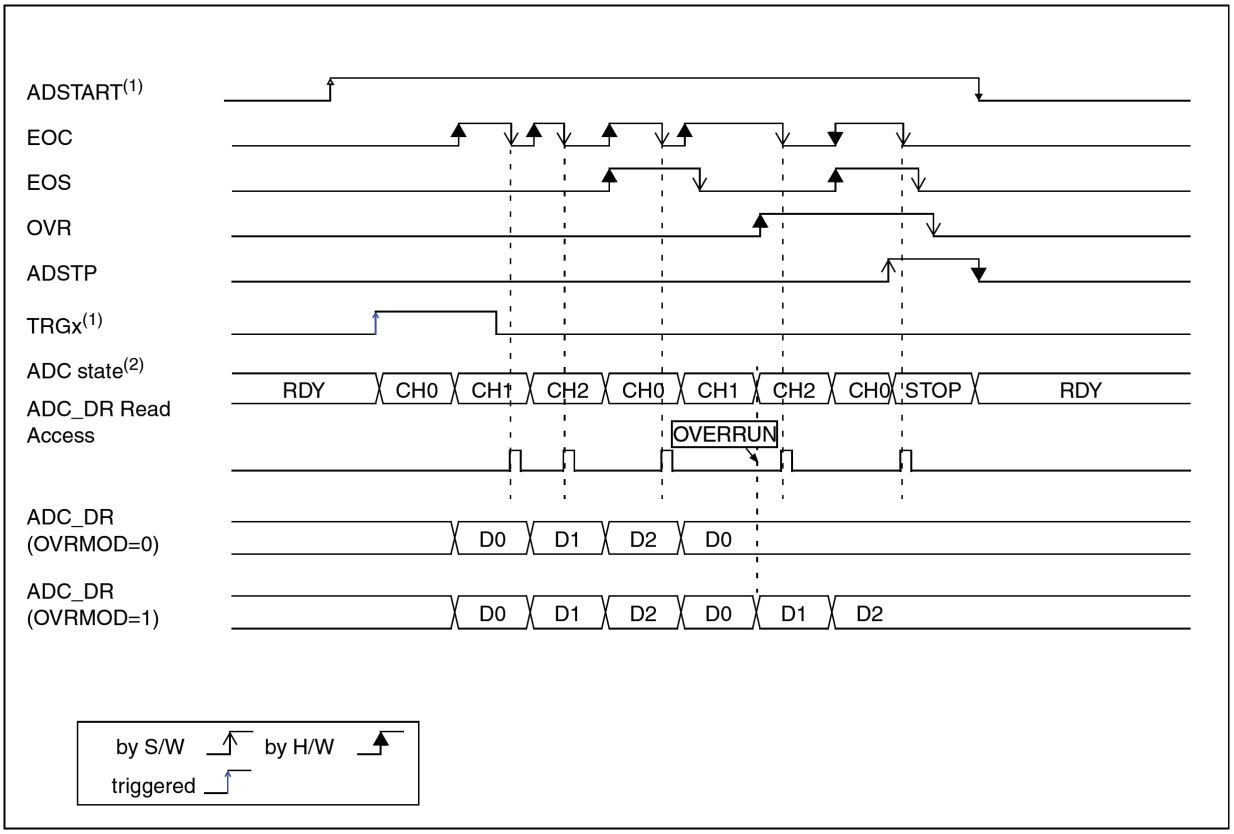
当过冲事件发生时，ADC 会继续操作并且继续转换除非软件决定停止并复位这个序列转换，可用软件设置 ADC_CR 寄存器中的 ADSTP 为 1 来停止 ADC 转换

OVR 标志可用软件写 1 清除。

当发生过冲事件时，可通过对 ADC_CFGR1 寄存器中的 OVRMOD 位来设置 ADC 数据寄存器中的数据是被保持还是被覆盖：

- OVRMOD=0
  - 一个过冲事件保持数据寄存器的值防止被覆盖：老数据被保持新的转换数据丢弃。若 OVR 保持为 1，则后续的转换会被执行但结果都被丢弃。.
- OVRMOD=1
  - 数据寄存器用最后的转换结果覆盖但先前未读的数据丢失，若 OVR 保持为 1，则后续的转换被执行且 ADC_DR 寄存器存放着最后转换的结果值。

图 32. 过冲示例 (OVR)



12.6.3 在无 DMA 参于的情况下管理一序列的转换数据

若 ADC 的转换足够慢，转换序列可由软件来控制。这种情况下，软件应用 EOC 标志及其关联的中断去处理每个转换数据。当每次转换结束时，在 ADC_ISR 寄存器中的 EOC 位置位，此时可读 ADC_DR 寄存器的转换值。ADC_CFGR1 寄存器中的 OVRMOD 位可配为 0 来管理过冲事件。

12.6.4 在无 DAM 和无过冲条件下来管理转换数据

存在着转换一个或多个通道且不用每次转换结果都要读取的应用。这种情况下，OVRMOD 位必须置为 1 且软件应忽略 OVR 标志。当 OVRMOD=1 时，过冲事件不能阻止 ADC 继续转换且 ADC_DR 寄存器中的数据一直为最后转换的数据。

12.6.5 用 DMA 管理转换数据

因为所有通道的转换结果数据存放到一个单一的数据寄存器中，故当转换通道超过 1 个时用 DMA 方式会更有效。这样可以避免丢失存在 ADC_DR 寄存器中的转换结果。

当 DMA 模式开启时 (ADC_CFGR1 寄存器中的 DMAEN =1), 每次转换结束时都会产生一个 DMA 请求。这样就允许把在 ADC_DR 寄存器中的转换数据传送到软件指定的目标地址中。

尽管如此，因 DMA 不能够及时为 DMA 请求服务而产生的过冲 (OVR=1) 时，ADC 就会停止产生 DMA 请求且相应结果是新的转换数据也不会再由 DMA 进行传输 (当 OVR=0 时，会继续传输)。这也可以认为所有传输到 RAM 中的数据都是有效的 (因无效的数据再也不传输了)。

根据 OVRMOD 位的配置，ADC_DR 寄存器中的数据可选择为：保持或覆盖 (参见 12.6.2 章节：ADC 过冲 (OVR, OVRMOD))。

DMA 传输请求会被阻止直到软件清除 OVR 位。

有两种不同的 DMA 模式，其取决于 ADC_CFGR1 寄存器中的 DMACFG 位的配置：

- DMA 一次模式 (one shot mode)(DMACFG=0).
  当 DMA 编程用于传输固定长度的数据时，可选用该模式。
- DMA 循环模式 (DMACFG=1)
  当 DMA 编程为循环模式或双缓冲模式时，可选用该模式。

DMA 一次模式 (DMACFG=0)

在这种模式下，ADC 在每次转换的数据有效时产生一次 DMA 请求。一旦 DMA 已达到最后一个 DMA 传输 (此时产生一个 DMA_EOT 中断，参见 10 章：DMA 控制器有关内容) 时，即使 ADC 转换已重新启动，但 ADC 停止产生 DMA 请求。(产生 DMA_EOT 中断时，下一次的 ADC 转换有可能已开始)

当 DMA 传输完成 (配置在 DMA 控制器中的所有传输已经完成)：

- ADC 数据寄存器的内容冻结
- 任何进行中的转换中止且结果值丢弃
- 不给 DMA 控制器发出新的 DMA 请求。假如仍有 ADC 转换启动，这种方式可避免产生一个 ADC 过冲错误。
- ADC 扫描序列停止并复位
- DMA 停止

DMA 循环模式 (DMACFG=1)

在这种模式下，即使 DMA 达到最后一个 DMA 的传输，ADC 也会在每次转换的数据有效时产生一次 DMA 请求。这允许 DMA 配置为循环模式来处理连续模拟输入数据流。

12.7 低功耗特性

12.7.1 自动延迟转换模式 (AUTDLY)

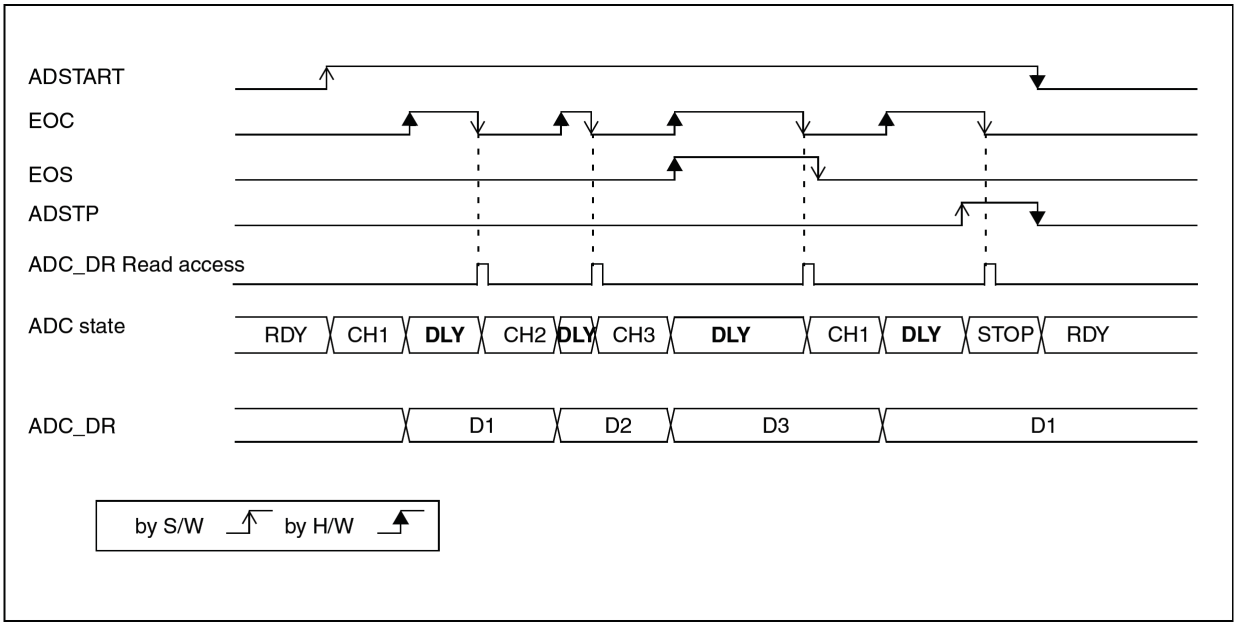
自动延迟转换模式可用于在低速运行时简化软件以及优化应用程序的性能，当然在这种模式下下容易产生 ADC 过冲的情况。

当在 ADC_CFGR1 寄存器中设置 AUTDLY 为 1 时， 一个新的转换只有在刚才的 ADC 数据处理完后 (比如 ADC_DR 寄存器中的数据被读取或 EOC 标志已被清除) 才开始。

这是一种自适应 ADC 速度和自适应系统读取 ADC 数据速度的方法。

注： 当正在转换中或自动延迟产生的情况下，任一硬件产生的触发都会被忽略。

图 33. 自动延迟转换 (连续模式 , 软件触发)



1. EXTEN=0x0, CONT=1
2. CHSEL=0x3, SCANDIR=0, AUTDLY=1, AUTOFF=0

12.7.2 自动关断模式 (AUTOFF)

ADC 含有一个本身电源自动管理的功能，该功能为自关断模式。当设置 ADC_CFGR1 寄存器中的 AUTOFF=1 时，开启该功能。

当 AUTOFF=1 时 , 无转换时 ADC 经常自断电；转换开始 (由软件或硬件触发) 时 ADC 自动唤醒。一个启动时间在触发开始转换与采样之间自动插入。一旦序列转换结束后 ADC 自动关闭。

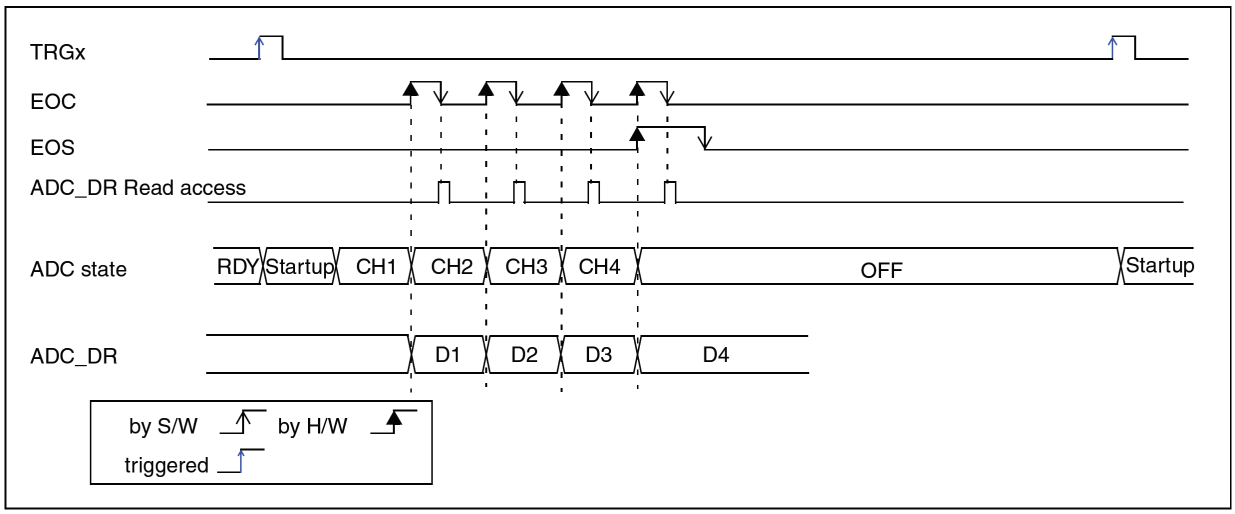
自动关断模式可大大降低应用的功耗， 其适用于需要相对较少转换或转换请求的时间间隔足够长 (如低频的硬件触发) 的应用。

自关掉模式可与自动延迟 (AUTDLY=1) 转换模式联合使用于低频的应用当中。在这种情况下，有望达到 1mA 的节能。

ADC 会自动在转换完成与用读 ADC_DR 寄存器的方法启动 ADC 的期间断电 (参见图 35: 在 AUTDLY=1, AUTOFF=1 下的行为).

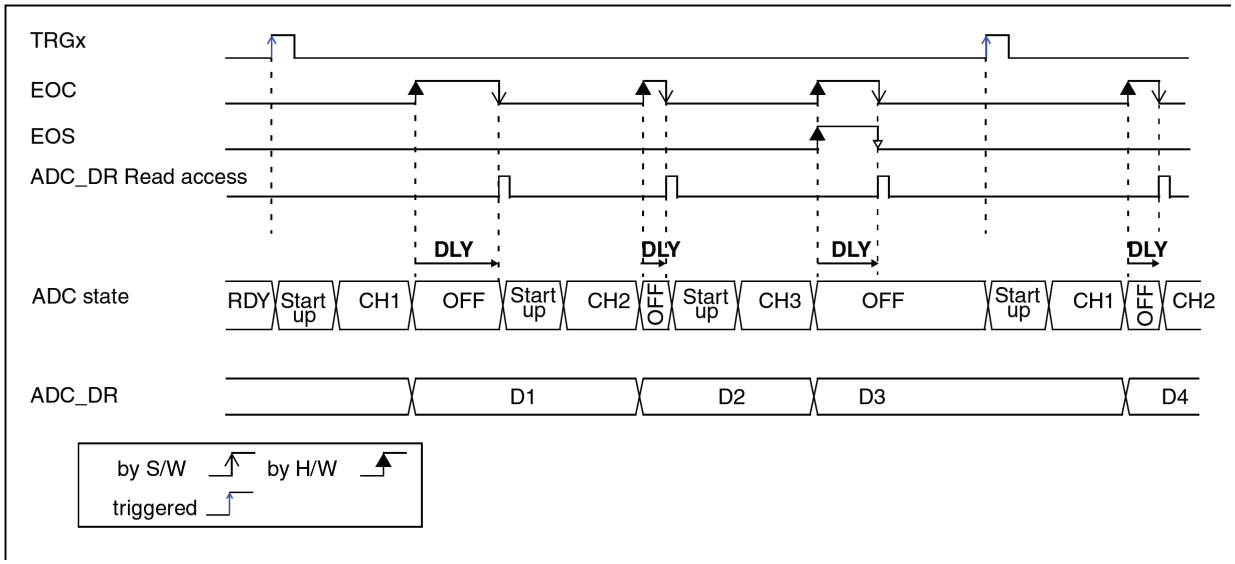
注： 请参考第 7 章节：复位与时钟控制 (RCC) 了解有关管理专用的 14MHz 内部振荡器的内容。ADC 接口可自动切换开启 / 关闭 14 MHz 内部振荡器来节能。

图 34. 在 AUTDLY=0, AUTOFF=1 下的行为



1. EXTSEL=TRGx, EXTEN=0x1 (上升沿), CONT=x, ADSTART=1, CHSEL=0xF, SCANDIR=0, AUTDLY=1, AUTOFF=1

图 35. 在 AUTDLY=1, AUTOFF=1 下的行为



1. EXTSEL=TRGx, EXTEN=0x1 (上升沿), CONT=x, ADSTART=1, CHSEL=0xF, SCANDIR=0, AUTDLY=1, AUTOFF=1

12.8 模拟窗口看门狗 (AWDEN, AWDSGL, AWDCH, AWD_HTR/LTR, AWD)

AWD 模拟看门狗的功能由在 ADC_CFGR1 寄存器中的 AWDEN 位置位来开启。 它可用于监控所选的单一通道或所有使能通道 (参见表 35: 模拟看门狗通道选择) 所配置电压范围 (窗口), 如图 36 所示。

如果模拟电压转换由 ADC 低于低阀值或高于高阀值时，AWD 模拟看门狗的状态位被置位。阀值被编程到最多具有 12 位有效数据的 ADC_HTR 和 ADC_LTR 16- 位寄存器中。模拟看门狗中断可用设置 ADC_IER 寄存器中的 AWDIE 位来使能。

AWD 标志位可用软件写 1 来清除。

当转换的数据分辨率小于 12- 位 (由 DRES[1:0] 位来决定), 被编程阀值的低位必须保持清零，因为内部转换数据的比较都是按左对齐全 12 位的方式进行比较。

表 34 所有可能分辨率下的比较说明

表 34. 模拟看门狗比较

分辨率位 RES[1:0]	模拟看门狗比较值：		说明
	Raw 转换数据 , 左对齐	阀值	
00: 12- 位	DATA[11:0]	LT[11:0] and HT[11:0]	-
01: 10- 位	DATA[11:2],00	LT[11:0] and HT[11:0]	用户必须配置 LT1[1:0] 和 HT1[1:0] 为 “00”
10: 8- 位	DATA[11:4],0000	LT[11:0] and HT[11:0]	用户必须配置 LT1[3:0] 和 HT1[3:0] 为“0000”
11: 6- 位	DATA[11:6],000000	LT[11:0] and HT[11:0]	用户必须配置 LT1[5:0] 和 HT1[5:0] 为“000000”

表 35 给出了如何配置 ADC_CFGR1 寄存器中的 AWDSGL 和 AWDEN 位来使能模拟看门狗在单一通道或在多通道上。

图 36. 模拟看门狗监控区域

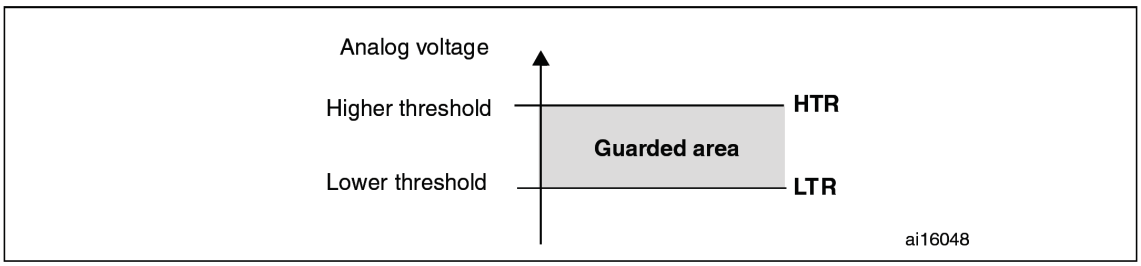


表 35. 看门狗通道选择

由模拟看门狗监控的通道	AWDSGL 位	AWDEN 位
无	x	0
所有通道	0	1
单一通道 (1)	1	1

1. 由 AWDCH[4:0] 位来选择具体通道

12.9 温度传感器

温度传感器可以用来测量器件的接点温度 (TJ)。

温度传感器内部连接到 ADC1_IN16 输入通道，可用于转换传感器的电压值到一个数值。温度传感器的采样时间必须长于 2.2 µs。

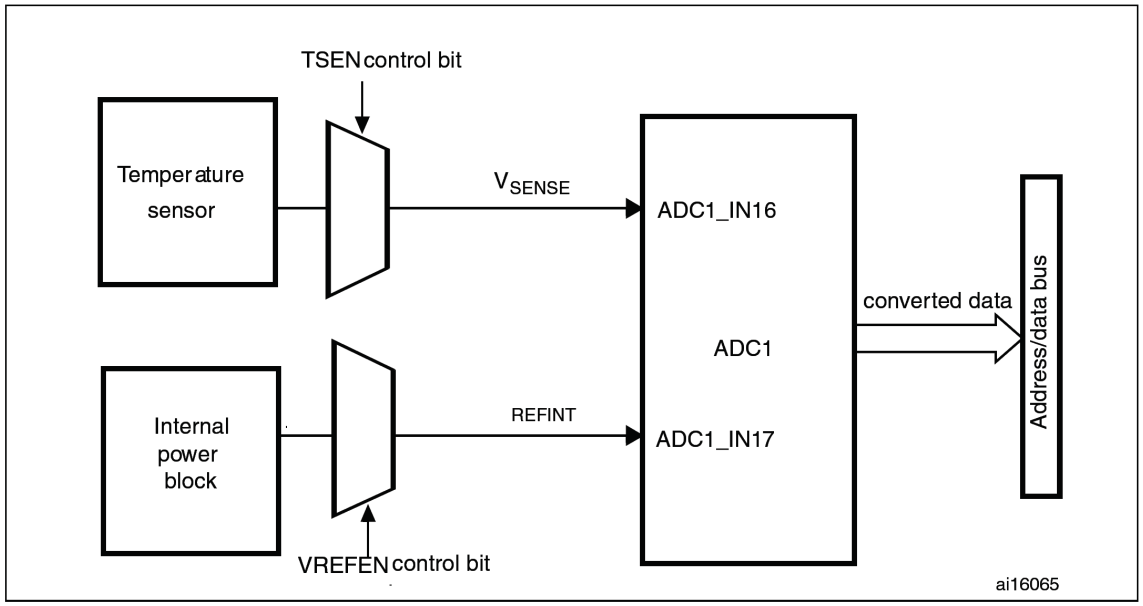
图 37 给出了温度传感器的框图 .
当温度传感器没被使用时，传感器可以置于断电模式。

注： 必须设置 TSVREFE 位来激活两内部通道： ADC1_IN16 (温度传感器) ADC1_IN17 (VREFINT).

主要特性

- 支持的温度范围： –40 到 125℃
- 精度：±2℃ max，精度取决于校准

图 37. 温度传感器和 V_{REFINT} 通道框图



读温度

如何用温度传感器：

1. 选择 ADC1_IN16 输入通道
2. 选择 17.1 µs 的采样时间
3. 在 ADC_CCR 寄存器中设置 TSEN 位用来唤醒从断电模式下的温度传感器
4. 用设置在 ADC_CR 寄存器中的 ADSTART 位 (也可用外部触发) 来启动 ADC 转换
5. 从 ADC_DR 寄存器中读取 VSENSE 转换数据
6. 用下列公式计数温度：

   温度 (℃) = {(V_{25} – V_{SENSE}) / Avg_Slope} + 25

   其中 :

   — V_{25} 为 V_{SENSE} 25℃时的值

   — Avg_Slope 为 V_{SENSE} 和温度曲线的平均斜率值 (单位为 mV/℃或 µV/℃)

参考数据手册有关电气特性章节中的 V_{25} 和 Avg_Slope 的值。

注： 传感器从断电模式下唤醒时到能正确输出 V_{SENSE} 的值要有一个启动时间，ADC 从上电后启动也有一个启动时间，若要减少这个延时，则需要在同一时间设置 ADEN 和 TSEN 位。

12.10 电池电压监测

在 ADC_CCR 寄存器中的 VBATEN 位置位允许应用监测从 V_{BAT} 引脚进来的后备电池电压。由于 V_{BAT} 电压可能比 V_{DDA} 高，为了确保 ADC 的正确操作， V_{BAT} 引脚内部连接到 2 分压桥。当 VBATEN 置 1 时该桥自动开启，连接 V_{BAT}/2 到 ADC1_IN18 输入通道。因此，转换后的数值为 V_{BAT} 电压的一半。为了防止不必要的电池能量消耗，推荐仅在需要转换电池电压时才打开 2 分压桥。

12.11 ADC 中断

ADC 中断可由以下任一事件产生：

- ADC 上电，当 ADC 准备好 (ADRDY 标志)
- 任何一次的转换结束 (EOC 标志)
- 序列转换结束 (EOS 标志)
- 当模拟看门狗检测发生 (AWD 标志)
- 当采样阶段结束发生 (EOSMP 标志)
- 当数据过冲发生 (OVR 标志)

独自的中断使能位用于灵活设置 ADC 中断

表 36. ADC 中断

中断事件	事件标志	使能控制位
ADC 准备好	ADRDY	ADRDYIE
转换结束	EOC	EOCIE
序列转换结束	EOS	EOSIE
模拟看门狗状态置位	AWD	AWDIE
采样阶段结束	EOSMP	EOSMPIE
过冲	OVR	OVRIE

12.12 ADC 寄存器

有关寄存器描述的缩写请参考 1.1 章节内容。

12.12.1 ADC 中断和状态寄存器 (ADC_ISR)

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD	Res.	Res.	OVR	EOS	EOC	EOSMP	ADRDY
								r_w1			r_w1	r_w1	rc_w1	r_w1	r_w1

位 31:8 保留，必须设置为复位值 .

位 7 AWD: 模拟看门狗标志

当转换后的电压超过 ADC_LTR 和 ADC_HTR 寄存器编程设定的电压时，该位由硬件置位。用软件对该位写 1 请除。

0: 无模拟看门狗事件产生 (或该事件标志由软件获取并清零)

1: 产生了模拟看门狗事件

位 6:5 保留，必须设置为复位值 .

位 4 OVR: ADC 过冲

当过冲产生时该位由硬件置位，置位说明新的转换结束但 EOC 位还是为 1 。该位可由软件写 1 请零。

0: 无过冲事件产生 (或该事件标志由软件获取并清零)

1: 产生了过冲事件

位 3 EOS: 序列转换结束标志

由 CHSEL 位所选的通道序列转换结束后，该位由硬件置位。其由软件对该位写 1 请零

0: 序列转换未完成 (或该事件标志由软件获取并清零)

1: 序列转换完成

位 2 EOC: 转换结束标志

当每个通道新转换结果有效时 (存放在 ADC_DR 中) 该位由硬件置位。可由软件对该位写 1 清零或读取 ADC_DR 寄存器来清零。

0: 通道转换未结束 (或该事件标志由软件获取并清零或由读 ADC_DR 寄存器清零)

1: 通道转换结束

位 1 EOSMP: 采样结束标志

在转换期间的采样阶段结束时该位由硬件置 1 。

0: 不在采样结束阶段 (或该事件标志由软件获取并清零)

1: 达到采样阶段结束条件

位 0 ADRDY: ADC 准备好

This bit is set by hardware after the ADC has been enabled (bit ADEN=1) and when the ADC reaches a state where it is ready to accept conversion requests.

该位用软件写 1 清零。

0: ADC 未准备好 (或该标志事件由软件获取并清零)

1: ADC 已准备好开始转换

12.12.2 ADC 中断使能寄存器 (ADC_IER)

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD IE	Res.	Res.	OVRIE	EOSIE	EOCIE	EOSMP IE	ADRDY IE
								rw			rw	rw	rw	rw	rw

位 31:8 保留，必须设置为复位值 .

位 7 AWDIE: 模拟看门狗中断使能

该位由软件设置和清除来开启 / 关闭模拟看门狗中断。

0: 模拟看门狗中断禁止

1: 模拟看门狗中断使能

注：只有当 ADSTART=0 时 (确定无进行中的转换) 才允许改写这些位。

位 6:5 保留，必须设置为复位值 .

位 4 OVRIE: 过冲中断使能

该位由软件设置和清除来开启 / 关闭过冲中断。

0: 过冲中断关闭

1: 过冲中断开启。当 OVR 位置位时产生中断。

注：只有当 ADSTART=0 时 (确定无进行中的转换) 才允许改写这些位。

位 3 EOSIE: 序列转换结束中断使能

该位由软件设置和清除来开启 / 关闭序列转换结束中断

0: EOS 中断关闭

1: EOS 中断开启。当 EOS 置位时产生中断。

注：只有当 ADSTART=0 时 (确定无进行中的转换) 才允许改写这些位。

位 2 EOCIE: 转换结束中断使能
该位由软件设置和清除来开启 / 关闭转换结束中断。
0: EOC 中断关闭
1: EOC 中断开启。当 EOC 置位时产生中断。
注：只有当 ADSTART=0 时 (确定无进行中的转换) 才允许改写这些位。

位 1 EOSMPIE: 采样结束中断使能
该位由软件设置和清除来开启 / 关闭采样阶段结束中断。
0: EOSMP 中断关闭 .
1: EOSMP 中断开启。当 EOSMP 置位时产生中断。
注：只有当 ADSTART=0 时 (确定无进行中的转换) 才允许改写这些位。

位 0 ADRDYIE: ADC 准备好中断使能
该位由软件设置和清除来开启 / 关闭 ADC 准备好中断
0: ADRDY 中断关闭
1: ADRDY 中断开启。当 ADRDY 置位时产生中断。
注：只有当 ADSTART=0 时 (确定无进行中的转换) 才允许改写这些位。

12.12.3 ADC 控制寄存器 (ADC_CR)

偏移地址：0x08
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AD CAL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AD STP	Res.	AD START	AD DIS	AD EN
											rs		rs	rs	rs

位 31 ADCAL: ADC 校准
该位由软件设置来启动 ADC 校准。当校准完成后，由硬件清零。.
0: 校准完成
1: 写 1 时校准 ADC，读为 1 时意味着校准进行中。
注： 只有在 ADC 禁止下 (ADCAL=0, ADSTART=0, ADSTP=0, ADDIS=0 and ADEN=0) 才允许软件设置 ADCL。

位 30:5 保留，必须保持为复位值 .

位 4 ADSTP: ADC 停止转换命令

该位由软件设置来停止和丢弃正在进行中的转换。

当转换停止结束时，该位由硬件清零且 ADC 已准备好接受新的转换命令。

0: 不发 ADC 停止转换命令

1: 写 1 用来停止 ADC，读为 1 时表明 ADSTP 命令正在执行中。

注：只有当 ADSTART=1 和 ADDIS=0(ADC 开启且可能正在转换，但无禁止 ADC 挂起的请求) 软件才能对该位进行设置。

位 3 保留，必须保持为复位值 .

位 2 ADSTART: ADC 开始转换命令

该位由软件设置来启动 ADC 转换。一次转换可由立即启动 (由软件配置) 或硬件触发产生 (硬件触发配置) 两种方式来启动，启动方式由 EXTEN[1:0] 位的配置来决定。

其位由硬件清零：

- 在单次转换模式中，当选择为软件触发 (EXTSEL=0x0) 时：序列转换结束 (EOS 置位) 后该位清零。
- 当执行完 ADSTP 命令后， 同时 ADSTP 位由硬件清零。

0: 无进行中的 ADC 转换。

1: 写 1 开始 ADC 转换。读为 1 表明 ADC 正在进行转换。

注：只有当 ADEN=1 和 ADDIS=0(ADC 开启且可能正在转换，但无禁止 ADC 挂起的请求) 时才允许软件设置 ADSTART 位。

位 1 ADDIS: ADC 禁止命令

该位由软件设置来禁止 ADC (ADDIS 命令) 并让 ADC 处于掉电状态 (关断状态)。

一旦 ADC 有效关闭 (ADEN 同时被硬件清零) 后由硬件清除该位。.

0: 无 ADDIS 命令进行中

1: 写 1 为关闭 ADC。读为 1 时表明 ADDIS 命令正在执行中。

注：只有当 ADEN=1 和 ADSTART=0 (确信无进行中的转换) 时才允许软件设置 ADDIS 位。

位 0 ADEN: ADC 使能命令

由软件设置该位来使能 ADC。一旦 ADRDY 标志置为 1 时表明 ADC 可供使用了。

执行 ADDIS 命令后，ADC 关断且该位被硬件清零。

0: ADC 禁用 (关断状态)

1: 写 1 来使能 ADC.

注：只有在 ADC_CR 寄存器所有位为 0 (ADCAL=0, ADSTP=0, ADSTART=0, ADDIS=0 and ADEN=0) 的情况下，软件才能设置 ADEN 位。

12.12.4 ADC 配置寄存器 1 (ADC_CFGR1)

偏移地址：0x0C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	AWDCH[4:0]					Res.	Res.	AWD EN	AWD SGL	Res.	Res.	Res.	Res.	Res.	DISC EN
	rw	rw	rw	rw	rw			rw	rw						rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AUT OFF	AUT DLY	CONT	OVR MOD	EXTEN[1:0]		Res.	EXTSEL[2:0]			ALIGN	RES[1:0]		SCAN DIR	DMA CFG	DMA EN
rw	rw	rw	rw	rw			rw			rw	rw		rw	rw	rw

位 31 保留，必须设置为复位值 .

位 30:26 AWDCH[4:0]: 模拟看门狗通道选择

这些位由软件设置和清除。它们设置模拟看门狗监视的输入通道。

00000: 由模拟看门狗监视的 ADC 模拟输入通道 0

00001: 由模拟看门狗监视的 ADC 模拟输入通道 1

.....

10011：由模拟看门狗监视的 ADC 模拟输入通道 18

其它值：保留，不会被使用

注：被 AWDCH[4:0] 位所选择的通道必须同样写入 CHSELR 寄存器

注：只有当 ADSTART=0 时 (确定无进行中的转换) 才允许改写这些位。

位 29:24 保留，必须保持为复位值 .

位 23 AWDEN: 模拟看门狗使能位

该位由软件设置和清除。

0: 模拟看门狗关闭

1: 模拟看门狗开启

注：只有当 ADSTART=0 时 (确定无进行中的转换) 才允许改写该位。

位 22 AWDSGL: 在单一通道或所有通道使能看门狗

该位由软件设置和清除来使能由 AWDCH[4:0] 位指定的通道或所有通道。

0: 在所有通道上使能模拟看门狗

1: 在单一通道上使能模拟看门狗

注：只有当 ADSTART=0 时 (确定无进行中的转换) 才允许改写这些位。

位 21:17 保留，必须保持为复位值 .

位 16 DISCEN: 断续模式
该位由软件设置和清除来开启 / 禁止断续模式。
0: 断续模式禁止
1: 断续模式开启
注： 不可能同时开启断续模式和连续模式，在这种情况下 (若 DISCEN=1, CONT=1), 则 ADC 认为连续模式禁止。
注：只有当 ADSTART=0 时 (确定无进行中的转换) 才允许改写这些位。

位 15 AUTOFF: 自动关断模式
该位由软件设置和清除来开启 / 禁止自动关断模式。.
0: 自动关断模式禁止
1: 自动关断模式开启
注：只有当 ADSTART=0 时 (确定无进行中的转换) 才允许改写这些位。

位 14 AUTDLY: 自动延迟转换模式
该位由软件设置和清除来使开启 / 关闭自动延迟转换模式。.
0: 自动延迟转换模式关闭
1: 自动延迟转换模式开启
注：只有当 ADSTART=0 时 (确定无进行中的转换) 才允许改写这些位。

位 13 CONT: 单次 / 连续转换模式
该位由软件设置和清除。 若该位置位，转换为连续模式直到该位清零。
0: 单次转换模式
1: 连续转换模式
注： 不可能同时开启断续模式和连续模式，在这种情况下 (若 DISCEN=1, CONT=1), 则 ADC 认为连续模式禁止。
注：只有当 ADSTART=0 时 (确定无进行中的转换) 才允许改写这些位。

位 12 OVRMOD: 过冲管理模式
该位由软件设置和清除来配置数据过冲管理。
0: 当检测到过冲事件时，ADC_DR 寄存器保持为老数据
1: 当检测到过冲事件时，ADC_DR 寄存器用最后一次的转换数据覆盖
注：只有当 ADSTART=0 时 (确定无进行中的转换) 才允许改写这些位。

位 11:10 EXTEN[1:0]: 外部触发使能和极性选择
这些位可由软件设置和清除来选择外部触发的极性并使能触发器。
00: 硬件触发检测关闭 (可由软件启动转换)
01: 在上升沿进行硬件触发检测
10: 在下降沿进行硬件触发检测
11: 在上升和下降沿进行硬件触发检测
注：只有当 ADSTART=0 时 (确定无进行中的转换) 才允许改写这些位。

位 9 保留，必须保持为复位值 .

位 8:6 EXTSEL[2:0]: 外部触发选择
这些位用于选择触发 ADC 转换的外部事件：
000: 事件 0
001: 事件 1
010: 事件 2
011: 事件 3
100: 事件 4
101: 事件 5
110: 事件 6
111: 事件 7
注：只有当 ADSTART=0 时 (确定无进行中的转换) 才允许改写这些位。

位 5 ALIGN: 数据对齐
该位由软件设置和清除用来选择数据的左或右对齐能见图 31: 数据对齐与分辩率 .
0: 右对齐
1: 左对齐
注：只有当 ADSTART=0 时 (确定无进行中的转换) 才允许改写这些位。

位 4:3 RES[1:0]: 数据分辩率
这些位由软件改写，用于选择 ADC 转换的数据分辩率。
00: 12 位
01: 10 位
10: 8 位
11: 6 位
注：只有当 ADSTART=0 时 (确定无进行中的转换) 才允许改写这些位。

位 2 SCANDIR: 扫描序列方向
该位由软件设置和清除来选择通道序列中的通道扫描方向。
0: 向前扫描 (从 CHSEL0 到 CHSEL16)
1: 向后扫描 (从 CHSEL16 到 CHSEL0)
注：只有当 ADSTART=0 时 (确定无进行中的转换) 才允许改写这些位。

位 1 DMACFG: DMA 配置
该位由软件设置和清除来选择 DMA 模式，只有在 DMAEN=1 时起作用。
0: DMA 单次模式
1: DMA 循环模式
详情请参考 12.6.5 章节：用 DMA 管理转换的数据。
注：只有当 ADSTART=0 时 (确定无进行中的转换) 才允许改写这些位。

位 0 DMAEN: DMA 使能

该位由软件设置和清除来使能 DMA 的请求。允许用 DMA 控制器来自动管理转换的结果数据。详情请参考 12.6.5 章节：用 DMA 管理转换的数据。

0: DMA disabled

1: DMA enabled

注：只有当 ADSTART=0 时 (确定无进行中的转换) 才允许改写这些位。

12.12.5ADC 配置寄存器 2 (ADC_CFGR2)

偏移地址：0x10

复位值：0x0000 8000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
JITOFF_D4	JITOFF_D2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31 JITOFF_DIV4: 移除当 ADC 时钟为 PLCK/4 时在 “触发到启动转换” 延迟中产生的抖动

该位由软件设置或清除。只有当 ADC 时钟为 PLCK/4 时，该位才由软件设置。

当 ADC 时钟是专用的 14 MHz 振荡器时，该位须保持为 0。

当设为 1 时，打开 ADC 移除在触发到启动转换期间抖动的机制。

0: 不消除抖动

1: 当 ADC 时钟由 PCLK/4 驱动时，移除抖动

注： 只有当 ADSTART=0 时 (确定无进行中的转换) 才允许改写该位。

注： 当 JITOFF_DIV4 置位时， JITOFF_DIV2 位必须保持清零。

位 30 JITOFF_DIV2: 移除当 ADC 时钟为 PLCK/2 时在 “触发到启动转换” 延迟中产生的抖动

该位由软件设置或清除。只有当 ADC 时钟为 PLCK/2 时，该位才由软件设置。

当 ADC 时钟是专用的 14 MHz 振荡器时，该位须保持为 0。

当设为 1 时，打开 ADC 移除在触发到启动转换期间抖动的机制。

0: 不消除抖动

1: 当 ADC 时钟由 PCLK/2 驱动时，移除抖动

注：只有当 ADSTART=0 时 (确定无进行中的转换) 才允许改写该位。

注： 当 JITOFF_DIV2 置位时， JITOFF_DIV4 位必须保持清零。

位 29:0 保留，必须保持为复位值 .

12.12.6 ADC 采样时间寄存器 (ADC_SMPR)

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SMP[2:0]		
													rw		

位 31:3 保留，必须保持为复位值 .

位 2:0 SMP[2:0]: 采样时间选择

这位用软件改写，用于选择所选通道的采样时间。

000: 1.5 ADC 时钟周期

001: 7.5 ADC 时钟周期

010: 13.5 ADC 时钟周期

011: 28.5 ADC 时钟周期

100: 41.5 ADC 时钟周期

101: 55.5 ADC 时钟周期

110: 71.5 ADC 时钟周期

111: 239.5 ADC 时钟周期

注： 只有当 ADSTART=0 时 (确定无进行中的转换) 才允许改写这些位。

12.12.7 ADC 看门狗阀值寄存器 (ADC_TR)

偏移地址：0x20

复位值：0x0000 0FFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	HT[11:0]											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	LT[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持为复位值 .

位 27:16 HT[11:0]: 模拟看门狗的高阀值

这些位由软件改写，用来定义模拟看门狗的高阀值。

参见 12.8 章节：模拟看门狗 (AWDEN, AWDSGL, AWDCH, AWD_HTR/LTR, AWD)

注： 只有当 ADSTART=0 时 (确定无进行中的转换) 才允许改写这些位。

位 15:12 保留，必须保持为复位值 .

位 11:0 LT[11:0]: 模拟看门狗低阀值

这些位由软件改写，用来定义模拟看门狗的低阀值。

参见 12.8 章节：模拟看门狗 (AWDEN, AWDSGL, AWDCH, AWD_HTR/LTR, AWD)

注： 只有当 ADSTART=0 时 (确定无进行中的转换) 才允许改写这些位。

12.12.8 ADC 通道选择寄存器 (ADC_CHSELR)

偏移地址：0x28

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CHSEL 17	CHSEL 16
														rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHSEL 15	CHSEL 14	CHSEL 13	CHSEL 12	CHSEL 11	CHSEL 10	CHSEL 9	CHSEL 8	CHSEL 7	CHSEL 6	CHSEL 5	CHSEL 4	CHSEL 3	CHSEL 2	CHSEL 1	CHSEL 0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:18 保留，必须保持为复位值 .

位 17:0 CHSELx: 通道选择

这些位可由软件改写，用来定义所要转换序列的通道。

0: 输入通道 x 不被选为转换通道

1: 输入通道 x 被选为转换通道

注：只有当 ADSTART=0 时 (确定无进行中的转换) 才允许改写这些位。

12.12.9 ADC 数据寄存器 (ADC_DR)

偏移地址：0x40

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持为复位值 .

位 15:0 DATA[15:0]: 转换数据

这些位只读。其包含最后转换通道的转换结果值。该数据左对齐或右对齐数据格式如图 31(数据对课齐与分辩率) 所示。

仅在校准完成时，DATA[6:0] 值为校准因子。

12.12.10 ADC 通用配置寄存器 (ADC_CCR)

偏移地址：0x308

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	VBAT EN	TS EN	VREF EN	Res.	Res.	Res.	Res.	Res.	Res.
							rw	rw	rw						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:25 保留，必须保持为复位值 .

位 24 VBATEN: V_{BAT} 使能

由软件设置和清除该位来打开 / 关闭 V_{BAT} 通道。

0: V_{BAT} 通道关闭

1: V_{BAT} 通道开启

注： 只有当 ADSTART=0 时 (确定无进行中的转换) 才允许改写该位。

位 23 TSEN: 温度传感使能

由软件设置和清除来打开 / 关闭温度传感通道。

0: 温度传感通道关闭

1: 温度传感通道开启

注： 只有当 ADSTART=0 时 (确定无进行中的转换) 才允许改写该位。

位 22 VREFEN: V_{REFINT} 使能

由软件设置和清除来打开 / 关闭 V_{REFINT} 通道。

0: V_{REFINT} 通道关闭

1: V_{REFINT} 通道开启

注： 只有当 ADSTART=0 时 (确定无进行中的转换) 才允许改写该位。

位 21:0 保留，必须设置为复位值 .

12.12.11 ADC 寄存器映像

ADC 寄存器一览表。

表 37. ADC 寄存器映像和复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	ADC_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD	Res.	Res.	OVR	EOS	EOC	EOSMP	ADRDY
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	ADC_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDIE	Res.	Res.	OVRIE	EOSIE	EOCIE	EOSMPIE	ADRDYIE
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	ADC_CR	ADCAL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADSTP	Res.	ADSTART	ADDIS	ADEN
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	ADC_CFGR1	Res.	AWDCH[4:0]					Res.	Res.	AWDEN	AWDSGL	Res.	Res.	Res.	Res.	Res.	DISCEN	AUTOFF	AUTDLY	CONT	OVRMOD	EXTEN[1:0]		Res.	EXTSEL [2:0]			ALIGN	RES [1:0]		SCANDIR	DMACFG	DMAEN
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	ADC_CFGR2	JITOFF_D4	JITOFF_D2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	ADC_SMPR	SMPR [2:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	Reserved	Reserved																															
0x1C	Reserved	Reserved																															
0x20	ADC_TR	Res.	Res.	Res.	Res.	HT[11:0]												Res.	Res.	Res.	Res.	LT[11:0]											
	Reset value	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	Reserved	Reserved																															
0x28	ADC_CHSELR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CHSEL18	CHSEL17	CHSEL16	CHSEL15	CHSEL14	CHSEL13	CHSEL12	CHSEL11	CHSEL10	CHSEL9	CHSEL8	CHSEL7	CHSEL6	CHSEL5	CHSEL4	CHSEL3	CHSEL2	CHSEL1	CHSEL0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C 0x30 0x34 0x38 0x3C	Reserved	Reserved																															
0x40	ADC_DR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATA[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 37. ADC 寄存器映像和复位值 (续)

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x44 ... 0x2FC	Reserved	Reserved																															
0x308	ADC_CCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VBATEN	TSEN	VREFEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

注：Res 为保留

有关寄存器起始地址参见 2.2.2 章节。

13 数字模拟转换器

13.1 DAC 简介

数字 / 模拟转换模块 (DAC) 是 12 位电压输出的数字 / 模拟转换器。DAC 可以配置为 8 位或 12 位模式，也可以与 DMA 控制器配合使用。DAC 在 12 位模式工作时，数据可以被设置成左对齐或右对齐。输入参考电压 VDDA(与 ADC 共享的) 可用。可选择通过输出缓冲器来实现较高的驱动电流。

13.2 DAC 主要特性

- 12 位模式下数据左对齐或者右对齐
- 同步更新功能
- 有 DMA 功能
- DMA 欠载出错检测
- 外部触发转换
- 可编程内部缓存
- 输入参考电压 V_{DDA}

单个 DAC 通道的框图见图 38，表 38 给出了引脚的说明。

图 38. DAC 通道模块框图

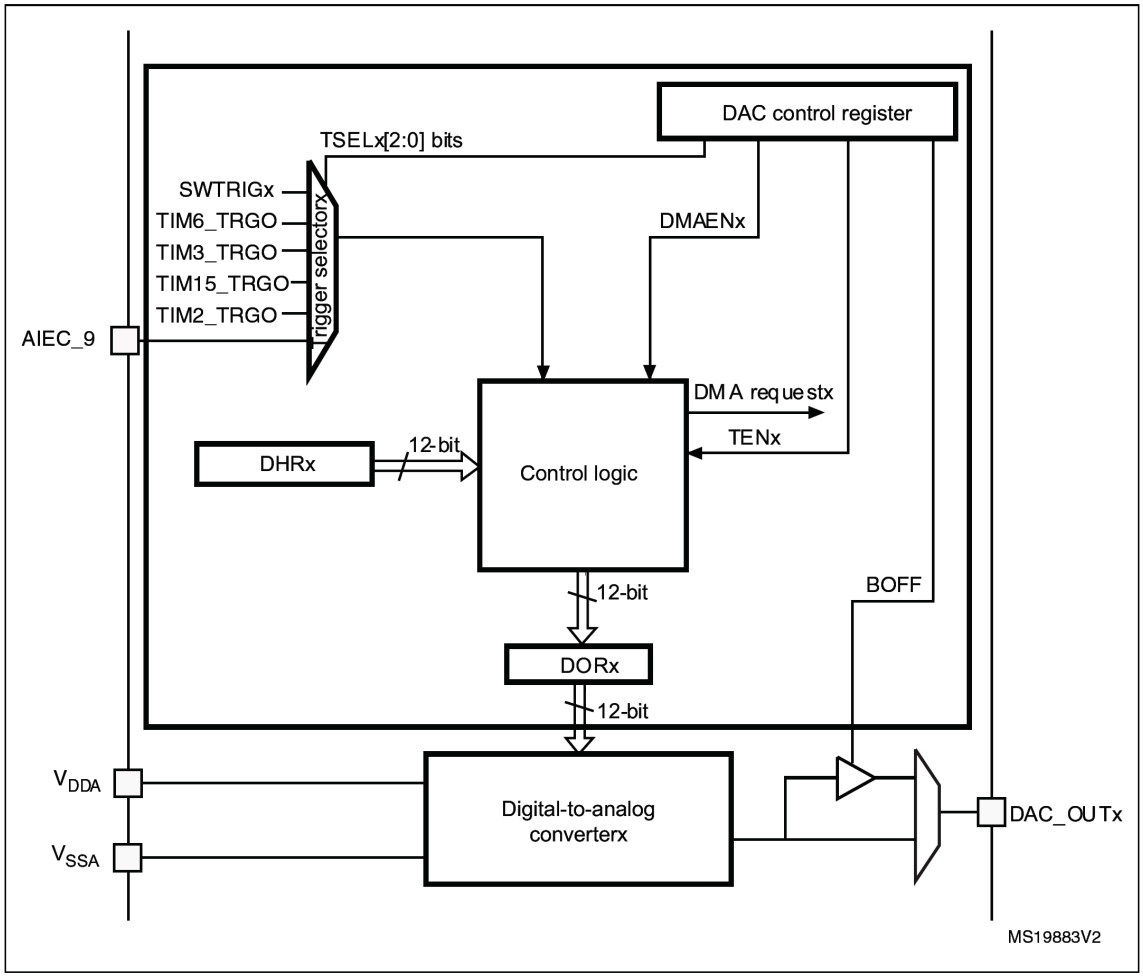


表 38. DAC 引脚

名称	信号类型	注释
V_{DDA}	输入，模拟电源	模拟电源
V_{SSA}	输入，模拟电源地	模拟电源的地线
DAC_OUTx	模拟输出信号	DAC 通道 x 的模拟输出

注：一旦使能 DAC 通道 x，相应的 GPIO 引脚 (PA4 或 PA5) 就会自动与 DAC 的模拟输出相连 (DAC_OUTx)。为了避免寄生的干扰和额外的功耗，PA4 或 PA5 引脚首先应设置成模拟输入 (AIN)。

13.3 DAC 功能描述

13.3.1 使能 DAC 通道

将 DAC_CR 寄存器的 EN 位置’1’即可打开对 DAC 通道的供电。经过一段启动时间 tWAKEUP，DAC 通道即被使能。

注： ENx 位只会使能 DAC 通道 x 的模拟部分，即便该位被置’0’，DAC 通道 x 的数字接口仍然被使能。

13.3.2 使能 DAC 输出缓存

DAC 集成了 1 个输出缓存，可以用来减少输出阻抗，无需外部运放即可直接驱动外部负载。DAC 通道输出缓存可以通过设置 DAC_CR 寄存器的 BOFFx 位来启用或者关闭。

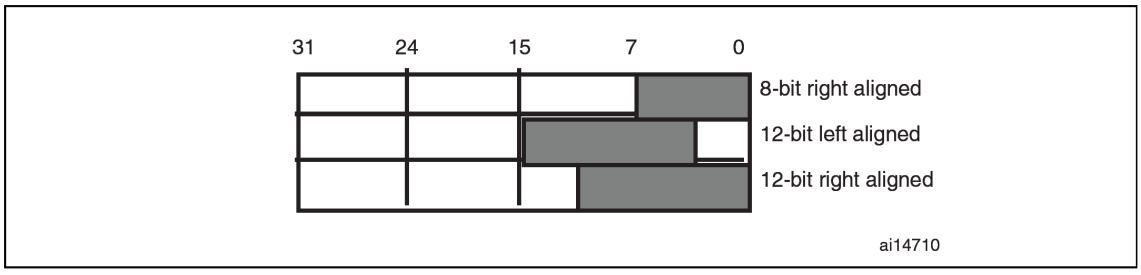
13.3.3 DAC 数据格式

根据选择的配置模式，数据按照下文所述写入指定的寄存器：

- 有 3 种情况 :
  - 8 位数据右对齐：软件须将数据写入寄存器 DAC_DHR8Rx[7:0] 位 (实际是存入寄存器 DHRx[11:4] 位)
  - 12 位数据左对齐：软件须将数据写入寄存器 DAC_DHR12Lx[15:4] 位 (实际是存入寄存器 DHRx[11:0] 位)
  - 12 位数据右对齐：软件须将数据写入寄存器 DAC_DHR12Rx[11:0] 位 (实际是存入寄存器 DHRx[11:0] 位)

根据加载的 DAC_DHRyyyx 寄存器，用户写入的数据被转存到相应的 DHRx 寄存器中 (DHRx 是内部非内存映射的数据保存寄存器 x)。随后，通过软件触发或外部事件触发，DHRx 寄存器的内容或被自动地传送到 DORx 寄存器。

图 39. 单 DAC 通道模式的数据寄存器



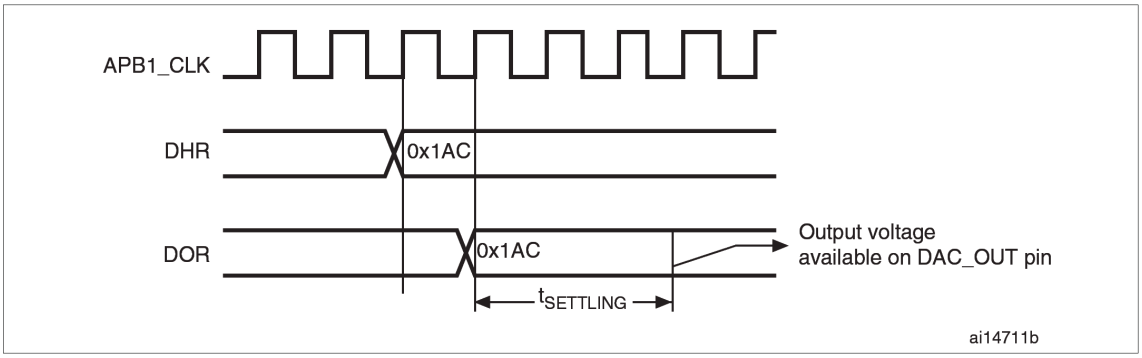
13.3.4 DAC 转换

不能直接对寄存器 DAC_DORx 写入数据，任何输出到 DAC 通道 x 的数据都必须写入 DAC_DHRx 寄存器 (数据实际写入 DAC_DHR8Rx, DAC_DHR12Lx, DAC_DHR12Rx, DAC_DHR8RD, DAC_DHR12LD 或 DAC_DHR12LD 寄存器)。

如果没有选择硬件触发 (寄存器 DAC_CR 的 TENx 位置’0’)，存入寄存器 DAC_DHRx 的数据会在一个 APB1 时钟周期后自动传至寄存器 DAC_DORx。如果选择硬件触发 (寄存器 DAC_CR 的 TENx 位置’1’)，数据传输在触发发生后 3 个 PLCK1 时钟周期后完成。

当 DAC_DHRx 寄存器的数据加载到 DAC_DORx 寄存器，在经过时间 tSETTLING 之后，模拟输出有效，tSETTLING 依电源电压和模拟输出负载的不同会有所变化。

图 40. TEN=0 触发关闭时转换的时间框图



13.3.5 DAC 输出电压

数字输入被线性地转换为模拟输出电压，其范围为 0 到 VDDA。 任一 DAC 通道引脚上的模拟输出电压满足以下关系：

DAC 输出 = V_{DDA} × DOR / 4095

13.3.6 选择 DAC 触发

如果 TENx 控制位被置 1，DAC 转换可以由某一外部事件触发 (定时器计数器、外部中断线)。配置 TSELx[2:0] 控制位可以选择下表 39 的 8 个触发事件之一触发 DAC 转换。

表 39. 外部触发

触发源	类型	TSEL[2:0]
定时器 6 TRGO 事件	来自片上定时器的内部信号	000
定时器 3TRGO 事件		001
Reserved 保留		010
定时器 15 TRG 事件		011
定时器 2 TRGO 事件		100
Reserved 保留		101
AIEC 线路 9	外部引脚	110
SWTRIG 软件触发	软件控制位	111

每次 DAC 接口检测到来自选定的定时器 TRGO 输出，或外部中断线 9 的上升沿，最近存放在寄存器 DAC_DHRx 中的数据会被传送到寄存器 DAC_DORx 中。在 3 个 APB1 时钟周期后，寄存器 DAC_DORx 更新为新值。

如果选择软件触发，一旦 SWTRIG 位置 ’1’，转换即开始。在寄存器 DAC_DHRx 的数据加载到寄存器 DAC_DORx 后，SWTRIG 位由硬件自动清 ’0’。

注： *1 不能在 ENx 为’1’时改变 TSELx[2:0] 位。*

2 如果选择软件触发，数据从寄存器 DAC_DHRx 传送到寄存器 DAC_DORx 只需要一个 APB1 时钟周期。

13.3.7 DMA 请求

每个 DAC 通道都具有 DMA 功能。2 个 DMA 通道可用于服务 DAC 通道的 DMA 请求。

如果 DMAENx 位置’1’，一旦有外部触发 (但不是软件触发) 发生，则产生一个 DMA 请求，然后 DAC_DHRx 寄存器的数据被传送到 DAC_DORx 寄存器。

DMA 欠载

DAC 的 DMA 请求没有队列，所以，在收到第一个外部触发确认之前（第一个请求），如果第二个外部触发到达，然后再没有新的请求发出时，DAC_SR 寄存器的 DMA 通道 x 的欠载标志 DMAUDRx 被置“1”，报告错误状态。DMA 数据传输功能关闭，不再处理 DMA 请求。DAC 通道 x 继续转换旧数据。

软件应通过写“1”清除 DMAUDRx 标志，清除对应的 DMA 通道的 DMAEN 位，并重新初始化 DMA 和 DAC 通道 x，以用正确的方式重新启动数据传输。软件应先修改 DAC 触发转换的频率或减轻 DMA 的工作量，以避免产生新的 DMA 请求。最后，使能 DMA 数据传送和转换触发器来恢复 DAC 转换。

对于每个 DAC 通道，如果相应的 DAC_CR 寄存器的 DMAUDRIEx 位被使能，都会产生一个中断。

13.4 DAC 寄存器

请参考第 31 页第 1.1 节 寄存器描述使用的缩写列表。

必须以字 (32 位) 的方式操作这些外设寄存器。

13.4.1 DAC 控制寄存器 (DAC_CR)

地址偏移 : 0x00

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15		13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	DMAUDRIE1	DMAEN1	Res.	Res.	Res.	Res.	Res.	Res.	TSEL12	TSEL11	TSEL10	TEN1	BOFF1	EN1
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:14 保留。

位 13 DMAUDRIE1: DAC 通道 1 DMA 欠载中断使能
该位由软件设置和清除。
0: DAC 通道 1 DMA 欠载中断关闭
1: DAC 通道 1 DMA 欠载中断使能

位 12 DMAEN1: DAC 通道 1 DMA 使能
该位由软件设置和清除。
0: 关闭 DAC 通道 1 DMA 模式
1: 使能 DAC 通道 1 DMA 模式

位 11:8 保留。

位 7:6 保留。

位 5:3 TSEL1[2:0]: DAC 通道 1 触发器
通道 1 触发选择
该位用于选择 DAC 通道 1 的外部触发事件 .
000: Timer 6 TRGO 事件
001: Timer 8 TRGO 事件
010: Timer 7 TRGO 事件
011: Timer 5 TRGO 事件
100: Timer 2 TRGO 事件
101: Timer 4 TRGO 事件
110: 外部中断线 9
111: 软件触发
注：该位只能在 TEN1= 1(DAC 通道 1 触发使能) 时使用。

位 2 TEN1: DAC 通道 1 触发使能
该位由软件设置和清除，用来使能 / 关闭 DAC 通道 1 的触发。
0：关闭 DAC 通道 1 触发，写入寄存器 DAC_DHRx 的数据在 1 个 APB1 时钟周期后传入寄存器 DAC_DOR1；
1：使能 DAC 通道 1 触发，写入寄存器 DAC_DHRx 的数据在 3 个 APB1 时钟周期后传入寄存器 DAC_DOR1。
注：当选择软件触发时，写入寄存器 DAC_DHRx 的数据只需要 1 个 APB1 时钟周期就可以传入寄存器 DAC_DOR1。

位 1 BOFF1: 关闭 DAC 通道 1 输出缓存
该位由软件设置和清除，用来使能 / 关闭 DAC 通道 1 的输出缓存。
0：使能 DAC 通道 1 输出缓存；
1：关闭 DAC 通道 1 输出缓存。

位 0 EN1: ：DAC 通道 1 使能
该位由软件设置和清除，用来使能 / 关闭 DAC 通道 1。
0：关闭 DAC 通道 1；
1：使能 DAC 通道 1。

13.4.2 DAC 软件触发寄存器 (DAC_SWTRIGR)

地址偏移 : 0x04

复位置 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWTRIG1
															w

位 31:1 保留。

位 0 SWTRIG1: DAC 通道 1 软件触发

该位由软件设置和清除，用来使能 / 关闭软件触发。

0：关闭软件触发；

1：使能软件触发。

注：一旦寄存器 DAC_DHR1 的数据传入寄存器 DAC_DOR1，(1 个 APB1 时钟周期后) 该位由硬件置’0’。

13.4.3 DAC 通道 1 的 12 位右对齐数据保持寄存器 (DAC_DHR12R1)

地址偏移 : 0x08

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DACC1DHR[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:12 保留。

位 11:0 DACC1DHR[11:0]: DAC 通道 1 的 12 位右对齐数据

这些位由软件写入，表示 DAC 通道 1 的 12 位数据。

13.4.4 DAC 通道 1 的 12 位左对齐数据保持寄存器 (DAC_DHR12L1)

地址偏移 : 0x0C

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC1DHR[11:0]												Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				

位 31:16 保留。

位 15:4 DACC1DHR[11:0]: DAC 通道 1 的 12 位左对齐数据
这些位由软件写入，表示 DAC 通道 1 的的 12 位数据 .

位 3:0 保留 .

13.4.5 DAC 通道 1 的 8 位右对齐数据保持寄存器 (DAC_DHR8R1)

地址偏移 : 0x10

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC1DHR[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8 保留。

位 7:0 DACC1DHR[7:0]: DAC 通道 1 的 8 位右对齐数据
这些位由软件写入，表示 DAC 通道 1 的的 8 位数据 .

13.4.6 双 DAC 的 8 位右对齐数据保持寄存器 (DAC_DHR8RD)

地址偏移 : 0x28

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC1DHR[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8 保留 , 必须设为复位值 .

位 7:0 DACC1DHR[7:0]: DAC 通道 1 的 8 位右对齐数据
该位由软件写入，表示 DAC 通道 1 的的 8 位数据。

13.4.7 DAC 通道 1 数据输出寄存器 (DAC_DOR1)

地址偏移 : 0x2C

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DACC1DOR[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

位 31:12 保留 .

位 11:0 DACC1DOR[11:0]: DAC 通道 1 输出数据

只读，表示 DAC 通道 1 的输出数据。

13.4.8 DAC 状态寄存器 (DAC_SR)

地址偏移 : 0x34

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	DMAUDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
		rc_w1													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	DMAUDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
		rc_w1													

位 31:14 保留。

位 13 DMAUDR1: DAC 通道 1 DMA 欠载标志

该位由硬件置“1”，由软件清“0”

0: DAC 通道 1 无 DMA 欠载出错

1: DAC 通道 1 发生 DMA 欠载出错，(DAC 通道 1 选择的触发频率高于 DMA 的响应速度)。

位 12:0 保留。

13.4.9 DAC 寄存器映射

表 40 是 DAC 寄存器的概述

表 40. DAC 寄存器映射和复位值

Address offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	DAC_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAUDRIE1	DMAEN1	Res.	Res.	Res.	Res.	Res.	Res.	TSEL1[2:0]			TEN1	BOFF1	EN1
	Reset value																			0	0							0	0	0	0	0	0
0x04	DAC_SWTRIGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWTRIG1
	Reset value																																0
0x08	DAC_DHR12R1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC1DHR[11:0]											
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x0C	DAC_DHR12L1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC1DHR[11:0]												Res.	Res.	Res.	Res.
	Reset value																																
0x10	DAC_DHR8R1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC1DHR[7:0]							
	Reset value																																
0x14	DAC_DHR12R2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC2DHR[11:0]											
	Reset value																																
0x28	DAC_DHR8RD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC1DHR[7:0]							
	Reset value																																
0x2C	DAC_DOR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC1DOR[11:0]											
	Reset value																																
	Reset value																																
0x34	DAC_SR	Res.	Res.	Res.	Res.	Res.	Res	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAUDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																			0													

寄存器起始地址请参考第 35 页第 2.2.2 节。

14 比较器 (COMP)

14.1 COMP 说明

STM32F0xx 内嵌两个通用比较器 COMP1 和 COMP2，可独立使用（适用所有终端上的 I / O 口），也可与定时器结合使用。它们可用于多种功能，包括：

- 由模拟信号触发的从低功耗模式唤醒
- 模拟信号调理
- 与 DAC 和定时器输出的 PWM 相结合，组成逐周期的电流控制回路

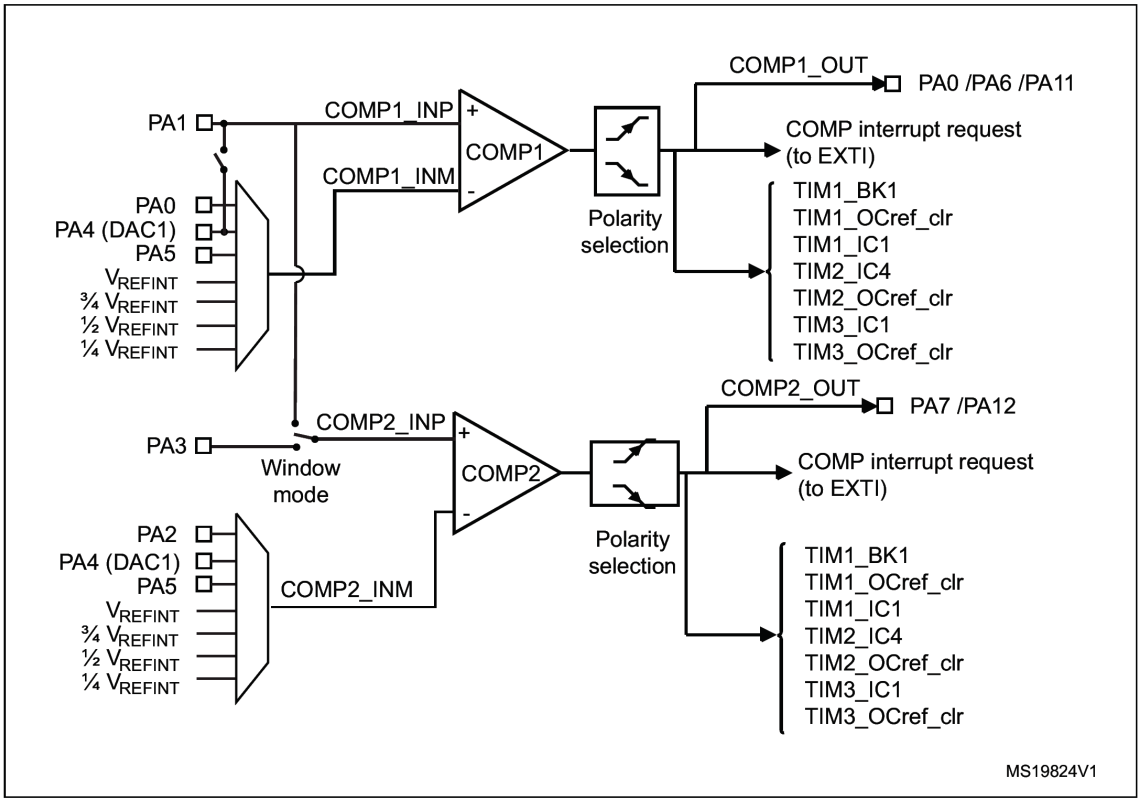
14.2 比较器主要特性

- 轨对轨比较器
- 每个比较器有可选门限
  - 3 个 I/O 引脚
  - DAC1
  - 内部电压和三个等分电压值 (1/4, 1/2, 3/4)
- 可编程迟滞
- 可编程的速率和损耗
- 输出端可以重定向到一个 I/O 端口或多个定时器输入端，可以触发以下事件：
  - 捕获事件
  - OCref_clr 事件（逐周期电流控制）
  - 为实现快速 PWM 关断的中断事件
- 两个比较器可以组合在一个窗口比较器中使用。
- 每个比较器都可产生中断，并支持从 Sleep 和 Stop 模式唤醒 (通过 EXTI 控制器)。

14.3 比较器功能描述

14.3.1 简介

图 41. 比较器框图



14.3.2 时钟

COMP 时钟控制器提供的时钟与 PCLK 同步（APB 时钟）。

在 RCC 控制器中没有提供这个外设的时钟使能控制位。

注： 重要信息：极性选择逻辑与输出端口的重定向工作独立于 PCLK 时钟，这使得比较器可以在停止模式下工作。

14.4 比较器的寄存器

14.4.1 比较器的控制和状态寄存器 (COMP_CSR)

地址偏移 : 0x1C

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
COMP 2LOCK	COMP 2OUT	COMP2HYST [1:0]		COMP 2POL	COMP2OUTSEL[2:0]			WNDW EN	COMP2INSEL[2:0]			COMP2MODE [1:0]		Res.	COMP2 EN
rwo	r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r		rw/r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP 1LOCK	COMP 1OUT	COMP1HYST [1:0]		COMP 1POL	COMP1OUTSEL[2:0]			Res.	COMP1INSEL[2:0]			COMP1MODE [1:0]		COMP1 _INP_D AC	COMP1 EN
rwo	r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r		rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r

位 31 COMP2LOCK: 比较器 2 锁

该位只能写一次，由软件置 1，由系统复位清零。

它令比较器 2 的所有控制位为只读。

0 : COMP_CSR[31:16] 可读 / 可写。

1 : COMP_CSR[31:16] 只读位 .

位 30 COMP2OUT: 比较器 2 输出

只读，反映应比较器 2 输出状态 .

0 : 低输出 (同相输入低于反相输入).

1 : 高输出 (同相输入高于反相输入).

位 29:28 COMP2HYST[1:0] 比较器 2 迟滞

这些位控制比较器 2 的迟滞程度

00 : 没有迟滞

01 : 低度迟滞

10 : 中度迟滞

11 : 高度迟滞

参见迟滞程度的电气特性。

位 27 COMP2POL: 比较器 2 输出极性

该位用于切换比较器 2 输出极性。

0 : 同相输出

1 : 反相输出

位 26:24 COMP2OUTSEL[2:0]: 比较器 2 输出选择

这些位用来选择比较器输出方向。

000: 无选择

001: 定时器 1 中断输入

010: 定时器 1 输入捕捉 1

011: 定时器 1 Ocrefclear 输入

100: 定时器 2 输入捕捉 4

101: 定时器 2 OCrefclear 输入

110: 定时器 3 输入捕捉 1

111: 定时器 3 Ocrefclear 输入

位 23 WNDWEN: 窗口模式使能
该位使 COMP2 和 COMP1 的同相输入端相连，同时与 PA3 分离。
0 : 窗口模式关闭
1 : 窗口模式启用

位 22:20 COMP2INSEL[2:0]: 比较器 2 反相输入选择
这些位用于选择连接到比较器 2 的反相输入的信号源。
000: Vrefint 的 1/4
001: Vrefint 的 1/2
010: Vrefint 的 3/4
011: Vrefint
100: COMP2_INM4 (PA4 或 DAC 输出)
101: COMP2_INM5 (PA5)
110: COMP2_INM6 (PA2)
111: 保留

位 19:18 COMP2MODE[1:0]: 比较器 2 模式
比较器 2 的工作模式控制位，允许调整速率和损耗。
00: 极低功率
01: 低功率
10: 中等速率
11: 高速率

位 17 保留

位 16 COMP2EN: 比较器 2 使能
比较器 2 的开关位
0 : 比较器 2 关闭
1 : 比较器 2 打开

位 15 COMP1LOCK: 比较器 1 锁
该位只能写一次，由软件置 1，由系统复位清零。
它令比较器 1 的所有控制位为只读。
0 : COMP_CSR[15:0] 可读可写。
1 : COMP_CSR[15:0] 只读。

位 14 COMP1OUT: 比较器 1 输出
只读，反映应比较器 1 输出状态
0 : 低输出 (同相输入低于反相输入).
1 : 高输出 (同相输入高于反相输入).

位 13:12 COMP1HYST[1:0] 比较器 1 迟滞
这些位用于控制比较器 1 的迟滞程度
00: 无迟滞
01: 低度迟滞
10: 中度迟滞
11: 高度迟滞
参见迟滞程度的电气特性。

位 11 COMP1POL: 比较器 1 输出极性
该位用于切换比较器 1 输出极性。
0 : 非反相输出
1 : 反相输出

位 10:8 COMP1OUTSEL[2:0]: 比较器 1 输出选择
这些位用来选择比较器 1 的输出方向。
000: 无选择
001: 定时器 1 中断输入
010: 定时器 1 输入捕捉 1
011: 定时器 1 Ocrefclear 输入
100: 定时器 2 输入捕捉 4
101: 定时器 2 OCrefclear 输入
110: 定时器 3 输入捕捉 1
111: 定时器 3 Ocrefclear 输入

位 7 保留

位 6:4 COMP1INSEL[2:0]: 比较器 1 反相输入选择
这些位用于选择连接到比较器 1 的反相输入的信号源。
000: Vrefint 的 1/4
001: Vrefint 的 1/2
010: Vrefint 的 3/4
011: Vrefint
100: COMP1_INM4 (PA4 或 DAC 输出)
101: COMP1_INM5 (PA5)
110: COMP1_INM6 (PA0)
111: 保留

位 3:2 COMP1MODE[1:0]: 比较器 1 模式
比较器 1 的工作模式控制位，允许调整速率和损耗。
00: 极低功率
01: 低功率
10: 中等速率
11: 高速率

位 1 COMP1_SW1: 比较器 1 使能
该位关闭 PA0 上比较器 1 的同相输入端和 PA4（DAC）的 I/O 之间的开关 .
0 : 开关断开
1 : 开关闭合
注：此开关仅用于重定向信号到高阻输入，例如比较器 1 的同相输入（高阻开关）。

位 0 COMP1EN: 比较器 1 使能
该位是比较器开关控制位。
0 : 比较器 1 关闭
1 : 比较器 1 打开

14.4.2 比较器寄存器映像

下表是 COMP 寄存器概述

表 41. 比较器寄存器映像和复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1C	COMP_CSR	COMP2LOCK	COMP2OUT	COMP2HYST[1:0]		COMP2POL	COMP2OUTSEL[2:0]			WNDWEN	COMP2INSEL[2:0]			COMP2MODE[1:0]		Res.	COMP2EN	COMP1LOCK	COMP1OUT	COMP1HYST[1:0]		COMP1POL	COMP1OUTSEL[2:0]			Res.	COMP1INSEL[2:0]			COMP1MODE[1:0]		COMP1_INP_DAC	COMP1EN
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

寄存器边界地址请参考第 35 页的 2.2.2

15 高级控制定时器 (TIM1)

15.1 TIM1 简介

高级控制定时器 (TIM) 由一个 16 位的自动装载计数器组成，它由一个可编程的预分频器 驱动。它适合多种用途，包含测量输入信号的脉冲宽度 (输入捕获)，或者产生输出波形 (输出比较、PWM、嵌入死区时间的互补 PWM 等)。

使用定时器预分频器和 RCC 时钟控制预分频器，可以实现脉冲宽度和波形周期从几个微秒到几个毫秒的调节。

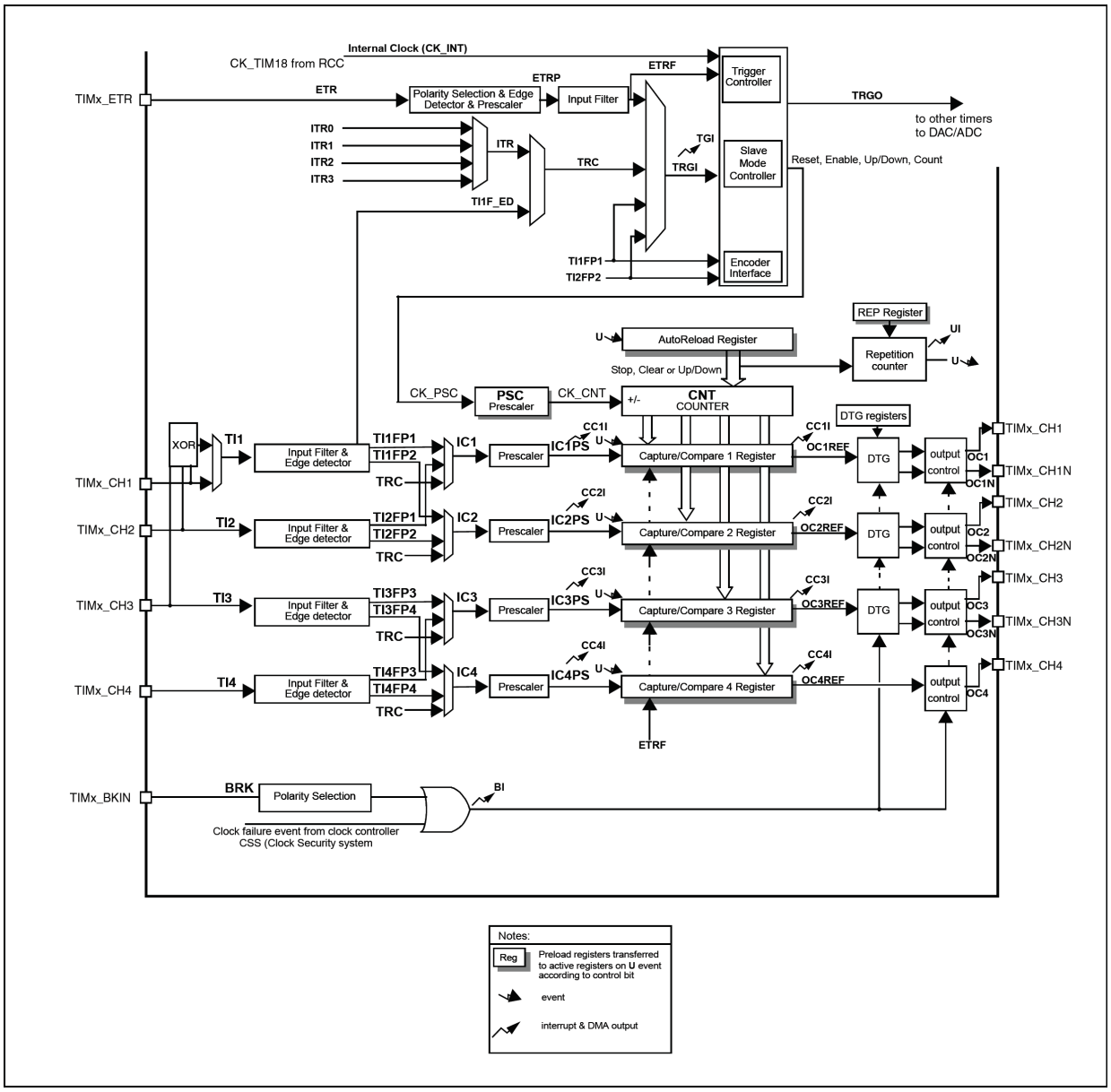
高级控制定时器 (TIM1) 和通用定时器 (TIMx) 是完全独立的，它们不共享任何资源。它们可以同步操作，具体描述参看 15.3.20 节。

15.2 TIM1 主要特性

TIM1 定时器的功能包括：

- 16 位向上、向下、向上 / 下自动装载计数器
- 16 位可编程 (可以实时修改) 预分频器，计数器时钟频率的分频系数为 1 ～ 65535 之间的任意数值
- 多达 4 个独立通道：
  - 输入捕获
  - 输出比较
  - PWM 生成 (边缘或中间对齐模式)
  - 单脉冲模式输出
- 死区时间可编程的互补输出
- 使用外部信号控制定时器和定时器互联的同步电路
- 允许在指定数目的计数器周期之后更新定时器寄存器的重复计数器
- 刹车输入信号可以将定时器输出信号置于复位状态或者一个已知状态
- 如下事件发生时产生中断 /DMA：
  - 更新：计数器向上溢出 / 向下溢出，计数器初始化 (通过软件或者内部 / 外部触发)
  - 触发事件 (计数器启动、停止、初始化或者由内部 / 外部触发计数)
  - 输入捕获
  - 输出比较
  - 刹车信号输入
- 支持针用于定位的增量 (正交) 编码器和霍尔传感器电路
- 触发输入作为外部时钟或者按周期的电流管理

图 42. 高级控制定时器框图



15.3 TIM1 功能描述

15.3.1 时基单元

可编程高级控制定时器的主要部分是一个 16 位计数器和与其相关的自动装载寄存器。这个计数器可以向上计数、向下计数或者向上向下双向计数。此计数器时钟由预分频器分频得到。计数器、自动装载寄存器和预分频器寄存器可以由软件读写，即使计数器还在运行读写仍然有效。

时基单元包含：

- 计数器寄存器 (TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动装载寄存器 (TIMx_ARR)
- 重复次数寄存器 (TIMx_RCR)

自动装载寄存器是预先装载的，写或读自动重装载寄存器将访问预装载寄存器。根据在 TIMx_CR1 寄存器中的自动装载预装载使能位 (ARPE) 的设置，预装载寄存器的内容被立即或在每次的更新事件 UEV 时传送到影子寄存器。当计数器达到溢出条件 (或向下计数时的下溢条件) 并当 TIMx_CR1 寄存器中的 UDIS 位等于 0 时，产生更新事件。更新事件也可以由软件产生。随后会详细描述每一种配置下更新事件的产生。

计数器由预分频器的时钟输出 CK_CNT 驱动，仅当设置了计数器 TIMx_CR1 寄存器中的计数器使能位 (CEN) 时，CK_CNT 才有效。(更多有关使能计数器的细节，请参见控制器的从模式描述)。

注意，在设置了 TIMx_CR 寄存器的 CEN 位的一个时钟周期后，计数器开始计数。

预分频器描述

预分频器可以将计数器的时钟频率按 1 到 65536 之间的任意值分频。它是基于一个 (在 TIMx_PSC 寄存器中的)16 位寄存器控制的 16 位计数器。因为这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

图 44 和图 45 给出了在运行时更改预分频器因子，计数器的动作的例子。

图 43. 当预分频器的参数从 1 变到 2 时，计数器的时序图

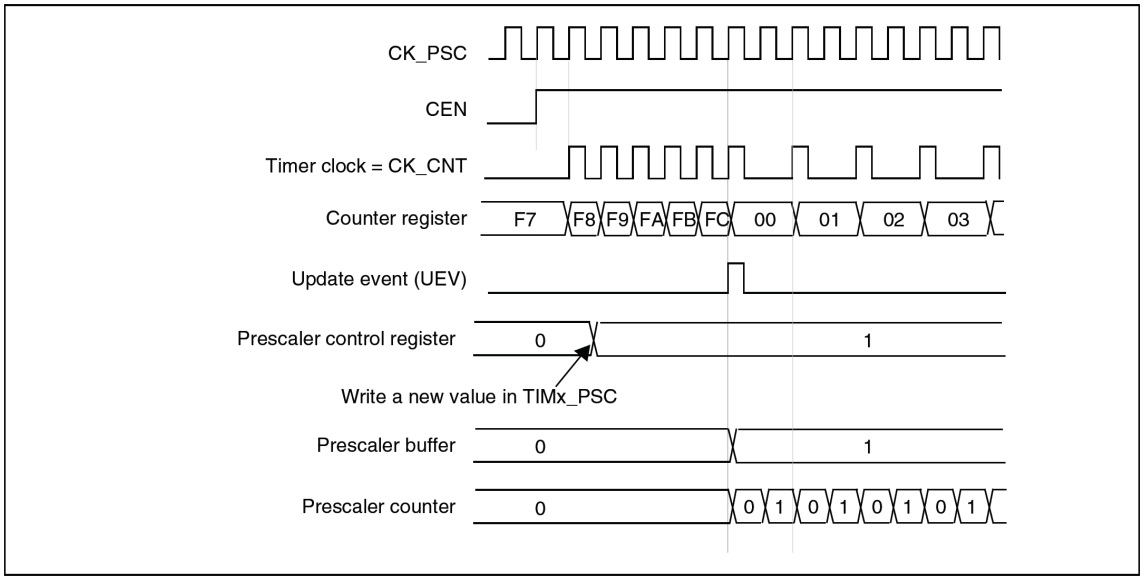
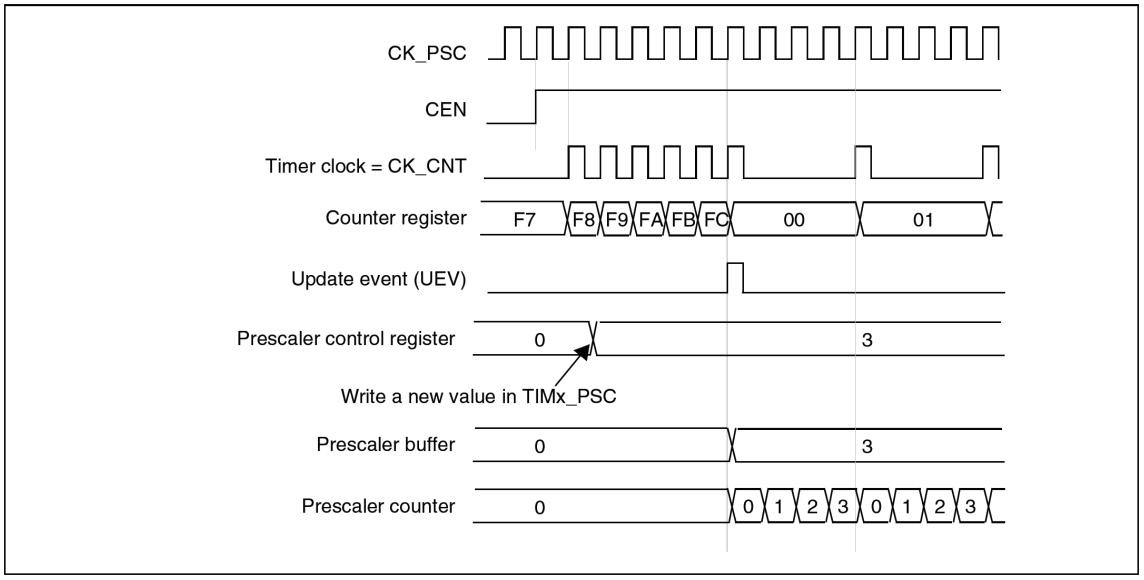


图 44. 当预分频器的参数从 1 变到 4 时，计数器的时序图



15.3.2 计数器模式

向上计数模式

在向上计数模式中，计数器从 0 计数到自动加载值 (TIMx_ARR 计数器的内容)，然后重新从 0 开始计数并且产生一个计数器溢出事件。

如果使用了重复计数器功能，在向上计数达到设置的重复计数次数 (TIMx_RCR) 时，才产生更新事件 (UEV)；否则每次计数器溢出时都会产生更新事件。

在 TIMx_EGR 寄存器中 (通过软件方式或者使用从模式控制器) 设置 UG 位也同样可以产生一个更新事件。

通过软件设置 TIMx_CR1 寄存器中的 UDIS 位，可以禁止更新事件；这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。在 UDIS 位被清’0’之前，将不产生更新事件。但是在应该产生更新事件时，计数器会被清’0’，同时预分频器的计数也被清 0(但预分频器的数值不变)。此外，如果设置了 TIMx_CR1 寄存器中的 URS 位 (选择更新请求)，设置 UG 位将产生一个更新事件 UEV，但硬件不设置 UIF 标志 (即不产生中断或 DMA 请求)。这是为了避免在捕获模式下清除计数器时，同时产生更新和捕获中断。

当发生一个更新事件时，所有的寄存器都被更新，同时 (依据 URS 位) 设置更新标志位 (TIMx_SR 寄存器中的 UIF 位)。

- 重复计数器被重新加载为 TIMx_RCR 寄存器的内容。
- 自动装载影子寄存器被重新为预装载寄存器的值 (TIMx_ARR)。
- 预分频器的缓冲区被写入预装载寄存器的值 (TIMx_PSC 寄存器的内容)。

下图给出一些例子，当 TIMx_ARR=0x36 时计数器在不同时钟频率下计数器的动作。

图 45. 计数器时序图，内部时钟分频因子为 1

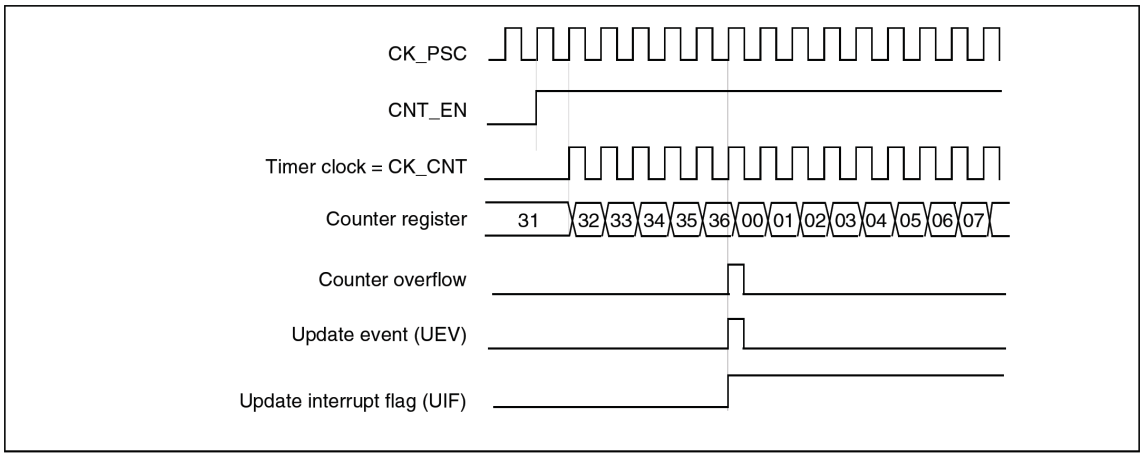


图 46. 计数器时序图，内部时钟分频因子为 2

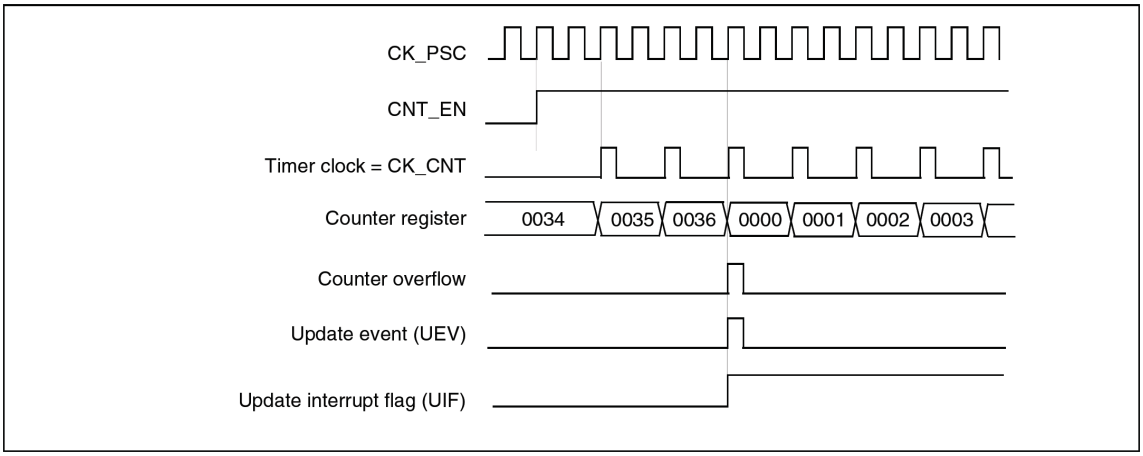


图 47. 计数器时序图，内部时钟分频因子为 4

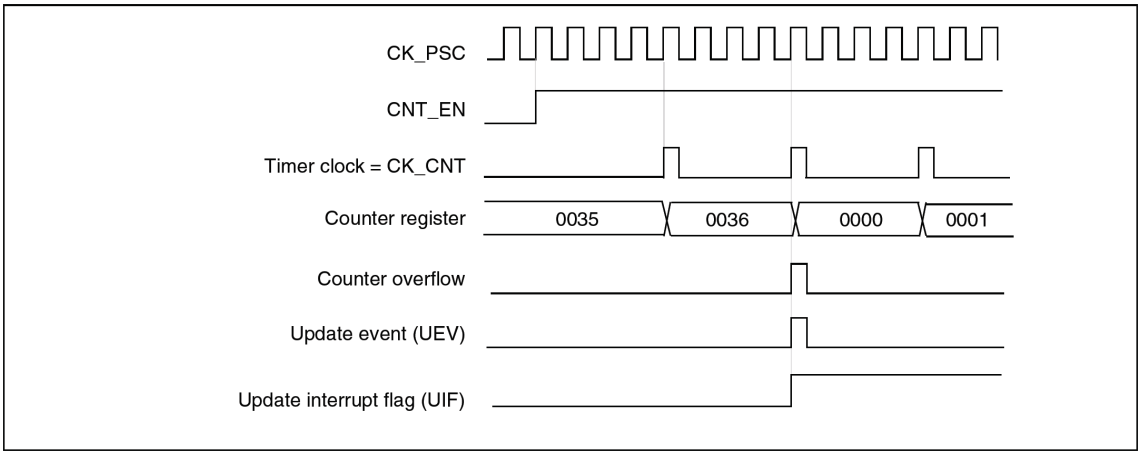


图 48. 计数器时序图，内部时钟分频因子为 N

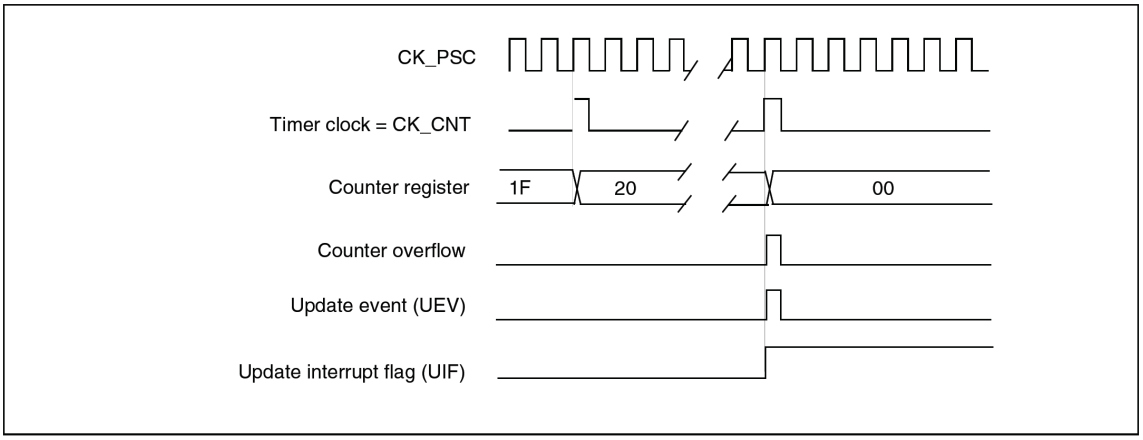


图 49. 计数器时序图，当 ARPE=0 时的更新事件 (TIMx_ARR 没有预装入)

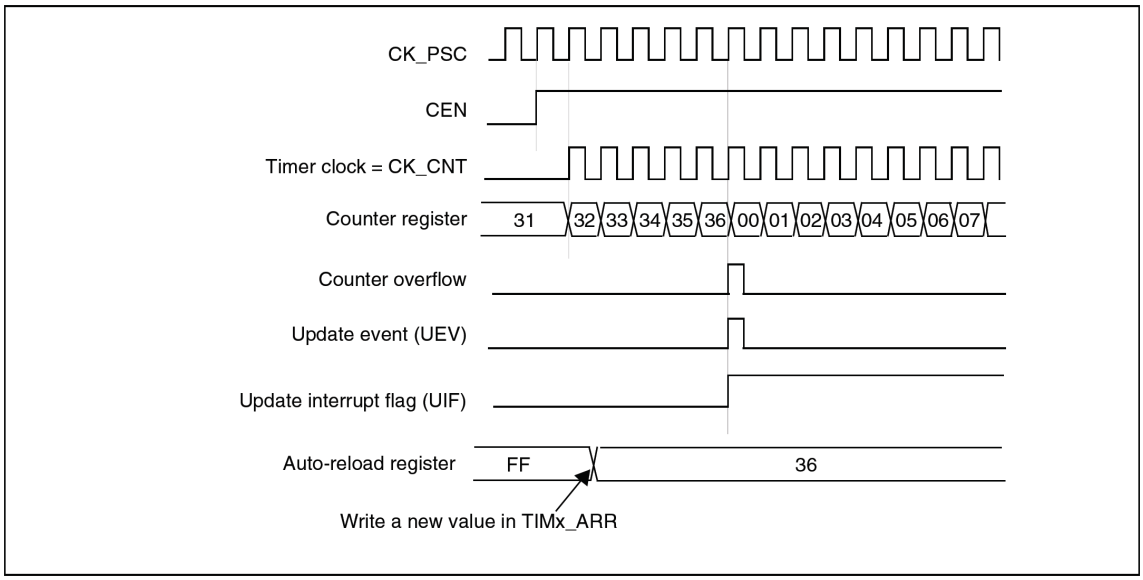
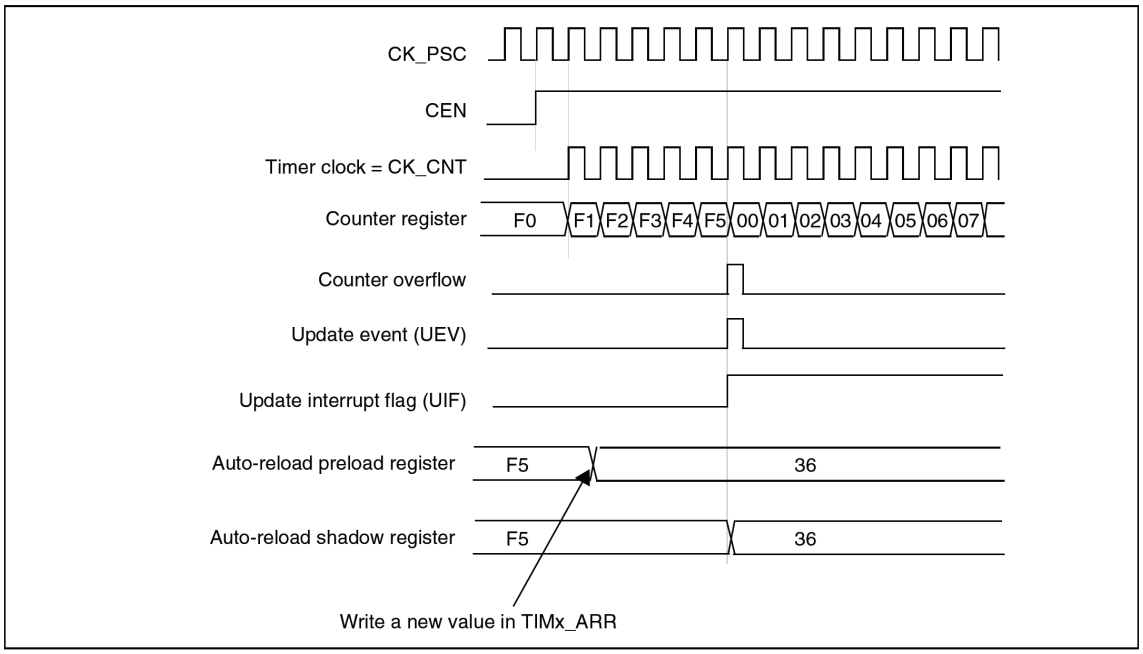


图 50. 计数器时序图，当 ARPE=1 时的更新事件 (TIMx_ARR 已预装入)



向下计数模式

在向下模式中，计数器从自动装入的值 (TIMx_ARR 计数器的值) 开始向下计数到 0，然后从自动装入的值重新开始并且产生一个计数器向下溢出事件。
如果使用了重复计数器，当向下计数重复了重复计数寄存器 (TIMx_RCR) 中设定的次数后，才产生更新事件 (UEV)，否则每次计数器下溢时都会产生更新事件。
设置 TIMx_EGR 寄存器的 (通过软件方式或者使用从模式控制器) UG 位，也同样可以产生一个更新事件。

通过软件设置 TIMx_CR1 寄存器中的 UDIS 位，可以禁止更新事件；这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。在 UDIS 位被清’0’之前，将不产生更新事件。但是在应该产生更新事件时，计数器会被清’0’，同时预分频器的计数也被清 0(但预分频器的数值不变)。

此外，如果设置了 TIMx_CR1 寄存器中的 URS 位 (选择更新请求)，设置 UG 位将产生一个更新事件 UEV，但硬件不设置 UIF 标志 (即不产生中断或 DMA 请求)。这是为了避免在捕获模式下清除计数器时，同时产生更新和捕获中断。

当发生一个更新事件时，所有的寄存器都被更新，同时 (依据 URS 位) 设置更新标志位 (TIMx_SR 寄存器中的 UIF 位)。

- 重复计数器被重新加载为 TIMx_RCR 寄存器的内容。
- 预分频器的缓冲区被写入预装载寄存器的值 (TIMx_PSC 寄存器的内容)。
- 自动装载影子寄存器被重新为预装载寄存器的值 (TIMx_ARR)。注，如果在计数器重载之前自动重装被更新，此时在下一个周期时才是预期的值。

下图给出一些例子，当 TIMx_ARR=0x36 时计数器在不同时钟频率下计数器的动作。

图 51. 计数器时序图，内部时钟分频因子为 1

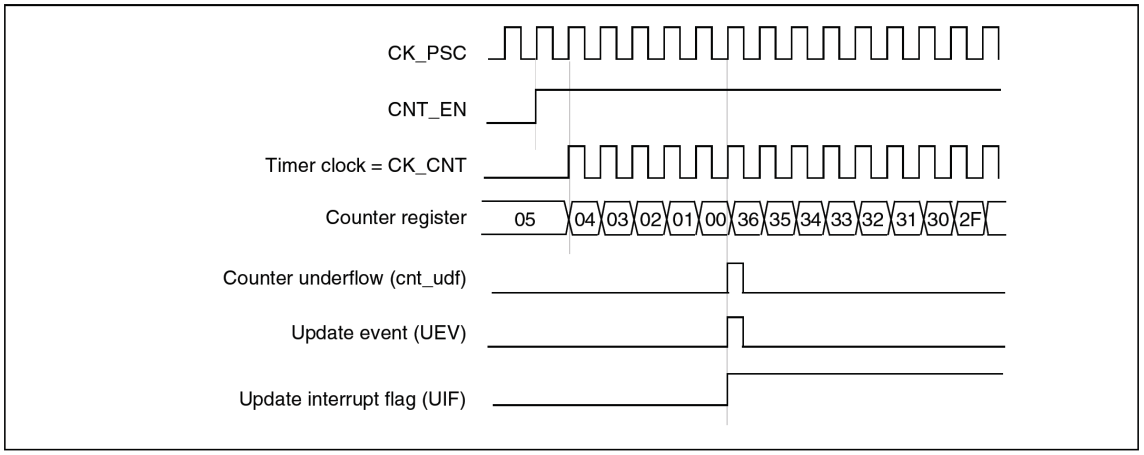


图 52. 计数器时序图，内部时钟分频因子为 2

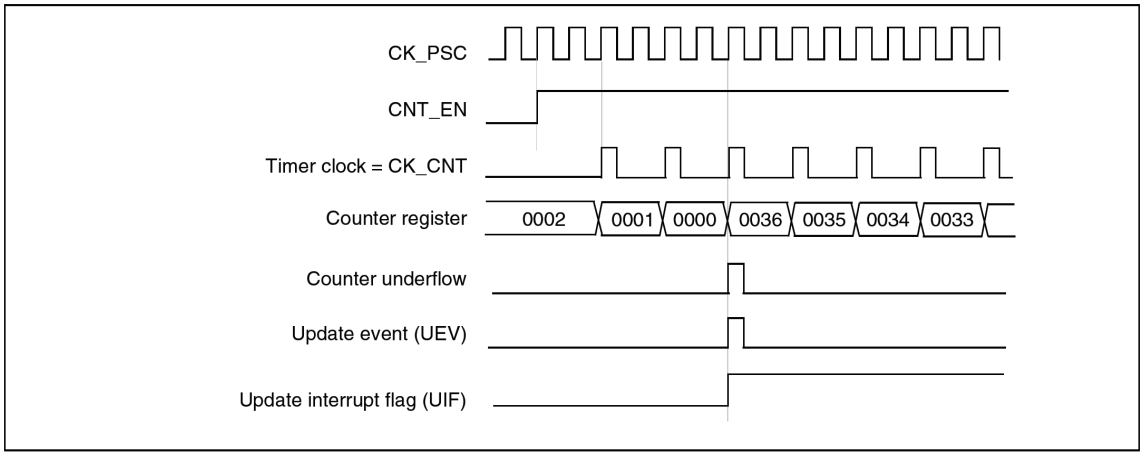


图 53. 计数器时序图，内部时钟分频因子为 4

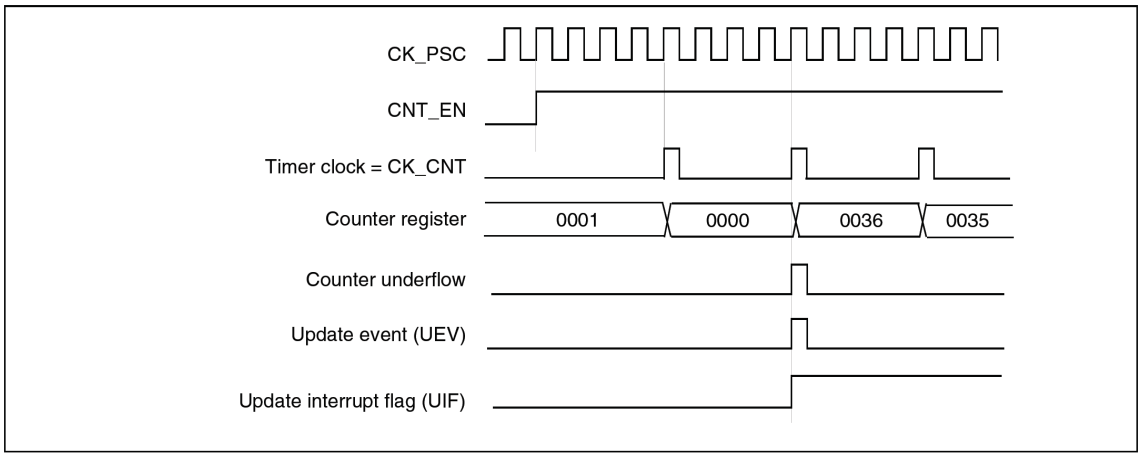


图 54. 计数器时序图，内部时钟分频因子为 N

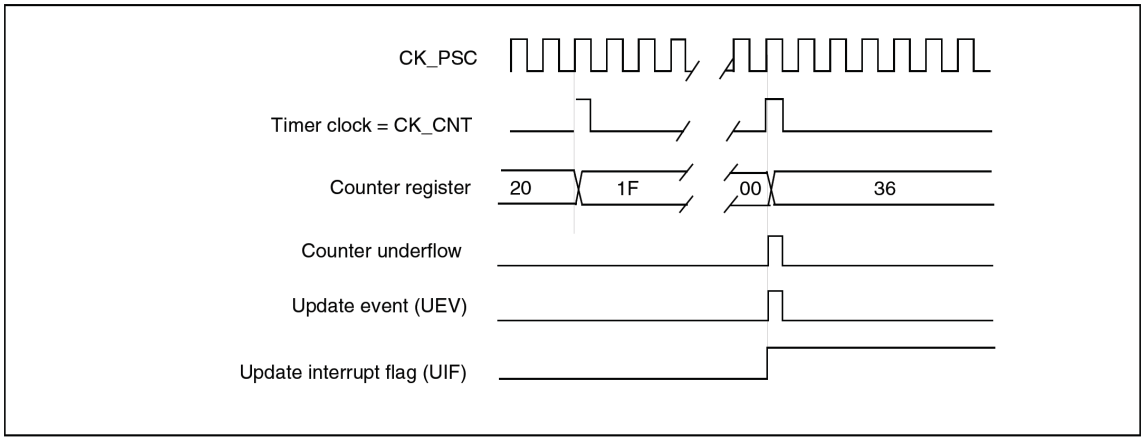
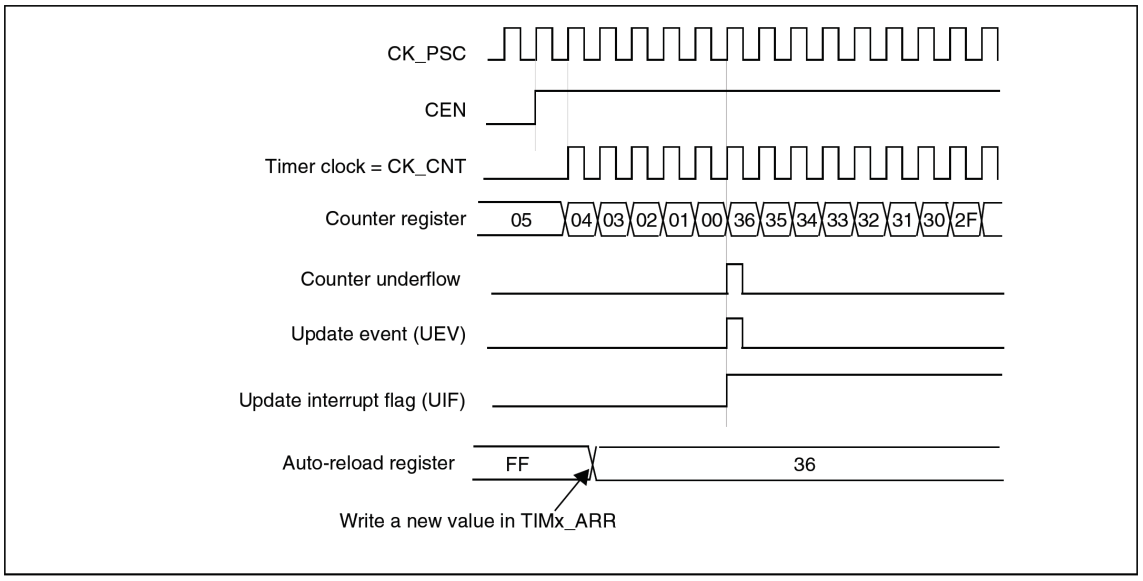


图 55. 计数器时序图，没有使用重复计数器时的更新事件



中央对齐模式 (向上 / 向下计数)

在中央对齐模式，计数器从 0 开始计数到自动加载的值 (TIMx_ARR 寄存器) 一 1，产生一个计数器溢出事件，然后向下计数到 1 并且产生一个计数器下溢事件；然后再从 0 开始重新计数。当 TIMx_CR1 的 CMS 位不为 "00" 时，使能中央对齐模式。已配置为输出的通道的输出比较中断标志在以下情况下被置位：计数器向下计数（中央对齐模式 1, CMS = "01"），计数器向上计数（中央对齐模式 2, CMS = "10"），计数器向下和向上计数（中央对齐模式 3, CMS = "11"），

在此模式下，不能写入 TIMx_CR1 中的 DIR 方向位。它由硬件更新并指示当前的计数方向。可以在每次计数上溢和每次计数下溢时产生更新事件；也可以通过 (软件或者使用从模式控制器) 设置 TIMx_EGR 寄存器中的 UG 位产生更新事件。此时，计数器重新从 0 开始计数，预分频器也重新从 0 开始计数。

设置 TIMx_CR1 寄存器中的 UDIS 位可以禁止 UEV 事件。这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。因此 UDIS 位被清为 0 之前不会产生更新事件。然而，计数器仍会根据当前自动重加载的值，继续向上或向下计数。

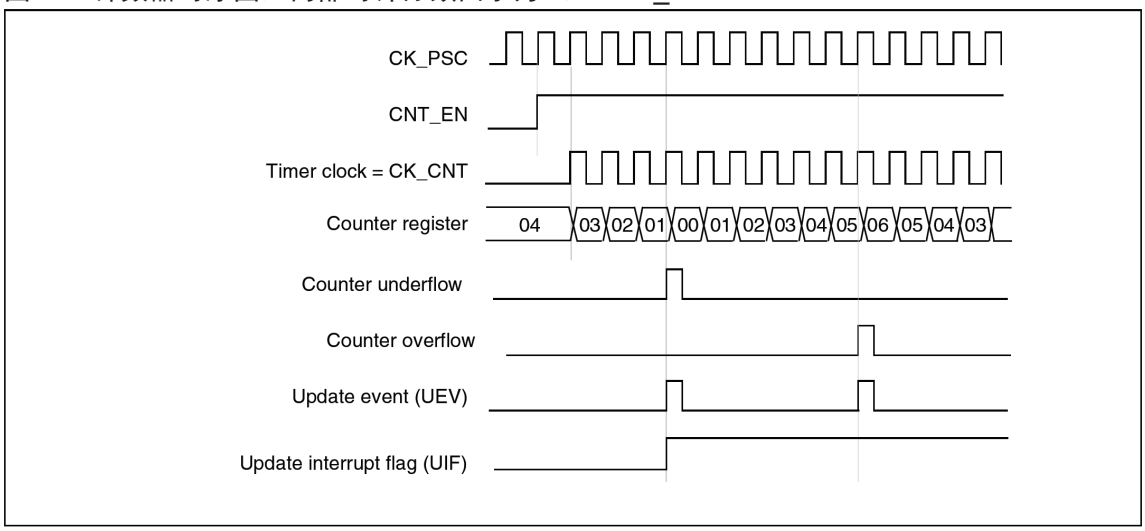
此外，如果设置了 TIMx_CR1 寄存器中的 URS 位 (选择更新请求) ，设置 UG 位将产生一个更新事件 UEV，但不设置 UIF 标志 (因此不产生中断和 DMA 请求)，这是为了避免在发生捕获事件并清除计数器时，同时产生更新和捕获中断。

当发生更新事件时，所有的寄存器都被更新，并且 (根据 URS 位) 更新标志位 (TIMx_SR 寄存器中的 UIF 位) 也被设置。

- 重复计数器被重置为 TIMx_RCR 寄存器中的内容
- 预分频器的缓存器被加载为预装载 (TIMx_PSC 寄存器) 的值。
- 当前的自动加载寄存器被更新为预装载值 (TIMx_ARR 寄存器中的内容)。注：如果因为计数器溢出而产生更新，自动重装载将在计数器重载入之前被更新，因此下一个周期才是预期的值 (计数器被装载为新的值)。

以下是一些计数器在不同时钟频率下的操作的例子：

图 56. 计数器时序图，内部时钟分频因子为 1，TIMx_ARR=0x6



1. 这里使用了中心对齐模式 1(详见 261 页 15.4：TIM1 寄存器)。

图 57. 计数器时序图，内部时钟分频因子为 2

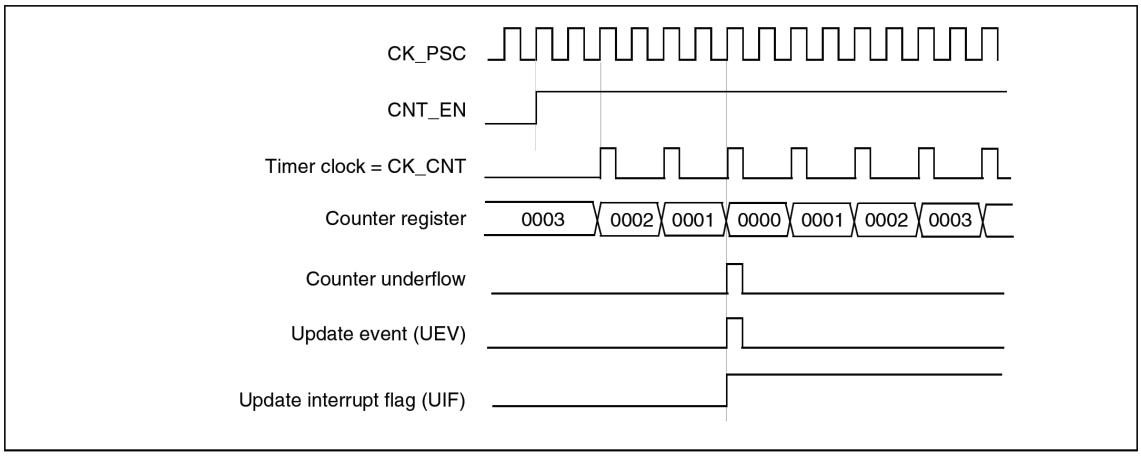
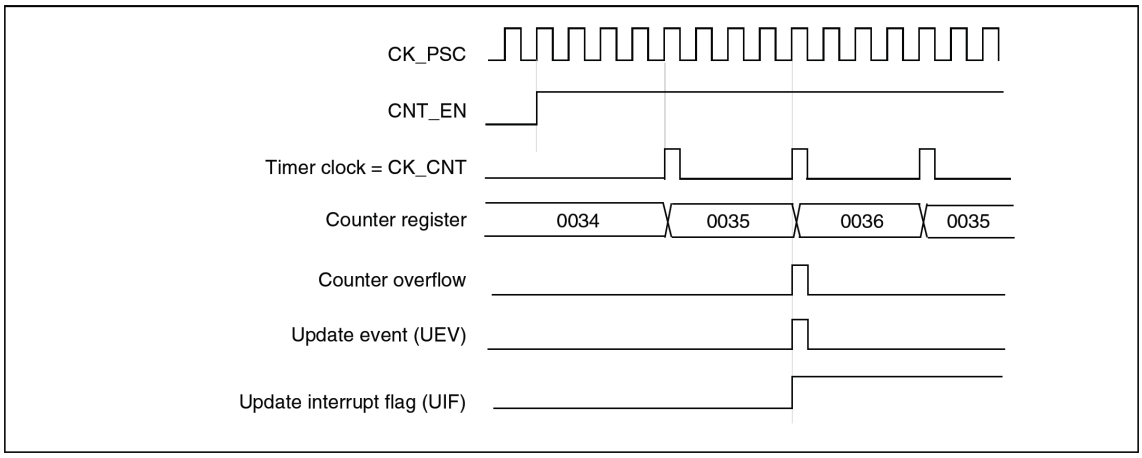


图 58. 计数器时序图，内部时钟分频因子为 4, TIMx_ARR=0x36



1. 这里使用中心对齐模式 2 或 3 带一个 UIF。

图 59. 计数器时序图，内部时钟分频因子为 N

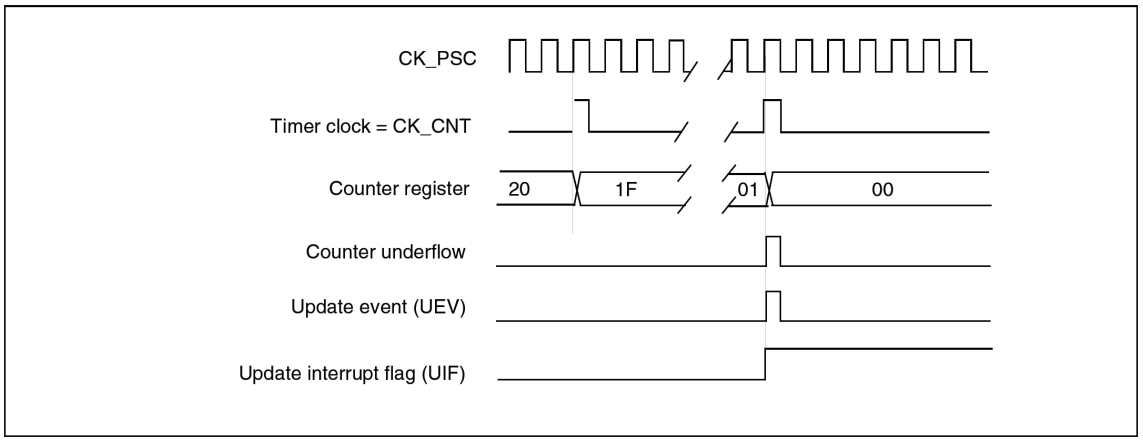


图 60. 计数器时序图，ARPE=1 时的更新事件（计数器下溢）

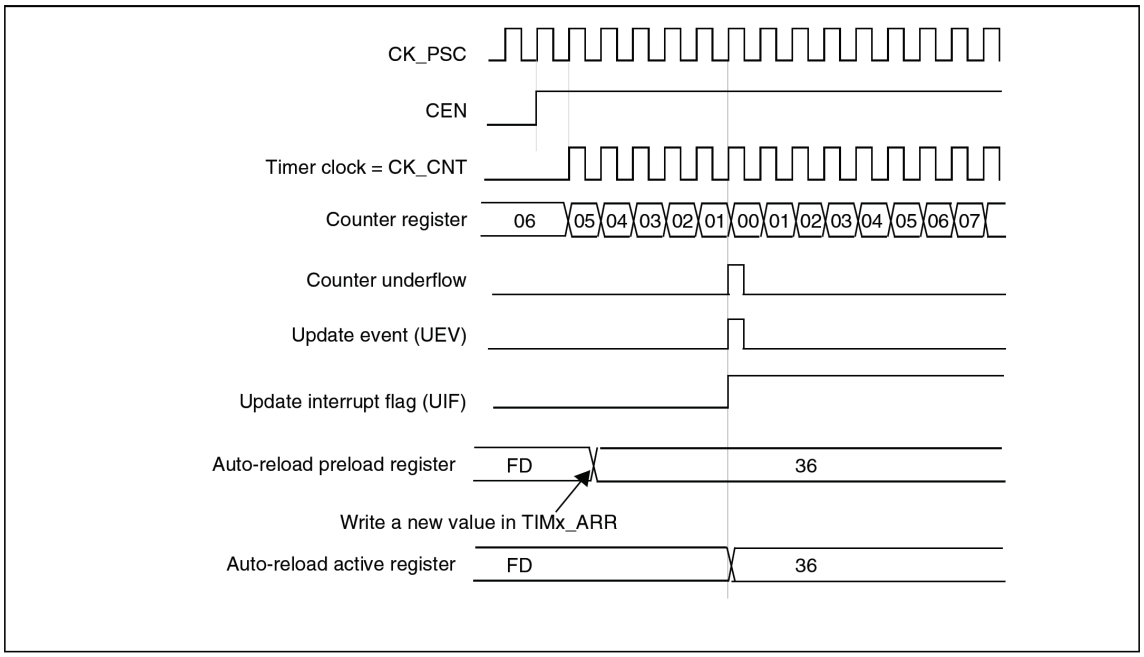
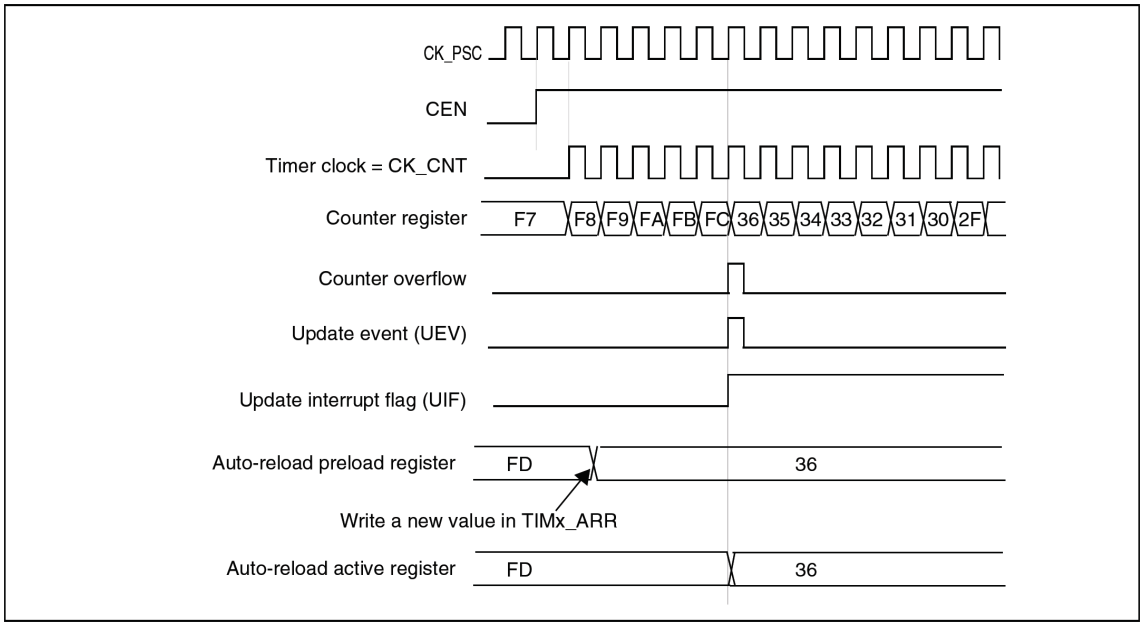


图 61. 计数器时序图，ARPE=1 时的更新事件（计数器溢出）



15.3.3 重复计数器

15.3.1 节“时基单元”解释了计数器上溢 / 下溢时是如何产生更新事件 (UEV) 的，然而事实上它只能在重复计数达到 0 的时候产生。这个特性对产生 PWM 信号非常有用。

这意味着在每 N 次计数上溢或下溢时，数据从预装载寄存器传输到影子寄存器 (TIMx_ARR 自动重载入寄存器，TIMx_PSC 预装 载寄存器，还有在比较模式下的捕获 / 比较寄存器 TIMx_CCRx)，N 是 TIMx_RCR 重复计数寄存器中的值。

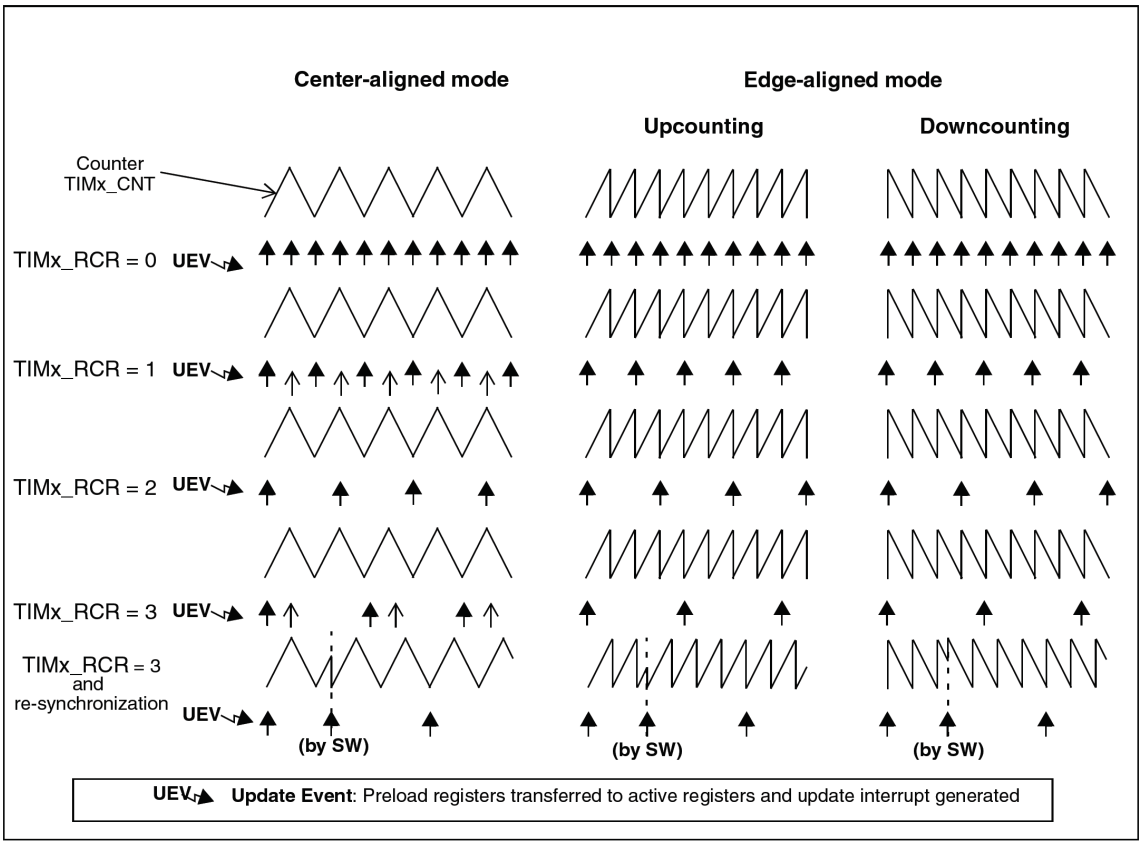
重复计数器在下述任一条件成立时递减：

- 向上计数模式下每次计数器溢出时，
- 向下计数模式下每次计数器下溢时，
- 中央对齐模式下每次上溢和每次下溢时。虽然这样限制了 PWM 的最大循环周期为 128，但它能够在每个 PWM 周期 2 次更新占空比。在中央对齐模式下，因为波形是对称的，如果每个 PWM 周期中仅刷新一次比较寄存器，则最大的分辨率为 2xTck。

重复计数器是自动加载的，重复速率是由 TIMx_RCR 寄存器的值定义 (参看图 62)。当更新事件由软件产生 (通过设置 TIMx_EGR 中的 UG 位) 或者通过硬件的从模式控制器产生，则无论重复计数器的值是多少，立即发生更新事件，并且 TIMx_RCR 寄存器中的内容被重载入到重复计数器。

在中央对齐模式下，如果 RCR 是奇数，一旦 RCR 寄存器被写入并且计数器已经启动，每次溢出和下溢时都发生更新事件。如果 RCR 在计数器开始之前被写入，在溢出时发生 UEV 事件。如果 RCR 在计数器开始之后被写入，在下溢时发生 UEV 事件。例如 RCR = 3，则在 RCR 被写入后的每 4 个溢出和下溢产生 UEV 事件。

图 62. 不同模式及不同 TIMx_RCR 寄存器设置下更新速率的例子



15.3.4 时钟源

计数器时钟可由下列时钟源提供：

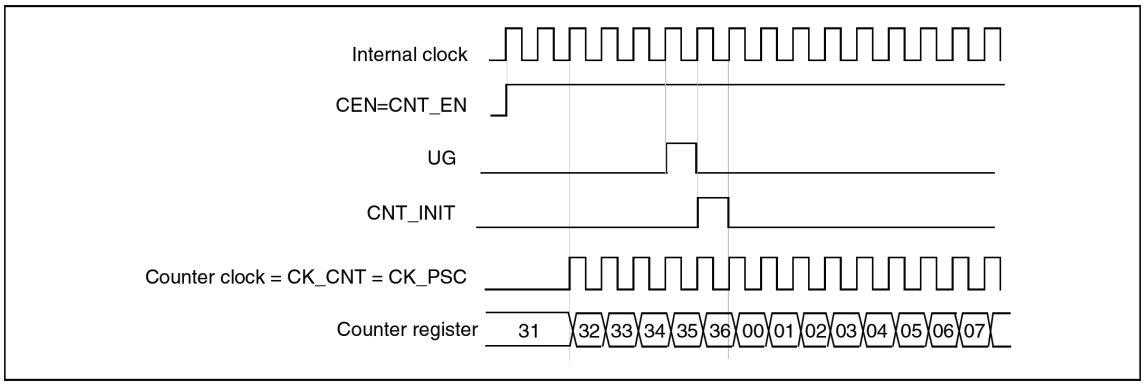
- 内部时钟 (CK_INT)
- 外部时钟模式 1：外部输入引脚
- 外部时钟模式 2：外部触发输入 ETR
- 内部触发输入 (ITRx)：使用一个定时器作为另一个定时器的预分频器。例如，可以配置定时器 Timer1 作为定时器 Timer2 的预分频器。详见使用一个定时器作为另一个定时器的预分频器。

内部时钟源 (CK_INT)

如果禁止了从模式控制器 (SMS=000)，则 CEN、DIR(TIMx_CR1 寄存器) 和 UG 位 (TIMx_EGR 寄存器) 是事实上的控制位，并且只能被软件修改 (除非 UG 位自动被清除)。一旦 CEN 位被写成’1’，预分频器的时钟就由内部时钟 CK_INT 提供。

图 63 显示了在不带预分频器时，控制电路和向上计数器在正常模式下的动作。

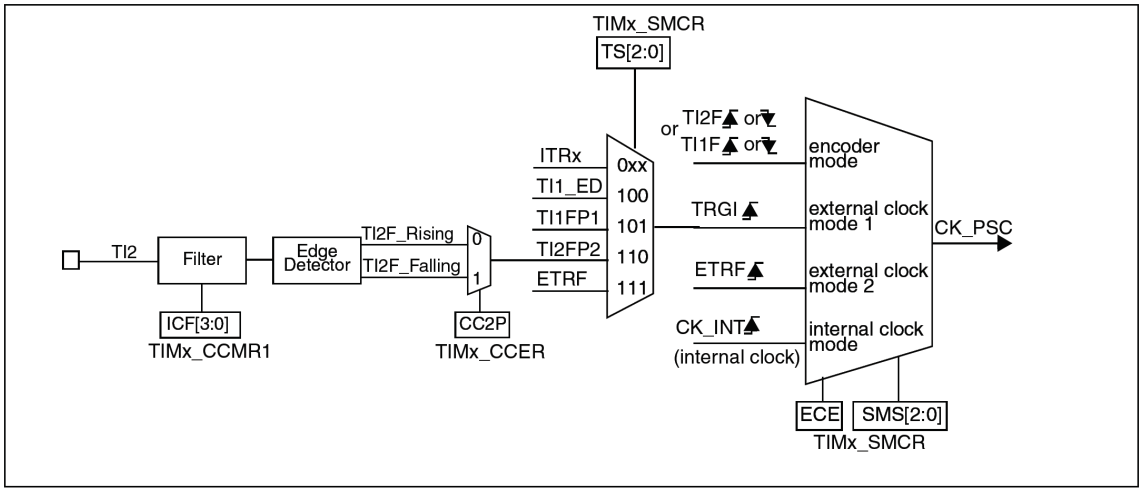
图 63. 内部时钟分频是 1 时正常模式下的控制电路



外部时钟源模式 1

当 TIMx_SMCR 寄存器的 SMS=111 时，此模式被选中。计数器在选定输入端的每个上升沿或下降沿计数。

图 64. TI2 外部时钟连接示例



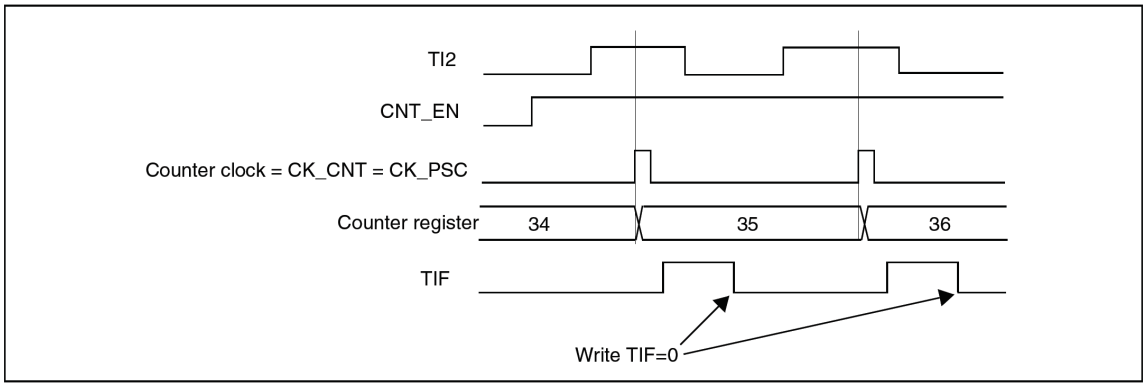
例如，要配置向上计数器在 TI2 输入端的上升沿计数，使用下列步骤：

1. 写 TIMx_CCMR1 寄存器 CC2S=01，配置通道 2 检测 TI2 输入的上升沿
2. 写 TIMx_CCMR1 寄存器的 IC2F[3:0]，选择输入滤波器带宽 (如果不需要滤波器，保持 IC2F=0000)
3. 写 TIMx_CCER 寄存器的 CC2P=0，选定上升沿极性
4. 写 TIMx_SMCR 寄存器的 SMS=111，选择定时器外部时钟模式 1
5. 写 TIMx_SMCR 寄存器中的 TS=110，选定 TI2 作为触发输入源
6. 写 TIMx_CR1 寄存器的 CEN=1，启动计数器

注： *捕获预分频器不用作触发，所以不需要对它进行配置*

当 TI2 上升沿出现时，计数器计数一次，且 TIF 标志被设置
在 TI2 的上升沿和计数器实际时钟之间的延时，取决于在 TI2 输入端的重新同步电路。

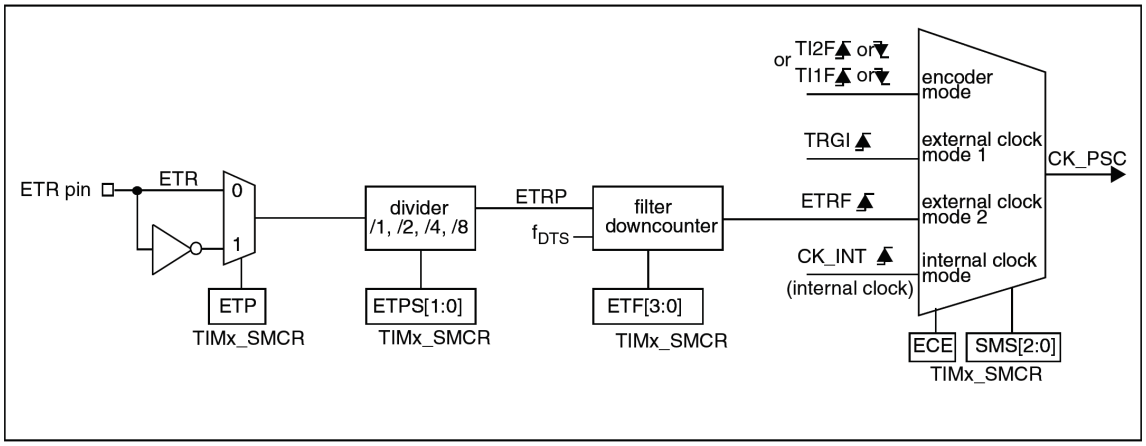
图 65. 外部时钟模式 1 时的控制电路



外部时钟源模式 2

当 TIMx_SMCR 寄存器的 ECE=1 时，此模式被选中。计数器在外部触发 ETR 的每个上升沿或下降沿计数。
图 66 是外部触发输入的框图

图 66. 外部触发输入框图

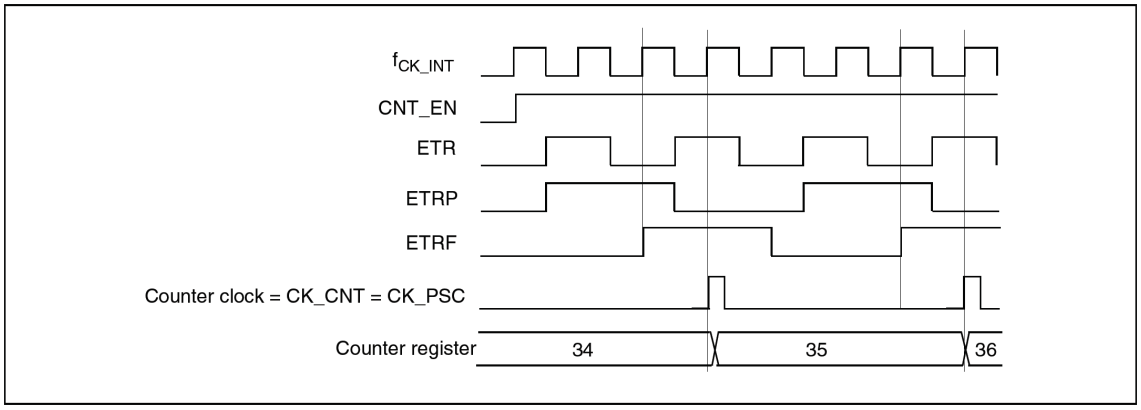


例如，要配置向上计数器在 ETR 的每 2 个上升沿计数一次，使用下列步骤：

1. 本例中不需要滤波器，写 TIMx_SMCR 寄存器中的 ETF[3:0]=0000
2. 写 TIMx_SMCR 寄存器中的 ETPS[1:0]=01，设置预分频器
3. 写 TIMx_SMCR 寄存器中的 ETP=0 选择检测 ETR 的上升沿
4. 写 TIMx_SMCR 寄存器中的 ECE=1，开启外部时钟模式 2
5. 写 TIMx_CR1 寄存器中的 CEN=1，启动计数器

计数器在每 2 个 ETR 上升沿计数一次。
ETR 的上升沿和计数器实际时钟之间的延时取决于在 ETRP 信号端的重新同步电路。

图 67. 外部时钟模式 2 时的控制电路



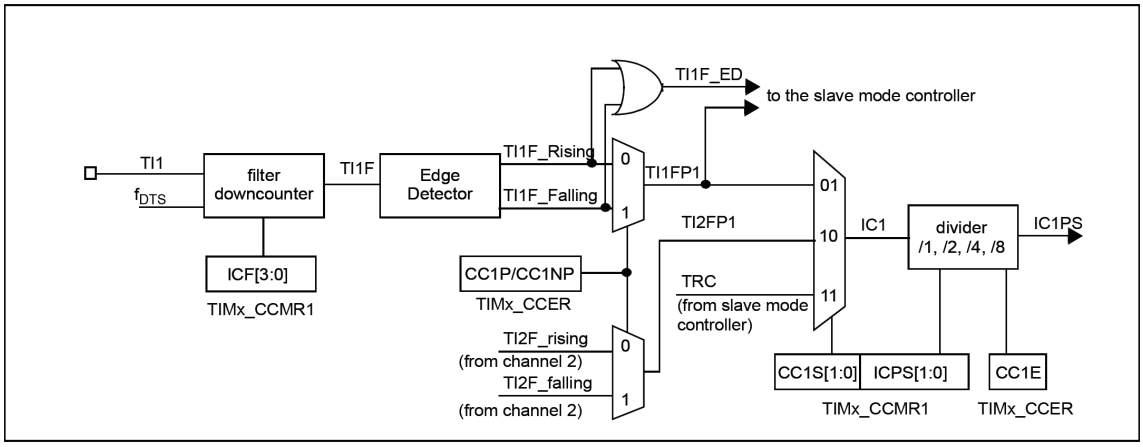
15.3.5 捕获 / 比较通道

每一个捕获 / 比较通道都是包括一个捕获 / 比较寄存器 (包含影子寄存器)，一个捕获的输入部分 (数字滤波、多路复用和预分频器)，和一个输出部分 (比较器和输出控制)。

图 68 至图 71 是一个捕获 / 比较通道的概览。

输入部分采样相对相应的 TIx 输入信号，产生一个滤波后的信号 TIxF。然后，一个带极性选择的边缘监测器产生一个信号 (TIxFPx)，它可以作为从模式控制器的输入触发或者作为捕获控制。该信号被预分频后进入捕获寄存器 (ICxPS)。

图 68. 捕获 / 比较通道 (如：通道 1 输入部分)



输出部分产生一个中间波形 OCxRef(高有效) 作为基准，链的末端决定最终输出信号的极性。

图 69. 捕获 / 比较通道 1 主要框图

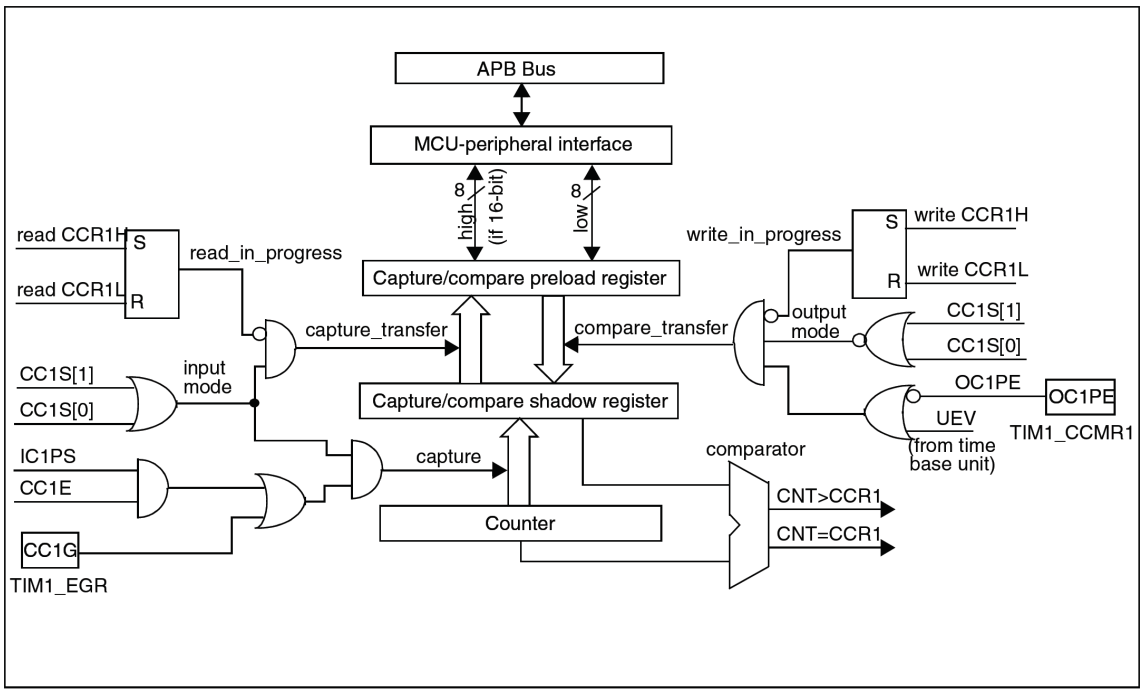


图 70. 捕获 / 比较通道的输出部分 (通道 1 至 3)

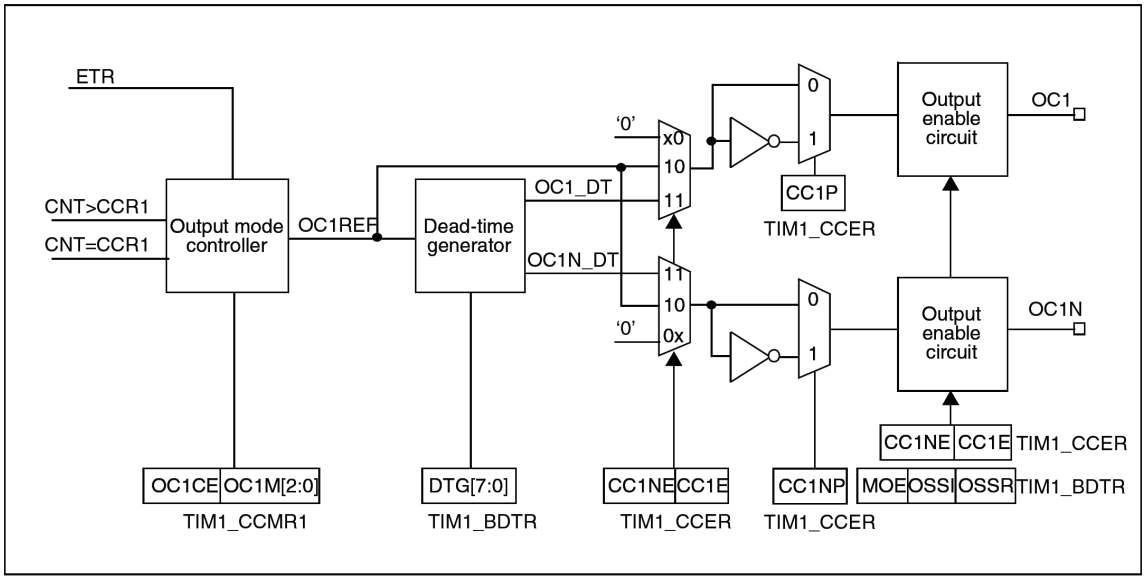
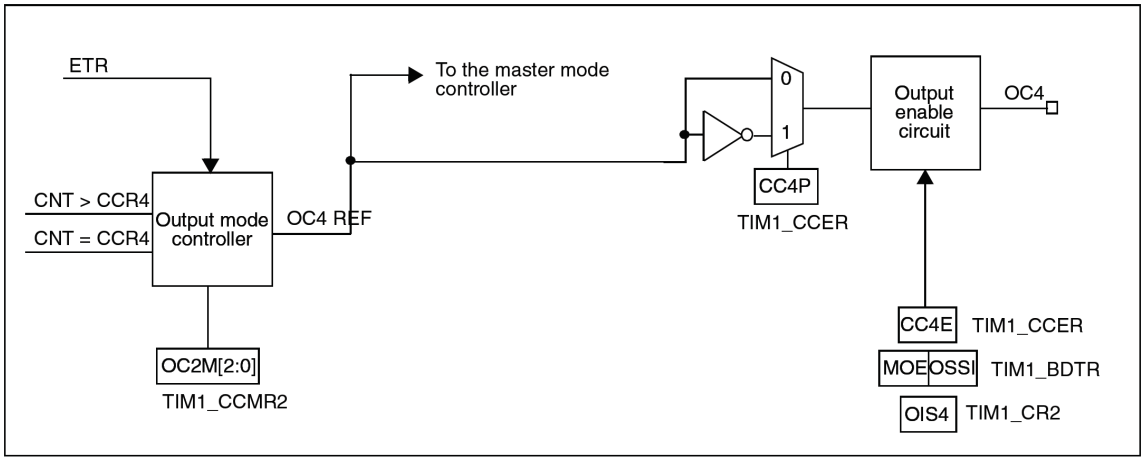


图 71. 捕获 / 比较通道的输出部分 (通道 4)



捕获 / 比较模块由一个预装载寄存器和一个影子寄存器组成。读写过程仅操作预装载寄存器。

在捕获模式下，捕获发生在影子寄存器上，然后再复制到预装载寄存器中。

在比较模式下，预装载寄存器的内容被复制到影子寄存器中，然后影子寄存器的内容和计数器进行比较。

15.3.6 输入捕获模式

在输入捕获模式下，当检测到 ICx 信号上相应的边沿跳变时，计数器的当前值被锁存到捕获 / 比较寄存器 (TIMx_CCRx) 中。当发生捕获事件时，相应的 CCxIF 标志 (TIMx_SR 寄存器) 被置 1，如果开放了中断或者 DMA 则产生一个中断或者一个 DMA 请求。如果发生捕获事件时 CCxIF 标志已经为高，那么重复捕获标志 CCxOF(TIMx_SR 寄存器) 被置 1。软件写 CCxIF=0 可清除 CCxIF，或读取存储在 TIMx_CCRx 寄存器中的捕获数据也可清除 CCxIF。写 CCxOF=0 可清除 CCxOF。

以下例子说明如何在 TI1 输入的上升沿时捕获计数器的值到 TIMx_CCR1 寄存器中，步骤如下：

- 选择有效输入端：TIMx_CCR1 必须链接到 TI1 输入，所以写入 TIMx_CCR1 寄存器中的 CC1S=01，只要 CC1S 不为’00’，通道被配置为输入并且 TIMx_CCR1 寄存器变为只读。
- 根据连接到计数器的输入信号特点，配置输入滤波器为所需的带宽 (输入为 TIx 时，输入滤波器控制位是 TIMx_CCMRx 寄存器中的 ICxF 位)。假设输入信号在最多 5 个内部时钟周期的时间内抖动， 我们须配置滤波器的带宽长于 5 个时钟周期；因此我们可以 (以 fDTS 频率采样) 连续采样 8 次，以确认在 TI1 上一次新的边沿变换。然后在 TIMx_CCMR1 寄存器中写入 IC1F=0011。
- 选择 TI1 通道的有效转换边沿。通过向 TIMx_CCER 寄存器中写入 CC1P=0 和 CC1NP=0(本例中为上升沿)。
- 配置输入预分频器。在本例中，我们希望在每个有效的电平转换时发生一次捕获，因此预分频器被禁止 (写 TIMx_CCMR1 寄存器的 IC1PS=00)。
- 写 TIMx_CCER 寄存器的 CC1E=1，允许捕获计数器的值到捕获寄存器中。
- 如果需要，通过设置 TIMx_DIER 寄存器中的 CC1IE 位允许相关中断请求，通过设置 TIMx_DIER 寄存器中的 CC1DE 位允许 DMA 请求。

当发生一个输入捕获时：

- 产生有效的电平转换时，计数器的值被传送到 TIMx_CCR1 寄存器。
- CC1IF 标志被设置 (中断标志)。当发生至少 2 个连续的捕获时，而 CC1IF 未曾被清除，CC1OF 也被置 1。
- 如设置了 CC1IE 位，则会产生一个中断。
- 如设置了 CC1DE 位，则还会产生一个 DMA 请求。

为了处理捕获溢出，建议在读出捕获溢出标志之前读取数据，这是为了避免丢失在读出捕获溢出标志之后和读取数据之前可能产生的捕获溢出。

注：　通过软件设置 TIMx_EGR 寄存器中相应的 CCxG 位，也可以产生输入捕获中断和 / 或 DMA 请求。

15.3.7 PWM 输入模式

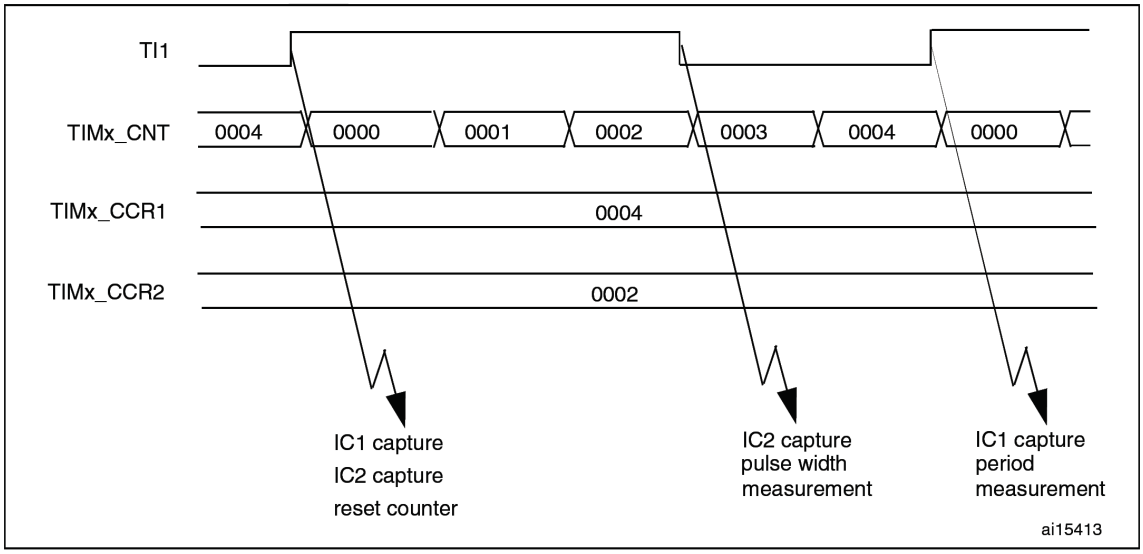
该模式是输入捕获模式的一个特例，除下列区别外，操作与输入捕获模式相同：

- 两个 ICx 信号被映射至同一个 TIx 输入。
- 这 2 个 ICx 信号为边沿有效，但是极性相反。
- 其中一个 TIxFP 信号被作为触发输入信号，而从模式控制器被配置成复位模式。

例如，你需要测量输入到 TI1 上的 PWM 信号的周期 (TIMx_CCR1 寄存器) 和占空比 (TIMx_CCR2 寄存器)，具体步骤如下 (取决于 CK_INT 的频率和预分频器的值)

- 选择 TIMx_CCR1 的有效输入：置 TIMx_CCMR1 寄存器的 CC1S=01(选中 TI1)。
- 选择 TI1FP1 的有效极性 (用来捕获数据到 TIMx_CCR1 中和清除计数器)：置 CC1P=0(上升沿有效)。
- 选择 TIMx_CCR2 的有效输入：置 TIMx_CCMR1 寄存器的 CC2S=10(选中 TI1)。
- 选择 TI1FP2 的有效极性 (捕获数据到 TIMx_CCR2)：置 CC2P=1(下降沿有效)。
- 选择有效的触发输入信号：置 TIMx_SMCR 寄存器中的 TS=101(选择 TI1FP1)。
- 配置从模式控制器为复位模式：置 TIMx_SMCR 中的 SMS=100。
- 使能捕获：置 TIMx_CCER 寄存器中 CC1E=1 且 CC2E=1。

图 72. PWM 输入模式时序



15.3.8 强置输出模式

在输出模式 (TIMx_CCMRx 寄存器中 CCxS=00) 下，输出比较信号 (OCxREF 和相应的 OCx/OCxN) 能够直接由软件强置为有效或无效状态，而不依赖于输出比较寄存器和计数器间的比较结果。

置 TIMx_CCMRx 寄存器中相应的 OCxM=101，即可强置输出比较信号 (OCxREF/OCx) 为有效状态。这样 OCxREF 被强置为高电平 (OCxREF 始终为高电平有效)，同时 OCx 得到 CCxP 极性相反的信号。

例如：CCxP=0(OCx 高电平有效)，则 OCx 被强置为高电平。

置 TIMx_CCMRx 寄存器中的 OCxM=100，可强置 OCxREF 信号为低。

该模式下，在 TIMx_CCRx 影子寄存器和计数器之间的比较仍然在进行，相应的标志也会被修改。因此仍然会产生相应的中断和 DMA 请求。这将会在下面的输出比较模式一节中介绍。

15.3.9 输出比较模式

此项功能是用来控制一个输出波形，或者指示一段给定的的时间已经到时。

当计数器与捕获 / 比较寄存器的内容相同时，输出比较功能做如下操作：

- 将输出比较模式 (TIMx_CCMRx 寄存器中的 OCxM 位) 和输出极性 (TIMx_CCER 寄存器中的 CCxP 位) 定义的值输出到对应的引脚上。在比较匹配时，输出引脚可以保持它的电平 (OCxM=000)、被设置成有效电平 (OCxM=001)、被设置成无效电平 (OCxM=010) 或进行翻转 (OCxM=011)。
- 设置中断状态寄存器中的标志位 (TIMx_SR 寄存器中的 CCxIF 位)。
- 若设置了相应的中断屏蔽 (TIMx_DIER 寄存器中的 CCxIE 位)，则产生一个中断。
- 若设置了相应的使能位 (TIMx_DIER 寄存器中的 CCxDE 位，TIMx_CR2 寄存器中的 CCDS 位选择 DMA 请求功能)，则产生一个 DMA 请求。

TIMx_CCMRx 中的 OCxPE 位选择 TIMx_CCRx 寄存器是否需要使用预装载寄存器。

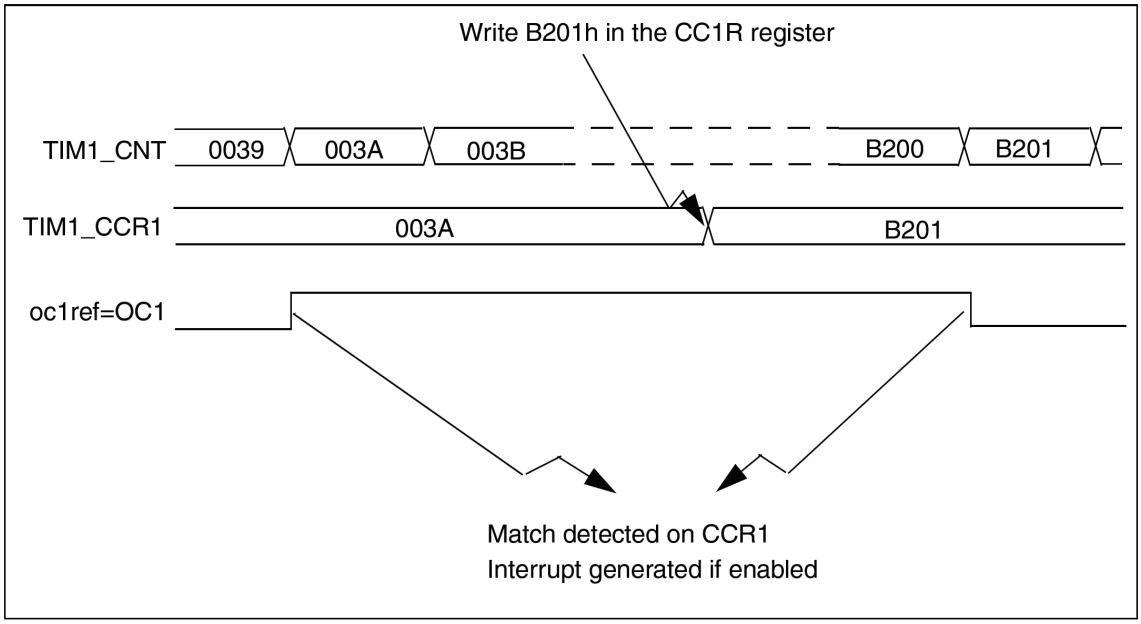
在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出没有影响。同步的精度可以达到计数器的一个计数周期。输出比较模式 (在单脉冲模式下) 也能用来输出一个单脉冲。

输出比较模式的配置步骤：

1. 选择计数器时钟 (内部，外部，预分频器)。
2. 将相应的数据写入 TIMx_ARR 和 TIMx_CCRx 寄存器中。
3. 如果要产生一个中断请求，设置 CCxIE 位。
4. 选择输出模式，例如：
   - 要求计数器与 CCRx 匹配时翻转 OCx 的输出引脚，设置 OCxM=011
   - 置 OCxPE = 0 禁用预装载寄存器
   - 置 CCxP = 0 选择极性为高电平有效
   - 置 CCxE = 1 使能输出
5. 设置 TIMx_CR1 寄存器的 CEN 位启动计数器

TIMx_CCRx 寄存器能够在任何时候通过软件进行更新以控制输出波形，条件是未使用预装载寄存器 (OCxPE=’0’，否则 TIMx_CCRx 的影子寄存器只能在发生下一次更新事件时被更新)。下图给出了一个例子。

图 73. 输出比较模式，翻转 OC1



15.3.10 PWM 模式

脉冲宽度调制模式可以产生一个由 TIMx_ARR 寄存器确定频率、由 TIMx_CCRx 寄存器确定占空比的信号。

在 TIMx_CCMRx 寄存器中的 OCxM 位写入’110’(PWM 模式 1) 或’111’(PWM 模式 2)，能够独立地设置每个 OCx 输出通道产生一路 PWM。必须通过设置 TIMx_CCMRx 寄存器的 OCxPE 位使能相应的预装载寄存器，最后还要设置 TIMx_CR1 寄存器的 ARPE 位，(在向上计数或中心对称模式中) 使能自动重装载的预装载寄存器。

仅当发生一个更新事件的时候，预装载寄存器才能被传送到影子寄存器，因此在计数器开始计数之前，必须通过设置 TIMx_EGR 寄存器中的 UG 位来初始化所有的寄存器。 OCx 的极性可以通过软件在 TIMx_CCER 寄存器中的 CCxP 位设置，它可以设置为高电平有效或低电平有效。OCx 的输出使能通过 (TIMx_CCER 和 TIMx_BDTR 寄存器中)CCxE 、CCxNE 、MOE、OSSI 和 OSSR 位的组合控制。详见 TIMx_CCER 寄存器的描述。

在 PWM 模式 (模式 1 或模式 2) 下，TIMx_CNT 和 TIMx_CCRx 始终在进行比较，(依据计数器的计数方向) 以确定是否符合 TIMx_CCRx ≤ TIMx_CNT 或者 TIMx_CNT ≤ TIMx_CCRx。 根据 TIMx_CR1 寄存器中 CMS 位的状态，定时器能够产生边沿对齐的 PWM 信号或中央对齐的 PWM 信号。

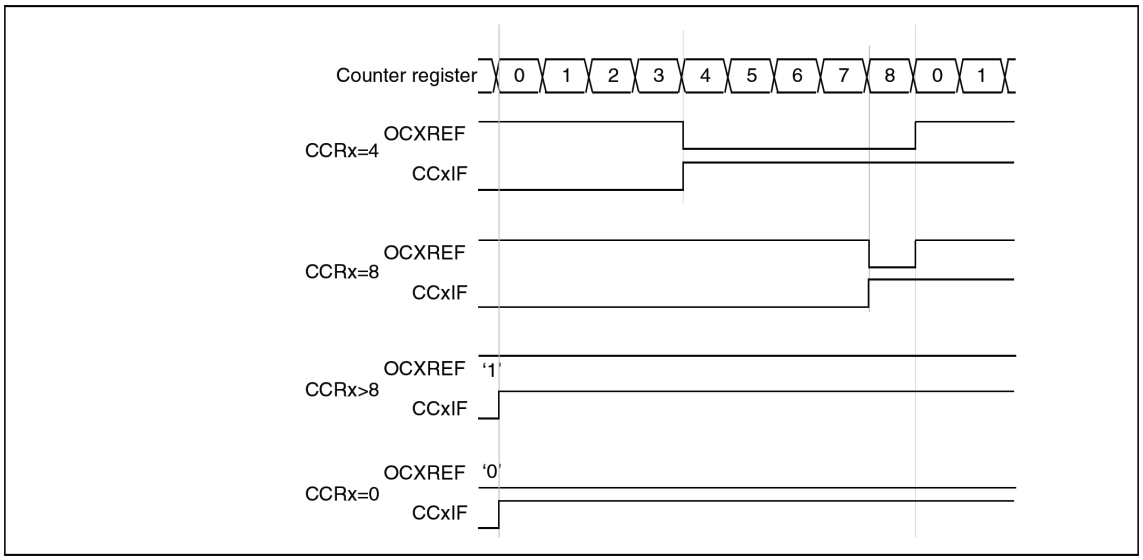
PWM 边沿对齐模式

- 向上计数配置

当 TIMx_CR1 寄存器中的 DIR 位为低的时候执行向上计数。参看 223 页向上计数配置节。下面是一个 PWM 模式 1 的例子。

当 TIMx_CNT<TIMx_CCRx 时，PWM 参考信号 OCxREF 为高，否则为低。如果 TIMx_CCRx 中的比较值大于自动重装载值 (TIMx_ARR)，则 OCxREF 保持为’1’。如果比较值为 0，则 OCxREF 保持为’0’。 图 74 为 TIMx_ARR=8 时边沿对齐的 PWM 波形实例。

图 74. 边沿对齐的 PWM 波形 (ARR=8)



- 向下计数配置

当 TIMx_CR1 寄存器的 DIR 位为高时执行向下计数。参看 225 页向下计数配置节。 在 PWM 模式 1，当 TIMx_CNT>TIMx_CCRx 时参考信号 OCxREF 为低，否则为高。如果 TIMx_CCRx 中的比较值大于 TIMx_ARR 中的自动重装载值，则 OCxREF 保持为’1’。该模式下不能产生 0% 的 PWM 波形。

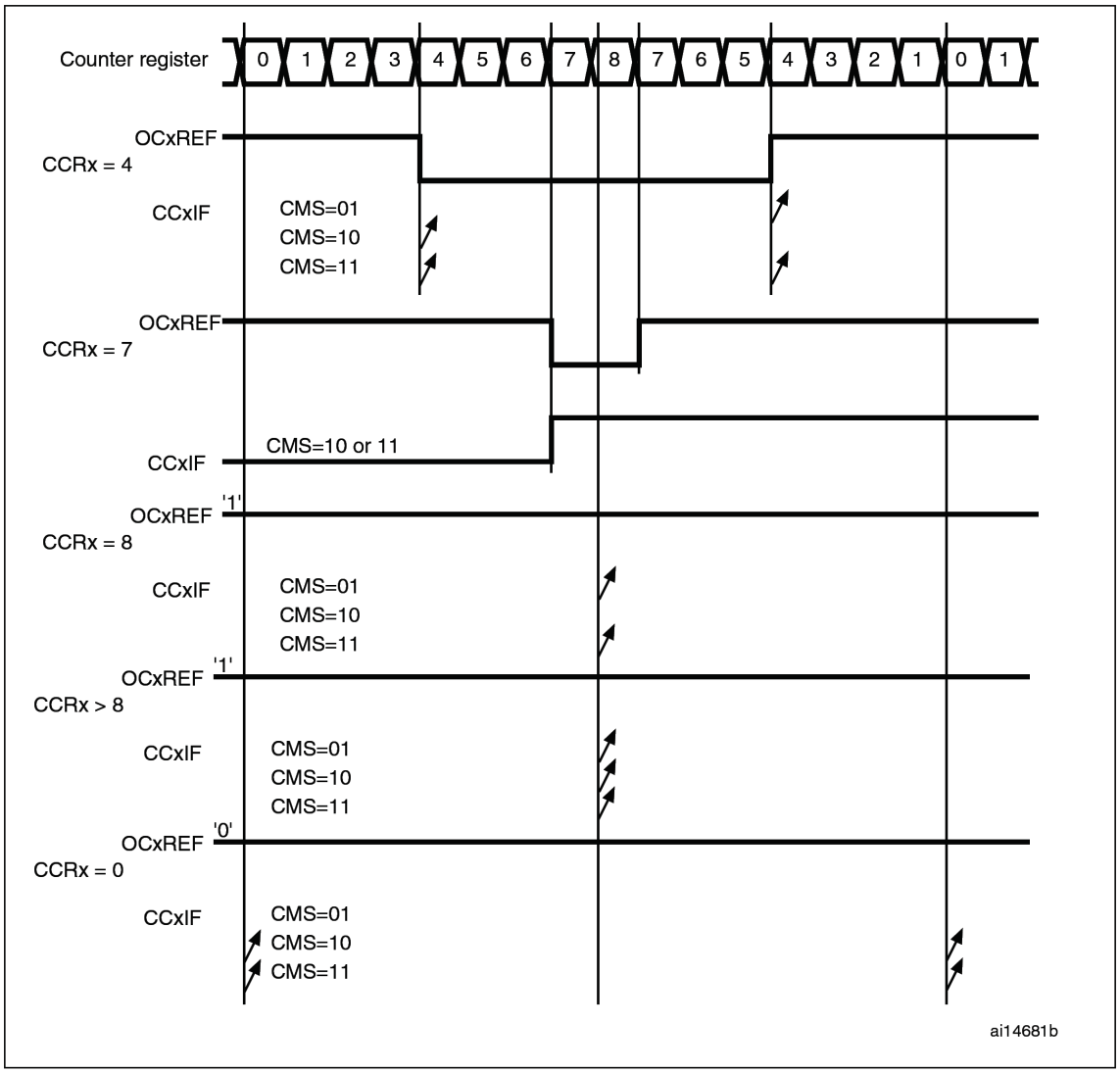
PWM 中央对齐模式

当 TIMx_CR1 寄存器中的 CMS 位不为’00’时为中央对齐模式 (所有其他的配置对 OCxREF/OCx 信号都有相同的作用)。根据不同的 CMS 位设置，比较标志可以在计数器向上计数时，在计数器向下计数时，或在计数器向上和向下计数时被置 1。TIMx_CR1 寄存器中的计数方向位 (DIR) 由硬件更新，不要用软件修改它。参看 228 页的中央对齐模式。

图 75 给出了一些中央对齐的 PWM 波形的例子

- TIMx_ARR=8
- PWM 模式 1
- TIMx_CR1 寄存器的 CMS=01，在中央对齐模式 1 下，当计数器向下计数时设置比较标志。

图 75. 中央对齐的 PWM 波形 (APR=8)



使用中央对齐模式的提示：

- 进入中央对齐模式时，使用当前的向上 / 向下计数配置；这就意味着计数器向上还是向下计 数取决于TIMx_CR1寄存器中DIR位的当前值。此外，软件不能同时修改DIR和CMS位。
- 不推荐当运行在中央对齐模式时改写计数器，因为这会产生不可预知的结果。特别地：
  - 如果写入计数器的值大于自动重加载的值 (TIMx_CNT>TIMx_ARR)，则方向不会被更新。 例如，如果计数器正在向上计数，它就会继续向上计数。
  - 如果将 0 或者 TIMx_ARR 的值写入计数器，方向被更新，但不产生更新事件 UEV。
- 使用中央对齐模式最保险的方法，就是在启动计数器之前产生一个软件更新 (设置 TIMx_EGR 位中的 UG 位)，并且不要在计数进行过程中修改计数器的值。

15.3.11 互补输出和死区插入

高级控制定时器 (TIM1) 能够输出两路互补信号，并且能够管理输出的瞬时关断和接通。

这段时间通常被称为死区，用户应该根据连接的输出器件和它们的特性 (电平转换的延时、电源开关的延时等) 来调整死区时间。

配置 TIMx_CCER 寄存器中的 CCxP 和 CCxNP 位，可以为每一个输出独立地选择极性 (主输出 OCx 或互补输出 OCxN)。

互补信号 OCx 和 OCxN 通过下列控制位的组合进行控制：TIMx_CCER 寄存器的 CCxE 和 CCxNE 位，TIMx_BDTR 和 TIMx_CR2 寄存器中的 MOE、OISx、OISxN、OSSI 和 OSSR 位，详见 278 页表 44：带刹车功能的互补输出通道 OCx 和 OCxN 的控制位。特别的是，在转换到 IDLE 状态时 (MOE 下降到 0) 死区被激活。

同时设置 CCxE 和 CCxNE 位将插入死区，如果存在刹车电路，则还要设置 MOE 位。每一个通道都有一个 10 位的死区发生器。参考信号 OCxREF 可以产生 2 路输出 OCx 和 OCxN。如果 OCx 和 OCxN 为高有效：

- OCx 输出信号与参考信号相同，只是它的上升沿相对于参考信号的上升沿有一个延迟。
- OCxN 输出信号与参考信号相反，只是它的上升沿相对于参考信号的下降沿有一个延迟。

如果延迟大于当前有效的输出宽度 (OCx 或者 OCxN)，则不会产生相应的脉冲。

下列几张图 显示了死区 发生器的输 出信号和当 前参考信号 OCxREF 之 间的关系。 (假设 CCxP=0、CCxNP=0、MOE=1、CCxE=1 并且 CCxNE=1)

图 76. 带死区插入的互补输出

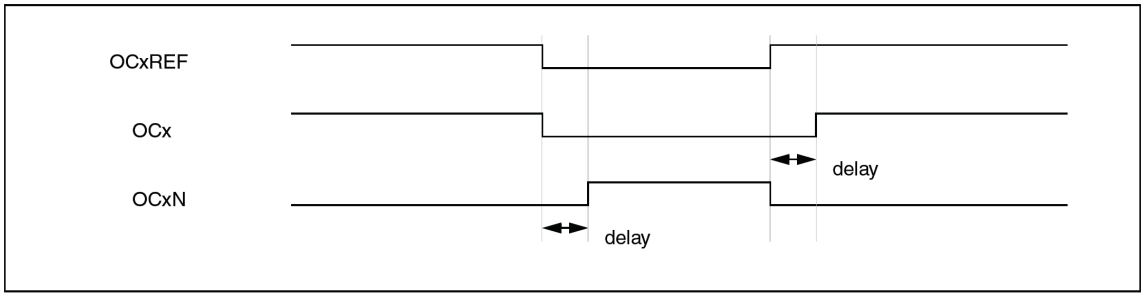


图 77. 死区波形延迟大于负脉冲

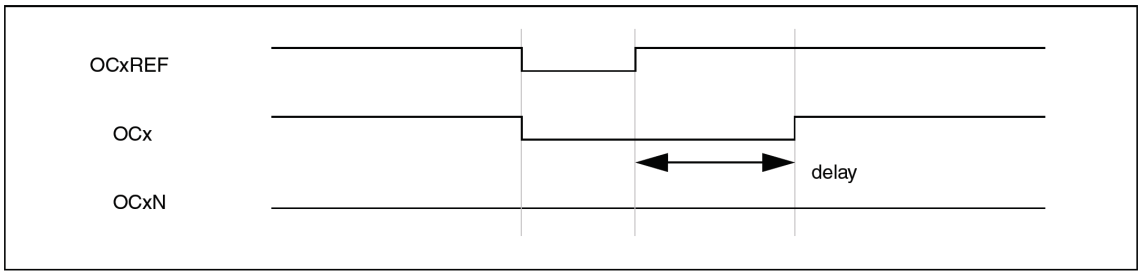
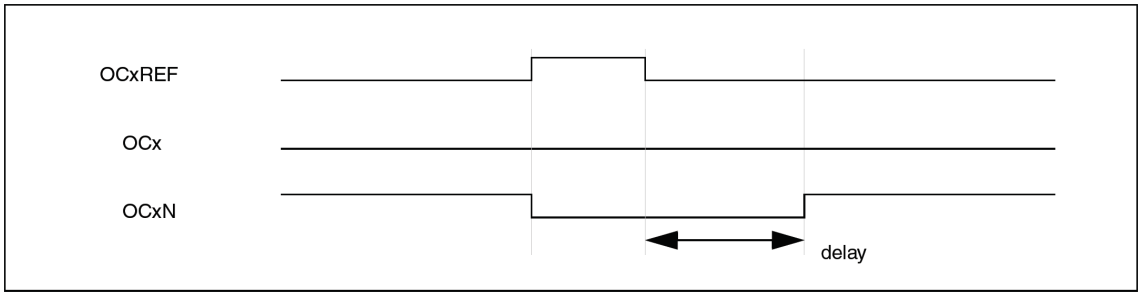


图 78. 死区波形延迟大于正脉冲



每一个通道的死区延时都是相同的，是由 TIMx_BDTR 寄存器中的 DTG 位编程配置。详见 282 页 15.4.18 节 TIM1 刹车和死区寄存器 (TIMx_BDTR) 中的延时计算。

重定向 OCxREF 到 OCx 或 OCxN

在输出模式下 (强置、输出比较或 PWM)，通过配置 TIMx_CCER 寄存器的 CCxE 和 CCxNE 位，OCxREF 可以被重定向到 OCx 或者 OCxN 的输出。

这个功能可以在互补输出处于无效电平时，在某个输出上送出一个特殊的波形 (例如 PWM 或者静态有效电平)。另一个作用是，让两个输出同时处于无效电平，或处于有效电平和带死区的互补输出。

注： 当只使能 OCxN(CCxE=0, CCxNE=1) 时，它不会反相，当 OCxREF 有效时立即变高。例如，如果 CCxNP=0，则 OCxN=OCxREF。另一方面，当 OCx 和 OCxN 都被使能时 (CCxE=CCxNE=1)， 当 OCxREF 为高时 OCx 有效；而 OCxN 相反，当 OCxREF 低时 OCxN 变为有效。

15.3.12 使用刹车功能

当使用刹车功能时，依据相应的控制位 (TIMx_BDTR 寄存器中的 MOE 、OSSI 和 OSSR 位，TIMx_CR2 寄存器中的 OISx 和 OISxN 位)，输出使能信号和无效电平都会被修改。但无论何时，OCx 和 OCxN 输出不能在同一时间同时处于有效电平上。详见 278 页表 44 带刹车功能的互补输出通道 OCx 和 OCxN 的控制位。

刹车源既可以是刹车输入引脚又可以是一个时钟失败事件。时钟失败事件由复位时钟控制器中的时钟安全系统 (CSS) 产生，详见 7.2.7 节时钟安全系统。

系统复位后，刹车电路被禁止，MOE 位为低。设置 TIMx_BDTR 寄存器中的 BKE 位可以使能刹车功能，刹车输入信号的极性可以通过配置同一个寄存器中的 BKP 位选择。BKE 和 BKP 可以同时被修改。当写入 BKE 和 BKP 位时，在真正写入之前会有 1 个 APB 时钟周期的延迟，因此需要等待一个 APB 时钟周期之后，才能正确地读回写入的位。

因为 MOE 下降沿可以是异步的，在实际信号 (作用在输出端) 和同步控制位 (在 TIMx_BDTR 寄存器中) 之间设置了一个再同步电路。这个再同步电路会在异步信号和同步信号之间产生延迟。特别的，如果当它为低时写 MOE=1，则读出它之前必须先插入一个延时 (空指令) 才能读到正确的值。这是因为写入的是异步信号而读的是同步信号。

当发生刹车时 (在刹车输入端出现选定的电平)，有下述动作：

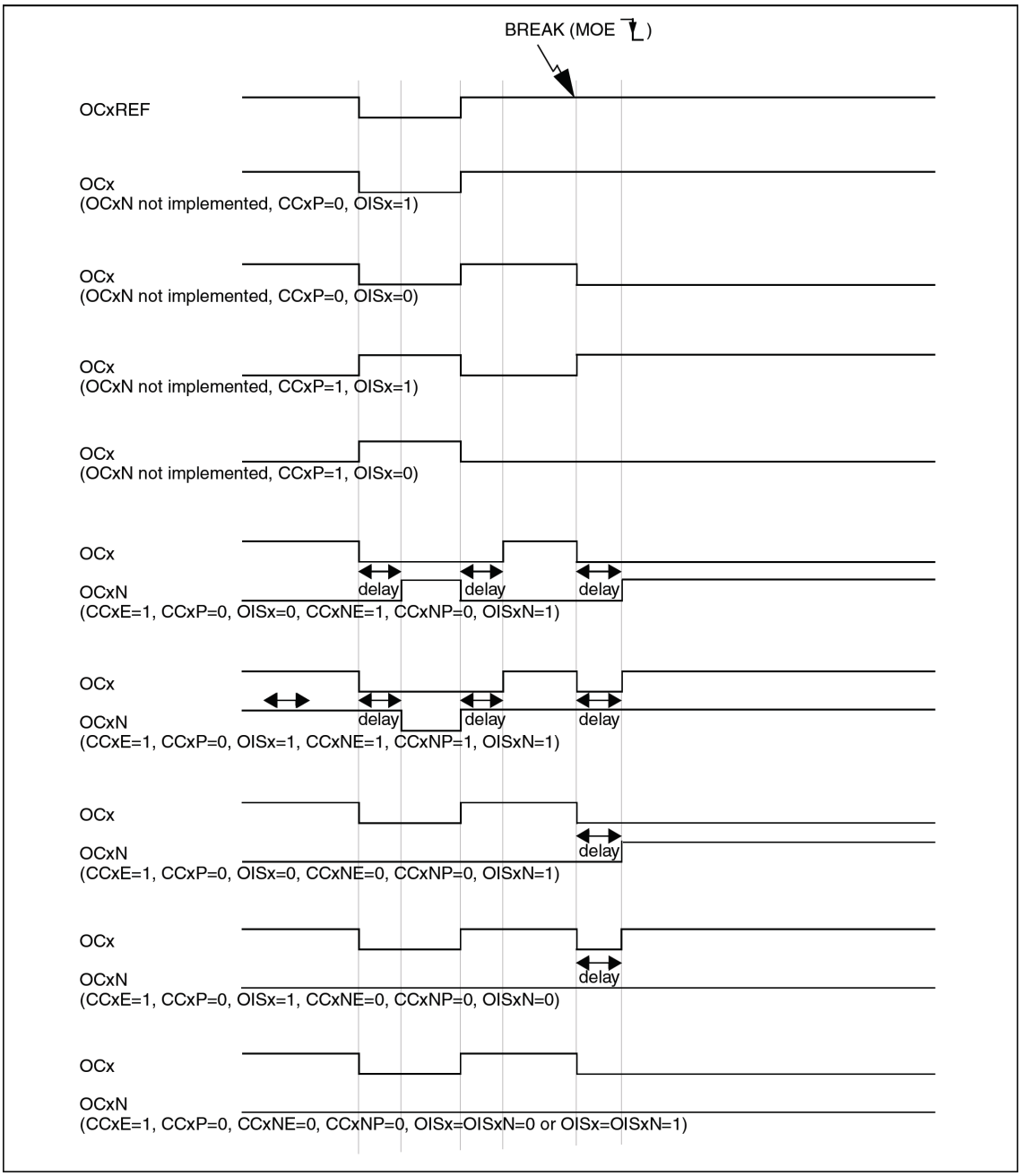
- MOE 位被异步地清除，将输出置于无效状态、空闲状态或者复位状态 (由 OSSI 位选择)。这个特性在 MCU 的振荡器关闭时依然有效。
- 一旦 MOE=0，每一个输出通道输出由 TIMx_CR2 寄存器中的 OISx 位设定的电平。如果 OSSI=0，则定时器释放使能输出，否则使能输出始终为高。
- 当使用互补输出时：
  - 输出首先被置于复位状态即无效的状态 (取决于极性)。这是异步操作，即使定时器没有时钟时，此功能也有效。
  - 如果定时器的时钟依然存在，死区生成器将会重新生效，在死区之后根据 OISx 和 OISxN 位指示的电平驱动输出端口。即使在这种情况下，OCx 和 OCxN 也不能被同时驱动到有效的电平。注，因为重新同步 MOE，死区时间比通常情况下长一些 (大约 2 个 ck_tim 的时钟周期)。
  - 如果 OSSI=0，定时器释放使能输出，否则保持使能输出；或一旦 CCxE 与 CCxNE 之一变高时，使能输出变为高。
- 如果设置了 TIMx_DIER 寄存器中的 BIE 位，当刹车状态标志 (TIMx_SR 寄存器中的 BIF 位) 为’1’时，则产生一个中断。如果设置了 TIMx_DIER 寄存器中的 BDE 位，则产生一个 DMA 请求。
- 如果设置了 TIMx_BDTR 寄存器中的 AOE 位，在下一个更新事件 UEV 时 MOE 位被自动置位； 例如，这可以用来进行整形。否则，MOE 始终保持低直到被再次置’1’；此时，这个特性可以被用在安全方面，你可以把刹车输入连到电源驱动的报警输出、热敏传感器或者其他安全器件上。

注：　刹车输入为电平有效。所以，当刹车输入有效时，不能同时 (自动地或者通过软件) 设置 MOE。同时，状态标志 BIF 不能被清除。

刹车由 BRK 输入产生，它的有效极性是可编程的，且由 TIMx_BDTR 寄存器中的 BKE 位开启。除了刹车输入和输出管理，刹车电路中还实现了写保护以保证应用程序的安全。它允许用户冻结几个配置参数 (死区长度，OCx/OCxN 极性和被禁止的状态，OCxM 配置，刹车使能和极性)。用户可以通过 TIMx_BDTR 寄存器中的 LOCK 位，从三级保护中选择一种，参看 282 页 15.4.18 节 TIM1 刹车和死区寄存器 (TIMx_BDTR)。在 MCU 复位后 LOCK 位只能被修改一次。

下图显示响应刹车的输出实例。

图 79. 响应刹车的输出


BREAK (MOE)
OCxREF
OCx
(OCxN not implemented, CCxP=0, OISx=1)
OCx
(OCxN not implemented, CCxP=0, OISx=0)
OCx
(OCxN not implemented, CCxP=1, OISx=1)
OCx
(OCxN not implemented, CCxP=1, OISx=0)
OCx
OCxN
(CCxE=1, CCxP=0, OISx=0, CCxNE=1, CCxNP=0, OISxN=1)
delay
delay
delay
OCx
OCxN
(CCxE=1, CCxP=0, OISx=1, CCxNE=1, CCxNP=1, OISxN=1)
delay
delay
delay
OCx
OCxN
(CCxE=1, CCxP=0, OISx=0, CCxNE=0, CCxNP=0, OISxN=1)
delay
OCx
OCxN
(CCxE=1, CCxP=0, OISx=1, CCxNE=0, CCxNP=0, OISxN=0)
delay
OCx
OCxN
(CCxE=1, CCxP=0, CCxNE=0, CCxNP=0, OISx=OISxN=0 or OISx=OISxN=1)

15.3.13 在外部事件时清除 OCxREF 信号

对于一个给定的通道，设置 TIMx_CCMRx 寄存器中对应的 OCxCE 位为’1’，能够用 ETRF 输入端的高电平把 OCxREF 信号拉低，OCxREF 信号将保持为低直到发生下一次的更新事件 UEV。

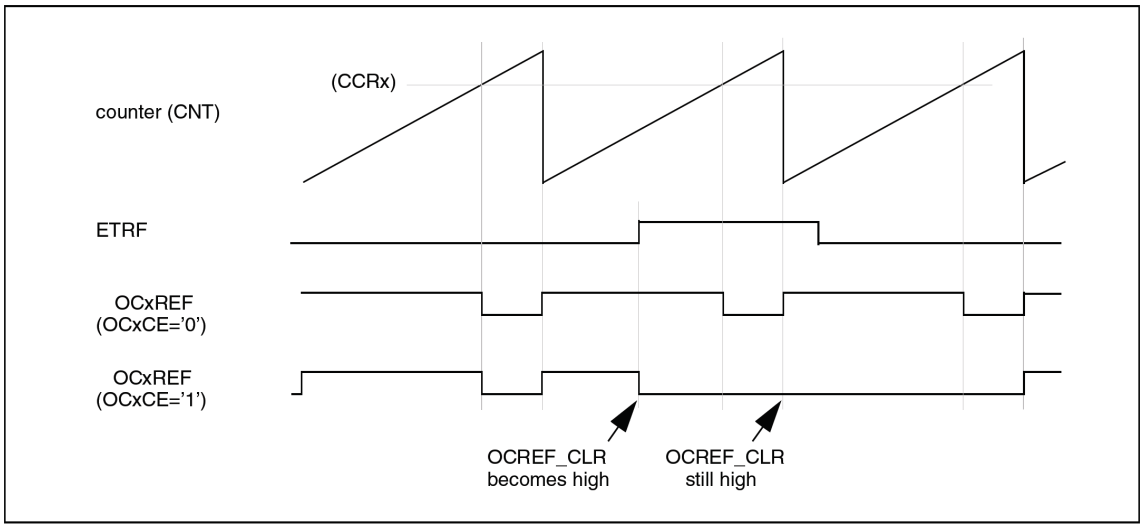
该功能只能用于输出比较和 PWM 模式，而不能用于强置模式。

例如，OCxREF 信号可以联到一个比较器的输出，用于控制电流。这时，ETR 必须配置如下：

1. 外部触发预分频器必须处于关闭：TIMx_SMCR 寄存器中的 ETPS[1:0]=00。
2. 必须禁止外部时钟模式 2：TIMx_SMCR 寄存器中的 ECE=0。
3. 外部触发极性 (ETP) 和外部触发滤波器 (ETF) 可以根据需要配置。

图 80 显示了当 ETRF 输入变为高时，对应不同 OCxCE 的值，OCxREF 信号的动作。在这个例子中，定时器 TIMx 被置于 PWM 模式。

图 80. 清除 TIMx 的 OCxREF



注： *在 100%的 PWM 时（如果 CCRx>ARR），OCxREF 在下一次计数溢出时被再次使能。*

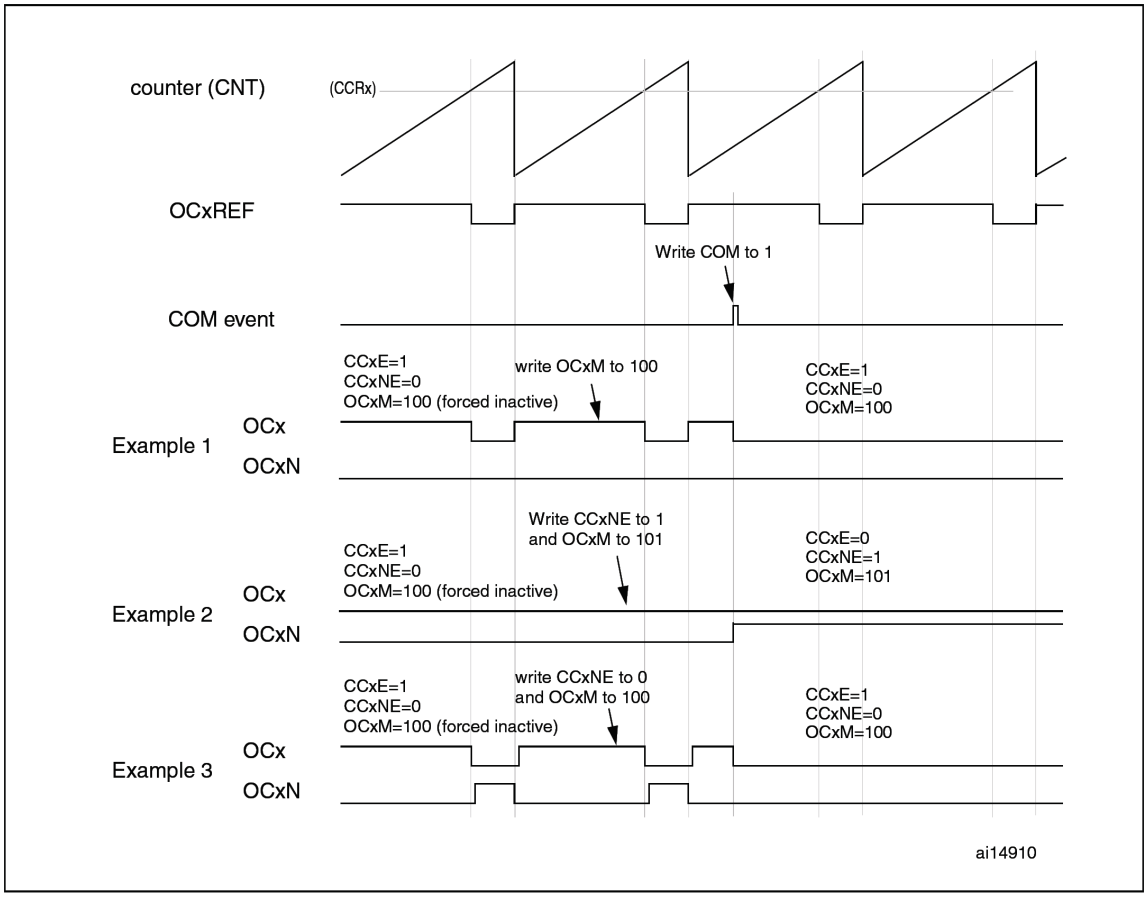
15.3.14 产生六步 PWM 输出

当在一个通道上需要互补输出时，预装载位有 OCxM、CCxE 和 CCxNE。在发生 COM 换相事件时，这些预装载位被传送到影子寄存器位。这样你就可以预先设置好下一步骤配置，并在同一个时刻同时修更改所有通道的配置。COM 可以通过设置 TIMx_EGR 寄存器的 COM 位由软件产生，或在 TRGI 上升沿由硬件产生。

当发生 COM 事件时会设置一个标志位 (TIMx_SR 寄存器中的 COMIF 位)，这时如果已设置了 TIMx_DIER 寄存器的 COMIE 位，则产生一个中断；如果已设置了 TIMx_DIER 寄存器的 COMDE 位，则产生一个 DMA 请求。

图 81 显示当发生 COM 事件时，三种不同配置下 OCx 和 OCxN 输出。

图 81. 产生六步 PWM，使用 COM 的例子 (OSSR=1)



15.3.15 单脉冲模式

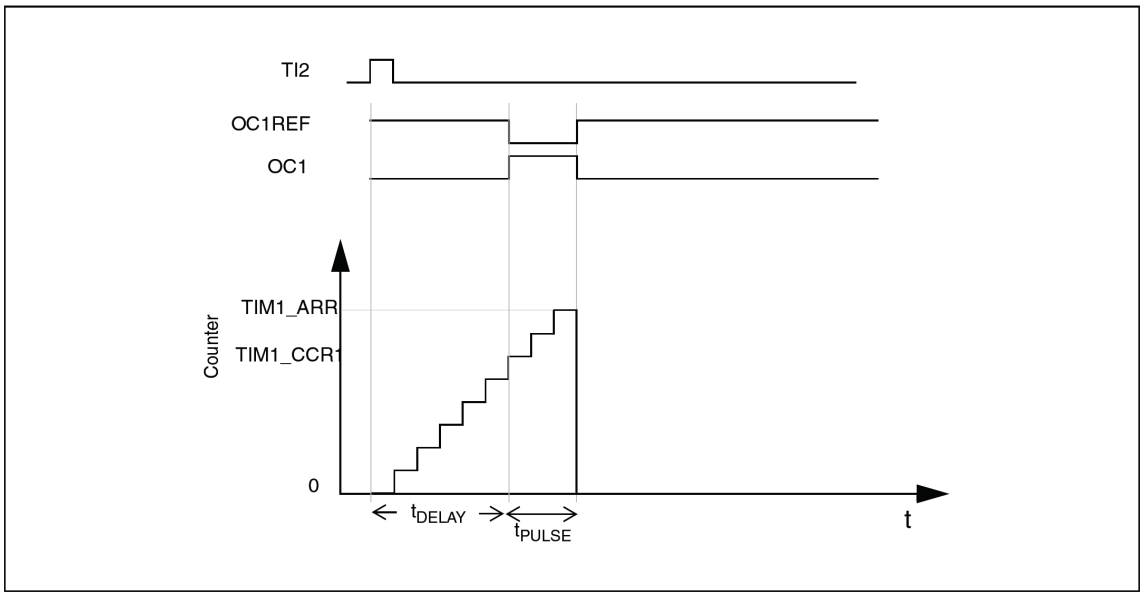
单脉冲模式 (OPM) 是前述众多模式的一个特例。这种模式允许计数器通过响应一个激励来启动，并在一个程序可控的延时之后产生一个脉宽可程序控制的脉冲。

可以通过从模式控制器启动计数器，在输出比较模式或者 PWM 模式下产生波形。设置 TIMx_CR1 寄存器中的 OPM 位将选择单脉冲模式，这样可以让计数器在产生下一个更新事件 UEV 时自动停止。

仅当比较值与计数器的初始值不同时，才能产生一个脉冲。启动之前 (当定时器正在等待触发)，配置必须满足：

- 向上计数方式：计数器 CNT < CCRx ≤ ARR (特别地 , 0 < CCRx)，
- 向下计数方式：计数器 CNT > CCRx。

图 82. 单脉冲模式的例子



例如，你需要在在 OC1 上产生一个长度为 tPULSE 的正脉冲，从 TI2 输入脚上检测到一个上升沿开始，延迟 tDELAY 之后。

假定 TI2FP2 作为触发 1:

- 置 TIMx_CCMR1 寄存器中的 CC2S=01，把 TI2FP2 映像到 TI2。
- 置 TIMx_CCER 寄存器中的 CC2P=0 和 CC2NP=0，使 TI2FP2 能够检测上升沿。
- 置 TIMx_SMCR 寄存器中的 TS=110，TI2FP2 作为从模式控制器的触发 (TRGI)。
- 置 TIMx_SMCR 寄存器中的 SMS=110(触发模式)，TI2FP2 被用来启动计数器。

OPM 的波形由写入比较寄存器决定 (要考虑时钟频率和计数器预分频器)

- t_{DELAY} 由 TIMx_CCR1 寄存器中的值定义。
- t_{PULSE} 由自动装载值和比较值之间的差值定义 (TIMx_ARR - TIMx_CCR1)。
- 假定当发生比较匹配时要产生从 0 到 1 的变换，当计数器达到预装载值时要产生一个从 1 到 0 的变换；首先要置 TIMx_CCMR1 寄存器的 OC1M=111，选择 PWM 模式 2；根据需要有选择地使能预装载寄存器：置 TIMx_CCMR1 中的 OC1PE=1 和 TIMx_CR1 寄存器中的 ARPE；此时必须向 TIMx_CCR1 寄存器中填写比较值，在 TIMx_ARR 寄存器中填写自动装载值，设置 UG 位来产生一个更新事件，然后等待在 TI2 上的一个外部触发事件。本例中，CC1P=0。

在这个例子中，TIMx_CR1 寄存器中的 DIR 和 CMS 位应该置低。

因为只需要一个脉冲，所以必须设置 TIMx_CR1 寄存器中的 OPM=1，在下一个更新事件 (当计数器从自动装载值翻转到 0) 时停止计数。当 TIMx_CR1 寄存器中的 OPM=0 时，进入重复模式。

特殊情况：OCx 快速使能：

在单脉冲模式下，在 TIx 输入脚的边沿检测逻辑设置 CEN 位来启动计数器。然后计数器和比较值间的比较结果驱动输出的转换。但是这些操作需要一定的时钟周期，因此它限制了可得到的最小延时 t_{DELAY}。

如果需要以最小延时输出波形，可以设置 TIMx_CCMRx 寄存器中的 OCxFE 位；此时 OCxREF(和 OCx) 直接响应激励而不再依赖比较的结果，输出的波形与比较匹配时的波形一样。OCxFE 只在通道配置为 PWM 模式 1 和 PWM 模式 2 时起作用。

15.3.16 编码器接口模式

选择编码器接口模式的方法是：如果计数器只在 TI2 的边沿计数，则置 TIMx_SMCR 寄存器中的 SMS=001；如果只在 TI1 边沿计数，则置 SMS=010；如果计数器同时在 TI1 和 TI2 边沿计数，则置 SMS=011。

通过设置 TIMx_CCER 寄存器中的 CC1P 和 CC2P 位，可以选择 TI1 和 TI2 极性；如果需要，还可以对输入滤波器编程。CC1P 和 CC2P 位必须保持为低。

两个输入 TI1 和 TI2 被用来作为增量编码器的接口，参看表 42。假定计数器已经启动 (TIMx_CR1 寄存器中的 CEN=1)，则计数器由每次在 TI1FP1 或 TI2FP2 上的有效跳变驱动。TI1FP1 和 TI2FP2 是 TI1 和 TI2 在通过输入滤波器和极性控制后的信号；如果没有滤波和变相，则 TI1FP1=TI1，TI2FP2=TI2。根据两个输入信号的跳变顺序，产生了计数脉冲和方向信号。依据两个输入信号的跳变顺序，计数器向上或向下计数，同时硬件对 TIMx_CR1 寄存器的 DIR 位修改。 不管计数器是依靠 TI1 还是 TI2 或者同时依靠 TI1 和 TI2 计数，在任一输入端 (TI1 或者 TI2) 的跳变都会重新计算 DIR 位。

编码器接口模式基本上相当于使用了一个带有方向选择的外部时钟。这意味着计数器只在 0 到 TIMx_ARR 寄存器的自动装载值之间连续计数 (根据方向，或是 0 到 ARR 计数，或是 ARR 到 0 计 数)。所以在开始计数之前必须配置 TIMx_ARR；同样，捕获器、比较器、预分频器、重复计数器、触发输出特性等仍工作如常。编码器模式和外部时钟模式 2 不兼容，因此不能同时操作。

在这个模式下，计数器依照增量编码器的速度和方向被自动的修改，因此计数器的内容始终指示着编码器的位置。计数方向与相连的传感器旋转的方向对应。下表列出了所有可能的组合，假设 TI1 和 TI2 不同时变换。

表 42. 计数方向与编码器信号的关系

Active edge	Level on opposite signal (TI1FP1 for TI2, TI2FP2 for TI1)	TI1FP1 signal		TI2FP2 signal	
		Rising	Falling	Rising	Falling
Counting on TI1 only	High	Down	Up	No Count	No Count
	Low	Up	Down	No Count	No Count
Counting on TI2 only	High	No Count	No Count	Up	Down
	Low	No Count	No Count	Down	Up
Counting on TI1 and TI2	High	Down	Up	Up	Down
	Low	Up	Down	Down	Up

一个外部的增量编码器可以直接与 MCU 连接而不需要外部接口逻辑。但是，一般会使用比较器将编码器的差动输出转换到数字信号，这大大增加了抗噪声干扰能力。编码器输出的第三个信号表示机械零点，可以把它连接到一个外部中断输入并触发一个计数器复位。

图 83 是一个计数器操作的实例，显示了计数信号的产生和方向控制。它还显示了当选择了双边沿时，输入抖动是如何被抑制的；抖动可能会在传感器的位置靠近一个转换点时产生。在这个例子中，我们假定配置如下：

- CC1S=’01’ (TIMx_CCMR1 寄存器，IC1FP1 映射到 TI1)
- CC2S=’01’ (TIMx_CCMR2 寄存器，IC2FP2 映射到 TI2)
- CC1P=’0’ (TIMx_CCER 寄存器，IC1FP1 不反相，IC1FP1=TI1)
- CC2P=’0’ (TIMx_CCER 寄存器，IC2FP2 不反相，IC2FP2=TI2)
- SMS=’011’ (TIMx_SMCR 寄存器，所有的输入均在上升沿和下降沿有效).
- CEN=’1’ (TIMx_CR1 寄存器，计数器使能)

图 83. 编码器模式下的计数器操作实例

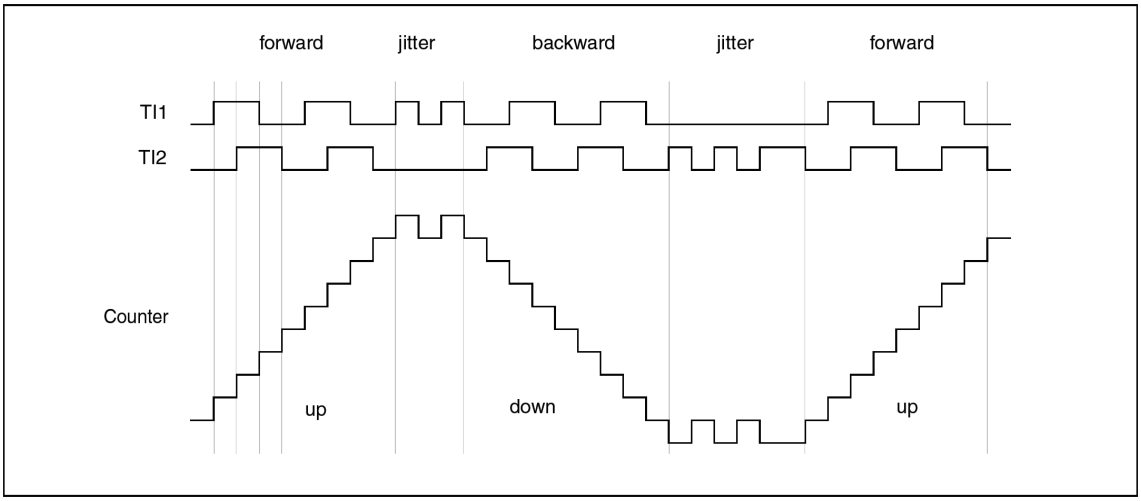
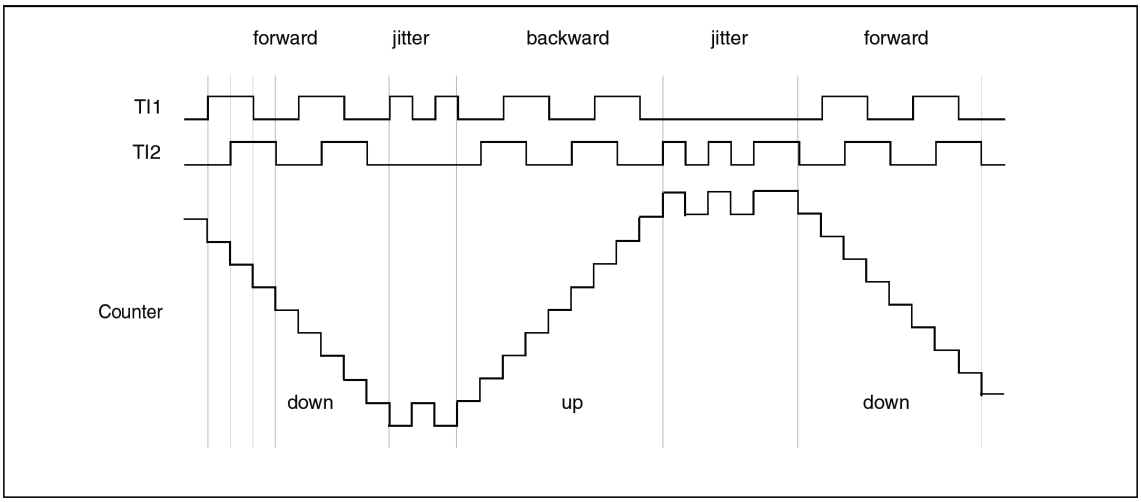


图 84 为当 IC1FP1 极性反相时计数器的操作实例 (CC1P=’1’，其他配置与上例相同)

图 84. IC1FP1 反相的编码器接口模式实例



当定时器配置成编码器接口模式时，用于指示传感器当前位置。配合使用第二个定时器配置为捕获模式，通过测量两个编码器事件的间隔，从而获得动态信息 (速度，加速度，减速度)。编码器输出还可用来指示机械零点。根据两个事件间的间隔，可以按照固定的时间读出计数器。如果有必要，可以把计数器的值锁存到第三个输入捕获寄存器 (由另一个定时器产生的捕获信号必须是周期性)；也可以通过一个由实时时钟产生的 DMA 请求来读取它的值。

15.3.17 定时器输入异或功能

TIMx_CR2 寄存器中的 TI1S 位，允许通道 1 的输入滤波器连接到一个异或门的输出端，异或门的 3 个输入端为 TIMx_CH1、TIMx_CH2 和 TIMx_CH3。

异或输出能够被用于所有定时器的输入功能，如触发或输入捕获。15.3.18 节给出了此特性用于连接霍尔传感器的例子。

15.3.18 与霍尔传感器的接口

使用高级控制定时器 (TIM1) 产生 PWM 信号驱动马达，用另一个 TIMx(TIM2 或 TIM3) 定时器作为“接口定时器”来连接霍尔传感器，请参见图 85。3 个定时器输入脚 (CC1、CC2、CC3) 通过一个异或门连接到 TI1 输入通道 (通过设置 TIMx_CR2 寄存器中的 TI1S 位来选择)，“接口定时器”捕获这个信号。

从模式控制器被配置于复位模式，从输入是 TI1F_ED。这样每当 3 个输入之一变化时，计数器从 0 重新开始计数。由此产生一个由霍尔输入端的任何变化而触发的时间基准。

“接口定时器”模式下，捕获 / 比较通道 1 被配置为捕获模式，捕获信号为 TRC(见 236 页图 68，捕获 / 比较通道 (如：通道 1 输入部分))。捕获值反映了输入端两次变化之间的时间间隔，指示出了马达速度的信息。

“接口定时器”可以用来在输出模式产生一个脉冲，这个脉冲可以 (通过触发一个 COM 事件) 用于改变高级定时器（TIM1）各个通道的属性，TIM1 产生 PWM 信号驱动马达。 因此，“接口定时器”通道必须编程为在一个指定的延时 (输出比较或 PWM 模式) 之后产生一个正脉冲，这个脉冲通过 TRGO 输出被送到高级控制定时器（TIM1）。

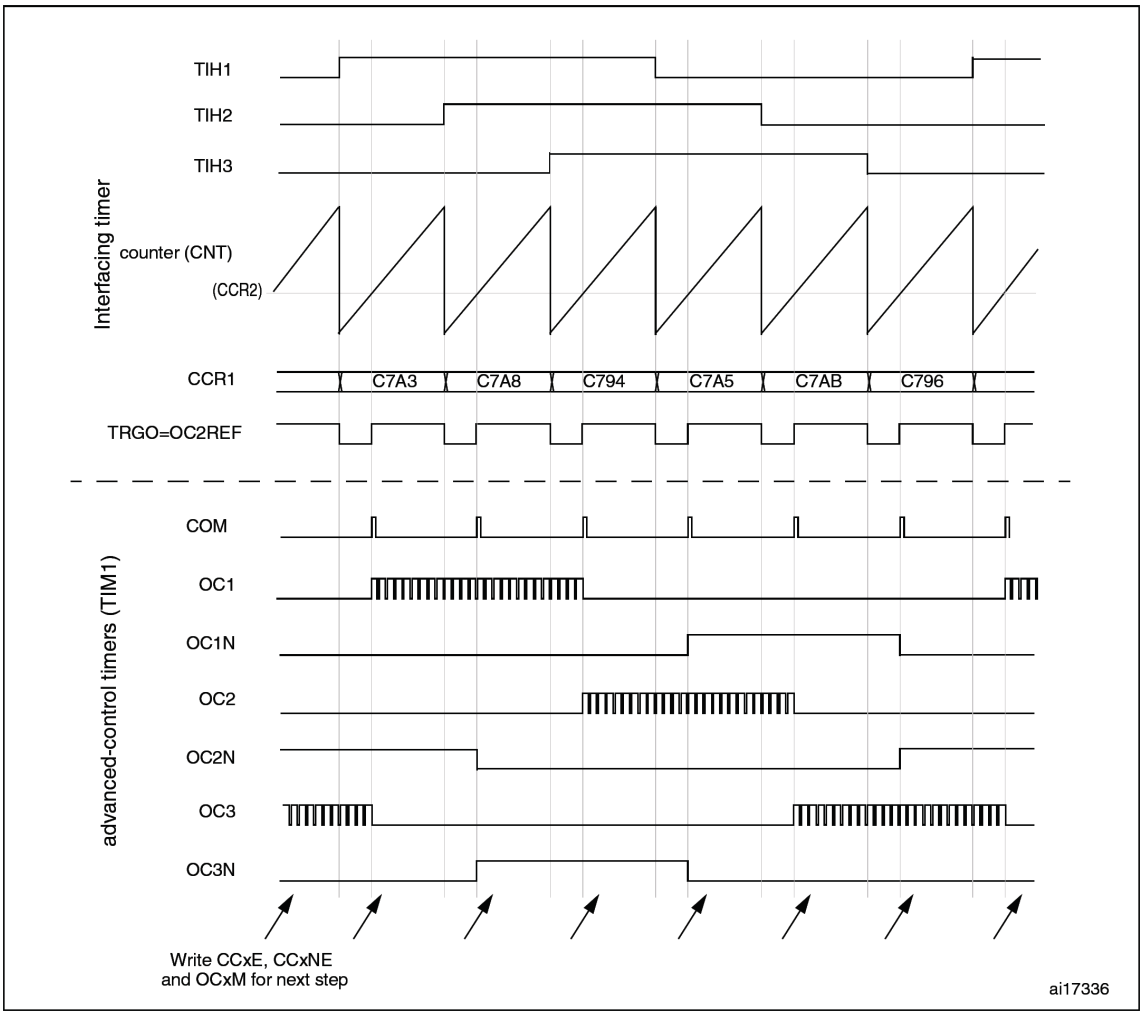
举例：霍尔输入连接到 TIMx 定时器，要求在每次霍尔输入上发生变化之后的一个指定的时刻，改变高级控制定时器 TIMx 的 PWM 配置。

- 置 TIMx_CR2 寄存器的 TI1S 位为’1’，配置三个定时器输入逻辑异或到 TI1 输入，
- 时基编程：置 TIMx_ARR 为其最大值 (计数器必须通过 TI1 的变化清零)。设置预分频器得到一个最大的计数器周期，它长于传感器上的两次变化的时间间隔。
- 设置通道 1 为捕获模式 (选中 TRC)：置 TIMx_CCMR1 寄存器中 CC1S=01，如果需要，还可以设置数字滤波器。
- 设置通道 2 为 PWM 模式 2，带指定的延时：置 TIMx_CCMR1 寄存器中的 OC2M=111 和 CC2S=00。
- 选择 OC2REF 作为 TRGO 上的触发输出：置 TIMx_CR2 寄存器中的 MMS=101。

在高级控制寄存器 TIM1 中，正确的 ITR 输入必须是触发器输入，定时器被编程为产生 PWM 信号，捕获 / 比较控制信号为预装载的 (TIMx_CR2 寄存器中 CCPC=1)，由触发输入控制 COM 事件 (TIMx_CR2 寄存器中 CCUS=1)。在一次 COM 事件后，下一步再写入 PWM 控制位 (CCxE、OCxM)，这可以在处理 OC2REF 上升沿的中断子程序里实现。

图 85 显示了这个实例

图 85. 霍尔传感器接口的实例



15.3.19 TIMx 定时器和外部触发的同步

TIMx 定时器能够在多种模式下和一个外部的触发同步：复位模式、门控模式和触发模式。

从模式：复位模式

在一个触发输入发生事件时，计数器和它的预分频器能够重新被初始化；同时，如果 TIMx_CR1 寄存器的 URS 位为低，还产生一个更新事件 UEV ；然后所有的预装载寄存器 (TIMx_ARR ， TIMx_CCRx) 都被更新。

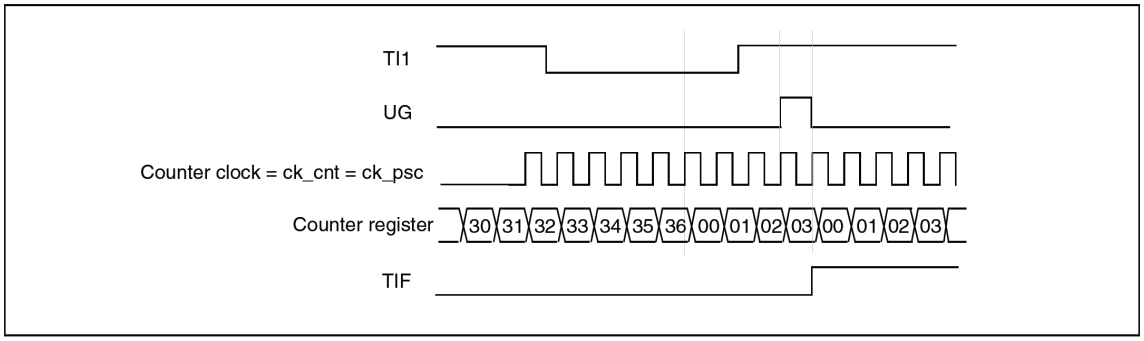
在以下的例子中，TI1 输入端的上升沿导致向上计数器被清零：

- 配置通道 1 以检测 TI1 的上升沿。配置输入滤波器的带宽 (在本例中，不需要任何滤波器，因此保持 IC1F=0000)。触发操作中不使用捕获预分频器，所以不需要配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中 CC1S=01。置 TIMx_CCER 寄存器中 CC1P=0 和 CC1NP=0 以确定极性 (只检测上升沿)。
- 置 TIMx_SMCR 寄存器中 SMS=100，配置定时器为复位模式；置 TIMx_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。
- 置 TIMx_CR1 寄存器中 CEN=1，启动计数器。

计数器开始依据内部时钟计数，然后正常运转直到 TI1 出现一个上升沿；此时，计数器被清零然后从 0 重新开始计数。同时，触发标志 (TIMx_SR 寄存器中的 TIF 位) 被设置，并根据 TIMx_DIER 寄存器中 TIE(中断使能) 位和 TDE(DMA 使能) 位的设置，产生一个中断请求或一个 DMA 请求。

下图显示当自动重装载寄存器 TIMx_ARR=0x36 时的动作。在 TI1 上升沿和计数器的实际复位之间的延时取决于 TI1 输入端的重同步电路。

图 86. 复位模式下的控制电路



从模式：门控模式

按照选中的输入端电平使能计数器。

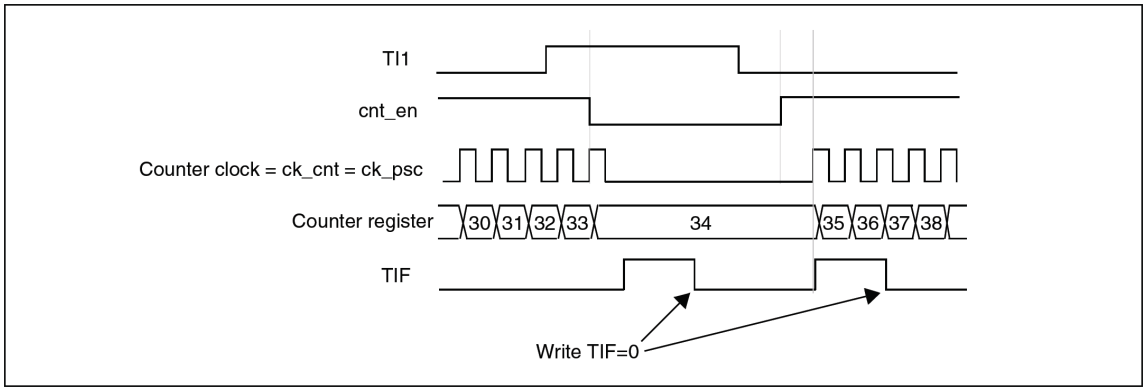
在如下的例子中，计数器只在 TI1 为低时向上计数：

- 配置通道 1 以检测 TI1 上的低电平。配置输入滤波器带宽 (本例中，不需要滤波，所以保持 IC1F=0000)。触发操作中不使用捕获预分频器，所以不需要配置。CC1S 位用于选择输入捕获源，置 TIMx_CCMR1 寄存器中 CC1S=01。置 TIMx_CCER 寄存器中 CC1P=1 和 CC1NP=0 以确定极性 (只检测低电平)。
- 置 TIMx_SMCR 寄存器中 SMS=101，配置定时器为门控模式；置 TIMx_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。
- 置 TIMx_CR1 寄存器中 CEN=1，启动计数器。（在门控模式下，如果 CEN=0，则计数器不能启动，不论触发输入电平如何）

只要 TI1 为低，计数器开始依据内部时钟计数，一旦 TI1 变高则停止计数。当计数器开始或停止时都设置 TIMx_SR 中的 TIF 标置。

TI1 上升沿和计数器实际停止之间的延时取决于 TI1 输入端的重同步电路。

图 87. 门控模式下的控制电路



从模式：触发模式

输入端上选中的事件使能计数器。

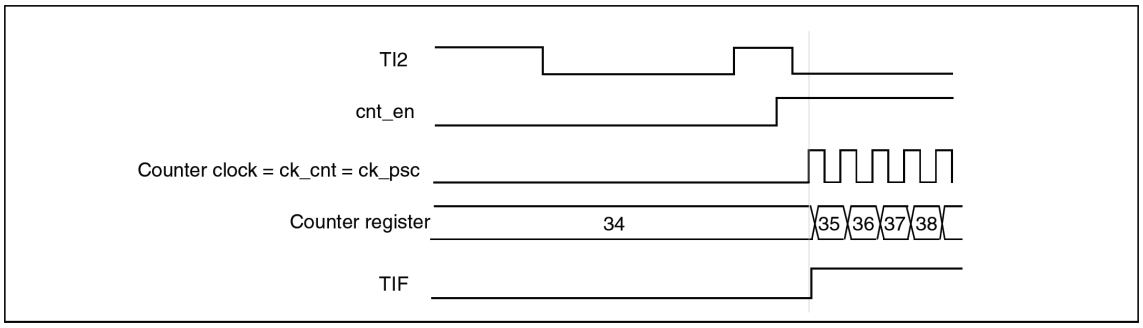
在下面的例子中，计数器在 TI2 输入的上升沿开始向上计数：

- 配置通道 2 检测 TI2 的上升沿。配置输入滤波器带宽 (本例中，不需要任何滤波器，保持 IC2F=0000)。触发操作中不使用捕获预分频器，不需要配置。CC2S 位只用于选择输入捕获源，置 TIMx_CCMR1 寄存器中 CC2S=01。置 TIMx_CCER 寄存器中 CC2P=1 和 CC1NP=0 以确定极性 (只检测低电平)。
- 置 TIMx_SMCR 寄存器中 SMS=110，配置定时器为触发模式；置 TIMx_SMCR 寄存器中 TS=110，选择 TI2 作为输入源。

当 TI2 出现一个上升沿时，计数器按内部时钟开始计数，同时设置 TIF 标志。

TI2 上升沿和计数器启动计数之间的延时，取决于 TI2 输入端的重同步电路。

图 88. 触发器模式下的控制电路



从模式：外部时钟模式 2 + 触发模式

外部时钟模式 2 可以与另一种从模式 (外部时钟模式 1 和编码器模式除外) 一起使用。这时，ETR 信号被用作外部时钟的输入，可以选择另一个输入作为触发输入（在复位模式、门控模式或触发模式）。不建议使用 TIMx_SMCR 寄存器的 TS 位选择 ETR 作为 TRGI。

在下面的例子中，在 TI1 上出现一个上升沿后，向上计数器在 ETR 的每一个上升沿加一：

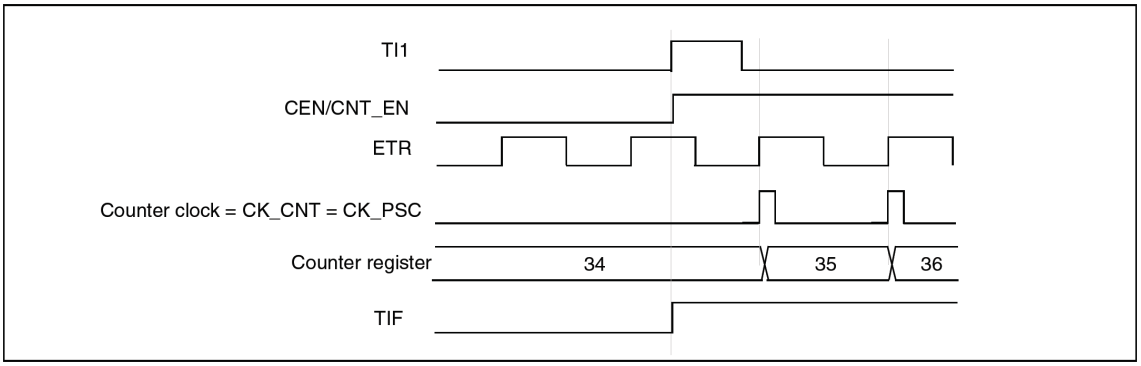
1. 通过 TIMx_SMCR 寄存器配置外部触发输入电路：
   - ETF=0000：没有滤波
   - ETPS=00：不用预分频器
   - ETP=0：检测 ETR 的上升沿，置 ECE=1 使能外部时钟模式 2。

2. 按如下配置通道 1，检测 TI 的上升沿：
   - IC1F=0000：没有滤波
   - 触发操作中不使用捕获预分频器，不需要配置
   - 置 TIMx_CCMR1 寄存器中 CC1S=01，选择输入捕获源
   - 置 TIMx_CCER 寄存器中 CC1P=0 和 CC1NP=0 以确定极性 (只检测上升沿)
3. 置 TIMx_SMCR 寄存器中 SMS=110，配置定时器为触发模式。置 TIMx_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。

当 TI1 上出现一个上升沿时，TIF 标志被设置，计数器开始在 ETR 的上升沿计数。

ETR 信号的上升沿和计数器实际复位间的延时，取决于 ETRP 输入端的重同步电路。

图 89. 外部时钟模式 2 ＋触发模式下的控制电路



15.3.20 定时器同步

TIM 定时器在内部相连，用于定时器同步或链接。详见 319 页 16.3.15 节，定时器同步。

15.3.21 调试模式

当微控制器进入调试模式时 (Cortex-M0 核心停止)，根据 DBG 模块中 DBG_TIMx_STOP 的设置，TIMx 计数器可以或者继续正常操作，或者停止。

15.4 TIM1 寄存器描述

关于在寄存器描述里面所用到的缩写，参见 31 页第 1.1 章。

15.4.1 TIM1 控制寄存器 1 (TIM1_CR1)

偏移地址 : 0x00

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CKD[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:10 保留，必须始终为复位值 .

位 9:8 CKD[1:0]: 时钟分频因子（Clock division）

这 2 位定义在定时器时钟 (CK_INT) 频率、死区时间和由死区发生器与数字滤波器 (ETR,TIx) 所用的采样时钟之间的分频比例。

00：tDTS = tCK_INT

01：tDTS = 2 x tCK_INT

10：tDTS = 4 x tCK_INT

11：保留，不要使用这个配置

位 7 ARPE: 自动重装载预装载允许位（Auto-reload preload enable）

0: TIMx_ARR 寄存器没有缓冲

1: TIMx_ARR 寄存器有缓冲

位 6:5 CMS[1:0]: 选择中央对齐模式 (Center-aligned mode selection)

00：边沿对齐模式。计数器依据方向位 (DIR) 向上或向下计数。

01：中央对齐模式 1。计数器交替地向上和向下计数。配置为输出的通道 (TIMx_CCMRx 寄存器中 CCxS=00) 的输出比较中断标志位，只在计数器向下计数时被设置。

10：中央对齐模式 2。计数器交替地向上和向下计数。配置为输出的通道 (TIMx_CCMRx 寄存器中 CCxS=00) 的输出比较中断标志位，只在计数器向上计数时被设置。

11：中央对齐模式 3。计数器交替地向上和向下计数。配置为输出的通道 (TIMx_CCMRx 寄存器中 CCxS=00) 的输出比较中断标志位，在计数器向上和向下计数时均被设置。

注：在计数器开启时 (CEN=1)，不允许从边沿对齐模式转换到中央对齐模式。

位 4 DIR: 方向 (Direction)

0：计数器向上计数；

1：计数器向下计数。

注：当计数器配置为中央对齐模式或编码器模式时，该位为只读。

位 3 OPM: 单脉冲模式 (One pulse mode)

0：在发生更新事件时，计数器不停止；

1：在发生下一次更新事件 (清除 CEN 位) 时，计数器停止。

位 2 URS: 更新请求源 (Update request source)
软件通过该位选择 UEV 事件的源
0：如果使能了更新中断或 DMA 请求，则下述任一事件产生更新中断或 DMA 请求：
— 计数器溢出 / 下溢
— 设置 UG 位
— 从模式控制器产生的更新
1：如果使能了更新中断或 DMA 请求，则只有计数器溢出 / 下溢才产生更新中断或 DMA 请求。

位 1 UDIS: 禁止更新 (Update disable)
软件通过该位允许 / 禁止 UEV 事件的产生
0：允许 UEV。更新 (UEV) 事件由下述任一事件产生：
— 计数器溢出 / 下溢
— 设置 UG 位
— 从模式控制器产生的更新
具有缓存的寄存器被装入它们的预装载值。
1：禁止 UEV。不产生更新事件，影子寄存器 (ARR、PSC、CCRx) 保持它们的值。如果设置了 UG 位或从模式控制器发出了一个硬件复位，则计数器和预分频器被重新初始化。

位 0 CEN: 使能计数器 (Counter enable)
0：禁止计数器；
1：使能计数器。
注：在软件设置了 CEN 位后，外部时钟、门控模式和编码器模式才能工作。触发模式可以自动 地通过硬件设置 CEN 位。

15.4.2 TIM1 控制寄存器 2 (TIM1_CR2)

偏移地址 : 0x04
复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS[2:0]			CCDS	CCUS	Res.	CCPC
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

位 15 保留，必须始终为复位值 .
位 14 OIS4: 输出空闲状态 4(OC4 输出)
参见 OIS1 位
位 13 OIS3N: 输出空闲状态 3(OC3N 输出)
参见 OIS1N 位
位 12 OIS3: 输出空闲状态 3(OC3 输出)
参见 OIS1 位
位 11 OIS2N: 输出空闲状态 2(OC2N 输出)
参见 OIS1N 位

位 10 OIS2: 输出空闲状态 2(OC2 输出)
参见 OIS1 位

位 9 OIS1N: 输出空闲状态 1(OC1N 输出) (Output Idle state 1)
0：当 MOE=0 时，死区后 OC1N=0；
1：当 MOE=0 时，死区后 OC1N=1。
注：已经设置了 LOCK(TIMx_BKR 寄存器 LOCK 位) 级别 1、2 或 3 后，该位不能被修改。

位 8 OIS1: 输出空闲状态 1(OC1 输出) (Output Idle state 1)
0：当 MOE=0 时，如果完成了 OC1N，则死区后 OC1=0；
1：当 MOE=0 时，如果完成了 OC1N，则死区后 OC1=1。
注：已经设置了 LOCK(TIMx_BKR 寄存器 LOCK 位) 级别 1、2 或 3 后，该位不能被修改。

位 7 TI1S: TI1 选择 (TI1 selection)
0：TIMx_CH1 引脚连到 TI1 输入；
1：TIMx_CH1、TIMx_CH2 和 TIMx_CH3 引脚经异或后连到 TI1 输入。

位 6:4 MMS[1:0]: 主模式选择 (Master mode selection)
这 3 位用于选择在主模式下送到从定时器的同步信息 (TRGO)。可能的组合如下：
000：复位 – TIMx_EGR 寄存器的 UG 位被用于作为触发输出 (TRGO)。如果是触发输入产生的复位 (从模式控制器处于复位模式)，则 TRGO 上的信号相对实际的复位会有一个延迟。
001：使能 – 计数器使能信号 CNT_EN 被用于作为触发输出 (TRGO)。可用于同时启动多个定时器或控制在一段时间内使能从定时器。在门控模式下，计数器使能信号是 CEN 控制位和的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时，TRGO 上会有一个延迟，除非选择了主 / 从模式 (见 TIMx_SMCR 寄存器中 MSM 位的描述)。
010：更新 – 更新事件被选为触发输入 (TRGO)。例如，一个主定时器的时钟可以被用作一个从定时器的预分频器。
011：比较脉冲 – 在发生一次捕获或一次比较成功时，当要设置 CC1IF 标志时 (即使它已经为高)，触发输出送出一个正脉冲 (TRGO)。
100：比较 – OC1REF 信号被用于作为触发输出 (TRGO)。
101：比较 – OC2REF 信号被用于作为触发输出 (TRGO)。
110：比较 – OC3REF 信号被用于作为触发输出 (TRGO)。
111：比较 – OC4REF 信号被用于作为触发输出 (TRGO)。

位 3 CCDS: 捕获 / 比较的 DMA 选择 (捕捉 / 比较 DMA selection)
0：当发生 CCx 事件时，送出 CCx 的 DMA 请求；
1：当发生更新事件时，送出 CCx 的 DMA 请求。

位 2 CCUS: 捕获 / 比较控制更新选择 (捕捉 / 比较 control update selection)
0：如果捕获 / 比较控制位是预装载的 (CCPC=1)，只能通过设置 COMG 位更新它们；
1：如果捕获 / 比较控制位是预装载的 (CCPC=1)，可以通过设置 COMG 位或 TRGI 上的一个上升沿更新它们。
注：该位只对具有互补输出的通道起作用。

位 1 保留，必须始终为复位值 .

位 0 CCPC: 捕获 / 比较预装载控制位 (捕捉 / 比较 preloaded control)
0：CCxE，CCxNE 和 OCxM 位不是预装载的；
1：CCxE，CCxNE 和 OCxM 位是预装载的；设置该位后，只能在发生通信 (COM) 事件（COMG 设置或 TRGI 是检测到上升沿，取决于 CCUS 位）位时被更新。
注：该位只对具有互补输出的通道起作用。

15.4.3 TIM1 从模式控制寄存器 (TIM1_SMCR)

偏移地址 : 0x08
复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			Res.	SMS[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw

位 15 ETP: 外部触发极性 (External trigger polarity)
该位选择是用 ETR 还是 ETR 的反相来作为触发操作
0：ETR 不反相，高电平或上升沿有效；
1：ETR 被反相，低电平或下降沿有效。

位 14 ECE: 外部时钟使能位 (External clock enable)
该位启用外部时钟模式 2
0：禁止外部时钟模式 2；
1：使能外部时钟模式 2。计数器由 ETRF 信号上的任意有效边沿驱动。
注 1：设置 ECE 位与选择外部时钟模式 1 并将 TRGI 连到 ETRF(SMS=111 和 TS=111) 具有相同功效。
注 2：下述从模式可以与外部时钟模式 2 同时使用：复位模式，门控模式和触发模式；但是，这时 TRGI 不能连到 ETRF(TS 位不能是’111’)。
注 3：外部时钟模式 1 和外部时钟模式 2 同时被使能时，外部时钟的输入是 ETRF。

位 13:12 ETPS[1:0]: 外部触发预分频 (External trigger prescaler)
外部触发信号 ETRP 的频率最多是 TIMxCLK 频率的 1/4。当输入较快的外部时钟时，可以使用预分频降低 ETRP 的频率。
00：关闭预分频；
01：ETRP 频率除以 2；
10：ETRP 频率除以 4；
11：ETRP 频率除以 8。

位 11:8 ETF[3:0]: 外部触发滤波 (External trigger filter)

这些位定义了对 ETRP 信号采样的频率和对 ETRP 数字滤波的带宽。数字滤波器是一个事件计数器，它记录到 N 个事件后会产生一个输出的跳变。

0000：无滤波器，以 $f_{SAMPLING}=f_{DTS}$ 采样

0001：采样频率 $f_{SAMPLING}=f_{CK\_INT}$，N=2

0010：采样频率 $f_{SAMPLING}=f_{CK\_INT}$，N=4

0011：采样频率 $f_{SAMPLING}=f_{CK\_INT}$，N=8

0100：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=6

0101：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=8

0110：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=6

0111：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=8

1000：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=6

1001：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=8

1010：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=5

1011：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=6

1100：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=8

1101：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=5

1110： 采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=6

1111：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=8

位 7 MSM: 主 / 从模式 (Master/slave mode)

0：无作用；

1：触发输入 (TRGI) 上的事件被延迟了，以允许在当前定时器与它的从定时器间的完美同步 (通过 TRGO)。这对要求把几个定时器同步到一个单一的外部事件时是非常有用的。

位 6:4 TS[2:0]: 触发选择 (Trigger selection)

这 3 位选择用于同步计数器的触发输入。

000：内部触发 0(ITR0)

001：内部触发 1(ITR1)

010：内部触发 2(ITR2)

011：内部触发 3(ITR3)

100：TI1 的边沿检测器 (TI1F_ED)

101：滤波后的定时器输入 1(TI1FP1)

110：滤波后的定时器输入 2(TI2FP2)

111：外部触发输入 (ETRF)

更多有关 ITRx 的细节，参见 266 页表 74，TIMx 内部触发的连接。

注：这些位只能在未用到 (如 SMS=000) 时被改变，以避免在改变时产生错误的边沿检测。

位 3 保留，必须始终为复位值 .

位 2:0 SMS: 从模式选择 (Slave mode selection)
当选择了外部信号，触发信号 (TRGI) 的有效边沿与选中的外部输入极性相关 (见输入控制寄存器和控制寄存器的说明)
000：关闭从模式 – 如果 CEN=1，则预分频器直接由内部时钟驱动。
001：编码器模式 1 – 根据 TI1FP1 的电平，计数器在 TI2FP2 的边沿向上 / 下计数。
010：编码器模式 2 – 根据 TI2FP2 的电平，计数器在 TI1FP1 的边沿向上 / 下计数。
011：编码器模式 3 – 根据另一个信号的输入电平，计数器在 TI1FP1 和 TI2FP2 的边沿向上 / 下计数。
100：复位模式 – 选中的触发输入 (TRGI) 的上升沿重新初始化计数器，并且产生一次更新寄存器。
101：门控模式 – 当触发输入 (TRGI) 为高时，计数器的时钟开启。一旦触发输入变为低，则计数器停止 (但不复位)。计数器的启动和停止都是受控的。
110：触发模式 – 计数器在触发输入 TRGI 的上升沿启动 (但不复位)，只有计数器的启动是受控的。
111：外部时钟模式 1 – 选中的触发输入 (TRGI) 的上升沿驱动计数器。
注：如果 TI1F_ED 被选为触发输入 (TS=100) 时，不要使用门控模式。这是因为，TI1F_ED 在每 次 TI1F 变化时输出一个脉冲，然而门控模式是要检查触发输入的电平。

表 43. TIMx 内部触发连接

Slave TIM	ITR0 (TS = 000)	ITR1 (TS = 001)	ITR2 (TS = 010)	ITR3 (TS = 011)
TIM1	TIM15	TIM2	TIM3	TIM17

15.4.4 TIM1 DMA/ 中断使能寄存器 (TIM1_DIER)

偏移地址 : 0x0C
复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15 保留，必须始终为复位值 .
位 14 TDE: 允许触发 DMA 请求 (Trigger DMA request enable)
0: 触发 DMA 请求禁止
1: 触发 DMA 请求允许
位 13 COMDE: COM DMA 请求使能
0: COM DMA 请求禁止
1: COM DMA 请求允许
位 12 CC4DE: 捕捉 / 比较 4 DMA 请求使能
0: CC4 DMA 请求禁止
1: CC4 DMA 请求允许

位 11 CC3DE: 捕捉 / 比较 3 DMA 请求使能
0: CC3 DMA 请求禁止
1: CC3 DMA 请求允许
位 10 CC2DE: 捕捉 / 比较 2 DMA 请求使能
0: CC2 DMA 请求禁止
1: CC2 DMA 请求允许
位 9 CC1DE: 捕捉 / 比较 1 DMA 请求使能
0: CC1 DMA 请求禁止
1: CC1 DMA 请求允许
位 8 UDE: 更新 DMA 请求使能
0: 更新 DMA 请求禁止
1: 更新 DMA 请求允许
位 7 BIE: 刹车中断使能
0: 刹车中断禁止
1: 刹车中断允许
位 6 TIE: 触发中断使能
0: 触发中断禁止
1: 触发中断允许
位 5 COMIE: COM 中断使能
0: COM 中断禁止
1: COM 中断允许
位 4 CC4IE: 捕捉 / 比较 4 中断使能
0: CC4 中断禁止
1: CC4 中断允许
位 3 CC3IE: 捕捉 / 比较 3 中断使能
0: CC3 中断禁止
1: CC3 中断允许
位 2 CC2IE: 捕捉 / 比较 2 中断使能
0: CC2 中断禁止
1: CC2 中断允许
位 1 CC1IE: 捕捉 / 比较 1 中断使能
0: CC1 中断禁止
1: CC1 中断允许
位 0 UIE: 更新中断使能
0: 更新中断禁止
1: 更新中断允许

15.4.5 TIM1 状态寄存器 (TIM1_SR)

偏移地址 : 0x10

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CC4OF	CC3OF	CC2OF	CC1OF	Res.	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
			rc_w0	rc_w0	rc_w0	rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

位 15:13 保留，必须始终为复位值 .

位 12 CC4OF: 捕捉 / 比较 4 重复捕捉标志 ， 参考 CC1OF 描述

位 11 CC3OF: 捕捉 / 比较 3 重复捕捉标志 ， 参考 CC1OF 描述

位 10 CC2OF: 捕捉 / 比较 2 重复捕捉标志 ， 参考 CC1OF 描述

位 9 CC1OF: 捕捉 / 比较 1 重复捕捉标志 (Capture/Compare 1 overcapture flag)

仅当相应的通道被配置为输入捕获时，该标记可由硬件置 1。软件写 0 可清除该位。

0：无重复捕获产生；

1：当 CC1IF 的状态已经为’1’，计数器的值被捕获到 TIMx_CCR1 寄存器。

位 8 保留，必须始终为复位值 .

位 7 BIF: 刹车中断标志 (Break interrupt flag)

一旦刹车输入有效，由硬件对该位置’1’。如果刹车输入无效，则该位可由软件清’0’。

0：无刹车事件产生；

1：刹车输入上检测到有效电平。

位 6 TIF: 触发器中断标志 (Trigger interrupt flag)

当发生触发事件 (当从模式控制器处于除门控模式外的其它模式时，在 TRGI 输入端检测到有效 边沿，或门控模式下的任一边沿) 时由硬件对该位置’1’。它由软件清’0’。

0：无触发器事件产生；

1：触发中断等待响应。

位 5 COMIF: COM 中断标志 (COM interrupt flag)

一旦产生 COM 事件 (当捕获 / 比较控制位：CCxE、CCxNE、OCxM 已被更新) 该位由硬件置’1’。 它由软件清’0’。

0：无 COM 事件产生；

1：COM 中断等待响应。

位 4 CC4IF: 捕捉 / 比较 4 中断标志 ， 参考 CC1IF 描述

位 3 CC3IF: 捕捉 / 比较 3 中断标志 ， 参考 CC1IF 描述

位 2 CC2IF: 捕捉 / 比较 2 中断标志 ，参考 CC1IF 描述

位 1 CC1IF: 捕捉 / 比较 1 中断标志 (Capture/Compare 1 interrupt flag)

CC1IF：捕获 / 比较 1 中断标记

如果通道 CC1 配置为输出模式：

当计数器值与比较值匹配时该位由硬件置 1，但在中心对称模式下除外 (参考 TIMx_CR1 寄存器的 CMS 位)。它由软件清’0’ 。

0：无匹配发生；

1：TIMx_CNT 的值与 TIMx_CCR1 的值匹配。 当 TIMx_CCR1 的内容大于 TIMx_APR 的内容时，在向上或向上 / 下计数模式时计数器溢出，或向下计数模式时的计数器下溢条件下，CC1IF 位变高

如果通道 CC1 配置为输入模式：

当捕获事件发生时该位由硬件置’1’，它由软件清’0’或通过读 TIMx_CCR1 清’0’。

0：无输入捕获产生；

1：计数器值被捕获至 TIMx_CCR1(在 IC1 上检测到与所选极性相同的边沿)。

位 0 UIF: 更新中断标志 (Update interrupt flag)

更新中断标记 (Update interrupt flag)

当产生更新事件时该位由硬件置’1’ 。它由软件清’0’ 。

0：无更新事件产生；

1：更新中断等待响应。当寄存器被更新时该位由硬件置’1’ ：

- 若 TIMx_CR1 寄存器的 UDIS=0，当重复计数器数值上溢或下溢时 (重复计数器 =0 时产生更新事件)。
- 若 TIMx_CR1 寄存器的 URS=0、UDIS=0， 当设置 TIMx_EGR 寄存器的 UG=1 时产生更新事件，通过软件对计数器 CNT 重新初始化时。
- 若 TIMx_CR1 寄存器的 URS=0、UDIS=0，当计数器 CNT 被触发事件重新初始化时。(参考 15.4.3: TIM1 从模式控制寄存器 (TIM1_SMCR)

15.4.6 TIM1 事件产生寄存器 (TIM1_EGR)

偏移地址 : 0x14

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
								w	w	w	w	w	w	w	w

位 15:8 保留，必须始终为复位值 .

位 7 BG: 产生刹车事件 (Break generation)

该位由软件置’1’ ，用于产生一个刹车事件，由硬件自动清’0’ 。

0：无动作；

1：产生一个刹车事件。此时 MOE=0、BIF=1，若开启对应的中断和 DMA，则产生相应的中断 和 DMA。

位 6 TG: 触发产生（Trigger generation）
该位由软件置’1’，用于产生一个事件，由硬件自动清’0’。
0：无动作；
1： TIMx_SR 中 TIF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。

位 5 COMG: 捕捉 / 比较控制更新产生（(Capture/Compare control update generation）
该位由软件置’1’，由硬件自动清’0’。
0：无动作；
1：当 CCPC=1，允许更新 CCxE、CCxNE、OCxM 位。
注：该位只对拥有互补输出的通道有效。

位 4 CC4G: 捕捉 / 比较 4 发生
参考 CC1G 描述

位 3 CC3G: 捕捉 / 比较 3 发生
参考 CC1G 描述

位 2 CC2G: 捕捉 / 比较 2 发生
参考 CC1G 描述

位 1 CC1G: 捕捉 / 比较 1 发生 (Capture/Compare 1 generation)
该位由软件置’1’，用于产生一个捕获 / 比较事件，由硬件自动清’0’。
0：无动作；
1：在通道 1 上产生一个捕获 / 比较事件
若通道 CC1 配置为输出：
设置 CC1IF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。
若通道 CC1 配置为输入：
当前的计数器值被捕获至 TIMx_CCR1 寄存器；设置 CC1IF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。若 CC1IF 已经为 1，则设置 CC1OF=1。

位 0 UG: 产生更新事件 (Update generation)
该位由软件置’1’，由硬件自动清’0’。
0：无动作；
1：重新初始化计数器，并产生一个 (寄存器) 更新事件。注意预分频器的计数器也被清’0’(但是预分频系数不变)。若在中心对称模式下或 DIR=0(向上计数) 则计数器被清’0’；若 DIR=1(向下计数) 则计数器取 TIMx_ARR 的值。

15.4.7 TIM1 捕捉 / 比较模式寄存器 1 (TIM1_CCMR1)

偏移地址 : 0x18

复位值 : 0x0000

通道可用于输入 (捕获模式) 或输出 (比较模式)，通道的方向由相应的 CCxS 位定义。该寄存器其它位的作用在输入和输出模式下不同。OCxx 描述了通道在输出模式下的功能，ICxx 描述了通道 在输入模式下的功能。因此必须注意，同一个位在输出模式和输入模式下的功能是不同的。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]				IC1PSC[1:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

输出比较模式 :

位 15 OC2CE: 输出比较 2 清 0 允许

位 14:12 OC2M[2:0]: 输出比较模式 2

位 11 OC2PE: 输出比较 2 预装允许

位 10 OC2FE: 输出比较 2 快速允许

位 9:8 CC2S[1:0]: 捕捉 / 比较 2 选择

该位定义通道的方向 (输入 / 输出)，及输入信号的选择：

00：CC2 通道被配置为输出；

01：CC2 通道被配置为输入，IC2 映射在 TI2 上；

10：CC2 通道被配置为输入，IC2 映射在 TI1 上；

11：CC2 通道被配置为输入，IC2 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)。

注：CC2S 仅在通道关闭时 (TIMx_CCER 寄存器的 CC2E=0) 才是可写的。

位 7 OC1CE: 输出比较 1 清 0 允许

0: OC1REF 不受 ETRF 输入的影响；

1: 一旦检测到 ETRF 输入高电平，清除 OC1REF=0。

位 6:4 OC1M: 输出比较模式 1 (Output Compare 1 mode)

该 3 位定义了输出参考信号 OC1REF 的动作，而 OC1REF 决定了 OC1、OC1N 的值。OC1REF 是高电平有效，而 OC1、OC1N 的有效电平取决于 CC1P、CC1NP 位。

000：冻结。输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 间的比较对 OC1REF 不起作用；

001 ：匹配 时 设置通道 1 为 有效电平。 当计数器 TIMx_CNT 的值与 捕获 / 比较寄存器 1 (TIMx_CCR1) 相同时，强制 OC1REF 为高。

010 ：匹配 时 设置通道 1 为 无效电平。 当计数器 TIMx_CNT 的值与 捕获 / 比较寄存器 1 (TIMx_CCR1) 相同时，强制 OC1REF 为低。

011：翻转。当 TIMx_CCR1=TIMx_CNT 时，翻转 OC1REF 的电平。

100：强制为无效电平。强制 OC1REF 为低。

101：强制为有效电平。强制 OC1REF 为高。

110：PWM 模式 1 － 在向上计数时，一旦 TIMx_CNT<TIMx_CCR1 时通道 1 为有效电平，否则为无效电平；在向下计数时，一旦 TIMx_CNT>TIMx_CCR1 时通道 1 为无效电平 (OC1REF=0)，否则为有效电平 (OC1REF=1)。

111：PWM 模式 2 － 在向上计数时，一旦 TIMx_CNT<TIMx_CCR1 时通道 1 为无效电平，否则为有效电平；在向下计数时，一旦 TIMx_CNT>TIMx_CCR1 时通道 1 为有效电平，否则为无效电平。

注 1：一旦 LOCK 级别设为 3(TIMx_BDTR 寄存器中的 LOCK 位) 并且 CC1S=00(该通道配置成输出) 则该位不能被修改。

注 2：在 PWM 模式 1 或 PWM 模式 2 中，只有当比较结果改变了或在输出比较模式中从冻结模式切换到 PWM 模式时，OC1REF 电平才改变。

注 3：在通道有互补输出时，此时被预装载。如果设置了 TIMx_CR2 中的 CCPC 位，OC1M 的有效位只能在发生 COM 事件时才能从预装位得到新的值。

位 3 OC1PE: 输出比较 1 预装允许 (Output Compare 1 preload enable)

0：禁止 TIMx_CCR1 寄存器的预装载功能，可随时写入 TIMx_CCR1 寄存器，并且新写入的数值立即起作用。

1：开启 TIMx_CCR1 寄存器的预装载功能，读写操作仅对预装载寄存器操作，TIMx_CCR1 的 预装载值在更新事件到来时被加载至当前寄存器中。

注 1：一旦 LOCK 级别设为 3(TIMx_BDTR 寄存器中的 LOCK 位) 并且 CC1S=00(该通道配置成输出) 则该位不能被修改。

注 2：仅在单脉冲模式下 (TIMx_CR1 寄存器的 OPM=1)，可以在未确认预装载寄存器情况下使用 PWM 模式，否则其动作不确定

位 2 OC1FE: 输出比较 1 快速使能 (Output Compare 1 fast enable)

该位用于加快 CC 输出对触发输入事件的响应。

0：CC1 的正常操作依赖于计数器与 CCR1 的值，即使工作于触发器状态。当触发器的输入有一个有效沿时，激活 CC1 输出的最小延时为 5 个时钟周期。

1：输入到触发器的有效沿的作用就象发生了一次比较匹配。因此，OC 被设置为比较电平而与比较结果无关。采样触发器的有效沿和 CC1 输出间的延时被缩短为 3 个时钟周期。 OCFE 只在通道被配置成 PWM1 或 PWM2 模式时起作用。

位 1:0 CC1S: 捕捉 / 比较 1 选择 (Capture/Compare 1 selection)
这 2 位定义通道的方向 (输入 / 输出)，及输入脚的选择：
00：CC1 通道被配置为输出；
01：CC1 通道被配置为输入，IC1 映射在 TI1 上；
10：CC1 通道被配置为输入，IC1 映射在 TI2 上；
11：CC1 通道被配置为输入，IC1 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)。
注：CC1S 仅在通道关闭时 (TIMx_CCER 寄存器的 CC1E=0) 才是可写的。

输入捕捉模式

位 15:12 IC2F: 输入捕捉 2 滤波器
位 11:10 IC2PSC[1:0]: 输入捕捉 2 预分频器
位 9:8 CC2S: 捕捉 / 比较 2 选择
这 2 位定义通道的方向 (输入 / 输出)，及输入脚的选择：
00：CC2 通道被配置为输出；
01：CC2 通道被配置为输入，IC2 映射在 TI2 上；
10：CC2 通道被配置为输入，IC2 映射在 TI1 上；
11：CC2 通道被配置为输入，IC2 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)。
注：CC2S 仅在通道关闭时 (TIMx_CCER 寄存器的 CC2E=0) 才是可写的。
位 7:4 IC1F[3:0]: 输入捕捉 1 滤波器
这几位定义了 TI1 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成，它记录到 N 个事件后会产生一个输出的跳变：
0000：无滤波器，以 f_{DTS} 采样
0001：采样频率 $f_{SAMPLING}=f_{CK\_INT}$，N=2
0010：采样频率 $f_{SAMPLING}=f_{CK\_INT}$，N=4
0011：采样频率 $f_{SAMPLING}=f_{CK\_INT}$，N=8
0100：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=6
0101：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=8
0110：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=6
0111：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=8
1000：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=6
1001：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=8
1010：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=5
1011：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=6
1100：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=8
1101：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=5
1110：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=6
1111：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=8
位 3:2 IC1PSC: 输入捕捉 1 预分频器
这 2 位定义了 CC1 输入 (IC1) 的预分频系数。一旦 CC1E=0(TIMx_CCER 寄存器中)，则预分频器复位。
00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获；
01：每 2 个事件触发一次捕获；
10：每 4 个事件触发一次捕获；
11：每 8 个事件触发一次捕获。

位 1:0 CC1S: 捕捉 / 比较 1 选择
这 2 位定义通道的方向 (输入 / 输出)，及输入脚的选择：
00：CC1 通道被配置为输出；
01：CC1 通道被配置为输入，IC1 映射在 TI1 上；
10：CC1 通道被配置为输入，IC1 映射在 TI2 上；
11：CC1 通道被配置为输入，IC1 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)。
注：CC1S 仅在通道关闭时 (TIMx_CCER 寄存器的 CC1E=0) 才是可写的。

15.4.8 TIM1 捕捉 / 比较模式寄存器 2 (TIM1_CCMR2)

偏移地址 : 0x1C
复位值 : 0x0000
参看以上 CCMR1 寄存器的描述

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4 CE	OC4M[2:0]			OC4 PE	OC4 FE	CC4S[1:0]		OC3 CE.	OC3M[2:0]			OC3 PE	OC3 FE	CC3S[1:0]	
IC4F[3:0]				IC4PSC[1:0]				IC3F[3:0]				IC3PSC[1:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

输出比较模式

位 15 OC4CE: 输出比较 4 清除允许
位 14:12 OC4M: 输出比较 4 模式
位 11 OC4PE: 输出比较 4 预分频允许
位 10 OC4FE: 输出比较 4 快速使能
位 9:8 CC4S: 捕捉 / 比较 4 选择
该 2 位定义通道的方向 (输入 / 输出)，及输入脚的选择：
00：CC4 通道被配置为输出；
01：CC4 通道被配置为输入，IC4 映射在 TI4 上；
10：CC4 通道被配置为输入，IC4 映射在 TI3 上；
11：CC4 通道被配置为输入，IC4 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)。
注：CC4S 仅在通道关闭时 (TIMx_CCER 寄存器的 CC4E=0) 才是可写的。
位 7 OC3CE: 输出比较 3 清除允许
位 6:4 OC3M: 输出比较 3 模式
位 3 OC3PE: 输出比较 3 预分频允许
位 2 OC3FE: 输出比较 3 快速使能
位 1:0 CC3S: 捕捉 / 比较 3 选择
该 2 位定义通道的方向 (输入 / 输出)，及输入脚的选择：
00：CC3 通道被配置为输出；
01：CC3 通道被配置为输入，IC4 映射在 TI3 上；
10：CC3 通道被配置为输入，IC4 映射在 TI4 上；
11：CC3 通道被配置为输入，IC4 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)。
注：CC3S 仅在通道关闭时 (TIMx_CCER 寄存器的 CC3E=0) 才是可写的。

输入捕捉模式

位 15:12 IC4F: 输入捕捉 4 滤波器
位 11:10 IC4PSC: 输入捕捉 4 预分频器
位 9:8 CC4S: 捕捉 / 比较 4 选择
这 2 位定义通道的方向 (输入 / 输出)，及输入脚的选择：
00：CC4 通道被配置为输出；
01：CC4 通道被配置为输入，IC3 映射在 TI4 上；
10：CC4 通道被配置为输入，IC3 映射在 TI3 上；
11：CC4 通道被配置为输入，IC3 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)。
注：CC4S 仅在通道关闭时 (TIMx_CCER 寄存器的 CC4E=0) 才是可写的。
位 7:4 IC3F: 输入捕捉 3 滤波器
位 3:2 IC3PSC: 输入比较 3 预分频器
位 1:0 CC3S: 捕捉 / 比较 3 选择
这 2 位定义通道的方向 (输入 / 输出)，及输入脚的选择：
00：CC3 通道被配置为输出；
01：CC3 通道被配置为输入，IC3 映射在 TI3 上；
10：CC3 通道被配置为输入，IC3 映射在 TI4 上；
11：CC3 通道被配置为输入，IC3 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)。
注：CC3S 仅在通道关闭时 (TIMx_CCER 寄存器的 CC3E=0) 才是可写的。

15.4.9 TIM1 捕捉 / 比较使能寄存器 (TIM1_CCER)

偏移地址 : 0x20
复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:14 保留，必须始终为复位值 .
位 13 CC4P: 捕捉 / 比较 4 输出极性，详见 CC1P
位 12 CC4E: 捕捉 / 比较 4 输出使能，详见 CC1E
位 11 CC3NP: 捕捉 / 比较 3 互补输出极性，详见 CC1NP
位 10 CC3NE: 捕捉 / 比较 3 互补输出使能，详见 CC1NE
位 9 CC3P: 捕捉 / 比较 3 输出极性，详见 CC1P
位 8 CC3E: 捕捉 / 比较 3 输出使能，详见 CC1E

位 7 CC2NP: 捕捉 / 比较 2 互补输出极性，详见 CC1NP

位 6 CC2NE: 捕捉 / 比较 2 互补输出使能，详见 CC1NE

位 5 CC2P: 捕捉 / 比较 2 输出极性，详见 CC1P

位 4 CC2E: 捕捉 / 比较 2 输出使能，详见 CC1E

位 3 CC1NP: 捕捉 / 比较 1 互补输出极性

CC1 通道配置为输出

0：OC1N 高电平有效；

1：OC1N 低电平有效。

CC1 通道配置为输入

本位用于和 CC1P 联合定义 TI1FP1 和 TI2FP1 的极性，详见 CC1P

注：在通道有互补输出时，本位被预装载。如果寄存器 TIMx_CR2 的 CCPC 被置 1，则 CC1NP 的有效位从预装载复制的新值只有在通信事件发生时才起作用。

注：一旦 LOCK 级别 (TIMx_BDTR 寄存器中的 LOCK 位) 设为 3 或 2 且 CC1S=00(通道配置为输出) 则该位不能被修改。

位 2 CC1NE: 捕捉 / 比较 1 互补输出使能

0：关闭－ OC1N 禁止输出，因此 OC1N 的电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值。

1：开启－ OC1N 信号输出到对应的输出引脚，其输出电平依赖于 MOE、OSSI、OSSR 、

OIS1、OIS1N 和 CC1E 位的值。

注：在通道有互补输出时，本位被预装载。如果寄存器 TIMx_CR2 的 CCPC 被置 1，则 CC1NP 的有效位从预装载复制的新值只有在通信事件发生时才起作用。

位 1 CC1P: 捕捉 / 比较 1 输出极性
CC1 通道配置为输出：
0：OC1 高电平有效；
1：OC1 低电平有效。
CC1 通道配置为输入：
CC1NP/CC1P 位选择在触发或捕捉模式下 TI1FP1 和 TI2FP1 的有效极性。
00: 非反相 / 上升沿
电路作用于 TIxFP1 的 上升沿 (在复位、外部时钟或触发模式下的捕捉或触发操作), TIxFP1 非反相 (在门控模式或编码模式)。
01: 反想 / 下降沿
电路作用于 TIxFP1 的 下降沿 (在复位、外部时钟或触发模式下的捕捉或触发操作), TIxFP1 反相 (在门控模式或编码模式)。
00: 保留不用
11: 非反相 / 上升或下降沿
电路作用于 TIxFP1 的 上升沿 和下降沿 (在复位、外部时钟或触发模式下的捕捉或触发操作), TIxFP1 非反相 (在门控模式)。在编码模式下不能使用此配置。
注：在通道有互补输出时，本位被预装载。如果寄存器 TIMx_CR2 的 CCPC 被置 1，则 CC1NP 的有效位从预装载复制的新值只有在通信事件发生时才起作用。
注：一旦 LOCK 级别 (TIMx_BDTR 寄存器中的 LOCK 位) 设为 3 或 2，则该位不能被修改。

位 0 CC1E: 捕捉 / 比较 1 输出使能
CC1 通道配置为输出：
0：关闭－ OC1N 禁止输出，因此 OC1N 的电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值。
1：开启－ OC1N 信号输出到对应的输出引脚，其输出电平依赖于 MOE、OSSI、OSSR 、
OIS1、OIS1N 和 CC1E 位的值。
CC1 通道配置为输入：
本位用于决定是否一个定时器值的捕捉要装载到捕捉 / 比较寄存器 1(TIMx_CCR1)。
0: 捕捉禁止 .
1: 捕捉允许 .
注：在通道有互补输出时，本位被预装载。如果寄存器 TIMx_CR2 的 CCPC 被置 1，则 CC1NP 的有效位从预装载复制的新值只有在通信事件发生时才起作用。

表 44. 带刹车功能的互补输出通道 OCx 和 OCxN 的控制位

控制位					输出状态 (1)	
MOE bit	OSSI bit	OSSR bit	CCxE bit	CCxNE bit	OCx 输出状态	OCxN 输出状态
1	X	0	0	0	输出禁止 (不由定时器驱动) OCx=0, OCx_EN=0	输出禁止 (不由定时器驱动)OCxN=0, OCxN_EN=0
		0	0	1	输出禁止 (不由定时器驱动) OCx=0, OCx_EN=0	OCxREF + 极性 OCxN=OCxREFxor CCxNP, OCxN_EN=1
		0	1	0	OCxREF + 极性 OCx=OCxREF xor CCxP, OCx_EN=1	输出禁止 (不由定时器驱动)OCxN=0, OCxN_EN=0
		0	1	1	OCREF + 极性 + 死区 OCx_ EN=1	OCREF 互补输出 (逻辑非 OCREF) + 极性 + 死区 OCxN_EN=1
		1	0	0	输出禁止 (不由定时器驱动) OCx=CCxP, OCx_EN=0	输出禁止 (不由定时器驱动) OCxN=CCxNP, OCxN_EN=0
		1	0	1	Off-State (输出使能但无效状态) OCx=CCxP, OCx_EN=1	OCxREF + 极性 OCxN=OCxREF xor CCxNP, OCxN_EN=1
		1	1	0	OCxREF + 极性 OCx=OCxREF xor CCxP, OCx_EN=1	Off-State (输出使能但无效状态) OCxN=CCxNP, OCxN_EN=1
		1	1	1	OCREF + 极性 + 死区 OCx_ EN=1	OCREF 互补输出 (逻辑非 OCREF) + 极性 + 死区 OCxN_EN=1
0	0	X	0	0	输出禁止 (不由定时器驱动) OCx=CCxP, OCx_EN=0	输出禁止 (不由定时器驱动) OCxN=CCxNP, OCxN_EN=0
	0		0	1	输出禁止 (不由定时器驱动) 异步 : OCx=CCxP, OCx_EN=0, OCxN=CCxNP, OCxN_EN=0 若时钟存 在：经过一个死区时间后 OCx=OISx，OCxN=OISxN，假设 OISx 与 OISxN 并不都对应 OCx 和 OCxN 的有效电平。	
	0		1	0		
	0		1	1		
	1		0	0	输出禁止 (不由定时器驱动) OCx=CCxP, OCx_EN=0	输出禁止 (不由定时器驱动) OCxN=CCxNP, OCxN_EN=0
	1		0	1	Off-State (输出使能但无效状态) 异步 : OCx=CCxP, OCx_EN=1, OCxN=CCxNP, OCxN_EN=1 若时钟存 在：经过 一个死区 时间后 OCx=OISx ， OCxN=OISxN，假设 OISx 与 OISxN 并不都对应 OCx 和 OCxN 的 有效电平。	
	1		1	0		
	1		1	1		

1. 如果一个通道的 2 个输出都没有使用 (CCxE = CCxNE = 0)，那么 OISx, OISxN, CCxP 和 CCxNP 都必须清零。

注：引脚连接到互补的 OCx 和 OCxN 通道的外部 I/O 引脚的状态，取决于 OCx 和 OCxN 通道状态和 GPIO 寄存器。

15.4.10 TIM1 计数器 (TIM1_CNT)

偏移地址 : 0x24

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 CNT[15:0]: 计数器值

15.4.11 TIM1 预分频器 (TIM1_PSC)

偏移地址 : 0x28

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 PSC[15:0]: 预分频值

预分频器的值 (Prescaler value) 计数器的时钟频率 (CK_CNT) 等于 fCK_PSC/(PSC[15:0]+1)。每次当更新事件产生时，PSC 的值被装入当前预分频器寄存器；更新事件包括计数器被 TIM_EGR 的 UG 位清’0’或被工作在复位模式的从控制器清’0’。

15.4.12 TIM1 自动重装载寄存器 (TIM1_ARR)

偏移地址 : 0x2C

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 ARR[15:0]: 自动重装载的值 (Prescaler value)

ARR 包含了将要装载入实际的自动重装载寄存器的值。

详细参考 221 页的 15.3.1 节：有关 ARR 的更新和动作。

当自动重装载的值为空时，计数器不工作。

15.4.13 TIM1 重复计数寄存器 (TIM1_RCR)

偏移地址 : 0x30

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 15:8 保留，必须始终为复位值 .

位 7:0 REP[7:0]: 重复计数器的值 (Repetition counter value)

预装载寄存器被使能后，这些位允许用户设置比较寄存器的更新速率 (即周期性地从预装载寄存器传输到当前寄存器)；如果允许产生更新中断，则会同时影响产生更新中断的速率。

每次向下计数器 REP_CNT 达到 0，会产生一个更新事件并且计数器 REP_CNT 重新从 REP 值开始计数。由于 REP_CNT 只有在周期更新事件 U_RC 发生时才重载 REP 值，因此对 TIMx_RCR 寄存器写入的新值只在下次周期更新事件发生时才起作用。

这意味着在 PWM 模式中，(REP+1) 对应着：

- 在边沿对齐模式下，PWM 周期的数目；
- 在中心对称模式下，PWM 半周期的数目；

15.4.14 TIM1 捕捉 / 比较寄存器 1 (TIM1_CCR1)

偏移地址 : 0x34

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 CCR1[15:0]: 捕捉 / 比较通道 1 的值

若 CC1 通道配置为输出：

CCR1 决定了装入当前捕获 / 比较 1 寄存器的值 (预装载值)。

如果在 TIMx_CCMR1 寄存器 (OC1PE 位) 中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获 / 比较 1 寄存器中。 当前捕获 / 比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC1 端口上产生输出信号。

若 CC1 通道配置为输入：

CCR1 包含了由上一次输入捕获 1 事件 (IC1) 传输的计数器值。

15.4.15 TIM1 捕捉 / 比较寄存器 2 (TIM1_CCR2)

偏移地址 : 0x38

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 CCR2[15:0]: 捕捉 / 比较通道 2 的值

若 CC2 通道配置为输出：

CCR2 决定了装入当前捕获 / 比较 2 寄存器的值 (预装载值)。

如果在 TIMx_CCMR2 寄存器 (OC2PE 位) 中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获 / 比较 2 寄存器中。 当前捕获 / 比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC2 端口上产生输出信号。

若 CC2 通道配置为输入：

CCR2 包含了由上一次输入捕获 2 事件 (IC2) 传输的计数器值。

CCR2 is the value to be loaded in the actual 捕捉 / 比较 2 寄存器 (preload value).

15.4.16 TIM1 捕捉 / 比较寄存器 3 (TIM1_CCR3)

偏移地址 : 0x3C

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 CCR3[15:0]: 捕捉 / 比较通道 3 的值

若 CC3 通道配置为输出：

CCR3 决定了装入当前捕获 / 比较 3 寄存器的值 (预装载值)。

如果在 TIMx_CCMR3 寄存器 (OC3PE 位) 中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获 / 比较 3 寄存器中。 当前捕获 / 比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC3 端口上产生输出信号。

若 CC3 通道配置为输入：

CCR3 包含了由上一次输入捕获 3 事件 (IC3) 传输的计数器值。

15.4.17 TIM1 捕捉 / 比较寄存器 4 (TIM1_CCR4)

偏移地址 : 0x40

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 CCR4[15:0]: 捕捉 / 比较通道 4 的值

若 CC4 通道配置为输出：

CCR4 决定了装入当前捕获 / 比较 4 寄存器的值 (预装载值)。

如果在 TIMx_CCMR4 寄存器 (OC4PE 位) 中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获 / 比较 4 寄存器中。 当前捕获 / 比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC4 端口上产生输出信号。

若 CC4 通道配置为输入：

CCR4 包含了由上一次输入捕获 4 事件 (IC4) 传输的计数器值。

15.4.18 TIM1 刹车和死区寄存器 (TIM1_BDTR)

偏移地址 : 0x44

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

注： 根据锁定设置，AOE、BKP、BKE、OSSI、OSSR 和 DTG[7:0] 位均可被写保护，有必要在第一次写入 TIMx_BDTR 寄存器时对它们进行配置。

位 15 MOE: 主输出使能 (Main output enable)

一旦刹车输入有效，该位被硬件异步清’0’ 。根据 AOE 位的设置值，该位可以由软件清’0’或被自动置 1。它仅对配置为输出的通道有效。

0：禁止 OC 和 OCN 输出或强制为空闲状态；

1：如果设置了相应的使能位 (TIMx_CCER 寄存器的 CCxE、CCxNE 位)，则开启 OC 和 OCN 输出。

有关 OC/OCN 使能的细节，参见 275 页 15.4.9 节，TIM1 捕获 / 比较使能寄存器 (TIMx_CCER)。

位 14 AOE: 自动输出使能 (Automatic output enable)

0：MOE 只能被软件置’1’；

1：MOE 能被软件置’1’或在下一个更新事件被自动置’1’ (如果刹车输入无效)。

注：一旦 LOCK 级别 (TIMx_BDTR 寄存器中的 LOCK 位) 设为’1’，则该位不能被修改。

位 13 BKP: 刹车输入极性 (Break polarity)
0：刹车输入低电平有效；
1：刹车输入高电平有效。
注：一旦 LOCK 级别 (TIMx_BDTR 寄存器中的 LOCK 位) 设为’ 1’，则该位不能被修改。
注：任何对该位的写操作都需要一个 APB 时钟的延迟以后才能起作用。

位 12 BKE: 刹车使能 (Break enable)
0: 刹车输入禁止 (BRK 和 CCS 时钟失效事件)
1; 刹车输入允许 (BRK 和 CCS 时钟失效事件)
注：一旦 LOCK 级别 (TIMx_BDTR 寄存器中的 LOCK 位) 设为’ 1’，则该位不能被修改。
注：任何对该位的写操作都需要一个 APB 时钟的延迟以后才能起作用。

位 11 OSSR: 运行模式下“关闭状态”选择 (Off-state selection for Run mode)
该位用于当 MOE=1 且通道为互补输出时。没有互补输出的定时器中不存在 OSSR 位。
参考 OC/OCN 使能的详细说明 (275 页 15.4.9 节，TIM1 捕获 / 比较使能寄存器 (TIMx_CCER))。
0：当定时器不工作时，禁止 OC/OCN 输出 (OC/OCN 使能输出信号 =0)；
1：当定时器不工作时，一旦 CCxE=1 或 CCxNE=1，OC/OCN 使能并输出无效电平，然后置 OC/OCN 使能输出信号 =1。
注：一旦 LOCK 级别 (TIMx_BDTR 寄存器中的 LOCK 位) 设为’ 2’，则该位不能被修改。

位 10 OSSI: 运行模式下“空闲状态”选择 (Off-state selection fo Idle mode)
该位用于当 MOE=0 时通道为输出。
参考 OC/OCN 使能的详细说明 (275 页 15.4.9 节，TIM1 捕获 / 比较使能寄存器 (TIMx_CCER))。
0：当定时器不工作时，禁止 OC/OCN 输出 (OC/OCN 使能输出信号 =0)；
1：当定时器不工作时，一旦 CCxE=1 或 CCxNE=1，OC/OCN 被强制输出空闲电平，置 OC/OCN 使能输出信号 =1。
注：一旦 LOCK 级别 (TIMx_BDTR 寄存器中的 LOCK 位) 设为’ 2’，则该位不能被修改。

位 9:8 LOCK[1:0]: 锁定设置 (Lock configuration)
该位为防止软件错误而提供写保护。
00：锁定关闭，寄存器无写保护；
01：锁定级别 1，不能写入 TIMx_BDTR 寄存器的 DTG、BKE、BKP、AOE 位和 TIMx_CR2 寄存器的 OISx/OISxN 位；
10：锁定级别 2，不能写入锁定级别 1 中的各位，也不能写入 CC 极性位 (一旦相关通道通过 CCxS 位设为输出，CC 极性位是 TIMx_CCER 寄存器的 CCxP/CCNxP 位) 以及 OSSR/OSSI 位；
11：锁定级别 3，不能写入锁定级别 2 中的各位，也不能写入 CC 控制位 (一旦相关通道通过 CCxS 位设为输出，CC 控制位是 TIMx_CCMRx 寄存器的 OCxM/OCxPE 位)；
注：在系统复位后，只能写一次 LOCK 位，一旦写入 TIMx_BDTR 寄存器，则其内容冻结直至复位。

位 7:0 DTG[7:0]: 死区发生器设置 (Dead-time generator setup)
这些位定义了插入互补输出之间的死区持续时间。假设 DT 表示其持续时间：
DTG[7:5]=0xx => DT=DTG[7:0] × Tdtg，Tdtg = TDTS；
DTG[7:5]=10x => DT=(64+DTG[5:0]) × Tdtg，Tdtg = 2 × TDTS；
DTG[7:5]=110 => DT=(32+DTG[4:0]) × Tdtg，Tdtg = 8 × TDTS；
DTG[7:5]=111 => DT=(32+DTG[4:0])× Tdtg，Tdtg = 16 × TDTS；
例：若 TDTS = 125ns(8MHZ)，可能的死区时间为：
0 到 15875ns，若步长时间为 125ns；
16us 到 31750ns，若步长时间为 250ns；
32us 到 63us，若步长时间为 1us；
64us 到 126us，若步长时间为 2us；
注：一旦 LOCK 级别 (TIMx_BDTR 寄存器中的 LOCK 位) 设为 1、2 或 3，则不能修改这些位。

15.4.19 TIM1 DMA 控制寄存器 (TIM1_DCR)

偏移地址 : 0x48
复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

位 15:13 保留，必须始终为复位值 .
位 12:8 DBL[4:0]: DMA 连续传送长度 (DMA burst length)
这 5 位定义了 DMA 在连续模式下的传送长度 (当对 TIMx_DMAR 寄存器进行读或写时，定时器则进行一次连续传送)，
00000：1 次传输
00001：2 次传输
00010：3 次传输
....
10001：18 次传输。
位 7:5 保留，必须始终为复位值 .
位 4:0 DBA[4:0]: DMA 基地址 (DMA base address)
这 5 位定义了 DMA 传送的基地址 (当对 TIMx_DMAR 寄存器进行读或写时)，DBA 定义为从 TIMx_CR1 寄存器所在地址开始的偏移量：
例如：
00000：TIMx_CR1，
00001：TIMx_CR2，
00010：TIMx_SMCR，
.....
例：要完成如下的传输：DBL = 7 ，DBA = TIMx_CR1
此时传送从 TIMx_CR1 的地址开始向 / 自连续 7 个寄存器进行操作。

15.4.20 TIM1 全部传输时 DMA 地址 (TIM1_DMAR)

偏移地址 : 0x4C

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAB[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 DMAB[15:0]: DMA 并发（连续）传送寄存器

对 TIMx_DMAR 寄存器的读或写会导致对以下地址所在寄存器的访问：

（TIMx_CR1 地址） + （DBA + DMA 索引）x 4，

其中： “TIMx_CR1 地址”是控制寄存器 1(TIMx_CR1) 所在的地址；“DBA”是 TIMx_DCR 寄存器中定义的基地址；“DMA 索引”是由 DMA 自动控制的偏移量，它取决于 TIMx_DCR 寄存器中定义的 DBL。

如何使用 DMA 并发操作的例子

本例中使用定时器 DMA 的并发功能，将 CCRx 寄存器 (x = 2, 3, 4) 的内容以半字方式进行 DMA 传输，更新到 CCRx 寄存器。

按如下步骤进行操作：

1. 配置相关的 DMA 通道 :
   - DMA 通道设备地址为 DMAR 寄存器地址
   - DMA 通道存储器地址为包含要通过 DMA 传送到 CCRx 寄存器的数据 RAM 缓冲区地址
   - 传送数据数量 = 3 (见下面的注).
   - 通告模式禁止 .
2. 配置 DCR 寄存器的 DBA 和 DBL 位 : DBL = 3 次传送 , DBA = 0xE.
3. 使能 TIMx 更新 DMA 请求 (设置 DIER 寄存器的 UDE 位).
4. 使能 TIMx
5. 使能 DMA 通道

注： 在本例中所有 CCRx 寄存器被一次性全部更新。如果需要更新 CCRx 寄存器两次，传送的数据数量应该是 6 ，而 RAM 缓冲区要包含 data1, data2, data3, data4, data5 和 data6。数据按如下过程被传送到 CCRx 寄存器：在第一个更新 DMA 请求时，data1 被传送到 CCR2, data2 被传送到 CCR3, data3 被传送到 CCR4，在第二个 DMA 更新中断请求时，data4 被传送到 CCR2, data5 被传送到 CCR3，data6 被传送到 CCR4

15.4.21 TIM1 寄存器映射

下表中将 TIM1 的所有寄存器映射到一个 16 位可寻址 (编址) 空间。

表 45. TIM1 寄存器映射与复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**TIM1_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKD [1:0]		ARPE	CMS [1:0]		DIR	OPM	URS	UDIS	CEN
	Reset value																							0	0	0	0	0	0	0	0	0	0
0x04	**TIM1_CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS[2:0]			CCDS	CCUS	Res.	CCPC
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0		0
0x08	**TIM1_SMCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETP	ECE	ETPS [1:0]		ETF[3:0]				MSM	TS[2:0]			Res.	SMS[2:0]		
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0
0x0C	**TIM1_DIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	**TIM1_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC4OF	CC3OF	CC2OF	CC1OF	Res.	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
	Reset value																				0	0	0	0		0	0	0	0	0	0	0	0
0x14	**TIM1_EGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BG	TG	COM	CC4G	CC3G	CC2G	CC1G	UG
	Reset value																									0	0	0	0	0	0	0	0
0x18	**TIM1_CCMR1** Output compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2CE	OC2M [2:0]			OC2PE	OC2FE	CC2S [1:0]		OC1CE	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIM1_CCMR1 Input capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC2F[3:0]				IC2 PSC [1:0]		CC2S [1:0]		IC1F[3:0]				IC1 PSC [1:0]		CC1S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	**TIM1_CCMR2** Output compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	O24CE	OC4M [2:0]			OC4PE	OC4FE	CC4S [1:0]		OC3CE	OC3M [2:0]			OC3PE	OC3FE	CC3S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIM1_CCMR2 Input capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC4F[3:0]				IC4 PSC [1:0]		CC4S [1:0]		IC3F[3:0]				IC3 PSC [1:0]		CC3S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	**TIM1_CCER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	**TIM1_CNT**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	**TIM1_PSC**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**TIM1_ARR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	**TIM1_RCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[7:0]							
	Reset value																									0	0	0	0	0	0	0	0

表 45. TIM1 寄存器映射与复位值 (续)

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x34	**TIM1_CCR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	**TIM1_CCR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR2[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	**TIM1_CCR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR3[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	**TIM1_CCR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR4[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	**TIM1_BDTR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK [1:0]		DT[7:0]							
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	**TIM1_DCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
	Reset value																				0	0	0	0	0				0	0	0	0	0
0x4C	**TIM1_DMAR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAB[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

关于寄存器的起始地址，请参见 35 页 2.2.2 节

16 通用定时器 (TIM2 和 TIM3)

16.1 TIM2 和 TIM3 简介

通用定时器由一个 16 位或 32 位的自动装载计数器组成，它由一个可编程的预分频器驱动。它适合多种用途，包含测量输入信号的脉冲宽度 (输入捕获)，或者产生输出波形 (输出比较和 PWM)。

使用定时器预分频器和 RCC 时钟控制预分频器，可以实现脉冲宽度和波形周期从几个微秒到几个毫秒的调节。

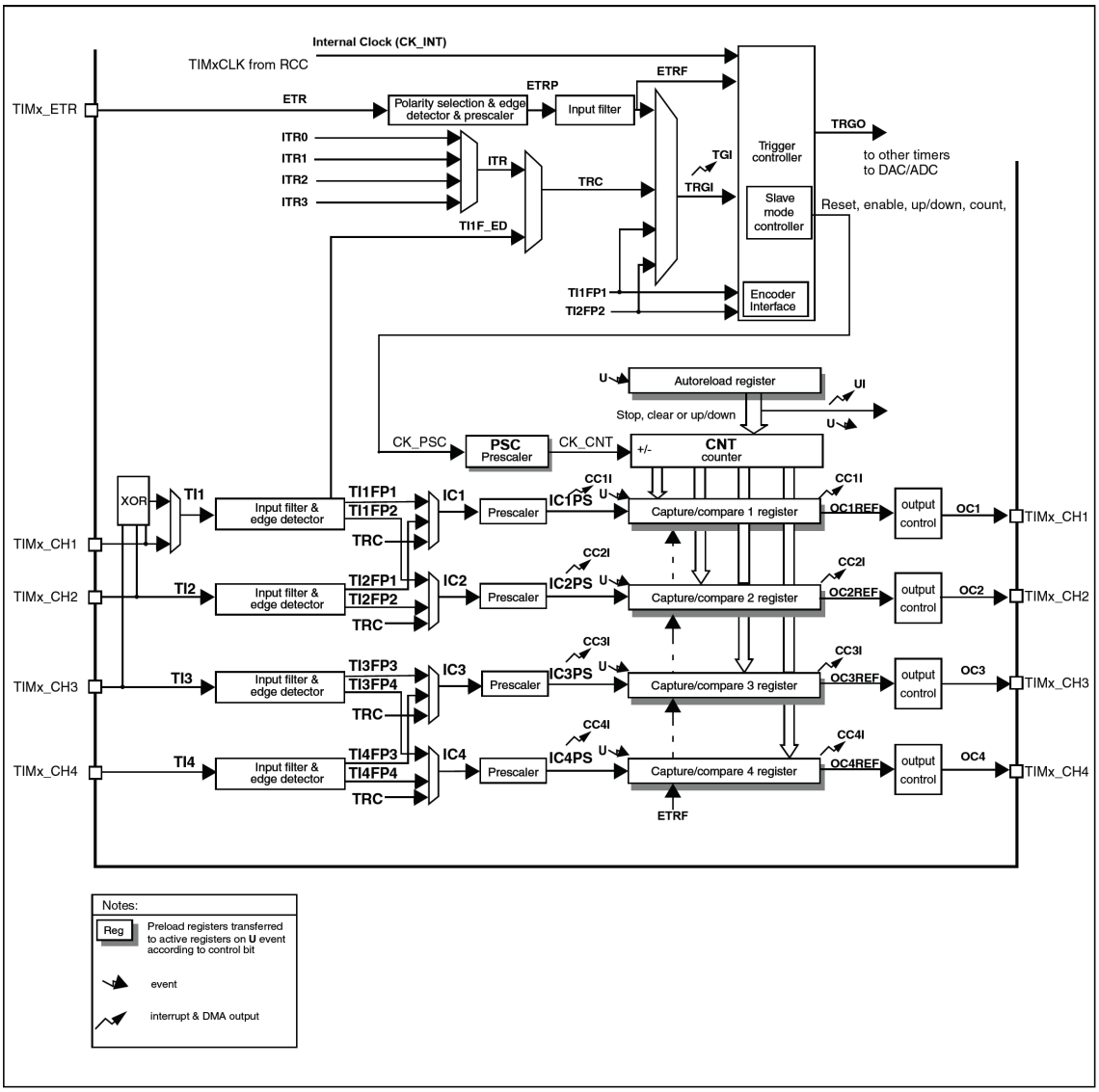
通用定时器是完全独立的，它们不共享任何资源。它们可以同步操作，具体描述参看 16.3.15 节。

16.2 TIM2 和 TIM3 主要功能

通用 TIMx 定时器功能包括：

- 16 位 (TIM3) 或 32-bit (TIM2) 向上、向下、向上 / 向下自动装载计数器
- 16 位可编程 (可以实时修改) 预分频器，计数器时钟频率的分频系数为 1 ～ 65536 之间的任意数值
- 4 个独立通道：
  - 输入捕获
  - 输出比较
  - PWM 生成 (边缘或中间对齐模式)
  - 单脉冲模式输出
- 使用外部信号控制定时器和定时器互连的同步电路
- 如下事件发生时产生中断 /DMA：
  - 更新：计数器向上溢出 / 向下溢出，计数器初始化 (通过软件或者内部 / 外部触发)
  - 触发事件 (计数器启动、停止、初始化或者由内部 / 外部触发计数)
  - 输入捕获
  - 输出比较
- 支持针对定位的增量 (正交) 编码器和霍尔传感器电路
- 触发输入作为外部时钟或者按周期的电流管理

图 90. 通用定时器框图 (TIM2 和 TIM3)



16.3 TIM2 和 TIM3 功能描述

16.3.1 时基单元

可编程通用定时器的主要部分是一个 16/32 位计数器和与其相关的自动装载寄存器。这个计数器可以向上计数、向下计数或者向上向下双向计数。此计数器时钟由预分频器分频得到。计数器、自动装载寄存器和预分频器寄存器可以由软件读写，在计数器运行时仍可以读写。

时基单元包含：

- 计数器寄存器 (TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动装载寄存器 (TIMx_ARR)

自动装载寄存器是预先装载的，写或读自动重装载寄存器将访问预装载寄存器。根据在 TIMx_CR1 寄存器中的自动装载预装载使能位 (ARPE) 的设置，预装载寄存器的内容被立即或在 每次的更新事件 UEV 时传送到影子寄存器。当计数器达到溢出条件 (向下计数时的下溢条件) 并当 TIMx_CR1 寄存器中的 UDIS 位等于’0’时，产生更新事件。更新事件也可以由软件产生。随后会详细描述每一种配置下更新事件的产生。

计数器由预分频器的时钟输出 CK_CNT 驱动，仅当设置了计数器 TIMx_CR1 寄存器中的计数器使能位 (CEN) 时，CK_CNT 才有效。(有关计数器使能的细节，请参见控制器的从模式描述)。
注：真正的计数器使能信号 CNT_EN 是在 CEN 的一个时钟周期后被设置。

预分频器描述

预分频器可以将计数器的时钟频率按 1 到 65536 之间的任意值分频。它是基于一个 (在 TIMx_PSC 寄存器中的)16-32 位寄存器控制的 16 位计数器。这个控制寄存器带有缓冲器，它能够在工作时被改变。新的预分频器参数在下一次更新事件到来时被采用。

图 91 和图 92 给出了在预分频器运行时更改计数器参数的例子。

图 91. 当预分频器的参数从 1 变到 2 时，计数器的时序图

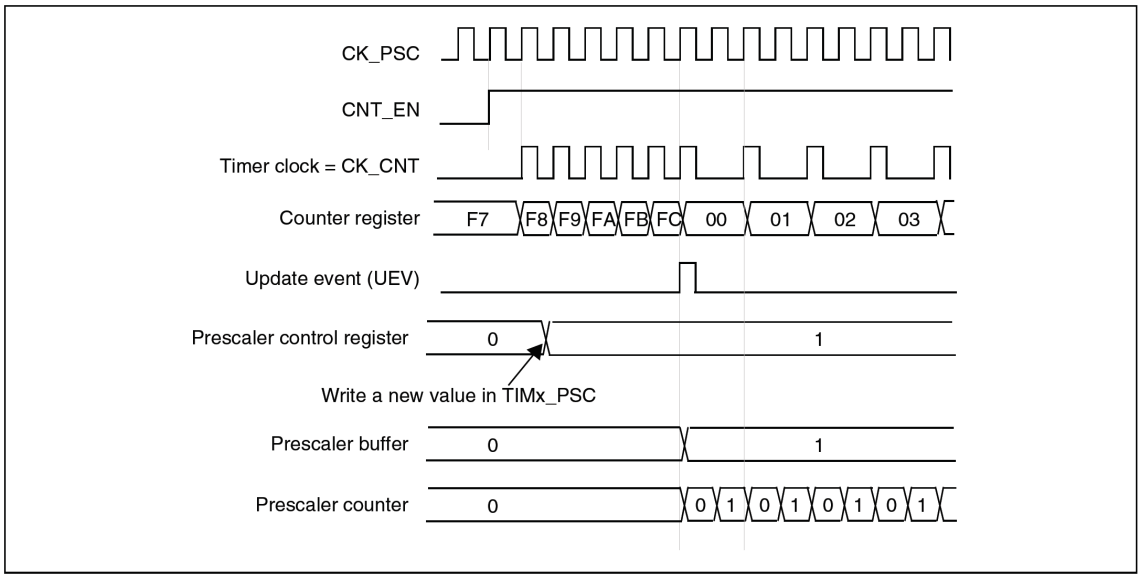
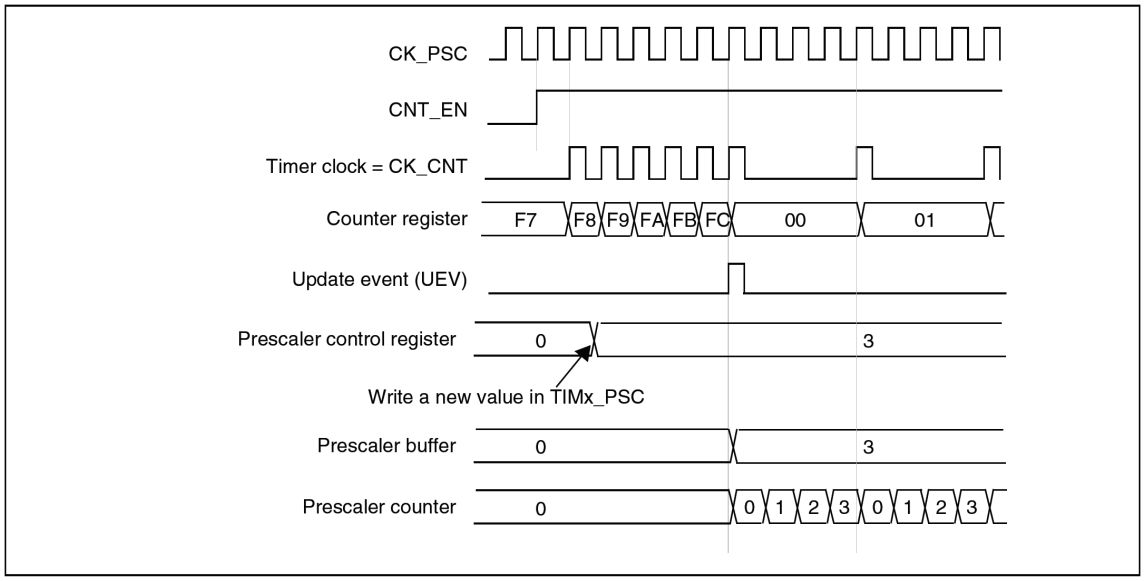


图 92. 当预分频器的参数从 1 变到 4 时，计数器的时序图



16.3.2 计数模式

向上计数模式

在向上计数模式中，计数器从 0 计数到自动加载值 (TIMx_ARR 计数器的内容)，然后重新从 0 开始计数并且产生一个计数器溢出事件。

每次计数器溢出时可以产生更新事件，在 TIMx_EGR 寄存器中 (通过软件方式或者使用从模式控 制器) 设置 UG 位也同样可以产生一个更新事件。

设置 TIMx_CR1 寄存器中的 UDIS 位，可以禁止更新事件；这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。在 UDIS 位被清’0’之前，将不产生更新事件。但是在应该产生更新事件时，计数器仍会被清’0’，同时预分频器的计数也被请 0(但预分频系数不变)。此外，如果设置了 TIMx_CR1 寄存器中的 URS 位 (选择更新请求)，设置 UG 位将产生一个更新事件 UEV，但硬件不设置 UIF 标志 (即不产生中断或 DMA 请求)；这是为了避免在捕获模式下清除计数器时，同时产生更新和捕获中断。

当发生一个更新事件时，所有的寄存器都被更新，硬件同时 (依据 URS 位) 设置更新标志位 (TIMx_SR 寄存器中的 UIF 位)。

- 预分频器的缓冲区被置入预装载寄存器的值 (TIMx_PSC 寄存器的内容)。
- 自动装载影子寄存器被重新置入预装载寄存器的值 (TIMx_ARR)。

下图给出一些例子，当 TIMx_ARR=0x36 时计数器在不同时钟频率下的动作。

图 93. 计数器时序图，内部时钟分频因子为 1

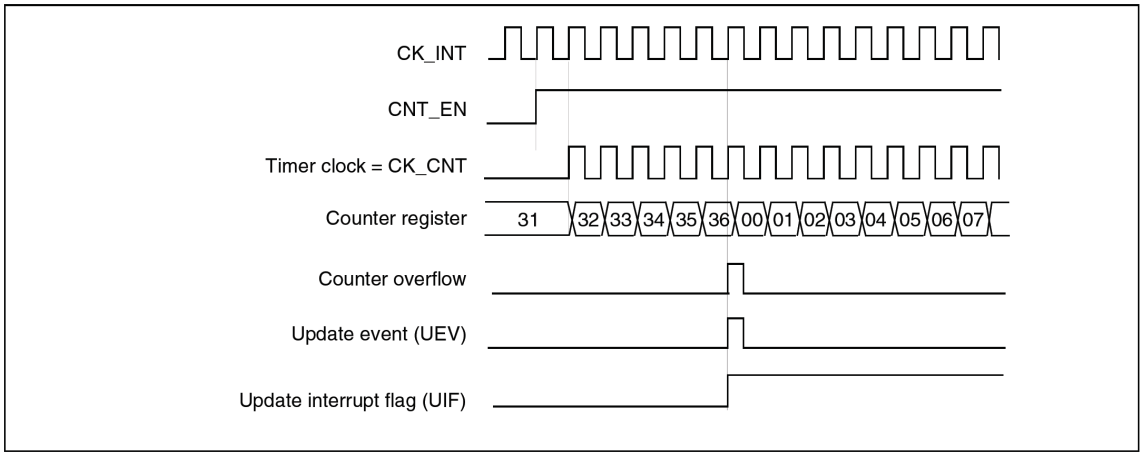


图 94. 计数器时序图，内部时钟分频因子为 2

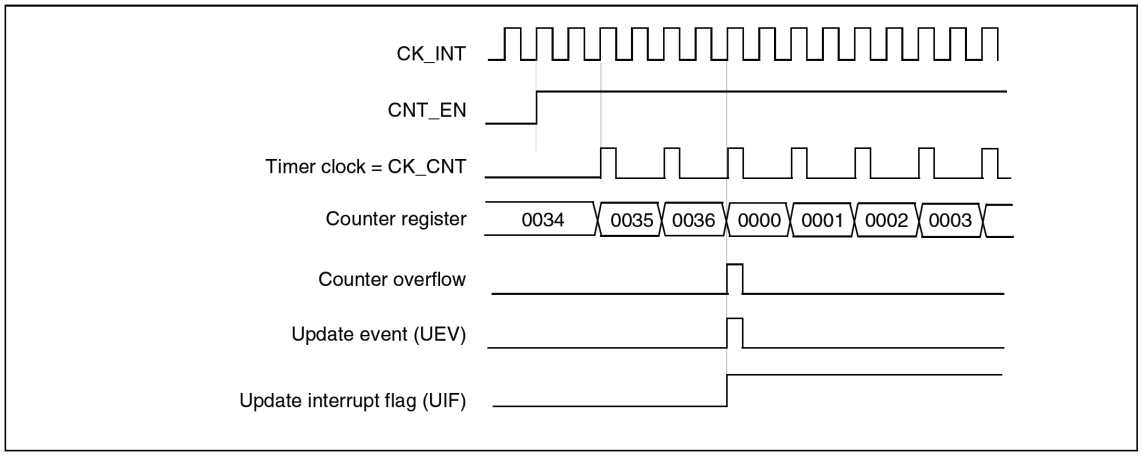


图 95. 计数器时序图，内部时钟分频因子为 4

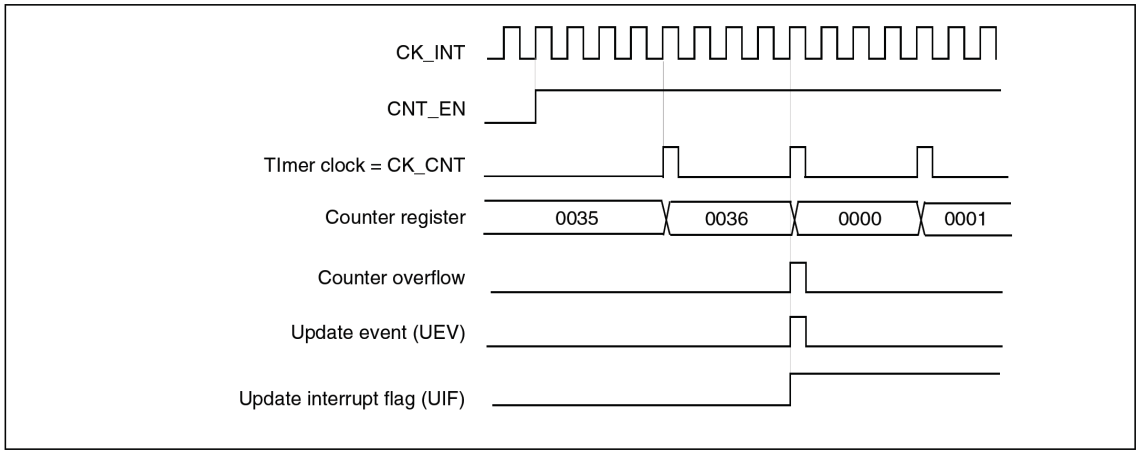


图 96. 计数器时序图，内部时钟分频因子为 N

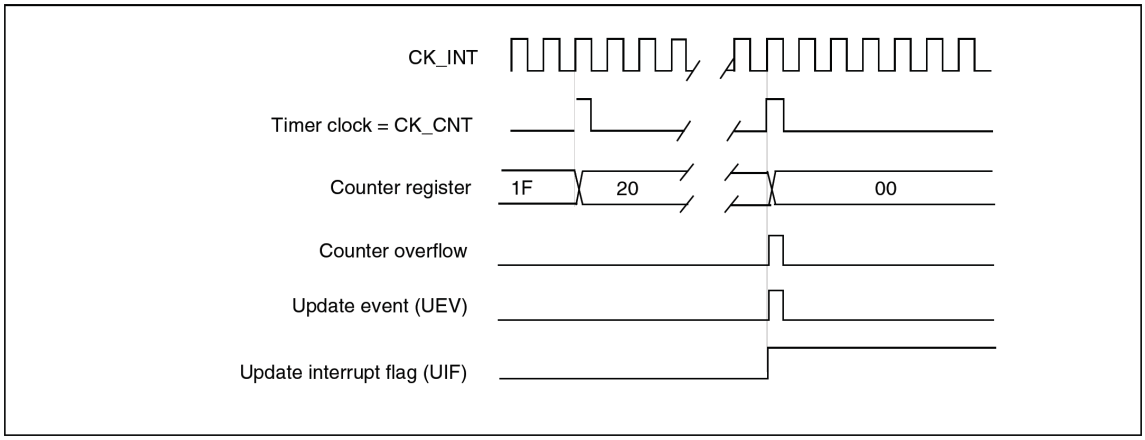


图 97. 计数器时序图，当 ARPE=0 时的更新事件 (TIMx_ARR 没有预装入)

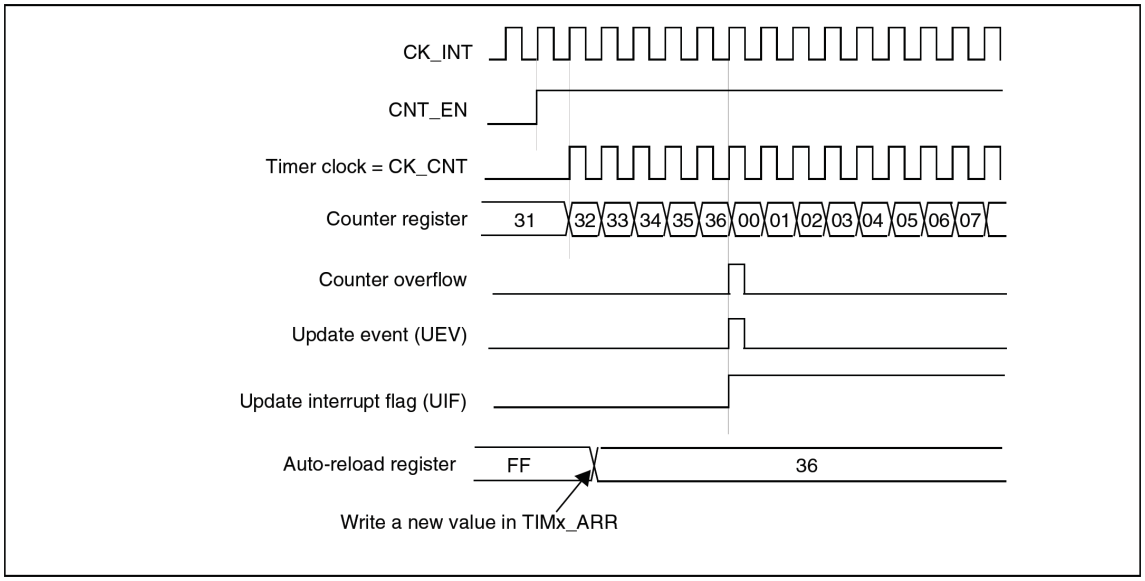
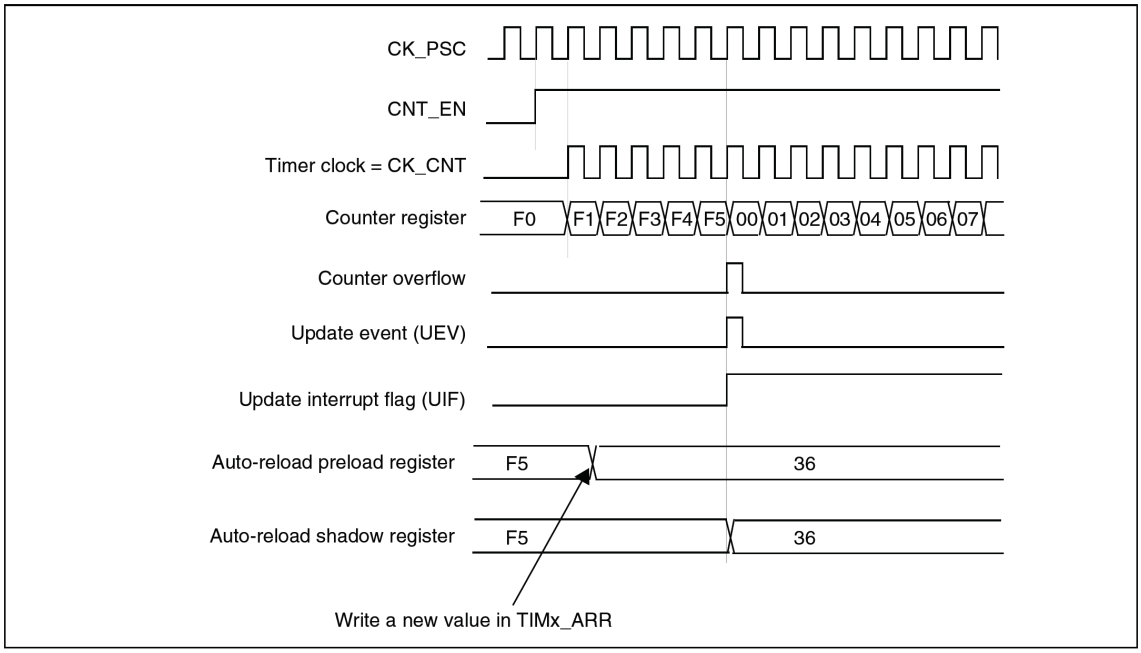


图 98. 计数器时序图，当 ARPE=1 时的更新事件 (预装入了 TIMx_ARR)



向下计数模式

在向下模式中，计数器从自动装入的值 (TIMx_ARR 计数器的值) 开始向下计数到 0，然后从自动装入的值重新开始并且产生一个计数器向下溢出事件。

每次计数器溢出时可以产生更新事件，在 TIMx_EGR 寄存器中 (通过软件方式或者使用从模式控 制器) 设置 UG 位，也同样可以产生一个更新事件。

设置 TIMx_CR1 寄存器的 UDIS 位可以禁止 UEV 事件。这样可以避免向预装载寄存器中写入新值时更新影子寄存器。因此 UDIS 位被清为’0’之前不会产生更新事件。然而，计数器仍会从当前自动加载值重新开始计数，同时预分频器的计数器重新从 0 开始 (但预分频系数不变)。

此外，如果设置了 TIMx_CR1 寄存器中的 URS 位 (选择更新请求) ，设置 UG 位将产生一个更新事件 UEV 但不设置 UIF 标志 (因此不产生中断和 DMA 请求)，这是为了避免在发生捕获事件并清除计数器时，同时产生更新和捕获中断。

当发生更新事件时，所有的寄存器都被更新，并且 (根据 URS 位的设置) 更新标志位 (TIMx_SR 寄 存器中的 UIF 位) 也被设置。

- 预分频器的缓存器被置入预装载寄存器的值 (TIMx_PSC 寄存器的值)。
- 当前的自动加载寄存器被更新为预装载值 (TIMx_ARR 寄存器中的内容)。注：自动装载在计数器重载入之前被更新，因此下一个周期将是预期的值。

以下是一些当 TIMx_ARR=0x36 时，计数器在不同时钟频率下的操作例子。

图 99. 计数器时序图，内部时钟分频因子为 1

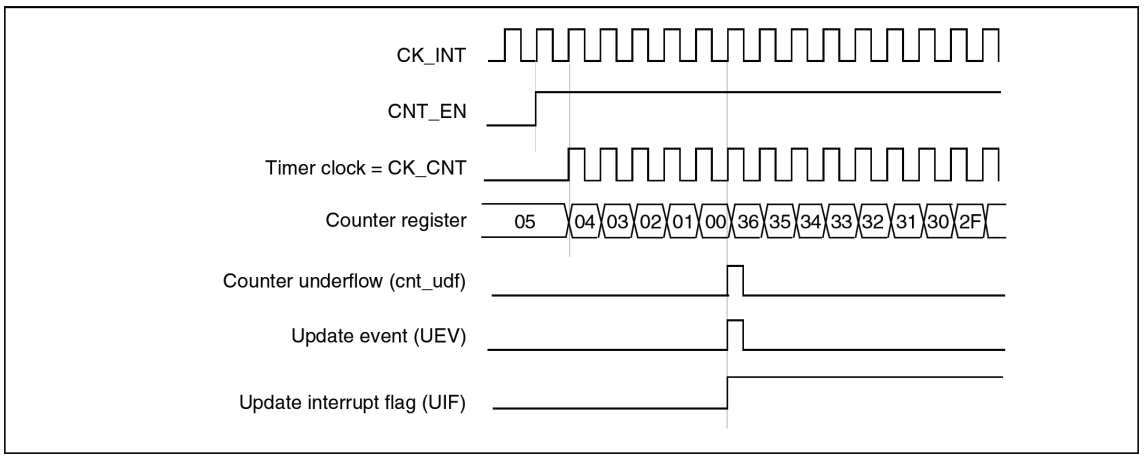


图 100. 计数器时序图，内部时钟分频因子为 2

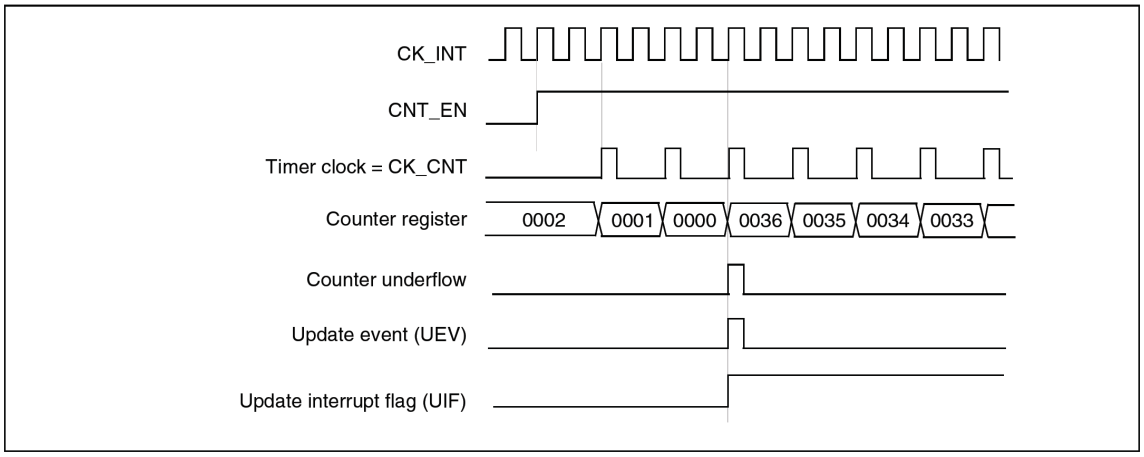


图 101. 计数器时序图，内部时钟分频因子为 4

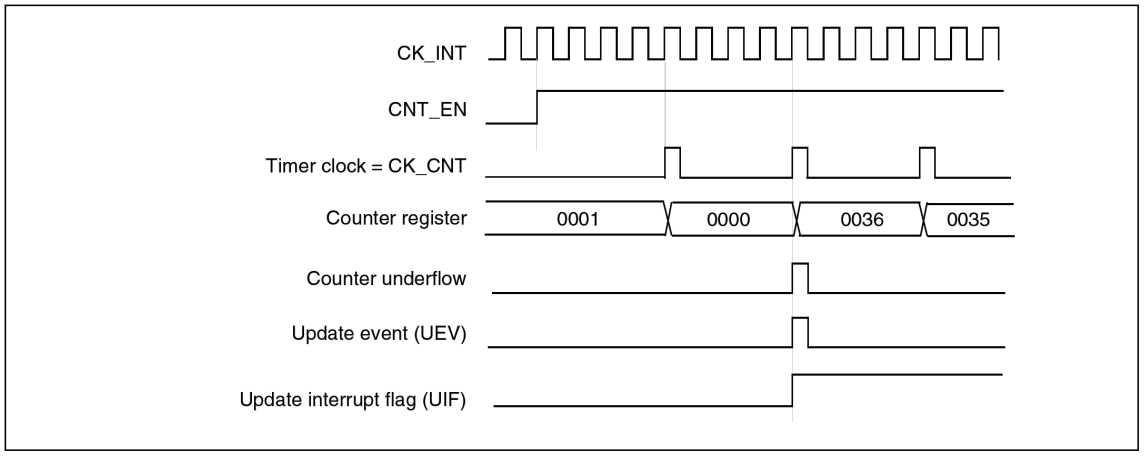


图 102. 计数器时序图，内部时钟分频因子为 N

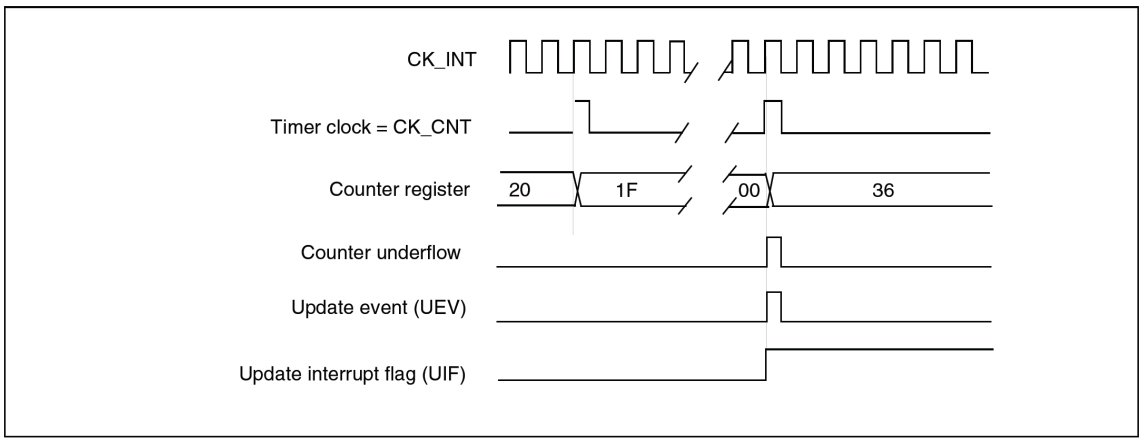
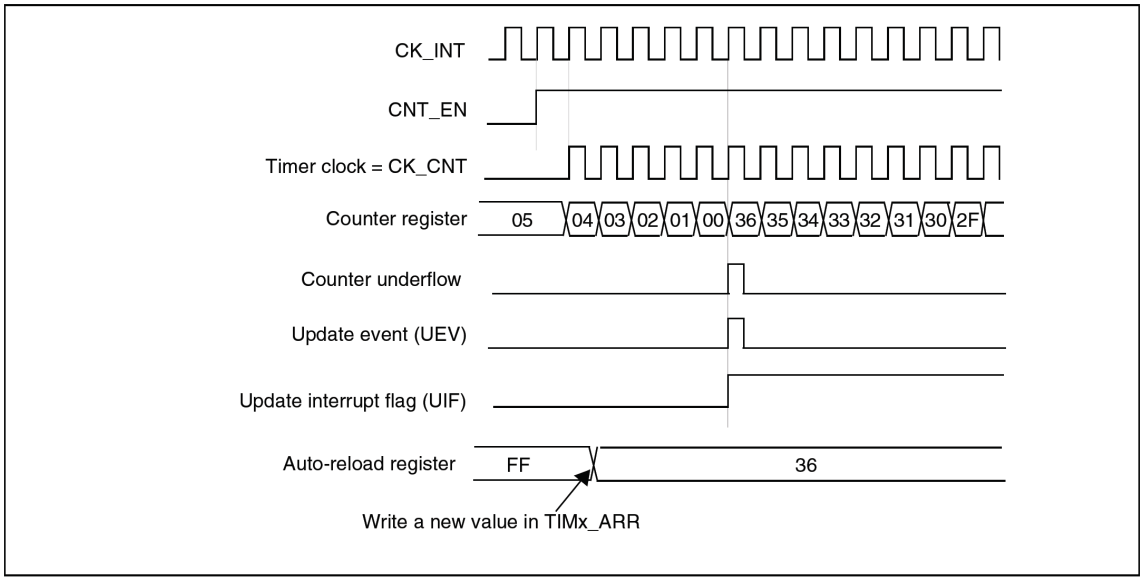


图 103. 计数器时序图，当没有使用重复计数器时的更新事件



中央对齐模式 (向上 / 向下计数)

在中央对齐模式，计数器从 0 开始计数到自动加载的值 (TIMx_ARR 寄存器) 一 1，产生一个计数器溢出事件，然后向下计数到 1 并且产生一个计数器下溢事件；然后再从 0 开始重新计数。当 TIMx_CR1 的 CMS 位不为 "00" 时，使能中央对齐模式。已配置为输出的通道的输出比较中断标志在以下情况下被置位：计数器向下计数（中央对齐模式 1, CMS = "01"），计数器向上计数（中央对齐模式 2, CMS = "10"），计数器向下和向上计数（中央对齐模式 3, CMS = "11"），

在这个模式，不能写入 TIMx_CR1 中的 DIR 方向位。它由硬件更新并指示当前的计数方向。可以在每次计数上溢和每次计数下溢时产生更新事件；也可以通过 (软件或者使用从模式控制器) 设置 TIMx_EGR 寄存器中的 UG 位产生更新事件。然后，计数器重新从 0 开始计数，预分频器也重新从 0 开始计数。

设置 TIMx_CR1 寄存器中的 UDIS 位可以禁止 UEV 事件。这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。因此 UDIS 位被清为’0’之前不会产生更新事件。然而，计数器仍会根据当前自动重加载的值，继续向上或向下计数。

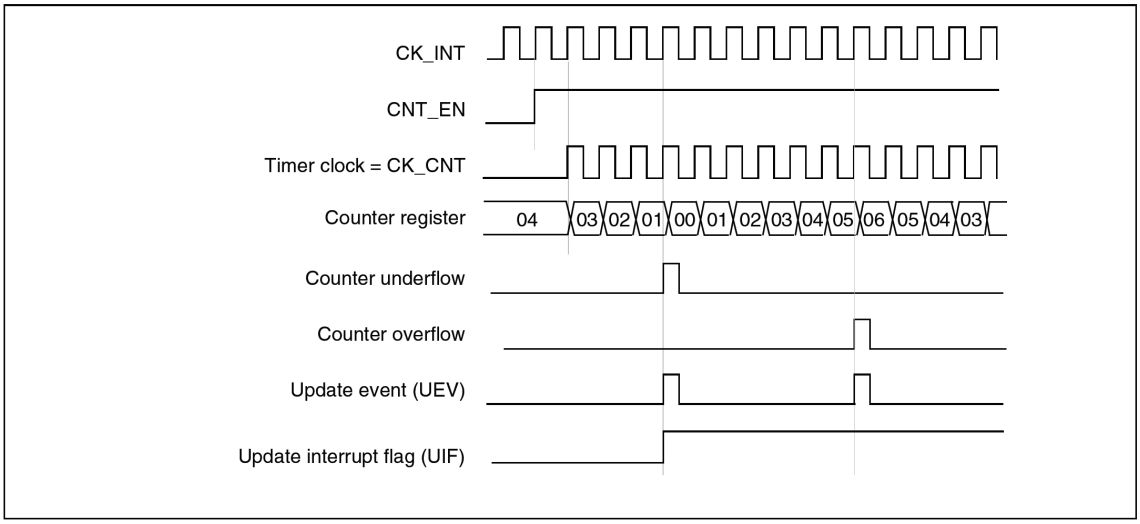
此外，如果设置了 TIMx_CR1 寄存器中的 URS 位 (选择更新请求) ，设置 UG 位将产生一个更新事件 UEV 但不设置 UIF 标志 (因此不产生中断和 DMA 请求)，这是为了避免在发生捕获事件并清除计数器时，同时产生更新和捕获中断。

当发生更新事件时，所有的寄存器都被更新，并且 (根据 URS 位的设置) 更新标志位 (TIMx_SR 寄 存器中的 UIF 位) 也被设置。

- 预分频器的缓存器被加载为预装载 (TIMx_PSC 寄存器) 的值。
- 当前的自动加载寄存器被更新为预装载值 (TIMx_ARR 寄存器中的内容)。注：如果因为计数器溢出而产生更新，自动重装载将在计数器重载入之前被更新，因此下一个周期将是预期的值 (计数器被装载为新的值)。

以下是一些计数器在不同时钟频率下的操作的例子：

图 104. 计数器时序图，内部时钟分频因子为 1，TIMx_ARR=0x6



1. 这里使用了中心对齐模式 1(详见 16.4.1 节)。

图 105. 计数器时序图，内部时钟分频因子为 2

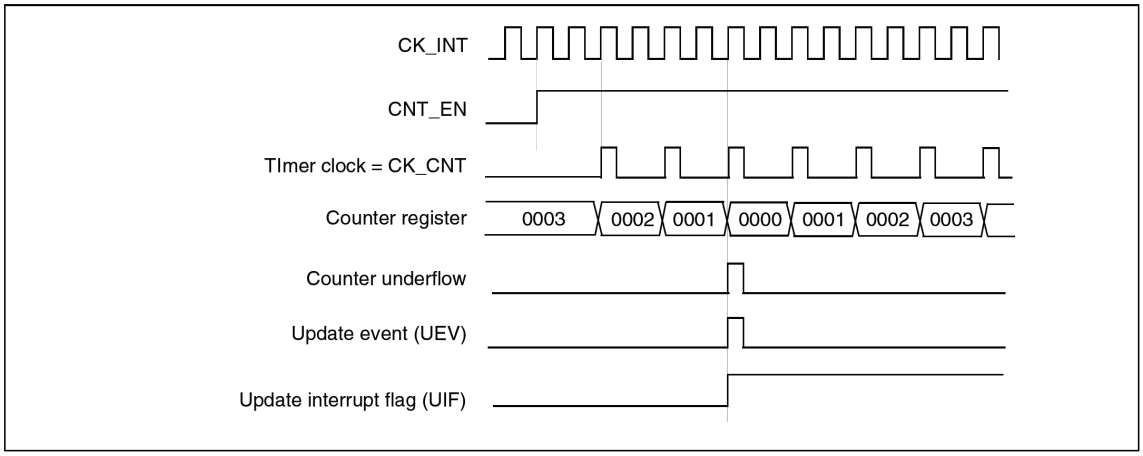
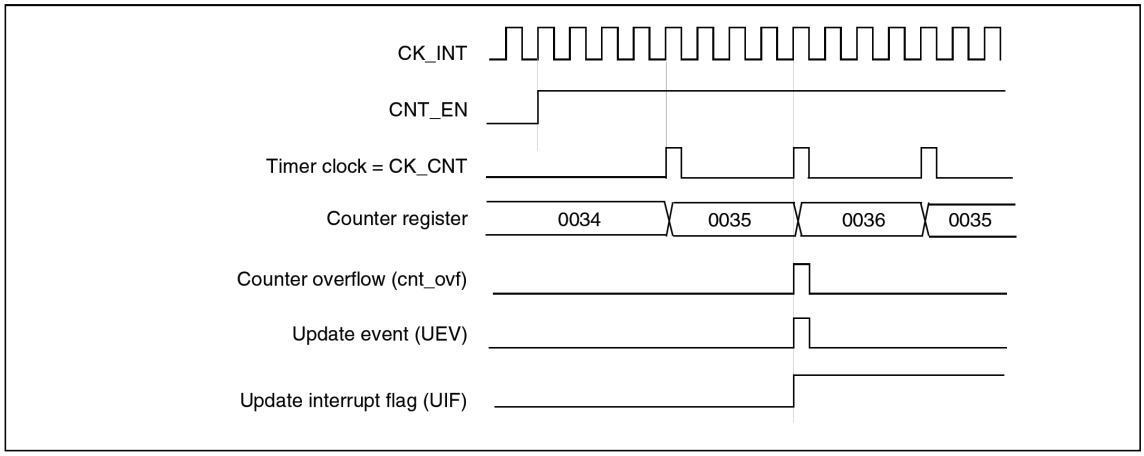


图 106. 计数器时序图，内部时钟分频因子为 4，TIMx_ARR=0x36



1. 这里使用了中央对称模式 2 或 3，在溢出时产生 UIF

图 107. 计数器时序图，内部时钟分频因子为 N

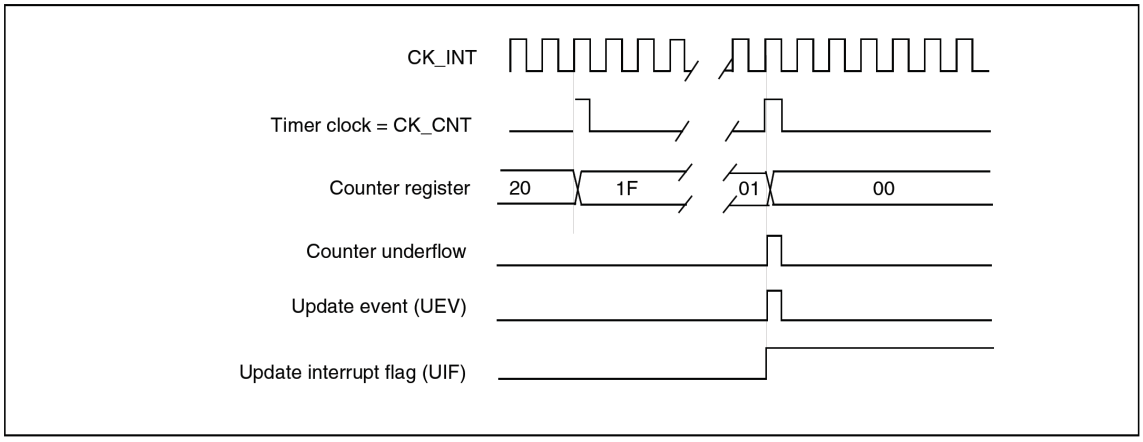


图 108. 计数器时序图，更新事件 ARPE=1（定时器向下溢出）

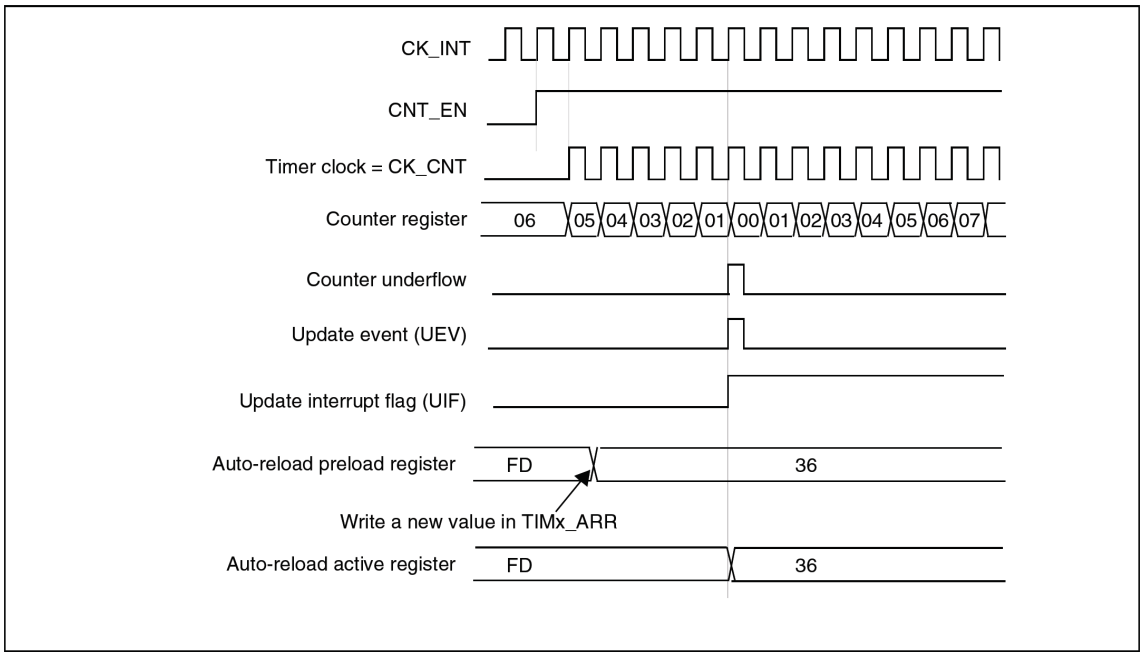
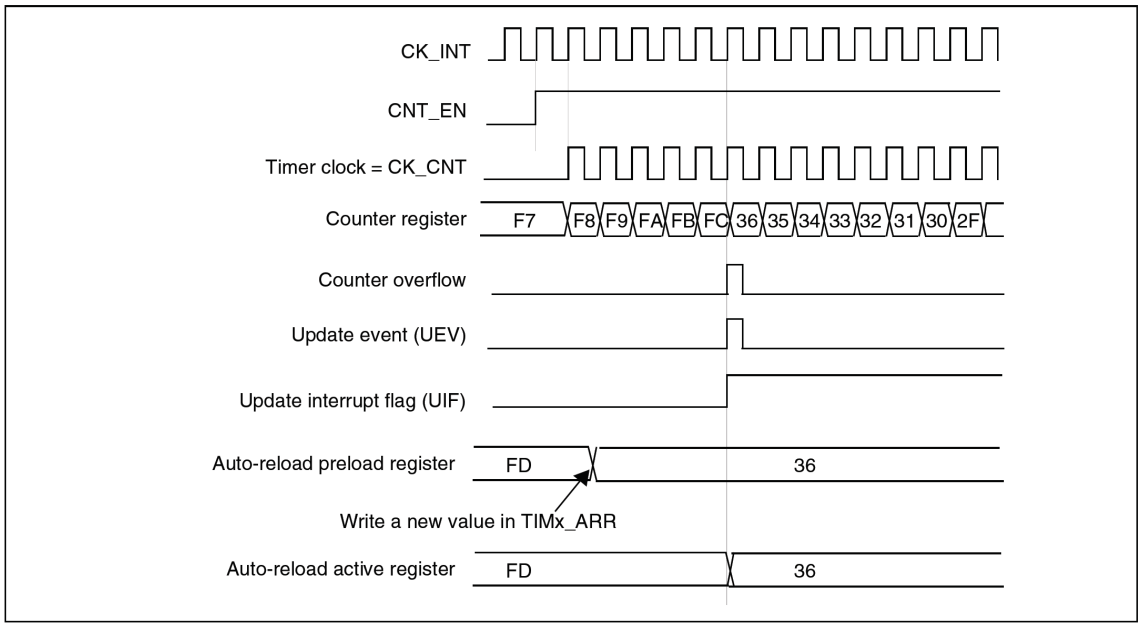


图 109. 计数器时序图，更新事件 ARPE=1（定时器向上溢出）



16.3.3 时钟源

计数器时钟可由下列时钟源提供：

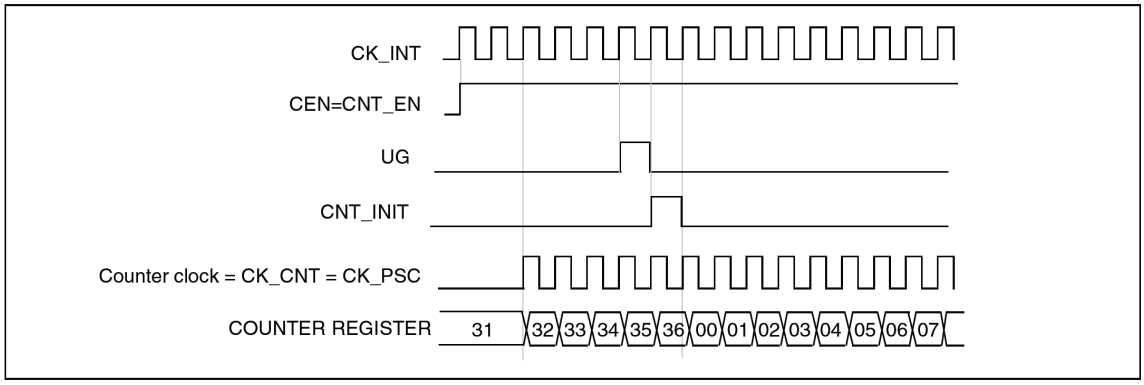
- 内部时钟 (CK_INT)
- 外部时钟模式 1：外部输入脚 (TIx)
- 外部时钟模式 2：外部触发输入 (ETR)
- 内部触发输入 (ITRx)：使用一个定时器作为另一个定时器的预分频器，如可以配置一个定时器 Timer1 而作为另一个定时器 Timer2 的预分频器。参见 319 页使用一个定时器作为另一个定时器的预分频器。

内部时钟源 (CK_INT)

如果禁止了从模式控制器 (TIMx_SMCR 寄存器的 SMS=000)，则 CEN、DIR(TIMx_CR1 寄存器) 和 UG 位 (TIMx_EGR 寄存器) 是事实上的控制位，并且只能被软件修改 (UG 位仍被自动清除)。只要 CEN 位被写成’1’，预分频器的时钟就由内部时钟 CK_INT 提供。

图 110 显示了控制电路和向上计数器在一般模式下，不带预分频器时的操作。

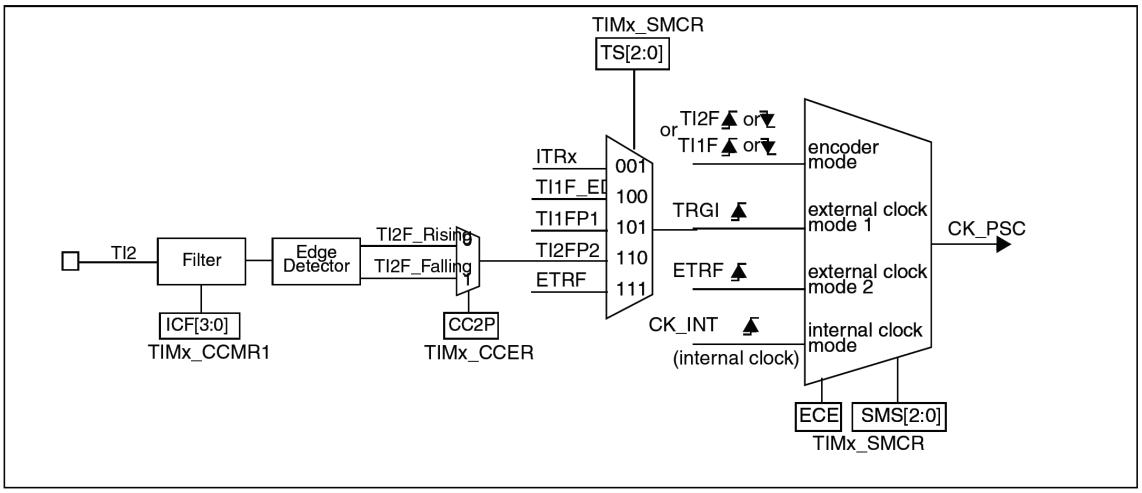
图 110. 一般模式下的控制电路，内部时钟分频因子为 1



外部时钟源模式 1

当 TIMx_SMCR 寄存器的 SMS=111 时，此模式被选中。计数器可以在选定输入端的每个上升沿或下降沿计数。

图 111. TI2 外部时钟连接示例



例如，要配置向上计数器在 T12 输入端的上升沿计数，使用下列步骤：

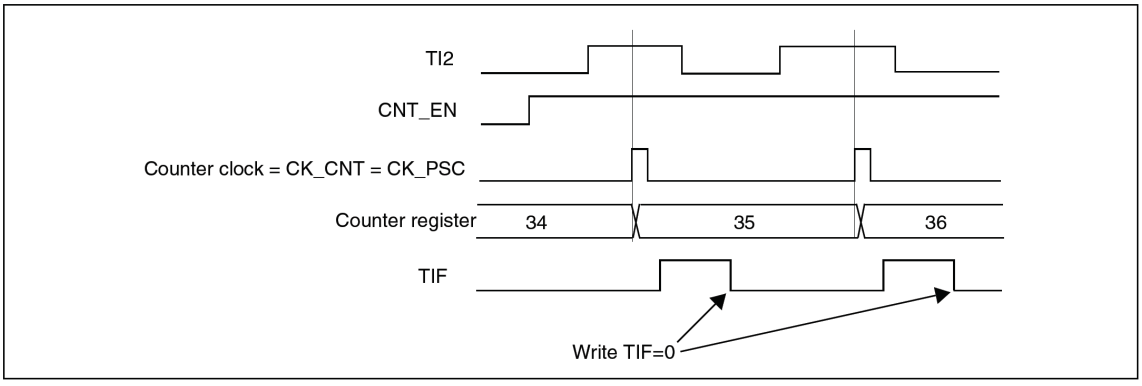
1. 配置 TIMx_CCMR1 寄存器 CC2S=’01’，配置通道 2 检测 TI2 输入的上升沿
2. 配置 TIMx_CCMR1 寄存器的 IC2F[3:0]，选择输入滤波器带宽 (如果不需要滤波器，保持 IC2F=0000)

注： *捕获预分频器不用作触发，所以不需要对它进行配置*

3. 配置 TIMx_CCER 寄存器的 CC2P=’0’，选定上升沿极性
4. 配置 TIMx_SMCR 寄存器的 SMS=’111’，选择定时器外部时钟模式 1
5. 配置 TIMx_SMCR 寄存器中的 TS=’110’，选定 TI2 作为触发输入源
6. 设置 TIMx_CR1 寄存器的 CEN=’1’，启动计数器 当上升沿出现在 TI2，计数器计数一次，且 TIF 标志被设置。

在 TI2 的上升沿和计数器实际时钟之间的延时，取决于在 TI2 输入端的重新同步电路。

图 112. 外部时钟模式 1 下的控制电路



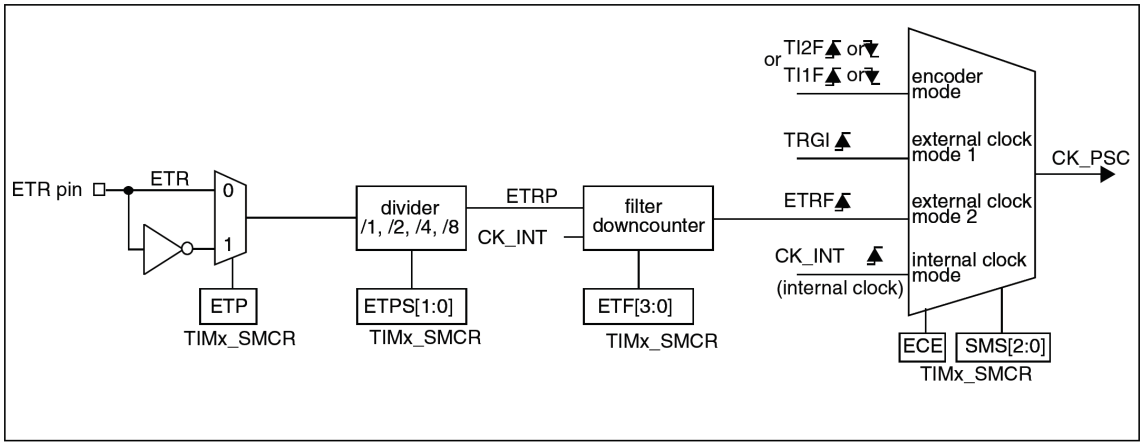
外部时钟源模式 2

选定此模式的方法为：令 TIMx_SMCR 寄存器中的 ECE=1

计数器能够在外部触发 ETR 的每一个上升沿或下降沿计数。

下图是外部触发输入的框图

图 113. 外部触发输入框图



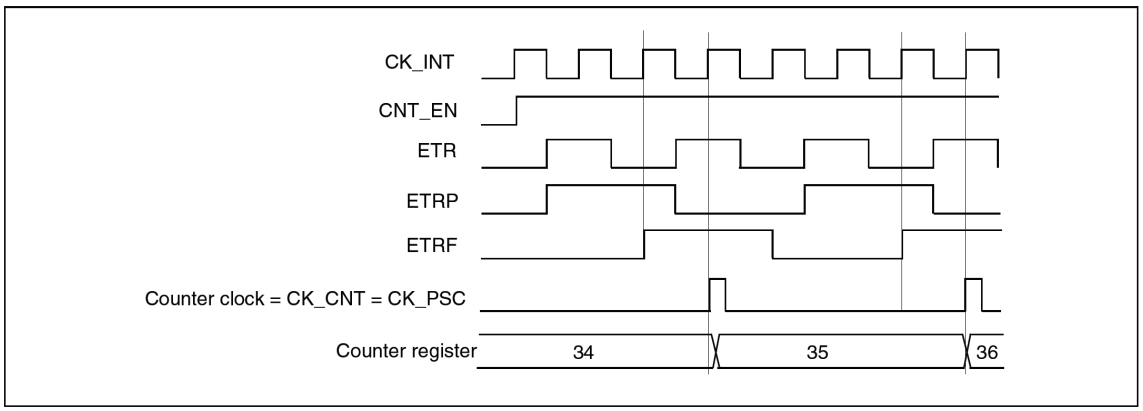
例如，要配置在 ETR 下每 2 个上升沿计数一次的向上计数器，使用下列步骤：

1. 本例中不需要滤波器，置 TIMx_SMCR 寄存器中的 ETF[3:0]=0000
2. 设置预分频器，置 TIMx_SMCR 寄存器中的 ETPS[1:0]=01
3. 设置在 ETR 的上升沿检测，置 TIMx_SMCR 寄存器中的 ETP=0
4. 开启外部时钟模式 2，置 TIMx_SMCR 寄存器中的 ECE=1
5. 启动计数器，置 TIMx_CR1 寄存器中的 CEN=1

计数器在每 2 个 ETR 上升沿计数一次。

在 ETR 的上升沿和计数器实际时钟之间的延时取决于在 ETRP 信号端的重新同步电路。

图 114. 外部时钟模式 2 下的控制电路

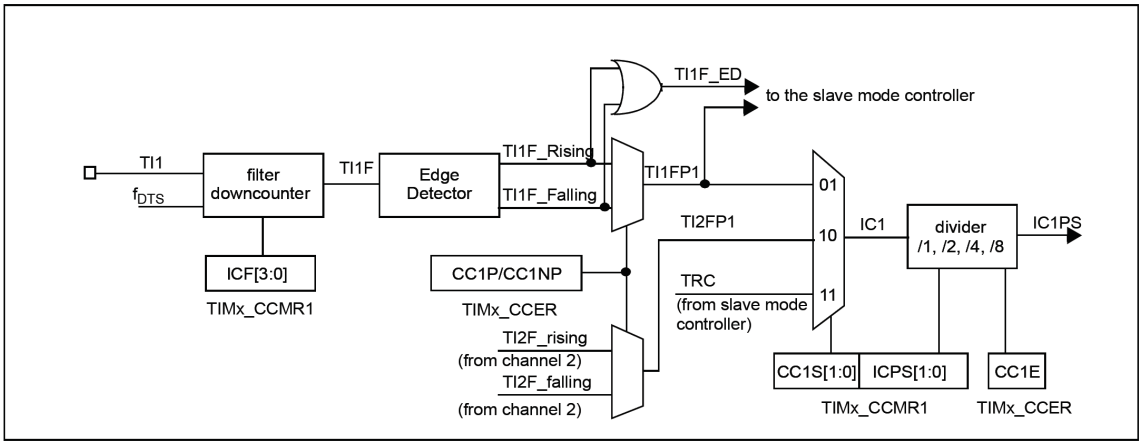


16.3.4 捕获 / 比较通道

每一个捕获 / 比较通道都是围绕着一个捕获 / 比较寄存器 (包含影子寄存器)，包括捕获的输入部分 (数字滤波、多路复用和预分频器)，和输出部分 (比较器和输出控制)。
下面几张图是一个捕获 / 比较通道概览。

输入部分对相应的 TIx 输入信号采样，并产生一个滤波后的信号 TIxF。然后，一个带极性选择的边缘检测器产生一个信号 (TIxFPx)，它可以作为从模式控制器的输入触发或者作为捕获控制。该信号通过预分频进入捕获寄存器 (ICxPS)。

图 115. 捕获 / 比较通道 (如：通道 1 输入部分)



输出部分产生一个中间波形 OCxRef(高有效) 作为基准，链的末端决定最终输出信号的极性。

图 116. 捕获 / 比较通道 1 的主电路

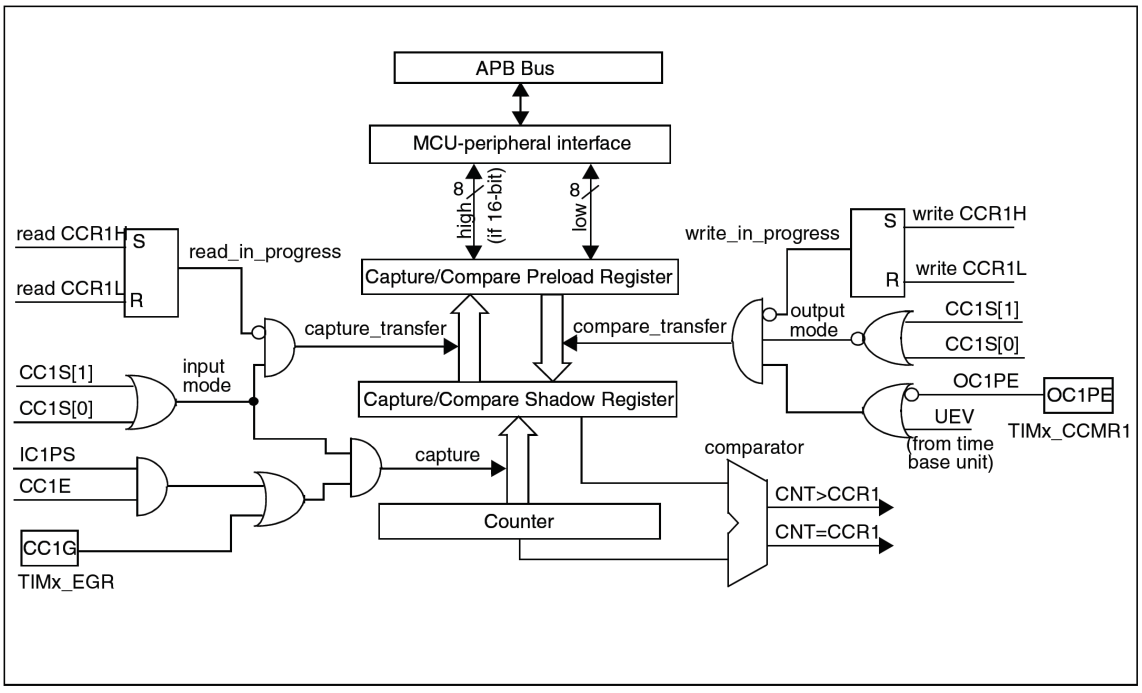
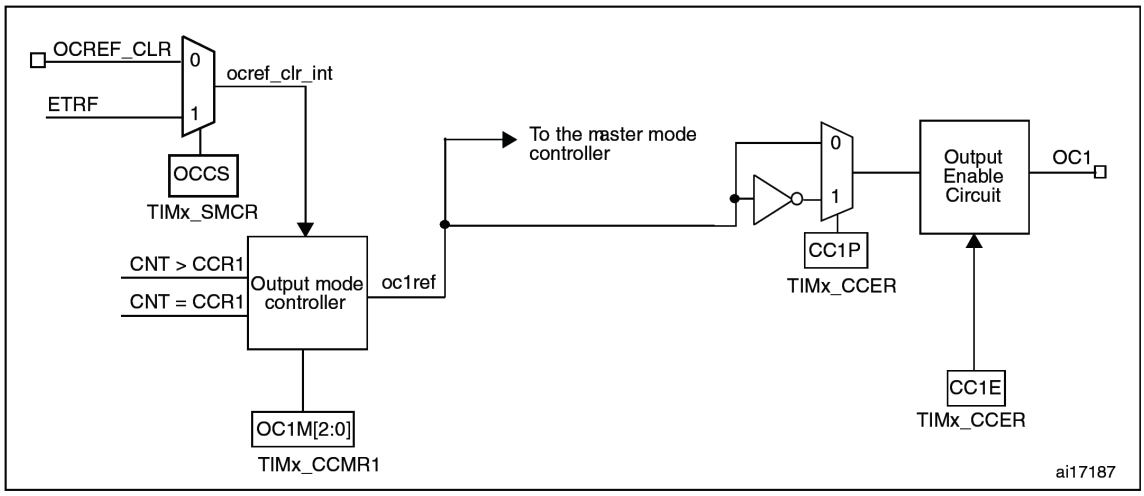


图 117. 捕获 / 比较通道的输出部分 (通道 1)



捕获 / 比较模块由一个预装载寄存器和一个影子寄存器组成。读写过程仅操作预装载寄存器。

在捕获模式下，捕获发生在影子寄存器上，然后再复制到预装载寄存器中。

在比较模式下，预装载寄存器的内容被复制到影子寄存器中，然后影子寄存器的内容和计数器 进行比较。

16.3.5 输入捕获模式

在输入捕获模式下，当检测到 ICx 信号上相应的边沿后，计数器的当前值被锁存到捕获 / 比较寄存 器 (TIMx_CCRx) 中。当捕获事件发生时，相应的 CCxIF 标志 (TIMx_SR 寄存器) 被置’1’，如果使能了中断或者 DMA 操作，则将产生中断或者 DMA 操作。如果捕获事件发生时 CCxIF 标志已经为高，那么重复捕获标志 CCxOF(TIMx_SR 寄存器) 被置’1’ 。写 CCxIF=0 可清除 CCxIF，或读取存储在 TIMx_CCRx 寄存器中的捕获数据也可清除 CCxIF。写 CCxOF=0 可清除 CCxOF。

以下例子说明如何在 TI1 输入的上升沿时捕获计数器的值到 TIMx_CCR1 寄存器中，步骤如下：

- 选择有效输入端：TIMx_CCR1 必须连接到 TI1 输入，所以写入 TIMx_CCR1 寄存器中的 CC1S=01，只要 CC1S 不为’00’，通道被配置为输入，并且 TM1_CCR1 寄存器变为只读。
- 根据输入信号的特点，配置输入滤波器为所需的带宽 (即输入为 TIx 时，输入滤波器控制位是 TIMx_CCMRx 寄存器中的 ICxF 位)。假设输入信号在最多 5 个内部时钟周期的时间内抖动， 我们须配置滤波器的带宽长于 5 个时钟周期。因此我们可以 (以 fDTS 频率) 连续采样 8 次，以确 认在 TI1 上一次真实的边沿变换，即在 TIMx_CCMR1 寄存器中写入 IC1F=0011。

- 选择 TI1 通道的有效转换边沿，在 TIMx_CCER 寄存器中写入 CC1P=0 和 CC1NP=0 (上升沿)。
- 配置输入预分频器。在本例中，我们希望捕获发生在每一个有效的电平转换时刻，因此预分频器被禁止 (写 TIMx_CCMR1 寄存器的 IC1PS=00)。
- 设置 TIMx_CCER 寄存器的 CC1E=1，允许捕获计数器的值到捕获寄存器中。
- 如果需要，通过设置 TIMx_DIER 寄存器中的 CC1IE 位允许相关中断请求，通过设置 TIMx_DIER 寄存器中的 CC1DE 位允许 DMA 请求。

当发生一个输入捕获时：

- 产生有效的电平转换时，计数器的值被传送到 TIMx_CCR1 寄存器。
- CC1IF 标志被设置 (中断标志)。当发生至少 2 个连续的捕获时，而 CC1IF 未曾被清除，CC1OF 也被置’1’。
- 如设置了 CC1IE 位，则会产生一个中断。
- 如设置了 CC1DE 位，则还会产生一个 DMA 请求。

为了处理捕获溢出，建议在读出捕获溢出标志之前读取数据，这是为了避免丢失在读出捕获溢出标志之后和读取数据之前可能产生的捕获溢出信息。

注： *设置 TIMx_EGR 寄存器中相应的 CCxG 位，可以通过软件产生输入捕获中断和 / 或 DMA 请求。*

16.3.6 PWM 输入模式

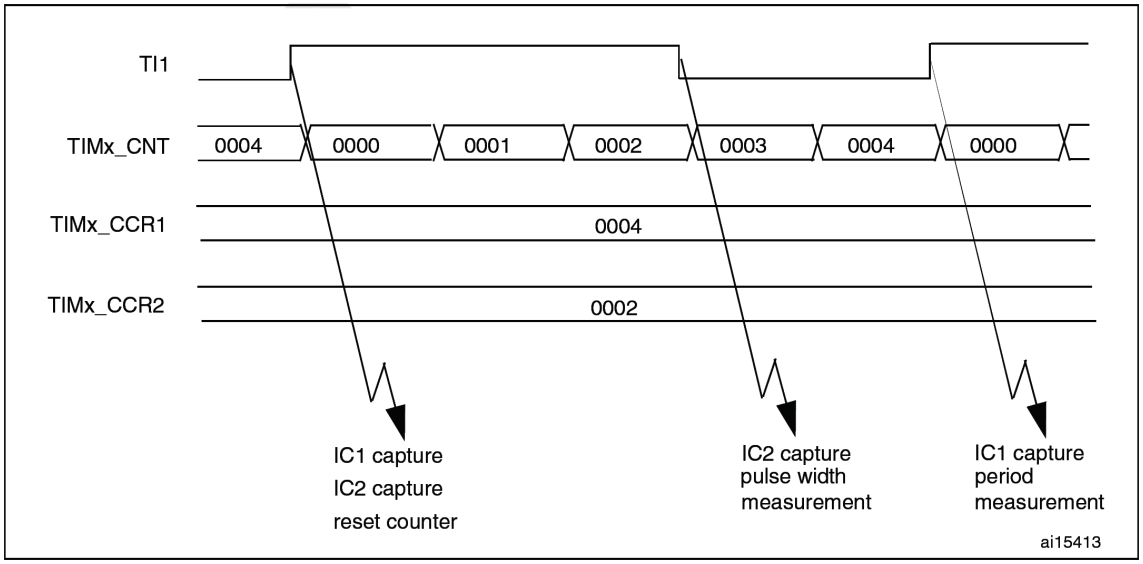
该模式是输入捕获模式的一个特例，除下列区别外，操作与输入捕获模式相同：

- 两个 ICx 信号被映射至同一个 TIx 输入。
- 这 2 个 ICx 信号为边沿有效，但是极性相反。
- 其中一个 TIxFP 信号被作为触发输入信号，而从模式控制器被配置成复位模式。

例如，你需要测量输入到 TI1 上的 PWM 信号的长度 (TIMx_CCR1 寄存器) 和占空比 (TIMx_CCR2 寄存器)，具体步骤如下 (取决于 CK_INT 的频率和预分频器的值)

- 选择 TIMx_CCR1 的有效输入：置 TIMx_CCMR1 寄存器的 CC1S=01(选择 TI1)。
- 选择 TI1FP1 的有效极性 (用来捕获数据到 TIMx_CCR1 中和清除计数器)：置 CC1P=0 和 CC1NP=0 (上升沿有效)。
- 选择 TIMx_CCR2 的有效输入：置 TIMx_CCMR1 寄存器的 CC2S=10(选择 TI1)。
- 选择 TI1FP2 的有效极性 (捕获数据到 TIMx_CCR2)：置 CC2P=1 和 CC1NP=0 (下降沿有效)。
- 选择有效的触发输入信号：置 TIMx_SMCR 寄存器中的 TS=101(选择 TI1FP1)。
- 配置从模式控制器为复位模式：置 TIMx_SMCR 中的 SMS=100。
- 使能捕获：置 TIMx_CCER 寄存器中 CC1E=1 且 CC2E=1。

图 118. PWM 输入模式时序



16.3.7 强置输出模式

在输出模式 (TIMx_CCMRx 寄存器中 CCxS=00) 下，输出比较信号 (OCxREF 和相应的 OCx) 能够直接由软件强置为有效或无效状态，而不依赖于输出比较寄存器和计数器间的比较结果。置 TIMx_CCMRx 寄存器中相应的 OCxM=101，即可强置输出比较信号 (OCxREF/OCx) 为有效状态。这样 OCxREF 被强置为高电平 (OCxREF 始终为高电平有效)，同时 OCx 得到 CCxP 极性位相反的值。

例如：CCxP=0(OCx 高电平有效)，则 OCx 被强置为高电平。
置 TIMx_CCMRx 寄存器中的 OCxM=100，可强置 OCxREF 信号为低。
该模式下，在 TIMx_CCRx 影子寄存器和计数器之间的比较仍然在进行，相应的标志也会被修改。因此仍然会产生相应的中断和 DMA 请求。这将会在下面的输出比较模式一节中介绍。

16.3.8 输出比较模式

此功能是用来控制一个输出波形，或者指示一段给定的时间已经到时。
当计数器与捕获 / 比较寄存器的内容相同时，输出比较功能做如下操作：

- 将输出比较模式 (TIMx_CCMRx 寄存器中的 OCxM 位) 和输出极性 (TIMx_CCER 寄存器中的 CCxP 位) 定义的值输出到对应的引脚上。在比较匹配时，输出引脚可以保持它的电平 (OCxM=000)、被设置成有效电平 (OCxM=001)、被设置成无效电平 (OCxM=010) 或进行翻 转 (OCxM=011)。
- 设置中断状态寄存器中的标志位 (TIMx_SR 寄存器中的 CCxIF 位)。
- 若设置了相应的中断屏蔽 (TIMx_DIER 寄存器中的 CCxIE 位)，则产生一个中断。
- 若设置了相应的使能位 (TIMx_DIER 寄存器中的 CCxDE 位，TIMx_CR2 寄存器中的 CCDS 位选择 DMA 请求功能)，则产生一个 DMA 请求。

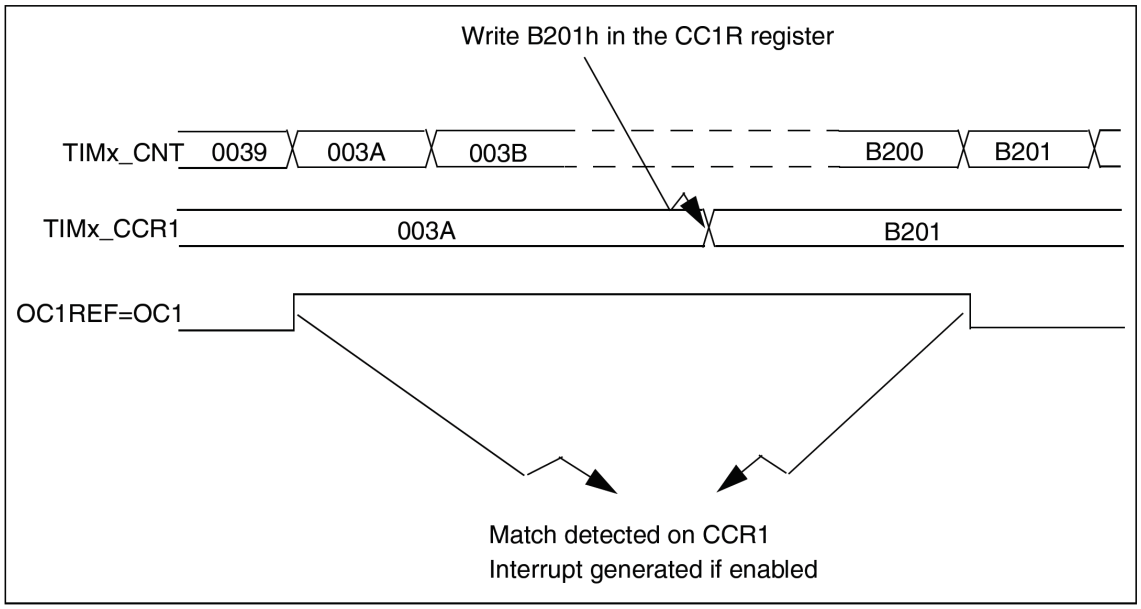
TIMx_CCMRx 中的 OCxPE 位选择 TIMx_CCRx 寄存器是否需要使用预装载寄存器。
在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出没有影响。 同步的精度可以达到计数器的一个计数周期。输出比较模式 (在单脉冲模式下) 也能用来输出一个单脉冲。

输出比较模式的配置步骤：

1. 选择计数器时钟 (内部，外部，预分频器)
2. 将相应的数据写入 TIMx_ARR 和 TIMx_CCRx 寄存器中
3. 如果要产生一个中断请求和 / 或一个 DMA 请求，设置 CCxIE 位和 / 或 CCxDE 位。
4. 选择输出模式，例如当计数器 CNT 与 CCRx 匹配时翻转 OCx 的输出引脚，CCRx 预装载未用，开启 OCx 输出且高电平有效，则必须设置 OCxM=’011’、OCxPE=’0’、CCxP=’0’和 CCxE=’1’。
5. 设置 TIMx_CR1 寄存器的 CEN 位启动计数器

TIMx_CCRx 寄存器能够在任何时候通过软件进行更新以控制输出波形，条件是未使用预装载寄存器 (OCxPE=’0’，否则 TIMx_CCRx 影子寄存器只能在发生下一次更新事件时被更新)。下图给出了一个例子。

图 119. 输出比较模式，翻转 OC1



16.3.9 PWM 模式

脉冲宽度调制模式可以产生一个由 TIMx_ARR 寄存器确定频率、由 TIMx_CCRx 寄存器确定占空比的信号。

在 TIMx_CCMRx 寄存器中的 OCxM 位写入’110’(PWM 模式 1) 或’111’(PWM 模式 2)，能够独立地设置每个 OCx 输出通道产生一路 PWM。必须设置 TIMx_CCMRx 寄存器 OCxPE 位以使能相应的预装载寄存器，最后还要设置 TIMx_CR1 寄存器的 ARPE 位，(在向上计数或中心对称模式中) 使能自动重装载的预装载寄存器。

仅当发生一个更新事件的时候，预装载寄存器才能被传送到影子寄存器，因此在计数器开始计数之前，必须通过设置 TIMx_EGR 寄存器中的 UG 位来初始化所有的寄存器。

OCx 的极性可以通过软件在 TIMx_CCER 寄存器中的 CCxP 位设置，它可以设置为高电平有效或低电平有效。TIMx_CCER 寄存器中的 CCxE 位控制 OCx 输出使能。详见 TIMx_CCERx 寄存器的描述。

在 PWM 模式 (模式 1 或模式 2) 下，TIMx_CNT 和 TIMx_CCRx 始终在进行比较，(依据计数器的计数方向) 以确 定是否符合 TIMx_CCRx ≤ TIMx_CNT 或者 TIMx_CNT ≤ TIMx_CCRx 。然而 为了 与 OCREF_CLR 的功能 (在下一个 PWM 周期之前，ETR 信号上的一个外部事件能够清除 OCxREF) 一致，OCxREF 信号只能在下述条件下产生：

- 当比较的结果改变，或
- 当输出比较模式 (TIMx_CCMRx 寄存器中的 OCxM 位) 从“冻结”(无比较，OCxM=’000’) 切换到某个 PWM 模式 (OCxM=’110’ 或’111’)。

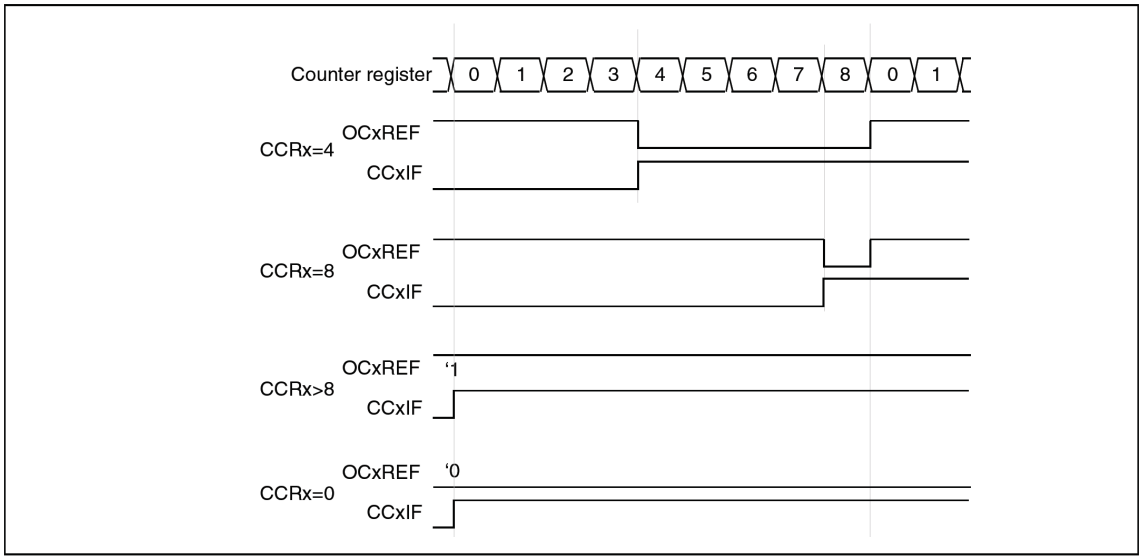
这样在运行中可以通过软件强置 PWM 输出。
根据 TIMx_CR1 寄存器中 CMS 位的状态，定时器能够产生边沿对齐的 PWM 信号或中央对齐的 PWM 信号。

PWM 边沿对齐模式

向上计数配置
当 TIMx_CR1 寄存器中的 DIR 位为低的时候执行向上计数。参看 291 页：向上计数模式

下面是一个 PWM 模式 1 的例子。当 TIMx_CNT<TIMx_CCRx 时 PWM 信号参考 OCxREF 为高，否则为低。如果 TIMx_CCRx 中的比较值大于自动重装载值 (TIMx_ARR)，则 OCxREF 保持为’1’。 如果比较值为 0，则 OCxREF 保持为’0’。 图 120 为 TIMx_ARR=8 时边沿对齐的 PWM 波形实例。

图 120. 边沿对齐的 PWM 波形 (ARR=8)



向下计数的配置

当 TIMx_CR1 寄存器的 DIR 位为高时执行向下计数。参看 294 页：向下计数模式
在 PWM 模式 1，当 TIMx_CNT>TIMx_CCRx 时参考信号 OCxREF 为低，否则为高。如果 TIMx_CCRx 中的比较值大于 TIMx_ARR 中的自动重装载值，则 OCxREF 保持为’1’。该模式下不能产生 0%的 PWM 波形。

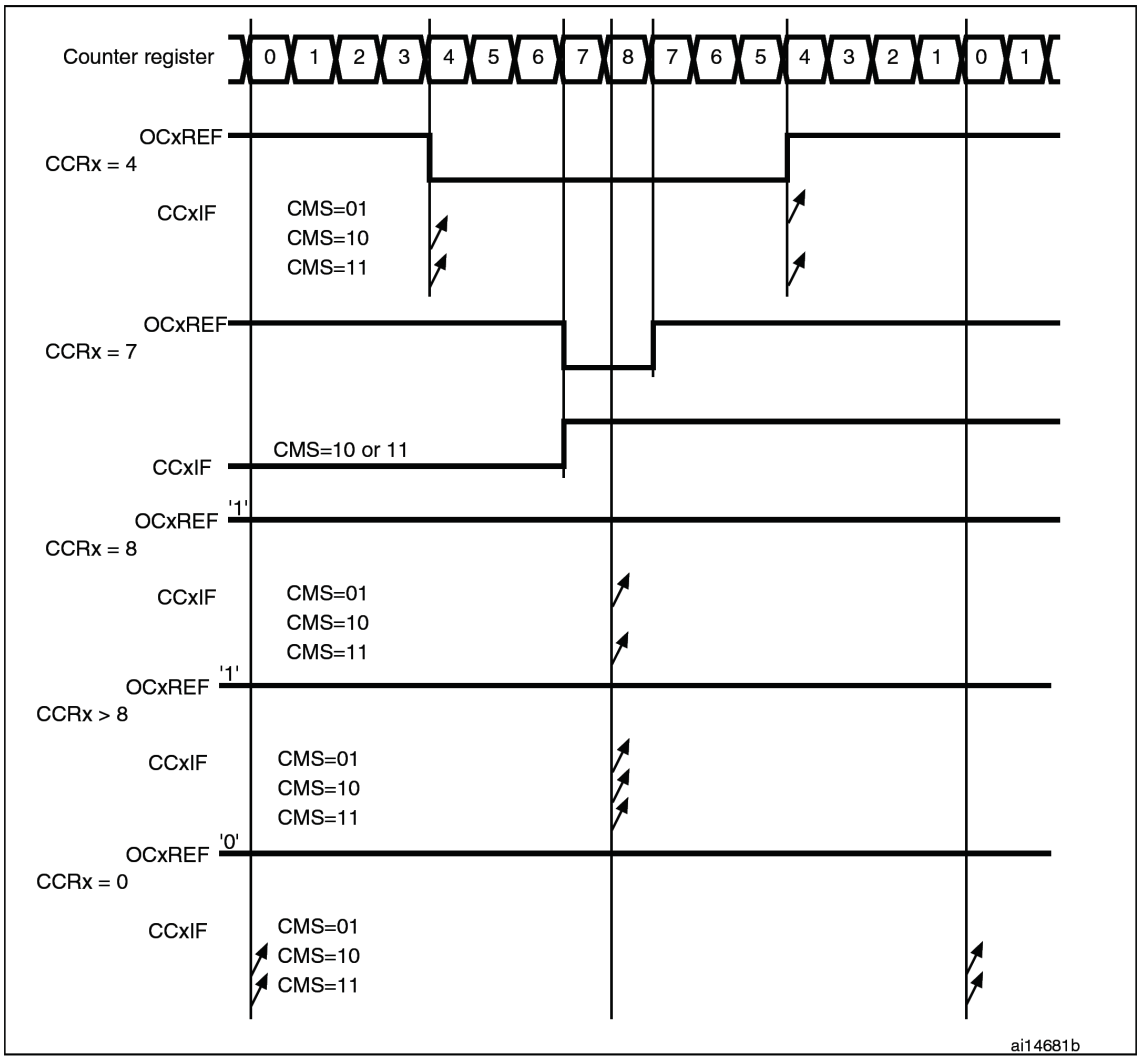
PWM 中央对齐模式

当 TIMx_CR1 寄存器中的 CMS 位不为’00’时，为中央对齐模式 (所有其他的配置对 OCxREF/OCx 信号都有相同的作用)。根据不同的 CMS 位设置，比较标志可以在计数器向上计数时被置’1’、在计数器向下计数时被置’1’、或在计数器向上和向下计数时被置’1’。TIMx_CR1 寄存器中的计数方向位 (DIR) 由硬件更新，不要用软件修改它。参看 296 页的中央对齐模式。

图 121 给出了一些中央对齐的 PWM 波形的例子

- TIMx_ARR=8
- PWM 模式 1
- TIMx_CR1 寄存器中的 CMS=01，在中央对齐模式 1 时，当计数器向下计数时设置比较标志。

图 121. 中央对齐的 PWM 波形 (ARR=8)



使用中央对齐模式的提示：

- 进入中央对齐模式时，使用当前的向上 / 向下计数配置；这就意味着计数器向上还是向下计数取决于 TIMx_CR1 寄存器中 DIR 位的当前值。此外，软件不能同时修改 DIR 和 CMS 位。
- 不推荐当运行在中央对齐模式时改写计数器，因为这会产生不可预知的结果。特别地：
  - 如果写入计数器的值大于自动重加载的值 (TIMx_CNT>TIMx_ARR)，则方向不会被更新。 例如，如果计数器正在向上计数，它就会继续向上计数。
  - 如果将 0 或者 TIMx_ARR 的值写入计数器，方向被更新，但不产生更新事件 UEV。
- 使用中央对齐模式最保险的方法，就是在启动计数器之前产生一个软件更新 (设置 TIMx_EGR 位中的 UG 位)，不要在计数进行过程中修改计数器的值。

16.3.10 单脉冲模式

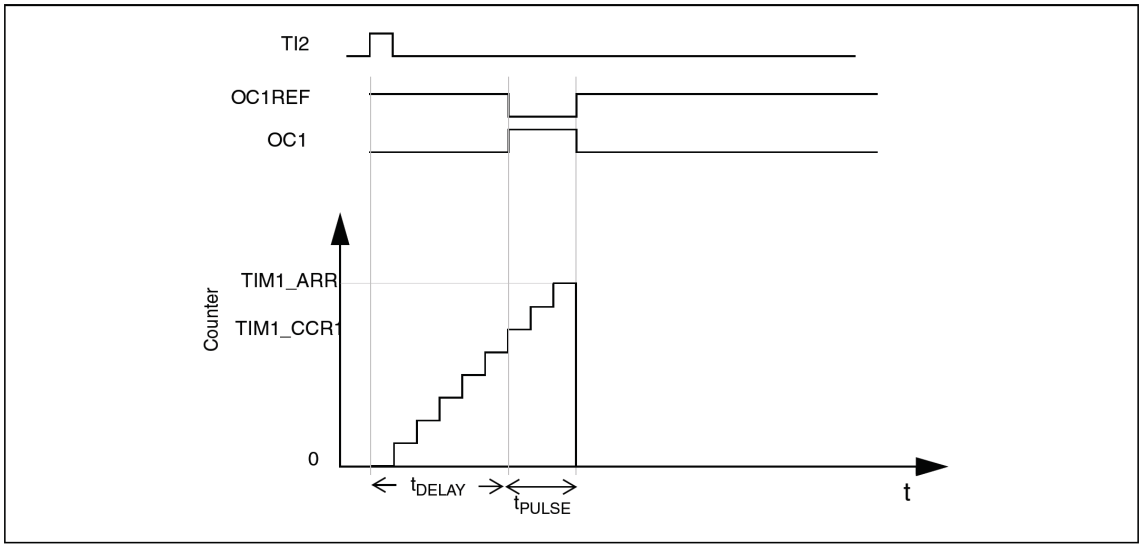
单脉冲模式 (OPM) 是前述众多模式的一个特例。这种模式允许计数器响应一个激励，并在一个程序可控的延时之后，产生一个脉宽可程序控制的脉冲。

可以通过从模式控制器启动计数器，在输出比较模式或者 PWM 模式下产生波形。设置 TIMx_CR1 寄存器中的 OPM 位将选择单脉冲模式，这样可以让计数器自动地在产生下一个更新事件 UEV 时停止。

仅当比较值与计数器的初始值不同时，才能产生一个脉冲。启动之前 (当定时器正在等待触发)，必须如下配置：

- 向上计数方式：CNT < CCRx ≤ ARR (特别地，0 < CCRx)，
- 向下计数方式：CNT > CCRx。

图 122. 单脉冲模式的例子



例如，你需要在从 TI2 输入脚上检测到一个上升沿开始，延迟 tDELAY 之后，在 OC1 上产生一个长度为 tPULSE 的正脉冲。

假定 TI2FP2 作为触发 1:

- 置 TIMx_CCMR1 寄存器中的 CC2S=’01’，把 TI2FP2 映像到 TI2。
- 置 TIMx_CCER 寄存器中的 CC2P=’0’ 和 CC2NP=’0’，使 TI2FP2 能够检测上升沿。
- 置 TIMx_SMCR 寄存器中的 TS=’110’，TI2FP2 作为从模式控制器的触发 (TRGI)。
- 置 TIMx_SMCR 寄存器中的 SMS=’110’(触发模式)，TI2FP2 被用来启动计数器。

OPM 波形由写入比较寄存器的数值决定 (要考虑时钟频率和计数器预分频器)

- t_{DELAY} 由写入 TIMx_CCR1 寄存器中的值定义。
- t_{PULSE} 由自动装载值和比较值之间的差值定义 (TIMx_ARR - TIMx_CCR1)。
- 假定当发生比较匹配时要产生从’0’到’1’的波形，当计数器到达预装载值时要产生一个从’1’ 到’0’的波形；首先要置 TIMx_CCMR1 寄存器的 OC1M=’111’，进入 PWM 模式 2；根据需要有选择地使能预装载寄存器：置 TIMx_CCMR1 中的 OC1PE=’1’ 和 TIMx_CR1 寄存器中的 ARPE；然后在 TIMx_CCR1 寄存器中填写比较值，在 TIMx_ARR 寄存器中填写自动装载值，修改 UG 位来产生一个更新事件，然后等待在 TI2 上的一个外部触发事件。本例中，CC1P=’0’。

在这个例子中，TIMx_CR1 寄存器中的 DIR 和 CMS 位应该置低。因为只需一个脉冲，所以必须设置 TIMx_CR1 寄存器中的 OPM=’1’，在下一个更新事件 (当计数器从自动装载值翻转到 0) 时停止计数。设置 TIMx_CR1 寄存器中的 OPM=’0’，就会选择重复模式。

特殊情况：OCx 快速使能：

在单脉冲模式下，在 TIx 输入脚的边沿检测逻辑设置 CEN 位以启动计数器。然后计数器和比较值间的比较操作产生了输出的转换。但是这些操作需要一定的时钟周期，因此它限制了可得到的最小延时 tDELAY。

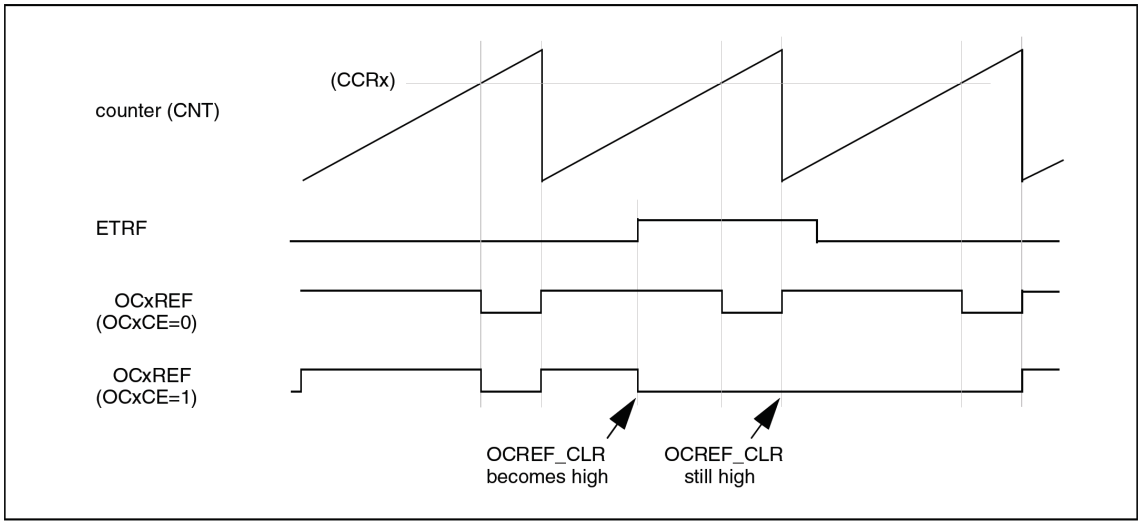
如果要以最小延时输出波形，可以设置 TIMx_CCMRx 寄存器中的 OCxFE 位；此时 OCxREF(和 OCx) 被强制响应激励而不再依赖比较的结果，输出的波形与比较匹配时的波形一样。OCxFE 只在通道配置为 PWM1 和 PWM2 模式时起作用。

16.3.11 外部事件时清除 OCxREF 信号

1. 外部触发预分频器必须处于关闭：TIMx_SMCR 寄存器中的 ETPS[1:0]=’00’。
2. 必须禁止外部时钟模式 2：TIMx_SMCR 寄存器中的 ECE=’0’。
3. 外部触发极性 (ETP) 和外部触发滤波器 (ETF) 可以根据需要配置。

图 123 显示了当 ETRF 输入变为高时，对应不同 OCxCE 的值，OCxREF 信号的动作。在这个例子 中，定时器 TIMx 被置于 PWM 模式。

图 123. 清除 TIMx 的 OCxREF



1. In case of a PWM with a 100% duty cycle (if CCRx>ARR), OCxREF is enabled again at the next counter overflow.

16.3.12 编码器接口模式

选择编码器接口模式的方法是：如果计数器只在 TI2 的边沿计数，则置 TIMx_SMCR 寄存器中的 SMS=001；如果只在 TI1 边沿计数，则置 SMS=010；如果计数器同时在 TI1 和 TI2 边沿计数，则置 SMS=011。

通过设置 TIMx_CCER 寄存器中的 CC1P 和 CC2P 位，可以选择 TI1 和 TI2 极性，CC1NP 和 CC2NP 位必须保持为‘0’；如果需要，还可以对输入滤波器编程。
两个输入 TI1 和 TI2 被用来作为增量编码器的接口。

参看表 46，假定计数器已经启动 (TIMx_CR1 寄存器中的 CEN=’1’)，计数器由每次在 TI1FP1 或 TI2FP2 上的有效跳变驱动。TI1FP1 和 TI2FP2 是 TI1 和 TI2 在通过输入滤波器和极性控制后的信号；如果没有滤波和变相，则 TI1FP1=TI1， TI2FP2=TI2。根据两个输入信号的跳变顺序，产生了计数脉冲和方向信号。依据两个输入信号的跳变顺序，计数器向上或向下计数，同时硬件对 TIMx_CR1 寄存器的 DIR 位进行相应的设置。不管计数器是依靠 TI1 计数、依靠 TI2 计数或者同时依靠 TI1 和 TI2 计数。在任一输入端 (TI1 或者 TI2) 的跳变都会重新计算 DIR 位。

编码器接口模式基本上相当于使用了一个带有方向选择的外部时钟。这意味着计数器只在 0 到 TIMx_ARR 寄存器的自动装载值之间连续计数 (根据方向，或是 0 到 ARR 计数，或是 ARR 到 0 计数)。所以在开始计数之前必须配置 TIMx_ARR；同样，捕获器、比较器、预分频器、触发输出特性等仍工作如常。

在这个模式下，计数器依照增量编码器的速度和方向被自动的修改，因此计数器的内容始终指示着编码器的位置。计数方向与相连的传感器旋转的方向对应。下表列出了所有可能的组合，假设 TI1 和 TI2 不同时变换。

表 46. 计数方向与编码器信号的关系

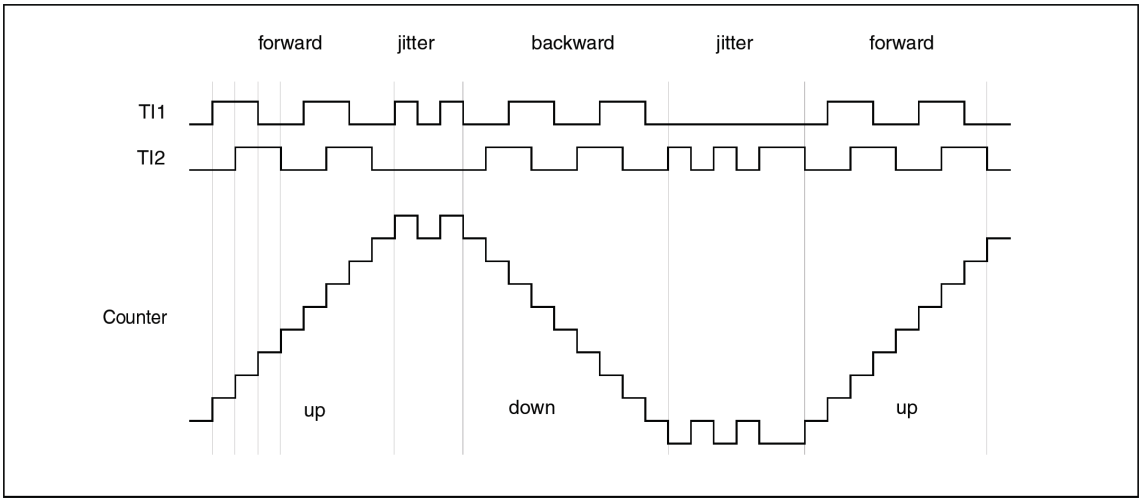
Active edge	Level on opposite signal (TI1FP1 for TI2, TI2FP2 for TI1)	TI1FP1 signal		TI2FP2 signal	
		Rising	Falling	Rising	Falling
Counting on TI1 only	High	Down	Up	No Count	No Count
	Low	Up	Down	No Count	No Count
Counting on TI2 only	High	No Count	No Count	Up	Down
	Low	No Count	No Count	Down	Up
Counting on TI1 and TI2	High	Down	Up	Up	Down
	Low	Up	Down	Down	Up

一个外部的增量编码器可以直接与 MCU 连接而不需要外部接口逻辑。但是，一般会使用比较器将编码器的差动输出转换到数字信号，这大大增加了抗噪声干扰能力。编码器输出的第三个信号表示机械零点，可以把它连接到一个外部中断输入并触发一个计数器复位。

图 124 是一个计数器操作的实例，显示了计数信号的产生和方向控制。它还显示了当选择了双边沿时，输入抖动是如何被抑制的；抖动可能会在传感器的位置靠近一个转换点时产生。在这个例子中，我们假定配置如下：

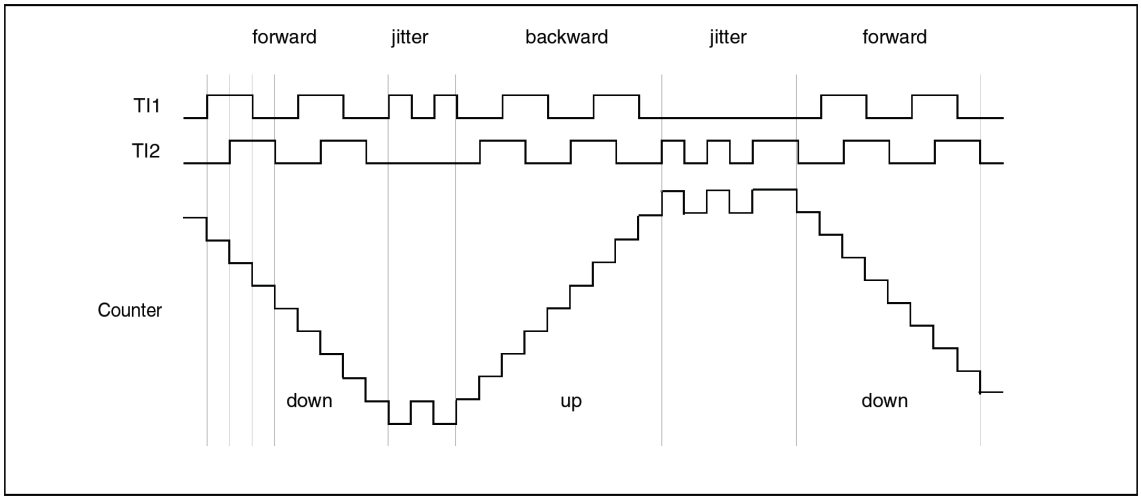
- CC1S=’01’ (TIMx_CCMR1 寄存器，TI1FP1 映射到 TI1)
- CC2S=’01’ (TIMx_CCMR2 寄存器，TI2FP2 映射到 TI2)
- CC1P=’0’ (TIMx_CCER 寄存器，TI1FP1 不反相，TI1FP1=TI1)
- CC2P=’0’ (TIMx_CCER 寄存器，TI2FP2 不反相，TI2FP2=TI2)
- SMS=’011’ (TIMx_SMCR 寄存器，所有的输入均在上升沿和下降沿有效).
- CEN=’1’ (TIMx_CR1 寄存器，计数器使能)

图 124. 编码器模式下的计数器操作实例



下图为当 TI1FP1 极性反相时计数器的操作实例 (CC1P=’1’，其他配置与上例相同)

图 125. TI1FP1 反相的编码器接口模式实例



当定时器配置成编码器接口模式时，提供传感器当前位置的信息。使用第二个配置在捕获模式的定时器，可以测量两个编码器事件的间隔，获得动态的信息 (速度，加速度，减速度)。指示机械零点的编码器输出可被用做此目的。根据两个事件间的间隔，可以按照固定的时间读出计数器。如果可能的话，你可以把计数器的值锁存到第三个输入捕获寄存器 (捕获信号必须是周期的并且可以由另一个定时器产生)；也可以通过一个由实时时钟产生的 DMA 请求来读取它的值。

16.3.13 定时器输入异或功能

TIMx_CR2 寄存器中的 TI1S 位，允许通道 1 的输入滤波器连接到一个异或门的输出端，异或门的 3 个输入端为 TIMx_CH1、TIMx_CH2 和 TIMx_CH3。

异或输出能够被用于所有定时器的输入功能，如触发或输入捕获。

上一章 15.3.18 节给出了此特性用于连接霍尔传感器的例子。

16.3.14 定时器和外部触发的同步

TIMx 定时器能够在多种模式下和一个外部的触发同步：复位模式、门控模式和触发模式。

从模式：复位模式

在发生一个触发输入事件时，计数器和它的预分频器能够重新被初始化；同时，如果 TIMx_CR1 寄存器的 URS 位为低，还会产生一个更新事件 UEV；然后所有的预装载寄存器 (TIMx_ARR，TIMx_CCRx) 都会被更新。

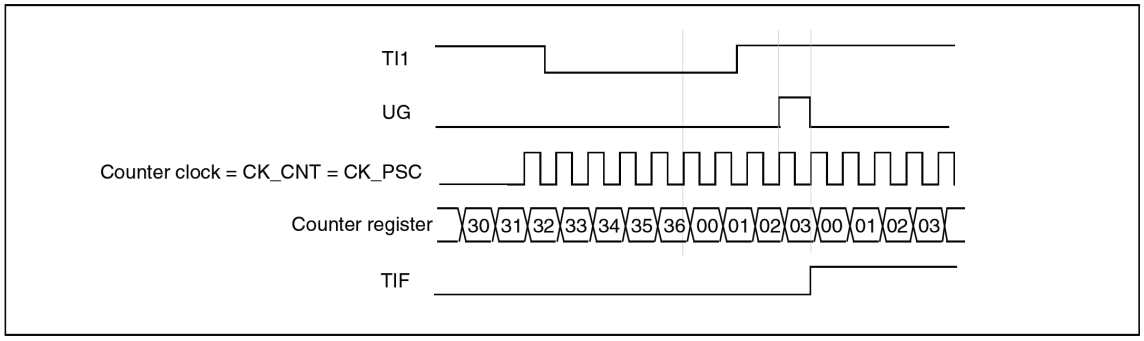
在下面的例子中，TI1 输入端的上升沿导致向上计数器被清零：

- 配置通道 1 以检测 TI1 的上升沿。配置输入滤波器的带宽 (在本例中，不需要任何滤波器，因此保持 IC1F=0000)。触发操作中不使用捕获预分频器，所以不需要配置它。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中 CC1S=01。置 TIMx_CCER 寄存器中 CC1P=0 和 CC1NP=0 以确定极性 (只检测上升沿)。
- 置 TIMx_SMCR 寄存器中 SMS=100，配置定时器为复位模式；置 TIMx_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。
- 置 TIMx_CR1 寄存器中 CEN=1，启动计数器。

计数器开始依据内部时钟计数，然后正常运转直到 TI1 出现一个上升沿；此时，计数器被清零然后从 0 重新开始计数。同时，触发标志 (TIMx_SR 寄存器中的 TIF 位) 被设置，根据 TIMx_DIER 寄存器中 TIE(中断使能) 位和 TDE(DMA 使能) 位的设置，产生一个中断请求或一个 DMA 请求。

下图显示当自动重装载寄存器 TIMx_ARR=0x36 时的动作。在 TI1 上升沿和计数器的实际复位之间的延时，取决于 TI1 输入端的重同步电路。

图 126. 复位模式下的控制电路



从模式：门控模式

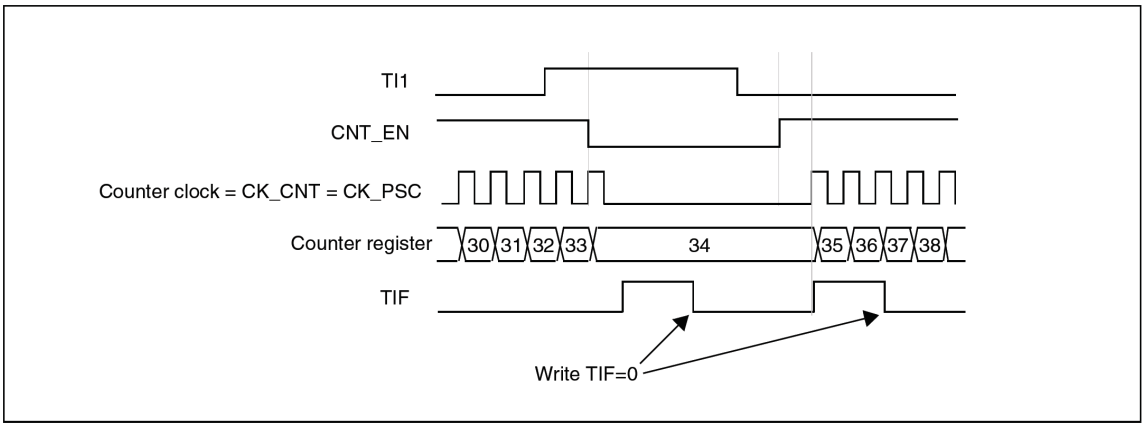
按照选中的输入端电平使能计数器。

在如下的例子中，计数器只在 TI1 为低时向上计数：

- 配置通道 1 以检测 TI1 上的低电平。配置输入滤波器带宽 (本例中，不需要滤波，所以保持 IC1F=0000)。触发操作中不使用捕获预分频器，所以不需要配置。CC1S 位用于选择输入捕获源，置 TIMx_CCMR1 寄存器中 CC1S=01。置 TIMx_CCER 寄存器中 CC1P=1 和 CC1NP=0 以确定极性 (只检测低电平)。
- 置 TIMx_SMCR 寄存器中 SMS=101，配置定时器为门控模式；置 TIMx_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。
- 置 TIMx_CR1 寄存器中 CEN=1，启动计数器。在门控模式下，如果 CEN=0，则计数器不能启动，不论触发输入电平如何。

只要 TI1 为低，计数器开始依据内部时钟计数，在 TI1 变高时停止计数。当计数器开始或停止时都设置 TIMx_SR 中的 TIF 标置。 TI1 上升沿和计数器实际停止之间的延时，取决于 TI1 输入端的重同步电路。

图 127. 门控模式下的控制电路



1. 配置“CCxP=CCxNP=1” (同时检测上升和下降沿) 在门控模式下无效，因为门控模式是电平控制而不是边沿。

从模式：触发模式

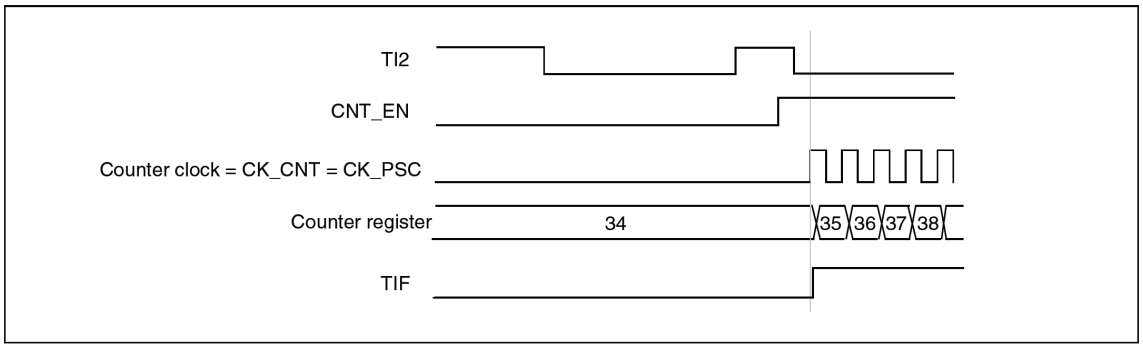
输入端上选中的事件启动计数器。

在下面的例子中，计数器在 TI2 输入的上升沿开始向上计数：

- 配置通道 2 检测 TI2 的上升沿。配置输入滤波器带宽 (本例中，不需要任何滤波器，保持 IC2F=0000)。触发操作中不使用捕获预分频器，不需要配置。CC2S 位只用于选择输入捕获源，置 TIMx_CCMR1 寄存器中 CC2S=01。置 TIMx_CCER 寄存器中 CC2P=1 和 CC1NP=0 以确定极性 (只检测低电平)。
- 置 TIMx_SMCR 寄存器中 SMS=110，配置定时器为触发模式；置 TIMx_SMCR 寄存器中 TS=110，选择 TI2 作为输入源。

当 TI2 出现一个上升沿时，计数器开始在内部时钟驱动下计数，同时设置 TIF 标志。
TI2 上升沿和计数器启动计数之间的延时，取决于 TI2 输入端的重同步电路。

图 128. 触发器模式下的控制电路



从模式：外部时钟模式 2 + 触发模式

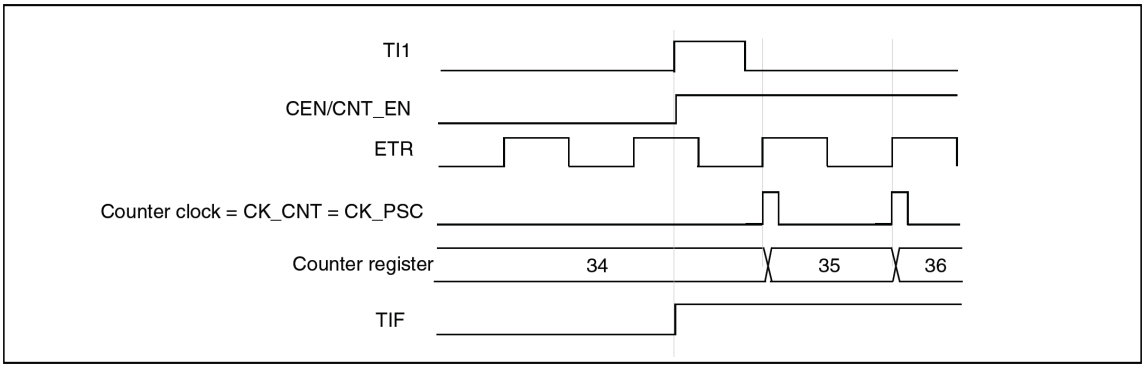
外部时钟模式 2 可以与另一种从模式 (外部时钟模式 1 和编码器模式除外) 一起使用。这时，ETR 信号被用作外部时钟的输入，在复位模式、门控模式或触发模式时可以选择另一个输入作为触发输入。不建议使用 TIMx_SMCR 寄存器的 TS 位选择 ETR 作为 TRGI。

下面的例子中，TI1 上出现一个上升沿之后，计数器即在 ETR 的每一个上升沿向上计数一次：

1. 通过 TIMx_SMCR 寄存器配置外部触发输入电路：
   - ETF=0000：没有滤波
   - ETPS=00：不用预分频器
   - ETP=0：检测 ETR 的上升沿，置 ECE=1 使能外部时钟模式 2
2. 按如下配置通道 1，检测 TI 的上升沿：
   - IC1F=0000：没有滤波
   - 触发操作中不使用捕获预分频器，不需要配置
   - 置 TIMx_CCMR1 寄存器中 CC1S=01，选择输入捕获源
   - 置 TIMx_CCER 寄存器中 CC1P=0 和 CC1NP=0 以确定极性 (只检测上升沿)
3. 置 TIMx_SMCR 寄存器中 SMS=110，配置定时器为触发模式。置 TIMx_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。

当 TI1 上出现一个上升沿时，TIF 标志被设置，计数器开始在 ETR 的上升沿计数。
ETR 信号的上升沿和计数器实际复位间的延时，取决于 ETRP 输入端的重同步电路。

图 129. 外部时钟模式 2 ＋触发模式下的控制电路

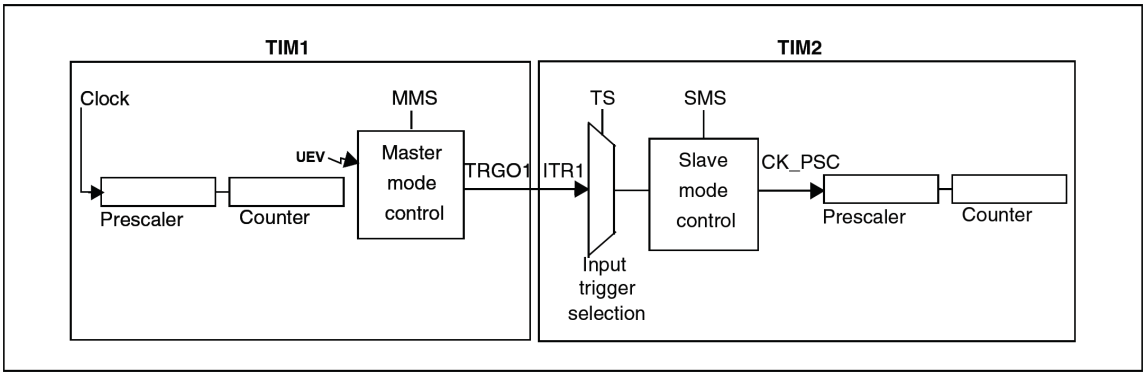


16.3.15定时器同步

所有 TIMx 定时器在内部相连，用于定时器同步或链接。当一个定时器处于主模式时，它可以对另一个处于从模式的定时器的计数器进行复位、启动、停止或提供时钟等操作。
下图显示了触发选择和主模式选择模块的概况

使用一个定时器作为另一个定时器的预分频器

图 130. 主 / 从定时器的例子



如：可以配置定时器 1 作为定时器 2 的预分频器。参考图 130，进行下述操作：

- 配置定时器 1 为主模式，它可以在每一个更新事件 UEV 时输出一个周期性的触发信号。在 TIM1_CR2 寄存器的 MMS=’010’时，每当产生一个更新事件时在 TRGO1 上输出一个上升沿信号。
- 连接定时器 1 的 TRGO1 输出至定时器 2，设置 TIM2_SMCR 寄存器的 TS=’000’，配置定时器 2 为使用 ITR1 作为内部触发的从模式。
- 然后把从模式控制器置于外部时钟模式 1(TIM2_SMCR 寄存器的 SMS=111)；这样定时器 2 即可由定时器 1 周期性的上升沿 (即定时器 1 的计数器溢出) 信号驱动。
- 最后，必须设置相应 (TIMx_CR1 寄存器) 的 CEN 位分别启动两个定时器。

注：　如果 OCx 已被选中为定时器 1 的触发输出 (MMS=1xx)，它的上升沿用于驱动定时器 2 的计数器。

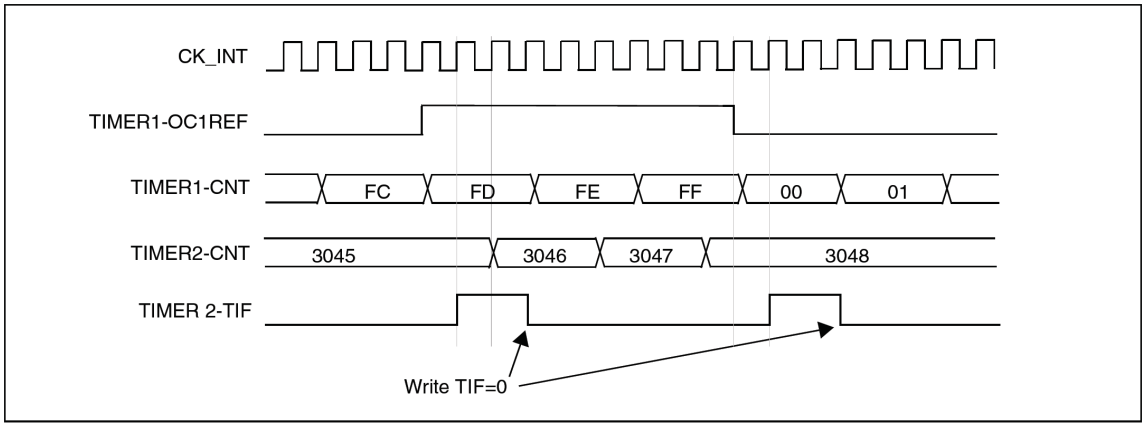
使用一个定时器使能另一个定时器

在这个例子中，定时器 2 的使能由定时器 1 的输出比较控制。参考图 130 的连接。只当定时器 1 的 OC1REF 为高时，定时器 2 才对分频后的内部时钟计数。两个定时器的时钟频率都是由预分频器 对 CK_INT 除以 3(fCK_CNT=fCK_INT/3) 得到。

- 配置定时器 1 为主模式，送出它的输出比较参考信号 (OC1REF) 为触发输出 (TIM1_CR2 寄存 器的 MMS=100)
- 配置定时器 1 的 OC1REF 波形 (TIM1_CCMR1 寄存器)
- 配置定时器 2 从定时器 1 获得输入触发 (TIM2_SMCR 寄存器的 TS=000)
- 配置定时器 2 为门控模式 (TIM2_SMCR 寄存器的 SMS=101)
- 置 TIM2_CR1 寄存器的 CEN=1 以使能定时器 2
- 置 TIM1_CR1 寄存器的 CEN=1 以启动定时器 1

注： 定时器 2 的时钟不与定时器 1 的时钟同步，这个模式只影响定时器 2 计数器的使能信号。

图 131. 定时器 1 的 OC1REF 控制定时器 2

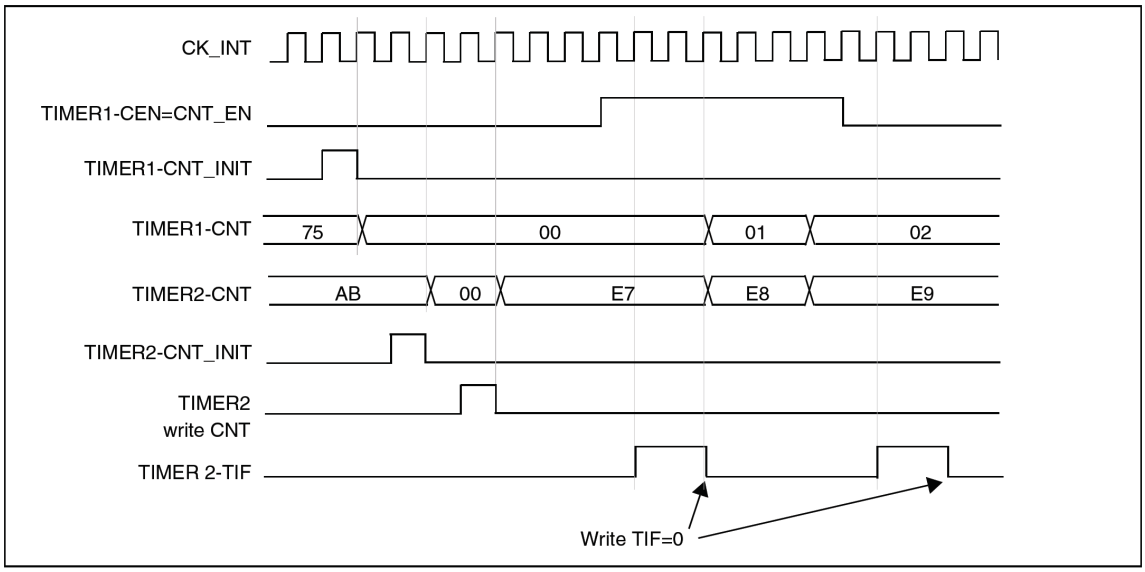


在图 131 的例子中，在定时器 2 启动之前，它们的计数器和预分频器未被初始化，因此它们从当前的数值开始计数。可以在启动定时器 1 之前复位 2 个定时器，使它们从给定的数值开始，即在定时器计数器中写入需要的任意数值。写 TIMx_EGR 寄存器的 UG 位即可复位定时器。

在下一个例子中，需要同步定时器 1 和定时器 2。定时器 1 是主模式并从 0 开始，定时器 2 是从模式并从 0xE7 开始；2 个定时器的预分频器系数相同。写’0’到 TIM1_CR1 的 CEN 位将禁止定时器 1，定时器 2 随即停止。

- 配置定时器 1 为主模式，送出输出比较 1 参考信号 (OC1REF) 做为触发输出 (TIM1_CR2 寄存器的 MMS=100)。
- 配置定时器 1 的 OC1REF 波形 (TIM1_CCMR1 寄存器)。
- 配置定时器 2 从定时器 1 获得输入触发 (TIM2_SMCR 寄存器的 TS=000)
- 配置定时器 2 为门控模式 (TIM2_SMCR 寄存器的 SMS=101)
- 置 TIM1_EGR 寄存器的 UG=’1’，复位定时器 1。
- 置 TIM2_EGR 寄存器的 UG=’1’，复位定时器 2。
- 写’0xE7’至定时器 2 的计数器 (TIM2_CNTL)，初始化它为 0xE7。
- 置 TIM2_CR1 寄存器的 CEN=’1’以使能定时器 2。
- 置 TIM1_CR1 寄存器的 CEN=’1’以启动定时器 1。
- 置 TIM1_CR1 寄存器的 CEN=’0’以停止定时器 1。

图 132. 通过使能定时器 1 可以控制定时器 2

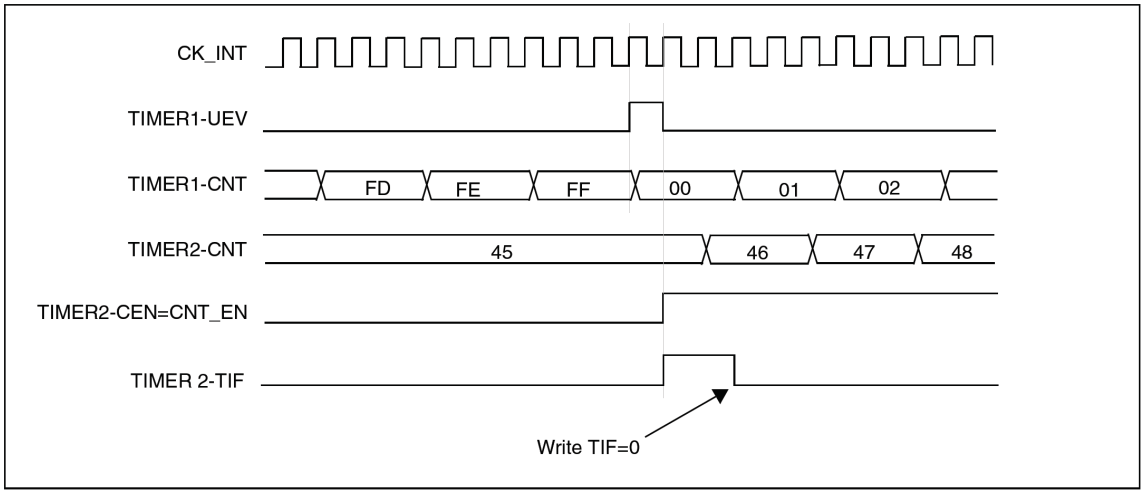


使用一个定时器去启动另一个定时器

在这个例子中，使用定时器 1 的更新事件使能定时器 2。参考图 130 的连接。一旦定时器 1 产生更新事件，定时器 2 即从它当前的数值 (可以是非 0) 按照分频的内部时钟开始计数。在收到触发信号时，定时器 2 的 CEN 位被自动地置’1’，同时计数器开始计数直到写’0’到 TIM2_CR1 寄存器的 CEN 位。两个定时器的时钟频率都是由预分频器对 CK_INT 除以 3(fCK_CNT=fCK_INT/3)。

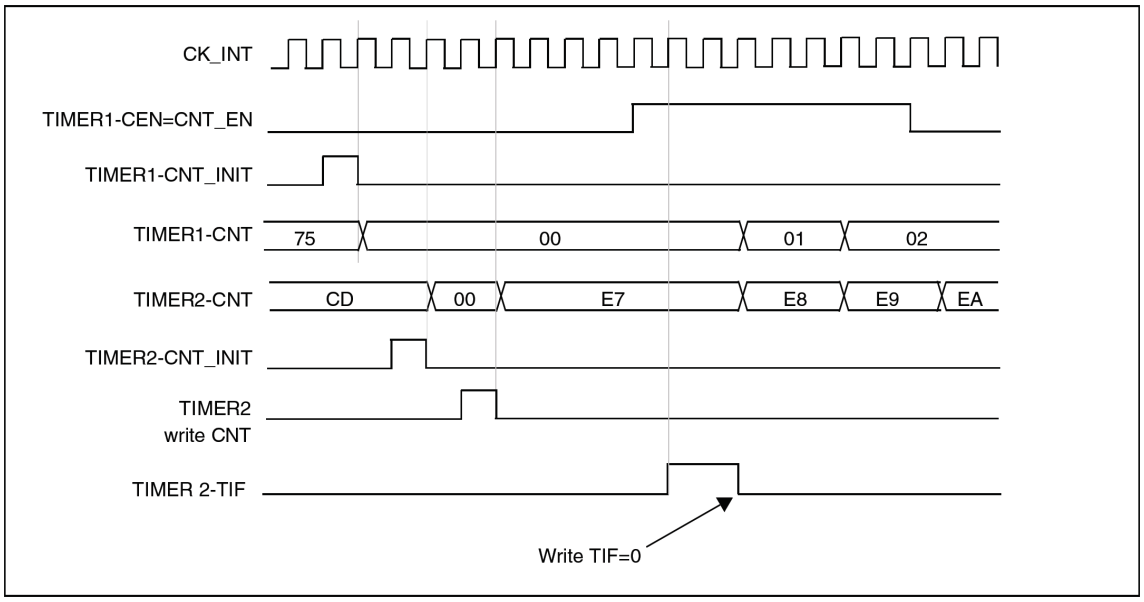
- 配置定时器 1 为主模式，送出它的更新事件 (UEV) 做为触发输出 (TIM1_CR2 寄存器的 MMS=010)。
- 配置定时器 1 的周期 (TIM1_ARR 寄存器)。
- 配置定时器 2 从定时器 1 获得输入触发 (TIM2_SMCR 寄存器的 TS=000)
- 配置定时器 2 为触发模式 (TIM2_SMCR 寄存器的 SMS=110)
- 置 TIM1_CR1 寄存器的 CEN=1 以启动定时器 1。

图 133. 使用定时器 1 的更新触发定时器 2



在上一个例子中，可以在启动计数之前初始化两个计数器。图 134 显示在与图 133 相同配置情况下，使用触发模式而不是门控模式 (TIM2_SMCR 寄存器的 SMS=110) 的动作。

图 134. 利用定时器 1 的使能触发定时器 2



使用一个定时器作为另一个的预分频器

这个例子使用定时器 1 作为定时器 2 的预分频器。参考图 130 的连接，配置如下：

- 配置定时器 1 为主模式，送出它的更新事件 UEV 做为触发输出 (TIM1_CR2 寄存器的 MMS=’010’)。然后每次计数器溢出时输出一个周期信号。
- 配置定时器 1 的周期 (TIM1_ARR 寄存器)。
- 配置定时器 2 从定时器 1 获得输入触发 (TIM2_SMCR 寄存器的 TS=000)
- 配置定时器 2 使用外部时钟模式 (TIM2_SMCR 寄存器的 SMS=111)
- 置 TIM1_CR2 寄存器的 CEN=1 以启动定时器 2。
- 置 TIM1_CR1 寄存器的 CEN=1 以启动定时器 1。

使用一个外部触发同步地启动 2 个定时器

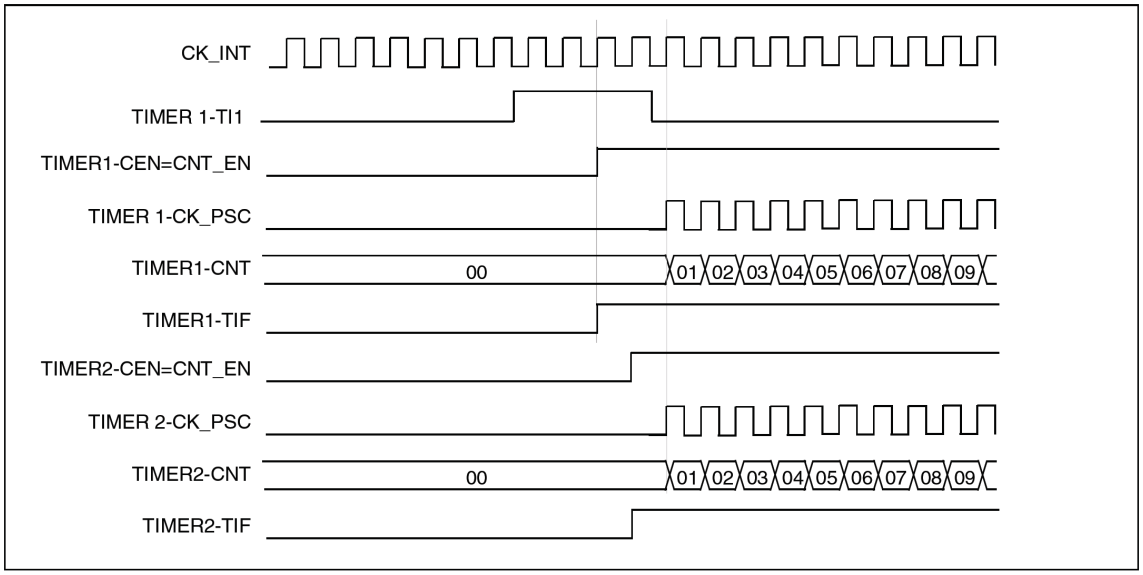
这个例子中当定时器 1 的 TI1 输入上升时使能定时器 1，使能定时器 1 的同时使能定时器 2，参见图 130。为保证计数器的对齐，定时器 1 必须配置为主 / 从模式 (对应 TI1 为从，对应定时器 2 为主)：

- 配置定时器 1 为主模式，送出它的使能做为触发输出 (TIM1_CR2 寄存器的 MS=’001’)。
- 配置定时器 1 为从模式，从 TI1 获得输入触发 (TIM1_SMCR 寄存器的 TS=’100’)。
- 配置定时器 1 为触发模式 (TIM1_SMCR 寄存器的 SMS=’110’)。
- 配置定时器 1 为主 / 从模式，TIM1_SMCR 寄存器的 MSM=’1’ 。
- 配置定时器 2 从定时器 1 获得输入触发 (TIM2_SMCR 寄存器的 TS=000)
- 配置定时器 2 为触发模式 (TIM2_SMCR 寄存器的 SMS=’110’)。

当定时器 1 的 TI1 上出现一个上升沿时，两个定时器同步地按照内部时钟开始计数，两个 TIF 标志也同时被设置。

注：　在这个例子中，在启动之前两个定时器都被初始化 (设置相应的 UG 位)，两个计数器都从 0 开始，但可以通过写入任意一个计数器寄存器 (TIMx_CNT) 在定时器间插入一个偏移。下图中能看到主 / 从模式下在定时器 1 的 CNT_EN 和 CK_PSC 之间有个延迟。

图 135. 使用定时器 1 的 TI1 输入触发定时器 1 和定时器 2



16.3.16 调试模式

当微控制器进入调试模式 (Cortex-M3 核心停止)，根据 DBGMCU 模块中 DBG_TIMx_STOP 的设置， TIMx 计数器或者继续正常操作，或者停止。

16.4 TIM2 和 TIM3 寄存器描述

关于在寄存器描述里面所用到的缩写，详见 31 页第 1.1 节。

可以用半字 (16 位) 或字 (32 位) 的方式操作这些外设寄存器。

16.4.1 TIM2 和 TIM3 控制寄存器 1 (TIM2_CR1 和 TIM3_CR1)

地址偏移 : 0x00

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CKD[1:0]		ARPE	CMS		DIR	OPM	URS	UDIS	CEN
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:10 保留，始终读为 ‘0’

位 9:8 CKD: 时钟分频因子 (Clock division)

定义在定时器时钟 (CK_INT) 频率与数字滤波器 (ETR，TIx) 使用的采样频率之间的分频比例。

00：tDTS = tCK_INT

01：tDTS = 2 x tCK_INT

10：tDTS = 4 x tCK_INT

11：保留

位 7 ARPE: 自动重装载预装载允许位 (Auto-reload preload enable)

0：TIMx_ARR 寄存器没有缓冲；

1：TIMx_ARR 寄存器缓冲器有效。

位 6:5 CMS: 选择中央对齐模式 (Center-aligned mode selection)

00：边沿对齐模式。计数器依据方向位 (DIR) 向上或向下计数。

01：中央对齐模式 1。计数器交替地向上和向下计数。配置为输出的通道 (TIMx_CCMRx 寄存器中 CCxS=00) 的输出比较中断标志位，只在计数器向下计数时被设置。

10：中央对齐模式 2。计数器交替地向上和向下计数。配置为输出的通道 (TIMx_CCMRx 寄存器中 CCxS=00) 的输出比较中断标志位，只在计数器向上计数时被设置。

11：中央对齐模式 3。计数器交替地向上和向下计数。配置为输出的通道 (TIMx_CCMRx 寄存器中 CCxS=00) 的输出比较中断标志位，在计数器向上和向下计数时均被设置。

注：在计数器开启时 (CEN=1)，不允许从边沿对齐模式转换到中央对齐模式。

位 4 DIR: 方向（Direction）

0：计数器向上计数；

1：计数器向下计数。

注：当计数器配置为中央对齐模式或编码器模式时，该位为只读。

位 3 OPM: 单脉冲模式 (One pulse mode)

0：在发生更新事件时，计数器不停止；

1：在发生下一次更新事件 (清除 CEN 位) 时，计数器停止

位 2 URS: 更新请求源 (Update request source)

软件通过该位选择 UEV 事件的源

0：如果使能了更新中断或 DMA 请求，则下述任一事件产生更新中断或 DMA 请求：

- 计数器溢出 / 下溢
- 设置 UG 位
- 从模式控制器产生的更新

1：如果使能了更新中断或 DMA 请求，则只有计数器溢出 / 下溢才产生更新中断或 DMA 请求。

位 1 UDIS: 禁止更新 (Update disable)

软件通过该位允许 / 禁止 UEV 事件的产生

0：允许 UEV。更新 (UEV) 事件由下述任一事件产生：

- 计数器溢出 / 下溢
- 设置 UG 位
- 从模式控制器产生的更新具有缓存的寄存器被装入它们的预装载值。

1：禁止 UEV。不产生更新事件，影子寄存器 (ARR、PSC、CCRx) 保持它们的值。如果设置了 UG 位或从模式控制器发出了一个硬件复位，则计数器和预分频器被重新初始化。

Bit 0 CEN: 使能计数器（Counter enable）

0：禁止计数器；

1：使能计数器。

注：在软件设置了 CEN 位后，外部时钟、门控模式和编码器模式才能工作。触发模式可以自动地通过硬件设置 CEN 位。

在单脉冲模式下，当发生更新事件时，CEN 被自动清除。

16.4.2 TIM2 和 TIM3 控制寄存器 2 (TIM2_CR2 和 TIM3_CR2)

地址偏移 : 0x04

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1S	MMS[2:0]			CCDS	Res.	Res.	Res.
								rw	rw	rw	rw	rw			

位 15:8 保留，必须保持为复位值 .

位 7 TI1S: TI1 选择（selection）

0: TIMx_CH1 引脚连到 TI1 输入；

1：TIMx_CH1、TIMx_CH2 和 TIMx_CH3 引脚经异或后连到 TI1 输入。

见 255 页 15.3.18 节：与霍尔传感器的接口。

位 6:4 MMS: 主模式选择 (Master mode selection)

这 3 位用于选择在主模式下送到从定时器的同步信息 (TRGO)。可能的组合如下：

000：复位 – TIMx_EGR 寄存器的 UG 位被用于作为触发输出 (TRGO)。如果是触发输入产生的 复位 (从模式控制器处于复位模式)，则 TRGO 上的信号相对实际的复位会有一个延迟。

001：使能 – 计数器使能信号 CNT_EN 被用于作为触发输出 (TRGO)。有时需要在同一时间启动 多个定时器或控制在一段时间内使能从定时器。计数器使能信号是通过 CEN 控制位和门控模式 下的触发输入信号的逻辑或产生。

当计数器使能信号受控于触发输入时，TRGO 上会有一个延迟，除非选择了主 / 从模式 (见 TIMx_SMCR 寄存器中 MSM 位的描述)。

010：更新 – 更新事件被选为触发输入 (TRGO)。例如，一个主定时器的时钟可以被用作一个从 定时器的预分频器。

011：比较脉冲 – 在发生一次捕获或一次比较成功时，当要设置 CC1IF 标志时 (即使它已经为 高)，触发输出送出一个正脉冲 (TRGO)。

100：比较 – OC1REF 信号被用于作为触发输出 (TRGO)。

101：比较 – OC2REF 信号被用于作为触发输出 (TRGO)。

110：比较 – OC3REF 信号被用于作为触发输出 (TRGO)。

111：比较 – OC4REF 信号被用于作为触发输出 (TRGO)。

位 3 CCDS: 捕获 / 比较的 DMA 选择（Capture/compare DMA selection）

0：当发生 CCx 事件时，送出 CCx 的 DMA 请求；

1：当发生更新事件时，送出 CCx 的 DMA 请求。

位 2:0 保留，始终读为‘0’

16.4.3 TIM2 和 TIM3 从模式控制寄存器 (TIM2_SMCR 和 TIM3_SMCR)

地址偏移 : 0x08

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			Res.	SMS[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw

位 15 ETP: 外部触发极性 (External trigger polarity)

该位选择是用 ETR 还是 ETR 的反相来作为触发操作

0：ETR 不反相，高电平或上升沿有效；

1：ETR 被反相，低电平或下降沿有效。

位 14 ECE: 外部时钟使能位 (External clock enable)

该位启用外部时钟模式 2

0：禁止外部时钟模式 2；

1：使能外部时钟模式 2。计数器由 ETRF 信号上的任意有效边沿驱动。

注 1：设置 ECE 位与选择外部时钟模式 1 并将 TRGI 连到 ETRF(SMS=111 和 TS=111) 具有相同功效。

注 2：下述从模式可以与外部时钟模式 2 同时使用：复位模式、门控模式和触发模式；但是，这时 TRGI 不能连到 ETRF(TS 位不能是’111’)。

注 3：外部时钟模式 1 和外部时钟模式 2 同时被使能时，外部时钟的输入是 ETRF。

位 13:12 ETPS: 外部触发预分频 (External trigger prescaler)

外部触发信号 ETRP 的频率必须最多是 CK_INT 频率的 1/4。当输入较快的外部时钟时，可以使用预分频降低 ETRP 的频率。

00：关闭预分频；

01：ETRP 频率除以 2；

10：ETRP 频率除以 4；

11：ETRP 频率除以 8。

位 11:8 ETF[3:0]: 外部触发滤波 (External trigger filter)

这些位定义了对 ETRP 信号采样的频率和对 ETRP 数字滤波的带宽。实际上，数字滤波器是一个 事件计数器，它记录到 N 个事件后会产生一个输出的跳变。

0000：无滤波器，以 f_{DTS} 采样

0001：采样频率 $f_{SAMPLING}=f_{CK\_INT}$，N=2

0010：采样频率 $f_{SAMPLING}=f_{CK\_INT}$，N=4

0011：采样频率 $f_{SAMPLING}=f_{CK\_INT}$，N=8

0100：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=6

0101：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=8

0110：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=6

0111：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=8

1000：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=6

1001：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=8

1010：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=5

1011：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=6

1100：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=8

1101：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=5

1110：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=6

1111：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=8

位 7 MSM: 主 / 从模式 (Master/slave mode)

0：无作用；

1：触发输入 (TRGI) 上的事件被延迟了，以允许在当前定时器 (通过 TRGO) 与它的从定时器间的完美同步。这对要求把几个定时器同步到一个单一的外部事件时是非常有用的。

位 6:4 TS: 触发选择 (Trigger selection)

这 3 位选择用于同步计数器的触发输入。

000：内部触发 0(ITR0)

001：内部触发 1(ITR1)

010：内部触发 2(ITR2)

011：内部触发 3(ITR3)

100：TI1 的边沿检测器 (TI1F_ED)

101：滤波后的定时器输入 1(TI1FP1)

110：滤波后的定时器输入 2(TI2FP2)

111：外部触发输入 (ETRF)

关于每个定时器中 ITRx 的细节，参见页 330 表 47。

注：这些位只能在未用到 (如 SMS=000) 时被改变，以避免在改变时产生错误的边沿检测。

位 3 保留，必须保持为复位值

位 2:0 SMS: 从模式选择 (Slave mode selection)

当选择了外部信号，触发信号 (TRGI) 的有效边沿与选中的外部输入极性相关 (见输入控制寄存器 和控制寄存器的说明)

000：关闭从模式 – 如果 CEN=1，则预分频器直接由内部时钟驱动。

001：编码器模式 1 – 根据 TI1FP1 的电平，计数器在 TI2FP2 的边沿向上 / 下计数。

010：编码器模式 2 – 根据 TI2FP2 的电平，计数器在 TI1FP1 的边沿向上 / 下计数。

011：编码器模式 3 – 根据另一个信号的输入电平，计数器在 TI1FP1 和 TI2FP2 的边沿向上 / 下计数。

100：复位模式 – 选中的触发输入 (TRGI) 的上升沿重新初始化计数器，并且产生一个更新寄存器的信号。

101：门控模式 – 当触发输入 (TRGI) 为高时，计数器的时钟开启。一旦触发输入变为低，则计数器停止 (但不复位)。计数器的启动和停止都是受控的。

110：触发模式 – 计数器在触发输入 TRGI 的上升沿启动 (但不复位)，只有计数器的启动是受控的。

111：外部时钟模式 1 – 选中的触发输入 (TRGI) 的上升沿驱动计数器。

注：如果 TI1F_EN 被选为触发输入 (TS=100) 时，不要使用门控模式。这是因为，TI1F_ED 在每次 TI1F 变化时输出一个脉冲，然而门控模式是要检查触发输入的电平。

表 47. TIM2 和 TIM3 内部触发连接

Slave TIM	ITR0 (TS = 000)	ITR1 (TS = 001)	ITR2 (TS = 010)	ITR3 (TS = 011)
TIM2	TIM1	TIM15	TIM3	TIM14
TIM3	TIM1	TIM2	TIM15	TIM14

16.4.4 TIM2 和 TIM3 DMA/ 中断允许寄存器 (TIM2_DIER 和 TIM3_DIER)

地址偏移 : 0x0C

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	Res.	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Res.	TIE	Res.	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	rw		rw	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw

位 15 保留，必须保持为复位值 .

位 14 TDE: 触发 DMA 请求允许

0: 触发 DMA 请求禁止 .

1: 触发 DMA 请求允许

位 13 保留，读始终为 ‘0’

位 12 CC4DE: 捕捉 / 比较 4 DMA 请求允许

0: CC4 DMA 请求禁止 .

1: CC4 DMA 请求允许

位 11 CC3DE: 捕捉 / 比较 3 DMA 请求允许

0: CC3 DMA 请求禁止 .

1: CC3 DMA 请求允许

位 10 CC2DE: 捕捉 / 比较 2 DMA 请求允许

0: CC2 DMA 请求禁止 .

1: CC2 DMA 请求允许

位 9 CC1DE: 捕捉 / 比较 1 DMA 请求允许

0: CC1 DMA 请求禁止 .

1: CC1 DMA 请求允许

位 8 UDE: 更新 DMA 请求允许

0: 更新 DMA 请求禁止 .

1: 更新 DMA 请求允许

位 7 保留，必须保持为复位值 .

位 6 TIE: 触发中断允许

0: 触发中断禁止 .

1: 触发中断允许

位 5 保留，必须保持为复位值 .

位 4 CC4IE: 捕捉 / 比较 4 中断允许

0: CC4 中断禁止 .

1: CC4 中断允许

位 3 CC3IE: 捕捉 / 比较 3 中断允许

0: CC3 中断禁止

1: CC3 中断允许

位 2 CC2IE: 捕捉 / 比较 2 中断允许
0: CC2 中断禁止
1: CC2 中断允许
位 1 CC1IE: 捕捉 / 比较 1 中断允许
0: CC1 中断禁止
1: CC1 中断允许
位 0 UIE: 更新 interrupt 允许
0: 更新中断禁止
1: 更新中断允许

16.4.5 TIM2 和 TIM3 状态寄存器 (TIM2_SR 和 TIM3_SR)

地址偏移 : 0x10
复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CC4OF	CC3OF	CC2OF	CC1OF	Res.	Res.	TIF	Res.	CC4IF	CC3IF	CC2IF	CC1IF	UIF
			rc_w0	rc_w0	rc_w0	rc_w0			rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

位 15:13 保留，始终读为‘0’
位 12 CC4OF: 捕捉 / 比较 4 重复捕捉标志
参见 CC1OF 描述
位 11 CC3OF: 捕捉 / 比较 3 重复捕捉标志
参见 CC1OF 描述
位 10 CC2OF: 捕捉 / 比较 2 重复捕捉标志
参见 CC1OF 描述
位 9 CC1OF: 捕捉 / 比较 1 重复捕捉标志 (Capture/Compare 1 overcapture flag)
仅当相应的通道被配置为输入捕获时，该标记可由硬件置’1’。写’0’可清除该位。
0：无重复捕获产生；
1：当计数器的值被捕获到 TIMx_CCR1 寄存器时，CC1IF 的状态已经为’1’。
位 8:7 保留，始终读为‘0’
位 6 TIF: 触发器中断标记 (Trigger interrupt flag)
当发生触发事件 (当从模式控制器处于除门控模式外的其它模式时，在 TRGI 输入端检测到有效边沿，或门控模式下的任一边沿) 时由硬件对该位置’1’。它由软件清’0’。
0：无触发器事件产生；
1：触发器中断等待响应。
位 5 保留，始终读为‘0’
位 4 CC4IF: 捕捉 / 比较 4 中断标志
参见 CC1IF 描述
位 3 CC3IF: 捕捉 / 比较 3 中断标志
参见 CC1IF 描述
位 2 CC2IF: 捕捉 / 比较 2 中断标志
参见 CC1IF 描述

位 1 CC1IF: 捕捉 / 比较 1 中断标志 (Capture/Compare 1 interrupt flag)

如果通道 CC1 配置为输出模式：

当计数器值与比较值匹配时该位由硬件置’1’，但在中心对称模式下除外 (参考 TIMx_CR1 寄存器的 CMS 位)。它由软件清’0’。

0：无匹配发生；

1：TIMx_CNT 的值与 TIMx_CCR1 的值匹配。

当TIMx_CCR1的内容大于TIMx_ARR时，在计数器溢出（向上或向上/向下计数模式）或计数器下溢出时（向下计数模式）时 CC1IF 位变高

如果通道 CC1 配置为输入模式：

当捕获事件发生时该位由硬件置’1’，它由软件清’0’或通过读TIMx_CCR1清’0’。

0：无输入捕获产生；

1：计数器值已被捕获 (拷贝) 至 TIMx_CCR1(在 IC1 上检测到与所选极性相同的边沿)。

位 0 UIF: 更新中断标记 (Update interrupt flag)

当产生更新事件时该位由硬件置’1’。它由软件清’0’。

0：无更新事件产生；

1：更新中断等待响应。

当寄存器被更新时该位由硬件置’1’：

- 若 TIMx_CR1 寄存器的 UDIS=0、URS=0，当 TIMx_EGR 寄存器的 UG=1 时产生更新事件 (软件对计数器 CNT 重新初始化)；
- 若 TIMx_CR1 寄存器的 UDIS=0、URS=0，当计数器 CNT 被触发事件（参见同步控制寄存器的描述）重初始化时产生

16.4.6 TIM2 和 TIM3 事件产生寄存器 (TIM2_EGR 和 TIM3_EGR)

地址偏移 : 0x14

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TG	Res.	CC4G	CC3G	CC2G	CC1G	UG
									w		w	w	w	w	w

位 15:7 保留，必须保持为复位值 .

位 6 TG: 产生触发事件 (Trigger generation)
该位由软件置’1’，用于产生一个触发事件，由硬件自动清’0’。
0：无动作；
1：TIMx_SR 寄存器的 TIF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。

位 5 保留，必须保持为复位值 .

位 4 CC4G: 捕捉 / 比较 4 产生
参见 CC1G 描述

位 3 CC3G: 捕捉 / 比较 3 产生
参见 CC1G 描述

位 2 CC2G: 捕捉 / 比较 2 产生
参见 CC1G 描述

位 1 CC1G: 捕捉 / 比较 1 产生
该位由软件置’1’，用于产生一个捕获 / 比较事件，由硬件自动清’0’。
0：无动作；
1：在通道 CC1 上产生一个捕获 / 比较事件：
若通道 CC1 配置为输出：
设置 CC1IF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。
若通道 CC1 配置为输入：
当前的计数器值捕获至 TIMx_CCR1 寄存器；设置 CC1IF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。若 CC1IF 已经为 1，则设置 CC1OF=1。

位 0 UG: 产生更新事件 (Update generation)
该位由软件置’1’，由硬件自动清’0’。
0：无动作；
1：重新初始化计数器，并产生一个更新事件。注意预分频器的计数器也被清’0’(但是预分频系数不变)。若在中心对称模式下或 DIR=0(向上计数) 则计数器被清’0’，若 DIR=1(向下计数) 则计数器取 TIMx_ARR 的值。

16.4.7 TIM2 和 TIM3 捕捉 / 比较模式寄存器 1 (TIM2_CCMR1 和 TIM3_CCMR1)

地址偏移 : 0x18

复位值 : 0x0000

通道可用于输入 (捕获模式) 或输出 (比较模式)，通道的方向由相应的 CCxS 定义。该寄存器其它位的作用在输入和输出模式下不同。OCxx 描述了通道在输出模式下的功能，ICxx 描述了通道在输入模式下的功能。因此必须注意，同一个位在输出模式和输入模式下的功能是不同的。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]				IC1PSC[1:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

输出比较模式

位 15 OC2CE: 输出比较 2 清除使能

位 14:12 OC2M[2:0]: 输出比较 2 模式

位 11 OC2PE: 输出比较 2 预装载使能

位 10 OC2FE: 输出比较 2 快速使能

位 9:8 CC2S[1:0]: 捕捉 / 比较 2 选择 (Capture/Compare 2 selection)

该位定义通道的方向 (输入 / 输出)，及输入脚的选择：

00：CC2 通道被配置为输出；

01：CC2 通道被配置为输入，IC2 映射在 TI2 上；

10：CC2 通道被配置为输入，IC2 映射在 TI1 上；

11：CC2 通道被配置为输入，IC2 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)。

注：CC2S 仅在通道关闭时 (TIMx_CCER 寄存器的 CC2E=’0’) 才是可写的。

位 7 OC1CE: 输出比较 1 清除使能

0: OC1REF 不受 ETRF 输入的影响

1: 一旦检测到 ETRF 输入高电平，清除 OC1REF=0

位 6:4 OC1M: 输出比较 1 模式 (Output compare 1 mode)

该 3 位定义了输出参考信号 OC1REF 的动作，而 OC1REF 决定了 OC1 和 OC1N 的值。OC1REF 是高电平有效，而 OC1 和 OC1N 的有效电平取决于 CC1P 和 CC1PN 位。

000：冻结。输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 间的比较对 OC1REF 不起作用；（此模式仅用于产生时基）
001 ：匹配时设置通道 1 为有效电平。 当计数器 TIMx_CNT 的值与捕获 / 比较寄存器 1 (TIMx_CCR1) 相同时，强制 OC1REF 为高。
010 ：匹配时设置通道 1 为无效电平。 当计数器 TIMx_CNT 的值与捕获 / 比较寄存器 1 (TIMx_CCR1) 相同时，强制 OC1REF 为低。
011：翻转。当 TIMx_CCR1=TIMx_CNT 时，翻转 OC1REF 的电平。
100：强制为无效电平。强制 OC1REF 为低。
101：强制为有效电平。强制 OC1REF 为高。
110：PWM 模式 1 － 在向上计数时，一旦 TIMx_CNT<TIMx_CCR1 时通道 1 为有效电平，否则为无效电平；在向下计数时，一旦 TIMx_CNT>TIMx_CCR1 时通道 1 为无效电平 (OC1REF=0)，否 则为有效电平 (OC1REF=1)。
111：PWM 模式 2 － 在向上计数时，一旦 TIMx_CNT<TIMx_CCR1 时通道 1 为无效电平，否则为有效电平；在向下计数时，一旦 TIMx_CNT>TIMx_CCR1 时通道 1 为有效电平，否则为无效电平。
注 1：一旦 LOCK 级别设为 3(TIMx_BDTR 寄存器中的 LOCK 位) 并且 CC1S=’00’(该通道配置成输出) 则该位不能被修改。
注 2：在 PWM 模式 1 或 PWM 模式 2 中，只有当比较结果改变了或在输出比较模式中从冻结模式 切换到 PWM 模式时，OC1REF 电平才改变。

位 3 OC1PE: 输出比较 1 预装载使能 (Output compare 1 preload enable)
0：禁止 TIMx_CCR1 寄存器的预装载功能，可随时写入 TIMx_CCR1 寄存器，并且新写入的数值立即起作用。
1：开启 TIMx_CCR1 寄存器的预装载功能，读写操作仅对预装载寄存器操作，TIMx_CCR1 的预装载值在更新事件到来时被传送至当前寄存器中。
注 1：一旦 LOCK 级别设为 3(TIMx_BDTR 寄存器中的 LOCK 位) 并且 CC1S=’00’(该通道配置成输出) 则该位不能被修改。
注 2：仅在单脉冲模式下 (TIMx_CR1 寄存器的 OPM=’1’)，可以在未确认预装载寄存器情况下使用 PWM 模式，否则其动作不确定。

位 2 OC1FE: 输出比较 1 快速使能 (Output compare 1 fast enable)
该位用于加快 CC 输出对触发器输入事件的响应。
0：根据计数器与 CCR1 的值，CC1 正常操作，即使触发器是打开的。当触发器的输入出现一个有效沿时，激活 CC1 输出的最小延时为 5 个时钟周期。
1：输入到触发器的有效沿的作用就象发生了一次比较匹配。因此，OC 被设置为比较电平而与比较结果无关。采样触发器的有效沿和 CC1 输出间的延时被缩短为 3 个时钟周期。 该位只在通道被配置成 PWM1 或 PWM2 模式时起作用。

位 1:0 CC1S: 捕捉 / 比较 1 选择 (Capture/Compare 1 selection)
这 2 位定义通道的方向 (输入 / 输出)，及输入脚的选择：
00：CC1 通道被配置为输出；
01：CC1 通道被配置为输入，IC1 映射在 TI1 上；
10：CC1 通道被配置为输入，IC1 映射在 TI2 上；
11：CC1 通道被配置为输入，IC1 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)。
注：CC1S 仅在通道关闭时 (TIMx_CCER 寄存器的 CC1E=’0’) 才是可写的。

输入捕捉模式

位 15:12 IC2F: 输入捕捉 2 滤波器

位 11:10 IC2PSC[1:0]: 输入捕捉 2 预分频

位 9:8 CC2S: 捕捉 / 比较 2 选择 (Capture/compare 2 selection)

这 2 位定义通道的方向 (输入 / 输出)，及输入脚的选择：

00：CC2 通道被配置为输出；

01：CC2 通道被配置为输入，IC2 映射在 TI2 上；

10：CC2 通道被配置为输入，IC2 映射在 TI1 上；

11：CC2 通道被配置为输入，IC2 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)。

注：CC2S 仅在通道关闭时 (TIMx_CCER 寄存器的 CC2E=’0’) 才是可写的。

位 7:4 IC1F: 输入捕获 1 滤波器 (Input capture 1 filter)

这几位定义了 TI1 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成，它记录到 N 个事件后会产生一个输出的跳变：

0000：无滤波器，以 f_{DTS} 采样

0001：采样频率 $f_{SAMPLING}$=fCK_INT，N=2

0010：采样频率 $f_{SAMPLING}$=fCK_INT，N=4

0011：采样频率 $f_{SAMPLING}$=fCK_INT，N=8

0100：采样频率 $f_{SAMPLING}$=fDTS/2，N=6

0101：采样频率 $f_{SAMPLING}$=fDTS/2，N=8

0110：采样频率 $f_{SAMPLING}$=fDTS/4，N=6

0111：采样频率 $f_{SAMPLING}$=fDTS/4，N=8

1000：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=6

1001：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=8

1010：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=5

1011：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=6

1100：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=8

1101：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=5

1110：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=6

1111：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=8

注：在现在的芯片版本中，当 ICxF[3:0]=1、2 或 3 时，公式中的 fDTS 由 CK_INT 替代。

位 3:2 IC1PSC: 输入捕捉 1 预分频

这 2 位定义了 CC1 输入 (IC1) 的预分频系数。 一旦 CC1E=’0’(TIMx_CCER 寄存器中)，则预分频器复位。

00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获；

01：每 2 个事件触发一次捕获；

10：每 4 个事件触发一次捕获；

11：每 8 个事件触发一次捕获。

位 1:0 CC1S: 捕捉 / 比较 1 选择

这 2 位定义通道的方向 (输入 / 输出)，及输入脚的选择：

00：CC1 通道被配置为输出；

01：CC1 通道被配置为输入，IC1 映射在 TI1 上；

10：CC1 通道被配置为输入，IC1 映射在 TI2 上；

11：CC1 通道被配置为输入，IC1 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)。

注：CC1S 仅在通道关闭时 (TIMx_CCER 寄存器的 CC1E=’0’) 才是可写的。

16.4.8 TIM2 和 TIM3 捕捉 / 比较模式寄存器 2 (TIM2_CCMR2 和 TIM3_CCMR2)

地址偏移 : 0x1C

复位值 : 0x0000

参考前面 CCMR1 寄存器描述

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M[2:0]			OC4PE	OC4FE	CC4S[1:0]		OC3CE	OC3M[2:0]			OC3PE	OC3FE	CC3S[1:0]	
IC4F[3:0]				IC4PSC[1:0]				IC3F[3:0]				IC3PSC[1:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

输出比较模式

位 15 OC4CE: 输出比较 4 清除允许

位 14:12 OC4M: 输出比较 4 模式

位 11 OC4PE: 输出比较 4 预装载允许

位 10 OC4FE: 输出比较 4 快速允许

位 9:8 CC4S: 捕捉 / 比较 4 选择

这 2 位定义通道的方向 (输入 / 输出)，及输入脚的选择：

00：CC4 通道被配置为输出；

01：CC4 通道被配置为输入，IC4 映射在 TI4 上；

10：CC4 通道被配置为输入，IC4 映射在 TI3 上；

11：CC4 通道被配置为输入，IC4 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)。

注：CC4S 仅在通道关闭时 (TIMx_CCER 寄存器的 CC4E=’0’) 才是可写的。

位 7 OC3CE: 输出比较 3 清除允许

位 6:4 OC3M: 输出比较 3 模式

位 3 OC3PE: 输出比较 3 预装载允许

位 2 OC3FE: 输出比较 3 快速允许

位 1:0 CC3S: 捕捉 / 比较 3 选择

这 2 位定义通道的方向 (输入 / 输出)，及输入脚的选择：

00：CC3 通道被配置为输出；

01：CC3 通道被配置为输入，IC3 映射在 TI3 上；

10：CC3 通道被配置为输入，IC3 映射在 TI4 上；

11：CC3 通道被配置为输入，IC3 映射在 TRGI 上。此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)。

注：CC3S 仅在通道关闭时 (TIMx_CCER 寄存器的 CC3E=’0’) 才是可写的。

输入捕捉模式

位 15:12 IC4F: 输入捕捉 4 滤波器
位 11:10 IC4PSC: 输入捕捉 4 预分频
位 9:8 CC4S: 捕捉 / 比较 4 选择
这 2 位定义通道的方向 (输入 / 输出)，及输入脚的选择：
00：CC4 通道被配置为输出；
01：CC4 通道被配置为输入，IC4 映射在 TI4 上；
10：CC4 通道被配置为输入，IC4 映射在 TI3 上；
11：CC4 通道被配置为输入，IC4 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)。
注：CC4S 仅在通道关闭时 (TIMx_CCER 寄存器的 CC4E=’0’) 才是可写的。
位 7:4 IC3F: 输入捕捉 3 滤波器
位 3:2 IC3PSC: 输入捕捉 3 预分频
位 1:0 CC3S: 捕捉 / 比较 3 选择
这 2 位定义通道的方向 (输入 / 输出)，及输入脚的选择：
00：CC3 通道被配置为输出；
01：CC3 通道被配置为输入，IC3 映射在 TI3 上；
10：CC3 通道被配置为输入，IC3 映射在 TI3 上；
11：CC3 通道被配置为输入，IC3 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)。
注：CC3S 仅在通道关闭时 (TIMx_CCER 寄存器的 CC3E=’0’) 才是可写的。

16.4.9 TIM2 和 TIM3 捕捉 / 比较使能寄存器 (TIM2_CCER 和 TIM3_CCER)

地址偏移 : 0x20
复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4NP	Res.	CC4P	CC4E	CC3NP	Res.	CC3P	CC3E	CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E
rw		rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw

位 15 CC4NP: 捕捉 / 比较 4 输出极性 .
参考 CC1NP 描述
位 14 保留，始终读为 ‘0’
位 13 CC4P: 捕捉 / 比较 4 输出极性 .
参考 CC1P 描述
位 12 CC4E: 捕捉 / 比较 4 输出使能 .
参考 CC1E 描述
位 13 CC3NP: 捕捉 / 比较 3 输出极性 .
参考 CC1NP 描述
位 12 保留，始终读为 ‘0’
位 11:10 保留，始终读为 ‘0’
位 9 CC3P: 捕捉 / 比较 3 输出极性 .
参考 CC1P 描述

位 8 CC3E: 捕捉 / 比较 3 输出使能 .
参考 CC1E 描述

位 7 CC2NP: 捕捉 / 比较 2 输出极性 .
参考 CC1NP 描述

位 6 保留，始终读为‘0’

位 5 CC2P: 捕捉 / 比较 2 输出极性 .
参考 CC1P 描述

位 4 CC2E: 捕捉 / 比较 2 输出使能 .
参考 CC1E 描述

位 3 CC1NP: 捕捉 / 比较 1 输出极性 .
通道 CC1 配置为输出时，CC1NP 必须操持为清除，CC1NP ＝ 0;
通道 CC1 配置为输入时，CC1NP 与 CC1P 联合控制 TI1FP1/TI2FP1 的极性
参考 CC1P 描述 .

位 2 保留，始终读为‘0’

位 1 CC1P: 捕捉 / 比较 1 输出极性 .
通道 CC1 配置为输出 :
0: OC1 高有效
1: OC1 低有效
通道 CC1 配置为输入 :
CC1NP/CC1P 用于选择作为触发或捕获的信号 TI1FP1 和 TI2FP1 的极性
该位 IC1 还是 IC1 的反相信号。
00：不反相 / 上升沿：捕获发生在 TIxFP1 的上升沿（复位、外部时钟或触发模式的捕捉或触发），TIxFP1 不反相（在门控、编码器模式下的触发）。
00：反相 / 下降沿：捕获发生在 TIxFP1 的下降沿（复位、外部时钟或触发模式的捕捉或触发），TIxFP1 反相（在门控、编码器模式下的触发）。
10：保留，不使用此配置
11：不反相 / 上升和下降沿：捕获发生在 TIxFP1 的上升沿和下降沿（复位、外部时钟或触发模式的捕捉或触发），TIxFP1 不反相（门控模式的触发）。此配置不能用于编码器模式。

位 0 CC1E: 捕捉 / 比较 1 输出使能 .
CC1 通道配置为输出：
0： 关闭－ OC1 禁止输出。
1： 开启－ OC1 信号输出到对应的输出引脚。
CC1 通道配置为输入：
该位决定了计数器的值是否能捕获入 TIMx_CCR1 寄存器。
0：捕获禁止；
1：捕获使能。

表 48. 标准 OCx 通道的输出控制位

CCxE 位	**OCx** 输出状态
0	禁止输出 (OCx=0，OCx_EN=0)
1	OCx = OCxREF + 极性，OCx_EN=1

注： 连接到标准 OCx 通道的外部 I/O 引脚状态，取决于 OCx 通道状态和 GPIO 寄存器。

16.4.10 TIM2 和 TIM3 计数器 (TIM2_CNT 和 TIM3_CNT)

地址偏移 : 0x24

复位值 : 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT[31:16] (TIM2 only)															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 CNT[31:16]: 高 16 位值 (仅 TIM2).

位 15:0 CNT[15:0]: 低 16 位值

16.4.11 TIM2 和 TIM3 预分频 (TIM2_PSC 和 TIM3_PSC)

地址偏移 : 0x28

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 PSC[15:0]: 预分频值

计数器的时钟频率 CK_CNT 等于 fCK_PSC/(PSC[15:0]+1)。 PSC 包含了当更新事件产生时装入当前预分频器寄存器的值。

16.4.12 TIM2 和 TIM3 自动重装寄存器 (TIM2_ARR 和 TIM3_ARR)

地址偏移 : 0x2C

复位值 : 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARR[31:16] (TIM2 only)															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 ARR[31:16]: 自动重装高 16 位值 (仅 TIM2).

位 15:0 ARR[15:0]: 自动重装低 16 位值

ARR 包含了将要传送至实际的自动重装载寄存器的数值。有关 ARR 的更新和动作详细参考 289 页 16.3.1 节。

当自动重装载的值为空时，计数器不工作。

16.4.13 TIM2 和 TIM3 捕捉 / 比较寄存器 1 (TIM2_CCR1 和 TIM3_CCR1)

地址偏移 : 0x34

复位值 : 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR1[31:16] (TIM2 only)															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 CCR1[31:16]: 捕捉 / 比较 1 高 16 位值 (仅 TIM2).

位 15:0 CCR1[15:0]: 捕捉 / 比较 1 低 16 位值

若 CC1 通道配置为输出：

CCR1 包含了装入当前捕获 / 比较 1 寄存器的值（预装载值）。

如果在 TIMx_CCMR1 寄存器（OC1PE 位）中未选择预装载特性，写入的数值会被立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获 / 比较 1 寄存器中。

当前捕获 / 比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC1 端口上产生输出信号。

若 CC1 通道配置为输入：

CCR1 包含了由上一次输入捕获 1 事件 (IC1) 传输的计数器值。

16.4.14 TIM2 和 TIM3 捕捉 / 比较寄存器 2 (TIM2_CCR2 和 TIM3_CCR2)

地址偏移 : 0x38

复位值 : 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR2[31:16] (TIM2 only)															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 CCR2[31:16]: 捕捉 / 比较 2 高 16 位值 (仅 TIM2).

位 15:0 CCR2[15:0]: 捕捉 / 比较 2 低 16 位值

若 CC2 通道配置为输出：

CCR2 包含了装入当前捕获 / 比较 2 寄存器的值（预装载值）。

如果在 TIMx_CCMR2 寄存器（OC2PE 位）中未选择预装载特性，写入的数值会被立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获 / 比较 2 寄存器中。

当前捕获 / 比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC2 端口上产生输出信号。

若 CC2 通道配置为输入：

CCR2 包含了由上一次输入捕获 2 事件 (IC2) 传输的计数器值。

16.4.15 TIM2 和 TIM3 捕捉 / 比较寄存器 3 (TIM2_CCR3 和 TIM3_CCR3)

地址偏移 : 0x3C

复位值 : 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR3[31:16] (TIM2 only)															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 CCR3[31:16]: 捕捉 / 比较 3 高 16 位值 (仅 TIM2).

位 15:0 CCR3[15:0]: 捕捉 / 比较 3 低 16 位值

若 CC3 通道配置为输出：

CCR3 包含了装入当前捕获 / 比较 3 寄存器的值（预装载值）。

如果在 TIMx_CCMR3 寄存器（OC3PE 位）中未选择预装载特性，写入的数值会被立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获 / 比较 3 寄存器中。

当前捕获 / 比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC3 端口上产生输出信号。

若 CC3 通道配置为输入：

CCR3 包含了由上一次输入捕获 3 事件 (IC3) 传输的计数器值。

16.4.16 TIM2 和 TIM3 捕捉 / 比较寄存器 4 (TIM2_CCR4 和 TIM3_CCR4)

地址偏移 : 0x40

复位值 : 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR4[31:16] (TIM2 only)															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 CCR4[31:16]: 捕捉 / 比较 4 高 16 位值 (仅 TIM2)

位 15:0 CCR4[15:0]: 捕捉 / 比较 4 低 16 位值

若 CC4 通道配置为输出：

CCR4 包含了装入当前捕获 / 比较 4 寄存器的值（预装载值）。

如果在 TIMx_CCMR4 寄存器（OC4PE 位）中未选择预装载特性，写入的数值会被立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获 / 比较 4 寄存器中。

当前捕获 / 比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC4 端口上产生输出信号。

若 CC4 通道配置为输入：

CCR4 包含了由上一次输入捕获 4 事件 (IC4) 传输的计数器值。

16.4.17 TIM2 和 TIM3 DMA 控制寄存器 (TIM2_DCR 和 TIM3_DCR)

地址偏移 : 0x48

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

位 15:13 保留，始终读为‘0’

位 12:8 DBL[4:0]: DMA 连续传送长度 (DMA burst length)

这些位定义了 DMA 在连续模式下的传送长度 (当对 TIMx_DMAR 寄存器进行读或写时，定时器则 进行一次连续传送)，即：定义传输的字节数目：

00000：1 个字节

00001：2 个字节

00010：3 个字节

.....

.....

10001：18 个字节

位 7:5 保留，始终读为‘0’

位 4:0 DBA[4:0]: DMA 基地址 (DMA base address)

这些位定义了 DMA 传送的基地址 (当对 TIMx_DMAR 寄存器进行读或写时)，DBA 定义为从 TIMx_CR1 寄存器所在地址开始的偏移量 .

例如：

00000：TIMx_CR1，

00001：TIMx_CR2，

00010：TIMx_SMCR，

.....

实例 : 假设传输配置如下： DBL = 7 & DBA = TIMx_CR1. 此时向或从 TIMx_CR1 地址连续传送 7 个寄存器的内容。

16.4.18 TIM2 和 TIM3 DMA 完全传送地址寄存器 (TIM2_DMAR 和 TIM3_DMAR)

地址偏移 : 0x4C

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAB[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 DMAB[15:0]: DMA 连续传送寄存器 (DMA register for burst accesses)

对 TIMx_DMAR 寄存器的读或写会导致对以下地址所在寄存器的存取操作：

TIMx_CR1 地址 + （DBA + DMA 索引）×4，

其中：“TIMx_CR1 地址”是控制寄存器 1(TIMx_CR1) 所在的地址；

“DBA”是 TIMx_DCR 寄存器中定义的基地址；

“DMA 索引”是由 DMA 自动控制的偏移量，它取决于 TIMx_DCR 寄存器中定义的 DBL。

如何使用 DMA 并发操作的例子

本例中使用定时器 DMA 的并发功能，将 CCRx 寄存器 (x = 2, 3, 4) 的内容以半字方式进行 DMA 传输，更新到 CCRx 寄存器。

按如下步骤进行操作：

1. 配置相关的 DMA 通道 :
   - DMA 通道设备地址为 DMAR 寄存器地址
   - DMA 通道存储器地址为包含要通过 DMA 传送到 CCRx 寄存器的数据 RAM 缓冲区地址
   - 传送数据数量 = 3 (见下面的注).
   - 通告模式禁止 .
2. 配置 DCR 寄存器的 DBA 和 DBL 位 : DBL = 3 次传送 , DBA = 0xE.
3. 使能 TIMx 更新 DMA 请求 (设置 DIER 寄存器的 UDE 位).
4. 使能 TIMx
5. 使能 DMA 通道

注： 在本例中所有 CCRx 寄存器被一次性全部更新。如果需要更新 CCRx 寄存器两次，传送的数据数量应该是 6 ，而 RAM 缓冲区要包含 data1, data2, data3, data4, data5 和 data6。数据按如下过程被传送到 CCRx 寄存器：在第一个更新 DMA 请求时，data1 被传送到 CCR2, data2 被传送到 CCR3, data3 被传送到 CCR4，在第二个 DMA 更新中断请求时，data4 被传送到 CCR2, data5 被传送到 CCR3，data6 被传送到 CCR4

16.4.19 TIM2 和 TIM3 寄存器映射

TIM2 和 TIM3 寄存器映射如下表所示：

表 49. TIM2 和 TIM3 寄存器映射及复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**TIM2_CR1 and TIM3_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKD [1:0]		ARPE	CMS [1:0]		DIR	OPM	URS	UDIS	CEN
	Reset value																							0	0	0	0	0	0	0	0	0	0
0x04	**TIM2_CR2 and TIM3_CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1S	MMS[2:0]			CCDS	Res.	Res.	Res.
	Reset value																									0	0	0	0	0			
0x08	**TIM2_SMCR and TIM3_SMCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETP	ECE	ETPS [1:0]		ETF[3:0]				MSM	TS[2:0]			Res.	SMS[2:0]		
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0
0x0C	**TIM2_DIER and TIM3_DIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDE	Res.	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Res.	TIE	Res.	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	Reset value																		0		0	0	0	0	0		0		0	0	0	0	0
0x10	**TIM2_SR and TIM3_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC4OF	CC3OF	CC2OF	CC1OF	Res.	Res.	TIF	Res.	CC4IF	CC3IF	CC2IF	CC1IF	UIF
	Reset value																				0	0	0	0			0		0	0	0	0	0
0x14	**TIM2_EGR and TIM3_EGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TG	Res.	CC4G	CC3G	CC2G	CC1G	UG
	Reset value																										0		0	0	0	0	0
0x18	**TIM2_CCMR1 and TIM3_CCMR1** Output compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2CE	OC2M [2:0]			OC2PE	OC2FE	CC2S [1:0]		OC1CE	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIM2_CCMR1 and TIM3_CCMR1 Input capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC2F[3:0]				IC2 PSC [1:0]		CC2S [1:0]		IC1F[3:0]				IC1 PSC [1:0]		CC1S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	**TIM2_CCMR2 and TIM3_CCMR2** Output compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	O24CE	OC4M [2:0]			OC4PE	OC4FE	CC4S [1:0]		OC3CE	OC3M [2:0]			OC3PE	OC3FE	CC3S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIM2_CCMR2 and TIM3_CCMR2 Input capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC4F[3:0]				IC4 PSC [1:0]		CC4S [1:0]		IC3F[3:0]				IC3 PSC [1:0]		CC3S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	**TIM2_CCER and TIM3_CCER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC4NP	Res.	CC4P	CC4E	CC3NP	Res.	CC3P	CC3E	CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E
	Reset value																	0		0	0	0		0	0	0		0	0	0		0	0
0x24	**TIM2_CNT and TIM3_CNT**	CNT[31:16] (TIM2 only)																CNT[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	**TIM2_PSC and TIM3_PSC**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**TIM2_ARR and TIM3_ARR**	ARR[31:16] (TIM2 only)																ARR[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 49. TIM2 和 TIM3 寄存器映射及复位值（续）

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x30	**Reserved**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x34	**TIM2_CCR1 and TIM3_CCR1**	CCR1[31:16] (TIM2 only)																CCR1[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	**TIM2_CCR2 and TIM3_CCR2**	CCR2[31:16] (TIM2 only)																CCR2[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	**TIM2_CCR3 and TIM3_CCR3**	CCR3[31:16] (TIM2 only)																CCR3[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	**TIM2_CCR4 and TIM3_CCR4**	CCR4[31:16] (TIM2 only)																CCR4[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	**Reserved**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x48	**TIM2_DCR and TIM3_DCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
	Reset value																				0	0	0	0	0				0	0	0	0	0
0x4C	**TIM2_DMAR and TIM3_DMAR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAB[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

参见 35 页 2.2.2 节关于寄存器的基地址

17 通用定时器 (TIM14)

17.1 TIM14 简介

通用定时器 TIM14 由一个 16 位的自动装载计数器组成，它由一个可编程的预分频器驱动。

它适合多种用途，包含测量输入信号的脉冲宽度 (输入捕获)，或者产生输出波形 (输出比较和 PWM)。

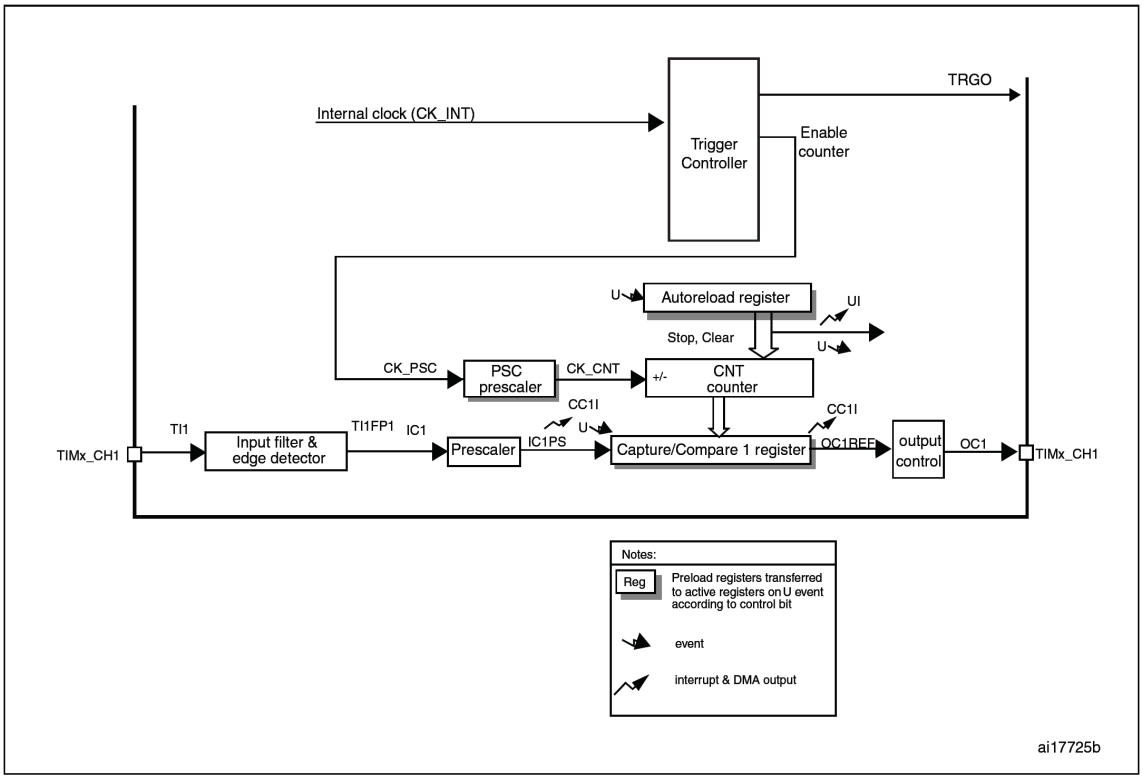
使用定时器预分频器和 RCC 时钟控制预分频器，可以实现脉冲宽度和波形周期从几个微秒到几个毫秒的调节。

通用定时器 TIM14 是完全独立的，不共享任何资源。它们可以同步操作，具体描述参看 16.3.15 节。

17.2 TIM14 主要特性

- 16 位自动装载计数器
- 16 位可编程 (可以实时修改) 预分频器，计数器时钟频率的分频系数为 1 ～ 65536 之间的任意数值
- 独立通道：
  - 输入捕获
  - 输出比较
  - PWM 生成 (边缘或中间对齐模式)
- 如下事件发生时产生中断 /DMA：
  - 更新：计数器溢出，计数器初始化 (通过软件)
  - 输入捕获
  - 输出比较

图 136. 通用定时器框图 (TIM14)



17.3 TIM14 功能描述

17.3.1 时基单元

可编程通用定时器的主要部分是一个 16 位计数器和与其相关的自动装载寄存器。这个计数器可以向上计数。此计数器时钟由预分频器分频得到。

计数器、自动装载寄存器和预分频器寄存器可以由软件读写，在计数器运行时仍可以读写。

时基单元包含：

- 计数器寄存器 (TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动装载寄存器 (TIMx_ARR)

自动装载寄存器是预先装载的，写或读自动重装载寄存器将访问预装载寄存器。根据在 TIMx_CR1 寄存器中的自动装载预装载使能位 (ARPE) 的设置，预装载寄存器的内容被立即或在每次的更新事件 UEV 时传送到影子寄存器。当计数器达到溢出条件并当 TIMx_CR1 寄存器中的 UDIS 位等于’0’时，产生更新事件。更新事件也可以由软件产生。随后会详细描述每一种配置下更新事件的产生。

计数器由预分频器的时钟输出 CK_CNT 驱动，仅当设置了计数器 TIMx_CR1 寄存器中的计数器使能位 (CEN) 时，CK_CNT 才有效。(有关计数器使能的细节，请参见控制器的从模式描述)。

注：真正的计数器使能信号 CNT_EN 是在 CEN 的一个时钟周期后被设置。

预分频器描述

预分频器可以将计数器的时钟频率按 1 到 65536 之间的任意值分频。它是基于一个 (在 TIMx_PSC 寄存器中的)16 位寄存器控制的 16 位计数器。这个控制寄存器带有缓冲器，它能够在工作时被改变。新的预分频器参数在下一次更新事件到来时被采用。

图 138 和图 139 给出了在预分频器运行时更改计数器参数的例子。

图 137. 当预分频器的参数从 1 变到 2 时，计数器的时序图

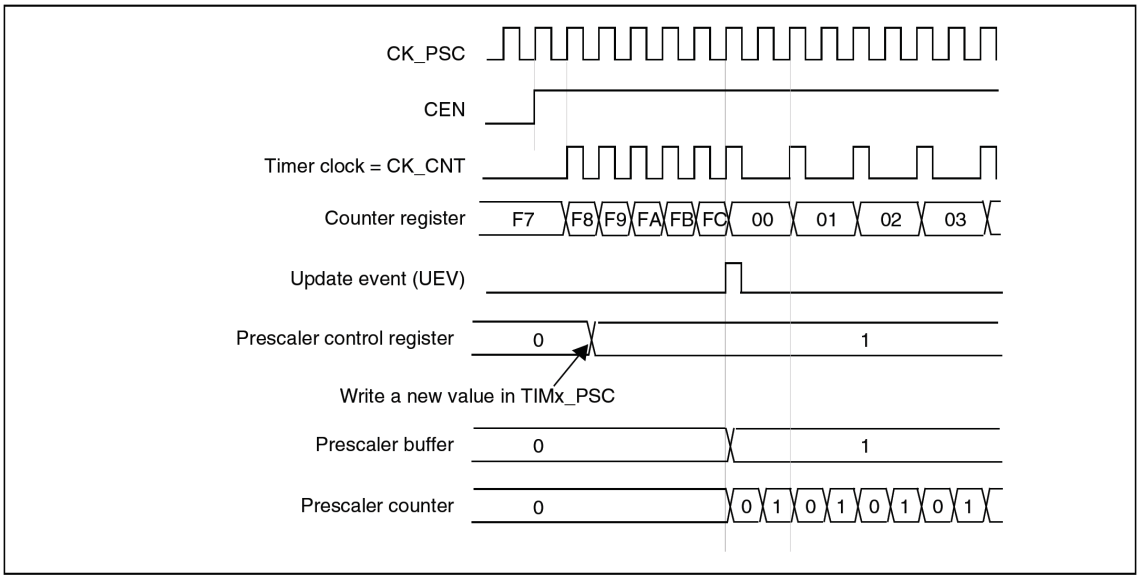
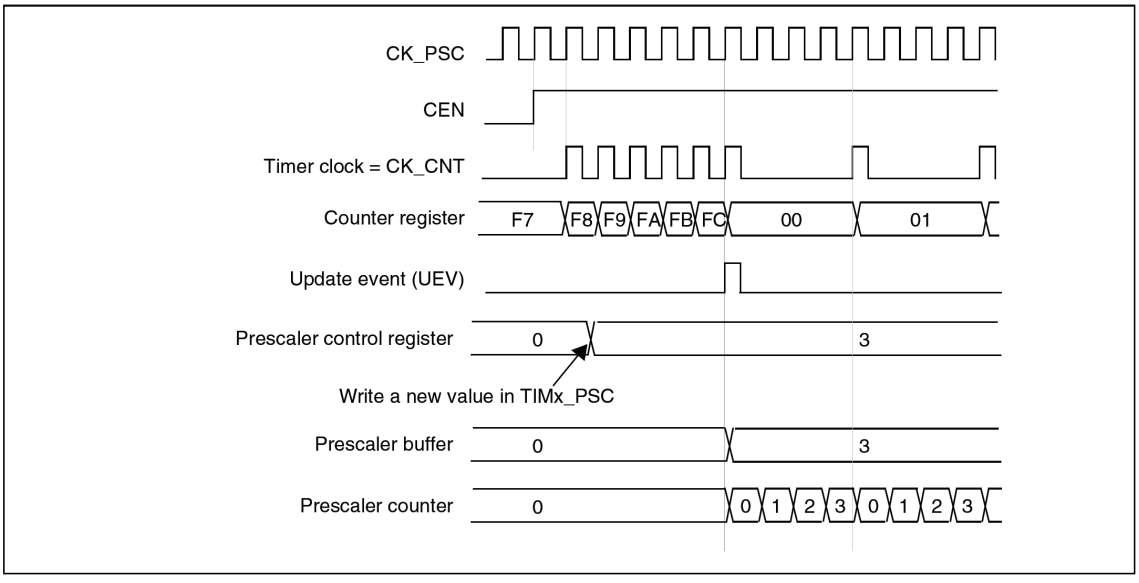


图 138. 当预分频器的参数从 1 变到 4 时，计数器的时序图



17.3.2 计数器模式

向上计数模式

在向上计数模式中，计数器从 0 计数到自动加载值 (TIMx_ARR 计数器的内容)，然后重新从 0 开始计数并且产生一个计数器溢出事件。

设置 TIMx_EGR 寄存器中 UG 位也同样可以产生一个更新事件。

设置 TIMx_CR1 寄存器中的 UDIS 位，可以禁止更新事件；这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。在 UDIS 位被清’0’之前，将不产生更新事件。但是在应该产生更新事件时，计数器仍会被清’0’，同时预分频器的计数也被请 0(但预分频系数不变)。此外，如果设置了 TIMx_CR1 寄存器中的 URS 位 (选择更新请求)，设置 UG 位将产生一个更新事件 UEV，但硬件不设置 UIF 标志 (即不产生中断)；这是为了避免在捕获模式下清除计数器时，同时产生更新和捕获中断。

当发生一个更新事件时，所有的寄存器都被更新，硬件同时 (依据 URS 位) 设置更新标志位 (TIMx_SR 寄存器中的 UIF 位)。

- 自动装载影子寄存器被重新置入预装载寄存器的值 (TIMx_ARR)。
- 预分频器的缓冲区被置入预装载寄存器的值 (TIMx_PSC 寄存器的内容)。

下图给出一些例子，当 TIMx_ARR=0x36 时计数器在不同时钟频率下的动作。

图 139. 计数器时序图，内部时钟分频因子为 1

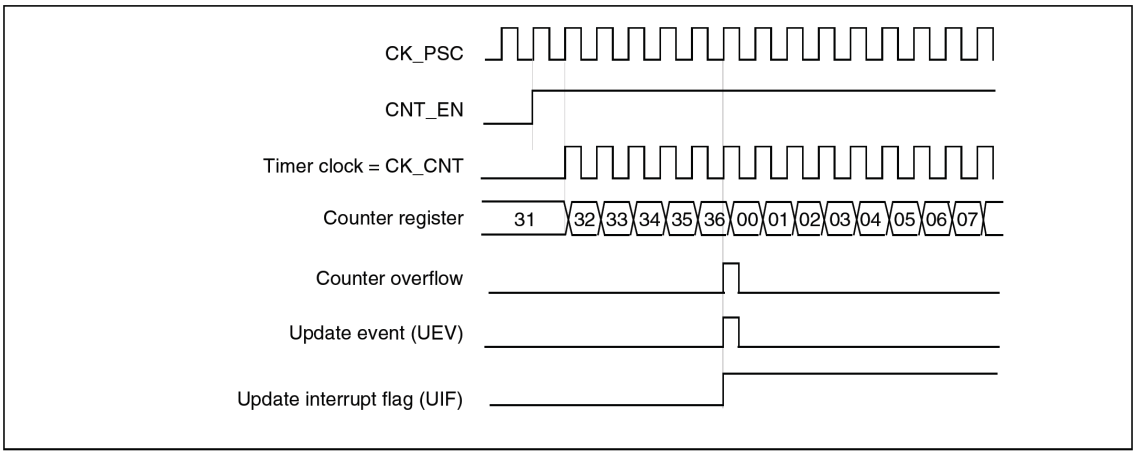


图 140. 计数器时序图，内部时钟分频因子为 2

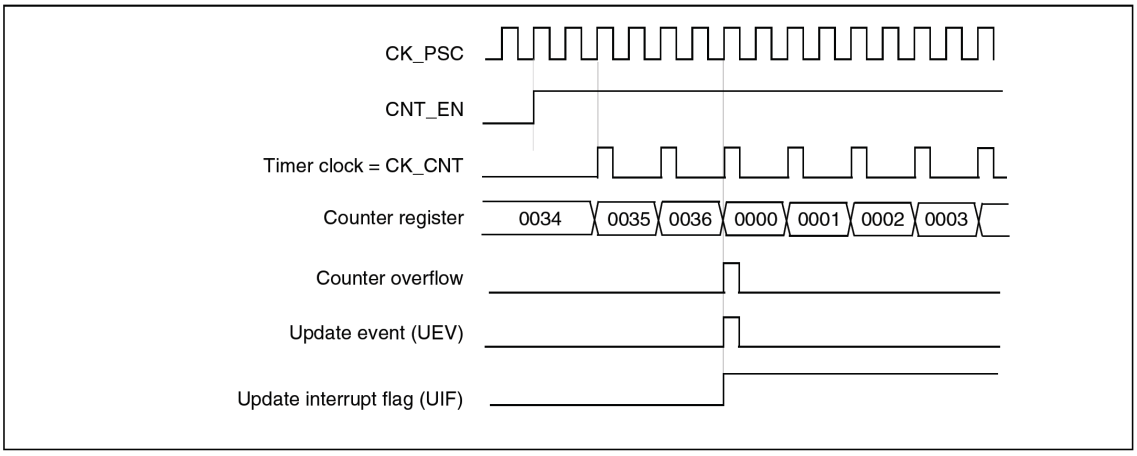


图 141. 计数器时序图，内部时钟分频因子为 4

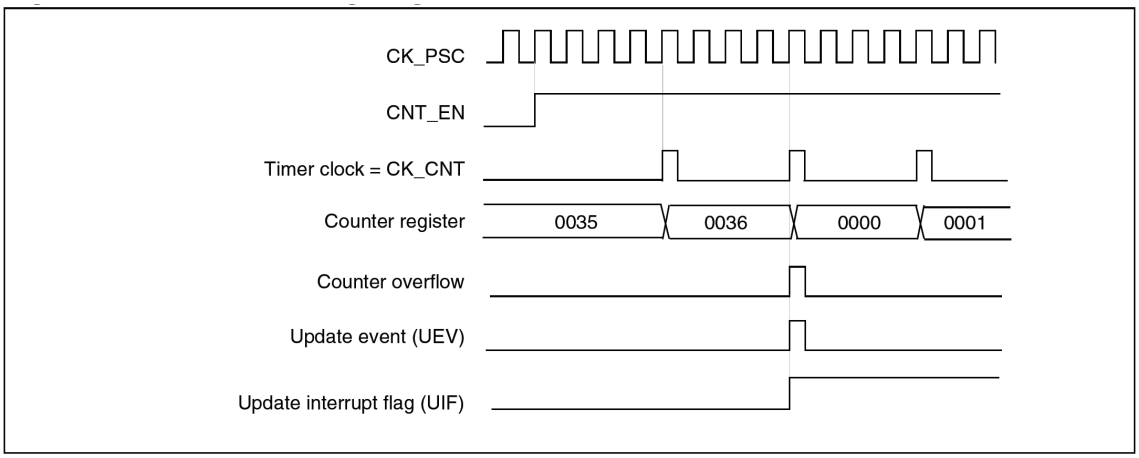


图 142. 计数器时序图，内部时钟分频因子为 N

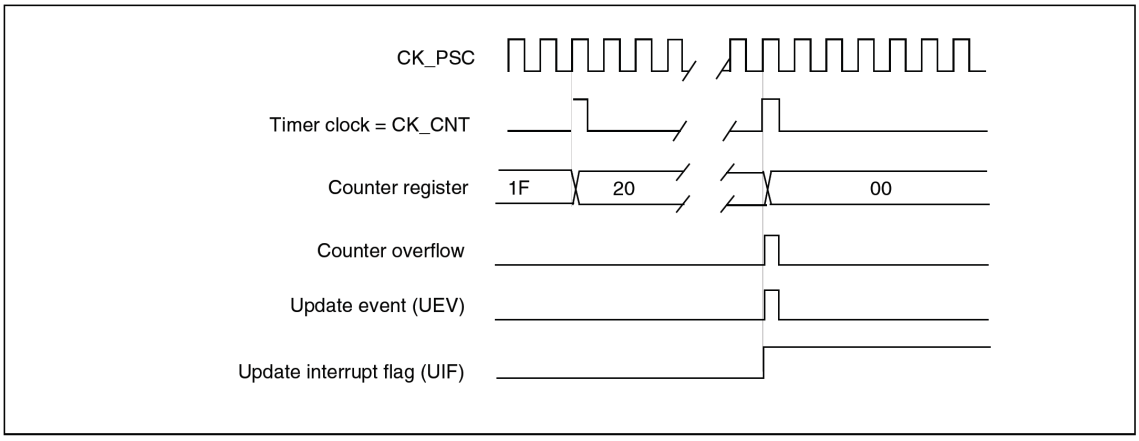


图 143. 计数器时序图，当 ARPE=0 时的更新事件 (TIMx_ARR 没有预装入)

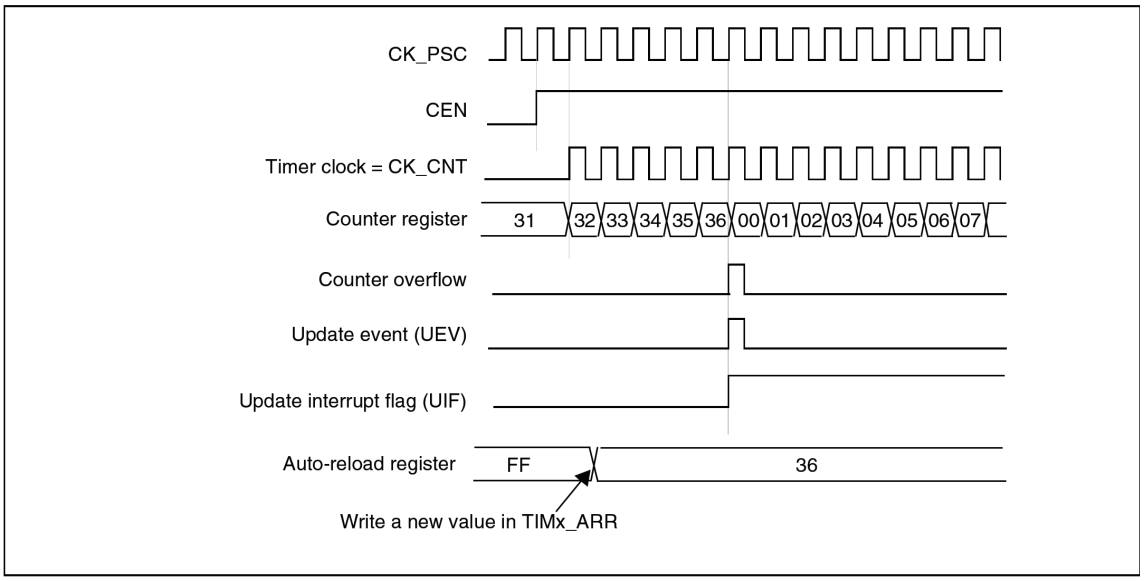
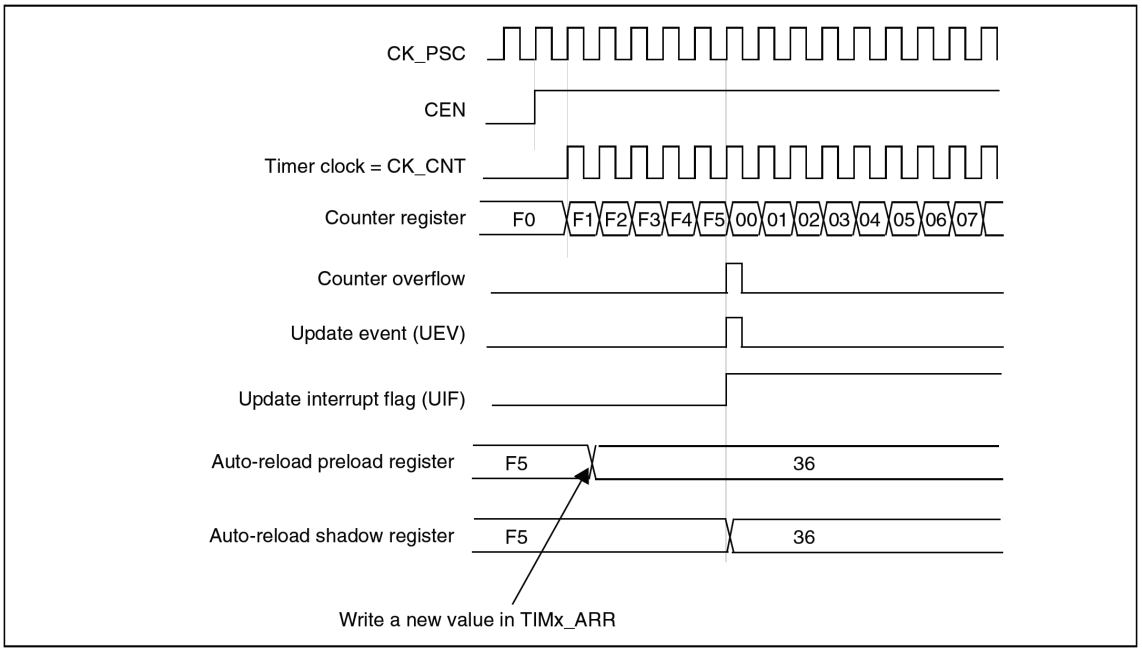


图 144. 计数器时序图，当 ARPE=1 时的更新事件 (TIMx_ARR 预装载)



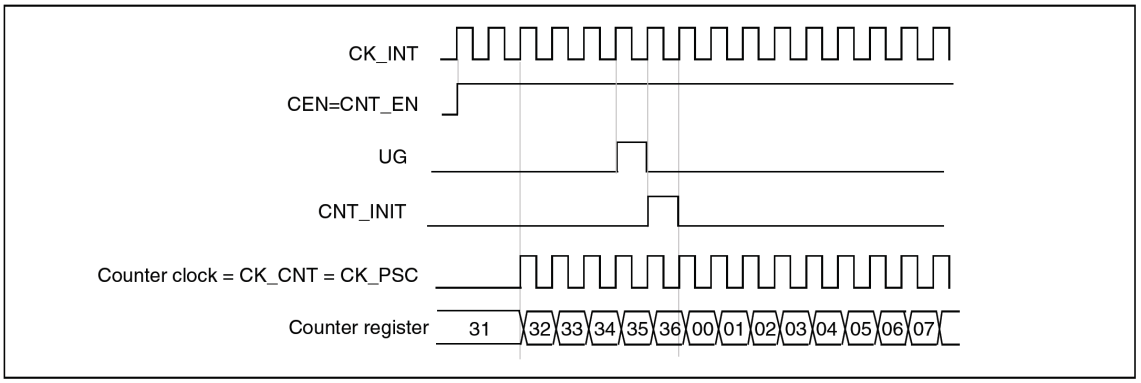
17.3.3 时钟源

计数器时钟由内部时钟 (CK_INT) 提供。

CEN(TIMx_CR1 寄存器) 和 UG 位 (TIMx_EGR 寄存器) 是事实上的控制位，并且只能被软件修改 (除了 UG 位仍被自动清除)。只要 CEN 位被写成’1’，预分频器的时钟就由内部时钟 CK_INT 提供。

图 145 显示了控制电路和向上计数器在一般模式下，不带预分频器时的操作。

图 145. 一般模式下的控制电路，内部时钟分频因子为 1



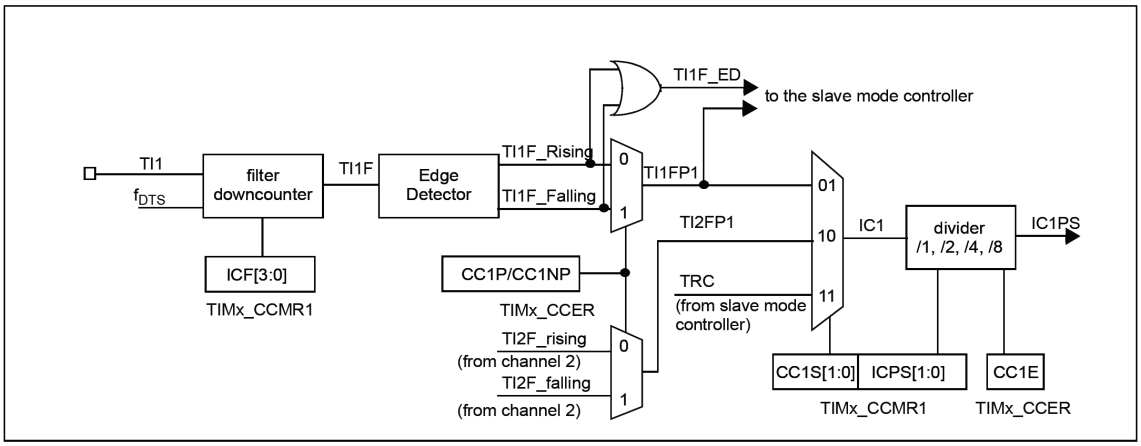
17.3.4 捕获 / 比较通道

每一个捕获 / 比较通道都是围绕着一个捕获 / 比较寄存器 (包含影子寄存器)，包括捕获的输入部分 (数字滤波、多路复用和预分频器)，和输出部分 (比较器和输出控制)。

图 146 至图 148 是一个捕获 / 比较通道概览。

输入部分对相应的 TIx 输入信号采样，并产生一个滤波后的信号 TIxF。然后，一个带极性选择的边缘检测器产生一个信号 (TIxFPx)，它可以作为从模式控制器的输入触发或者作为捕获控制。该信号通过预分频进入捕获寄存器 (ICxPS)。

图 146. 捕获 / 比较通道 (如：通道 1 输入部分)



输出部分产生一个中间波形 OCxRef(高有效) 作为基准，链的末端决定最终输出信号的极性。

图 147. 捕获 / 比较通道 1 的主电路

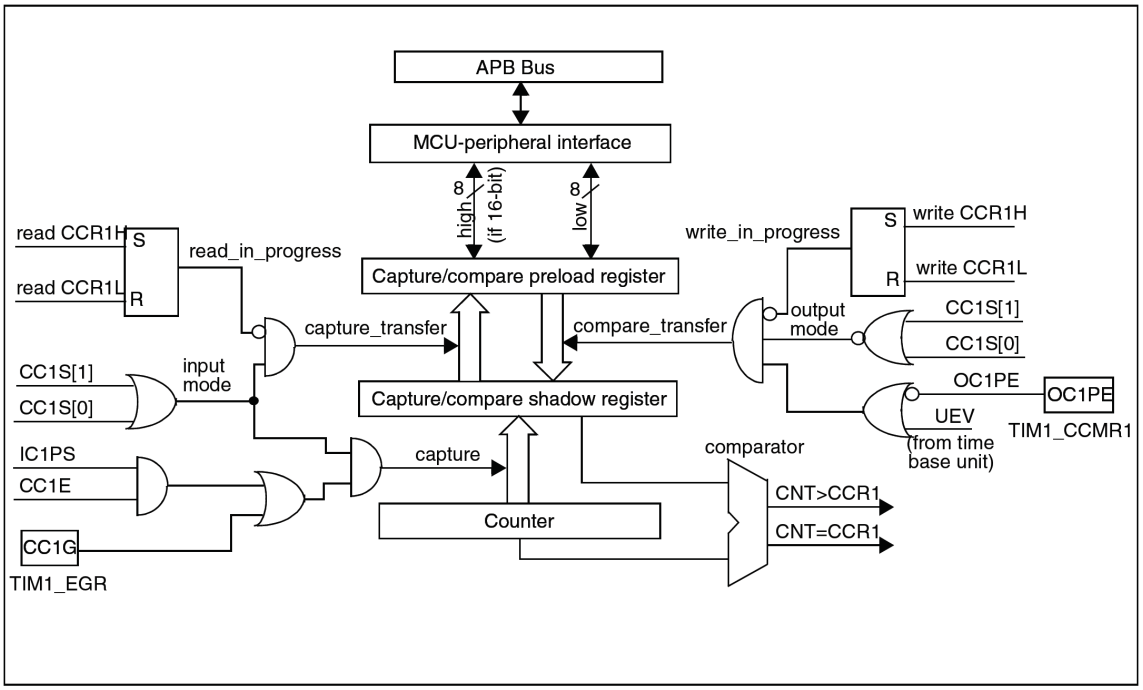
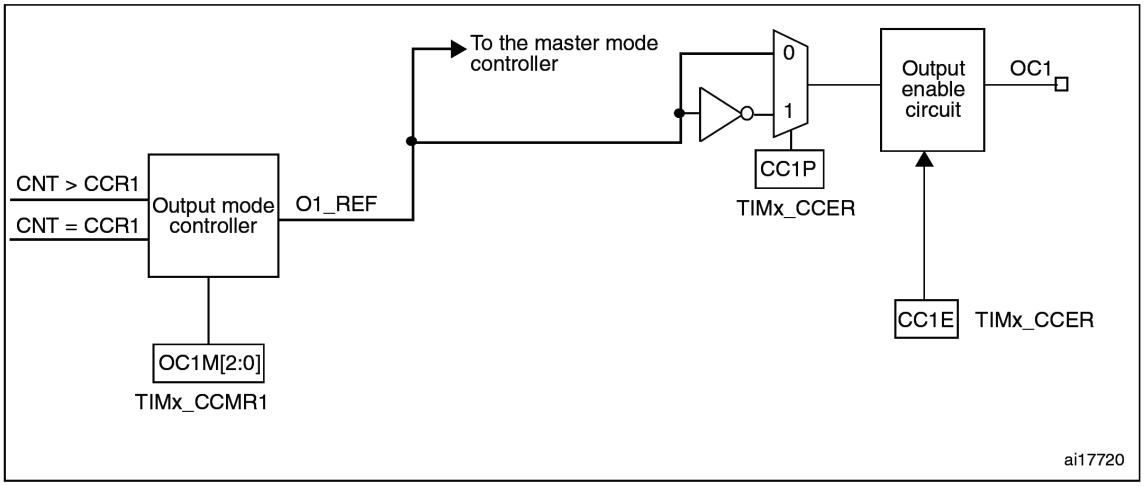


图 148. 捕获 / 比较通道的输出部分 (通道 1)



捕获 / 比较模块由一个预装载寄存器和一个影子寄存器组成。读写过程仅操作预装载寄存器。

在捕获模式下，捕获发生在影子寄存器上，然后再复制到预装载寄存器中。

在比较模式下，预装载寄存器的内容被复制到影子寄存器中，然后影子寄存器的内容和计数器进行比较。

17.3.5 输入捕捉模式

在输入捕获模式下，当检测到 ICx 信号上相应的边沿后，计数器的当前值被锁存到捕获 / 比较寄存器 (TIMx_CCRx) 中。当捕获事件发生时，相应的 CCxIF 标志 (TIMx_SR 寄存器) 被置’1’，如果使能了中断或者 DMA 操作，则将产生中断或者 DMA 操作。如果捕获事件发生时 CCxIF 标志已经为高，那么重复捕获标志 CCxOF(TIMx_SR 寄存器) 被置’1’。写 CCxIF=0 可清除 CCxIF，或读取存储在 TIMx_CCRx 寄存器中的捕获数据也可清除 CCxIF。写 CCxOF=0 可清除 CCxOF。

以下例子说明如何在 TI1 输入的上升沿时捕获计数器的值到 TIMx_CCR1 寄存器中，步骤如下：

1. 选择有效输入端：TIMx_CCR1 必须连接到 TI1 输入，所以写入 TIMx_CCR1 寄存器中的 CC1S=01，只要 CC1S 不为’00’，通道被配置为输入，并且 TM1_CCR1 寄存器变为只读。
2. 根据输入信号的特点，配置输入滤波器为所需的带宽 (即输入为 TIx 时，输入滤波器控制位是 TIMx_CCMRx 寄存器中的 ICxF 位)。假设输入信号在最多 5 个内部时钟周期的时间内抖动， 我们须配置滤波器的带宽长于 5 个时钟周期。因此我们可以 (以 fDTS 频率) 连续采样 8 次，以确 认在 TI1 上一次真实的边沿变换，即在 TIMx_CCMR1 寄存器中写入 IC1F=0011。
3. 选择 TI1 通道的有效转换边沿，在 TIMx_CCER 寄存器中写入 CC1P=0 和 CC1NP=0 (上升沿)。
4. 配置输入预分频器。在本例中，我们希望捕获发生在每一个有效的电平转换时刻，因此预分频器被禁止 (写 TIMx_CCMR1 寄存器的 IC1PS=00)。
5. 设置 TIMx_CCER 寄存器的 CC1E=1，允许捕获计数器的值到捕获寄存器中。
6. 如果需要，通过设置 TIMx_DIER 寄存器中的 CC1IE 位允许相关中断请求。

当发生一个输入捕获时：

- 产生有效的电平转换时，计数器的值被传送到 TIMx_CCR1 寄存器。
- CC1IF 标志被设置 (中断标志)。当发生至少 2 个连续的捕获时，而 CC1IF 未曾被清除，CC1OF 也被置’1’。
- 如设置了 CC1IE 位，则会产生一个中断。

为了处理捕获溢出，建议在读出捕获溢出标志之前读取数据，这是为了避免丢失在读出捕获溢出标志之后和读取数据之前可能产生的捕获溢出信息。

注： 设置 TIMx_EGR 寄存器中相应的 CCxG 位，可以通过软件产生输入捕获中断请求。

17.3.6 强制输出模式

在输出模式 (TIMx_CCMRx 寄存器中 CCxS=00) 下，输出比较信号 (OCxREF 和相应的 OCx) 能够直接由软件强置为有效或无效状态，而不依赖于输出比较寄存器和计数器间的比较结果。

置 TIMx_CCMRx 寄存器中相应的 OCxM=101，即可强置输出比较信号 (OCxREF/OCx) 为有效状态。这样 OCxREF 被强置为高电平 (OCxREF 始终为高电平有效)，同时 OCx 得到 CCxP 极性位相反的值。

例如：CCxP=0(OCx 高电平有效)，则 OCx 被强置为高电平。
置 TIMx_CCMRx 寄存器中的 OCxM=100，可强置 OCxREF 信号为低。
该模式下，在 TIMx_CCRx 影子寄存器和计数器之间的比较仍然在进行，相应的标志也会被修改。因此仍然会产生相应的中断请求。这将会在下面的输出比较模式一节中介绍。

17.3.7 输出比较模式

此功能是用来控制一个输出波形，或者指示一段给定的间已经到时。
当计数器与捕获 / 比较寄存器的内容相同时，输出比较功能做如下操作：

1. 将输出比较模式 (TIMx_CCMRx 寄存器中的 OCxM 位) 和输出极性 (TIMx_CCER 寄存器中的 CCxP 位) 定义的值输出到对应的引脚上。在比较匹配时，输出引脚可以保持它的电平 (OCxM=000)、被设置成有效电平 (OCxM=001)、被设置成无效电平 (OCxM=010) 或进行翻 转 (OCxM=011)。
2. 设置中断状态寄存器中的标志位 (TIMx_SR 寄存器中的 CCxIF 位)。
3. 若设置了相应的中断屏蔽 (TIMx_DIER 寄存器中的 CCxIE 位)，则产生一个中断。

TIMx_CCMRx 中的 OCxPE 位选择 TIMx_CCRx 寄存器是否需要使用预装载寄存器。

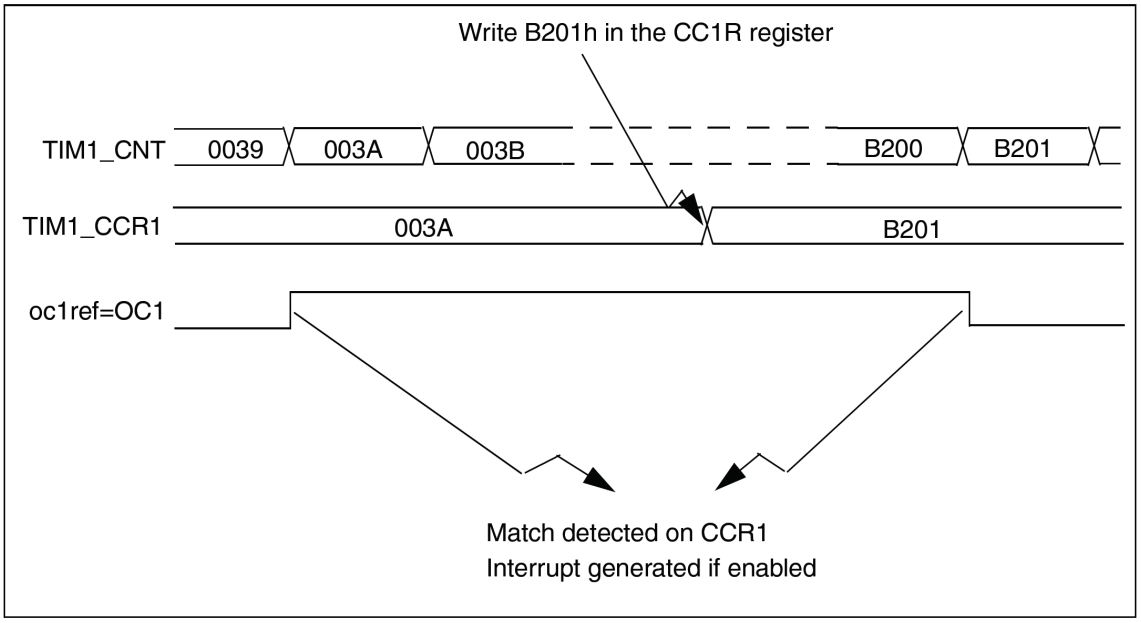
在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出没有影响。 同步的精度可以达到计数器的一个计数周期。输出比较模式 (在单脉冲模式下) 也能用来输出一个单脉冲。

输出比较模式的配置步骤：

1. 选择计数器时钟 (内部，外部，预分频器)
2. 将相应的数据写入 TIMx_ARR 和 TIMx_CCRx 寄存器中
3. 如果要产生一个中断请求请求，设置 CCxIE 位。
4. 选择输出模式：
   - 写 OCxM = ‘011’， 当计数器 CNT 与 CCRx 匹配时翻转 OCx 的输出引脚
   - 写 OCxPE = ‘0’ ，禁止预装载
   - 写 CCxP = ‘0’， 选择高电平有效
   - 写 CCxE = ‘1’， 允许输出
5. 设置 TIMx_CR1 寄存器的 CEN 位启动计数器

TIMx_CCRx 寄存器能够在任何时候通过软件进行更新以控制输出波形，条件是未使用预装载寄存器 (OCxPE=’0’，否则 TIMx_CCRx 影子寄存器只能在发生下一次更新事件时被更新)。图 149 给出了一个例子。

图 149. 输出比较模式，翻转 OC1



17.3.8 PWM 模式

脉冲宽度调制模式可以产生一个由 TIMx_ARR 寄存器确定频率、由 TIMx_CCRx 寄存器确定占空比的信号。

在 TIMx_CCMRx 寄存器中的 OCxM 位写入’110’(PWM 模式 1) 或’111’(PWM 模式 2)，能够独立地设置每个 OCx 输出通道产生一路 PWM。必须设置 TIMx_CCMRx 寄存器 OCxPE 位以使能相应的预装载寄存器，最后还要设置 TIMx_CR1 寄存器的 ARPE 位，(在向上计数或中心对称模式中) 使能自动重装载的预装载寄存器。

仅当发生一个更新事件的时候，预装载寄存器才能被传送到影子寄存器，因此在计数器开始计数之前，必须通过设置 TIMx_EGR 寄存器中的 UG 位来初始化所有的寄存器。

OCx 的极性可以通过软件在 TIMx_CCER 寄存器中的 CCxP 位设置，它可以设置为高电平有效或低电平有效。TIMx_CCER 寄存器中的 CCxE 位控制 OCx 输出使能。详见 TIMx_CCERx 寄存器的描述。

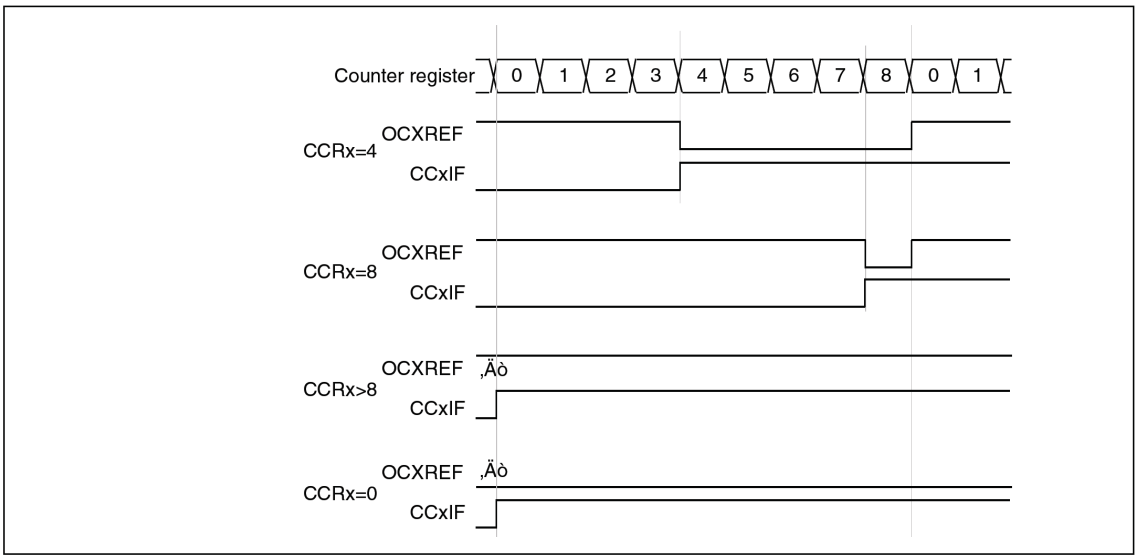
在 PWM 模式 (模式 1 或模式 2) 下，TIMx_CNT 和 TIMx_CCRx 始终在进行比较是否符合 TIMx_CNT ≤ TIMx_CCRx 。

因为本计数器是向上计数只能产生边沿对齐的 PWM 模式。

PWM 边沿对齐模式

下面是一个 PWM 模式 1 的例子。当 TIMx_CNT<TIMx_CCRx 时 PWM 信号参考 OCxREF 为高，否则为低。如果 TIMx_CCRx 中的比较值大于自动重装载值 (TIMx_ARR)，则 OCxREF 保持为’1’。 如果比较值为 0，则 OCxREF 保持为’0’。 图 150 为 TIMx_ARR=8 时边沿对齐的 PWM 波形实例。

图 150. 边沿对齐的 PWM 波形 (ARR=8)



17.3.9 调试模式

当微控制器进入调试模式 (Cortex-M3 核心停止)，根据 DBGMCU 模块中 DBG_TIMx_STOP 的设置，TIMx 计数器或者继续正常操作，或者停止。

17.4 TIM14 寄存器

17.4.1 TIM14 控制寄存器 1 (TIM14_CR1)

偏移地址 : 0x00

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CKD[1:0]		ARPE	Res.	Res.	Res.	Res.	URS	UDIS	CEN
						rw	rw	rw					rw	rw	rw

位 15:10 保留，始终保持复位值 .

位 9:8 CKD: 时钟分频因子 (Clock division)

定义在定时器时钟 (CK_INT) 频率与数字滤波器 (ETR，TIx) 使用的采样频率之间的分频比例。

00：tDTS = tCK_INT

01：tDTS = 2 x tCK_INT

10：tDTS = 4 x tCK_INT

11：保留

位 7 ARPE: 自动重装载预装载允许位 (Auto-reload preload enable)

0：TIMx_ARR 寄存器没有缓冲；

1：TIMx_ARR 寄存器缓冲器有效。

位 6:3 保留，始终保持复位值 .

位 2 URS: 更新请求源 (Update request source)

通过软件置位或清除来选择 UEV 事件的源

0：如果使能了 UEV 则下述任一事件产生 UEV：

- 计数器溢出
- 设置 UG 位

1：如果使能了 UEV 只有计数器溢出产生 UEV 事件。

位 1 UDIS: 禁止更新 (Update disable)

软件通过该位允许 / 禁止 UEV 事件的产生

0：允许 UEV。更新 (UEV) 事件由下述任一事件产生：

- 计数器溢出
- 设置 UG 位

1：禁止 UEV。不产生更新事件，影子寄存器 (ARR、PSC、CCRx) 保持它们的值。如果设置了 UG 位则计数器和预分频器被重新初始化。

位 0 CEN: 使能计数器（Counter enable）

0：禁止计数器；

1：使能计数器。

17.4.2 TIM14 中断使能寄存器 (TIM14_DIER)

偏移地址 : 0x0C

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1IE	UIE
														rw	rw

位 15:2 保留，始终保持复位值 .

位 1 CC1IE: 捕捉 / 比较 1 中断使能

0: CC1 中断禁止

1: CC1 中断允许

位 0 UIE: 更新中断使能

0: 更新中断禁止

1: 更新中断允许

17.4.3 TIM14 状态寄存器 (TIM14_SR)

偏移地址 : 0x10

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CC1OF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1IF	UIF
						rc_w0								rc_w0	rc_w0

位 15:10 保留，始终保持复位值 .

位 9 CC1OF: 捕捉 / 比较 1 重复捕捉标志 (Capture/Compare 1 overcapture flag)

仅当相应的通道被配置为输入捕获时，该标记可由硬件置’1’。写’0’可清除该位。

0：无重复捕获产生；

1：当计数器的值被捕获到 TIMx_CCR1 寄存器时，CC1IF 的状态已经为’1’ 。

位 8:2 保留，始终保持复位值 .

位 1 CC1IF: 捕捉 / 比较 1 中断标志 (Capture/Compare 1 interrupt flag)

如果通道 CC1 配置为输出模式：

当计数器值与比较值匹配时该位由硬件置’1’ ，由软件清’0’ 。

0：无匹配发生；

1：TIMx_CNT 的值与 TIMx_CCR1 的值匹配。

当 TIMx_CCR1 的内容大于 TIMx_ARR 时，在计数器溢出时 CC1IF 位变高

如果通道 CC1 配置为输入模式：

当捕获事件发生时该位由硬件置’1’，它由软件清’0’或通过读 TIMx_CCR1 清’0’。

0：无输入捕获产生；

1：计数器值已被捕获 (拷贝) 至 TIMx_CCR1(在 IC1 上检测到与所选极性相同的边沿)。

位 0 UIF: 更新中断标记 (Update interrupt flag)
当产生更新事件时该位由硬件置’1’。它由软件清’0’。
0：无更新事件产生；
1：更新中断等待响应。当寄存器被更新时该位由硬件置’1’：
— 若 TIMx_CR1 寄存器的 UDIS=0，当计数溢出时；
— 若 TIMx_CR1 寄存器的 UDIS=0、URS=0，通过软件写 TIMx_EGR 寄存器的 UG 位重初始化定时器时

17.4.4 TIM14 事件产生寄存器 (TIM14_EGR)

偏移地址 : 0x14
复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1G	UG
														w	w

位 15:2 保留，始终保持复位值 .
位 1 CC1G: 捕捉 / 比较 1 产生
该位由软件置’1’，用于产生一个捕获 / 比较事件，由硬件自动清’0’。
0：无动作；
1：在通道 CC1 上产生一个捕获 / 比较事件：
若通道 CC1 配置为输出：
设置 CC1IF=1，若开启对应的中断则产生相应的中断
若通道 CC1 配置为输入：
当前的计数器值捕获至 TIMx_CCR1 寄存器；设置 CC1IF=1，若开启对应的中断则产生相应的中断。若 CC1IF 已经为 1，则设置 CC1OF=1。
位 0 UG: 产生更新事件 (Update generation)
该位由软件置’1’，由硬件自动清’0’。
0：无动作；
1：重新初始化计数器，并产生一个更新事件。注意预分频器的计数器也被清’0’(但是预分频系数不变)。计数器被清’0’。

17.4.5 TIM14 捕捉 / 比较模式寄存器 1 (TIM14_CCMR1)

偏移地址 : 0x18

复位值 : 0x0000

通道可用于输入 (捕获模式) 或输出 (比较模式)，通道的方向由相应的 CCxS 定义。该寄存器其它位的作用在输入和输出模式下不同。OCxx 描述了通道在输出模式下的功能，ICxx 描述了通道在输入模式下的功能。因此必须注意，同一个位在输出模式和输入模式下的功能是不同的。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC1F[3:0]				IC1PSC[1:0]			
								rw	rw	rw	rw	rw	rw	rw	rw

输出比较模式

位 15:7 保留

位 6:4 OC1M: 输出比较 1 模式 (Output compare 1 mode)

该 3 位定义了输出参考信号 OC1REF 的动作，而 OC1REF 决定了 OC1 的值。OC1REF 是高电平有效，而 OC1 的有效电平取决于 CC1P 位。

000：冻结。输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 间的比较对 OC1REF 不起作用；

001 ：匹配时设置通道 1 为有效电平。 当计数器 TIMx_CNT 的值与捕获 / 比较寄存器 1 (TIMx_CCR1) 相同时，强制 OC1REF 为高。

010 ：匹配时设置通道 1 为无效电平。 当计数器 TIMx_CNT 的值与捕获 / 比较寄存器 1 (TIMx_CCR1) 相同时，强制 OC1REF 为低。

011：翻转。当 TIMx_CCR1=TIMx_CNT 时，翻转 OC1REF 的电平。

100：强制为无效电平。强制 OC1REF 为低。

101：强制为有效电平。强制 OC1REF 为高。

110：PWM 模式 1，一旦 TIMx_CNT<TIMx_CCR1 时通道 1 为有效电平，否则为无效电平；

111：PWM 模式 2，一旦 TIMx_CNT<TIMx_CCR1 时通道 1 为无效电平，否则为有效电平；

注：在 PWM 模式 1 或 PWM 模式 2 中，只有当比较结果改变了或在输出比较模式中从冻结模式切换到 PWM 模式时，OC1REF 电平才改变。

位 3 OC1PE: 输出比较 1 预装载使能 (Output compare 1 preload enable)

0：禁止 TIMx_CCR1 寄存器的预装载功能，可随时写入 TIMx_CCR1 寄存器，并且新写入的数值立即起作用。

1：开启 TIMx_CCR1 寄存器的预装载功能，读写操作仅对预装载寄存器操作，TIMx_CCR1 的预装载值在更新事件到来时被传送至当前寄存器中。

注：仅在单脉冲模式下 (TIMx_CR1 寄存器的 OPM=’1’)，可以在未确认预装载寄存器情况下使用 PWM 模式，否则其动作不确定。

位 2 OC1FE: 输出比较 1 快速使能 (Output compare 1 fast enable)
该位用于加快 CC 输出对触发器输入事件的响应。
0：根据计数器与 CCR1 的值，CC1 正常操作，即使触发器是打开的。当触发器的输入出现一个有效沿时，激活 CC1 输出的最小延时为 5 个时钟周期。
1：输入到触发器的有效沿的作用就象发生了一次比较匹配。因此，OC 被设置为比较电平而与比较结果无关。采样触发器的有效沿和 CC1 输出间的延时被缩短为 3 个时钟周期。该位只在通道被配置成 PWM1 或 PWM2 模式时起作用。

位 1:0 CC1S: 捕捉 / 比较 1 选择 (Capture/Compare 1 selection)
这 2 位定义通道的方向 (输入 / 输出)，及输入脚的选择：
00：CC1 通道被配置为输出；
01：CC1 通道被配置为输入，IC1 映射在 TI1 上；
10：保留
11：保留
注：CC1S 仅在通道关闭时 (TIMx_CCER 寄存器的 CC1E=’0’) 才是可写的。

输入捕捉模式

位 15:8 保留

位 7:4 IC1F: 输入捕获 1 滤波器 (Input capture 1 filter)
这几位定义了 TI1 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成，它记录到 N 个事件后会产生一个输出的跳变：

0000：无滤波器，以 f_{DTS} 采样
0001：采样频率 $f_{SAMPLING}=f_{CK\_INT}$，N=2
0010：采样频率 $f_{SAMPLING}=f_{CK\_INT}$，N=4
0011：采样频率 $f_{SAMPLING}=f_{CK\_INT}$，N=8
0100：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=6
0101：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=8
0110：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=6
0111：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=8
1000：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=6
1001：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=8
1010：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=5
1011：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=6
1100：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=8
1101：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=5
1110：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=6
1111：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=8

注：在现在的芯片版本中，当 ICxF[3:0]=1、2 或 3 时，公式中的 f_{DTS} 由 CK_INT 替代。

位 3:2 IC1PSC: 输入捕捉 1 预分频（Input capture 1 prescaler）
这 2 位定义了 CC1 输入 (IC1) 的预分频系数。 一旦 CC1E=’0’(TIMx_CCER 寄存器中)，则预分频器复位。
00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获；
01：每 2 个事件触发一次捕获；
10：每 4 个事件触发一次捕获；
11：每 8 个事件触发一次捕获。

位 1:0 CC1S: 捕捉 / 比较 1 选择（Capture/Compare 1 selection）
这 2 位定义通道的方向 (输入 / 输出)，及输入脚的选择：
00：CC1 通道被配置为输出；
01：CC1 通道被配置为输入，IC1 映射在 TI1 上；
10：
11：
注：CC1S 仅在通道关闭时 (TIMx_CCER 寄存器的 CC1E=’0’) 才是可写的。

17.4.6 TIM14 捕捉 / 比较使能寄存器 (TIM14_CCER)

偏移地址 : 0x20

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1NP	Res.	CC1P	CC1E
												rw		rw	rw

位 15:4 保留，始终保持复位值 .

位 3 CC1NP: 捕捉 / 比较 1 输出极性 (Capture/Compare 1 complementary output Polarity)
通道 CC1 配置为输出时，CC1NP 必须操持为清除，CC1NP ＝ 0;
通道 CC1 配置为输入时，CC1NP 与 CC1P 联合控制 TI1FP1 的极性
参考 CC1P 描述 .

位 2 保留，始终保持复位值 .

位 1 CC1P: 捕捉 / 比较 1 输出极性 .(Capture/Compare 1 output Polarity)
通道 CC1 配置为输出 :
0: OC1 高有效
1: OC1 低有效
通道 CC1 配置为输入 :
CC1P/CC1NP 用于选择作为触发或捕获的信号 TI1FP1 和 TI2FP1 的极性
该位 IC1 还是 IC1 的反相信号。
00：不反相 / 上升沿：捕获发生在 TIxFP1 的上升沿（捕捉模式），TIxFP1 不反相 ;
00：反相 / 下降沿：捕获发生在 TIxFP1 的下降沿（捕捉模式），TIxFP1 反相 ;
10：保留，不使用此配置
11：不反相 / 上升和下降沿：捕获发生在 TIxFP1 的上升沿和下降沿（捕捉模式），TIxFP1 不反相。

位 0 CC1E: 捕捉 / 比较 1 输出使能 (Capture/Compare 1 output enable)
CC1 通道配置为输出：
0： 关闭－ OC1 禁止输出。
1： 开启－ OC1 信号输出到对应的输出引脚。
CC1 通道配置为输入：
该位决定了计数器的值是否能捕获入 TIMx_CCR1 寄存器。
0：捕获禁止；
1：捕获使能。

表 50. 标准 OCx 通道的输出控制位

CCxE 位	OCx 输出状态
0	禁止输出 (OCx=0，OCx_EN=0)
1	OCx = OCxREF + 极性，OCx_EN=1

注： 连接到标准 OCx 通道的外部 I/O 引脚状态，取决于 OCx 通道状态和 GPIO 寄存器。

17.4.7 TIM14 计数器 (TIM14_CNT)

偏移地址 : 0x24

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 CNT[15:0]: 计数器的值

17.4.8 TIM14 预分频器 (TIM14_PSC)

偏移地址 : 0x28

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 PSC[15:0]: 预分频值

计数器的时钟频率 CK_CNT 等于 fCK_PSC/(PSC[15:0]+1)。 PSC 包含了当更新事件产生时装入当前预分频器寄存器的值。

17.4.9 TIM14 自动重装寄存器 (TIM14_ARR)

偏移地址 : 0x2C

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 ARR[15:0]: 自动重装寄存器的值

ARR 包含了将要传送至实际的自动重装载寄存器的数值。有关 ARR 的更新和动作详细参考 350 页 17.3.1 节。

当自动重装载的值为空时，计数器不工作。

17.4.10 TIM14 捕捉 / 比较寄存器 1 (TIM14_CCR1)

偏移地址 : 0x34

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 CCR1[15:0]: 捕捉 / 比较 1 的值

若 CC1 通道配置为输出：

CCR1 包含了装入当前捕获 / 比较 1 寄存器的值（预装载值）。

如果在 TIMx_CCMR1 寄存器（OC1PE 位）中未选择预装载特性，写入的数值会被立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获 / 比较 1 寄存器中。

当前捕获 / 比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC1 端口上产生输出信号。

若 CC1 通道配置为输入：

CCR1 包含了由上一次输入捕获 1 事件 (IC1) 传输的计数器值。

17.4.11 TIM14 选项寄存器 (TIM14_OR)

偏移地址 : 0x50

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1_RMP
															rw

位 15:1 保留，始终保持复位值 .

位 0 TI1_RMP: 定时器输入 1 重映射（Timer Input 1 remap）

软件清除或设置

0: TIM14 通道 1 连接到 GPIO: 参考器件数据手册中关于可变功能映射表 （ Alternate function mapping table）。

1: RTC_CLK 连接到 TIM14_CH1 输入端实现校正功能。

17.4.12 TIM14 寄存器映射

下表中将 TIM14 的所有寄存器映射到一个 16 位可寻址 (编址) 空间。

表 51. TIM14 寄存器映射与 复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**TIM14_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKD [1:0]		ARPE	Res.	Res.	Res.	Res.	URS	UDIS	CEN
	Reset value																							0	0	0					0	0	0
0x08	**Reserved**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x0C	**TIM14_DIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1IE	UIE
	Reset value																															0	0
0x10	**TIM14_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1OF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1IF	UIF
	Reset value																							0								0	0
0x14	**TIM14_EGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1G	UG
	Reset value																															0	0
0x18	**TIM14_CCMR1** Output compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
	Reset value																										0	0	0	0	0	0	0
	TIM14_CCMR1 Input capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC1F[3:0]				IC1 PSC [1:0]		CC1S [1:0]	
	Reset value																									0	0	0	0	0	0	0	0
0x1C	**Reserved**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x20	**TIM14_CCER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1NP	Res.	CC1P	CC1E
	Reset value																													0		0	0
0x24	**TIM14_CNT**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	**TIM14_PSC**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**TIM14_ARR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	**Reserved**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x34	**TIM14_CCR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38 to 0x4C	**Reserved**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																

表 51. TIM14 寄存器映射与复位值（续）

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x50	**TIM14_OR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1_RMP
	Reset value																																0

18 通用定时器 (TIM15/16/17)

18.1 TIM15/16/17 简介

通用定时器 TIM15/16/17 由一个 16 位自动装载计数器组成，由一个可编程的预分频器驱动。它适合多种用途，包含测量输入信号的脉冲宽度 (输入捕获)，或者产生输出波形 (输出比较和 PWM)。

使用定时器预分频器和 RCC 时钟控制预分频器，可以实现脉冲宽度和波形周期从几个微秒到几个毫秒的调节。

通用定时器 TIM15/16/17 是完全独立的，它们不共享任何资源。它们可以同步操作，具体描述参看 16.3.15 节。

18.2 TIM15 主要功能

通用 TIM15 定时器功能包括：

- 16 位向上自动装载计数器
- 16 位可编程 (可以实时修改) 预分频器，计数器时钟频率的分频系数为 1 ～ 65536 之间的任意数值
- 2 个独立通道：
  - 输入捕获
  - 输出比较
  - PWM 生成 (边缘或中间对齐模式)
  - 单脉冲模式输出
- 可编程死区时间的互补输出（仅通道 1）
- 使用外部信号控制定时器和定时器互连的同步电路
- 允许在指定数目的计数器周期之后更新定时器寄存器的重复计数器
- 刹车输入信号可以将定时器输出信号置于复位状态或者一个已知状态
- 如下事件发生时产生中断 /DMA：
  - 更新：计数器溢出，计数器初始化 (通过软件或者内部 / 外部触发)
  - 触发事件 (计数器启动、停止、初始化或者由内部 / 外部触发计数)
  - 输入捕获
  - 输出比较
  - 刹车输入（中断请求）

18.3 TIM16 和 TIM17 主要特性

通用 TIM16/TIM17 定时器功能包括：

- 16 位向上自动装载计数器
- 16 位可编程 (可以实时修改) 预分频器，计数器时钟频率的分频系数为 1 ～ 65536 之间的任意数值
- 2 个独立通道：
  - 输入捕获
  - 输出比较
  - PWM 生成 (边缘或中间对齐模式)
  - 单脉冲模式输出
- 可编程死区时间的互补输出
- 允许在指定数目的计数器周期之后更新定时器寄存器的重复计数器
- 刹车输入信号可以将定时器输出信号置于复位状态或者一个已知状态
- 如下事件发生时产生中断 /DMA：
  - 更新：计数器溢出
  - 触发事件 (计数器启动、停止、初始化或者由内部 / 外部触发计数)
  - 输入捕获
  - 输出比较
  - 刹车输入（中断请求）

图 151. TIM15 框图

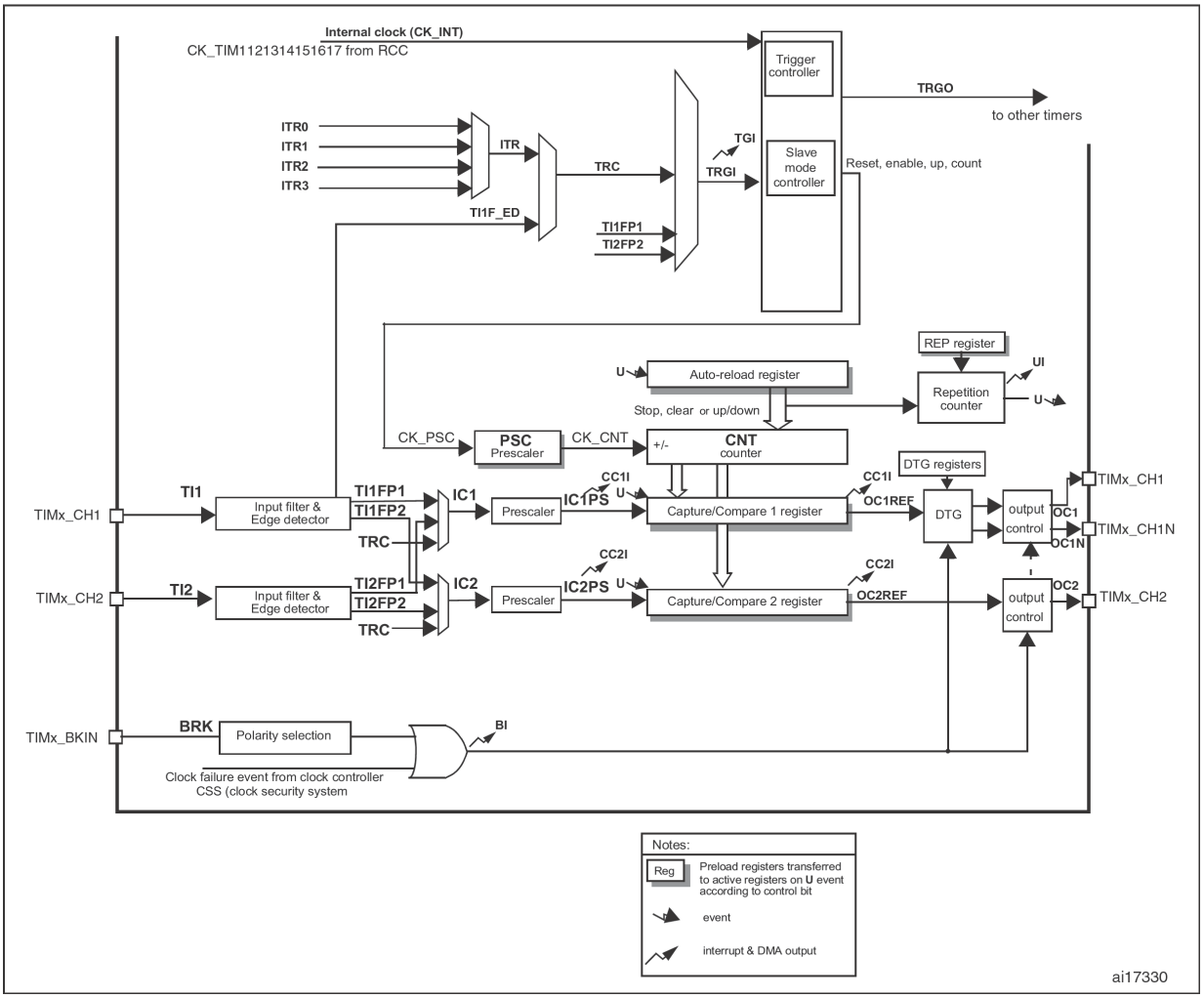
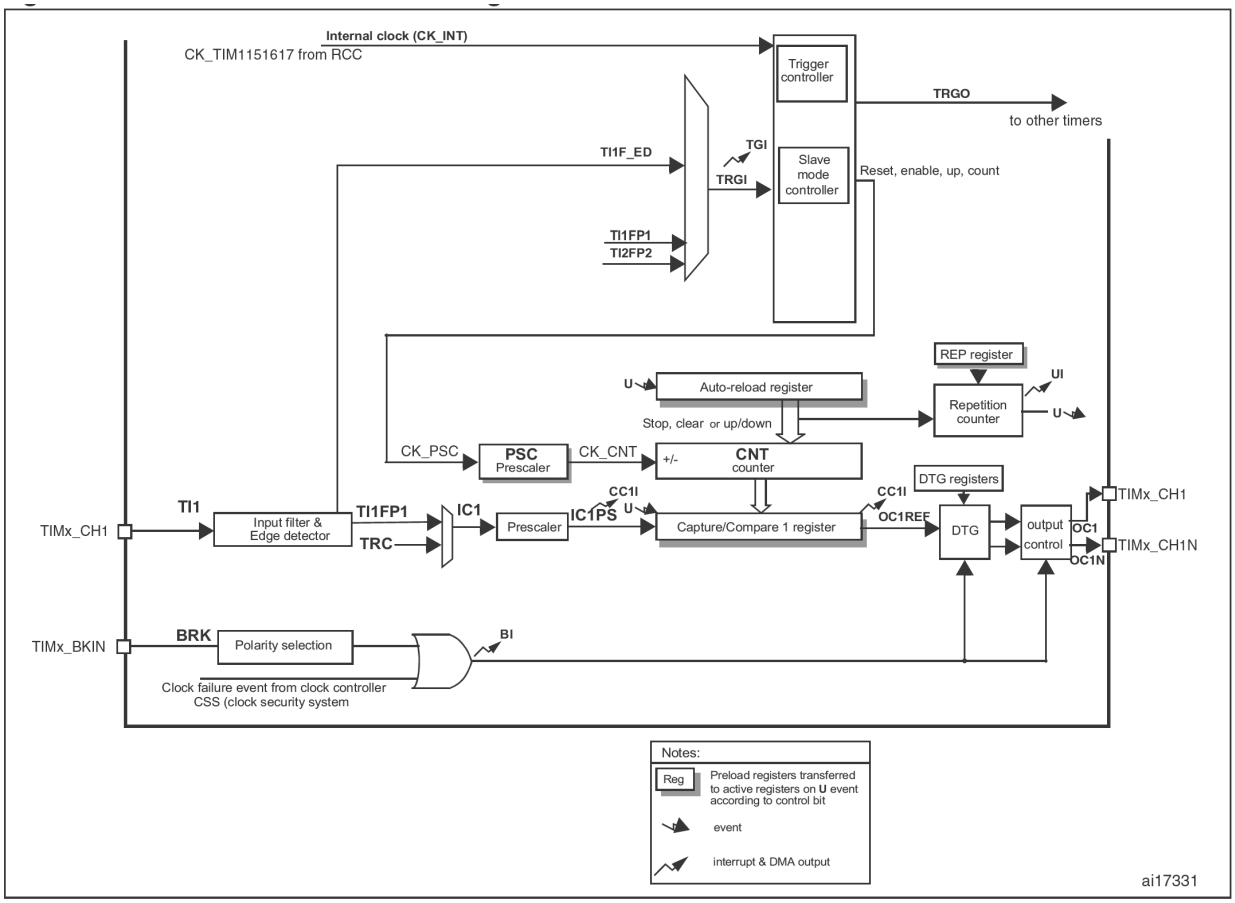


图 152. TIM16 和 TIM17 框图



18.4 TIM15/16/17 功能描述

18.4.1 时基单元

可编程通用定时器的主要部分是一个 16 位计数器和与其相关的自动装载寄存器。这个计数器可以向上计数、向下计数或者向上向下双向计数。此计数器时钟由预分频器分频得到。
计数器、自动装载寄存器和预分频器寄存器可以由软件读写，在计数器运行时仍可以读写。
时基单元包含：

- 计数器寄存器 (TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动装载寄存器 (TIMx_ARR)
- 重复寄存器 (TIMx_RCR)

自动装载寄存器是预先装载的，写或读自动重装载寄存器将访问预装载寄存器。根据在 TIMx_CR1 寄存器中的自动装载预装载使能位 (ARPE) 的设置，预装载寄存器的内容被立即或在 每次的更新事件 UEV 时传送到影子寄存器。当计数器达到溢出条件 (向下计数时的下溢条件) 并当 TIMx_CR1 寄存器中的 UDIS 位等于’0’时，产生更新事件。更新事件也可以由软件产生。随后会详细描述每一种配置下更新事件的产生。

计数器由预分频器的时钟输出 CK_CNT 驱动，仅当设置了计数器 TIMx_CR1 寄存器中的计数器使能位 (CEN) 时，CK_CNT 才有效。(有关计数器使能的细节，请参见控制器的从模式描述)。
注：真正的计数器开始计数在 CEN 的一个时钟周期后被设置。

预分频器描述
预分频器可以将计数器的时钟频率按 1 到 65536 之间的任意值分频。它是基于一个 (在 TIMx_PSC 寄存器中的)16 位寄存器控制的 16 位计数器。这个控制寄存器带有缓冲器，它能够在工作时被改变。新的预分频器参数在下一次更新事件到来时被采用。

图 138 和图 139 给出了在预分频器运行时更改计数器参数的例子。

图 153. 当预分频器的参数从 1 变到 2 时，计数器的时序图

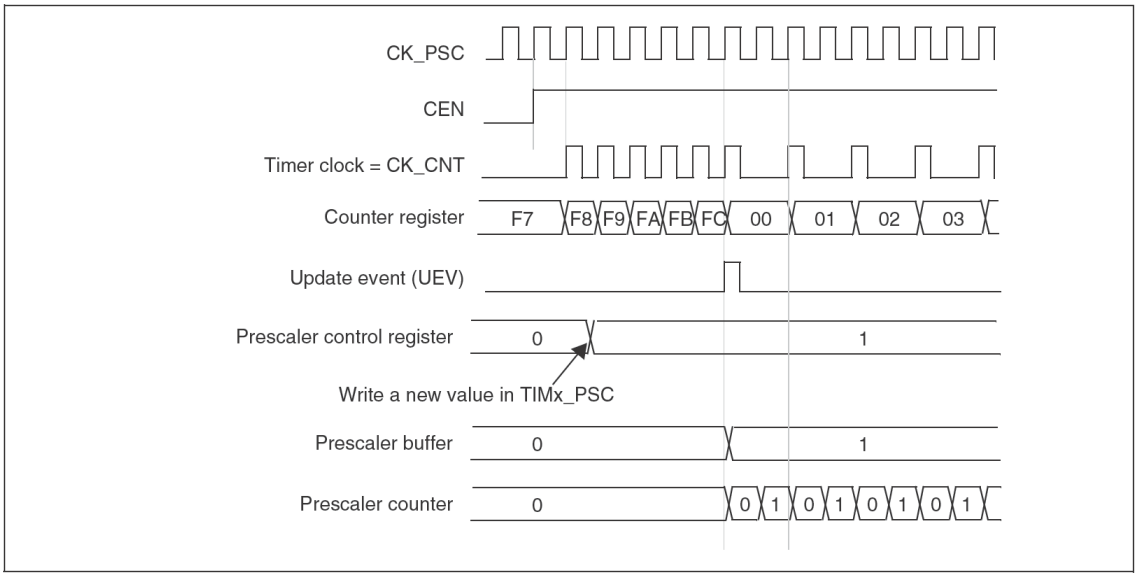
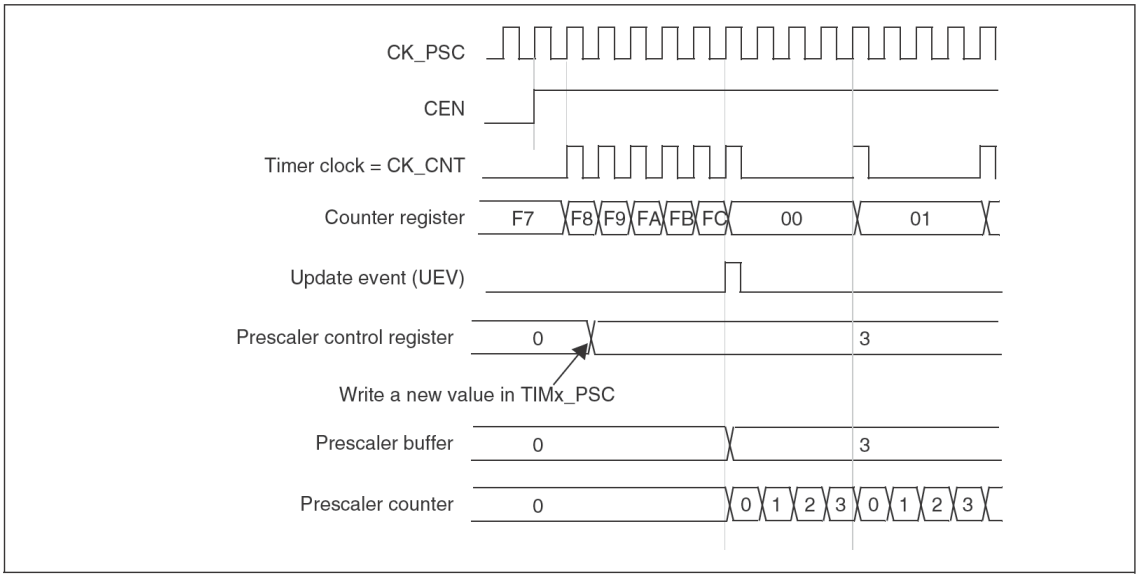


图 154. 当预分频器的参数从 1 变到 4 时，计数器的时序图



18.4.2 计数模式

向上计数模式

在向上计数模式中，计数器从 0 计数到自动加载值 (TIMx_ARR 计数器的内容)，然后重新从 0 开始计数并且产生一个计数器溢出事件。

如果使用了重复计数器功能，在向上计数达到设置的重复计数次数 (TIMx_RCR) 时，才产生更新事件 (UEV)；否则每次计数器溢出时都会产生更新事件。

每次计数器溢出时可以产生更新事件，在 TIMx_EGR 寄存器中 (通过软件方式或者使用从模式控制器) 设置 UG 位也同样可以产生一个更新事件。

设置 TIMx_CR1 寄存器中的 UDIS 位，可以禁止更新事件；这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。在 UDIS 位被清’0’之前，将不产生更新事件。但是在应该产生更新事件时，计数器仍会被清’0’，同时预分频器的计数也被请 0(但预分频系数不变)。此外，如果设置了 TIMx_CR1 寄存器中的 URS 位 (选择更新请求)，设置 UG 位将产生一个更新事件 UEV，但硬件不设置 UIF 标志 (即不产生中断或 DMA 请求)；这是为了避免在捕获模式下清除计数器时，同时产生更新和捕获中断。
当发生一个更新事件时，所有的寄存器都被更新，硬件同时 (依据 URS 位) 设置更新标志位 (TIMx_SR 寄存器中的 UIF 位)。

- 重复寄存器被置入寄存器的值 (TIMx_RCR)。
- 自动装载影子寄存器被重新置入预装载寄存器的值 (TIMx_ARR)。
- 预分频器的缓冲区被置入预装载寄存器的值 (TIMx_PSC 寄存器的内容)。

下图给出一些例子，当 TIMx_ARR=0x36 时计数器在不同时钟频率下的动作。

图 155. 计数器时序图，内部时钟分频因子为 1

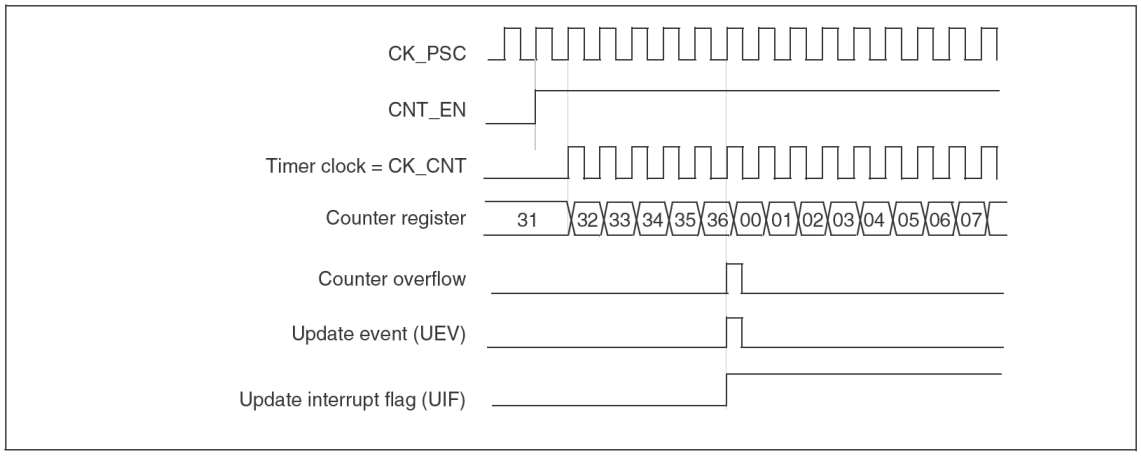


图 156. 计数器时序图，内部时钟分频因子为 2

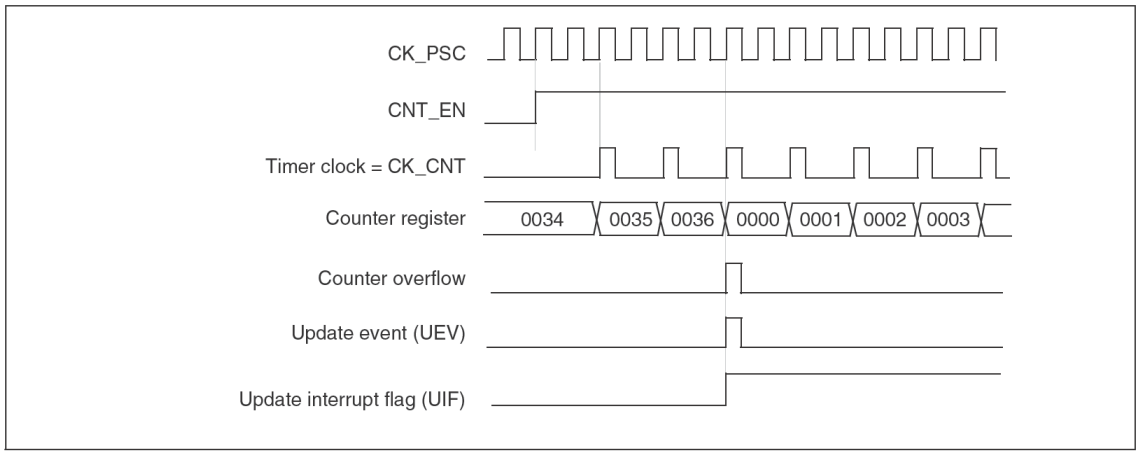


图 157. 计数器时序图，内部时钟分频因子为 4

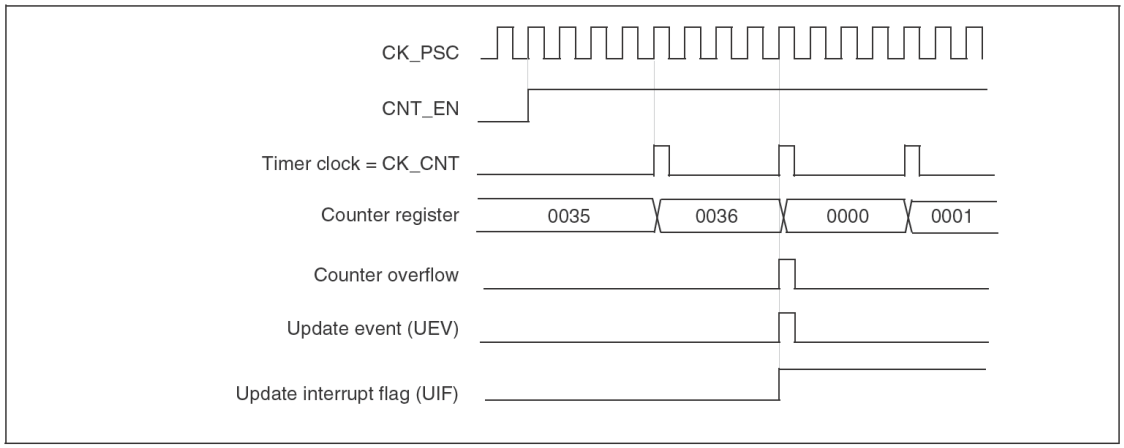


图 158. 计数器时序图，内部时钟分频因子为 N

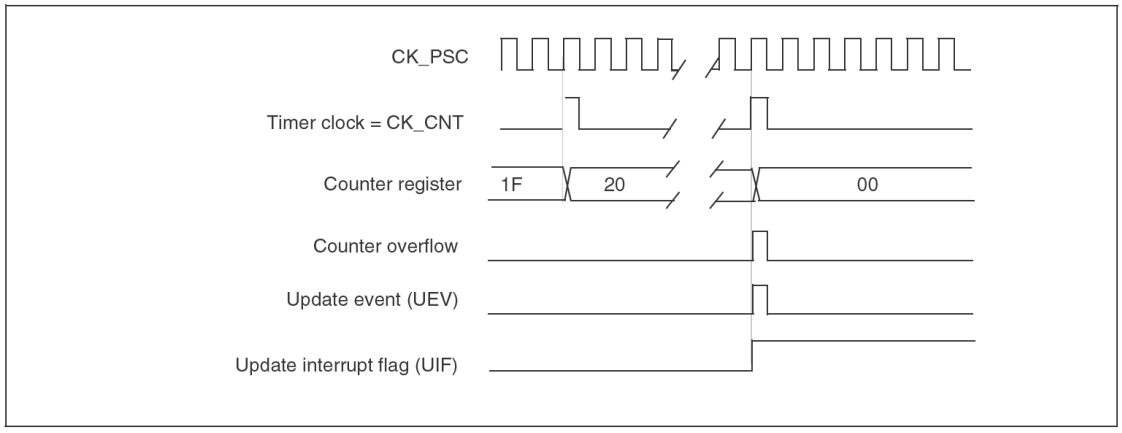


图 159. 计数器时序图，当 ARPE=0 时的更新事件 (TIMx_ARR 没有预装入)

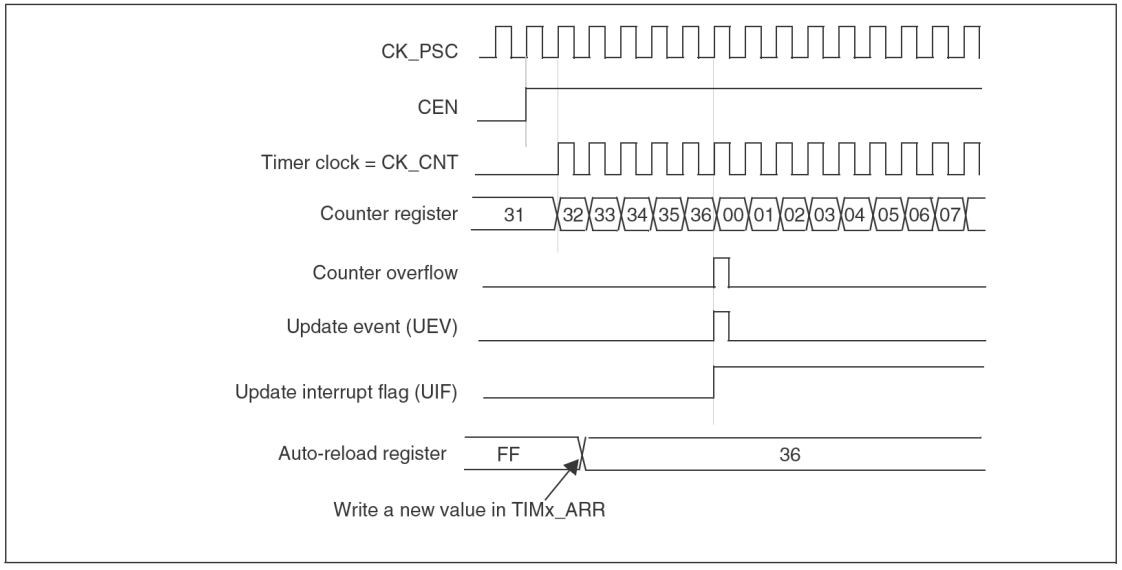
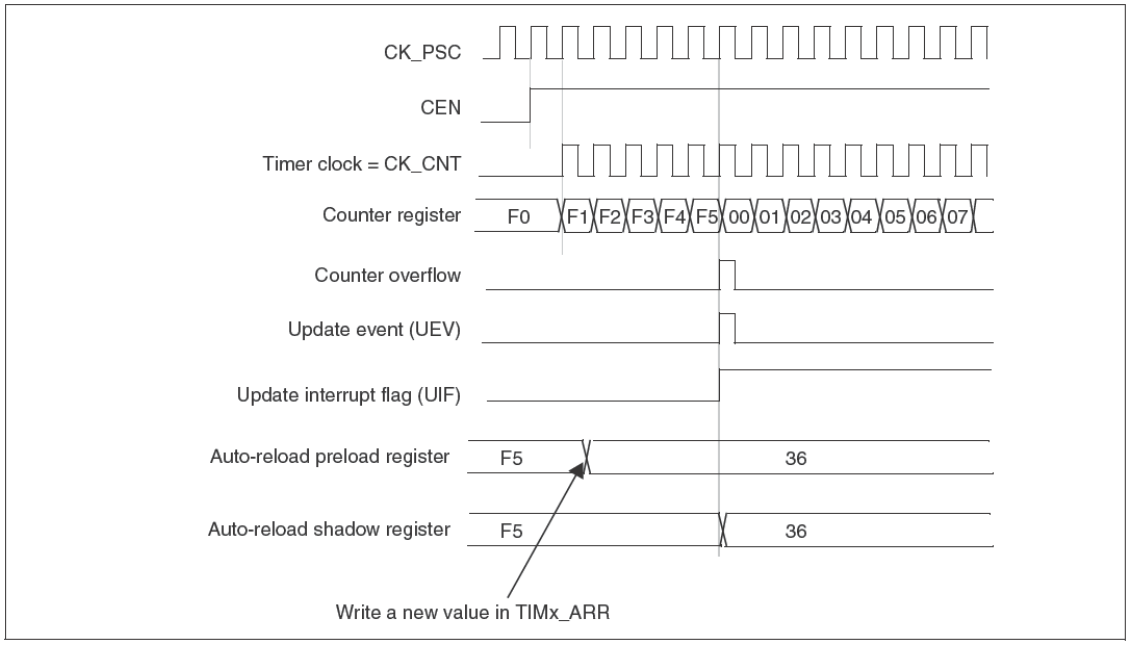


图 160. 计数器时序图，当 ARPE=1 时的更新事件 (预装入了 TIMx_ARR)



18.4.3 重复计数器

17.3.1 节“时基单元”解释了计数器上溢 / 下溢时是如何产生更新事件 (UEV) 的，然而事实上它只能在重复计数达到 0 的时候产生。这个特性对产生 PWM 信号非常有用。

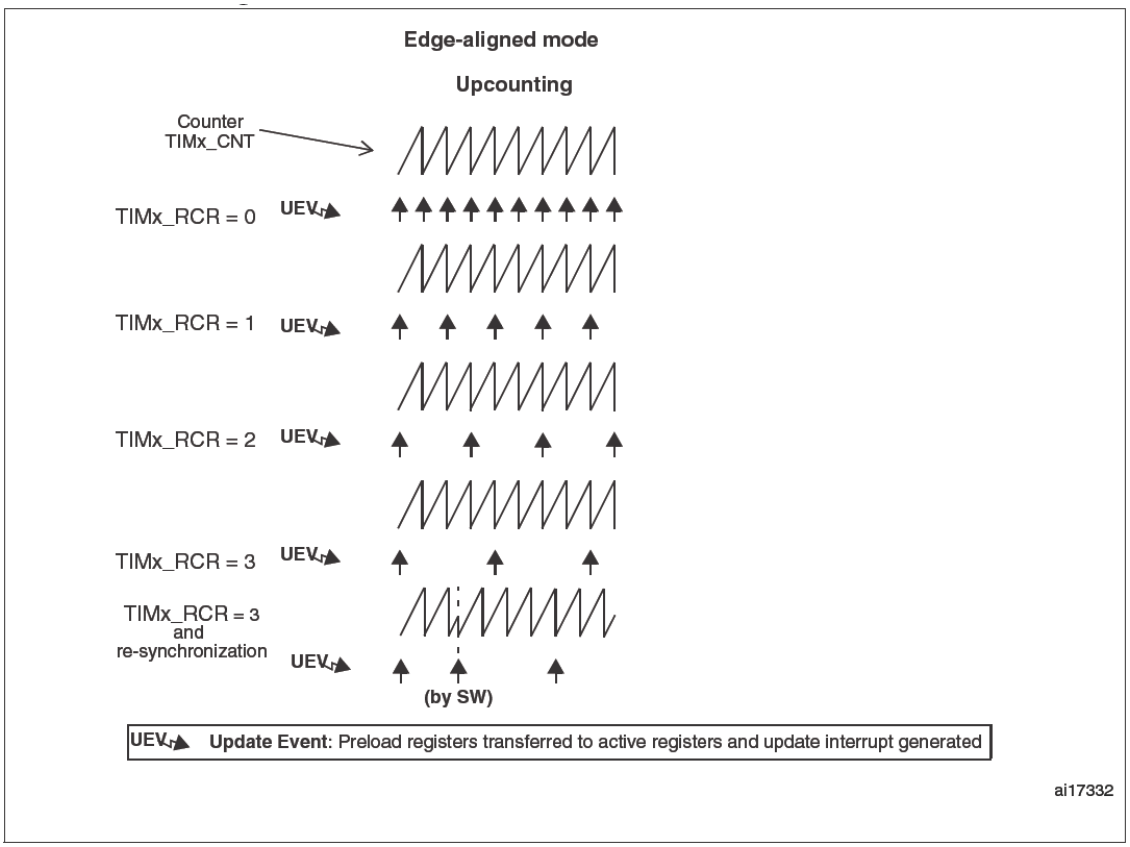
这意味着在每 N 次计数上溢或下溢时，数据从预装载寄存器传输到影子寄存器 (TIMx_ARR 自动重载入寄存器，TIMx_PSC 预装 载寄存器，还有在比较模式下的捕获 / 比较寄存器 TIMx_CCRx)，N 是 TIMx_RCR 重复计数寄存器中的值。

重复计数器在下述任一条件成立时递减：

- 向上计数模式下每次计数器溢出时，

重复计数器是自动加载的，重复速率是由 TIMx_RCR 寄存器的值定义 (参看图 161)。当更新事件由软件产生 (通过设置 TIMx_EGR 中的 UG 位) 或者通过硬件的从模式控制器产生，则无论重复计数器的值是多少，立即发生更新事件，并且 TIMx_RCR 寄存器中的内容被重载入到重复计数器。

图 161. 不同模式及不同 TIMx_RCR 寄存器设置下更新速率的例子



18.4.4 时钟源

计数器时钟可由下列时钟源提供：

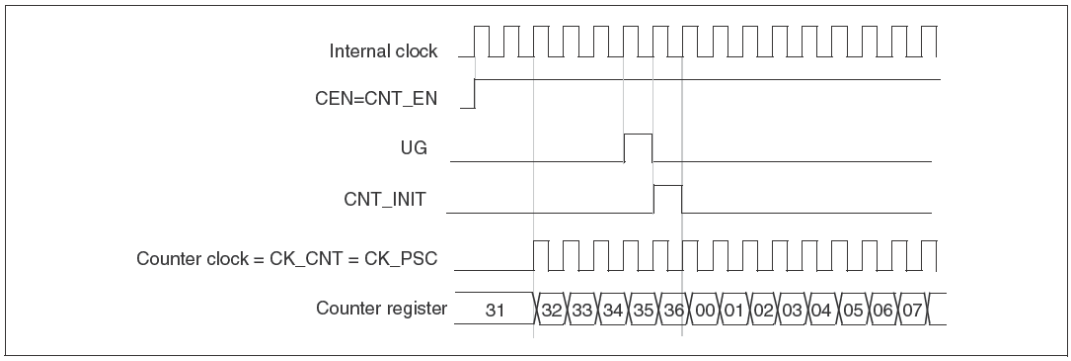
- 内部时钟 (CK_INT)
- 外部时钟模式 1：外部输入脚 (TIx)
- 内部触发输入 (ITRx)（仅 TIM15）：使用一个定时器作为另一个定时器的预分频器，如可以配置一个定时器 TIM1 而作为另一个定时器 TIM15 的预分频器。参见使用一个定时器作为另一个定时器的预分频器。

内部时钟源 (CK_INT)

如果禁止了从模式控制器 (TIMx_SMCR 寄存器的 SMS=000)，则 CEN、DIR(TIMx_CR1 寄存器) 和 UG 位 (TIMx_EGR 寄存器) 是事实上的控制位，并且只能被软件修改 (UG 位仍被自动清除)。只要 CEN 位被写成’1’，预分频器的时钟就由内部时钟 CK_INT 提供。

图 162 显示了控制电路和向上计数器在一般模式下，不带预分频器时的操作。

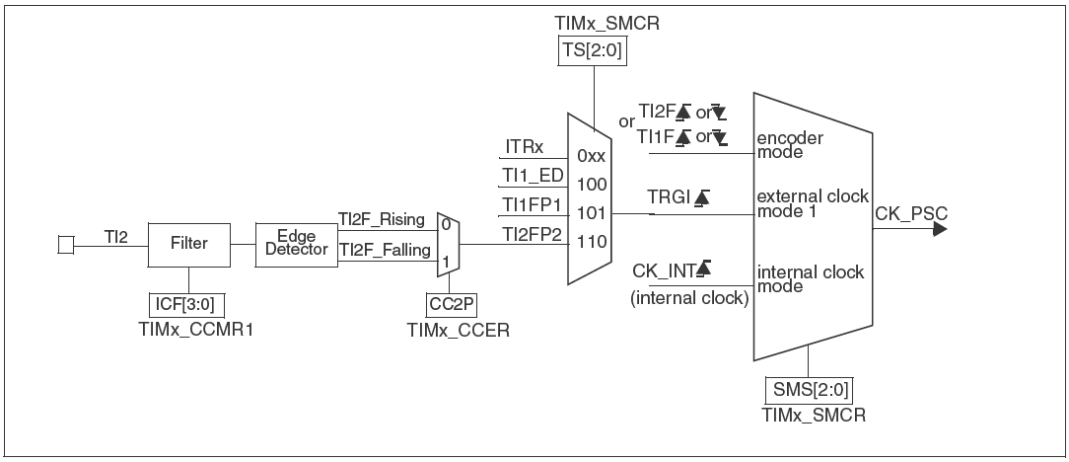
图 162. 一般模式下的控制电路，内部时钟分频因子为 1



外部时钟源模式 1

当 TIMx_SMCR 寄存器的 SMS=111 时，此模式被选中。计数器可以在选定输入端的每个上升沿或下降沿计数。

图 163. TI2 外部时钟连接示例

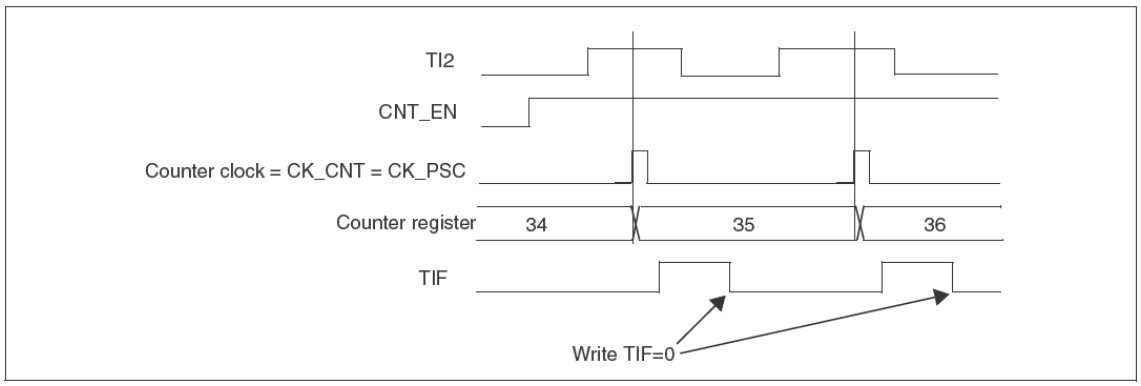


例如，要配置向上计数器在 T12 输入端的上升沿计数，使用下列步骤：

1. 配置 TIMx_CCMR1 寄存器 CC2S=’01’，配置通道 2 检测 TI2 输入的上升沿
2. 配置 TIMx_CCMR1 寄存器的 IC2F[3:0]，选择输入滤波器带宽 (如果不需要滤波器，保持 IC2F=0000)
3. 配置 TIMx_CCER 寄存器的 CC2P=’0’，选定上升沿极性
4. 配置 TIMx_SMCR 寄存器的 SMS=’111’，选择定时器外部时钟模式 1
5. 配置 TIMx_SMCR 寄存器中的 TS=’110’，选定 TI2 作为触发输入源
6. 设置 TIMx_CR1 寄存器的 CEN=’1’，启动计数器

注：捕获预分频器不用作触发，所以不需要对它进行配置当上升沿出现在 TI2，计数器计数一次，且 TIF 标志被设置。
在 TI2 的上升沿和计数器实际时钟之间的延时，取决于在 TI2 输入端的重新同步电路。

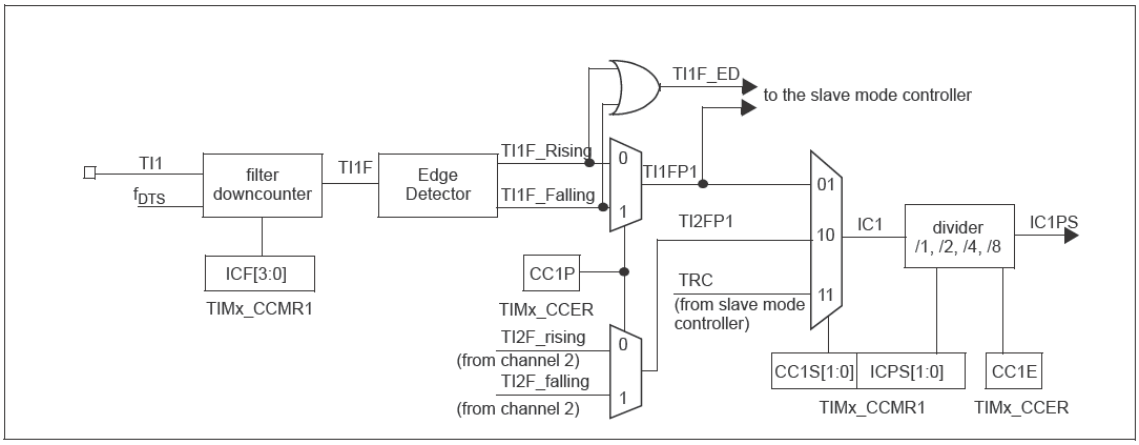
图 164. 外部时钟模式 1 下的控制电路



18.4.5 捕获 / 比较通道

每一个捕获 / 比较通道都是围绕着一个捕获 / 比较寄存器 (包含影子寄存器)，包括捕获的输入部分 (数字滤波、多路复用和预分频器)，和输出部分 (比较器和输出控制)。
下面几张图是一个捕获 / 比较通道概览。
输入部分对相应的 TIx 输入信号采样，并产生一个滤波后的信号 TIxF。然后，一个带极性选择的边缘检测器产生一个信号 (TIxFPx)，它可以作为从模式控制器的输入触发或者作为捕获控制。该信号通过预分频进入捕获寄存器 (ICxPS)。

图 165. 捕获 / 比较通道 (如：通道 1 输入部分)



输出部分产生一个中间波形 OCxRef(高有效) 作为基准，链的末端决定最终输出信号的极性。

图 166. 捕获 / 比较通道 1 的主电路

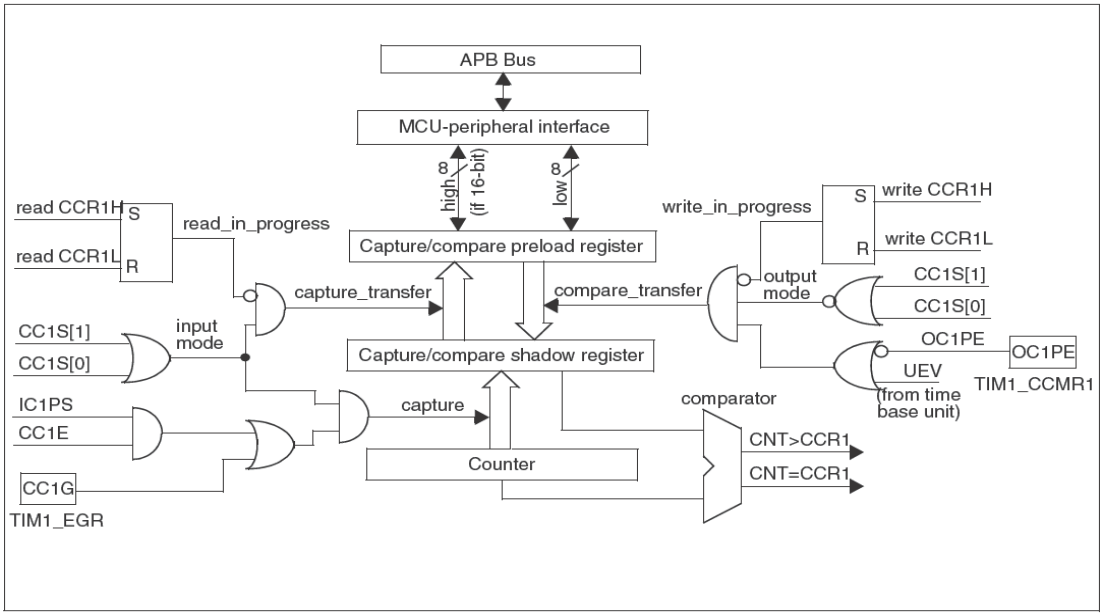


图 167. 捕获 / 比较通道的输出部分 (通道 1)

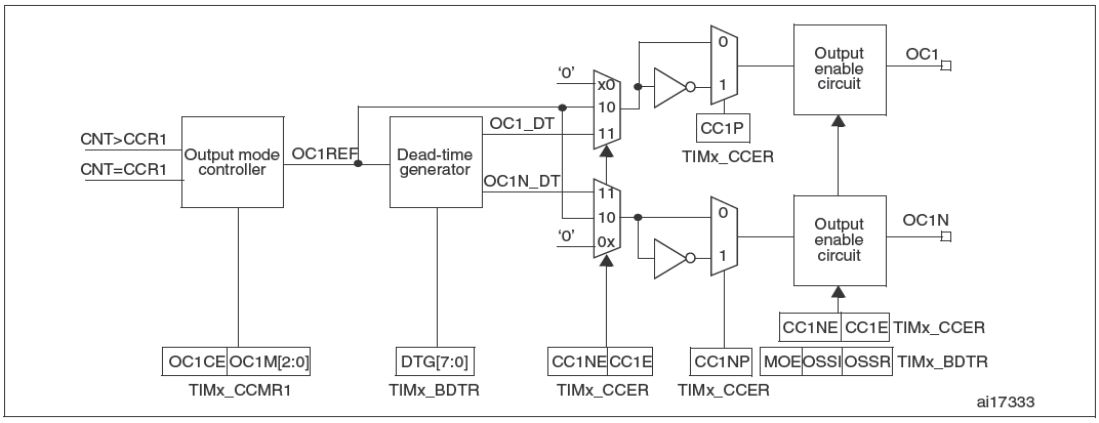
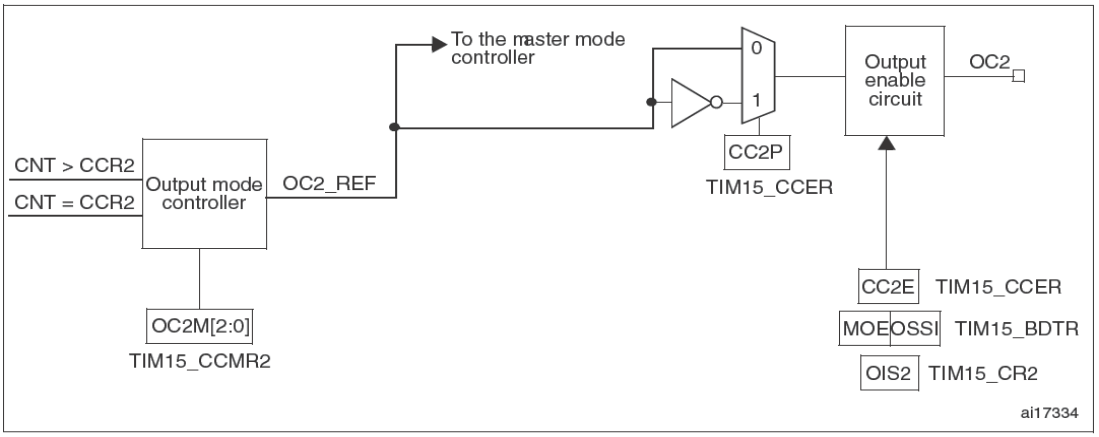


图 168. 捕获 / 比较通道的输出部分 (TIM15 的通道 2)



捕获 / 比较模块由一个预装载寄存器和一个影子寄存器组成。读写过程仅操作预装载寄存器。
在捕获模式下，捕获发生在影子寄存器上，然后再复制到预装载寄存器中。
在比较模式下，预装载寄存器的内容被复制到影子寄存器中，然后影子寄存器的内容和计数器进行比较。

18.4.6 输入捕获模式

在输入捕获模式下，当检测到 ICx 信号上相应的边沿后，计数器的当前值被锁存到捕获 / 比较寄存器 (TIMx_CCRx) 中。当捕获事件发生时，相应的 CCxIF 标志 (TIMx_SR 寄存器) 被置’1’，如果使能了中断或者 DMA 操作，则将产生中断或者 DMA 操作。如果捕获事件发生时 CCxIF 标志已经为高，那么重复捕获标志 CCxOF(TIMx_SR 寄存器) 被置’1’。写 CCxIF=0 可清除 CCxIF，或读取存储在 TIMx_CCRx 寄存器中的捕获数据也可清除 CCxIF。写 CCxOF=0 可清除 CCxOF。
以下例子说明如何在 TI1 输入的上升沿时捕获计数器的值到 TIMx_CCR1 寄存器中，步骤如下：

- 选择有效输入端：TIMx_CCR1 必须连接到 TI1 输入，所以写入 TIMx_CCMR1 寄存器中的 CC1S=01，只要 CC1S 不为’00’，通道被配置为输入，并且 TM1_CCR1 寄存器变为只读。
- 根据输入信号的特点，配置输入滤波器为所需的带宽 (即输入为 TIx 时，输入滤波器控制位是 TIMx_CCMRx 寄存器中的 ICxF 位)。假设输入信号在最多 5 个内部时钟周期的时间内抖动， 我们须配置滤波器的带宽长于 5 个时钟周期。因此我们可以 (以 fDTS 频率) 连续采样 8 次，以确 认在 TI1 上一次真实的边沿变换，即在 TIMx_CCMR1 寄存器中写入 IC1F=0011。
- 选择 TI1 通道的有效转换边沿，在 TIMx_CCER 寄存器中写入 CC1P=0 (本例中是上升沿)。
- 配置输入预分频器。在本例中，我们希望捕获发生在每一个有效的电平转换时刻，因此预分频器被禁止 (写 TIMx_CCMR1 寄存器的 IC1PS=00)。
- 设置 TIMx_CCER 寄存器的 CC1E=1，允许捕获计数器的值到捕获寄存器中。
- 如果需要，通过设置 TIMx_DIER 寄存器中的 CC1IE 位允许相关中断请求，通过设置
- TIMx_DIER 寄存器中的 CC1DE 位允许 DMA 请求。

当发生一个输入捕获时：

- 产生有效的电平转换时，计数器的值被传送到 TIMx_CCR1 寄存器。
- CC1IF 标志被设置 (中断标志)。当发生至少 2 个连续的捕获时，而 CC1IF 未曾被清除，
- CC1OF 也被置’1’。
- 如设置了 CC1IE 位，则会产生一个中断。
- 如设置了 CC1DE 位，则还会产生一个 DMA 请求。

为了处理捕获溢出，建议在读出捕获溢出标志之前读取数据，这是为了避免丢失在读出捕获溢出标志之后和读取数据之前可能产生的捕获溢出信息。

注：设置 TIMx_EGR 寄存器中相应的 CCxG 位，可以通过软件产生输入捕获中断和 / 或 DMA 请求。

18.4.7 PWM 输入 (仅 TIM15)

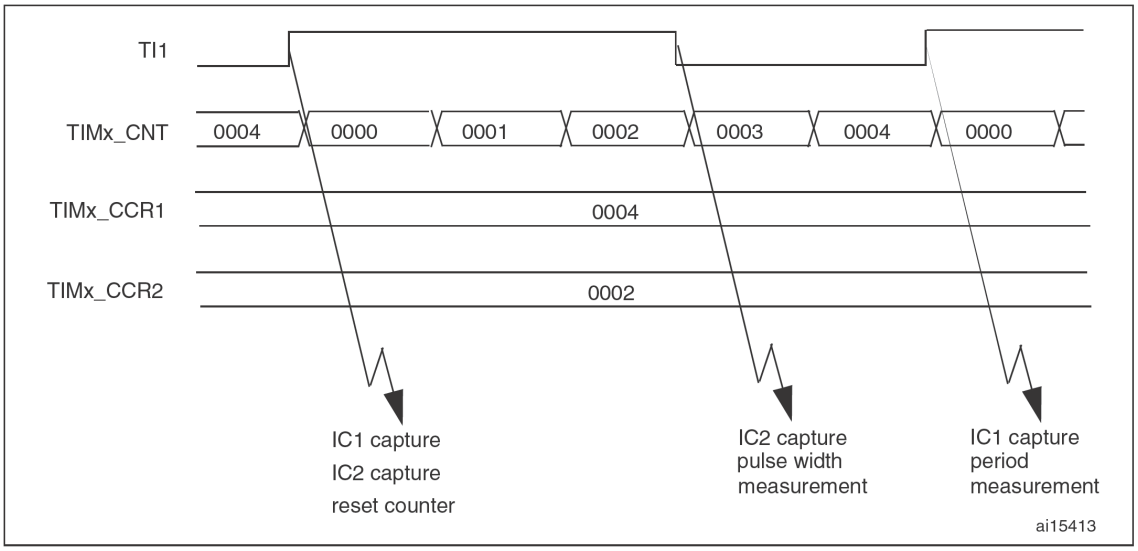
该模式是输入捕获模式的一个特例，除下列区别外，操作与输入捕获模式相同：

- 两个 ICx 信号被映射至同一个 TIx 输入。
- 这 2 个 ICx 信号为边沿有效，但是极性相反。
- 其中一个 TIxFP 信号被作为触发输入信号，而从模式控制器被配置成复位模式。

例如，你需要测量输入到 TI1 上的 PWM 信号的长度 (TIMx_CCR1 寄存器) 和占空比 (TIMx_CCR2 寄存器)，具体步骤如下 (取决于 CK_INT 的频率和预分频器的值)

- 选择 TIMx_CCR1 的有效输入：置 TIMx_CCMR1 寄存器的 CC1S=01(选择 TI1)。
- 选择 TI1FP1 的有效极性 (用来捕获数据到 TIMx_CCR1 中和清除计数器)：置 CC1P=0 (上升沿有效)。
- 选择 TIMx_CCR2 的有效输入：置 TIMx_CCMR1 寄存器的 CC2S=10(选择 TI1)。
- 选择 TI1FP2 的有效极性 (捕获数据到 TIMx_CCR2)：置 CC2P=1 (下降沿有效)。
- 选择有效的触发输入信号：置 TIMx_SMCR 寄存器中的 TS=101(选择 TI1FP1)。
- 配置从模式控制器为复位模式：置 TIMx_SMCR 中的 SMS=100。
- 使能捕获：置 TIMx_CCER 寄存器中 CC1E=1 且 CC2E=1。

图 169. PWM 输入模式时序



1. PWM 输入模式仅可用于 TIMx_CH1/TIMx_CH2 是因为事实上只有 TI1FP1 和 TI2FP2 被连接到从模式控制器。

18.4.8 强置输出模式

在输出模式 (TIMx_CCMRx 寄存器中 CCxS=00) 下，输出比较信号 (OCxREF 和相应的 OCx/OCxN 由软件强置为有效或无效状态，而不依赖于输出比较寄存器和计数器间的比较结果。
置 TIMx_CCMRx 寄存器中相应的 OCxM=101，即可强置输出比较信号 (OCxREF/OCx) 为有效状态。这样 OCxREF 被强置为高电平 (OCxREF 始终为高电平有效)，同时 OCx 得到 CCxP 极性位相反的值。
例如：CCxP=0(OCx 高电平有效)，则 OCx 被强置为高电平。
置 TIMx_CCMRx 寄存器中的 OCxM=100，可强置 OCxREF 信号为低。

该模式下，在 TIMx_CCRx 影子寄存器和计数器之间的比较仍然在进行，相应的标志也会被修改。因此仍然会产生相应的中断和 DMA 请求。这将会在下面的输出比较模式一节中介绍。

18.4.9 输出比较模式

此功能是用来控制一个输出波形，或者指示一段给定的时间已经到时。
当计数器与捕获 / 比较寄存器的内容相同时，输出比较功能做如下操作：

- 将输出比较模式 (TIMx_CCMRx 寄存器中的 OCxM 位) 和输出极性 (TIMx_CCER 寄存器中的 CCxP 位) 定义的值输出到对应的引脚上。在比较匹配时，输出引脚可以保持它的电平 (OCxM=000)、被设置成有效电平 (OCxM=001)、被设置成无效电平 (OCxM=010) 或进行翻转 (OCxM=011)。
- 设置中断状态寄存器中的标志位 (TIMx_SR 寄存器中的 CCxIF 位)。
- 若设置了相应的中断屏蔽 (TIMx_DIER 寄存器中的 CCxIE 位)，则产生一个中断。
- 若设置了相应的使能位 (TIMx_DIER 寄存器中的 CCxDE 位，TIMx_CR2 寄存器中的 CCDS 位选择 DMA 请求功能)，则产生一个 DMA 请求。

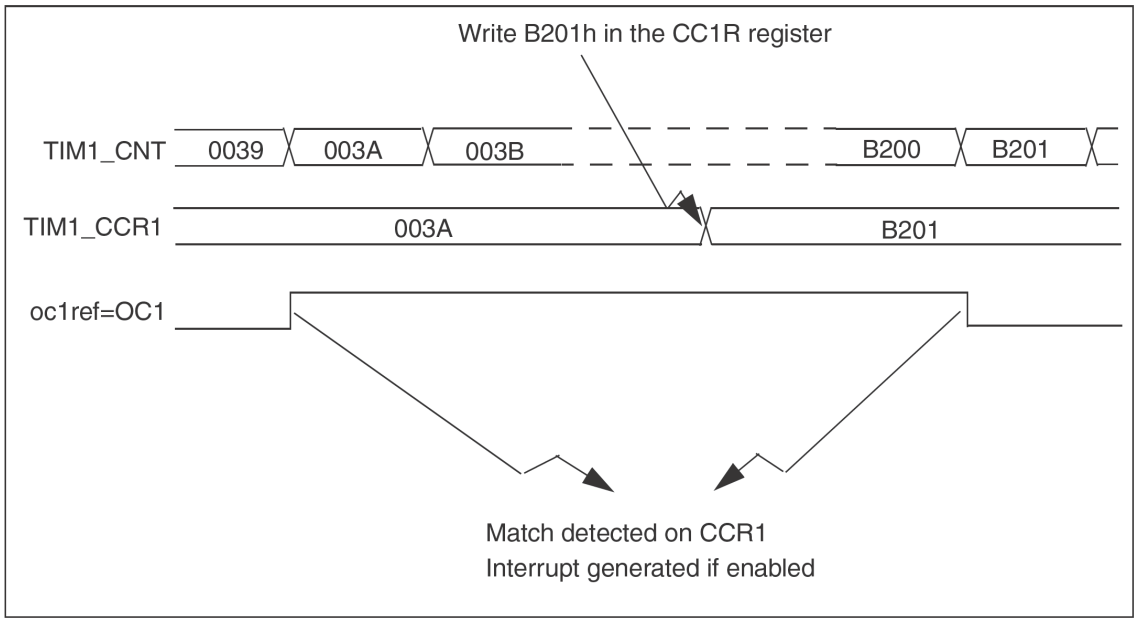
TIMx_CCMRx 中的 OCxPE 位选择 TIMx_CCRx 寄存器是否需要使用预装载寄存器。
在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出没有影响。 同步的精度可以达到计数器的一个计数周期。输出比较模式 (在单脉冲模式下) 也能用来输出一个单脉冲。

输出比较模式的配置步骤：

1. 选择计数器时钟 (内部，外部，预分频器)
2. 将相应的数据写入 TIMx_ARR 和 TIMx_CCRx 寄存器中
3. 如果要产生一个中断请求，设置 CCxIE 位。
4. 选择输出模式 :
   - 置 OCxM=’011’，CNT 与 CCRx 匹配时翻转 OCx 的输出引脚
   - 写 OCxPE=’0’，预装载未用
   - 写 CCxP=’0’，高电平有效
   - 写 CCxE=’1’，开启 OCx 输出
5. 设置 TIMx_CR1 寄存器的 CEN 位启动计数器

TIMx_CCRx 寄存器能够在任何时候通过软件进行更新以控制输出波形，条件是未使用预装载寄存器 (OCxPE=’0’，否则 TIMx_CCRx 影子寄存器只能在发生下一次更新事件时被更新)。下图给出了一个例子。

图 170. 输出比较模式，翻转 OC1



18.4.10 PWM 模式

脉冲宽度调制模式可以产生一个由 TIMx_ARR 寄存器确定频率、由 TIMx_CCRx 寄存器确定占空比的信号。

在 TIMx_CCMRx 寄存器中的 OCxM 位写入’110’(PWM 模式 1) 或’111’(PWM 模式 2)，能够独立地设置每个 OCx 输出通道产生一路 PWM。必须设置 TIMx_CCMRx 寄存器 OCxPE 位以使能相应的预装载寄存器，最后还要设置 TIMx_CR1 寄存器的 ARPE 位，(在向上计数或中心对称模式中) 使能自动重装载的预装载寄存器。

仅当发生一个更新事件的时候，预装载寄存器才能被传送到影子寄存器，因此在计数器开始计数之前，必须通过设置 TIMx_EGR 寄存器中的 UG 位来初始化所有的寄存器。

OCx 的极性可以通过软件在 TIMx_CCER 寄存器中的 CCxP 位设置，它可以设置为高电平有效或低电平有效。TIMx_CCER 和 TIMx_BDTR 寄存器中的 CCxE、CCxNE、MOE、OSSI OSSR 位控制 OCx 输出使能。详见 TIMx_CCERx 寄存器的描述。

在 PWM 模式 (模式 1 或模式 2) 下，TIMx_CNT 和 TIMx_CCRx 始终在进行比较，(依据计数器的计数方向) 以确 定是否符合 TIMx_CCRx ≤ TIMx_CNT 或者 TIMx_CNT ≤ TIMx_CCRx 。根据 TIMx_CR1 寄存器中 CMS 位的状态，定时器能够产生边沿对齐的 PWM 信号或中央对齐的 PWM 信号。

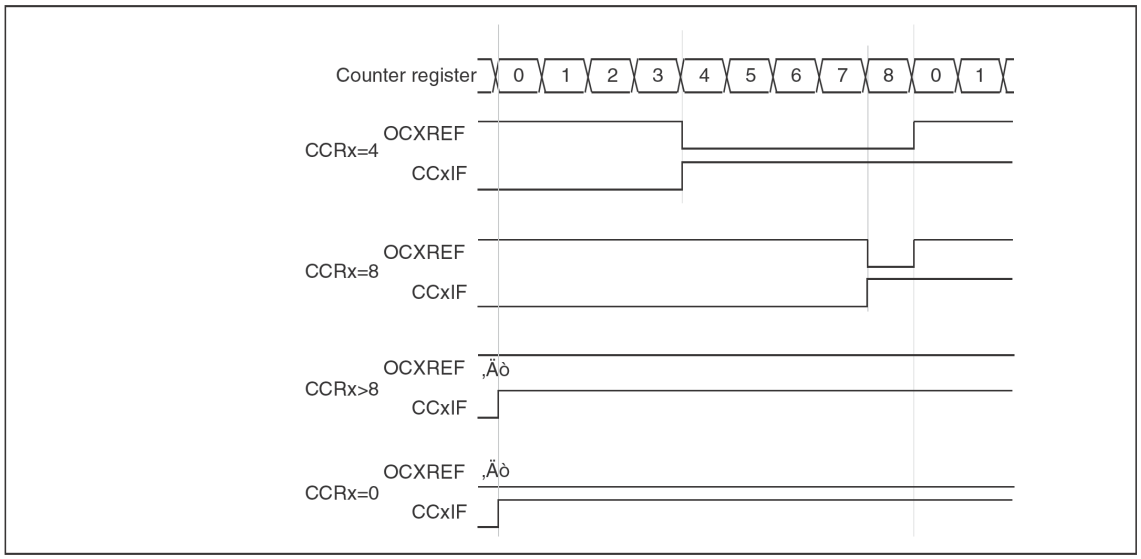
PWM 边沿对齐模式

向上计数配置

当 TIMx_CR1 寄存器中的 DIR 位为低的时候执行向上计数。参看 291 页：向上计数模式

下面是一个 PWM 模式 1 的例子。当 TIMx_CNT<TIMx_CCRx 时 PWM 信号参考 OCxREF 为高，否则为低。如果 TIMx_CCRx 中的比较值大于自动重装载值 (TIMx_ARR)，则 OCxREF 保持为’1’。 如果比较值为 0，则 OCxREF 保持为’0’。 图 171 为 TIMx_ARR=8 时边沿对齐的 PWM 波形实例。

图 171. 边沿对齐的 PWM 波形 (ARR=8)



向下计数的配置

当 TIMx_CR1 寄存器的 DIR 位为高时执行向下计数。参看 379 页：向下计数模式

在 PWM 模式 1 ，当 TIMx_CNT>TIMx_CCRx 时参考信 号 OCxREF 为低，否 则为高。 如果 TIMx_CCRx 中的比较值大于 TIMx_ARR 中的自动重装载值，则 OCxREF 保持为’1’。该模式下不能产生 0%的 PWM 波形。

18.4.11 互补输出和死区插入

通用定时器 TIM15/16/17 能够输出两路互补信号，并且能够管理输出的瞬时关断和接通。
这段时间通常被称为死区，用户应该根据连接的输出器件和它们的特性 (电平转换的延时、电源开关的延时等) 来调整死区时间。
配置 TIMx_CCER 寄存器中的 CCxP 和 CCxNP 位，可以为每一个输出独立地选择极性 (主输出 OCx 或互补输出 OCxN)。

互补信号 OCx 和 OCxN 通过下列控制位的组合进行控制：TIMx_CCER 寄存器的 CCxE 和 CCxNE 位，TIMx_BDTR 和 TIMx_CR2 寄存器中的 MOE、OISx、OISxN、OSSI 和 OSSR 位，详见 410 页表 53：带刹车功能的互补输出通道 OCx 和 OCxN 的控制位。特别的是，在转换到 IDLE 状态时 (MOE 下降到 0) 死区被激活。
同时设置 CCxE 和 CCxNE 位将插入死区，如果存在刹车电路，则还要设置 MOE 位。每一个通道都有一个 10 位的死区发生器。参考信号 OCxREF 可以产生 2 路输出 OCx 和 OCxN。如果 OCx 和 OCxN 为高有效：

- OCx 输出信号与参考信号相同，只是它的上升沿相对于参考信号的上升沿有一个延迟。
- OCxN 输出信号与参考信号相反，只是它的上升沿相对于参考信号的下降沿有一个延迟。

如果延迟大于当前有效的输出宽度 (OCx 或者 OCxN)，则不会产生相应的脉冲。
下列几张图 显示了死区 发生器的输 出信号和当 前参考信号 OCxREF 之 间的关系。 (假设 CCxP=0、CCxNP=0、MOE=1、CCxE=1 并且 CCxNE=1)

图 172. 带死区插入的互补输出

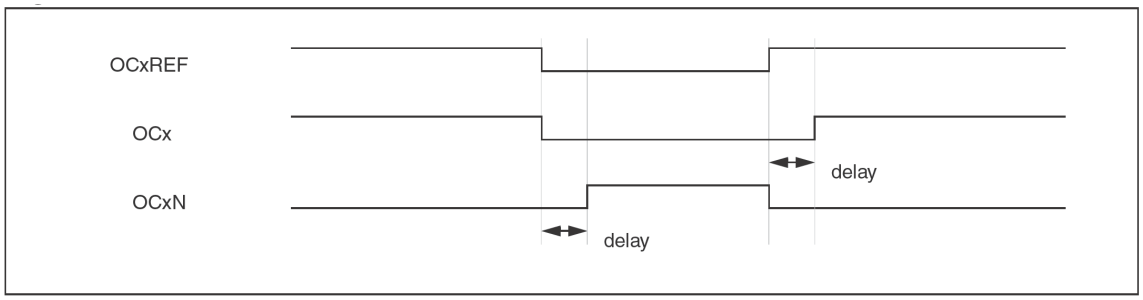


图 173. 死区波形延迟大于负脉冲

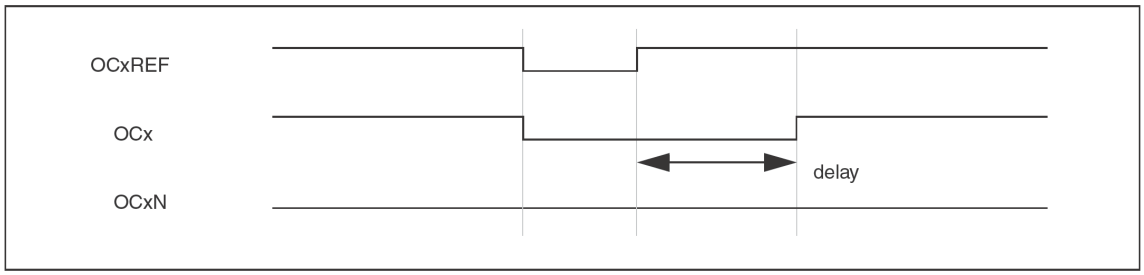
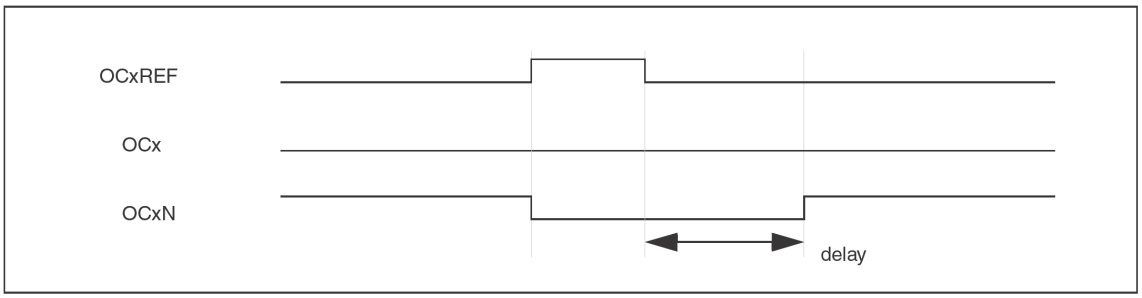


图 174. 死区波形延迟大于正脉冲



每一个通道的死区延时都是相同的，是由 TIMx_BDTR 寄存器中的 DTG 位编程配置。详见 413 页 18.5.15 节 TIM15 刹车和死区寄存器 (TIMx_BDTR) 中的延时计算。

重定向 OCxREF 到 OCx 或 OCxN

在输出模式下 (强置、输出比较或 PWM)，通过配置 TIMx_CCER 寄存器的 CCxE 和 CCxNE 位，OCxREF 可以被重定向到 OCx 或者 OCxN 的输出。

这个功能可以在互补输出处于无效电平时，在某个输出上送出一个特殊的波形 (例如 PWM 或者静态有效电平)。另一个作用是，让两个输出同时处于无效电平，或处于有效电平和带死区的互补输出。

注：当只使能 OCxN(CCxE=0, CCxNE=1) 时，它不会反相，当 OCxREF 变高时立即有效。例如，如果 CCxNP=0，则 OCxN=OCxREF。另一方面，当 OCx 和 OCxN 都被使能时 (CCxE=CCxNE=1)，当 OCxREF 为高时 OCx 有效；而 OCxN 相反，当 OCxREF 低时 OCxN 变为有效。

18.4.12 使用刹车功能

当使用刹车功能时，依据相应的控制位 (TIMx_BDTR 寄存器中的 MOE 、OSSI 和 OSSR 位，TIMx_CR2 寄存器中的 OISx 和 OISxN 位)，输出使能信号和无效电平都会被修改。但无论何时，OCx 和 OCxN 输出不能在同一时间同时处于有效电平上。详见 410 页表 53 带刹车功能的互补输出通道 OCx 和 OCxN 的控制位。

刹车源既可以是刹车输入引脚又可以是一个时钟失败事件。时钟失败事件由复位时钟控制器中的时钟安全系统 (CSS) 产生，详见 7.2.7 节时钟安全系统。

系统复位后，刹车电路被禁止，MOE 位为低。设置 TIMx_BDTR 寄存器中的 BKE 位可以使能刹车功能，刹车输入信号的极性可以通过配置同一个寄存器中的 BKP 位选择。BKE 和 BKP 可以同时被修改。当写入 BKE 和 BKP 位时，在真正写入之前会有 1 个 APB 时钟周期的延迟，因此需要等待一个 APB 时钟周期之后，才能正确地读回写入的位。

因为 MOE 下降沿可以是异步的，在实际信号 (作用在输出端) 和同步控制位 (在 TIMx_BDTR 寄存器中) 之间设置了一个再同步电路。这个再同步电路会在异步信号和同步信号之间产生延迟。特别的，如果当它为低时写 MOE=1，则读出它之前必须先插入一个延时 (空指令) 才能读到正确的值。这是因为写入的是异步信号而读的是同步信号。

当发生刹车时 (在刹车输入端出现选定的电平)，有下述动作：

- MOE 位被异步地清除，将输出置于无效状态、空闲状态或者复位状态 (由 OSSI 位选择)。这个特性在 MCU 的振荡器关闭时依然有效。
- 一旦 MOE=0，每一个输出通道输出由 TIMx_CR2 寄存器中的 OISx 位设定的电平。如果 OSSI=0，则定时器释放使能输出，否则使能输出始终为高。
- 当使用互补输出时：
  - 输出首先被置于复位状态即无效的状态 (取决于极性)。这是异步操作，即使定时器没有时钟时，此功能也有效。
  - 如果定时器的时钟依然存在，死区生成器将会重新生效，在死区之后根据 OISx 和 OISxN 位指示的电平驱动输出端口。即使在这种情况下，OCx 和 OCxN 也不能被同时驱动到有效的电平。注，因为重新同步 MOE，死区时间比通常情况下长一些 (大约 2 个 ck_tim 的时钟周期)。
  - 如果 OSSI=0，定时器释放使能输出，否则保持使能输出；或一旦 CCxE 与 CCxNE 之一变高时，使能输出变为高。
- 如果设置了 TIMx_DIER 寄存器中的 BIE 位，当刹车状态标志 (TIMx_SR 寄存器中的 BIF 位) 为’1’时，则产生一个中断。如果设置了 TIMx_DIER 寄存器中的 BDE 位，则产生一个 DMA 请求。
- 如果设置了 TIMx_BDTR 寄存器中的 AOE 位，在下一个更新事件 UEV 时 MOE 位被自动置位； 例如，这可以用来进行整形。否则，MOE 始终保持低直到被再次置’1’；此时，这个特性可以被用在安全方面，你可以把刹车输入连到电源驱动的报警输出、热敏传感器或者其他安全器件上。

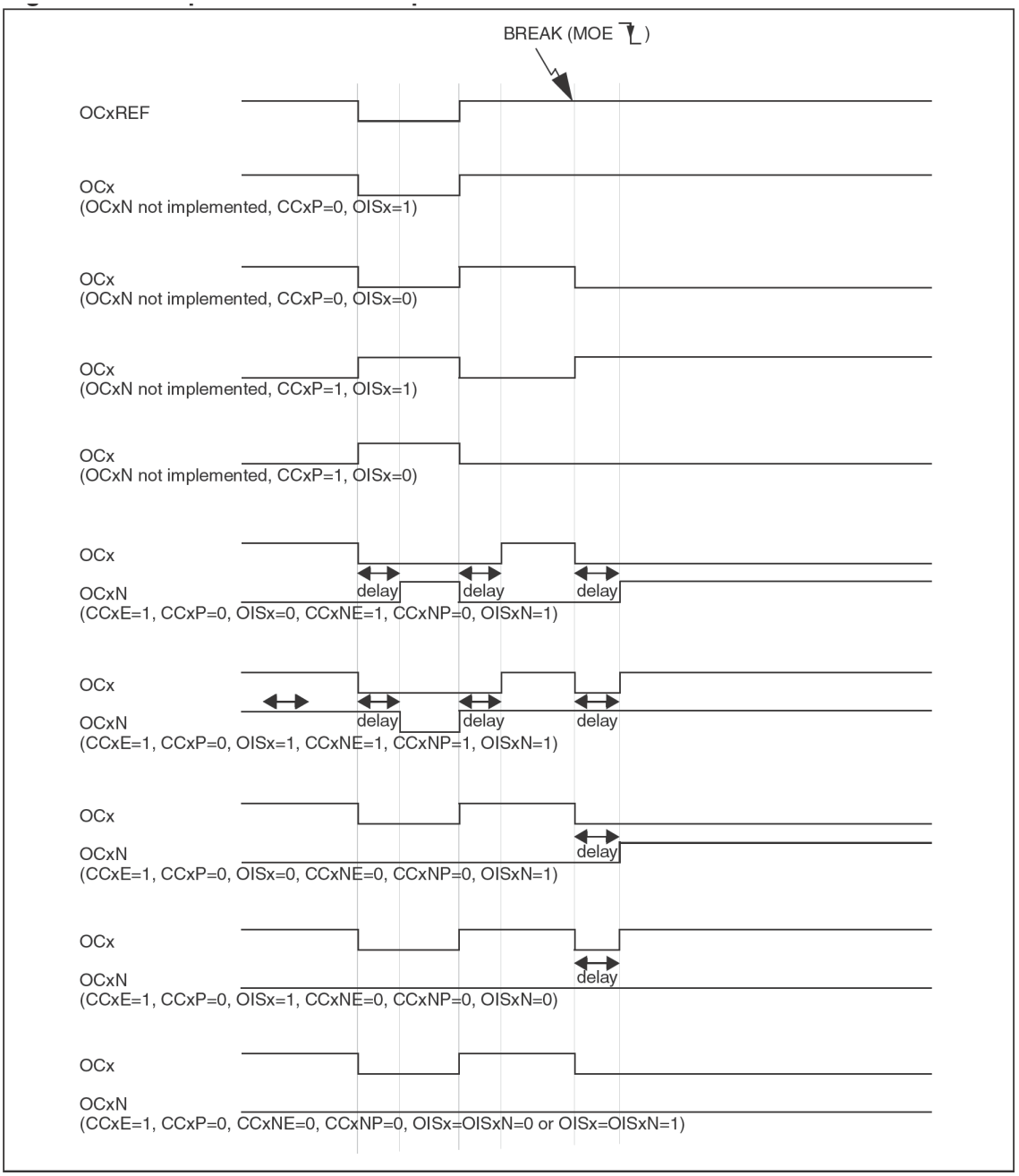
注：刹车输入为电平有效。所以，当刹车输入有效时，不能同时 (自动地或者通过软件) 设置 MOE。同时，状态标志 BIF 不能被清除。

刹车由 BRK 输入产生，它的有效极性是可编程的，且由 TIMx_BDTR 寄存器中的 BKE 位开启。

除了刹车输入和输出管理，刹车电路中还实现了写保护以保证应用程序的安全。它允许用户冻结几个配置参数 (死区长度，OCx/OCxN 极性和被禁止的状态，OCxM 配置，刹车使能和极性)。用户可以通过 TIMx_BDTR 寄存器中的 LOCK 位，从三级保护中选择一种，参看 413 页 18.5.15 节 TIM15 刹车和死区寄存器 (TIMx_BDTR)。在 MCU 复位后 LOCK 位只能被修改一次。

下图显示响应刹车的输出实例。

图 175. 响应刹车的输出

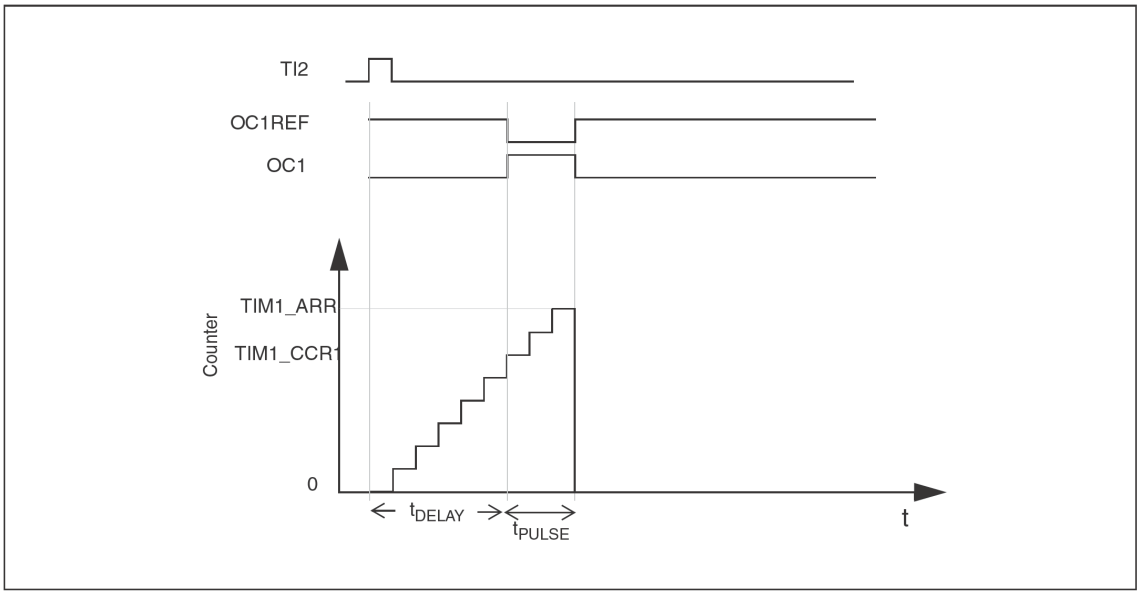


18.4.13 单脉冲模式

单脉冲模式 (OPM) 是前述众多模式的一个特例。这种模式允许计数器响应一个激励，并在一个程序可控的延时之后，产生一个脉宽可程序控制的脉冲。
可以通过从模式控制器启动计数器，在输出比较模式或者 PWM 模式下产生波形。设置 TIMx_CR1 寄存器中的 OPM 位将选择单脉冲模式，这样可以让计数器自动地在产生下一个更新事件 UEV 时停止。
仅当比较值与计数器的初始值不同时，才能产生一个脉冲。启动之前 (当定时器正在等待触发)，必须如下配置：

- 向上计数方式：CNT < CCRx ≤ ARR (特别地，0 < CCRx)，
- 向下计数方式：CNT > CCRx。

图 176. 单脉冲模式的例子



例如，你需要在从 TI2 输入脚上检测到一个上升沿开始，延迟 tDELAY 之后，在 OC1 上产生一个长度为 tPULSE 的正脉冲。

假定 TI2FP2 作为触发 1:

- 置 TIMx_CCMR1 寄存器中的 CC2S=’01’，把 TI2FP2 映像到 TI2。
- 置 TIMx_CCER 寄存器中的 CC2P=’0’，使 TI2FP2 能够检测上升沿。
- 置 TIMx_SMCR 寄存器中的 TS=’110’，TI2FP2 作为从模式控制器的触发 (TRGI)。
- 置 TIMx_SMCR 寄存器中的 SMS=’110’(触发模式)，TI2FP2 被用来启动计数器。

OPM 波形由写入比较寄存器的数值决定 (要考虑时钟频率和计数器预分频器)

- tDELAY 由写入 TIMx_CCR1 寄存器中的值定义。
- tPULSE 由自动装载值和比较值之间的差值定义 (TIMx_ARR - TIMx_CCR1)。
- 假定当发生比较匹配时要产生从’0’到’1’的波形，当计数器到达预装载值时要产生一个从’1’ 到’0’的波形；首先要置 TIMx_CCMR1 寄存器的 OC1M=’111’，进入 PWM 模式 2；根据需要有选择地使能预装载寄存器：置 TIMx_CCMR1 中的 OC1PE=’1’和 TIMx_CR1 寄存器中的 ARPE；然后在 TIMx_CCR1 寄存器中填写比较值，在 TIMx_ARR 寄存器中填写自动装载值，修改 UG 位来产生一个更新事件，然后等待在 TI2 上的一个外部触发事件。本例中，CC1P=’0’。

在这个例子中，TIMx_CR1 寄存器中的 DIR 和 CMS 位应该置低。
因为只需一个脉冲，所以必须设置 TIMx_CR1 寄存器中的 OPM=’1’，在下一个更新事件 (当计数器从自动装载值翻转到 0) 时停止计数。设置 TIMx_CR1 寄存器中的 OPM=’0’，就会选择重复模式。

特殊情况：OCx 快速使能：

在单脉冲模式下，在 TIx 输入脚的边沿检测逻辑设置 CEN 位以启动计数器。然后计数器和比较值间的比较操作产生了输出的转换。但是这些操作需要一定的时钟周期，因此它限制了可得到的最小延时 tDELAY。
如果要以最小延时输出波形，可以设置 TIMx_CCMRx 寄存器中的 OCxFE 位；此时 OCxREF(和 OCx) 被强制响应激励而不再依赖比较的结果，输出的波形与比较匹配时的波形一样。OCxFE 只在通道配置为 PWM1 和 PWM2 模式时起作用。

18.4.14 TIM15 外部触发的同步

TIM15 定时器能够在多种模式下和一个外部的触发同步：复位模式、门控模式和触发模式。

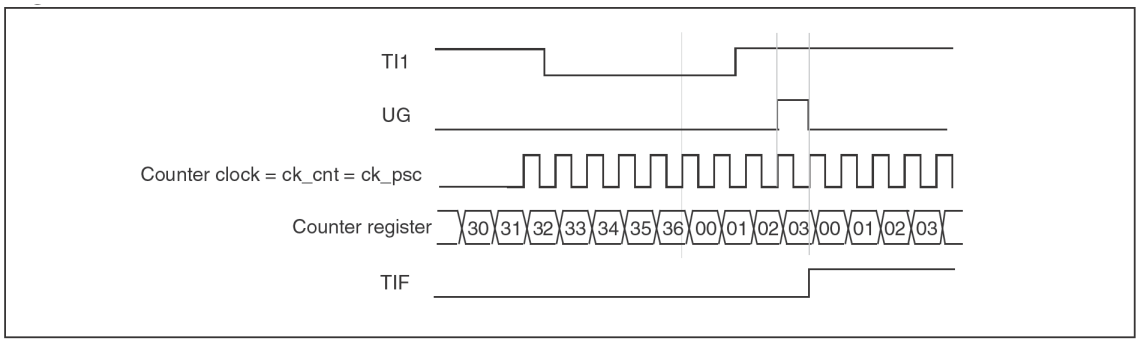
从模式：复位模式
在一个触发输入发生事件时，计数器和它的预分频器能够重新被初始化；同时，如果 TIMx_CR1 寄存器的 URS 位为低，还产生一个更新事件 UEV ；然后所有的预装载寄存器 (TIMx_ARR ，TIMx_CCRx) 都被更新。

在以下的例子中，TI1 输入端的上升沿导致向上计数器被清零：

- 配置通道 1 以检测 TI1 的上升沿。配置输入滤波器的带宽 (在本例中，不需要任何滤波器，因此保持 IC1F=0000)。触发操作中不使用捕获预分频器，所以不需要配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中 CC1S=01。置 TIMx_CCER 寄存器中 CC1P=0 以确定极性 (只检测上升沿)。
- 置 TIMx_SMCR 寄存器中 SMS=100，配置定时器为复位模式；置 TIMx_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。
- 置 TIMx_CR1 寄存器中 CEN=1，启动计数器。

计数器开始依据内部时钟计数，然后正常运转直到 TI1 出现一个上升沿；此时，计数器被清零然后从 0 重新开始计数。同时，触发标志 (TIMx_SR 寄存器中的 TIF 位) 被设置，并根据 TIMx_DIER 寄存器中 TIE(中断使能) 位和 TDE(DMA 使能) 位的设置，产生一个中断请求或一个 DMA 请求。
下图显示当自动重装载寄存器 TIMx_ARR=0x36 时的动作。在 TI1 上升沿和计数器的实际复位之间的延时取决于 TI1 输入端的重同步电路。

图 177. 复位模式下的控制电路



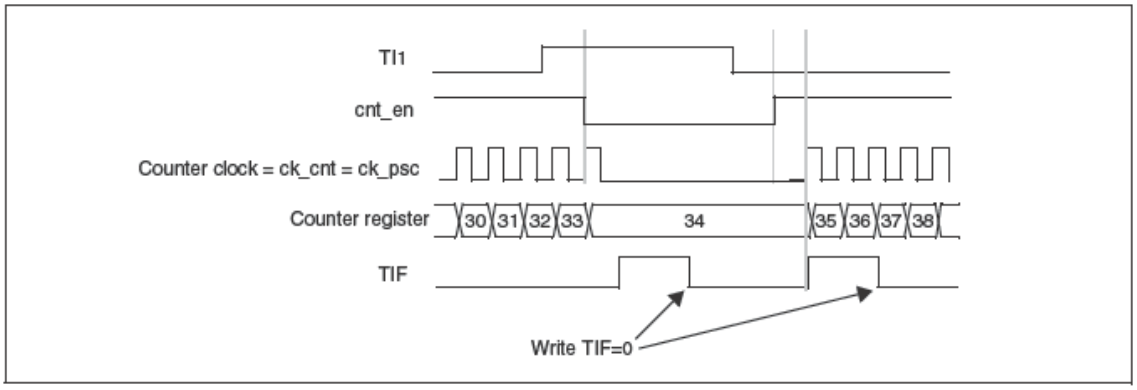
从模式：门控模式
按照选中的输入端电平控制计数器。

在如下的例子中，计数器只在 TI1 为低时向上计数：

- 配置通道 1 以检测 TI1 上的低电平。配置输入滤波器带宽 (本例中，不需要滤波，所以保持 IC1F=0000)。触发操作中不使用捕获预分频器，所以不需要配置。CC1S 位用于选择输入捕获源，置 TIMx_CCMR1 寄存器中 CC1S=01。置 TIMx_CCER 寄存器中 CC1P=1 以确定极性 (只检测低电平)。
- 置 TIMx_SMCR 寄存器中 SMS=101，配置定时器为门控模式；置 TIMx_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。
- 置 TIMx_CR1 寄存器中 CEN=1，启动计数器。（在门控模式下，如果 CEN=0，则计数器不能启动，不论触发输入电平如何）

只要 TI1 为低，计数器开始依据内部时钟计数，一旦 TI1 变高则停止计数。当计数器开始或停止时都设置 TIMx_SR 中的 TIF 标置。
TI1 上升沿和计数器实际停止之间的延时取决于 TI1 输入端的重同步电路。

图 178. 门控模式下的控制电路



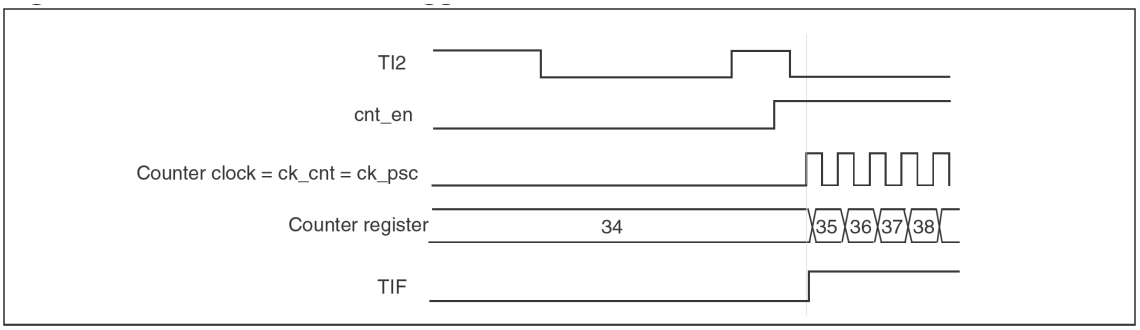
从模式：触发模式
输入端上选中的事件启动计数器。

在下面的例子中，计数器在 TI2 输入的上升沿开始向上计数：

- 配置通道 2 检测 TI2 的上升沿。配置输入滤波器带宽 (本例中，不需要任何滤波器，保持 IC2F=0000)。触发操作中不使用捕获预分频器，不需要配置。CC2S 位只用于选择输入捕获源，置 TIMx_CCMR1 寄存器中 CC2S=01。置 TIMx_CCER 寄存器中 CC2P=1 以确定极性 (只检测低电平)。
- 置 TIMx_SMCR 寄存器中 SMS=110，配置定时器为触发模式；置 TIMx_SMCR 寄存器中 TS=110，选择 TI2 作为输入源。

当 TI2 出现一个上升沿时，计数器按内部时钟开始计数，同时设置 TIF 标志。
TI2 上升沿和计数器启动计数之间的延时，取决于 TI2 输入端的重同步电路。

图 179. 触发器模式下的控制电路



TIM 定时器在内部相连，用于定时器同步或链接。详见 319 页 16.3.15 节，定时器同步。

18.4.15 定时器同步 (TIM15)

TIM 定时器在内部相连，用于定时器同步或链接。详见 308 页 13.3.15 节，定时器同步。

18.4.16 调试模式

当微控制器进入调试模式时 (Cortex-M0 核心停止)，根据 DBG 模块中 DBG_TIMx_STOP 的设置，TIMx 计数器可以或者继续正常操作，或者停止。

18.5 TIM15 寄存器描述

关于在寄存器描述里面所用到的缩写，参见 31 页第 1.1 章。

18.5.1 TIM15 控制寄存器 1 (TIM15_CR1)

偏移地址 : 0x00

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CKD[1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
						rw	rw	rw				rw	rw	rw	rw

Bits 15:10 保留，读始终为‘0’.

Bits 9:8 CKD[1:0]: 时钟分频因子 (Clock division)

定义在定时器时钟 (CK_INT) 频率与死区时间、用于死区时间发生器和数字滤波器 (ETR，TIx) 的采样时钟（tDTS）之间的分频比例。

00：tDTS = tCK_INT

01：tDTS = 2 x tCK_INT

10：tDTS = 4 x tCK_INT

11：保留，不可编程此值

Bit 7 ARPE: 自动重装载预装载允许位 (Auto-reload preload enable)

0：TIMx_ARR 寄存器没有缓冲；

1：TIMx_ARR 寄存器缓冲器有效。

Bits 6:4 保留，读始终为‘0’.

Bit 3 OPM: 单脉冲模式 (One pulse mode)

0：在发生更新事件时，计数器不停止；

1：在发生下一次更新事件 (清除 CEN 位) 时，计数器停止

Bit 2 URS: 更新请求源 (Update request source)

软件通过该位选择 UEV 事件的源

0：如果使能了更新中断请求，则下述任一事件产生更新中断请求：

— 计数器溢出 / 下溢

— 设置 UG 位

— 从模式控制器产生的更新

1：如果使能了更新中断请求，则只有计数器溢出 / 下溢才产生更新中断请求。

Bit 1 UDIS: 禁止更新 (Update disable)

软件通过该位允许 / 禁止 UEV 事件的产生

0：允许 UEV。更新 (UEV) 事件由下述任一事件产生：

— 计数器溢出 / 下溢

— 设置 UG 位

— 从模式控制器产生的更新具有缓存的寄存器被装入它们的预装载值。

1：禁止 UEV。不产生更新事件，影子寄存器 (ARR、PSC、CCRx) 保持它们的值。如果设置了 UG 位或从模式控制器发出了一个硬件复位，则计数器和预分频器被重新初始化。

Bit 0 CEN: 使能计数器（Counter enable）

0：禁止计数器；

1：使能计数器。

注：在软件设置了 CEN 位后，外部时钟和门控模式模式才能工作。触发模式可以自动地通过硬件设置 CEN 位。

18.5.2 TIM15 控制寄存器 2 (TIM15_CR2)

偏移地址 : 0x04

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	OIS2	OIS1N	OIS1	Res.	MMS[2:0]			CCDS	CCUS	Res.	CCPC
					rw	rw	rw		rw	rw	rw	rw	rw		rw

Bit 15:11 保留，读始终为‘0’.

Bit 10 OIS2: 输出空闲状态 2 (OC2 输出)

0: OC2=0 ， 当 MOE=0

1: OC2=1 ， 当 MOE=0

注： 一旦通过 TIMx_BKR 寄存器的 LOCK 编程设置了 LOCK 等级 1，2 或 3，此位不可再被修改。

Bit 9 OIS1N: 输出空闲状态 1 (OC1N 输出)

0: OC1N=0 ， 当 MOE=0 后一个死区时间

1: OC1N=1 ， 当 MOE=0 后一个死区时间

注： 一旦通过 TIMx_BKR 寄存器的 LOCK 编程设置了 LOCK 等级 1，2 或 3，此位不可再被修改。

Bit 8 OIS1: 输出空闲状态 1 (OC1 输出)

0: OC1=0 ， 当 MOE=0（如果 OC1N 被使用还要等一个死区时间）

1: OC1=1 ， 当 MOE=0（如果 OC1N 被使用还要等一个死区时间）

注： 一旦通过 TIMx_BKR 寄存器的 LOCK 编程设置了 LOCK 等级 1，2 或 3，此位不可再被修改。

Bit 7 保留，读始终为‘0’

Bits 6:4 MMS[1:0]: 主模式选择 (Master mode selection)

这 3 位用于选择在主模式下送到从定时器的同步信息 (TRGO)。可能的组合如下：

000：复位 – TIMx_EGR 寄存器的 UG 位被用于作为触发输出 (TRGO)。如果是触发输入产生的复位 (从模式控制器处于复位模式)，则 TRGO 上的信号相对实际的复位会有一个延迟。

001：使能 – 计数器使能信号 CNT_EN 被用于作为触发输出 (TRGO)。有时需要在同一时间启动多个定时器或控制在一段时间内使能从定时器。计数器使能信号是通过 CEN 控制位和门控模式下的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时，TRGO 上会有一个延迟，除非选择了主 / 从模式 (见 TIMx_SMCR 寄存器中 MSM 位的描述)。

010：更新 – 更新事件被选为触发输入 (TRGO)。例如，一个主定时器的时钟可以被用作一个从定时器的预分频器。

011：比较脉冲 – 在发生一次捕获或一次比较成功时，当要设置 CC1IF 标志时 (即使它已经为 高)，触发输出送出一个正脉冲 (TRGO)。

100：比较 – OC1REF 信号被用于作为触发输出 (TRGO)。

101：比较 – OC2REF 信号被用于作为触发输出 (TRGO)。

Bit 3 CCDS: 捕获 / 比较的 DMA 选择（Capture/compare DMA selection）

0：当发生 CCx 事件时，送出 CCx 的 DMA 请求；

1：当发生更新事件时，送出 CCx 的 DMA 请求。

Bit 2 CCUS: 捕捉 / 比较控制更新选择（Capture/compare control update selection）

0: 当捕捉 / 比较控制位被 预装载时 (CCPC=1)，只有通过设置 COMG 位才会被 更新

1: 当捕捉 / 比较控制位被 预装载时 (CCPC=1)，在设置 COMG 位或在 TRGI 上产生上升沿时都会被更新

注：此位仅汉通道有互补输出时才起作用

Bit 1 保留，读始终为‘0’.

Bit 0 CCPC: 捕捉 / 比较预装载控制（Capture/compare preloaded control）

0: CCxE, CCxNE 和 OCxM 位不进行预装载

1: CCxE, CCxNE 和 OCxM 位在被写入后被预装载，只有在 COM 位被设置为‘1’时才更新

注：此位仅汉通道有互补输出时才起作用

18.5.3 TIM15 从模式控制寄存器 (TIM15_SMCR)

偏移地址 : 0x08

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MSM	TS[2:0]			Res.	SMS[2:0]		
								rw	rw	rw	rw		rw	rw	rw

Bits 15:8 保留，读始终为‘0’.

Bit 7 MSM: 主 / 从模式 (Master/slave mode)

0：无作用；

1：触发输入 (TRGI) 上的事件被延迟了，以允许在当前定时器 (通过 TRGO) 与它的从定时器间的完美同步。这对要求把几个定时器同步到一个单一的外部事件时是非常有用的。

Bits 6:4 TS[2:0]: 触发选择 (Trigger selection)

这 3 位选择用于同步计数器的触发输入。

000：内部触发 0(ITR0)　　100：TI1 的边沿检测器 (TI1F_ED)

001：内部触发 1(ITR1)　　101：滤波后的定时器输入 1(TI1FP1)

010：内部触发 2(ITR2)　　110：滤波后的定时器输入 2(TI2FP2)

011：内部触发 3(ITR3)

关于每个定时器中 ITRx 的细节，参见页 401 表 52，TIMx 内部触发连接。

注：这些位只能在未用到 (如 SMS=000) 时被改变，以避免在改变时产生错误的边沿检测。

Bit 3　保留，读始终为‘0’

Bits 2:0 SMS: 从模式选择 (Slave mode selection)

当选择了外部信号，触发信号 (TRGI) 的有效边沿与选中的外部输入极性相关 (见输入控制寄存器 和控制寄存器的说明)

000：关闭从模式 – 如果 CEN=1，则预分频器直接由内部时钟驱动。

001：编码器模式 1 – 根据 TI1FP1 的电平，计数器在 TI2FP2 的边沿向上 / 下计数。

010：编码器模式 2 – 根据 TI2FP2 的电平，计数器在 TI1FP1 的边沿向上 / 下计数。

011：编码器模式 3 – 根据另一个信号的输入电平，计数器在 TI1FP1 和 TI2FP2 的边沿向上 / 下计数。

100：复位模式 – 选中的触发输入 (TRGI) 的上升沿重新初始化计数器，并且产生一个更新寄存器的信号。

101：门控模式 – 当触发输入 (TRGI) 为高时，计数器的时钟开启。一旦触发输入变为低，则计数器停止 (但不复位)。计数器的启动和停止都是受控的。

110：触发模式 – 计数器在触发输入 TRGI 的上升沿启动 (但不复位)，只有计数器的启动是受控的。

111：外部时钟模式 1 – 选中的触发输入 (TRGI) 的上升沿驱动计数器。

注：如果 TI1F_ED 被选为触发输入 (TS=100) 时，不要使用门控模式。这是因为，TI1F_ED 在每次 TI1F 变化时输出一个脉冲，然而门控模式是要检查触发输入的电平。

表 52.　TIMx 内部触发连接

Slave TIM	ITR0 (TS = 000)	ITR1 (TS = 001)	ITR2 (TS = 010)	ITR3 (TS = 011)
TIM15	TIM2	TIM3	TIM16	TIM17

18.5.4 TIM15 DMA/ 中断使能寄存器 (TIM15_DIER)

偏移地址 : 0x0C

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	Res.	Res.	Res.	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	Res.	Res.	CC2IE	CC1IE	UIE
	rw				rw	rw	rw	rw	rw	rw			rw	rw	rw

Bit 15 保留，读始终为‘0’

Bit 14 TDE: 触发 DMA 请求允许

0: 触发 DMA 请求禁止 .

1: 触发 DMA 请求允许

Bit 13:11 保留，读始终为‘0’

Bit 10 CC2DE: 捕捉 / 比较 2 DMA 请求允许

0: CC2 DMA 请求禁止 .

1: CC2 DMA 请求允许

Bit 9 CC1DE: 捕捉 / 比较 1 DMA 请求允许

0: CC1 DMA 请求禁止 .

1: CC1 DMA 请求允许

Bit 8 UDE: 更新 DMA 请求允许

0: 更新 DMA 请求禁止 .

1: 更新 DMA 请求允许

Bit 7 BIE: 刹车中断允许

0: 刹车中断禁止 .

1: 刹车中断允许

Bit 6 TIE: 触发中断允许

0: 触发中断禁止 .

1: 触发中断允许

Bit 5 COMIE: COM 中断允许

0: COM 中断禁止 .

1: COM 中断允许

Bit 4:3 保留，始终读为‘0’

Bit 2 CC2IE: 捕捉 / 比较 2 中断允许

0: CC2 中断禁止

1: CC2 中断允许

Bit 1 CC1IE: 捕捉 / 比较 1 中断允许

0: CC1 中断禁止

1: CC1 中断允许

Bit 0 UIE: 更新中断允许

0: 更新中断禁止

1: 更新中断允许

18.5.5 TIM15 状态寄存器 (TIM15_SR)

偏移地址 : 0x10

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CC2OF	CC1OF	Res.	BIF	TIF	COMIF	Res.	Res.	CC2IF	CC1IF	UIF
					rc_w0	rc_w0		rc_w0	rc_w0				rc_w0	rc_w0	rc_w0

Bits 15:10 保留，读始终为‘0’.

Bit 9 CC1OF: 捕捉 / 比较 1 重复捕捉标志 (Capture/Compare 1 overcapture flag)

仅当相应的通道被配置为输入捕获时，该标记可由硬件置’1’。写’0’可清除该位。

0：无重复捕获产生；

1：当计数器的值被捕获到 TIMx_CCR1 寄存器时，CC1IF 的状态已经为’1’。

Bit 8 保留，读始终为‘0’.

Bit 7 BIF: 刹车中断标记 (Trigger interrupt flag)

当刹车输入信号有效时由硬件对该位置’1’。刹车信号无效时可由软件清’0’。

0：无刹车事件产生；

1：刹车输入端检测到有效电平。

Bit 6 TIF: 触发器中断标记 (Trigger interrupt flag)

当发生触发事件 (当从模式控制器处于除门控模式外的其它模式时，在 TRGI 输入端检测到有效边沿，或门控模式下的任一边沿) 时由硬件对该位置’1’。它由软件清’0’。

0：无触发器事件产生；

1：触发器中断等待响应。

Bit 5 COMIF:COM 中断标记 (COM interrupt flag)

当发生 COM 事件 (一旦捕捉 / 比较控制位——CCxE, CCxNE, OCxM，被 更新) 时由硬件对该位置’1’。它由软件清’0’。

0：无 COM 事件产生；

1：COM 中断等待响应。

Bits 4:3 保留，读始终为‘0’.

Bit 2 CC2IF: 捕捉 / 比较 2 中断标志

参见 CC1IF 描述

Bit 1 CC1IF: 捕捉 / 比较 1 中断标志 (Capture/Compare 1 interrupt flag)

如果通道 CC1 配置为输出模式：
当计数器值与比较值匹配时该位由硬件置’1’，但在中心对称模式下除外 (参考 TIMx_CR1 寄存器的 CMS 位)。它由软件清’0’。
0：无匹配发生；
1：TIMx_CNT 的值与 TIMx_CCR1 的值匹配。
当 TIMx_CCR1 的内容大于 TIMx_ARR 时，在计数器溢出（向上或向上 / 向下计数模式）或计数器下溢出时（向下计数模式）时 CC1IF 位变高

如果通道 CC1 配置为输入模式：
当捕获事件发生时该位由硬件置’1’，它由软件清’0’或通过读 TIMx_CCR1 清’0’。
0：无输入捕获产生；
1：计数器值已被捕获 (拷贝) 至 TIMx_CCR1(在 IC1 上检测到与所选极性相同的边沿)。

Bit 0 UIF: 更新中断标记 (Update interrupt flag)
当产生更新事件时该位由硬件置’1’。它由软件清’0’。
0：无更新事件产生；
1：更新中断等待响应。
当寄存器被更新时该位由硬件置’1’：
- 若 TIMx_CR1 寄存器的 UDIS=0，当对应的重复计数器的值溢出（更新如果重复计数器＝ 0）时产生更新事件；
- 若 TIMx_CR1 寄存器的 UDIS=0、URS=0，当通过软件对 TIMx_EGR 寄存器的 UG 对计数器 CNT 重新初始化时产生更新事件；
- 若 TIMx_CR1 寄存器的 UDIS=0、URS=0，当计数器 CNT 被触发事件（参见 18.5.3：TIM15 从模式控制寄存器 TIM15_SMCR 的描述）重初始化时产生

18.5.6 TIM15 事件产生寄存器 (TIM15_EGR)

偏移地址 : 0x14
复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BG	TG	COMG	Res.	Res.	CC2G	CC1G	UG
								w	w	rw			w	w	w

Bits 15:8 保留，读始终为‘0’.

Bit 7 BG: 刹车产生（Break generation）
该位由软件置’1’，用于产生一个刹车事件，由硬件自动清’0’。
0：无动作；
1：刹车事件产生，清除 MOET，设置 TIMx_SR 寄存器的 TIF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。

Bit 6 TG: 产生触发事件 (Trigger generation)
该位由软件置’1’，用于产生一个触发事件，由硬件自动清’0’。
0：无动作；
1：TIMx_SR 寄存器的 TIF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。

Bit 5 COMG: 捕捉 / 比较控制更新产生（Capture/Compare control update generation）
该位由软件置’1’，由硬件自动清’0’。
0：无动作；
1: 当 CCPC 位被设置时，才可能更新 CCxE, CCxNE 和 OCxM 位
注：此位仅当通道有互补输出时才起作用

Bits 4:3 保留，读始终为‘0’

Bit 2 CC2G: 捕捉 / 比较 2 产生
参见 CC1G 描述

Bit 1 CC1G: 捕捉 / 比较 1 产生
该位由软件置’1’，用于产生一个捕获 / 比较事件，由硬件自动清’0’。
0：无动作；
1：在通道 CC1 上产生一个捕获 / 比较事件：
若通道 CC1 配置为输出：
设置 CC1IF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。
若通道 CC1 配置为输入：
当前的计数器值捕获至 TIMx_CCR1 寄存器；设置 CC1IF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。若 CC1IF 已经为 1，则设置 CC1OF=1。

Bit 0 UG: 产生更新事件 (Update generation)
该位由软件置’1’，由硬件自动清’0’。
0：无动作；
1：重新初始化计数器，并产生一个更新事件。注意预分频器的计数器也被清’0’(但是预分频系数不变)。若在中心对称模式下或 DIR=0(向上计数) 则计数器被清’0’，若 DIR=1(向下计数) 则计数器取 TIMx_ARR 的值。

18.5.7 TIM15 捕捉 / 比较模式寄存器 1 (TIM15_CCMR1)

偏移地址 : 0x18
复位值 : 0x0000

通道可用于输入 (捕获模式) 或输出 (比较模式)，通道的方向由相应的 CCxS 定义。该寄存器其它位的作用在输入和输出模式下不同。OCxx 描述了通道在输出模式下的功能，ICxx 描述了通道在输入模式下的功能。因此必须注意，同一个位在输出模式和输入模式下的功能是不同的。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OC2M[2:0]			OC2 PE	OC2 FE	CC2S[1:0]		Res.	OC1M[2:0]			OC1 PE	OC1 FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]				IC1PSC[1:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

输出比较模式

Bit 15 保留，读始终为‘0’.
Bits 14:12 OC2M[2:0]: 输出比较 2 模式
Bit 11 OC2PE: 输出比较 2 预装载使能
Bit 10 OC2FE: 输出比较 2 快速使能
Bits 9:8 CC2S[1:0]: 捕捉 / 比较 2 选择 (Capture/Compare 2 selection)
该位定义通道的方向 (输入 / 输出)，及输入脚的选择：
00：CC2 通道被配置为输出；
01：CC2 通道被配置为输入，IC2 映射在 TI2 上；
10：CC2 通道被配置为输入，IC2 映射在 TI1 上；
11：CC2 通道被配置为输入，IC2 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)。
注：CC2S 仅在通道关闭时 (TIMx_CCER 寄存器的 CC2E=’0’) 才是可写的。
Bit 7 保留，读始终为‘0’.
Bits 6:4 OC1M: 输出比较 1 模式 (Output compare 1 mode)
该 3 位定义了输出参考信号 OC1REF 的动作，而 OC1REF 决定了 OC1 和 OC1N 的值。OC1REF 是高电平有效，而 OC1 和 OC1N 的有效电平取决于 CC1P 和 CC1PN 位。
000：冻结。输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 间的比较对 OC1REF 不起作用；（此模式仅用于产生时基）
001 ：匹配时设置通道 1 为有效电平。 当计数器 TIMx_CNT 的值与捕获 / 比较寄存器 1 (TIMx_CCR1) 相同时，强制 OC1REF 为高。
010 ：匹配时设置通道 1 为无效电平。 当计数器 TIMx_CNT 的值与捕获 / 比较寄存器 1 (TIMx_CCR1) 相同时，强制 OC1REF 为低。
011：翻转。当 TIMx_CCR1=TIMx_CNT 时，翻转 OC1REF 的电平。
100：强制为无效电平。强制 OC1REF 为低。
101：强制为有效电平。强制 OC1REF 为高。
110：PWM 模式 1 － 在向上计数时，一旦 TIMx_CNT<TIMx_CCR1 时通道 1 为有效电平，否则为无效电平；在向下计数时，一旦 TIMx_CNT>TIMx_CCR1 时通道 1 为无效电平 (OC1REF=0)，否 则为有效电平 (OC1REF=1)。
111：PWM 模式 2 － 在向上计数时，一旦 TIMx_CNT<TIMx_CCR1 时通道 1 为无效电平，否则为有效电平；在向下计数时，一旦 TIMx_CNT>TIMx_CCR1 时通道 1 为有效电平，否则为无效电平。
注 1：一旦 LOCK 级别设为 3(TIMx_BDTR 寄存器中的 LOCK 位) 并且 CC1S=’00’ (该通道配置成输出) 则该位不能被修改。
注 2：在 PWM 模式 1 或 PWM 模式 2 中，只有当比较结果改变了或在输出比较模式中从冻结模式 切换到 PWM 模式时，OC1REF 电平才改变。
Bit 3 OC1PE: 输出比较 1 预装载使能 (Output compare 1 preload enable)
0：禁止 TIMx_CCR1 寄存器的预装载功能，可随时写入 TIMx_CCR1 寄存器，并且新写入的数值立即起作用。
1：开启 TIMx_CCR1 寄存器的预装载功能，读写操作仅对预装载寄存器操作，TIMx_CCR1 的预装载值在更新事件到来时被传送至当前寄存器中。
注 1：一旦 LOCK 级别设为 3(TIMx_BDTR 寄存器中的 LOCK 位) 并且 CC1S=’00’ (该通道配置成输出) 则该位不能被修改。
注 2：仅在单脉冲模式下 (TIMx_CR1 寄存器的 OPM=’1’)，可以在未确认预装载寄存器情况下使用 PWM 模式，否则其动作不确定。

Bit 2 OC1FE: 输出比较 1 快速使能 (Output compare 1 fast enable)

该位用于加快 CC 输出对触发器输入事件的响应。

0：根据计数器与 CCR1 的值，CC1 正常操作，即使触发器是打开的。当触发器的输入出现一个有效沿时，激活 CC1 输出的最小延时为 5 个时钟周期。

1：输入到触发器的有效沿的作用就象发生了一次比较匹配。因此，OC 被设置为比较电平而与比较结果无关。采样触发器的有效沿和 CC1 输出间的延时被缩短为 3 个时钟周期。 该位只在通道被配置成 PWM1 或 PWM2 模式时起作用。

Bits 1:0 CC1S: 捕捉 / 比较 1 选择 (Capture/Compare 1 selection)

这 2 位定义通道的方向 (输入 / 输出)，及输入脚的选择：

00：CC1 通道被配置为输出；

01：CC1 通道被配置为输入，IC1 映射在 TI1 上；

10：CC1 通道被配置为输入，IC1 映射在 TI2 上；

11：CC1 通道被配置为输入，IC1 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)。

注：CC1S 仅在通道关闭时 (TIMx_CCER 寄存器的 CC1E=’0’) 才是可写的。

输入捕捉模式

Bits 15:12 IC2F: 输入捕捉 2 滤波器

Bits 11:10 IC2PSC[1:0]: 输入捕捉 2 预分频

Bits 9:8 CC2S: 捕捉 / 比较 2 选择 (Capture/compare 2 selection)

这 2 位定义通道的方向 (输入 / 输出)，及输入脚的选择：

00：CC2 通道被配置为输出；

01：CC2 通道被配置为输入，IC2 映射在 TI2 上；

10：CC2 通道被配置为输入，IC2 映射在 TI1 上；

11：CC2 通道被配置为输入，IC2 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)。

注：CC2S 仅在通道关闭时 (TIMx_CCER 寄存器的 CC2E=’0’) 才是可写的。

Bits 7:4 IC1F: 输入捕获 1 滤波器 (Input capture 1 filter)

这几位定义了 TI1 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成，它记录到 N 个事件后会产生一个输出的跳变：

0000：无滤波器，以 fDTS 采样

0001：采样频率 fSAMPLING=fCK_INT，N=2

0010：采样频率 fSAMPLING=fCK_INT，N=4

0011：采样频率 fSAMPLING=fCK_INT，N=8

0100：采样频率 fSAMPLING=fDTS/2，N=6

0101：采样频率 fSAMPLING=fDTS/2，N=8

0110：采样频率 fSAMPLING=fDTS/4，N=6

0111：采样频率 fSAMPLING=fDTS/4，N=8

1000：采样频率 fSAMPLING=fDTS/8，N=6

1001：采样频率 fSAMPLING=fDTS/8，N=8

1010：采样频率 fSAMPLING=fDTS/16，N=5

1011：采样频率 fSAMPLING=fDTS/16，N=6

1100：采样频率 fSAMPLING=fDTS/16，N=8

1101：采样频率 fSAMPLING=fDTS/32，N=5

1110：采样频率 fSAMPLING=fDTS/32，N=6

1111：采样频率 fSAMPLING=fDTS/32，N=8

Bits 3:2 IC1PSC: 输入捕捉 1 预分频

这 2 位定义了 CC1 输入 (IC1) 的预分频系数。 一旦 CC1E=’0’(TIMx_CCER 寄存器中)，则预分频器复位。

00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获；

01：每 2 个事件触发一次捕获；

10：每 4 个事件触发一次捕获；

11：每 8 个事件触发一次捕获。

Bits 1:0 CC1S: 捕捉 / 比较 1 选择

这 2 位定义通道的方向 (输入 / 输出)，及输入脚的选择：

00：CC1 通道被配置为输出；

01：CC1 通道被配置为输入，IC1 映射在 TI1 上；

10：CC1 通道被配置为输入，IC1 映射在 TI2 上；

11：CC1 通道被配置为输入，IC1 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)。

注：CC1S 仅在通道关闭时 (TIMx_CCER 寄存器的 CC1E=’0’) 才是可写的。

18.5.8 TIM15 捕捉 / 比较使能 M15_CCER)

偏移地址 : 0x20

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1NP	Res.	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
								rw		rw	rw	rw	rw	rw	rw

Bits 15:8 保留，读始终为‘0’.

Bit 7 CC2NP: 捕捉 / 比较 2 输出极性 .
参考 CC1NP 描述
Bit 6 保留，读始终为‘0’
Bit 5 CC2P: 捕捉 / 比较 2 输出极性 .
参考 CC1P 描述
Bit 4 CC2E: 捕捉 / 比较 2 输出使能 .
参考 CC1E 描述
Bit 3 CC1NP: 捕捉 / 比较 1 输出极性 .
0: OC1N 高有效
1: OC1N 低有效
注：一旦 LOCK 级别设为 2 或 3(TIMx_BDTR 寄存器中的 LOCK 位) 并且 CC1S=’00’(该通道配置成输出) 则该位不能被修改。
Bit 2 CC1NE: 捕捉 / 比较 1 互补输出使能（Capture/Compare 1 complementary output enable）
0: 关闭 - OC1N 无效 .
1: 开启 - OC1N 输出到相关输出引脚的信号取决于 MOE, OSSI, OSSR, OIS1, OIS1N 和 CC1E 位
Bit 1 CC1P: 捕捉 / 比较 1 输出极性 .
通道 CC1 配置为输出 :
0: OC1 高有效
1: OC1 低有效
通道 CC1 配置为输入 :
CC1NP/CC1P 用于选择作为触发或捕获的信号 TI1FP1 和 TI2FP1 的极性
该位 IC1 还是 IC1 的反相信号。
00：不反相 / 上升沿：捕获发生在 TIxFP1 的上升沿（复位、外部时钟或触发模式的捕捉或触发），TIxFP1 不反相（在门控、编码器模式下的触发）。
00：反相 / 下降沿：捕获发生在 TIxFP1 的下降沿（复位、外部时钟或触发模式的捕捉或触发），TIxFP1 反相（在门控、编码器模式下的触发）。
10：保留，不使用此配置
11：不反相 / 上升和下降沿：捕获发生在 TIxFP1 的上升沿和下降沿（复位、外部时钟或触发模式的捕捉或触发），TIxFP1 不反相（门控模式的触发）。此配置不能用于编码器模式。
注：一旦 LOCK 级别设为 2 或 3(TIMx_BDTR 寄存器中的 LOCK 位) 并且 CC1S=’00’(该通道配置成输出) 则该位不能被修改。
Bit 0 CC1E: 捕捉 / 比较 1 输出使能 .
CC1 通道配置为输出：
0： 关闭－ OC1 禁止输出。
1： 开启－ 输出到对应的输出引脚的 OC1 信号取决于 MOE, OSSI, OSSR, OIS1, OIS1N 和 CC1E 位。
CC1 通道配置为输入：
该位决定了计数器的值是否能捕获入 TIMx_CCR1 寄存器。
0：捕获禁止；
1：捕获使能。

表 53. 带刹车功能的互补 OCx 和 OCxN 通道的输出控制位

Control bits					Output states(1)	
MOE bit	OSSI bit	OSSR bit	CCxE bit	CCxNE bit	OCx output state	OCxN output state
1	X	0	0	0	Output Disabled (not driven by the timer) OCx=0, OCx_EN=0	Output Disabled (not driven by the timer) OCxN=0, OCxN_EN=0
		0	0	1	Output Disabled (not driven by the timer) OCx=0, OCx_EN=0	OCxREF + Polarity OCxN=OCxREF xor CCxNP, OCxN_EN=1
		0	1	0	OCxREF + Polarity OCx=OCxREF xor CCxP, OCx_EN=1	Output Disabled (not driven by the timer) OCxN=0, OCxN_EN=0
		0	1	1	OCREF + Polarity + dead-time OCx_EN=1	Complementary to OCREF (not OCREF) + Polarity + dead-time OCxN_EN=1
		1	0	0	Output Disabled (not driven by the timer) OCx=CCxP, OCx_EN=0	Output Disabled (not driven by the timer) OCxN=CCxNP, OCxN_EN=0
		1	0	1	Off-State (output enabled with inactive state) OCx=CCxP, OCx_EN=1	OCxREF + Polarity OCxN=OCxREF xor CCxNP, OCxN_EN=1
		1	1	0	OCxREF + Polarity OCx=OCxREF xor CCxP, OCx_EN=1	Off-State (output enabled with inactive state) OCxN=CCxNP, OCxN_EN=1
		1	1	1	OCREF + Polarity + dead-time OCx_EN=1	Complementary to OCREF (not OCREF) + Polarity + dead-time OCxN_EN=1
0	0	X	0	0	Output Disabled (not driven by the timer) Asynchronously: OCx=CCxP, OCx_EN=0, OCxN=CCxNP, OCxN_EN=0 Then if the clock is present: OCx=OISx and OCxN=OISxN after a dead-time, assuming that OISx and OISxN do not correspond to OCX and OCxN both in active state.	
	0		0	1		
	0		1	0		
	0		1	1		
	1		0	0		
	1		0	1	Off-State (output enabled with inactive state) Asynchronously: OCx=CCxP, OCx_EN=1, OCxN=CCxNP, OCxN_EN=1 Then if the clock is present: OCx=OISx and OCxN=OISxN after a dead-time, assuming that OISx and OISxN do not correspond to OCX and OCxN both in active state	
	1		1	0		
	1		1	1		

1. 当某通道的两个输出都没有使用时 (CCxE = CCxNE = 0), OISx, OISxN, CCxP 和 CCxNP 必有保持清除。

注：连接到互补 OCx 和 OCxN 通道的外部 I/O 引脚状态，取决于 OCx 和 OCxN 通道状态和 GPIO + AFIO 寄存器。

18.5.9 TIM15 计数器 (TIM15_CNT)

偏移地址 : 0x24

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:0 CNT[15:0]: 计数器的值

18.5.10 TIM15 预分频寄存器 (TIM15_PSC)

偏移地址 : 0x28

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:0 PSC[15:0]: 预分寄存器的值

计数器的时钟频率 CK_CNT 等于 fCK_PSC/(PSC[15:0]+1)。 PSC 包含了当更新事件（包括使用 TIMx_EGR 寄存器的 UG 或者当配置为复位模式时通过触发控制器清除计数器）产生时装入当前预分频器寄存器的值。

18.5.11 TIM15 自动重装寄存器 (TIM15_ARR)

偏移地址 : 0x2C

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:0 ARR[15:0]: 自动重装寄存器（Auto-reload value）

ARR 包含了将要传送至实际的自动重装载寄存器的数值。有关 ARR 的更新和动作详细参考 350 页 17.3.1 节：时基单元。

当自动重装载的值为空时，计数器不工作。

18.5.12 TIM15 重复计数寄存器 (TIM15_RCR)

偏移地址 : 0x30

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:8 保留，读始终为‘0’.

Bits 7:0 REP[7:0]: 重复计数器的值 (Repetition counter value)

预装载寄存器被使能后，这些位允许用户设置比较寄存器的更新速率 (即周期性地从预装载寄存器传输到当前寄存器)；如果允许产生更新中断，则会同时影响产生更新中断的速率。

每次向下计数器 REP_CNT 达到 0，会产生一个更新事件并且计数器 REP_CNT 重新从 REP 值开始计数。由于 REP_CNT 只有在周期更新事件 U_RC 发生时才重载 REP 值，因此对 TIMx_RCR 寄存器写入的新值只在下次周期更新事件发生时才起作用。这意味着在 PWM 模式中，(REP+1) 对应着在边沿对齐模式下，PWM 周期的数目；

18.5.13 TIM15 捕捉 / 比较寄存器 1 (TIM15_CCR1)

偏移地址 : 0x34

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:0 CCR1[15:0]: 捕捉 / 比较 1 的值

若 CC1 通道配置为输出：

CCR1 包含了装入当前捕获 / 比较 1 寄存器的值（预装载值）。

如果在 TIMx_CCMR1 寄存器（OC1PE 位）中未选择预装载特性，写入的数值会被立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获 / 比较 1 寄存器中。

当前捕获 / 比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC1 端口上产生输出信号。

若 CC1 通道配置为输入：

CCR1 包含了由上一次输入捕获 1 事件 (IC1) 传输的计数器值。

18.5.14 TIM15 捕捉 / 比较寄存器 2 (TIM15_CCR2)

偏移地址 : 0x38

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:0 CCR2[15:0]: 捕捉 / 比较 2 的值

若 CC2 通道配置为输出：

CCR2 包含了装入当前捕获 / 比较 2 寄存器的值（预装载值）。

如果在 TIMx_CCMR2 寄存器（OC2PE 位）中未选择预装载特性，写入的数值会被立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获 / 比较 2 寄存器中。

当前捕获 / 比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC2 端口上产生输出信号。

若 CC2 通道配置为输入：

CCR2 包含了由上一次输入捕获 2 事件 (IC2) 传输的计数器值。

18.5.15 TIM15 刹车与死区时间寄存器 (TIM15_BDTR)

偏移地址 : 0x44

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

注：根据锁定设置，AOE、BKP、BKE、OSSI、OSSR 和 DTG[7:0] 位均可被写保护，有必要在第一次写入 TIMx_BDTR 寄存器时对它们进行配置。

Bit 15 MOE: 主输出使能 (Main output enable)

一旦刹车输入有效，该位被硬件异步清’0’。根据 AOE 位的设置值，该位可以由软件清’0’或被自动置 1。它仅对配置为输出的通道有效。

0：禁止 OC 和 OCN 输出或强制为空闲状态；

1：如果设置了相应的使能位 (TIMx_CCER 寄存器的 CCxE、CCxNE 位)，则开启 OC 和 OCN 输出。

有关 OC/OCN 使能的细节，参见 408 页 18.5.8 节，TIM15 捕获 / 比较使能寄存器 (TIM15_CCER)。

Bit 14 AOE: 自动输出使能 (Automatic output enable)

0：MOE 只能被软件置’1’；

1：MOE 能被软件置’1’或在下一个更新事件被自动置’1’(如果刹车输入无效)。

注：一旦 LOCK 级别 (TIMx_BDTR 寄存器中的 LOCK 位) 设为’1’，则该位不能被修改。

Bit 13 BKP: 刹车输入极性 (Break polarity)

0：刹车输入低电平有效；

1：刹车输入高电平有效。

注：一旦 LOCK 级别 (TIMx_BDTR 寄存器中的 LOCK 位) 设为’1’，则该位不能被修改。

注：任何对该位的写操作都需要一个 APB 时钟的延迟以后才能起作用。

Bit 12 BKE: 刹车使能 (Break enable)

0: 刹车输入禁止 (BRK 和 CCS 时钟失效事件)

1; 刹车输入允许 (BRK 和 CCS 时钟失效事件)

注：一旦 LOCK 级别 (TIMx_BDTR 寄存器中的 LOCK 位) 设为’1’，则该位不能被修改。

注：任何对该位的写操作都需要一个 APB 时钟的延迟以后才能起作用。

Bit 11 OSSR: 运行模式下“关闭状态”选择 (Off-state selection for Run mode)

该位用于当 MOE=1 且通道为互补输出时。没有互补输出的定时器中不存在 OSSR 位。参考 OC/OCN 使能的详细说明 (275 页 15.4.9 节，TIM1 捕获 / 比较使能寄存器 (TIMx_CCER))。

0：当定时器不工作时，禁止 OC/OCN 输出 (OC/OCN 使能输出信号 =0)；

1：当定时器不工作时，一旦 CCxE=1 或 CCxNE=1，OC/OCN 使能并输出无效电平，然后置 OC/OCN 使能输出信号 =1。

注：一旦 LOCK 级别 (TIMx_BDTR 寄存器中的 LOCK 位) 设为’2’，则该位不能被修改。

Bit 10 OSSI: 运行模式下“空闲状态”选择 (Off-state selection fo Idle mode)

该位用于当 MOE=0 时通道为输出。

参考 OC/OCN 使能的详细说明 (275 页 15.4.9 节，TIM1 捕获 / 比较使能寄存器 (TIMx_CCER))。

0：当定时器不工作时，禁止 OC/OCN 输出 (OC/OCN 使能输出信号 =0)；

1：当定时器不工作时，一旦 CCxE=1 或 CCxNE=1，OC/OCN 被强制输出空闲电平，置 OC/OCN 使能输出信号 =1。

注：一旦 LOCK 级别 (TIMx_BDTR 寄存器中的 LOCK 位) 设为’2’，则该位不能被修改。

Bits 9:8 LOCK[1:0]: 锁定设置 (Lock configuration)

该位为防止软件错误而提供写保护。

00：锁定关闭，寄存器无写保护；

01：锁定级别 1，不能写入 TIMx_BDTR 寄存器的 DTG、BKE、BKP、AOE 位和 TIMx_CR2 寄存器的 OISx/OISxN 位；

10：锁定级别 2，不能写入锁定级别 1 中的各位，也不能写入 CC 极性位 (一旦相关通道通过 CCxS 位设为输出，CC 极性位是 TIMx_CCER 寄存器的 CCxP/CCNxP 位) 以及 OSSR/OSSI 位；

11：锁定级别 3，不能写入锁定级别 2 中的各位，也不能写入 CC 控制位 (一旦相关通道通过 CCxS 位设为输出，CC 控制位是 TIMx_CCMRx 寄存器的 OCxM/OCxPE 位)；

注：在系统复位后，只能写一次 LOCK 位，一旦写入 TIMx_BDTR 寄存器，则其内容冻结直至复位。

Bits 7:0 DTG[7:0]: 死区发生器设置 (Dead-time generator setup)

这些位定义了插入互补输出之间的死区持续时间。假设 DT 表示其持续时间：

DTG[7:5]=0xx => DT=DTG[7:0] × Tdtg，Tdtg = TDTS； DTG[7:5]=10x => DT=(64+DTG[5:0]) × Tdtg，Tdtg = 2 × TDTS； DTG[7:5]=110 => DT=(32+DTG[4:0]) × Tdtg，Tdtg = 8 × TDTS； DTG[7:5]=111 => DT=(32+DTG[4:0])× Tdtg，Tdtg = 16 × TDTS；

例：若 TDTS = 125ns(8MHZ)，可能的死区时间为：

0 到 15875ns，若步长时间为 125ns；

16us 到 31750ns，若步长时间为 250ns；

32us 到 63us，若步长时间为 1us；

64us 到 126us，若步长时间为 2us；

注：一旦 LOCK 级别 (TIMx_BDTR 寄存器中的 LOCK 位) 设为 1、2 或 3，则不能修改这些位。

18.5.16 TIM15 DMA 控制寄存器 (TIM15_DCR)

偏移地址 : 0x48

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

Bits 15:13 保留，始终读为‘0’

Bits 12:8 DBL[4:0]: DMA 连续传送长度 (DMA burst length)

这些位定义了 DMA 在连续模式下的传送长度 (当对 TIMx_DMAR 寄存器进行读或写时，定时器则 进行一次连续传送)，即：定义传输的字节数目：

00000：1 个字节

00001：2 个字节

00010：3 个字节

……

10001：18 个字节

Bits 7:5 保留，始终读为‘0’

Bits 4:0 DBA[4:0]: DMA 基地址 (DMA base address)

这些位定义了 DMA 传送的基地址 (当对 TIMx_DMAR 寄存器进行读或写时)，DBA 定义为从 TIMx_CR1 寄存器所在地址开始的偏移量 .

例如：

00000：TIMx_CR1，

00001：TIMx_CR2，

00010：TIMx_SMCR，

……

18.5.17 TIM15 DMA 全传输地址寄存器 (TIM15_DMAR)

偏移地址 : 0x4C

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAB[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:0 DMAB[15:0]: DMA 连续传送寄存器 (DMA register for burst accesses)

对 TIMx_DMAR 寄存器的读或写会导致对以下地址所在寄存器的存取操作：

TIMx_CR1 地址 + （DBA + DMA 索引）×4，

其中：“TIMx_CR1 地址”是控制寄存器 1(TIMx_CR1) 所在的地址；

“DBA”是 TIMx_DCR 寄存器中定义的基地址；

“DMA 索引”是由 DMA 自动控制的偏移量，它取决于 TIMx_DCR 寄存器中定义的 DBL。

18.5.18 TIM15 寄存器映射

下表中将 TIM15 的所有寄存器映射到一个 16 位可寻址 (编址) 空间。

表 54. TIM15 寄存器映射与 复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**TIM15_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKD [1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
	Reset value																							0	0	0				0	0	0	0
0x04	**TIM15_CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIS2	OIS1N	OIS1	Res.	MMS[2:0]			CCDS	CCUS	Res.	CCPC
	Reset value																						0	0	0		0	0	0	0	0		0
0x08	**TIM15_SMCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MSM	TS[2:0]			Res.	SMS[2:0]		
	Reset value																									0	0	0	0		0	0	0
0x0C	**TIM15_DIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDE	Res.	Res.	Res.	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	Res.	Res.	CC2IE	CC1IE	UIE
	Reset value																		0				0	0	0	0	0	0			0	0	0
0x10	**TIM15_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2OF	CC1OF	Res.	BIF	TIF	COMIF	Res.	Res.	CC2IF	CC1IF	UIF
	Reset value																						0	0		0	0	0			0	0	0
0x14	**TIM15_EGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BG	TG	COMG	Res.	Res.	CC2G	CC1G	UG
	Reset value																									0	0	0			0	0	0
0x18	**TIM15_CCMR1** Output compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M [2:0]			OC2PE	OC2FE	CC2S [1:0]		Res.	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
	Reset value																		0	0	0	0	0	0	0		0	0	0	0	0	0	0
	TIM15_CCMR1 Input capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC2F[3:0]				IC2 PSC [1:0]	CC2S [1:0]		IC1F[3:0]				IC1 PSC [1:0]	CC1S [1:0]		
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	**TIM15_CCER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2NP	Res.	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
	Reset value																									0		0	0	0	0	0	0
0x24	**TIM15_CNT**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]														
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 54. TIM15 寄存器映射与复位值 (续)

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x28	TIM15_PSC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	TIM15_ARR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	TIM15_RCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x34	TIM15_CCR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	TIM15_CCR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR2[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	TIM15_BDTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK [1:0]		DT[7:0]							
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	TIM15_DCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
	Reset value																				0	0	0	0	0				0	0	0	0	0
0x4C	TIM15_DMAR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAB[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

关于寄存器的起始地址，请参见 35 页 2.2.2 节

18.6 TIM16 和 TIM17 寄存器

关于在寄存器描述里面所用到的缩写，详见 31 页第 1.1 节。

18.6.1 TIM16 和 TIM17 控制寄存器 1 (TIM16_CR1 and TIM17_CR1)

偏移地址 : 0x00

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CKD[1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
						rw	rw	rw				rw	rw	rw	rw

Bits 15:10 保留，读始终为‘0’.

Bits 9:8 CKD: 时钟分频因子 (Clock division)

定义在定时器时钟 (CK_INT) 频率与数字滤波器 (ETR，TIx) 使用的采样频率之间的分频比例。

00：tDTS = tCK_INT

01：tDTS = 2 x tCK_INT

10：tDTS = 4 x tCK_INT

11：保留

Bit 7 ARPE: 自动重装载预装载允许位 (Auto-reload preload enable)

0：TIMx_ARR 寄存器没有缓冲；

1：TIMx_ARR 寄存器缓冲器有效。

Bits 6:4 保留，读始终为‘0’

Bit 3 OPM: 单脉冲模式 (One pulse mode)

0：在发生更新事件时，计数器不停止；

1：在发生下一次更新事件 (清除 CEN 位) 时，计数器停止

Bit 2 URS: 更新请求源 (Update request source)

软件通过该位选择 UEV 事件的源

0：如果使能了更新中断或 DMA 请求，则下述任一事件产生更新中断或 DMA 请求：

— 计数器溢出 / 下溢

— 设置 UG 位

— 从模式控制器产生的更新

1：如果使能了更新中断或DMA请求，则只有计数器溢出/下溢才产生更新中断或DMA请求。

Bit 1 UDIS: 禁止更新 (Update disable)

软件通过该位允许 / 禁止 UEV 事件的产生

0：允许 UEV。更新 (UEV) 事件由下述任一事件产生：

— 计数器溢出 / 下溢

— 设置 UG 位

— 从模式控制器产生的更新具有缓存的寄存器被装入它们的预装载值。

1：禁止 UEV。不产生更新事件，影子寄存器 (ARR、PSC、CCRx) 保持它们的值。如果设置了 UG 位或从模式控制器发出了一个硬件复位，则计数器和预分频器被重新初始化。

Bit 0 CEN: 使能计数器（Counter enable）

0：禁止计数器；

1：使能计数器。

注：在软件设置了 CEN 位后，外部时钟、门控模式和编码器模式才能工作。触发模式可以自动地通过硬件设置 CEN 位。

在单脉冲模式下，当发生更新事件时，CEN 被自动清除。

18.6.2 TIM16 和 TIM17 控制寄存器 2 (TIM16_CR2 and TIM17_CR2)

偏移地址 : 0x04

复位值 : 0x0000

Bits 15:10 保留，读始终为‘0’.

Bit 9 OIS1N: 输出空闲状态 1 (OC1N 输出)

0: OC1N=0 ， 当 MOE=0 后一个死区时间

1: OC1N=1 ， 当 MOE=0 后一个死区时间

注： 一旦通过 TIMx_BKR 寄存器的 LOCK 编程设置了 LOCK 等级 1，2 或 3，此位不可再被修改。

Bit 8 OIS1: 输出空闲状态 1 (OC1 输出)

0: OC1=0 ， 当 MOE=0（如果 OC1N 被使用还要等一个死区时间）

1: OC1=1 ， 当 MOE=0（如果 OC1N 被使用还要等一个死区时间）

注： 一旦通过 TIMx_BKR 寄存器的 LOCK 编程设置了 LOCK 等级 1，2 或 3，此位不可再被修改。

Bit 7:4 保留，读始终为‘0’

Bit 3 CCDS: 捕获 / 比较的 DMA 选择（Capture/compare DMA selection）

0：当发生 CCx 事件时，送出 CCx 的 DMA 请求；

1：当发生更新事件时，送出 CCx 的 DMA 请求。

Bit 2 CCUS: 捕捉 / 比较控制更新选择（Capture/compare control update selection）

0: 当捕捉 / 比较控制位被 预装载时 (CCPC=1)，只有通过设置 COMG 位才会被 更新

1: 当捕捉 / 比较控制位被 预装载时 (CCPC=1)，在设置 COMG 位或在 TRGI 上产生上升沿时都会被更新

注：此位仅汉通道有互补输出时才起作用

Bit 1 保留，读始终为‘0’.

Bit 0 CCPC: 捕捉 / 比较预装载控制（Capture/compare preloaded control）

0: CCxE, CCxNE 和 OCxM 位不进行预装载

1: CCxE, CCxNE 和 OCxM 位在被写入后被预装载，只有在 COM 位被设置为‘1’时才更新

注：此位仅汉通道有互补输出时才起作用

18.6.3 TIM16 和 TIM17 DMA/ 中断允许寄存器 (TIM16_DIER 和 TIM17_DIER)

偏移地址 : 0x0C

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	Res.	Res.	Res.	Res.	CC1DE	UDE	BIE	TIE	COMIE	Res.	Res.	Res.	CC1IE	UIE
	rw					rw	rw	rw	rw	rw				rw	rw

Bit 15 保留，读始终为‘0’.

Bit 14 TDE: 触发 DMA 请求允许
0: 触发 DMA 请求禁止 .
1: 触发 DMA 请求允许

Bit 13:10 保留，读始终为‘0’

Bit 9 CC1DE: 捕捉 / 比较 1 DMA 请求允许
0: CC1 DMA 请求禁止 .
1: CC1 DMA 请求允许

Bit 8 UDE: 更新 DMA 请求允许
0: 更新 DMA 请求禁止 .
1: 更新 DMA 请求允许

Bit 7 BIE: 刹车中断允许
0: 刹车中断禁止 .
1: 刹车中断允许

Bit 6 TIE: 触发中断允许
0: 触发中断禁止 .
1: 触发中断允许

Bit 5 COMIE: COM 中断允许
0: COM 中断禁止 .
1: COM 中断允许

Bit 4:2 保留，始终读为‘0’

Bit 1 CC1IE: 捕捉 / 比较 1 中断允许
0: CC1 中断禁止
1: CC1 中断允许

Bit 0 UIE: 更新中断允许
0: 更新中断禁止
1: 更新中断允许

18.6.4 TIM16 和 TIM17 状态寄存器 (TIM16_SR 和 TIM17_SR)

偏移地址 : 0x10

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CC1OF	Res.	BIF	TIF	COMIF	Res.	Res.	Res.	CC1IF	UIF
						rc_w0		rc_w0	rc_w0	rc_w0				rc_w0	rc_w0

Bits 15:10 保留，读始终为‘0’.

Bit 9 CC1OF: 捕捉 / 比较 1 重复捕捉标志 (Capture/Compare 1 overcapture flag)

仅当相应的通道被配置为输入捕获时，该标记可由硬件置’1’。写’0’可清除该位。

0：无重复捕获产生；

1：当计数器的值被捕获到 TIMx_CCR1 寄存器时，CC1IF 的状态已经为’1’。

Bit 8 保留，读始终为‘0’.

Bit 7 BIF: 刹车中断标记 (Braek interrupt flag)

当刹车输入信号有效时由硬件对该位置’1’。刹车信号无效时可由软件清’0’。

0：无刹车事件产生；

1：刹车输入端检测到有效电平。

Bit 6 TIF: 触发器中断标记 (Trigger interrupt flag)

当发生触发事件 (当从模式控制器处于除门控模式外的其它模式时，在 TRGI 输入端检测到有效边沿，或门控模式下的任一边沿) 时由硬件对该位置’1’。它由软件清’0’。

0：无触发器事件产生；

1：触发器中断等待响应。

Bit 5 COMIF:COM 中断标记 (COM interrupt flag)

当发生 COM 事件 (一旦捕捉 / 比较控制位——CCxE, CCxNE, OCxM，被 更新) 时由硬件对该位置’1’。它由软件清’0’。

0：无 COM 事件产生；

1：COM 中断等待响应。

Bits 4:2 保留，读始终为‘0’.

Bit 1 CC1IF: 捕捉 / 比较 1 中断标志 (Capture/Compare 1 interrupt flag)

如果通道 CC1 配置为输出模式：

当计数器值与比较值匹配时该位由硬件置’1’，但在中心对称模式下除外 (参考 TIMx_CR1 寄存器的 CMS 位)。它由软件清’0’。

0：无匹配发生；

1：TIMx_CNT 的值与 TIMx_CCR1 的值匹配。

当 TIMx_CCR1 的内容大于 TIMx_ARR 时，在计数器溢出（向上或向上 / 向下计数模式）或计数器下溢出时（向下计数模式）时 CC1IF 位变高

如果通道 CC1 配置为输入模式：

当捕获事件发生时该位由硬件置’1’，它由软件清’0’或通过读 TIMx_CCR1 清’0’。

0：无输入捕获产生；

1：计数器值已被捕获 (拷贝) 至 TIMx_CCR1(在 IC1 上检测到与所选极性相同的边沿)。

Bit 0 UIF: 更新中断标记 (Update interrupt flag)

当产生更新事件时该位由硬件置’1’。它由软件清’0’。

0：无更新事件产生；

1：更新中断等待响应。

当寄存器被更新时该位由硬件置’1’：

— 若 TIMx_CR1 寄存器的 UDIS=0，当对应的重复计数器的值溢出（更新如果重复计数器＝ 0）时产生更新事件；

— 若 TIMx_CR1 寄存器的 UDIS=0、URS=0，当通过软件对 TIMx_EGR 寄存器的 UG 对计数器 CNT 重新初始化时产生更新事件；

— 若 TIMx_CR1 寄存器的 UDIS=0、URS=0，当计数器 CNT 被触发事件（参见 18.5.3：TIM15 从模式控制寄存器 TIM15_SMCR 的描述）重初始化时产生

18.6.5 TIM16 和 TIM17 事件产生寄存器 (TIM16_EGR 和 TIM17_EGR)

偏移地址 : 0x14

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BG	TG	COMG	Res.	Res.	Res.	CC1G	UG
								w	w	w				w	w

Bits 15:8 保留，读始终为‘0’.

Bit 7 BG: 刹车产生（Break generation）

该位由软件置’1’，用于产生一个刹车事件，由硬件自动清’0’。

0：无动作；

1：刹车事件产生，清除 MOET，设置 TIMx_SR 寄存器的 TIF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。

Bit 6 TG: 产生触发事件 (Trigger generation)

该位由软件置’1’，用于产生一个触发事件，由硬件自动清’0’。

0：无动作；

1：TIMx_SR 寄存器的 TIF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。

Bit 5 COMG: 捕捉 / 比较控制更新产生（Capture/Compare control update generation）

该位由软件置’1’，由硬件自动清’0’。

0：无动作；

1: 当 CCPC 位被设置时，才可能更新 CCxE, CCxNE 和 OCxM 位

注：此位仅当通道有互补输出时才起作用

Bits 4:2 保留，读始终为‘0’

Bit 1 CC1G: 捕捉 / 比较 1 产生

该位由软件置’1’，用于产生一个捕获 / 比较事件，由硬件自动清’0’。

0：无动作；

1：在通道 CC1 上产生一个捕获 / 比较事件：

若通道 CC1 配置为输出：

设置 CC1IF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。

若通道 CC1 配置为输入：

当前的计数器值捕获至 TIMx_CCR1 寄存器；设置 CC1IF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。若 CC1IF 已经为 1，则设置 CC1OF=1。

Bit 0 UG: 产生更新事件 (Update generation)

该位由软件置’1’，由硬件自动清’0’。

0：无动作；

1：重新初始化计数器，并产生一个更新事件。注意预分频器的计数器也被清’0’(但是预分频系数不变)。若在中心对称模式下或 DIR=0(向上计数) 则计数器被清’0’，若 DIR=1(向下计数) 则计数器取 TIMx_ARR 的值。

18.6.6 TIM16 和 TIM17 捕捉 / 比较模式寄存器 1 (TIM16_CCMR1 和 TIM17_CCMR1)

偏移地址 : 0x18

复位值 : 0x0000

通道可用于输入 (捕获模式) 或输出 (比较模式)，通道的方向由相应的 CCxS 定义。该寄存器其它位的作用在输入和输出模式下不同。OCxx 描述了通道在输出模式下的功能，ICxx 描述了通道在输入模式下的功能。因此必须注意，同一个位在输出模式和输入模式下的功能是不同的。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC1F[3:0]				IC1PSC[1:0]			
								rw	rw	rw	rw	rw	rw	rw	rw

输出比较模式

Bit 15:7 保留，读始终为‘0’.

Bits 6:4 OC1M: 输出比较 1 模式 (Output compare 1 mode)

该 3 位定义了输出参考信号 OC1REF 的动作，而 OC1REF 决定了 OC1 和 OC1N 的值。OC1REF 是高电平有效，而 OC1 和 OC1N 的有效电平取决于 CC1P 和 CC1PN 位。

000：冻结。输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 间的比较对 OC1REF 不起作用；（此模式仅用于产生时基）

001 ：匹配时设置通道 1 为有效电平。 当计数器 TIMx_CNT 的值与捕获 / 比较寄存器 1 (TIMx_CCR1) 相同时，强制 OC1REF 为高。

010 ：匹配时设置通道 1 为无效电平。 当计数器 TIMx_CNT 的值与捕获 / 比较寄存器 1 (TIMx_CCR1) 相同时，强制 OC1REF 为低。

011：翻转。当 TIMx_CCR1=TIMx_CNT 时，翻转 OC1REF 的电平。

100：强制为无效电平。强制 OC1REF 为低。

101：强制为有效电平。强制 OC1REF 为高。

110：PWM 模式 1 － 在向上计数时，一旦 TIMx_CNT<TIMx_CCR1 时通道 1 为有效电平，否则为无效电平；在向下计数时，一旦 TIMx_CNT>TIMx_CCR1 时通道 1 为无效电平 (OC1REF=0)，否 则为有效电平 (OC1REF=1)。

111：PWM 模式 2 － 在向上计数时，一旦 TIMx_CNT<TIMx_CCR1 时通道 1 为无效电平，否则为有效电平；在向下计数时，一旦 TIMx_CNT>TIMx_CCR1 时通道 1 为有效电平，否则为无效电平。

注 1：一旦 LOCK 级别设为 3(TIMx_BDTR 寄存器中的 LOCK 位) 并且 CC1S=’00’(该通道配置成输出) 则该位不能被修改。

注 2：在 PWM 模式 1 或 PWM 模式 2 中，只有当比较结果改变了或在输出比较模式中从冻结模式 切换到 PWM 模式时，OC1REF 电平才改变。

Bit 3 OC1PE: 输出比较 1 预装载使能 (Output compare 1 preload enable)

0：禁止 TIMx_CCR1 寄存器的预装载功能，可随时写入 TIMx_CCR1 寄存器，并且新写入的数值立即起作用。

1：开启 TIMx_CCR1 寄存器的预装载功能，读写操作仅对预装载寄存器操作，TIMx_CCR1 的预装载值在更新事件到来时被传送至当前寄存器中。

注 1：一旦 LOCK 级别设为 3(TIMx_BDTR 寄存器中的 LOCK 位) 并且 CC1S=’00’(该通道配置成输出) 则该位不能被修改。

注 2：仅在单脉冲模式下 (TIMx_CR1 寄存器的 OPM=’1’)，可以在未确认预装载寄存器情况下使用 PWM 模式，否则其动作不确定。

Bit 2 OC1FE: 输出比较 1 快速使能 (Output compare 1 fast enable)

该位用于加快 CC 输出对触发器输入事件的响应。

0：根据计数器与 CCR1 的值，CC1 正常操作，即使触发器是打开的。当触发器的输入出现一个有效沿时，激活 CC1 输出的最小延时为 5 个时钟周期。

1：输入到触发器的有效沿的作用就象发生了一次比较匹配。因此，OC 被设置为比较电平而与比较结果无关。采样触发器的有效沿和 CC1 输出间的延时被缩短为 3 个时钟周期。 该位只在通道被配置成 PWM1 或 PWM2 模式时起作用。

Bits 1:0 CC1S: 捕捉 / 比较 1 选择 (Capture/Compare 1 selection)

这 2 位定义通道的方向 (输入 / 输出)，及输入脚的选择：

00：CC1 通道被配置为输出；

01：CC1 通道被配置为输入，IC1 映射在 TI1 上；

10：CC1 通道被配置为输入，IC1 映射在 TI2 上；

11：CC1 通道被配置为输入，IC1 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)。

注：CC1S 仅在通道关闭时 (TIMx_CCER 寄存器的 CC1E=’0’) 才是可写的。

输入捕捉模式

Bits 15:8 保留，读始终为‘0’

Bits 7:4 IC1F: 输入捕获 1 滤波器 (Input capture 1 filter)

这几位定义了 TI1 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成，它记录到 N 个事件后会产生一个输出的跳变：

0000：无滤波器，以 fDTS 采样

0001：采样频率 fSAMPLING=fCK_INT，N=2

0010：采样频率 fSAMPLING=fCK_INT，N=4

0011：采样频率 fSAMPLING=fCK_INT，N=8

0100：采样频率 fSAMPLING=fDTS/2，N=6

0101：采样频率 fSAMPLING=fDTS/2，N=8

0110：采样频率 fSAMPLING=fDTS/4，N=6

0111：采样频率 fSAMPLING=fDTS/4，N=8

1000：采样频率 fSAMPLING=fDTS/8，N=6

1001：采样频率 fSAMPLING=fDTS/8，N=8

1010：采样频率 fSAMPLING=fDTS/16，N=5

1011：采样频率 fSAMPLING=fDTS/16，N=6

1100：采样频率 fSAMPLING=fDTS/16，N=8

1101：采样频率 fSAMPLING=fDTS/32，N=5

1110：采样频率 fSAMPLING=fDTS/32，N=6

1111：采样频率 fSAMPLING=fDTS/32，N=8

Bits 3:2 IC1PSC: 输入捕捉 1 预分频

这 2 位定义了 CC1 输入 (IC1) 的预分频系数。 一旦 CC1E=’0’(TIMx_CCER 寄存器中)，则预分频器复位。

00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获；

01：每 2 个事件触发一次捕获；

10：每 4 个事件触发一次捕获；

11：每 8 个事件触发一次捕获。

Bits 1:0 CC1S: 捕捉 / 比较 1 选择

这 2 位定义通道的方向 (输入 / 输出)，及输入脚的选择：

00：CC1 通道被配置为输出；

01：CC1 通道被配置为输入，IC1 映射在 TI1 上；

10：CC1 通道被配置为输入，IC1 映射在 TI2 上；

11：CC1 通道被配置为输入，IC1 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时 (由 TIMx_SMCR 寄存器的 TS 位选择)。

注：CC1S 仅在通道关闭时 (TIMx_CCER 寄存器的 CC1E=’0’) 才是可写的。

18.6.7 TIM16 和 TIM17 捕捉 / 比较使能寄存器 (TIM16_CCER 和 IM17_CCER)

偏移地址 : 0x20

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1NP	CC1NE	CC1P	CC1E
												rw	rw	rw	rw

Bits 15:4 保留，读始终为“0”.

Bit 3 CC1NP: 捕捉 / 比较 1 输出极性 .

0: OC1N 高有效

1: OC1N 低有效

注： 一旦 LOCK 级别设为 2 或 3(TIMx_BDTR 寄存器中的 LOCK 位) 并且 CC1S=’00’(该通道配置成输出) 则该位不能被修改。

Bit 2 CC1NE：捕捉 / 比较 1 互补输出使能（Capture/Compare 1 complementary output enable）

0: 关闭 - OC1N 无效 .

1: 开启 - OC1N 输出到相关输出引脚的信号取决于 MOE, OSSI, OSSR, OIS1, OIS1N 和 CC1E 位

Bit 1 CC1P: 捕捉 / 比较 1 输出极性 .

通道 CC1 配置为输出 :

0: OC1 高有效

1: OC1 低有效

通道 CC1 配置为输入 :

CC1NP/CC1P 用于选择作为触发或捕获的信号 TI1FP1 和 TI2FP1 的极性

该位 IC1 还是 IC1 的反相信号。

00：不反相 / 上升沿：捕获发生在 TIxFP1 的上升沿（复位、外部时钟或触发模式的捕捉或触发），TIxFP1 不反相（在门控、编码器模式下的触发）。

00：反相 / 下降沿：捕获发生在 TIxFP1 的下降沿（复位、外部时钟或触发模式的捕捉或触发），TIxFP1 反相（在门控、编码器模式下的触发）。

10：保留，不使用此配置

11：不反相 / 上升和下降沿：捕获发生在 TIxFP1 的上升沿和下降沿（复位、外部时钟或触发模式的捕捉或触发），TIxFP1 不反相（门控模式的触发）。此配置不能用于编码器模式。

注： 一旦 LOCK 级别设为 2 或 3(TIMx_BDTR 寄存器中的 LOCK 位) 并且 CC1S=’00’(该通道配置成输出) 则该位不能被修改。

Bit 0 CC1E: 捕捉 / 比较 1 输出使能 .

CC1 通道配置为输出：

0： 关闭－ OC1 禁止输出。

1： 开启－ 输出到对应的输出引脚的 OC1 信号取决于 MOE, OSSI, OSSR, OIS1, OIS1N 和 CC1E 位。

CC1 通道配置为输入：

该位决定了计数器的值是否能捕获入 TIMx_CCR1 寄存器。

0：捕获禁止；

1：捕获使能。

表 55. 带刹车功能的互补 OCx 和 OCxN 通道的输出控制位

Control bits					Output states(1)	
MOE bit	OSSI bit	OSSR bit	CCxE bit	CCxNE bit	OCx output state	OCxN output state
1	X	0	0	0	Output Disabled (not driven by the timer) OCx=0, OCx_EN=0	Output Disabled (not driven by the timer) OCxN=0, OCxN_EN=0
		0	0	1	Output Disabled (not driven by the timer) OCx=0, OCx_EN=0	OCxREF + Polarity OCxN=OCxREF xor CCxNP, OCxN_EN=1
		0	1	0	OCxREF + Polarity OCx=OCxREF xor CCxP, OCx_EN=1	Output Disabled (not driven by the timer) OCxN=0, OCxN_EN=0
		0	1	1	OCREF + Polarity + dead-time OCx_EN=1	Complementary to OCREF (not OCREF) + Polarity + dead-time OCxN_EN=1
		1	0	0	Output Disabled (not driven by the timer) OCx=CCxP, OCx_EN=0	Output Disabled (not driven by the timer) OCxN=CCxNP, OCxN_EN=0
		1	0	1	Off-State (output enabled with inactive state) OCx=CCxP, OCx_EN=1	OCxREF + Polarity OCxN=OCxREF xor CCxNP, OCxN_EN=1
		1	1	0	OCxREF + Polarity OCx=OCxREF xor CCxP, OCx_EN=1	Off-State (output enabled with inactive state) OCxN=CCxNP, OCxN_EN=1
		1	1	1	OCREF + Polarity + dead-time OCx_EN=1	Complementary to OCREF (not OCREF) + Polarity + dead-time OCxN_EN=1
0	0	X	0	0	Output Disabled (not driven by the timer) Asynchronously: OCx=CCxP, OCx_EN=0, OCxN=CCxNP, OCxN_EN=0 Then if the clock is present: OCx=OISx and OCxN=OISxN after a dead-time, assuming that OISx and OISxN do not correspond to OCX and OCxN both in active state.	
	0		0	1		
	0		1	0		
	0		1	1		
	1		0	0		
	1		0	1	Off-State (output enabled with inactive state) Asynchronously: OCx=CCxP, OCx_EN=1, OCxN=CCxNP, OCxN_EN=1 Then if the clock is present: OCx=OISx and OCxN=OISxN after a dead-time, assuming that OISx and OISxN do not correspond to OCX and OCxN both in active state	
	1		1	0		
	1		1	1		

1. 当某通道的两个输出都没有使用时 (CCxE = CCxNE = 0), OISx, OISxN, CCxP 和 CCxNP 必有保持清除。

注：连接到互补 OCx 和 OCxN 通道的外部 I/O 引脚状态，取决于 OCx 和 OCxN 通道状态和 GPIO + AFIO 寄存器。

18.6.8 TIM16 和 TIM17 计数器 (TIM16_CNT 和 TIM17_CNT)

偏移地址 : 0x24

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:0 CNT[15:0]: 计数器的值

18.6.9 TIM16 和 TIM17 预分频寄存器 (TIM16_PSC 和 TIM17_PSC)

偏移地址 : 0x28

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:0 PSC[15:0]：预分寄存器的值

计数器的时钟频率 CK_CNT 等于 fCK_PSC/(PSC[15:0]+1)。 PSC 包含了当更新事件（包括使用 TIMx_EGR 寄存器的 UG 或者当配置为复位模式时通过触发控制器清除计数器）产生时装入当前预分频器寄存器的值。

18.6.10 TIM16 和 TIM17 自动重装寄存器 (TIM16_ARR 和 TIM17_ARR)

偏移地址 : 0x2C 复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:0 ARR[15:0]: 自动重装寄存器（Auto-reload value）

ARR 包含了将要传送至实际的自动重装载寄存器的数值。有关 ARR 的更新和动作详细参考 350 页 17.3.1 节：时基单元。

当自动重装载的值为空时，计数器不工作。

18.6.11 TIM16 和 TIM17 重复计数寄存器 (TIM16_RCR 和 TIM17_RCR)

偏移地址 : 0x30

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:8 保留，读始终为“0”.

Bits 7:0 REP[7:0]: 重复计数器的值 (Repetition counter value)

预装载寄存器被使能后，这些位允许用户设置比较寄存器的更新速率 (即周期性地从预装载寄存器传输到当前寄存器)；如果允许产生更新中断，则会同时影响产生更新中断的速率。

每次向下计数器 REP_CNT 达到 0，会产生一个更新事件并且计数器 REP_CNT 重新从 REP 值开始计数。由于 REP_CNT 只有在周期更新事件 U_RC 发生时才重载 REP 值，因此对 TIMx_RCR 寄存器写入的新值只在下次周期更新事件发生时才起作用。

这意味着在 PWM 模式中，(REP+1) 对应着在边沿对齐模式下，PWM 周期的数目；

18.6.12 TIM16 和 TIM17 捕捉 / 比较寄存器 1 (TIM16_CCR1 和 TIM17_CCR1)

偏移地址 : 0x34

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:0 CCR1[15:0]: 捕捉 / 比较 1 的值

若 CC1 通道配置为输出：

CCR1 包含了装入当前捕获 / 比较 1 寄存器的值（预装载值）。

如果在 TIMx_CCMR1 寄存器（OC1PE 位）中未选择预装载特性，写入的数值会被立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获 / 比较 1 寄存器中。

当前捕获 / 比较寄存器参与同计数器 TIMx_CNT 的比较，并在 OC1 端口上产生输出信号。

若 CC1 通道配置为输入：

CCR1 包含了由上一次输入捕获 1 事件 (IC1) 传输的计数器值。

18.6.13 TIM16 和 TIM17 刹车与死区时间寄存器 (TIM16_BDTR 和 TIM17_BDTR)

偏移地址 : 0x44

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

注：根据锁定设置，AOE、BKP、BKE、OSSI、OSSR 和 DTG[7:0] 位均可被写保护，有必要在第一次写入 TIMx_BDTR 寄存器时对它们进行配置。

Bit 15 MOE: 主输出使能 (Main output enable)

一旦刹车输入有效，该位被硬件异步清’0’。根据 AOE 位的设置值，该位可以由软件清’0’或被自动置 1。它仅对配置为输出的通道有效。

0：禁止 OC 和 OCN 输出或强制为空闲状态；

1：如果设置了相应的使能位 (TIMx_CCER 寄存器的 CCxE、CCxNE 位)，则开启 OC 和 OCN 输出。

有关 OC/OCN 使能的细节，参见 408 页 18.5.8 节，TIM15 捕获 / 比较使能寄存器 (TIM15_CCER)。

Bit 14 AOE: 自动输出使能 (Automatic output enable)

0：MOE 只能被软件置’1’；

1：MOE 能被软件置’1’或在下一个更新事件被自动置’1’(如果刹车输入无效)。

注：一旦 LOCK 级别 (TIMx_BDTR 寄存器中的 LOCK 位) 设为’1’，则该位不能被修改。

Bit 13 BKP: 刹车输入极性 (Break polarity)

0：刹车输入低电平有效；

1：刹车输入高电平有效。

注：一旦 LOCK 级别 (TIMx_BDTR 寄存器中的 LOCK 位) 设为’1’，则该位不能被修改。

注：任何对该位的写操作都需要一个 APB 时钟的延迟以后才能起作用。

Bit 12 BKE: 刹车使能 (Break enable)

0: 刹车输入禁止 (BRK 和 CCS 时钟失效事件)

1; 刹车输入允许 (BRK 和 CCS 时钟失效事件)

注：一旦 LOCK 级别 (TIMx_BDTR 寄存器中的 LOCK 位) 设为’1’，则该位不能被修改。

注：任何对该位的写操作都需要一个 APB 时钟的延迟以后才能起作用。

Bit 11 OSSR: 运行模式下“关闭状态”选择 (Off-state selection for Run mode)

该位用于当 MOE=1 且通道为互补输出时。没有互补输出的定时器中不存在 OSSR 位。

参考 OC/OCN 使能的详细说明 (275 页 15.4.9 节，TIM1 捕获 / 比较使能寄存器 (TIMx_CCER))。

0：当定时器不工作时，禁止 OC/OCN 输出 (OC/OCN 使能输出信号 =0)；

1：当定时器不工作时，一旦 CCxE=1 或 CCxNE=1，OC/OCN 使能并输出无效电平，然后置 OC/OCN 使能输出信号 =1。

注：一旦 LOCK 级别 (TIMx_BDTR 寄存器中的 LOCK 位) 设为’ 2’，则该位不能被修改。

Bit 10 OSSI: 运行模式下“空闲状态”选择 (Off-state selection fo Idle mode)

该位用于当 MOE=0 时通道为输出。

参考 OC/OCN 使能的详细说明 (275 页 15.4.9 节，TIM1 捕获 / 比较使能寄存器 (TIMx_CCER))。

0：当定时器不工作时，禁止 OC/OCN 输出 (OC/OCN 使能输出信号 =0)；

1：当定时器不工作时，一旦 CCxE=1 或 CCxNE=1，OC/OCN 被强制输出空闲电平，置 OC/OCN 使能输出信号 =1。

注：一旦 LOCK 级别 (TIMx_BDTR 寄存器中的 LOCK 位) 设为’ 2’，则该位不能被修改。

Bits 9:8 LOCK[1:0]: 锁定设置 (Lock configuration)

该位为防止软件错误而提供写保护。

00：锁定关闭，寄存器无写保护；

01：锁定级别 1，不能写入 TIMx_BDTR 寄存器的 DTG、BKE、BKP、AOE 位和 TIMx_CR2 寄存器的 OISx/OISxN 位；

10：锁定级别 2，不能写入锁定级别 1 中的各位，也不能写入 CC 极性位 (一旦相关通道通过 CCxS 位设为输出，CC 极性位是 TIMx_CCER 寄存器的 CCxP/CCNxP 位) 以及 OSSR/OSSI 位；

11：锁定级别 3，不能写入锁定级别 2 中的各位，也不能写入 CC 控制位 (一旦相关通道通过 CCxS 位设为输出，CC 控制位是 TIMx_CCMRx 寄存器的 OCxM/OCxPE 位)；

注：在系统复位后，只能写一次 LOCK 位，一旦写入 TIMx_BDTR 寄存器，则其内容冻结直至复位。

Bits 7:0 DTG[7:0]: 死区发生器设置 (Dead-time generator setup)

这些位定义了插入互补输出之间的死区持续时间。假设 DT 表示其持续时间：

DTG[7:5]=0xx => DT=DTG[7:0] × Tdtg，Tdtg = TDTS； DTG[7:5]=10x => DT=(64+DTG[5:0]) × Tdtg，Tdtg = 2 × TDTS； DTG[7:5]=110 => DT=(32+DTG[4:0]) × Tdtg，Tdtg = 8 × TDTS； DTG[7:5]=111 => DT=(32+DTG[4:0])× Tdtg，Tdtg = 16 × TDTS；

例：若 TDTS = 125ns(8MHZ)，可能的死区时间为：

0 到 15875ns，若步长时间为 125ns；

16us 到 31750ns，若步长时间为 250ns；

32us 到 63us，若步长时间为 1us；

64us 到 126us，若步长时间为 2us；

注：一旦 LOCK 级别 (TIMx_BDTR 寄存器中的 LOCK 位) 设为 1、2 或 3，则不能修改这些位。

18.6.14 TIM16 和 TIM17 DMA 控制寄存器 (TIM16_DCR 和 TIM17_DCR)

偏移地址 : 0x48

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

Bits 15:13 保留，读始终为‘0’.

Bits 12:8 DBL[4:0]: DMA 连续传送长度 (DMA burst length)

这些位定义了 DMA 在连续模式下的传送长度 (当对 TIMx_DMAR 寄存器进行读或写时，定时器则 进行一次连续传送)，即：定义传输的字节数目：

00000：1 个字节

00001：2 个字节

00010：3 个字节

.....

.....

10001：18 个字节

Bits 7:5 保留，始终读为‘0’

Bits 4:0 DBA[4:0]: DMA 基地址 (DMA base address)

这些位定义了 DMA 传送的基地址 (当对 TIMx_DMAR 寄存器进行读或写时)，DBA 定义为从 TIMx_CR1 寄存器所在地址开始的偏移量 .

例如：

00000：TIMx_CR1，

00001：TIMx_CR2，

00010：TIMx_SMCR，

.....

例：要完成如下的传输：DBL = 7 ，DBA = TIMx_CR1

此时传送从 TIMx_CR1 的地址开始向 / 自连续 7 个寄存器进行操作。

18.6.15 TIM16 和 TIM17 DMA 全传输地址寄存器 (TIM16_DMAR 和 TIM17_DMAR)

偏移地址 : 0x4C

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAB[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:0 DMAB[15:0]: DMA 连续传送寄存器 (DMA register for burst accesses)

对 TIMx_DMAR 寄存器的读或写会导致对以下地址所在寄存器的存取操作：

TIMx_CR1 地址 + （DBA + DMA 索引）×4，

其中：“TIMx_CR1 地址”是控制寄存器 1(TIMx_CR1) 所在的地址；

“DBA”是 TIMx_DCR 寄存器中定义的基地址；

“DMA 索引”是由 DMA 自动控制的偏移量，它取决于 TIMx_DCR 寄存器中定义的 DBL。

如何使用 DMA 并发操作的例子

本例中使用定时器 DMA 的并发功能，将 CCRx 寄存器 (x = 2, 3, 4) 的内容以半字方式进行 DMA 传输，更新到 CCRx 寄存器。
按如下步骤进行操作：

1. 配置相关的 DMA 通道 :
   - DMA 通道设备地址为 DMAR 寄存器地址
   - DMA 通道存储器地址为包含要通过 DMA 传送到 CCRx 寄存器的数据 RAM 缓冲区地址
   - 传送数据数量 = 3 (见下面的注).
   - 通告模式禁止 .
2. 配置 DCR 寄存器的 DBA 和 DBL 位 : DBL = 3 次传送 , DBA = 0xE.
3. 使能 TIMx 更新 DMA 请求 (设置 DIER 寄存器的 UDE 位).
4. 使能 TIMx
5. 使能 DMA 通道

注：在本例中所有 CCRx 寄存器被一次性全部更新。如果需要更新 CCRx 寄存器两次，传送的数据数量应该是 6 ，而 RAM 缓冲区要包含 data1, data2, data3, data4, data5 和 data6。数据按如下过程被传送到 CCRx 寄存器：在第一个更新 DMA 请求时，data1 被传送到 CCR2, data2 被传送到 CCR3, data3 被传送到 CCR4，在第二个 DMA 更新中断请求时，data4 被传送到 CCR2, data5 被传送到 CCR3，data6 被传送到 CCR4

18.6.16 TIM16 和 TIM17 寄存器映射

下表中将 TIM16 和 TIM17 的所有寄存器映射到一个 16 位可寻址 (编址) 空间。

表 56. TIM16 和 TIM17 寄存器映射与 复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**TIM16_CR1 and TIM17_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKD [1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
	Reset value																							0	0	0				0	0	0	0
0x04	**TIM16_CR2 and TIM17_CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIS1N	OIS1	Res.	Res.	Res.	Res.	CCDS	CCUS	Res.	CCPC
	Reset value																							0	0					0	0		0
0x0C	**TIM16_DIER and TIM17_DIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDE	Res.	Res.	Res.	Res.	CC1DE	UDE	BIE	TIE	COMIE	Res.	Res.	Res.	CC1IE	UIE
	Reset value																		0					0	0	0	0	0				0	0
0x10	**TIM16_SR and TIM17_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1OF	Res.	BIF	TIF	COMIF	Res.	Res.	Res.	CC1IF	UIF
	Reset value																							0		0	0	0				0	0
0x14	**TIM16_EGR and TIM17_EGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BG	TG	COMG	Res.	Res.	Res.	CC1G	UG
	Reset value																									0	0	0				0	0
0x18	**TIM16_CCMR1 and TIM17_CCMR1** Output compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
	Reset value																										0	0	0	0	0	0	0
	TIM16_CCMR1 and TIM17_CCMR1 Input capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC1F[3:0]				IC1 PSC [1:0]		CC1S [1:0]	
	Reset value																									0	0	0	0	0	0	0	0
0x20	**TIM16_CCER and TIM17_CCER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1NP	CC1NE	CC1P	CC1E
	Reset value																													0	0	0	0
0x24	**TIM16_CNT and TIM17_CNT**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	**TIM16_PSC and TIM17_PSC**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**TIM16_ARR and TIM17_ARR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	**TIM16_RCR and TIM17_RCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x34	**TIM16_CCR1 and TIM17_CCR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	**TIM16_BDTR and TIM17_BDTR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK [1:0]		DT[7:0]							
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 56. TIM16 和 TIM17 寄存器映射与复位值 (续)

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x48	**TIM16_DCR and TIM17_DCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
	Reset value																				0	0	0	0	0				0	0	0	0	0
0x4C	**TIM16_DMAR and TIM17_DMAR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAB[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

关于寄存器的起始地址，请参见 35 页 2.2.2 节

19 基本定时器 (TIM6)

19.1 TIM6 简介

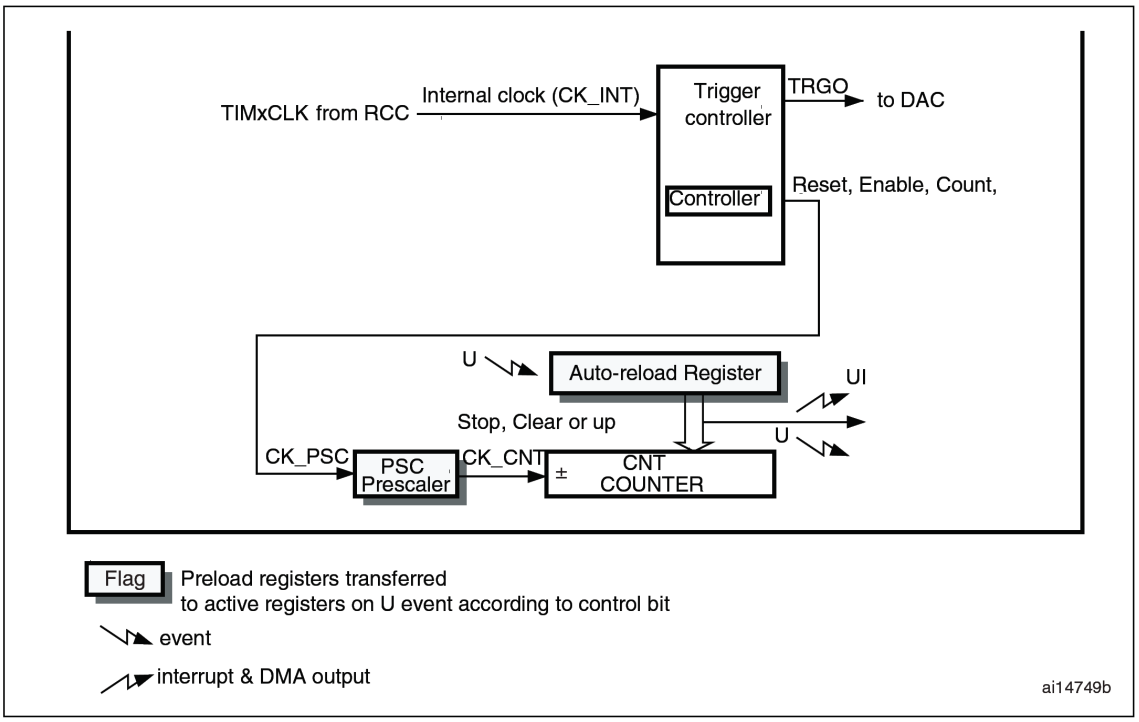
基本定时器 TIM6 包含一个 16 位自动装载计数器，由各自的可编程预分频器驱动。

它可以作为通用定时器提供时间基准，特别地可以为数模转换器 (DAC) 提供时钟。实际上，它在芯片内部直接连接到 DAC 并通过触发输出直接驱动 DAC。

19.2 TIM6 主要特性

- 16 位自动重装载累加计数器
- 16 位可编程 (可实时修改) 预分频器，用于对输入的时钟按系数为 1 ～ 65536 之间的任意数值分频
- 触发 DAC 的同步电路
- 在更新事件 (计数器溢出) 时产生中断 /DMA 请求

图 180. 基本寄存器框图



19.3 TIM6 功能描述

19.3.1 时基单元

这个可编程定时器的主要部分是一个带有自动重装载的 16 位累加计数器，计数器的时钟通过一个预分频器得到。

软件可以读写计数器、自动重装载寄存器和预分频寄存器，即使计数器运行时也可以操作。时基单元包含：

- 计数器寄存器 (TIMx_CNT)
- 预分频寄存器 (TIMx_PSC)
- 自动重装载寄存器 (TIMx_ARR)

自动重装载寄存器是预加载的，每次读写自动重装载寄存器时，实际上是通过读写预加载寄存器实现。根据 TIMx_CR1 寄存器中的自动重装载预加载使能位 (ARPE)，写入预加载寄存器的内容能够立即或在每次更新事件时，传送到它的影子寄存器。当 TIMx_CR1 寄存器的 UDIS 位 为’0’，则每当计数器达到溢出值时，硬件发出更新事件；软件也可以产生更新事件；关于更新 事件的产生，随后会有详细的介绍。

计数器由预分频输出 CK_CNT 驱动，设置 TIMx_CR1 寄存器中的计数器使能位 (CEN) 使能计数器计数。

注意：实际的设置计数器使能信号 CNT_EN 相对于 CEN 滞后一个时钟周期。

预分频器

预分频可以以系数介于 1 至 65536 之间的任意数值对计数器时钟分频。它是通过一个 16 位寄存器 (TIMx_PSC) 的计数实现分频。因为 TIMx_PSC 控制寄存器具有缓冲，可以在运行过程中改变它 的数值，新的预分频数值将在下一个更新事件时起作用。

以下两图是在运行过程中改变预分频系数的例子。

图 181. 预分频系数从 1 变到 2 的计数器时序图

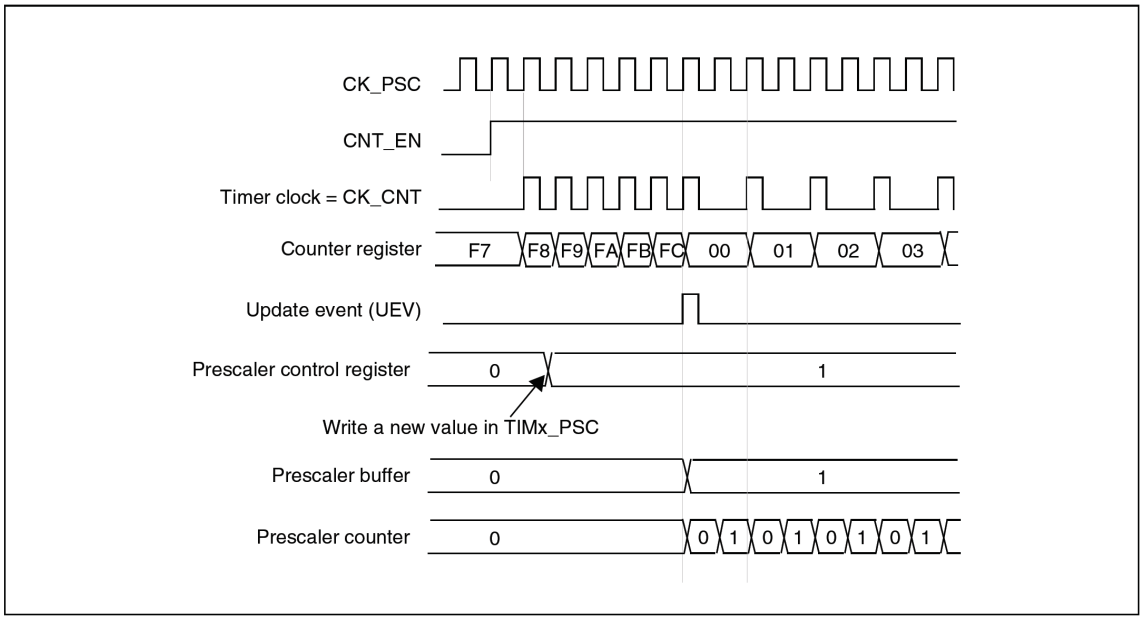
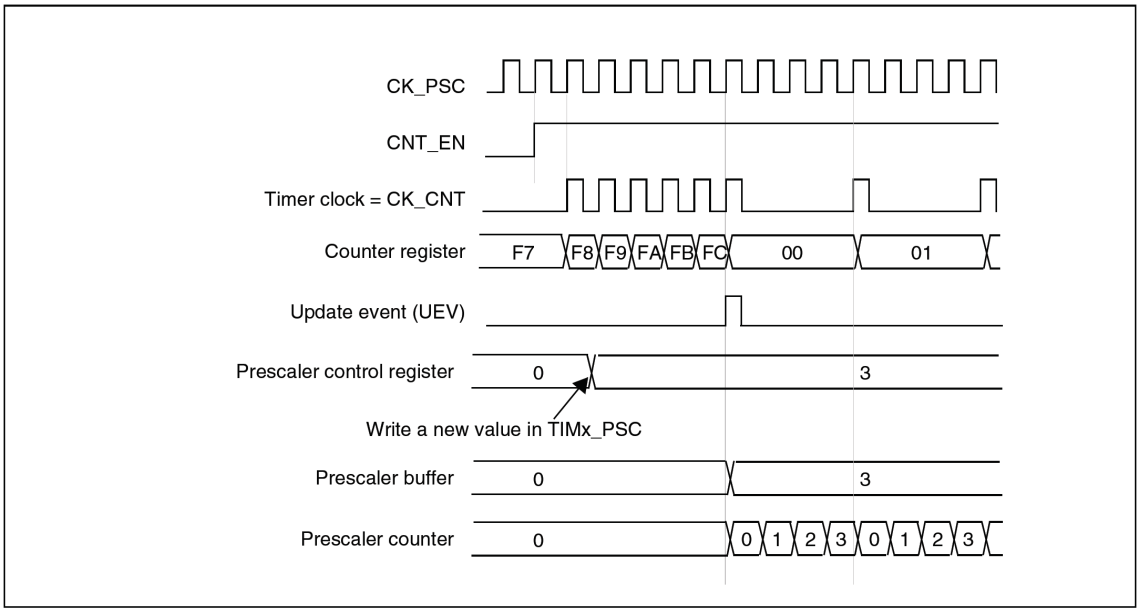


图 182. 预分频系数从 1 变到 4 的计数器时序图



19.3.2 计数模式

计数器从 0 累加计数到自动重装载数值 (TIMx_ARR 寄存器)，然后重新从 0 开始计数并产生一个计数器溢出事件。

每次计数器溢出时可以产生更新事件；(通过软件或使用从模式控制器) 设置 TIMx_EGR 寄存器的 UG 位也可以产生更新事件。

设置 TIMx_CR1 中的 UDIS 位可以禁止产生 UEV 事件，这可以避免在写入预加载寄存器时更改影子寄存器。在清除 UDIS 位为’0’之前，将不再产生更新事件，但计数器和预分频器依然会在应产生更新事件时重新从 0 开始计数 (但预分频系数不变)。另外，如果设置了 TIMx_CR1 寄存器中的 URS(选择更新请求)，设置 UG 位可以产生一次更新事件 UEV，但不设置 UIF 标志 (即没有中断或 DMA 请求)。

当发生一次更新事件时，所有寄存器会被更新并 (根据 URS 位) 设置更新标志 (TIMx_SR 寄存器的 UIF 位)：

- 传送预装载值 (TIMx_PSC 寄存器的内容) 至预分频器的缓冲区。
- 自动重装载影子寄存器被更新为预装载值 (TIMx_ARR)。

以下是一些在 TIMx_ARR=0x36 时不同时钟频率下计数器工作的图示例子。

图 183. 计数器时序图，内部时钟分频系数为 1

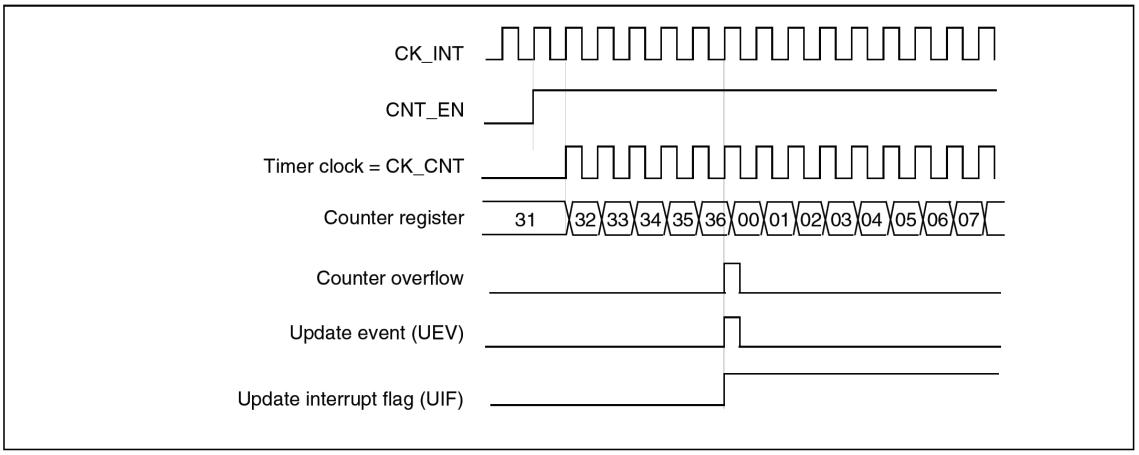


图 184. 计数器时序图，内部时钟分频系数为 2

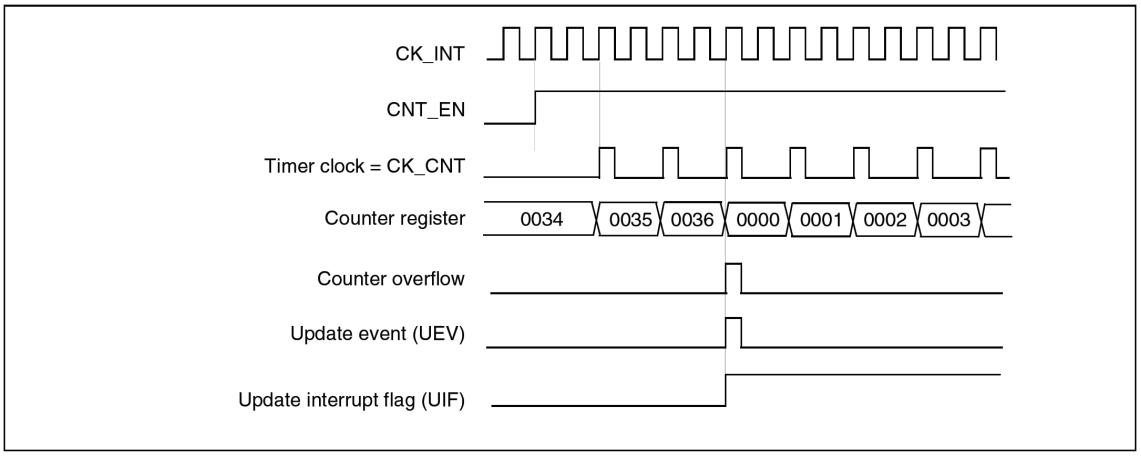


图 185. 计数器时序图，内部时钟分频系数为 4

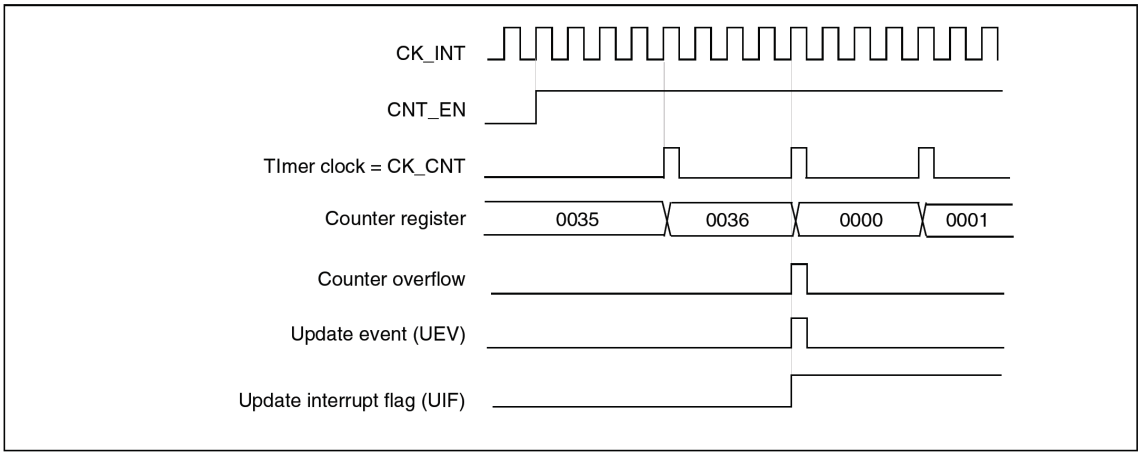


图 186. 计数器时序图，内部时钟分频系数为 N

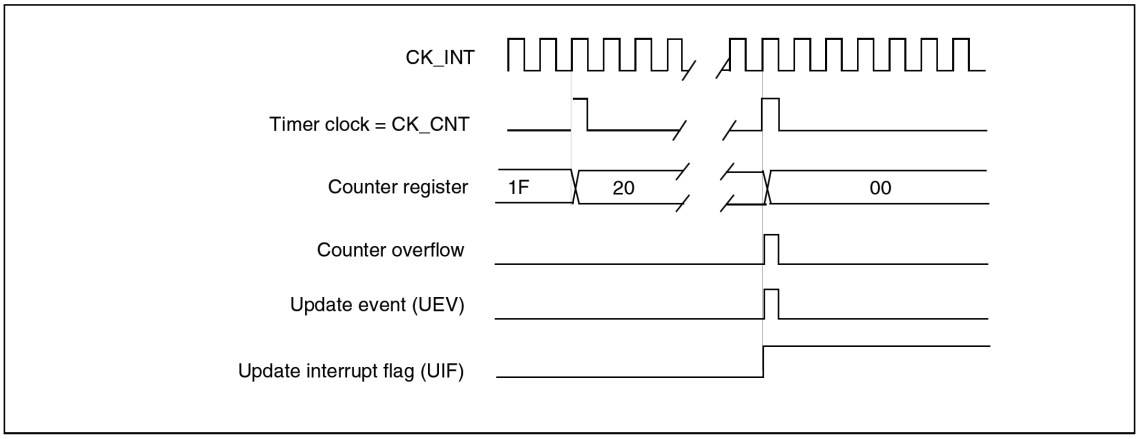


图 187. 计数器时序图，当 ARPE=1 时的更新事件 (无预装载 TIMx_ARR)

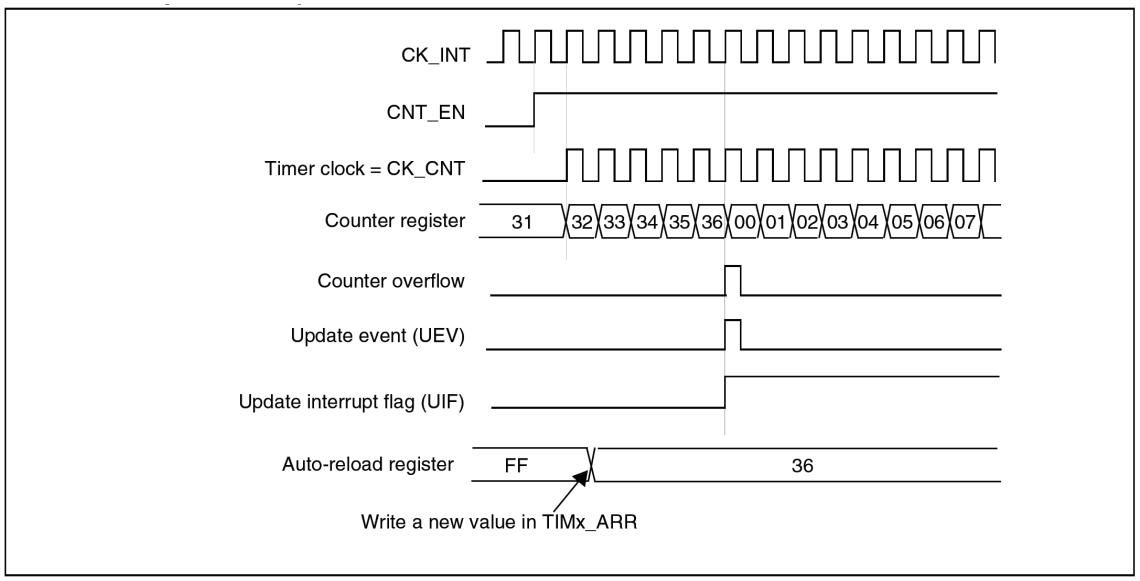
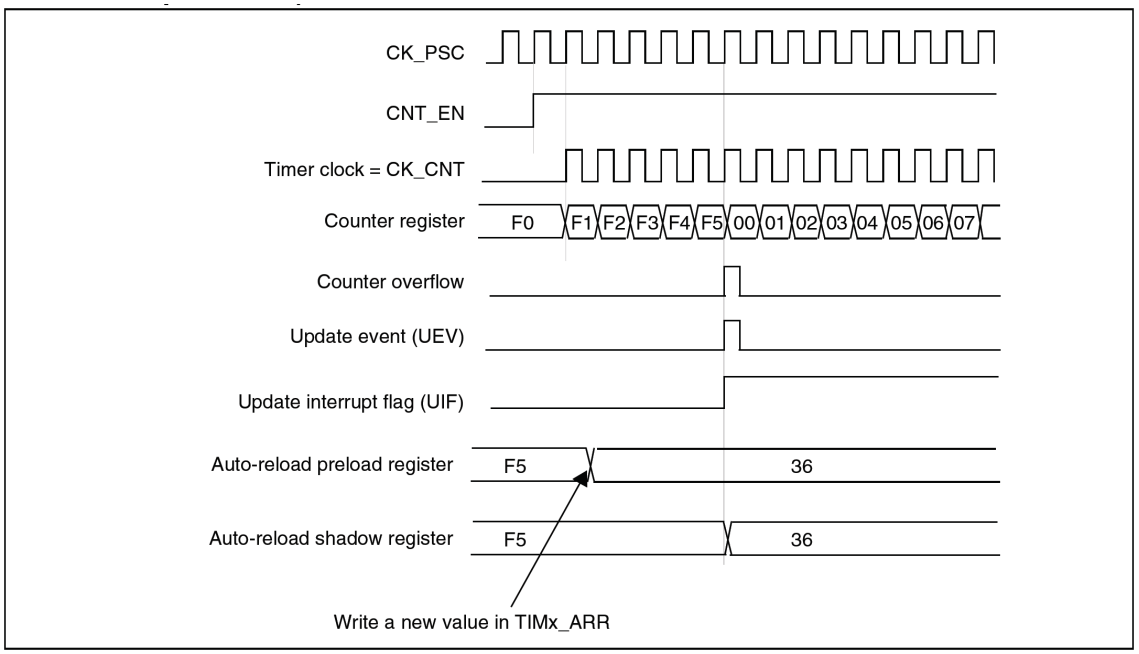


图 188. 计数器时序图，当 ARPE=1 时的更新事件 (预装载 TIMx_ARR)



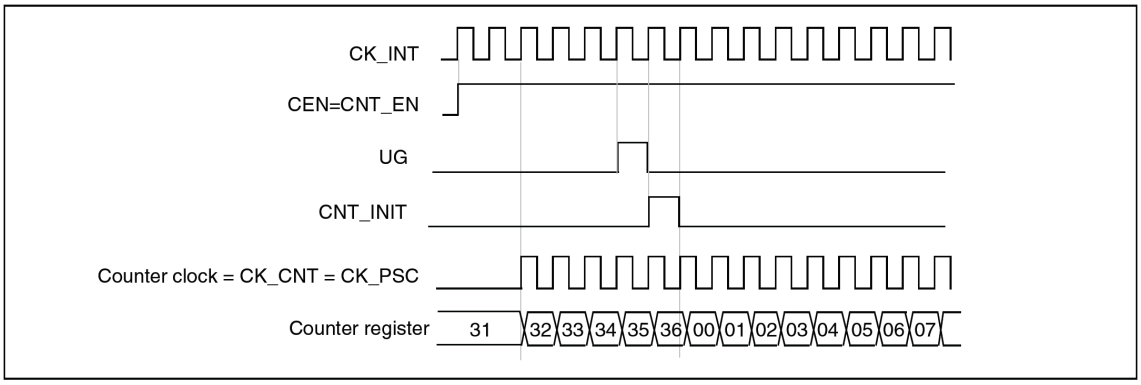
19.3.3 时钟源

计数器的时钟由内部时钟 (CK_INT) 提供。

TIMx_CR1 寄存器的 CEN 位和 TIMx_EGR 寄存器的 UG 位是实际的控制位，(除了 UG 位被自动清除外) 只能通过软件改变它们。一旦置 CEN 位为 ' 1' ，内部时钟即向预分频器提供时钟。

下图示出控制电路和向上计数器在普通模式下，没有预分频器时的操作。

图 189. 普通模式时序图，内部时钟分频系数为 1



19.3.4 调试模式

当微控制器进入调试模式 (Cortex-M3 核心停止) 时，根据 DBG 模块中的配置位 DBG_TIMx_STOP 的设置，TIMx 计数器或者继续计数或者停止工作。

19.4 TIM6 寄存器

有关寄存器描述中用到的缩写，请 31 页参考第 1.1 节。

可以用半字 (16 位) 或字 (32 位) 的方式操作这些外设寄存器。

19.4.1 TIM6 控制寄存器 1 (TIM6_CR1)

偏移地址 : 0x00

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
								rw				rw	rw	rw	rw

位 15:8 保留，读始终为‘0’

位 7 ARPE: 自动重装预装载使能（Auto-reload preload enable）

0: TIMx_ARR 寄存器无缓冲器

1: TIMx_ARR 寄存器有缓冲器

位 6:4 保留，读始终为‘0’

位 3 OPM: 单脉冲模式

0：在发生更新事件时，计数器不停止

1：在发生下次更新事件时，计数器停止计数 (清除 CEN 位)。

位 2 URS: 更新请求源 (Update request source)

该位由软件设置和清除，以选择 UEV 事件的请求源。

0：如果使能了中断或 DMA，以下任一事件可以产生一个更新中断或 DMA 请求：

- 计数器上溢或下溢
- 设置 UG 位
- 通过从模式控制器产生的更新

1：如果使能了中断或 DMA，只有计数器上溢或下溢可以产生更新中断或 DMA 请求。

位 1 UDIS: 禁止更新 (Update disable)

该位由软件设置和清除，以使能或禁止 UEV 事件的产生。

0：UEV 使能。更新事件 (UEV) 可以由下列事件产生：

- 计数器上溢或下溢
- 设置 UG 位
- 通过从模式控制器产生的更新产生更新事件后，带缓冲的寄存器被加载为预加载数值。

1：禁止 UEV。不产生更新事件 (UEV)，影子寄存器保持它的内容 (ARR、PSC)。但是如果设置 了 UG 位或从模式控制器产生了一个硬件复位，则计数器和预分频器将被重新初始化。

位 0 CEN: 计数器使能

0: 计数器禁止

1: 计数器使能

注：门控模式只能在软件已经设置了 CEN 位时有效，而触发模式可以自动地由硬件设置 CEN 位。

在单脉冲模式下，当产生更新事件时 CEN 被自动清除。

19.4.2 TIM6 控制寄存器 2 (TIM6_CR2)

偏移地址 : 0x04

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MMS[2:0]			Res.	Res.	Res.	Res.
									rw	rw	rw				

位 15:7 保留，始终保留为复位值

位 6:4 MMS: 主模式选择 (Master mode selection)

这些位用于选择在主模式下向从定时器发送的同步信息 (TRGO)，有以下几种组合：

000：复位 – 使用 TIMx_EGR 寄存器的 UG 位作为触发输出 (TRGO)。如果触发输入产生了复位 (从模式控制器配置为复位模式)，则相对于实际的复位信号，TRGO 上的信号有一定的延迟。

001：使能 – 计数器使能信号 CNT_EN 被用作为触发输出 (TRGO)。它可用于在同一时刻启动多个定时器，或控制使能从定时器的时机。计数器使能信号是通过 CEN 控制位和配置为门控模式时的触发输入的’逻辑或’产生。当计数器使能信号是通过触发输入控制时，在 TRGO 输出上会有一些延迟，除非选择了主 / 从模式 (见 TIMx_SMCR 寄存器的 MSM 位)。

010：更新 – 更新事件被用作为触发输出 (TRGO)。例如一个主定时器可以作为从定时器的预分频器使用。

位 3:0 保留，读始终为‘0’

19.4.3 TIM6 DMA/ 中断使能寄存器 (TIM6_DIER)

偏移地址 : 0x0C

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	UDE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIE
							rw								rw

位 15:9 保留，始终保持为复位值

位 8 UDE: 更新 DMA 请求使能

0: 更新 DMA 请求禁止

1: 更新 DMA 请求允许

位 7:1 保留，始终保持为复位值

位 0 UIE: 更新中断使能

0: 更新中断禁止

1: 更新中断允许

19.4.4 TIM6 状态寄存器 (TIM6_SR)

偏移地址 : 0x10

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIF
															rc_w0

位 15:1 保留，始终保持为复位值

位 0 UIF: 更新中断标志 (Update interrupt flag)

硬件在更新中断时设置该位，它由软件清除。

0：没有产生更新。

1：产生了更新中断。下述情况下由硬件设置该位：

- 计数器产生上溢或下溢并且 TIMx_CR1 中的 UDIS=0；
- 如果 TIMx_CR1 中的 URS=0 并且 UDIS=0，当使用 TIMx_EGR 寄存器的 UG 位重新初始化计数器 CNT 时。

19.4.5 TIM6 事件产生寄存器 (TIM6_EGR)

偏移地址 : 0x14

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UG
															w

位 15:1 保留，始终保持为复位值

位 0 UG: 产生更新事件 (Update generation)

该位由软件设置，由硬件自动清除。

0：无作用

1：重新初始化定时器的计数器并产生对寄存器的更新。注意：预分频器也被清除 (但预分频系 数不变)。

19.4.6 TIM6 定时器 (TIM6_CNT)

偏移地址 : 0x24

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 CNT[15:0]: 计数器值

19.4.7 TIM6 预分频器 (TIM6_PSC)

偏移地址 : 0x28

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 PSC[15:0]: 预分频器值

计数器的时钟频率 CK_CNT 等于 fCK_PSC/(PSC[15:0]+1)。

在每一次更新事件时，PSC 的数值被传送到实际的预分频寄存器中。

19.4.8 TIM6 自动重装寄存器 (TIM6_ARR)

偏移地址 : 0x2C

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 ARR[15:0]: 自动重装载数值 (Prescaler value)

ARR 的数值将传送到实际的自动重装载寄存器中。

关于 ARR 的更新和作用，详见 19.3.1。

如果自动重装载数值为空，则计数器停止。

19.4.9 TIM6 寄存器映射

下表中将 TIM6 的所有寄存器映射到一个 16 位可寻址 (编址) 空间。

表 57. TIM6 寄存器映射与位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**TIM6_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
	Reset value																									0				0	0	0	0
0x04	**TIM6_CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MMS[2:0]			Res.	Res.	Res.	Res.
	Reset value																										0	0	0				
0x08	**Reserved**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x0C	**TIM6_DIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UDE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIE
	Reset value																								0								0
0x10	**TIM6_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIF
	Reset value																																0
0x14	**TIM6_EGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Re	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UG
	Reset value																																0
0x18	**Reserved**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x1C	**Reserved**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x20	**Reserved**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x24	**TIM6_CNT**	Res.	Res.	Res.	Res.	Re	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	**TIM6_PSC**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**TIM6_ARR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

有关寄存器的起始地址，参见 35 页第 2.2.2 节

20 独立看门狗 (IWWDG)

20.1 简介

本器件集成了一个内嵌的看门狗外设，用集成的方式提供了高安全水平，精准的时间控制及运用的灵活性。 集成的看门狗外设是用来解决某些软件故障问题的，当它的定时计数值达到了预设的门限的时候，它会触发一个系统复位请求。

独立看门狗由它自己专有的低速时钟来驱动，因此就算是主时钟失效了，它仍然能保持工作状态。

该外设非常适合哪种需要在主程序以外还需要一个完全独立的看门狗，但对时钟精确度却又不太高的应用。 进一步的关于窗口看门狗的信息，请参见 456 页的第 21 章。

20.2 IWWDG 主要功能

- 自由运行的向下计数器
- 由一个独立的 RC 振荡器驱动 (在 Standby 和 Stop 状态下仍可操作)
- 复位条件
  - 当向下递减计数器的值达到 0 时产生复位请求。
  - 当向下递减计数器的值处于窗口值之外时，被重新加载就会导致复位请求。

20.3 IWWDG 功能描述

图 190 描述了独立看门狗模块的功能框图。

当向独立看门狗的关键字寄存器写入启动指令 0x0000CCCC 的时候，看门狗计数器开始由复位值 0xFFF 向下计数。
当计数值达到 0x000 的时候由独立看门狗发出复位信号。

任何时候将关键字 0x0000AAAA 写到 IWWDG_KR 寄存器中，都会使得 IWWDG_RLR 寄存器中的值被重加载到看门狗计数器中，从而阻止即将发生的复位动作。

20.3.1 窗口选项

IWWDG 也能够工作在窗口看门狗模式下，只要在 IWWDG_WINR 寄存器中设置适当的值即可。

如果重加载操作执行的同时，看门狗计数器的值超出了窗口寄存器 (IWWDG_WINR) 中存储的值，也会引起复位操作。

IWWDG_WINR 的默认值是 0x00000FFF, 所以如果没有改写它，那么窗口选项默认是关闭的。

窗口值一旦改变，立即就会引起看门狗计数器的一次重加载动作，将其置为 IWWDG_RLR 中所设置的值，从而一定程度上延缓目前到下次复位所需的时间周期。

当窗口选项使能时配置 IWWDG

1. 将 0x0000CCCC 写到 IWWDG_KR 寄存器，使能 IWWDG。
2. 向 IWWDG_KR 寄存器写 0x00005555 打开寄存器访问许可。
3. 向 IWWDG_PR 写 0 ～ 7 的值，以配置 IWWDG 的预分频器。
4. 配置重加载寄存器 (IWWDG_RLR)。
5. 等待状态寄存器 IWWDG_SR 的值更新为 0x00000000。
6. 配置窗口寄存器 IWWDG_WINR。 这将会引起自动将 IWWDG_RLR 的值更新到看门狗计数器。

注： 当 IWWDG_SR 的值为 0x00000000 时，写窗口值的动作会使 RLR 的值刷新计数器。

当窗口选项被禁止时配置 IWWDG

当窗口选项未被使用，可按下列顺序配置 IWWDG。

1. 向 IWWDG_KR 寄存器写 0x00005555 打开寄存器访问许可。
2. 向 IWWDG_PR 写 0 ～ 7 的值，以配置 IWWDG 的预分频器。
3. 配置重加载寄存器 (IWWDG_RLR)。
4. 等待状态寄存器 IWWDG_SR 的值更新为 0x00000000。
5. 将 IW_RLR 的值刷新到看门狗定时器 (IWWDG_KR = 0x0000 AAAA)。
6. 将 0x0000CCCC 写到 IWWDG_KR 寄存器，使能 IWWDG。

20.3.2 硬件看门狗

如果在 OPTION 字节中打开了 " 硬件看门狗 " 功能，那么在上电的时候看门狗就被自动打开。如果没有在看门狗计数器计数结束，或者向下计数的值超出窗口之前向关键字寄存器写入正确的值，那么一定会产生硬件复位请求。

20.3.3 寄存器访问保护

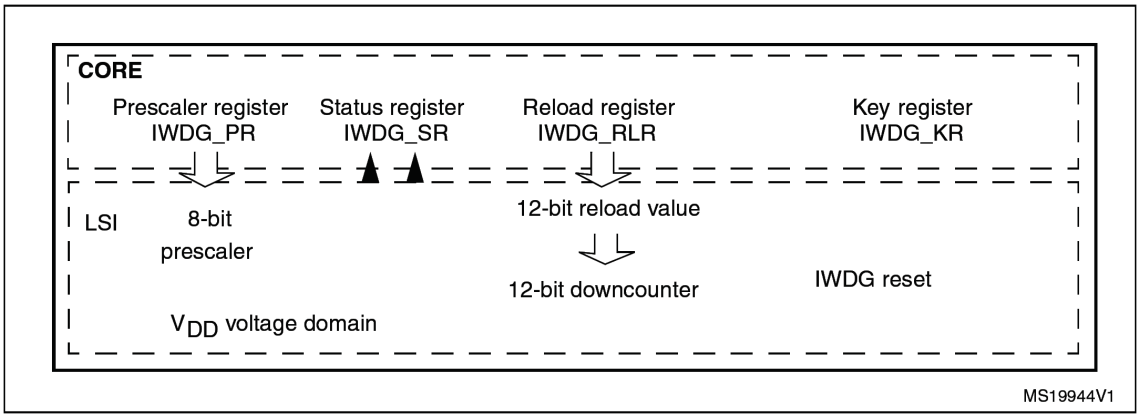
默认条件下，对 IWWDG_PR、IWWDG_RLR 和 IWWDG_WINR 的写访问操作都是受保护的。想要改变这一点，必须先向 IWWDG_KR 写入 0x0000 5555 解锁码。 如果写入别的值，将会打破这个顺序，使得对寄存器的访问保护重新生效。 这意味着在做重加载操作的时候（向该寄存器写入 0x0000 AAAA）就属于这种情况。

可以通过一个状态寄存器观察预分频器的更新、看门狗计数器的重加载或窗口值的重加载。

20.3.4 调试模式

当微控制器进入调试模式时 (内核被暂停)，看门狗计数器可以继续运行，也可以被停止。这取决于 DBG 模块中的 DBG_IWWDG_STOP 选项的配置。

图 190. 独立看门狗框图



注： 看门狗功能由内核供电区域供电，无论是在 Stop 模式下还是在 Standy 模式下。

20.4 IWWDG 寄存器

参见 31 页的章节 1.1 中对寄存器描述所采用的缩写列表 外设寄存器可以用 32 位或者 16 位的方式访问。

20.4.1 关键字寄存器 (IWWDG_KR)

地址偏移 : 0x00

复位值 : 0x0000 0000 (reset by Standby mode)

31	30	29	28	27	26	5	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:16 保留，必须保持复位时的值。

位 15:0 KEY[15:0]: 关键值（只写，读的话会是 0x0000）

这些位必须周期性的由软件写入 0xAAAA，否则当计数器向下计数到 0 的时候会产生硬件复位请求。

写入 0x5555 会使能对 IWWDG_PR,IWWDG_RLR 和 IWWDG_WINR 三个寄存器的访问许可。（参见章节 20.3.3）

写入 0xCCCC 启动看门狗（除非已经由选项字节在上电的时候就启动了它）

20.4.2 预分频寄存器(IWWDG_PR)

地址偏移 : 0x04

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PR[2:0]		
													rw	rw	rw

位 31:3 保留，必须保持复位时的值。

位 2:0 PR[2:0]: 预分频器

这些位平时处于写保护状态，参见章节 20.3.3 由软件写入，用来选择对输入时钟的预分频系数。 为了能够改变预分频器的分频系数，IWWDG_SR 寄存器中的 PVU 位必须先清零。

000: 四分频

001: 8 分频

010: 16 分频

011: 32 分频

100: 64 分频

101: 128 分频

110: 256 分频

111: 256 分频

注： 读这个寄存器会返回 VDD 供电区域的分频器的值。 如果处于写保护状态下，这个值不一定是最新的。 所以从这个地方读数据的时候要保证 IWWDG_SR 寄存器中的 PVU 位为 0 才行。

20.4.3 重加载寄存器(IWWDG_RLR)

地址偏移 : 0x08

复位值 : 0x0000 0FFF (reset by Standby mode)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	RL[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:12 保留，必须保持复位时的值。

位 11:0 RL[11:0]: 看门狗计数器重置值

这些位平时处于写保护状态，参见章节 20.3.3 这个值是由软件来设置的，并且每次向 IWWDG_KR 寄存器写入 0xAAAA 的时候，这个值会被更新到看门狗计数器中，如果想延时长一点，这个值就该大一些。 因为看门狗计数器正是从这个值开始向下计数。 定时的长度是由这个值和预分频器的设置值来共同决定的。 参见 datasheet 中关于超时的数据。

要想改变这个重加载值，必须先保证 IWWDG_SR 中的 RVU 位为 0。

注： 读这个寄存器会返回 VDD 供电区域的值。 如果处于写保护状态下，这个值不一定是最新的。 所以从这个地方读数据的时候要保证 IWWDG_SR 寄存器中的 RVU 位为 0 才行。

20.4.4 状态寄存器(IWWDG_SR)

地址偏移 : 0x0C

复位值 : 0x0000 0000 (not reset by Standby mode)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WVU	RVU	PVU
													r	r	r

位 31:3 保留，必须保持复位时的值。

位 2 WVU: 看门狗计数器窗口值更新

该位由硬件置位，用来表明正在更新窗口值。当 VDD 供电区域中完成了看门狗定时器的重加载时，该位会被硬件清零。（这需要 5 个 40KHz 的 RC 振荡周期）

只有当 WVU 的值为零时才能再次改写窗口值设置。

该位只在窗口功能打开的时候有效。

位 1 RVU: 看门狗计数器重置值更新

该位由硬件置位，用来表明正在更新定时器重置值。当 VDD 供电区域中完成了看门狗定时器的重加载时，该位会被硬件清零。（这需要 5 个 40KHz 的 RC 振荡周期）

只有当 RVU 的值为零时才能再次改写重置值设置。

位 0 PVU: 看门狗预分频器设置更新

该为由硬件置位，用来表明正在更新预分频器设置。 当 VDD 供电区域中完成了看门狗定时器的重加载时，该位会被硬件清零。（这需要 5 个 40KHz 的 RC 振荡周期）

只有当 WVU 的值为零时才能再次改写预分频器设置值。

20.4.5 窗口寄存器(IWWDG_WINR)

地址偏移 : 0x10

复位值 : 0x0000 0FFF (reset by Standby mode)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	WIN[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:12 保留，必须保持复位时的值。

位 11:0 WIN[11:0]: 看门狗计数器窗口值

这些位平时处于写保护状态，参见章节 20.3.3 这些位包含的是窗口值和向下计数器的比较上限。

为了阻止复位信号的产生，必须在向下计数器递减到窗口值和 0 之间的某个值时重加载它。

要想改变这个窗口值，必须先保证 IWWDG_SR 中的 WVU 位为 0。

注： 读这个寄存器会返回 VDD 供电区域的值。 如果刚刚做过写操作就读，这个值不一定是对的。 所以从这个地方读数据的时候要保证 IWWDG_SR 寄存器中的 WVU 位为 0 才行。

注： 如果重加载设置、预分频设置和窗口设置都已经生效，那么再次改变它们前一定要确认相关的标志位为零，那表明他们没有处于正在被加载的过程中。 更新这些设置之后却不需要等着三个标志位清零，除非马上要进入低功耗模式。

20.4.6 IWWDG 寄存器镜像

下表给出了 IWWDG 寄存器镜像和复位值。

表 58. CRC 寄存器镜像和复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**IWWDG_KR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEY[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	**IWWDG_PR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PR[2:0]		
	Reset value																														0	0	0
0x08	**IWWDG_RLR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RL[11:0]											
	Reset value																					1	1	1	1	1	1	1	1	1	1	1	1
0x0C	**IWWDG_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WVU	RVU	PVU
	Reset value																														0	0	0
0x10	**IWWDG_WINR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WIN[11:0]											
	Reset value																					1	1	1	1	1	1	1	1	1	1	1	1

寄存器边界地址请参见 35 页的章节 2.2.2

21 窗口看门狗 (WWDG)

21.1 WWDG 简介

窗口看门狗通常被用来监测，由外部干扰或不可预见的逻辑条件造成的应用程序背离正常的运行序列而产生的软件故障。除非递减计数器的值在 T6 位变成 0 前被刷新，看门狗电路在达到预置的时间周期时，会产生一个 MCU 复位。在递减计数器达到窗口寄存器数值之前，如果 7 位的递减计数器数值 (在控制寄存器中) 被刷新， 那么也将产生一个 MCU 复位。这表明递减计数器需要在一个有限的时间窗口中被刷新。

WWDG 时钟从 APB1 时钟预分频，并有一个可配置的时间窗口，这可通过编程来检测异常推迟或提前的应用行为。

WWDG 最适合要求看门狗在一个精确时间窗口内做出反应的应用程序。

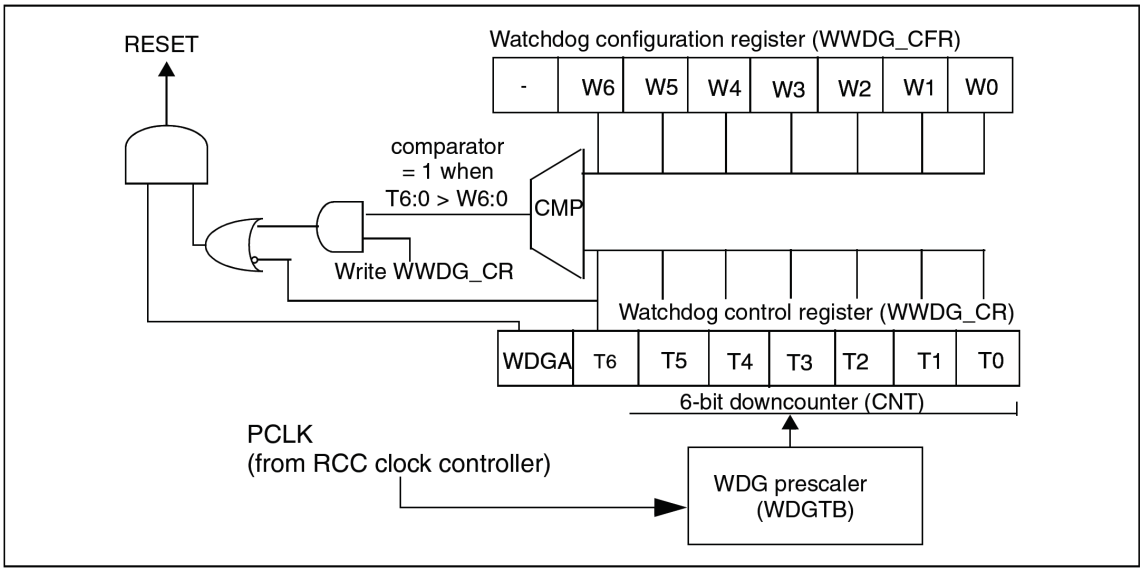
21.2 主要特性

- 可编程的自由运行递减计数器
- 复位条件
  - 当递减计数器的值小于 0x40，(若看门狗被启动) 则产生复位。
  - 当递减计数器在窗口外被重新装载，(若看门狗被启动) 则产生复位。见图 192。
- 提前唤醒中断 (EWI)：如果启动了看门狗，当递减计数器等于 0x40 时产生提前唤醒中断。

21.3 功能描述

如果看门狗被启动 (WWDG_CR 寄存器中的 WDGA 位被置’1’)， 并且当 7 位 (T[6:0]) 递减计数器从 0x40 翻转到 0x3F(T6 位清零) 时，则产生一个复位。如果软件在计数器值大于窗口寄存器中的数值时重新装载计数器，将产生一个复位。

图 191. 看门狗框图



应用程序在正常运行过程中必须定期地写入 WWDG_CR 寄存器以防止 MCU 发生复位。只有当计数器值小于窗口寄存器的值时，才能进行写操作。储存在 WWDG_CR 寄存器中的数值必须在 0xFF 和 0xC0 之间：

启动看门狗

在系统复位后，看门狗总是处于关闭状态，设置 WWDG_CR 寄存器的 WDGA 位能够开启看门狗，随后它不能再被关闭，除非发生复位。

控制递减计数器

递减计数器处于自由运行状态：即使看门狗被禁止，递减计数器仍继续递减计数。当看门狗被启用时，T6 位必须被设置，以防止立即产生一个复位。

T[5:0] 位包含了看门狗产生复位之前的计时数目；复位前的延时时间在一个最小值和一个最大值之间变化，这是因为写入 WWDG_CR 寄存器时，预分频值是未知的（见图 192）。 配置寄存器 (WWDG_CFR) 中包含窗口的上限值：要避免产生复位，递减计数器必须在其值小于窗口寄存器的数值并且大于 0x3F 时被重新装载，图 192 描述了窗口寄存器的工作过程。

注： *可以用 T6 位产生一个软件复位 (设置 WDGA 位为’1’，T6 位为’0’)。*

看门狗中断高级特性

如果在实际复位产生之前必须进行特定的安全操作或数据记录，可以用提前唤醒中断。设置 WWDG_CFR 寄存器中的 WEI 位开启该中断。在复位之前当递减计数器到达 0x40 时，则产生此中断，同时可以用相应的中断服务程序 (ISR) 来触发特定的行为（例如通信或数据记录）

在某些应用中，提前唤醒中断可以用来管理软件系统检测和 / 或系统恢复 / 故障弱化，但不产生 WWDG 复位。在这种情况下，相应的中断服务程序（ISR）将重加载 WWDG 计数器，以避免 WWDG 复位，然后触发必要的行为。

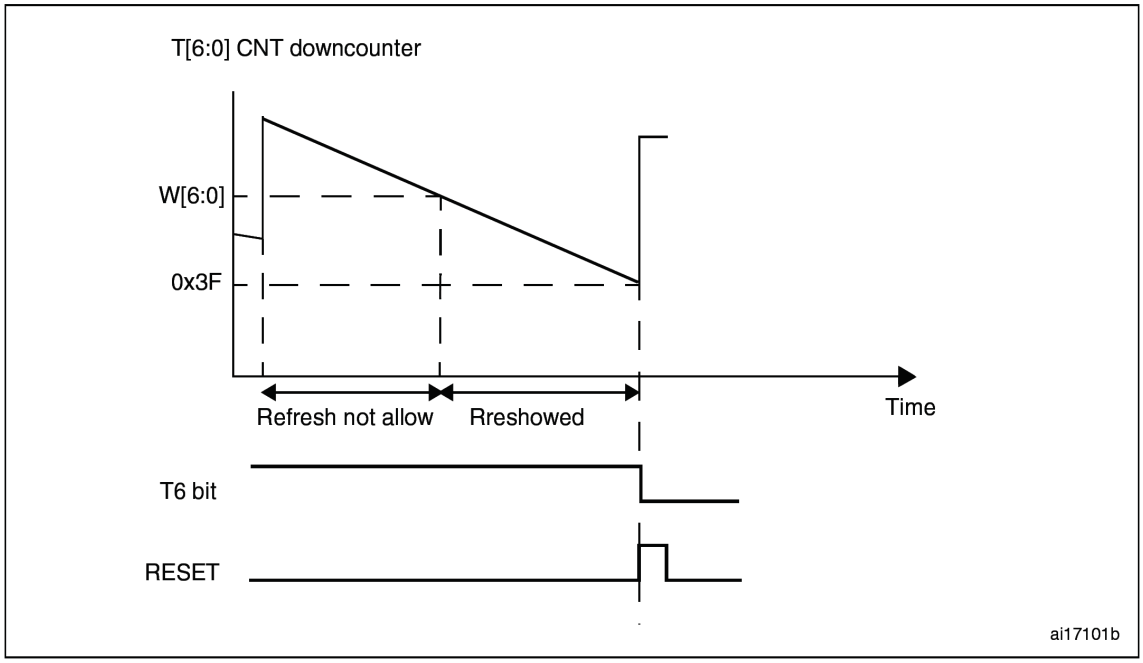
注： *当提前唤醒中断无法启用时，例如由于系统锁定在更高优先级任务，最终将产生 WWDG 复位。*

21.4 如何设置看门狗超时

可以使用图 192 提供的公式计算窗口看门狗的超时时间。

警告：当写入 WWDG_CR 寄存器时，始终置 T6 位为’1’以避免立即产生一个复位。

图 192. 窗口看门狗时序图



计算超时的公式如下：

$$t_{WWDG} = t_{PCLK1} \times 4096 \times 2^{WDGTB} \times (t[5:0] + 1) \quad (ms)$$

其中：

t_{WWDG}: WWDG 超时时间

t_{PCLK}: APB1 以 ms 为单位的时钟周期

T_{WWDG} 的最大值和最小值请参考数据手册。

21.5 调试模式

当微控制器进入调试模式时 (Cortex-M3 核心停止)，根据调试模块中的 DBG_WWDG_STOP 配置位的状态，WWDG 的计数器可以继续正常工作或停止。

21.6 WWDG registers 寄存器

关于在寄存器描述里面所用到的缩写，详见第 31 页第 1.1 节。可以用半字 (16 位) 或字 (32 位) 的方式操作这些外设寄存器。

21.6.1 Control register (WWDG_CR) 控制寄存器

地址偏移 : 0x00

复位值 : 0x0000 007F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDGA	T[6:0]						
								rs	rw						

位 31:8 保留。

位 7 WDGA: 激活位

此位由软件置’1’，但仅能由硬件在复位后清’0’。当 WDGA=1 时，看门狗可以产生复位。

0：禁止看门狗

1：启用看门狗

位 6:0 T[6:0]: 7-bit 计数器 (MSB 到 o LSB)

这些位用来存储看门狗的计数器值。每 (4096×2^{WDGTB}) 个 PCLK1 周期减 1。当计数器值从 0x40 变为 0x3F 时 (T6 变成 0)，产生看门狗复位。

21.6.2 配置寄存器 (WWDG_CFR)

地址偏移 : 0x04

复位值 : 0x0000 007F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	EWI	WDGTB[1:0]		W[6:0]						
						rs	rw		rw						

位 31:10 保留。

位 9 EWI: 提前唤醒中断

此位若置’1’，则当计数器值达到 0x40，即产生中断。 此中断只能由硬件在复位后清除。

位 8:7 WDGTB[1:0]: 时基

预分频器的时基可以设置如下：

00: CK 计时器时钟 (PCLK1 除以 4096) 除以 1

01: CK 计时器时钟 (PCLK1 除以 4096) 除以 2

10: CK 计时器时钟 (PCLK1 除以 4096) 除以 4

11: CK 计时器时钟 (PCLK1 除以 4096) 除以 8

位 6:0 W[6:0]: 7-bit 窗口值

这些位包含了用来与递减计数器进行比较用的窗口值。

21.6.3 Status register (WWDG_SR) 状态寄存器

地址偏移 : 0x08

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EWIF
															rc_w0

位 31:1 保留。

位 0 EWIF: 提前唤醒中断标志

当计数器的值达到 0x40 时，此位由硬件置’1’。它必须通过软件写’0’来清除。对此位写’1’无效。若中断未被启用，此位也会被置’1’。

21.6.4 WWDG 寄存器映像

下表是 WWDG 寄存器映像和复位值

表 59. WWDG 寄存器映像和复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**WWDG_CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDGA	T[6:0]						
	Reset value																									0	1	1	1	1	1	1	1
0x04	**WWDG_CFR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EWI	WDGTB1	WDGTB0	W[6:0]						
	Reset value																							0	0	0	1	1	1	1	1	1	1
0x08	**WWDG_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EWIF
	Reset value																																0

寄存器边界地址请参考第 35 页 2.2.2.

22 实时时钟 (RTC)

22.1 简介

实时时钟（RTC）是一个独立的 BCD 定时器 / 计数器。RTC 模块拥有一个具有可编程报警中断功能的时间日历时钟。

RTC 模块拥有自动唤醒功能，用于管理所有的低功耗模式。

两个 32 位寄存器以 BCD 格式存储秒、分、时（12/24 小时制）、日（星期数）、日期、月和年。亚秒值同样可用二进制存储。

RTC 具有自动月份天数补偿功能，还包括夏令时间补偿。

单独使用一个 32 位寄存器存储可编程报警相关信息，包括亚秒、秒、分、时、日、日期。

对由晶体本身的频偏、温度漂移及其他原因引起的任何误差，可以利用 RTC 本身的数字校准功能进行修正。

上电复位后，所有 RTC 寄存器将被禁止访问，以防止意外的写操作。

当设备处于运行模式、低功耗模式，或复位状态，只要电压在工作范围内， RTC 将保持正常运行。

22.2 主要特性

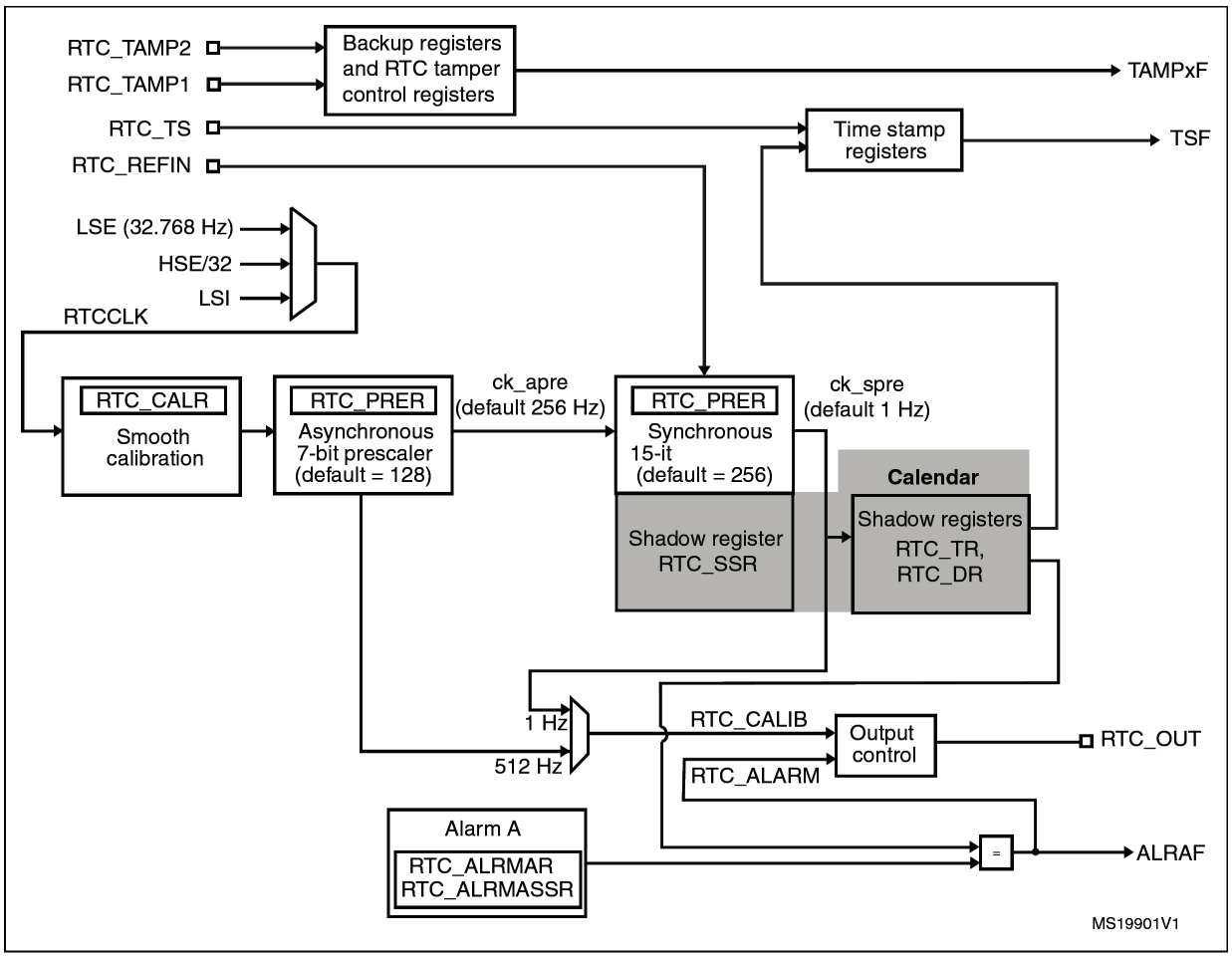
RTC 模块主要特性如下（见图 193：RTC 框图）：

- 日历功能，可显示亚秒、秒、分、时（12/24 小时制）、日（星期）、日期、月和年。
- 通过软件编程实现夏令时补偿。
- 可编程报警中断功能。报警设定可由任何日历字段来触发。
- 参考时钟检测：采用更高精度的时钟源（50 或 60Hz），用于扩展日历精度。
- 采用一个亚秒级外部时钟实现精确同步。
- 数字校准电路（计数器定期修正）：0.95ppm 精准度，得自一个几秒钟的校准窗口
- 事件记录时间戳。
- 侵入检测事件：带可配置的滤波器和内部上拉电阻。
- 可屏蔽中断 / 事件：
  - Alarm A
  - 时间戳
  - 侵入检测
- 备份寄存器 : 当产生一个侵入检测事件，备份寄存器被复位。

22.3 功能描述

22.3.1 RTC 框图

图 193. RTC 框图



RTC 模块包括 :

- 一个报警
- 两个侵入事件
- 5 个 32bit 备份寄存器
- 复用功能输出 : RTC_OUT 以下述两种形式中任意一种形式 输出 :
  - RTC_CALIB: 512 Hz 或 1Hz 时钟输出 (用 32.768 kHz 的 LSE 的时候). 用 RTC_CR 寄存器中 COE 位来打开这个输出。
  - RTC_ALARM: Alarm A. 这个输出用 RTC_CR 寄存器中 OSEL[1:0] 位来打开。
- 复用功能输入 :
  - RTC_TS : 时间戳事件
  - RTC_TAMP1: 侵入事件检测 1
  - RTC_TAMP2: 侵入事件检测 2
  - RTC_REFIN: 50 或 60 Hz 参考时钟输入

RTC_OUT, RTC_TS 和 RTC_TAMP1 映射到同一个引脚（PC13）.

通过 RTC_TAFCR 寄存器完成 RTC_ALARM 输出的选择，其中 PC13VALUE 位用于选择 RTC_ALARM 输出是推挽模式或开漏模式。

当 PC13 不用作 RTC 复用功能时，可通过设置 RTC_TAFCR 中 PC13MODE 位将 PC14 强制为推挽输出模式，输出值由 PC13VALUE 位给出。此时，PC13 的推挽输出状态和值在待机模式下是可以保持的。
输出机制遵循固定的优先顺序（见表 60）。

当 PC14 和 PC15 不用作 LSE 振荡器时，可通过分别设置 RTC_TAFCR 寄存器中 PC14MODE 位和 PC15MODE 位将 PC14 和 PC15 强制为推挽输出模式，输出值由 PC14VALUE 和 PC15VALUE 给出。此时，PC14 和 PC15 的推挽输出状态和值在待机模式下是可保持的。
输出机制遵循固定的优先顺序（见表 61 和表 62）。

表 60. RTC pin PC13 配置 (1)

引脚配置和功能	RTC_ALARM 输出使能	RTC_CALIB 输出使能	RTC_TAMP1 输入使能	RTC_TS 输入使能	PC13MODE 位	PC13VALUE 位
RTC_ALARM output OD	1	无影响	无影响	无影响	无影响	0
RTC_ALARM output PP	1	无影响	无影响	无影响	无影响	1
RTC_CALIB output PP	0	1	无影响	无影响	无影响	无影响
RTC_TAMP1 input floating	0	0	1	0	无影响	无影响
RTC_TS and RTC_TAMP1 input floating	0	0	1	1	无影响	无影响
RTC_TS input floating	0	0	0	1	无影响	无影响
Output PP forced	0	0	0	0	1	PC13 输出数据值
Wakeup pin or Standard GPIO	0	0	0	0	0	无影响

1. OD：开漏，open drain 的缩写；PP：推挽，push-pull 的缩写。

表 61. LSE pin PC14 配置 (1)

引脚配置和功能	RCC_BDCR 寄存器 LSEON 位	RCC_BDCR 寄存器 LSEBYP 位	PC14MODE 位	PC14VALUE 位
LSE oscillator	1	0	无影响	无影响
LSE bypass	1	1	无影响	无影响

表 61. LSE pin PC14 配置 (续)(1)

引脚配置和功能	RCC_BDCR 寄存器 LSEON 位	RCC_BDCR 寄存器 LSEBYP 位	PC14MODE 位	PC14VALUE 位
Output PP forced	0	无影响	1	PC14 输出数据值
Standard GPIO	0	无影响	0	无影响

1. OD：开漏，open drain 的缩写；PP：推挽，push-pull 的缩写。

表 62. LSE pin PC15 配置 (1)

引脚配置和功能	RCC_BDCR 寄存器 LSEON 位	RCC_BDCR 寄存器 LSEBYP 位	PC15MODE 位	PC15VALUE 位
LSE oscillator	1	0	无影响	无影响
Output PP forced	0	无影响	1	PC15 输出数据值
Standard GPIO	0	无影响	0	无影响

1. OD：开漏，open drain 的缩写；PP：推挽，push-pull 的缩写。

22.3.2 时钟和预分频器

由时钟控制器从以下 3 种时钟中选择 RTC 时钟源（RTCCLK）：

- LSE 振荡器作为 RTC 时钟；
- LSI 振荡器作为 RTC 时钟；
- HSE 振荡器作为 RTC 时钟。

更多有关 RTC 时钟源的配置信息，请参考第 7 章 — 复位和时钟控制（RCC）

一个可编程预分频器产生一个 1HZ 的时钟，用于更新日历。为最大限度地减少功耗，预分频器可分割成 2 个可编程预分频器。（见图 193：RTC 框图）：

- 通过控制 RTC_PRER 寄存器中 PREDIV_A 位来配置 7 位异步预分频器。
- 通过控制 RTC_PRER 寄存器中 PREDIV_S 位来配置 15 位同步预分频器。

注： 当所有分频器被启用时，建议将异步预分频器的值调高以最大限度地减少功率消耗

异步预分频器的分频系数设为 128，同步预分频器的分频系数设为 256，从而得到一个基于 32.768 kHz LSE 频率的 1 Hz (ck_spre) 内部时钟频率。

最小分频系数为 1，最大分频系数为 2^{22}，相当于最大输入频率大约 4MHz。

f_{ck_apre} 满足：

$$f_{CK\_APRE} = \frac{f_{RTCCLK}}{PREDIV\_A + 1}$$

The ck_apre 时钟为二进制 RTC_SSR 亚秒递减计数器提供时钟。当值为 0 时，RTC_SSR 的值将被重置为 PREDIV_S 里的值。

f_{ck_spre} 满足

$$f_{CK\_SPRE} = \frac{f_{RTCCLK}}{(PREDIV\_S + 1) \times (PREDIV\_A + 1)}$$

22.3.3 实时时钟和日历

RTC 的日历时间寄存器和日期寄存器是通过影子寄存器访问的，该影子寄存器与 PCLK (APB clock) 同步。 为避免同步等待时间，也可直接访问。

- RTC_SSR ：亚秒
- RTC_TR ：时间
- RTC_DR：日期

硬件会每两个 RTCCLK 时钟周期更新一次影子寄存器的日历值，并将 RTC_ISR 寄存器的 RSF 位置 1（见 22.6.4）。停止或待机模式下则不作这个更新，当退出上述两种模式后，影子寄存器将在最多两个 RTCCLK 周期后执行更新操作。

默认情况下，应用程序通过访问影子寄存器获取日历寄存器的内容。如需直接访问日历寄存器，可通过设置 RTC_CR 寄存器的 BYPSHAD 控制位（默认为零）来实现。

在 BYPSHAD=0 模式下，如需读取 RTC_SSR, RTC_TR 或 RTC_DR 寄存器的内容，APB 时钟频率 (fAPB) 至少是 RTC 时钟频率 (fRTCCLK) 的 7 倍。

系统复位后，影子寄存器也将自动复位。

22.3.4 可编程报警

RTC 模块拥有可编程报警功能：Alarm A

设置 RTC_CR register 寄存器的 ALRAE 位，启动可编程报警功能。当日历中亚秒 , 秒 , 分 , 时 , 日期或星期与报警寄存器 RTC_ALRMASSR 和 RTC_ALRMAR 中设定的值匹配，ALRAF 由硬件置 1 。所有日历字段都可以通过 RTC_ALRMAR 寄存器的 MSKx 位和 RTC_ALRMASSR 寄存器的 MASKSSx 位独立的被选择为报警源。设置 RTC_CR 寄存器的 ALRAIE 位，会产生报警中断。

注： 当“秒”字段被选中时，为确保正常运行，RTC_PRER 寄存器中同步预分频器的分频系数应*≥ 3*。

设置 RTC_CR 寄存器的 OSEL[0:1] 启动 Alarm A，可同步输出至 RTC_ALARM。RTC_ALARM 输出极性可由 RTC_CR 寄存器的 POL 位设置。

22.3.5 RTC 初始化及配置

访问 RTC 寄存器

RTC 寄存器是 32 位寄存器。除了 BYPSHAD=0 时对日历寄存器执行读操作外，APB 接口为其他对 RTC 寄存器的访问提供了 2 种等待状态。

RTC 寄存器的写保护

上电复位后，所有 RTC 寄存器将处于写保护状态。通过向写保护寄存器 RTC_WPR 写入指定关键字来启动 RTC 寄存器的写权限。

通过以下操作解除所有 RTC 寄存器（RTC_ISR[13:8],RTC_TAFCR 和 RTC_BKPxR 除外）的写保护。

1. 向 RTC_WPR 寄存器写入‘0xCA’；
2. 向 RTC_WPR 寄存器写入‘0x53’。

如果写入错误将重新激活写保护。
系统复位不影响写保护机制。

日历初始化及配置

按照以下顺序完成时间和日期值的初始化，包括时间格式和预分频器的配置：

1. RTC_ISR 寄存器的 INIT 位置“1”，进入初始初始化模式。在该模式下日历计数器暂停运行，此时可更新计数器的值。
2. 等待 RTC_ISR 寄存器的 INITF 位置“1”，确保已经正式进入初始化模式。由于时钟同步的延迟，该过程大约需要 2 个 RTCCLK 时钟周期。
3. 为了使日历计数器生成一个 1 Hz 的时钟，编写 RTC_PRER 寄存器中的分频系数。
4. 将初始时间和日期值加载到影子寄存器 (RTC_TR and RTC_DR)，通过 RTC_CR 寄存器的 FMT 位设置时间格式（12 小时制 /24 小时制）。
5. 清除 INIT 位的值退出初始化模式。日历计数器的实际值将会自动加载，并在 4 个 RTCCLK 时钟周期后重新启动。

在完成上述一系列初始化操作后，日历将开始计时。

注： 1. 系统复位后，应用程序可读取 RTC_ISR 寄存器的 INITS 标志，从而检测日历是否已经被初始化。如果标志位为“0”说明“年”字段恢复成其上电复位后的默认值 0x00，此时日历没被初始化。

2. 初始化后如果需要读日历，首先需要检测 RTC_ISR register 寄存器的 RSF 标志位是否为 1。

夏令时

通过 RTC_CR 寄存器的 SUB1H, ADD1H 和 BKP 位管理夏令时间。
通过设置 SUB1H 或 ADD1H 位，软件可在不通过初始化流程的情况下将日历中的时间单次增加 / 减少 1 小时。
此外，软件可通过 BKP 位记录该操作。

可编程报警

编程或更新可编程报警时，应遵循以下几点：

1. 清除 RTC_CR 的 ALRAE 位禁用 Alarm A；
2. 编程 Alarm A 寄存器 (RTC_ALRMASSR/RTC_ALRMAR)；
3. 设置 RTC_CR 的 ALRAE 位重新启用 Alarm A。

注： 由于时钟同步的延迟，所有对 RTC_CR 寄存器的更新将在大约 2 个 RTCCLK 时钟周期后正式有效。

22.3.6 读日历寄存器

- **当 RTC_CR 寄存器的 BYPSHAD 控制位被清除时：**

为确保在安全同步机制下正常读 RTC 日历寄存器 (RTC_SSR, RTC_TR 和 RTC_DR)，APB1 时钟频率 (fPCLK) 应至少为 RTC 时钟频率 (fRTCCLK) 的 7 倍以上。

当 APB1 时钟频率低于 7 倍 RTC 时钟频率时，软件必须两次读取日历时间和日期寄存器。如果第二次读取 RTC_TR 返回的值与第一次返回值相同，说明返回值是正确的。否则需再次读取。任何情况下， APB1 时钟频率都必须大于 RTC 时钟频率。

日历寄存器的内容被复制到 RTC_TR 和 RTC_DR 影子寄存器中时，RTC_ISR 寄存器的 RSF 位被置位。复制操作每两个 RTCCLK 周期执行一次。为确保三者的值保持一致，读 RTC_SSR 或 RTC_TR，锁定高阶日历影子寄存器中的值，直至 RTC_DR 中的值被读取。为避免软件在时间间隔少于 2 个 RTCCLK 周期的情况下多次访问日历，每次读日历 RSF 位应由软件清零，软件必须等待 RSF 位被置位后才能读 RTC_SSR，RTC_TR 和 RTC_DR 寄存器。

从低功耗模式（关机或待机）唤醒后，RSF 位应由软件清零，软件必须等待 RSF 位被再次置位后才能读 RTC_SSR，RTC_TR 和 RTC_DR 寄存器。

RSF 位应在唤醒后被清除，而不是进入低功耗模式前。

系统复位后，软件必须等待 RSF 位被置位后才能读 RTC_SSR，RTC_TR 和 RTC_DR 寄存器。事实上系统复位将导致影子寄存器复位至其默认值。

初始化（参考第 468 页：日历初始化及配置）后，软件必须等待 RSF 位被置位后才能读 RTC_SSR，RTC_TR 和 RTC_DR 寄存器。

同步（参考 22.3.8 节：RTC 同步）后，软件必须等待 RSF 位被置位后才能读 RTC_SSR，RTC_TR 和 RTC_DR 寄存器。

- 当 RTC_CR 寄存器的 BYPSHAD 控制位被置位时 (无需考虑影子寄存器):

读日历寄存器，直接从日历计数器获取值，无需等待 RSF 位被置位。此功能在刚退出低功耗模式（停止或待机模式）时非常有用，因为影子寄存器在低功耗模式下不会自动更新。

在 BYPSHAD 为“1”时，如果两次读寄存器之间出现 RTCCLK 边沿，不同寄存器中的结果可能会互不相关。此外，如果在读操作过程中遇到 RTCCLK 边沿，则某个寄存器的值可能不正确。软件必须读取所有的寄存器两次，并比较两次读取的结果，或者通过比较两组最低有效日历寄存器的结果，以检验数据是否正确且有一定关联。

注： BYPSHAD=1 时 , 读日历寄存器的指令的完成需另加一个额外的 APB 周期。

22.3.7 复位过程

任何可用的系统复位源都将导致日历影子寄存器 (RTC_SSR，RTC_TR 和 RTC_DR) 和 RTC 状态寄存器 (RTC_ISR) 复位至默认值。

然而，下列寄存器的复位于系统复位无任何关联，只与上电复位有关：RTC 当前日历寄存器、RTC 控制寄存器 (RTC_CR)、预分频器寄存器 (RTC_PRER)、RTC 校准寄存器 (RTC_CALR)、RTC 移位寄存器 (RTC_SHIFTR)、RTC 时间戳寄存器 (RTC_TSSSR，RTC_TSTR 和 RTC_TSDR)、RTC 侵入和复用功能配置寄存器 (RTC_TAFCR)、RTC 备份寄存器 (RTC_BKPxR)、Alarm A 寄存器 (RTC_ALRMASSR/RTC_ALRMAR).

除上电复位外，任何系统复位时 RTC 将维持运行状态。发生上电复位后，RTC 停止运行，所有 RTC 寄存器复位至默认值。

22.3.8 RTC 同步

RTC 可与一个高精度远程时钟同步。在读取亚秒字段 (RTC_SSR 或 RTC_TSSSR) 后，可计算出远程时钟和 RTC 中时间的精确偏差值。RTC 可以通过 RTC_SHIFTR 寄存器用时钟移位的方式瞬间剔除这个偏差。

RTC_SSR 用于存储同步预分频器的计数器值。这允许用 1/ (PREDIV_S + 1) 秒的分辨率去衡量 RTC 所保持的实际时间。因此，这个分辨率可以通过增加同步预分频值的方式提高 (PREDIV_S[14:0]. 允许的最大分辨率为 (32768 Hz 时钟时的 30.52 µs) 得自 PREDIV_S 被设为 0x7FFF 的情况。

然而，如果 PREDIV_S 增加，为了保证同步预分频器的输出为 1 Hz，PREDIV_A 必须减少。在该情况下，同步预分频器的输出频率将增加，同时动态功耗也会增加。

通过 RTC 移位控制寄存器 (RTC_SHIFTR) 可以微调 RTC。写 RTC_SHIFTR 寄存器可能移位（延迟或者提前）时钟信号，在分辨率为 1 / (PREDIV_S + 1) 秒的时候幅度可达 1 秒 . 移位操作包括将 SUBFS[14:0] 的值加到同步预分频计数器 SS[15:0]: 这将延迟时钟信号 . 如果同时 ADD1S 位是 1，结果会是加上一个整秒同时再减去不到 1 秒，所以这会提前时钟信号。

注： *在启动移位操作前，用户必须检查 SS[15] 是否为“0”以确保不会发生溢出。*

写 RTC_SHIFTR 寄存器启动移位操作后，通过硬件设置 SHPF 标志，表示一个移位操作正在等待。当移位操作完成后该位被硬件清除。

注： *同步功能与参考时钟检测功能不兼容，因此当 REFCKON=1 时，固件不能写 RTC_SHIFTR 。*

22.3.9 RTC 参考时钟检测

RTC_REFIN 参考时钟 (50Hz 或 60 Hz) 与 32.768 kHz LSE 时钟比较，具有更高的精确度。RTC_REFIN 检测启用后 (RTC_CR 的 REFCKON 位置“1”)，将用于补偿日历更新频率 (1 Hz) 的偏移。

每个 1Hz 的时钟沿会和最近的 RTC_REFIN 时钟沿（如果在给定的时间窗口中可以找到）进行比较。多数情况下两个时钟沿会恰好对齐。当发现对得不齐说明 LSE 不精确，RTC 将 1Hz 时钟移动一个最小位，从而下一个沿就就能对齐。得益于这种机制使得这个万年历的精度变得和参考时钟一样。

当参考时钟停止时，日历将完全遵照 LSE 时钟进行更新。RTC 等待参考时钟，并产生一个围绕 ck_spre 边沿的检测窗口。

启动 RTC_REFIN 检测后，PREDIV_A 和 PREDIV_S 必须复位至默认值：

- PREDIV_A = 0x007F
- PREVID_S = 0x00FF

注： *RTC_REFIN 时钟检测在待机模式下禁用。.*

22.3.10 RTC 平滑数字校准

RTC 频率可以按大约 0.954ppm 的分辨率进行校准，范围从 -487.1 ppm 到 +488.5 ppm。 通过一系列微调（如增加 / 减少单独的 RTCCLK 脉冲）进行频率校正。这些校准会分布得相当好，所以 RTC 会校准得很好，即便是再观察一段时间也一样。

在一个约 2^{20} RTCCLK 脉冲周期或 32 秒（输入频率为 32768 Hz 时）内，执行平滑数字校准。平滑校准寄存器 (RTC_CALR) 将明确记录在 32 秒周期中屏蔽的 RTCCLK 时钟周期的数目：

- 设置 CALM[0] 位为“1”，在 32 秒周期中一个脉冲被屏蔽。
- 设置 CALM[1] 位为“1”，另外两个周期被屏蔽。
- 设置 CALM[2] 位为“1”，另外四个周期被屏蔽。
- 以此类推，直至设置 SMC[8] 位为“1”，将有 256 个时钟被屏蔽。

最佳分辨率的情况下设置 CALM 可使 RTC 频率最多减少 487.1 ppm，通过设置 CALP 位可使频率增加 488.5 ppm。这时设置 CALP 为 1 的效果就是每 211 RTCCLK 周期插入一个额外的 RTCCLK 脉冲，也就意味着每 32 秒钟会插入 512 个时钟周期。

同时使用 CALM 和 CALP，每 32 秒可以插入 -511 到 +512 的 RTCCLK 时钟周期，换算成校准范围就是 -487.1 ppm 到 +488.5 ppm，校准步长为 0.954ppm。

下列公式用于计算有效校准频率 (F_{CAL})：

$F_{CAL} = F_{RTCCLK} \times [1 + (CALP \times 512 - CALM) / (2^{20} + CALM - CALP \times 512)]$

校准：当 PREDIV_A<3

当同步预分频器值 (RTC_PRER 寄存器的 PREDIV_A 位) 小于 3， CALP 位不能设为“1”。如果 CALP 已经被设为“1”，同时 PREDIV_A<3，则直接将 CALP 视为“0”，校准正常执行。

当 PREDIV_A<3 时执行一个校准，同步预分频器的值 (PREDIV_S) 应减少，以确保所有秒均加速 8 个 RTCCLK 时钟周期，相当于每 32 秒增加了 256 个时钟周期。因此在每 32 秒周期中仅使用 CALM 位便可完成在 255 和 256 时钟脉冲间（对应 243.3 ppm - 244.1ppm 的校准范围）有效增加。

正常情况下 RTCCLK 频率为 32768Hz，当 PREDIV_A=1（分频因子为 2），PREDIV_S 应该被设为 16379 而不是 16383（要减 4）。当 PREDIV_A=0, PREDIV_S 就应该设为 32759 而不是 32767（要减 8）。

如果 PREDIV_S 通过该方式减少，则校准输出始终的有效频率可由以下公式得出：

$F_{CAL} = F_{RTCCLK} \times [1 + (256 - CALM) / (2^{20} + CALM - 256)]$

在此情况下 , 如果 RTCCLK 为 32768.00 Hz，则 CALM[7:0] 的正确值应等于 0x100 (CALM 范围的中点)。

验证 RTC 校准

通过测量 RTCCLK 的精确频率，计算正确的 CALM 和 CALP 频率以确保 RTC 精度。有一个可选的 1HZ 输出用来测量和验证 RTC 精度。

如果测量 RTC 的精确频率过程超出时间间隔，可能会导致在测量期间内形成最多 2 个 RTCCLK 时钟周期的正误差（根据数字校准周期与测量周期的对齐方式而变）。

然而 , 如果测量期间校准周期的长度相同，可以消除这种测量误差。在此情况下，唯一能检测到的误差是由数字校准的分辨率所产生的。

- 默认情况下，校准周期为 32 秒。

该模式下，测量在整整 32 秒内 1HZ 输出的精确度，确保该测量精度在 0.477 ppm 内（0.5 个 RTCCLK 相对于 32 秒，受限于校准方法）。

- RTC_CALR 寄存器 CALW16 位设置为“1”，将强制使用 16 秒校准周期。

在此情况下，在 16 秒内测量 RTC 精确度，产生的错误最大值为 0.954 ppm（0.5 个 RTCCLK 周期相对于 16 秒）。然而，因为校准解决方案的限制，长期 RTC 精确度也将减少至 0.954 ppm：当 CALW16 为“1”时，CALM[0] 位将保持为“0”。

- RTC_CALR 寄存器 CALW8 位设置为“1”，将强制使用 8 秒校准周期。

在此情况下，在 8 秒内测量 RTC 精确度，产生的错误最大值为 1.907ppm（0.5 个 RTCCLK 周期相对于 8 秒）。长期 RTC 精确度也将变差至 1.907ppm：当 CALW16 为“1”时，CALM[1:0] 位将保持为“00”。

运行中重校准

当 RTC_ISR/INITF=0，校准寄存器 (RTC_CALR) 可通过以下流程实现运行中更新：

1. 查询 RTC_ISR/RECALPF (重校准挂起标志)；
2. 如果 RTC_ISR/RECALPF 置“0”，如果有必要通过向 RTC_CALR 写入一个新值，RECALPF 将置“1”。
3. 在向 RTC_CALR 写入新值后 3 个 ck_apre 周期内，新的校准设置生效。

22.3.11 时间戳功能

RTC_CR 寄存器 TSE 位置“1”启用时间戳功能。

当 RTC_TS 引脚检测到一个时间戳事件时候，会将日历存储在时间戳寄存器中 (RTC_TSSSR, RTC_TSTR, RTC_TSDR) 。当产生时间戳事件时，RTC_ISR 寄存器的时间戳标志位 (TSF) 被置位。

通过设置 RTC_CR 寄存器的 TSIE 位，在产生时间戳事件时，同时产生一个中断。

如果检测到一个新的时间戳事件时，时间戳标志位 (TSF) 已经被置位，则时间戳溢出标志位 (TSOVF) 标志为“溢出”，时间戳寄存器 (RTC_TSTR 和 RTC_TSDR) 维持上一事件的内容不变。

注： 1. 由于同步延迟，TSF 将在时间戳事件发生后 2 个 ck_apre 周期后被置位。

2. TSOVF 零延迟置位。如果两个时间戳事件发生的时间间隔非常短，TSOVF 位可视为“1”，而 TSF 位仍为“0”。因此建议当 TSF 位被置位后再检测 TSOVF 位。

警告： 如果在 TSF 位被清除后立即产生一个时间戳事件，TSF 位和 TSOVF 位都将被置位。为了避免被同一时间产生的时间戳事件覆盖，应用程序不得向 RSF 位写入“0”，除非已经从该位读取到“1”。

(可选) 一个侵入事件可导致一个时间戳被记录。参考 22.6.12 节：RTC 时间戳亚秒寄存器，了解 TAMPTS 控制位的描述。

22.3.12 侵入检测

RTC_TAMPx 输入事件可设置为边沿检测或带过滤功能的电平检测。

RTC 备份寄存器

备份寄存器 (RTC_BKPxR) 处在 VDD 备份域中，当 VDD 电源被切断，他们仍由 VBAT 维持供电。当系统复位或设备在待机模式下被唤醒时，他们不会被复位。上电复位时，备份寄存器将被复位。

当产生一个侵入检测事件时，备份寄存器将被复位。(参见 22.6.16 节：RTC 备份寄存器 (RTC_BKPxR) 和第 474 页中的侵入检测初始化)

侵入检测初始化

所有 RTC_TAMPx 侵入检测输入均与 RTC_ISR2 寄存器的 TAMPxF 标志相关。所有输入可通过设置 RTC_TAFCR 寄存器中相应的 TAMPxE 位为“1”来使能。

侵入检测事件可将所有备份寄存器 (RTC_BKPxR) 内容清除。

通过设置 RTC_TAFCR 寄存器的 TAMPIE 位，当检测到一个侵入检测事件时便产生一个中断。

侵入事件时间戳

设置 TAMPTS 为“1”，所有侵入事件将产生一个时间戳。这种情况下，RTC_ISR 中无论是 TSF 位还是 TSOVF 位被置 1，和正常的时间戳事件发生的效果相同。设置 TSF 或 TSOVF，与其关联的的侵入标志寄存器 TAMPxF 将同时被设置。

侵入输入的边沿检测

如果 TAMPFLT 位为“00”，根据相关 TAMPxTRG 位，当检测到上升沿或下降沿时，RTC_TAMPx 引脚将生成侵入检测事件。当边沿检测被选中后，RTC_TAMPx 输入的内部上拉电阻将被停用。

警告： 为避免侵入检测事件丢失并及时地发现侵入检测事件，用于边沿检测的信号与相应 TAMPxE 位执行逻辑与操作，以便在 RTC_TAMPx 引脚启用前检测到侵入检测事件。

- 当 TAMPxTRG = 0：如果在启用侵入检测（通过设置 TAMPxE 位为 1）前 RTC_TAMPx 已经为高电平，一旦启用 RTC_TAMPx 输入，则会产生一个侵入事件（尽管在 TAMPxE 位置“1”后 RTC_TAMPx 输入中并没有出现上升沿）。
- 当 TAMPxTRG = 1：如果在启用侵入检测前， RTC_TAMPx 已经为低电平，一旦启用 RTC_TAMPx，则会产生一个侵入事件（尽管在设置 TAMPxE 位后 RTC_TAMPx 输入中并没有出现下降沿）。

在一个侵入事件被检测到并清除后，RTC_TAMPx 复用功能应该被禁止。然后在再次写入备份寄存器前重新启用 RTC_TAMPx 复用功能（通过设置 TAMPxE 位为 1）。这样可以阻止软件在 RTC_TAMPx 检测出仍有侵入事件发生时对备份寄存器进行写操作。这相当于对 RTC_TAMPx 复用功能输入进行电平检测。

注： 当电源断开时，侵入检测功能仍然有效。为避免对备份寄存器产生不必要的复位，RTC_TAMPx 复用功能对应的引脚应该在片外连接到正确的电平。

针对 RTC_TAMPx 输入的带滤波功能电平检测

将 TAMPFLT 设置为非“0”值，会启用带滤波功能的电平检测。当检测到 2 个、4 个或 8 个（根据 TAMPFLT 设置）连续样本的指定电平（TAMPxTRG 位决定）时，将产生一个侵入检测事件。

RTC_TAMPx 输入在其状态被采样前，通过 I/O 内部上拉电阻实现预充电，直至 TAMPPUDIS 设置为“1”后，停止该功能。预充电时间由 TAMPPRCH 位决定，允许 RTC_TAMPx 输入拥有更大的电容。

可通过调整 TAMPFREQ 来改变电平检测的采样频率，从而在侵入检测延迟与通过上拉电阻产生的电源消耗间进行取舍。

注： 有关上拉电阻的电气特性请参考数据手册。

22.3.13 校准时钟输出

当 RTC_CR 寄存器 COE 位置“1”，RTC_CALIB 输出上会提供一个参考时钟。

如果 RTC_CR 寄存器 COSEL 位为 0， 且 PREDIV_A = 0x7F, RTC_CALIB 频率 =fRTCCLK/64，对应于 32.768kHz 的 RTCCLK 时，输出频率为 512HZ。因为下降沿存在轻微抖动，所以 RTC_CALIB 占空比是不规则的。因此建议使用上升沿。

如果 COSEL=1，“PREDIV_S+1”为非“0”的 256 的倍数 (例如 PREDIV_S[7:0] = 0xFF)，RTC_CALIB 频率等于 fRTCCLK/(256 * (PREDIV_A+1))。这样在 RTCCLK 频率为 32.768 kHz 的时候，对于的 RTC_CALIB 输出频率为 1HZ（默认 PREDIV_A = Ox7F, PREDIV_S =0xFF）。

22.3.14 报警输出

RTC_CR 寄存器的 OSEL[1:0] 控制位的功能是激活报警备用功能 RTC_ALARM 输出，并选择输出功能。这些功能与 RTC_ISR 寄存器中相应标志位的内容一致。

输出极性由 RTC_CR 寄存器的 POL 控制位决定，因此当 POL 设置为“1”时，就会输出选定标志位的取反后的值。

报警复用功能输出

通过设置 RTC_TAFCR 寄存器 ALARMOUTTYPE 控制位，RTC_ALARM 引脚可被设置为开漏输出或推挽输出。

注： 1. 一旦 RTC_ALARM 输出启用，其优先级将高于 RTC_CALIB (COE 位无任何影响，但必须保持清除状态)。

2. RTC_CALIB 或 RTC_ALARM 输出启用后，RTC_OUT 脚将自动被配置为复用输出功能。

22.4 RTC 低功耗模式

表 63. RTC 低功耗模式的影响

模式	描述
睡眠模式	无影响 RTC 中断使设备退出睡眠模式。
停止模式	如果 RTC 时钟源为 LSE 或 LSI，RTC 将保持运行状态。RTC 报警 , RTC 侵入事件 , RTC 时间戳事件和 RTC 唤醒事件使设备退出停止模式。
待机模式	如果 RTC 时钟源为 LSE 或 LSI，RTC 将保持运行状态。RTC 报警 , RTC 侵入事件 , RTC 时间戳事件和 RTC 唤醒事件使设备退出待机模式。

22.5 RTC 中断

所有 RTC 中断都连接到 EXTI 控制器，参见 11.2.6 节外部和内部中断 / 时间线图（第 159 页）。

按下列顺序启用 RTC 报警中断：

1. 在中断模式下设置并使能 RTC 报警事件对应的 EXTI 线，并选定上升沿极性。
2. 设置并使能 NVIC 的 RTC_ALARM IRQ 通道。
3. 设置 RTC，触发 RTC 报警 (Alarm A)。

按下列顺序启用侵入中断：

1. 在中断模式下设置并使能 RTC 侵入事件对应的 EXTI 线，并选定上升沿极性。
2. 设置并使能 NVIC 的 TAMP_STAMP IRQ 通道。
3. 设置 RTC，检测 RTC 侵入事件。

按下列顺序启用 RTC 时间戳中断：

1. 在中断模式下设置并使能 RTC 时间戳事件对应的 EXTI 线，并选定上升沿极性。
2. 设置并使能 NVIC 的 TAMP_STAMP IRQ 通道。
3. 设置 RTC，检测 RTC 时间戳事件。

表 64. 中断控制位

中断事件	事件标志	使能控制位	退出睡眠模式	退出停止模式	退出待机模式
Alarm A	ALRAF	ALRAIE	yes	yes(1)	yes(1)
RTC_TS输入 (时间戳)	TSF	TSIE	yes	yes(1)	yes(1)
RTC_TAMP1输入检测	TAMP1F	TAMPIE	yes	yes(1)	yes(1)
RTC_TAMP2 输入检测	TAMP2F	TAMPIE	yes	yes(1)	yes(1)

1. 仅当 RTC 时钟源为 LSE 或 LSI 时才有可能从停止和待机模式中唤醒。

22.6 RTC 寄存器

关于寄存器描述中的缩略词，请参考 1.1 节 (第 31 页)。

可以用字 (32 位) 的方式操作外设寄存器。

22.6.1 RTC 时间寄存器 (RTC_TR)

RTC_TR 是日历时间影子寄存器，该寄存器只能在初始化模式下写入。参见日历初始化和配置（第 468 页）以及读日历（第 469 页）。

该寄存器处于写保护状态，相关说明请参见 RTC 寄存器写保护（第 468 页）。

偏移地址 : 0x00
上电复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PM	HT[1:0]		HU[3:0]			
									rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	MNT[2:0]			MNU[3:0]				Res.	ST[2:0]			SU[3:0]			
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31-23 保留 , 必须保持复位时的值。
位 22 PM: AM/PM 标记
0: AM or 24 小时制
1: PM
位 21:20 HT[1:0]: 时，十位值以 BCD 格式存储。
位 16:16 HU[3:0]: 时，个位值以 BCD 格式存储。
位 15 保留 , 必须保持复位时的值。
位 14:12 MNT[2:0]: 分，十位值以 BCD 格式存储。
位 11:8 MNU[3:0]: 分，个位值以 BCD 格式存储。
位 7 保留 , 必须保持复位时的值。
位 6:4 ST[2:0]: 秒，十位值以 BCD 格式存储。
位 3:0 SU[3:0]: 秒，个位值以 BCD 格式存储。

22.6.2 RTC 日期寄存器 (RTC_DR)

RTC_DR 是日历日期影子寄存器，该寄存器只能在初始化模式下写入。参见日历初始化和配置（第 468 页）以及读日历（第 469 页）。

该寄存器处于写保护状态，相关说明请参见 RTC 寄存器写保护（第 468 页）。

偏移地址 : 0x04
上电复位值 : 0x0000 2101

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	YT[3:0]				YU[3:0]			
								rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDU[2:0]			MT	MU[3:0]				Res.	Res.	DT[1:0]		DU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw

位 31:24 保留 , 必须保持复位时的值。
位 23:20 YT[3:0]: 年，十位值以 BCD 格式存储。
位 19:16 YU[3:0]: 年，个位值以 BCD 格式存储。
位 15:13 WDU[2:0]: 星期
000: 禁用
001: 星期一
...
111: 星期日
位 12 MT: 月，十位值以 BCD 格式存储。
位 11:8 MU: 月，个位值以 BCD 格式存储。
位 7:6 保留 , 必须保持复位时的值。
位 5:4 DT[1:0]: 日，十位值以 BCD 格式存储。
位 3:0 DU[3:0]: 日，个位值以 BCD 格式存储。

22.6.3 RTC 控制寄存器 (RTC_CR)

偏移地址 : 0x08

上电复位值 : 0x0000 0000

系统复位 : 无影响。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COE	OSEL[1:0]		POL	COSEL	BKP	SUB1H	ADD1H
								rw	rw	rw	rw	rw	rw	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSIE	Res.	Res.	ALRAIE	TSE	Res.	Res.	ALRAE	Res.	FMT	BYPS HAD	REFCKON	TSEDGE	Res.		
rw			rw	rw			rw		rw	rw	rw	rw			

位 31:24 保留 , 必须保持复位时的值。

位 23 COE: 校准输出使能。

该位使能 RTC_CALIB 输出。

0: 校准输出禁用；

1: 校准输出启用。

位 22:21 OSEL[1:0]: 输出选择。

该位用于选择 RTC_ALARM 输出关联的标志位。

00: 输出禁用；

01: Alarm A 输出启用；

10: 保留；

11: 保留。

位 20 POL: 输出极性。

该位用于设置 RTC_ALARM 的输出极性。

0: 根据 OSEL[1:0] 位，ALRAF 有效时，引脚为高电平；

1: 根据 OSEL[1:0] 位，ALRAF 有效时，引脚为低电平。

位 19 COSEL : 校准输出选择。

COE=1 时，该位用于选择 RTC_CALIB 的输出信号。

0: 校准输出为 512 Hz；

1: 校准输出为 1 Hz。

上述频率在 RTCCLK 为 32.768 kHz 和预分频器处于默认值 (PREDIV_A=127，PREDIV_S=255) 时有效。参见 22.3.13 节：校准时钟输出。

位 18 BKP: 备份。

该位可由用户写入，用于记录夏令时是否已经发生变化。

位 17 SUB1H: 减少 1 小时 (冬季时间变化)。

读该位会恒为“0”。在初始化模式之外设置该位，如果当前小时位不为 0，日历时间将减少 1 小时。如果当前小时位为 0，则该位无效。

0: 无效。

1: 当前时间减少 1 小时，用于校准冬季时间变化。

位 16 ADD1H: 增加 1 小时 (夏季时间变化)。
读该位会恒为“0”。在初始化模式外设置该位，日历时间会增加 1 小时。
0: 无效；
1: 当前时间增加 1 小时，用于校准夏季时间变化。

位 15 TSIE: 时间戳中断使能。
0: 时间戳中断禁用；
1: 时间戳中断启用。

位 14 保留 , 必须保持复位时的值。

位 13 保留 , 必须保持复位时的值。

位 12 ALRAIE: Alarm A 中断使能。
0: Alarm A 中断禁用；
1: Alarm A 中断启用。

位 11 TSE: 时间戳使能
0: 时间戳禁用；
1: 时间戳启用。

位 10 保留 , 必须保持复位时的值。

位 9 保留 , 必须保持复位时的值。

位 8 ALRAE: Alarm A 使能
0: Alarm A 禁用；
1: Alarm A 启用。

位 7 保留 , 必须保持复位时的值。

位 6 FMT: 时间格式
0: 24 小时 / 天 格式；
1: AM/PM 时间格式。

位 5 BYPSHAD: 绕过影子寄存器
0: 从影子寄存器读取日历值，每两个 RTCCLK 周期更新一次；
1: 直接从日历计数器读取日历值。
注：如果 APB1 时钟频率低于 RTCCLK 频率的 7 倍，BYPSHAD 必须置“1”。

位 4 REFCKON: RTC_REFIN 参考时钟检测使能 (50 或 60 Hz)
0: RTC_REFIN 检测禁用；
1: RTC_REFIN 检测启用。
注：PREDIV_S 必须为 0x00FF.

位 3 TSEDGE: 时间戳事件有效边沿
0: RTC_TS 输入上升沿生成一个时间戳事件；
1: RTC_TS 输入下降沿生成一个时间戳事件。
当 TSEDGE 的值改变时 TSE 必须为零，以避免产生不必要的 TSF 值。

位 2:0 保留 , 必须保持复位时的值。

注：
1. *该寄存器的位 7，6 和 4 只能在初始化模式下写入 (RTC_ISR/INITF = 1)。.*
2. *不建议在日历的小时数在增加的时候去改变它，这有可能将正确的小时增量屏蔽掉。*
3. *ADD1H 和 SUB1H 值的改变将在下一秒生效。.*
4. *该寄存器处于写保护下。写访问的流程描述请参见 RTC 寄存器写保护（第 468 页）。*

22.6.4 RTC 初始化和状态寄存器 (RTC_ISR)

该寄存器（除 RTC_ISR[13:8] 位外）处在写保护状态。写访问的流程描述请参见 RTC 寄存器写保护（第 468 页）。

偏移地址 : 0x0C
复位值 : 0x0000 0007

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RECALPF
															r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TAMP2F	TAMP1F	TSOVF	TSF	Res.	Res.	ALRAF	INIT	INITF	RSF	INITS	SHPF	Res.	Res.	ALRAWF
	rc_w0	rc_w0	rc_w0	rc_w0			rc_w0	rw	r	rc_w0	r	rc_w0			r

位 31:17 保留 , 必须保持复位时的值。

位 16 RECALPF: 校准挂起标志
当软件向 RTC_CALR 做写操作时，RECALPF 状态标志位自动置“1”，表示 RTC_CALR 寄存器被封锁。当有其他新的校准设置执行时，该位回到“0”。参见“运行中重校准”。

位 15 保留 , 必须保持复位时的值。

位 14 TAMP2F: RTC_TAMP2 检测标志
当在 RTC_TAMP2 输出检测到侵入事件时该标志由硬件置位。
该标志由软件写入“0”后清除。

位 13 TAMP1F: RTC_TAMP1 检测标志
当在 RTC_TAMP1 输出检测到侵入事件时该标志由硬件置位。
该标志由软件写入“0”后清除。

位 12 TSOVF: 时间戳溢出标志
该标志在 TSF 已经被置 1，又产生时间戳事件时由硬件置位。由软件写入“0”后清除。建议在检查 TSF 位清除后再清除 TSOVF，否则如果在 TSF 位清除前产生一个时间戳事件，溢出可能会被忽略。

位 11 TSF: 时间戳标志
当产生时间戳事件时该标志由硬件置位。
该标志由软件写入“0”后清除。

位 10 保留 , 必须保持复位时的值。

位 9 保留 , 必须保持复位时的值。

位 8 ALRAF: Alarm A 标志

当时间 / 日期寄存器 (RTC_TR and RTC_DR) 与 Alarm A 寄存器 (RTC_ALRMAR) 匹配时，该标志由硬件置位。

该标志由软件写入“0”后清除。

位 7 INIT: 初始化模式

0: 运行模式

1: 在初始化模式下编程时间和日期寄存器 (RTC_TR 和 RTC_DR)，以及预分频器寄存器 (RTC_PRER)。计数器停止，直至 INIT 复位后计数器将从新值开始计数。

位 6 INITF: 初始化标志

该位置“1”，RTC 处在初始化状态，时间、日期和预分频器寄存器可被更新。

0: 日历寄存器不可更新；

1: 日历寄存器可更新。

位 5 RSF: 寄存器同步标志

每当日历寄存器中的内容复制到影子寄存器 (RTC_SSRx, RTC_TRx and RTC_DRx) 中时，该位由硬件置位。当移位操作被挂起 (SHPF=1) 或处于忽略影子寄存器模式 (BYPSHAD=1) 下时，该位由硬件在初始化模式下清除。该位也可由软件清除。

在初始化模式下，该位可由硬件 / 软件清除。

0: 日历影子寄存器尚未同步；

1: 日历影子寄存器已经同步。

位 4 INITS: 初始化状态标志

当日历中“年”字段不为“0”时，该位由硬件置位。

0: 日历尚未被初始化；

1: 日历已经被初始化。

位 3 SHPF: 移位操作挂起

0: 没有移位操作被挂起

1: 有移位操作被挂起

当通过向 RTC_SHIFTR 寄存器写入产生一个移位操作时，该位立即由硬件置位。当相应的移位操作执行完毕后，该位由软件清除。直接对 SHPF 写入是无效的。

位 2 保留 , 必须保持复位时的值。

位 1 保留 , 必须保持复位时的值。

位 0 ALRAWF: Alarm A 写标志

当 RTC_CR 的 ALRAE 被清“0”后，Alarm A 的值发生变化时，该位由硬件置位。该位由硬件在初始化模式下清除。

0: Alarm A 不可更新；

1: Alarm A 可以更新。

注： 1. ALRAF 位和 TSF 位被写“0”后，将在 2 个 APB 时钟周期后清零。

22.6.5 RTC 预分频器寄存器 (RTC_PRER)

该寄存器只能在初始化模式下写入，初始化必须由两次独立的写访问完成。参见日历初始化和配置（第 468 页）。

该寄存器处在写保护状态，相关说明请参见 RTC 寄存器写保护（第 468 页）。

偏移地址 : 0x10
上电复位值 : 0x007F 00FF
系统复位 : 无影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREDIV_A[6:0]						
									rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	PREDIV_S[14:0]														
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:23 保留 , 必须保持复位时的值。
位 22:16 PREDIV_A[6:0]: 异步预分频器系数
ck_apre 频率 = RTCCLK 频率 /(PREDIV_A+1)
位 15 保留 , 必须始终保位复位状态。
位 14:0 PREDIV_S[14:0]: 同步预分频器系数
ck_spre 频率 = ck_apre 频率 /(PREDIV_S+1)

22.6.6 RTC alarm A 寄存器 (RTC_ALRMAR)

该寄存器只能在 RTC_ISR 的 ALRAWF 置“1”或初始化模式下写入。

该寄存器处在写保护状态，相关说明请参见 RTC 寄存器写保护（第 468 页）。

偏移地址 : 0x1C
上电复位值 : 0x0000 0000
系统复位 : 无影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSK4	WDSEL	DT[1:0]		DU[3:0]				MSK3	PM	HT[1:0]		HU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSK2	MNT[2:0]			MNU[3:0]				MSK1	ST[2:0]			SU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 MSK4: Alarm A 日期屏蔽
0: 如果日期 / 星期匹配，Alarm A 置位
1: 日期 / 星期的值对 Alarm A 无影响
位 30 WDSEL: 日期选择
0: DU[3:0] 表示日期
1: DU[3:0] 表示星期数。 DT[1:0] 不起作用。
位 29:28 DT[1:0]: 日期，十位值。以 BCD 格式存储。
位 27:24 DU[3:0]: 日期，个位值。以 BCD 格式存储。
位 23 MSK3: Alarm A “时”屏蔽
0: 如果“小时”匹配，Alarm A 置位
1: “小时”的值对 Alarm A 无影响
位 22 PM: AM/PM 标志
0: AM 或 24 小时制
1: PM
位 21:20 HT[1:0]: 小时，十位值。以 BCD 格式存储。
位 19:16 HU[3:0]: 小时，个位值。以 BCD 格式存储。
位 15 MSK2: Alarm A “分”屏蔽
0: 如果“分”匹配，Alarm A 置位
1: “分”的值对 Alarm A 无影响
位 14:12 MNT[2:0]: 分，十位值。以 BCD 格式存储。
位 11:8 MNU[3:0]: 分，个位值。以 BCD 格式存储。
位 7 MSK1: Alarm A “秒”屏蔽
0: 如果“秒”匹配，Alarm A 置位
1: “秒”的值对 Alarm A 无影响
位 6:4 ST[2:0]: 秒，十位值。以 BCD 格式存储。
位 3:0 SU[3:0]: 秒，个位值。以 BCD 格式存储。

22.6.7 RTC 亚秒寄存器 (RTC_SSR)

偏移地址 : 0x28

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留 , 必须保持复位时的值。

位 15:0 SS: 亚秒值

SS[15:0] 是同步预分频器计数器中的值。“秒”的小数部分由下列公式给出：

Second fraction = (PREDIV_S - SS) / (PREDIV_S + 1)

注：仅当执行完一个移位操作后，SS 可能大于 PREDIV_S。此时，正确的时间 / 日期比 RTC_TR/RTC_DR 少一秒。

22.6.8 RTC 移位控制寄存器 (RTC_SHIFTR)

该寄存器处于写保护状态，相关说明请参见 RTC 寄存器写保护（第 468 页）。

偏移地址 : 0x2C
复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADD1S	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SUBFS[14:0]														
	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31 ADD1S: 增加一秒
0: 无效
1: 时钟 / 日历增加一秒。
该位只能写入且始终读为“0”。当移位操作挂起（RTC_ISR 中 SHPF=1），写操作对该位无影响。
该性质与 SUBFS 一起可用于在某次单独操作中，有效增加时钟的值， 增加值为若干分之一秒。

位 31:15 保留 , 必须保持复位时的值。

位 14:0 SUBFS: 减少若干分之一秒
该位只能写入，如果读则恒为“0”。当一个操作在执行时（RTC_ISR 的 SHPF=1），写该位无效。
写入 SUBFS 的值将添加到同步预分频器计数器。如果计数器是倒计时的，时钟将被延迟，延迟时间由以下公式得出：
Delay (seconds) = SUBFS / (PREDIV_S + 1)
当 ADD1S 功能与 SUBFS 一同使用时，时钟（推进时钟）将增加若干分之一秒，具体增加值有以下公式得出：
Advance (seconds) = (1 - (SUBFS / (PREDIV_S + 1))) .
注 : 写 SUBFS 将导致 RSF 被清除。软件持续运行直至 RSF=1，从而确保影子寄存器中相应值已与移位时间同步。

22.6.9 RTC 写保护寄存器 (RTC_WPR)

偏移地址 : 0x24

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEY							
								w	w	w	w	w	w	w	w

位 31:8 保留 , 必须保持复位时的值。

位 7:0 KEY: 写保护键

该字节由软件写入。始终读为“0x00”

有关如何解除 RTC 寄存器写保护的信息，请参见 RTC 寄存器写保护。

22.6.10 RTC 时间戳事件寄存器 (RTC_TSTR)

仅当 RTC_ISR 中 TSF 置“1”时，该寄存器的内容才有效。当 TSF 位复位时，该寄存器的内容将被清除。

偏移地址 : 0x30

上电复位值 : 0x0000 0000

系统复位 : 无影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PM	HT[1:0]		HU[3:0]			
									r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	MNT[2:0]			MNU[3:0]				Res.	ST[2:0]			SU[3:0]			
	r	r	r	r	r	r	r		r	r	r	r	r	r	r

位 31:23 保留 , 必须保持复位时的值。

位 22 PM: AM/PM 标记

0: AM 或 24 小时制

1: PM

位 21:20 HT[1:0]: 小时，十位值。以 BCD 格式存储 .

位 19:16 HU[3:0]: 小时，个位值。以 BCD 格式存储 .

位 15 保留 , 必须保持复位时的值。

位 14:12 MNT[2:0]: 分，十位值。以 BCD 格式存储 .

位 11:8 MNU[3:0]: 分，个位值。以 BCD 格式存储 .

位 7 保留 , 必须保持复位时的值。
位 6:4 ST[2:0]: 秒，十位值。以 BCD 格式存储 .
位 3:0 SU[3:0]: 秒，个位值。以 BCD 格式存储 .

22.6.11 RTC 时间戳日期寄存器 (RTC_TSDR)

仅当 RTC_ISR 中 TSF 置“1”时，该寄存器的内容才有效。当 TSF 位复位时，该寄存器的内容将被清除。

偏移地址 : 0x34
上电复位值 : 0x0000 0000
系统复位 : 无影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDU[1:0]			MT	MU[3:0]				Res.	Res.	DT[1:0]		DU[3:0]			
r	r	r	r	r	r	r	r			r	r	r	r	r	r

位 31:16 保留 , 必须保持复位时的值。
位 15:13 WDU[1:0]: 星期数
位 12 MT: 月，十位值。以 BCD 格式存储
位 11:8 MU[3:0]: 月，个位值。以 BCD 格式存储
位 7:6 保留 , 必须保持复位时的值。
位 5:4 DT[1:0]: 日期，十位值。以 BCD 格式存储
位 3:0 DU[3:0]: 日期，个位值。以 BCD 格式存储

22.6.12 RTC 时间戳亚秒寄存器 (RTC_TSSSR)

仅当设置 RTC_ISR/TSF 位时，该寄存器的内容才有效。当 RTC_ISR/TSF 位复位时，该寄存器的内容将被清除。

偏移地址 : 0x38
上电复位值 : 0x0000 0000
系统复位 : 无影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留 , 必须保持复位时的值。

位 15:0 SS: 亚秒值

当发生时间戳事件时，SS[15:0] 是同步预分频器计数器中的值。

22.6.13 RTC 校准寄存器 (RTC_CALR)

该寄存器处在写保护状态，相关说明请参见 RTC 寄存器写保护（第 468 页）

偏移地址 : 0x3C

上电复位值 : 0x0000 0000

系统复位 : 无影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CALP	CALW8	CALW 16	Res.	Res.	Res.	Res.	CALM[8:0]								
rw	rw	rw					rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留 , 必须保持复位时的值。

位 15 CALP: RTC 频率增加 488.5 ppm

0: 无 RTCCLK 脉冲增加。

1: 每 211 脉冲增加一个 RTCCLK 脉冲（频率增加 488.5 ppm）

该功能与 CALM 一起使用，采用好的分辨率的同时降低日历的频率。如果输入频率为 32768 Hz, 则在一个 32 秒窗口中 RTCCLK 脉冲增加数可由下列公式得出：(512 * CALP) – CALM。

参见 22.3.10 节：RTC 平滑数字校准。

注 :

位 14 CALW8: 采用 8 秒校准周期

CALW8 置“1”，选择 8 秒校准周期。

注：当 CALW8=’1’时，CALM[1:0] 锁定为“00”. 参见 22.3.10 节：RTC 平滑数字校准。

位 13 CALW16: 采用 16 秒校准周期

当 CALW16 置“1”，激活 16 秒校准周期。如果 CALW8=1，该位不能为“1”。

注：当 CALW16=’1’时，CALM[0] 锁定为 ‘0’。参见 22.3.10 节：RTC 平滑数字校准。

位 12:9 保留 , 必须保持复位时的值。

位 8:0 CALM[8:0]: 校准减

减少日历频率：在 220 RTCCLK 脉冲范围内 (如果输出频率为 32768 Hz，则为 32 秒) 通过屏蔽 CALM 将减少日历（按 0.9537 ppm 分辨率）的频率。

要增加日历频率：此功能应与 CALP 一同使用。参见 22.3.10 节：RTC 平滑数字校准（第 471 页）。

22.6.14 RTC 侵入和复用功能配置寄存器 (RTC_TAFCR)

偏移地址 : 0x40

上电复位值 : 0x0000 0000

系统复位 : 无影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PC15 MODE	PC15 VALUE	PC14 MODE	PC14 VALUE	PC13 MODE	PC13 VALUE	Res.	Res.
								rw	rw	rw	rw	rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAMPP UDIS	TAMPPRCH[1:0]		TAMPFLT[1:0]		TAMPFREQ[2:0]			TAMPT S	Res.	Res.	TAMP2 TRG	TAMP2 E	TAMPIE	TAMP1 TRG	TAMP1 E
rw	rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw

位 31:24 保留 , 必须保持复位时的值。

位 23 PC15MODE: PC15 模式

0: PC15 受 GPIO 配置寄存器约束，因此 PC13 在待机模式下浮空。

1: LSE 禁用时，PC15 强制为推挽输出模式 。

位 22 PC15VALUE: PC15 值

如果 LSE 禁用且 PC15MODE = 1, PC15VALUE 设置 PC15 的输出值。

位 21 PC14MODE: PC14 模式

0: PC14 受 GPIO 配置寄存器约束，因此 PC13 在待机模式下浮空。

1: LSE 禁用时，PC14 强制为推挽输出模式

位 20 PC14VALUE: PC14 值

如果 LSE 禁用且 PC14MODE = 1, PC14VALUE 设置 PC14 的输出值。

位 19 PC13MODE: PC13 模式

0: PC13 受 GPIO 配置寄存器约束，因此 PC13 在待机模式下浮空。

1: RTC 复用功能禁用时，PC13 强制为推挽输出模式。

位 18 PC13VALUE: RTC_ALARM 输出形式 / PC13 值

如果 PC13 用于输出 RTC_ALARM, PC13VALUE 设置输出：

0: RTC_ALARM 为开漏输出；

1: RTC_ALARM 为推挽输出

如果所有 RTC 复用功能禁用且 PC13MODE = 1, PC13VALUE 设置 PC13 输出值。

位 17:16 保留 , 必须保持复位时的值。

位 15 TAMPPUDIS: RTC_TAMPx 上拉禁用

由该位决定是否所有 RTC_TAMPx 引脚在采样前进行预充电。

0: RTC_TAMPx 引脚在采样前进行预充电 (使能内部上拉)

1: RTC_TAMPx 引脚的预充电功能禁用。

位 14:13 TAMPPRCH[1:0]: RTC_TAMPx 预充电时间
该位决定采样前上拉电阻启用时间。TAMPPRCH 适用于每次 RTC_TAMPx 输入。
0x0: 1 个 RTCCLK 周期
0x1: 2 个 RTCCLK 周期
0x2: 4 个 RTCCLK 周期
0x3: 8 个 RTCCLK 周期

位 12:11 TAMPFLT[1:0]: RTC_TAMPx 过滤器计数
该位决定在特定电平（TAMP*TRG）下（能够激活侵入事件）的连续样本数。TAMPFLT 适用于每次 RTC_TAMPx 输入。
0x0: 输入转换为有效电平 (RTC_TAMPx 输入无内部上拉) 的边沿现象将激活侵入事件。
0x1: 侵入事件在有效电平下 2 个连续样本后被激活。
0x2: 侵入事件在有效电平下 4 个连续样本后被激活。
0x3: 侵入事件在有效电平下 8 个连续样本后被激活。

位 10:8 TAMPFREQ[2:0]: 侵入抽样频率
确定 RTC_TAMPx 输入的采样频率。
0x0: RTCCLK / 32768 (当 RTCCLK = 32768 Hz 时为 1 Hz)
0x1: RTCCLK / 16384 (当 RTCCLK = 32768 Hz 时为 2 Hz)
0x2: RTCCLK / 8192 (当 RTCCLK = 32768 Hz 时为 4 Hz)
0x3: RTCCLK / 4096 (当 RTCCLK = 32768 Hz 时为 8 Hz)
0x4: RTCCLK / 2048 (当 RTCCLK = 32768 Hz 时为 16 Hz)
0x5: RTCCLK / 1024 (当 RTCCLK = 32768 Hz 时为 32 Hz)
0x6: RTCCLK / 512 (当 RTCCLK = 32768 Hz 时为 64 Hz)
0x7: RTCCLK / 256 (当 RTCCLK = 32768 Hz 时为 128 Hz)

位 7 TAMPTS: 侵入检测事件的有效时间戳
0: 侵入检测事件产生的时间戳不需保存。
1: 侵入检测事件产生的时间需保存。
即使 RTC_CR 中的 TSE=0，TAMPTS 依然有效。

位 6:5 保留 , 必须保持复位时的值。

位 2 TAMPIE: 侵入中断使能
0: 侵入中断禁用
1: 侵入中断启用

位 1 TAMP1TRG: RTC_TAMP1 输出的有效电平
如果 TAMPFLT != 00
0: RTC_TAMP1 输入保持低电平将触发一个侵入检测事件。
1: RTC_TAMP1 输入保持高电平将触发一个侵入检测事件。
如果 TAMPFLT = 00:
0: RTC_TAMP1 输入上升沿将触发一个侵入检测事件。
1: RTC_TAMP1 输入下降沿将触发一个侵入检测事件。

位 0 TAMP1E: RTC_TAMP1 输出检测使能
0: RTC_TAMP1 检测禁用
1: RTC_TAMP1 检测启用

警告： 当 TAMPFLT = 0，调整 TAMP1TRG 的值以避免产生不合逻辑的 TAMP1F 设置，此时 TAMP1E 的内容必须清除。

22.6.15 RTC alarm A 亚秒寄存器 (RTC_ALRMASSR)

仅当 RTC_CR ALRAE 复位时或在初始化模式下，该寄存器才可被写入。

该寄存器处在写保护状态，相关说明请参见 RTC 寄存器写保护（第 468 页）。

偏移地址 : 0x44

上电复位值 : 0x0000 0000

系统复位 : 无影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	MASKSS[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
				rw	rw	rw	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SS[14:0]														
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	w	rw	rw

位 31:28 保留 , 必须保持复位时的值。

位 27:24 MASKSS[3:0]: 从该位起品比最明显的位。

0: Alarm A 不匹配亚秒值。报警在秒单元增加 (假设其余字段是相互匹配的) 时置 1。

1: SS[14:1] 不参与 Alarm A 匹配。只有 SS[0] 参与匹配。

2: SS[14:2] 不参与 Alarm A 匹配。只有 SS[1:0] 参与匹配。

3: SS[14:2] 不参与 Alarm A 匹配。只有 SS[2:0] 参与匹配。

...

12: SS[14:12] 不参与 Alarm A 匹配。只有 SS[11:0] 参与匹配。

13: SS[14:13] 不参与 Alarm A 匹配。只有 SS[12:0] 参与匹配。

14: SS[14] 不参与 Alarm A 匹配。只有 SS[13:0] 参与匹配。

15: 15 个 SS 位均需参与匹配，且需匹配成功以激活报警。

同步计数器溢出位 (位 15) 始终不参与匹配。只有在移位操作后，该位才不为“0”。

位 23:15 保留 , 必须保持复位时的值。

位 14:0 SS[14:0]: 亚秒值

比较亚秒值与同步预分频器计数器中的内容，从而判断报警是否已经被激活。只有位 0 到 MASKSS-1 参与比较。

22.6.16 RTC 备份寄存器 (RTC_BKPxR)

偏移地址 : 0x50 to 0x60

上电复位值 : 0x0000 0000

系统复位 : 无影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BKP[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BKP[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	w	rw	rw

位 31:0 BKP[31:0]

应用程序可对这些寄存器进行读或写操作。

当 VDD 电源被切断他们仍由 VBAT 维持供电，因此这些位不会被系统复位所复位。

当设备在低功耗模式下运行时，这些位的内容仍然有效。

侵入检测事件发生时（即 TAMPxF=1）或闪存读出保护被禁用时，该寄存器复位

22.6.17 RTC 寄存器映像

表 65. RTC 寄存器映像和复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**RTC_TR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PM	HT [1:0]		HU[3:0]				Res.	MNT[2:0]			MNU[3:0]				Res.	ST[2:0]			SU[3:0]			
	Reset value										0	0	0	0	0	0	0		0	0	0	0	0	0	0		0	0	0	0	0	0	0
0x04	**RTC_DR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	YT[3:0]				YU[3:0]				WDU[2:0]			MT	MU[3:0]				Res.	Res.	DT [1:0]		DU[3:0]			
	Reset value									0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1			0	0	0	0	0	1
0x08	**RTC_CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COE	OSEL [1:0]		POL	Res.	BKP	SUB1H	ADD1H	TSIE	Res.	Res.	ALRAIE	TSE	Res.	Res.	ALRAE	DCE	FMT	FMT	BYPSHAD	TSEDGE	Res.		
	Reset value									0	0	0	0		0	0	0	0			0	0			0	0	0	0	0	0			
0x0C	**RTC_ISR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP2F	TAMP1F	TSOVF	TSF	Res.	Res.	ALRAF	INIT	INITF	RSF	INITS	SHPF	Res.	Res.	ALRAWF
	Reset value																	0	0	0	0	0			0	0	0	0	0	0			1
0x10	**RTC_PRER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREDIV_A[6:0]							PREDIV_S[14:0]															
	Reset value										1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0x1C	**RTC_ALRMAR**	MSK4	WDSEL	DT [1:0]		DU[3:0]				MSK3	PM	HT [1:0]		HU[3:0]				MSK2	MNT[2:0]			MNU[3:0]				MSK1	ST[2:0]			SU[3:0]			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	**RTC_WPR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEY							
	Reset value																									0	0	0	0	0	0	0	0

表 65. RTC 寄存器映像和复位值 (续)

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x28	**RTC_SSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SS[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**RTC_SHIFTR**	ADD1S	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBFS[14:0]														
	Reset value	0																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	**RTC_TSTR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PM	HT[1:0]		HU[3:0]				Res.	MNT[2:0]			MNU[3:0]				Res.	ST[2:0]			SU[3:0]			
	Reset value										0	0	0	0	0	0	0		0	0	0	0	0	0	0		0	0	0	0	0	0	0
0x34	**RTC_TSDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDU[1:0]			MT	MU[3:0]				Res.	Res.	DT [1:0]		DU[3:0]			
	Reset value																	0	0	0	0	0	0	0	0			0	0	0	0	0	0
0x38	**RTC_TSSSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SS[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	**RTC_ CALR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CALP	CALW8	CALW16	Res.	Res.	Res.	Res.	CALM[8:0]								
	Reset value																	0	0	0					0	0	0	0	0	0	0	0	0
0x40	**RTC_TAFCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PC15MODE	PC15MODE	PC14VALUE	PC14MODE	PC13VALUE	PC13VALUE	Res.	Res.	TAMP-PUDIS	TAMP-PRCH[1:0]		TAMPFLT[1:0]			TAMPFREQ[2:0]		TAMPTS	Res.	Res.	TAMP2-TRG	TAMP2E	TAMPIE	TAMP1TRG	TAMP1E
	Reset value									0	0	0	0	0	0			0	0	0	0	0	0	0	0	0			0	0	0	0	0
0x44	**RTC_ ALRMASSR**	Res.	Res.	Res.	Res.	MASKSS[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SS[14:0]														
	Reset value					0	0	0	0										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x50 to 0x60	**RTC_BKP0R**	BKP[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	to **RTC_BKP4R**	BKP[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

有关寄存器的边界地址，参见 2.2.2 节（第 35 页）。

23 I²C 接口

23.1 I²C 简介

I²C (芯片间) 总线接口连接微控制器和串行 I²C 总线。
它提供多主机功能，控制所有 I²C 总线特定的时序、协议、仲裁和定时。 . 它支持标准模式，快速模式和超快速模式。

同时与 SMBus（系统管理总线）和 PMBus（电源管理总线）保持兼容。

可以使用 DMA 以减轻 CPU 的负担。

23.2 I²C 的主要特点

- I²C 总线规范 rev03 兼容性：
  - 从机模式和主机模式
  - 多主机功能
  - 标准模式（高达 100kHz）
  - 快速模式（高达 400kHz）
  - 超快速模式（高达 1 MHz）
  - 7 位和 10 位地址模式
  - 多个 7 位从地址（2 个地址，其中一个可屏蔽）
  - 所有 7 位地址应答模式
  - 广播呼叫
  - 可编程建立和保持时间
  - 易用的事件管理
  - 可选的时钟延长
  - 软件复位
- 1 字节缓冲带 DMA 功能
- 可编程的模拟和数字噪声滤波器

以下附加功能根据产品具体配备（见 23.3 节： I2C 具体功能配备）：
- SMBus 规范 2.0 版的兼容性：
  - 硬件 PEC（包错误检查）的生成和验证，带 ACK 控制
  - 命令和数据的应答控制
  - 地址解析协议（ARP）的支持
  - 主机和设备支持
  - SMBus 报警
  - 超时和空闲状态检测

- 与 PMBus 版本 1.1 标准兼容
- 独立的时钟：允许 I²C 选择一个独立的时钟源通信速度相对于 PCLK 可独立调整
- 根据地址匹配事件从 STOP 模式唤醒。

23.3 I²C 具体功能配备

本文描述了 I²C1 所实现的全套功能。 I²C2 功能稍有裁剪，但有的功能都和 I²C1 一样。 差异如下表所示。

表 67. STM32F0xx 的 I²C1 和 2 的功能差异

I2C features(1)	I2C1	I2C2
Independent clock	X	
SMBus	X	
Wakeup from STOP	X	
20 mA output drive for FM+ mode	X	

1. X = supported.

23.4 I²C 功能描述

除了接收和发送数据，这个接口从串行转换成并行格式，以及其逆过程。 通过软件启用或禁用中断。该接口通过数据引脚（SDA）和时钟引脚（SCL）连接到 I²C 总线。它可以连接标准（高达 100 kHz），快速模式（高达 400 kHz）或快速模式加（高达 1 MHz）I²C 总线。

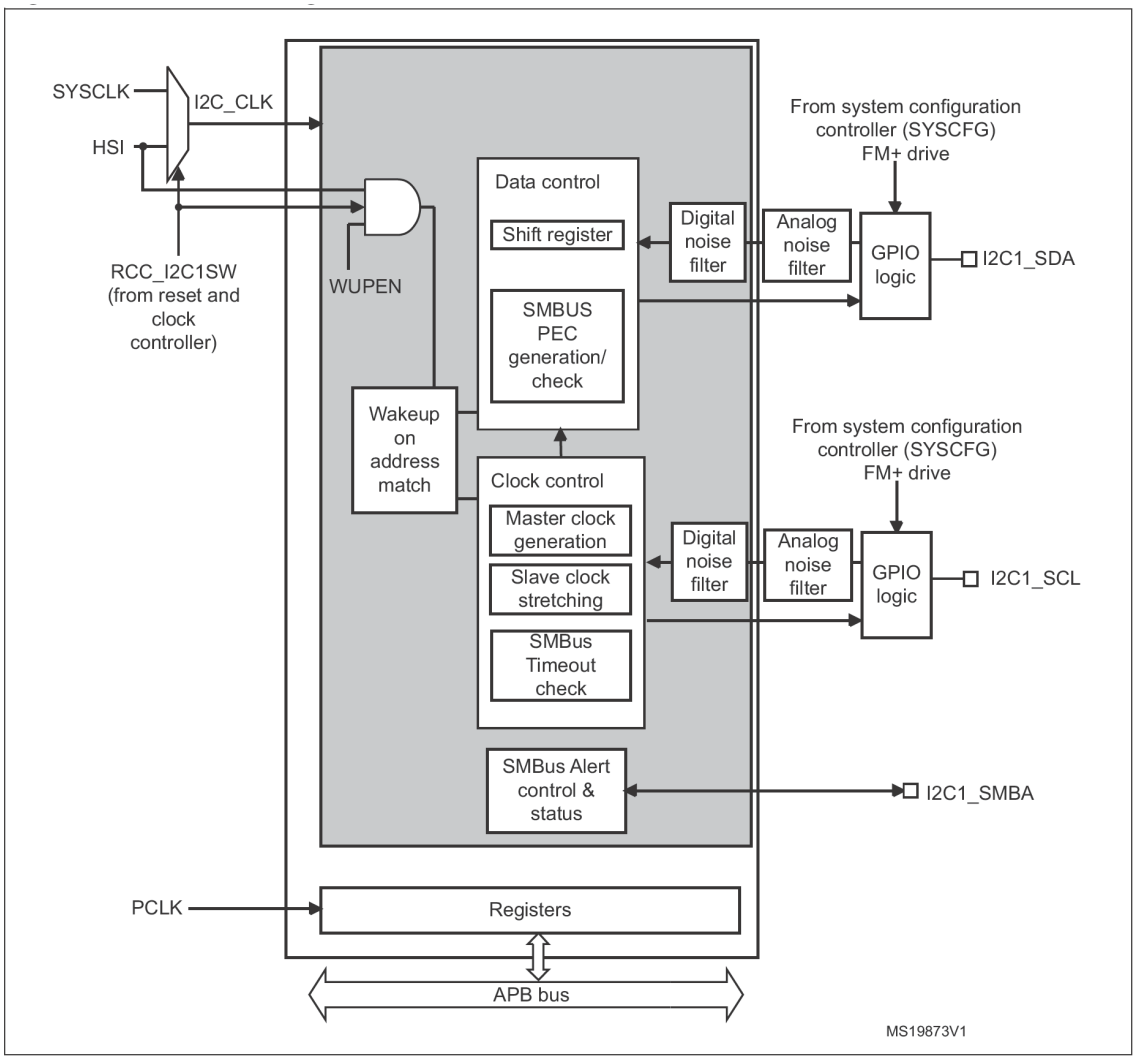
此接口也可以通过数据引脚（SDA）和时钟引脚（SCL）连接到的 SMBus。

如果有 SMBus 功能支持： 附加的 SMBus ALERT 引脚（SMBA）也是可用的。

23.4.1 I²C1 框图

I²C1 的框图如图 194 所示。

图 194. I²C1 框图



I²C1 由一个独立的时钟源驱动，它允许 I²C 相对于 PCLK 频率独立运作。

这个独立的时钟源可以选择以下两个时钟源之一：
HSI： 高速内部振荡器（默认值）
SYSCLK： 系统时钟

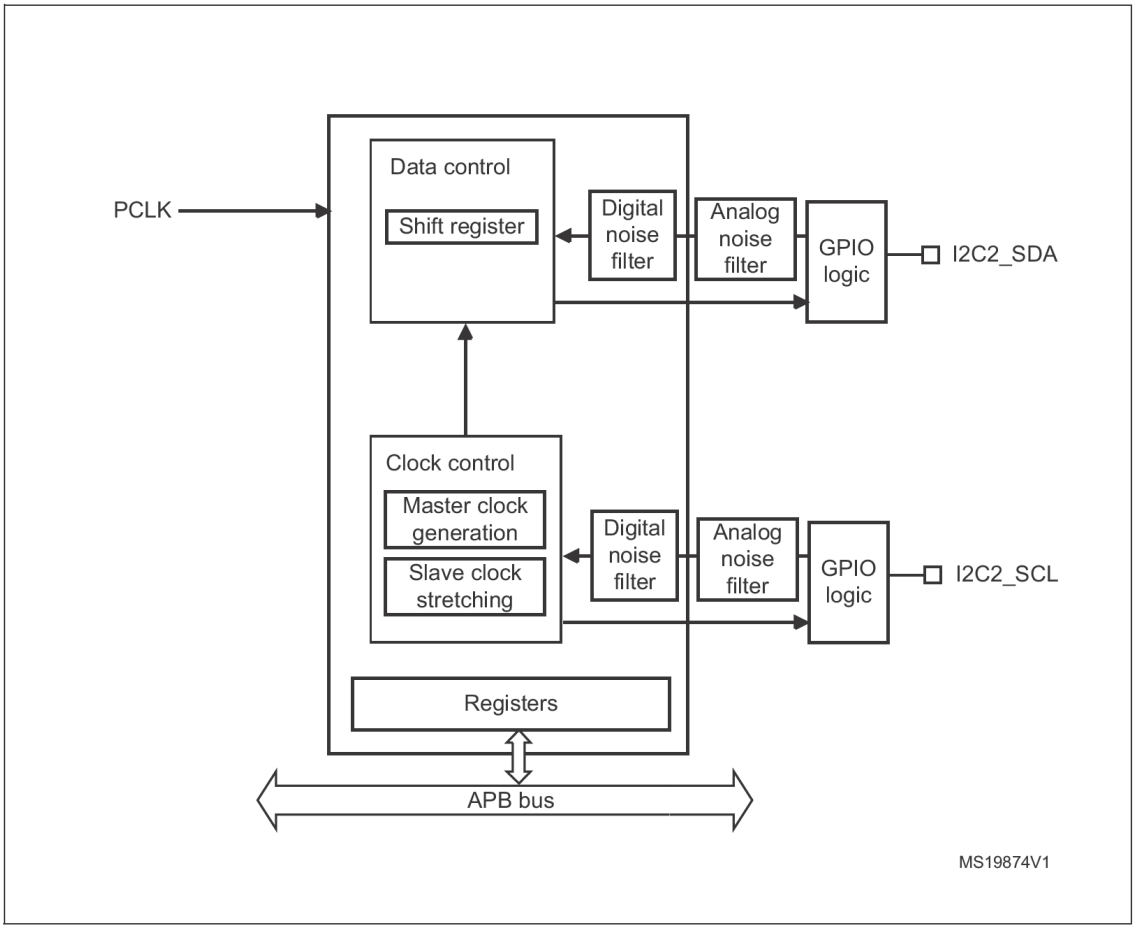
参见第 7 章： 复位和时钟控制系统（RCC）81 页。

I²C1 的 I/Os 支持 20 mA 的输出电流驱动以适应超快速模式的操作。 通过将 SCL 和 SDA 的驱动能力控制位置 1 来启用此设置，详见第 9.1.1 小节： SYSCFG 配置寄存器 1（SYSCFG_CFGR1）第 133 页。

23.4.2 I^2C2 框图

I^2C2 的框图如图 195 所示。

图 195. I^2C2 框图



23.4.3 I^2C 时钟要求

I2C 内核由 I2C_CLK 提供时钟。

I2C_CLK 的时钟周期 t_{I2C_CLK} 必须满足下列条件：

$t_{I2C\_CLK} < (t_{LOW}$ - tfilters) / 4 and $t_{I2C\_CLK} < t_{HIGH}$

其中：

t_{LOW}: SCL 低电平时间 和 tHIGH : SCL 高电平时间

tfilters: 启用时，模拟滤波器和数字滤波器所带来的延迟的总和。 模拟滤波器的延迟最大是 260 纳秒。 数字滤波器延时是 DNF x t_{I2C_CLK}.

PCLK 的时钟周期 t_{PCLK} 必须满足下列条件 :

$t_{PCLK} < 4/3\ t_{SCL}$

其中 t_{SCL}: SCL 周期

警告： 当 I2C 内核时钟由 PLCK 主频提供，PCLK 必须满足 t_{I2C_CLK} 的限制条件。

23.4.4 模式选择

接口可以工作在以下四个模式之一：

- 从机发送
- 从机接收
- 主机发送
- 主机接收

默认情况下，它在从机模式下运行。 接口在生成起始条件后自动地从从模式切换到主模式；当仲裁丢失或产生停止信号时，则从主模式切换到从模式。允许多主机功能。

通讯流

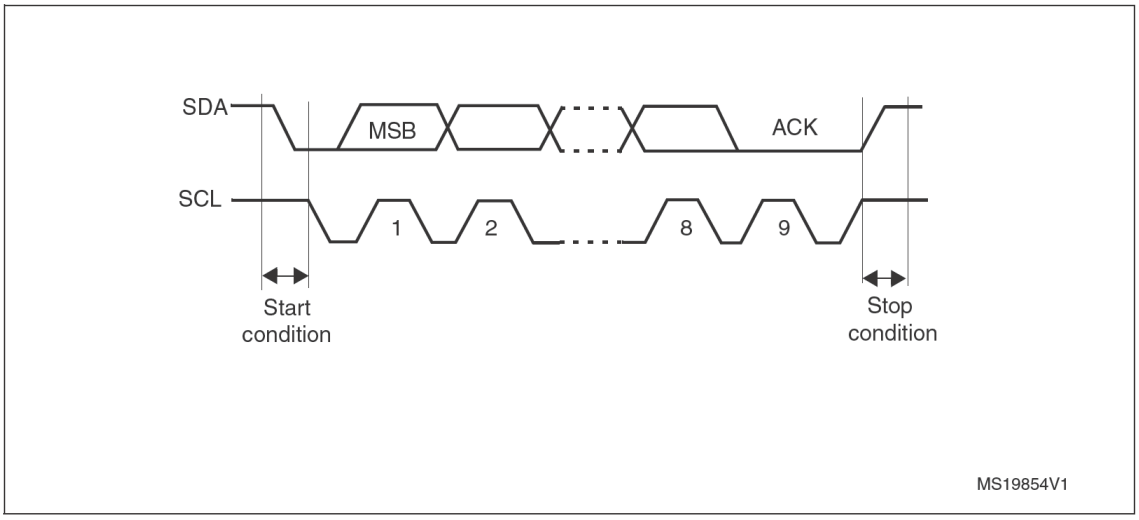
在主控模式下，I²C 接口启动数据传输，并产生时钟信号。 串行数据传输总是以起始条件开始并以停止条件结束。 起始条件和停止条件都是在主模式下由软件控制产生。

从模式时，I²C 接口能识别它自己的地址 (7 位或 10 位) 和广播呼叫地址。 软件能够控制开启或禁止广播呼叫地址的识别。 保留的 SMBus 地址，也可以由软件启用。

数据和地址按 8 位 / 字节进行传输，高位在前。 跟在起始条件后的 1 或 2 个字节是地址 (7 位模式为 1 个字节，10 位模式为 2 个字节)。 地址只在主模式发送。

在一个字节传输的 8 个时钟后的第 9 个时钟期间，接收器必须回送一个应答位 (ACK) 给发送器。请参阅下图。

图 196. I²C 总线协议



软件可以开启或禁止应答 (ACK)，并可以设置 I²C 接口的地址 (7 位、10 位地址或广播呼叫地址)。

23.4.5 I²C 的初始化

启用和禁用外设

I²C 外设时钟必须在时钟控制器中配置并启用（请参阅 7.2 节： 时钟 第 82 页。

然后，I²C 可以通过设置在 I²Cx_CR1 寄存器的 PE 位来使能。

当 I²C 被禁用（PE= 0）的时候，I²C 执行软件复位： I²C 线（SDA 和 SCL）都被释放。 内部状态机复位，所有的通信控制位和状态位回到它们的复位值。 受影响的位和 23.4.6 节列出那些是相同的： 软件复位

噪声滤波器

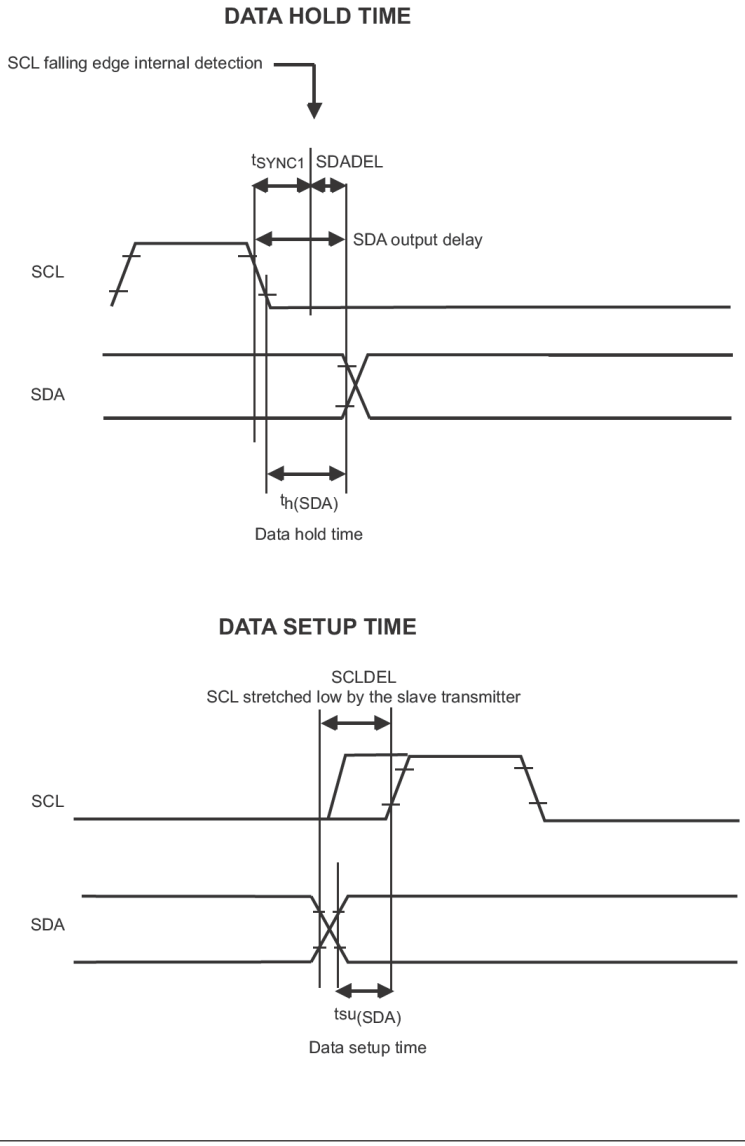
如有必要，在通过设置在 I²Cx_CR1 寄存器的 PE 位，启用 I²C 外设之前，你必须先配置噪声滤波器。 默认情况下，模拟噪声滤波器会处理 SDA 和 SCL 输入。 这个模拟滤波器符合快速模式和快速模式 Plus I²C 规范的需要，来抑制 50 纳秒以内的尖峰脉冲宽度。 可以通过设置的 ANFOFF 位来禁用这个模拟滤波器和 / 或配置 I²Cx_CR1 寄存器的 DNF[3:0] 位选择数字滤波器。

警告： 当 I²C 启用时不允许更改过滤器的配置。

的 I²C 时序

在主从模式中必须配置时序，以保证正确的数据保持和建立时间。 这通过编程 I²Cx_TIMINGR 寄存器中的 PRESC [3:0]，SCLDEL [3:0] 和 SDADEL [3:0] 位来实现。

图 197. 建立和保持时间



- 当内部检测 SCL 下降沿，在发送 SDA 输出前会插入一个延时。 这个延迟是 t_{SDADEL} = SDADEL x t_{PRESC} where t_{PRESC} = (PRESC+1) x t_{I2C_CLK}.

T_{SDADEL} 受保持时间的影响 tHD;DAT .

总的 SDA 输出延迟是：

tSYNC1 + [SDADEL x (PRESC+1) x tI2C_CLK]

tSYNC1 持续时间由下列参数决定：

— SCL 下降斜率

— 启用时，模拟滤波器所带来的输入延迟： tAF. 50ns < tAF < 260 ns.

— 启用时，数字滤波器所带来的输入延迟： tDNF = DNF x tI2C_CLK

— 由于 SCL 的同步 I2C_CLK 时钟（2 至 3 I2C_CLK 周期）造成的延迟

为了弥补 SCL 下降沿的未定义区域，你必须按下列方式，对 SDADEL 编程：

$\{t_{f\,(max)} + t_{HD;DAT\,(min)} - 50ns - [(DNF + 2) \times t_{I2C\_CLK}]\} / \{(PRESC + 1) \times t_{I2C\_CLK}\} <= SDADEL$

$SDADEL <= \{t_{HD;DAT\,(max)} - 260\ ns - [(DNF+3) \times t_{I2C\_CLK}]\} / \{(PRESC + 1) \times t_{I2C\_CLK}\}$

注： -50ns/-260ns 只有在启用了模拟滤波器的时候才是公式的一部分。

请参阅表 68： I2C-SMBUS 规范数据建立和保持时间来得到 tf 和 tHD;DAT 的标准值

- 在 SDA 发送输出后，在建立时间中，SCL 线保持在低电平。这个建立时间是
  $t_{SCLDEL} = (SCLDEL+1) \times t_{PRESC}$ where $t_{PRESC} = (PRESC+1) \times t_{I2C\_CLK}$.
  t_{SCLDEL} 受建立时间 $t_{SU;DAT}$. 的影响

为了弥补 SDA 发送（上升沿通常比较糟时）带来的的未定义区域，你必须按下列方式，对 SCLDEL 编程：

$\{[t_{r\,(max)} + t_{SU;DAT\,(min)}] / [(PRESC+1)] \times t_{I2C\_CLK}]\} - 1 <= SCLDEL$

请参阅表 68： I2C-SMBUS 规范数据建立和保持时间来得到 t_r and $t_{SU;DAT}$ 的标准值

表 68. I^2C SMBus 规范的数据建立和保持时间

	Parameter	Standard		Fast Mode		Fast Mode Plus		SMBUS	
		Min.	Max	Min.	Max	Min.	Max	Min.	Max
$t_{HD;DAT}$ (us)	Data hold time	0	3.45	0	0.9	0	0.45	300	
$t_{SU;DAT}$ (ns)	Data setup time	250		100		50		250	
tr(ns)	rise time of both SDA and SCL signals		1000		300		120		1000
t_f (ns)	fall time of both SDA and SCL signals		300		300		120		300

此外，在主模式下，SCL 时钟高和低电平，必须通过编程 I2Cx_TIMINGR 寄存器的 PRESC [3:0]、SCLH 的 [7:0] 和 SCLL [7:0] 位域来配置。

- 当内部检测 SCL 下降沿，在释放 SCL 输出之前会插入一个延时。 这个延迟是 $t_{SCLL} = (SCLL+1) \times t_{PRESC}$ 其中 $t_{PRESC} = (PRESC+1) \times t_{I2C\_CLK}$.
  t_{SCLL} 受 SCL 低时间 t_{LOW} 的影响 .
- 当内部检测 SCL 上升沿，在拉低 SCL 输出之前会插入一个延时。 这个延迟是 $t_{SCLH} = (SCLH+1) \times t_{PRESC}$ 其中 $t_{PRESC} = (PRESC+1) \times t_{I2C\_CLK}$.

t_{SCLH} 受 SCL 高时间 t_{HIGH} . 的影响

详情请参见： I^2C 主机初始化。

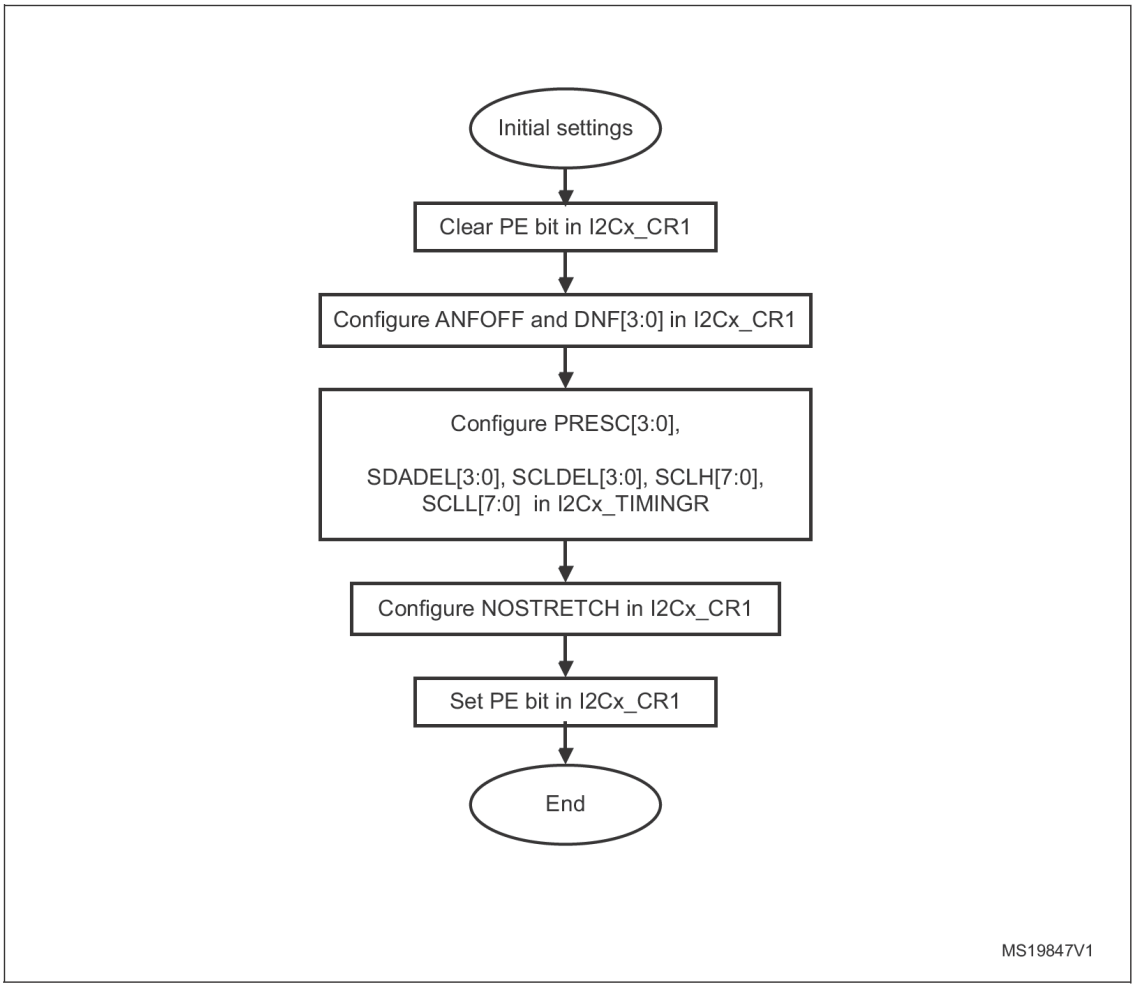
警告： 当 I^2C 启用时不允许更改时序配置。

I^2C 配置

I^2C 从机 NOSTRETCH 模式必须在在使能 I2C 外设之前配置。 参见： I2C 从机初始化。

警告： 当 I^2C 启用时不允许更改 NOSTRETCH 配置。

图 198. I^2C 初始化流程图



23.4.6 软件复位

将 I2Cx_CR1 寄存器的 SWRST 位置 1，可以实现软件复位。 在这种情况下，SCL 和 SDA I2C 线被释放。 内部状态机复位，所有的通信控制位和状态位回到它们的复位值。 而配置寄存器不会受到任何影响。

这是受影响的寄存器位的列表：
1. I2Cx_CR2 寄存器： START, STOP, NACK
2. I2Cx_ISR 寄存器： BUSY, TXE, TXIS, RXNE, ADDR, NACKF, TCR, TC, STOPF, BERR, ARLO, OVR

另外还有支持 SMBus 功能时：
1. I2Cx_CR2 寄存器： PECBYTE
2. I2Cx_ISR 寄存器： PECERR, TIMEOUT, ALERT

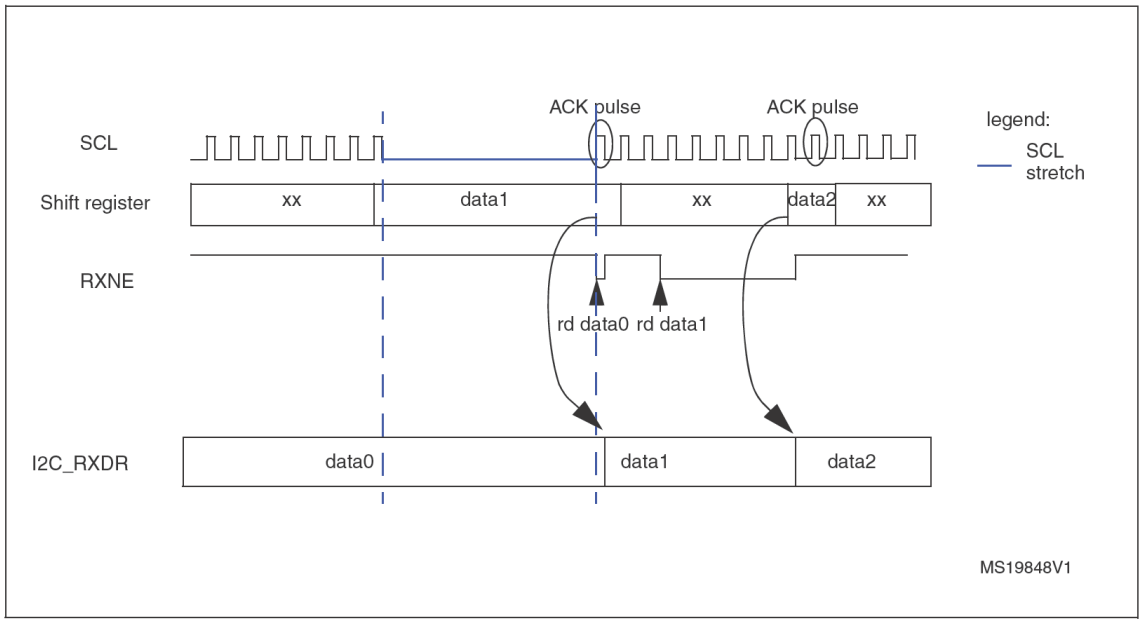
23.4.7 数据发送

要通过发送和接收数据寄存器和一个移位寄存器来管理数据发送。

接收

SDA 输入会填充移位寄存器。 在第 8 个 SCL 脉冲（当收到完整的数据字节）之后，如果 I2Cx_RXDR 寄存器是空的（RXNE = 0）移位寄存器的内容会被复制到 I2Cx_RXDR 寄存器。如果 RXNE = 1，也就是说，尚未读取之前收到的数据字节，这时 SCL 线被拉低直到 I2Cx_RXDR 被读取。 这个拉伸被插入到第 8 和第 9 个 SCL 脉冲之间（应答脉冲之前）。

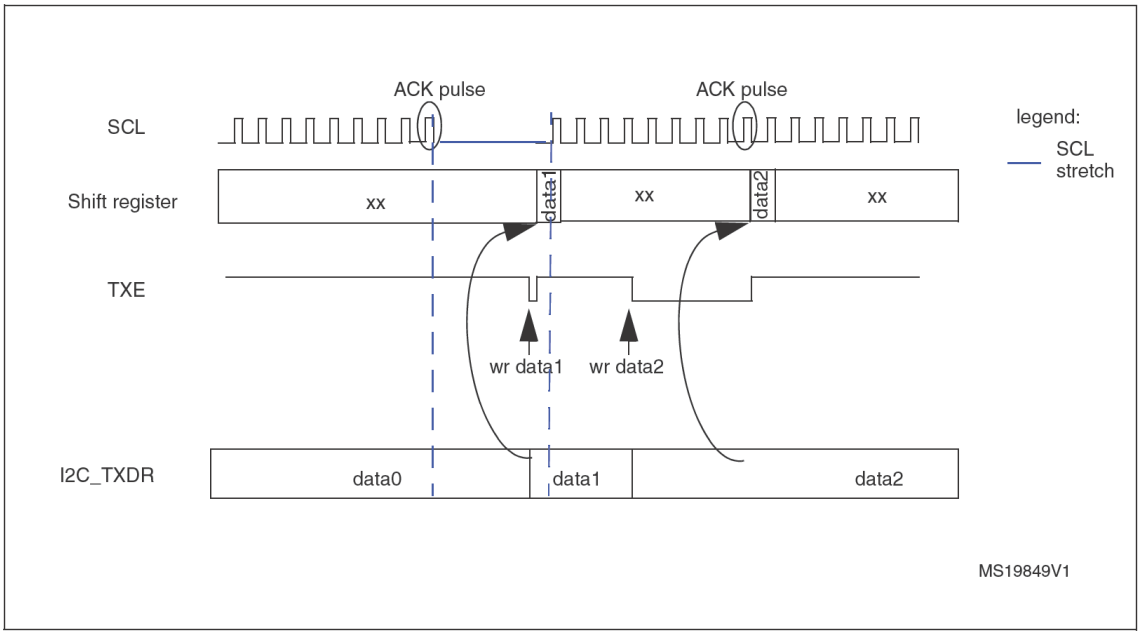
图 199. 数据接收



发送

如果 I2Cx_TXDR 寄存器不为空（的 TXE = 0），其内容将在第 9 个 SCL 脉冲（应答脉冲）之后被复制到移位寄存器。然后移位寄存器的内容被移到 SDA 线上。如果 TXE = 1，也就是说，I2Cx_TXDR 中没有数据被写入，SCL 线会被拉长为低直到 I2Cx_TXDR 有新数据被写入。 在第 9 个 SCL 脉冲之后完成拉伸。

图 200. 数据传输



硬件发送管理

I2C 有一个内嵌的字节计数器，可以管理发送的字节数，以便在各种模式中关闭通讯：

- 主模式下生成 NACK、STOP 和 ReSTART
- 从机接收模式下的 ACK 控制
- SMBus 的功能支持时的 PEC 的生成 / 检查

字节计数器总是在主模式下使用。 默认情况下，它在从模式下被禁用，但它可以通过软件设置 I2Cx_CR2 寄存器的 SBC 位（从字节控制）来启用。

要传输的字节数编程在 I2Cx_CR2 寄存器的 NBYTES[7:0] 位域。如果要传输的字节数（nBytes 个）大于 255，或一个接收器要控制收到的数据字节的应答值，就必须设置 I2Cx_CR2 寄存器的 RELOAD 位来选择重载模式。 在这种模式下，当 NBYTES 中编程的字节个数被发送完毕后，TCR 标志被置 1，如果 TCIE 为 1，则会产生一个中断。 SCL 在 TCR 标志为 1 期间被拉伸。 向 NBYTES 写入是一个非零的值时，TCR 标志由软件清除。

当 NBYTES 被重载为上次的字节数时，RELOAD 位必须被清零。

当主模式下 RELOAD = 0，计数器可以按下列 2 种模式工作：

- 自动结束模式（I2Cx_CR2 寄存器的 AUTOEND = 1）。 在这种模式下，一旦发送字节数达到了 NBYTES[7:0] 位域中设置的字节数，主机会自动发送一个停止条件。
- 软件结束模式（I2Cx_CR2 寄存器的 AUTOEND = 0）。 在这种模式下，发送字节数达到了 NBYTES[7:0] 位域中设置的字节数后需要软件的干预，这时 TC 标志会被置 1，如果 TCIE 位为 1，还会产生中断。 在 TC 标志为 1 期间，SCL 信号被拉伸。 在 I2Cx_CR2 寄存器的 START 或 STOP 位被置 1 时，TC 标志由软件清除。 主机要发送一个 RESTART 条件时，必须使用此模式。

警告： AUTOEND 位在重装位被置 1 时没有效果。

表 69. I2C 配置表

Function	SBC bit	RELOAD bit	AUTOEND bit
Master Tx/Rx NBYTES + STOP	x	0	1
Master Tx/Rx + NBYTES + RESTART	x	0	0
Slave Tx/Rx all received bytes ACKed	0	x	x
Slave Rx with ACK control	1	1	x

23.4.8 I²C 从机模式

I²C 从机初始化

要工作在从机模式下，您必须启用至少一个从机地址。 两个寄存器 I2Cx_OAR1 和 I2Cx_OAR2 都可以用来写入从机的本机地址 OA1 和 OA2。

- 通过设置 I2Cx_OAR1 寄存器的 OA1MODE 位，可将 OA1 配置在 7 位模式（默认）或 10 位地址模式。

通过设置 I2Cx_OAR1 寄存器的 OA1EN 位来启用 OA1。

- 如果需要额外的从机地址，你可以配置第二个从机地址 OA2。 配置 I2Cx_OAR2 寄存器的 OA2MSK [2:0] 位位域，可以最多屏蔽 OA2 的低 7 位。 因此从 1 到 6 为 OA2MSK 配置，只有 OA2 [7:2]，OA2 [7:3]，OA2 [7:4]，OA2 [7:5]，OA2 [7:6] 或 OA2 [7] 参加与接收到的地址的比较操作。 只要 OA2MSK 不等于 0，被 OA2 地址比较排除的 I2C 地址（0000 XXX 和 1111 xxx）不会被应答。 如果 OA2MSK = 7，收到的所有 7 位地址（保留地址除外）都会被应答。 OA2 始终是一个 7 位地址，不可以 10 位。如果他们启用特定的使能位，这些保留地址也可以应答，就是如果他们被写在 I2Cx_OAR1 或 I2Cx_OAR2 寄存器中并且 OA2MSK = 0。通过设置 I2Cx_OAR2 寄存器的 OA2EN 位来启用 OA2。
- I2Cx_CR1 寄存器的 GCEN 位被置 1 时，呼叫地址被启用。

当 I2C 由启用了的地址选中时，ADDR 中断状态标志被置 1，如果 ADDRIE 位为 1，就会产生一个中断。

默认情况下，从机使用它的时钟延长的功能，这意味着，它可根据需要在 SCL 信号拉低，以便执行软件动作。 如果主机不支持时钟延长，I2C 的 I2Cx_CR1 寄存器的 NOSTRETCH 必须配置为 1。

收到地址中断后，如果启用了多个地址，你必须读 I2Cx_ISR 寄存器中的 ADDCODE[6:0] 位以检查是哪一个地址匹配上了。 还要检查 DIR 标志，以了解数据传输的方向。

从时钟延长(NOSTRETCH = 0)

在默认模式下，I2C 从机于下列情况下拉低 SCL 时钟：

- 当址标志被置位： 接收到的地址和启用了的从机地址之一匹配上。 这种拖延在 ADDR 标志被软件清零后被释放。清零该位的办法是将 ADDRCF 位置 1。
- 在发送时，如果以前的数据传输完毕后，并没有新的数据被写到 I2Cx_TXDR 寄存器，或者如果在 ADDR 标志被清除（TXE = 1）后还没有写一个字节。 只要数据被写入 I2Cx_TXDR 寄存器，就会释放这个拖延。
- 在接收时如果 I2Cx_RXDR 寄存器的内容还没有被读走，又有一个新的数据被收进来。 这时延长在读取 I2Cx_RXDR 时被释放。
- 在从机字节控制模式，重载模式（SBC = 1 和 RELOAD= 1）时，如果 TCR = 1，意味着最后一个数据字节已发送完毕。在向 NBYTES[7:0] 位域写入一个非零值会清除 TCR 标志，这时延长也会释放。

从机不带时钟延长(NOSTRETCH = 1)

当 I2Cx_CR1 寄存器的 NOSTRETCH = 1 时，I2C 从机不会延长 SCL 信号。

- 在 ADDR 标志置位的时候 SCL 时钟不会延长。
- 发送中，数据必须在对于发送的第一个 SCL 脉冲之前写入到 I2Cx_TXDR 寄存器。 如果没有，会发生欠载，I2Cx_ISR 寄存器中的 UDR 标志被置位，如果 I2Cx_CR1 寄存器的 ERRIE 位为 1，会产生一个中断。 如果首个数据发送已经开始而 STOPF 位还是 1（未被清掉）时，OVR 标志也会被设置。 因此，如果你在写下次发送的首个发送数据之后才清除前一次传输的 STOPF 标志，必须先确认 OVR 状态，甚至于首个数据已经发送。
- 在接收时，必须在下一个字节的第 9 个 SCL 脉冲（ACK 脉冲）发生前，从 I2Cx_RXDR 寄存器将数据读走。 如果没有，会发生溢出，I2Cx_ISR 寄存器中的 OVR 标志被置位，如果 I2Cx_CR1 寄存器的 ERRIE 位为 1，会产生一个中断。

从机字节控制模式

为了允许从机接收模式的字节 ACK 控制，从机字节控制模式必须通过设置 I2Cx_CR1 寄存器的 SBC 位来启用。 这是与 SMBus 标准兼容所必需的。

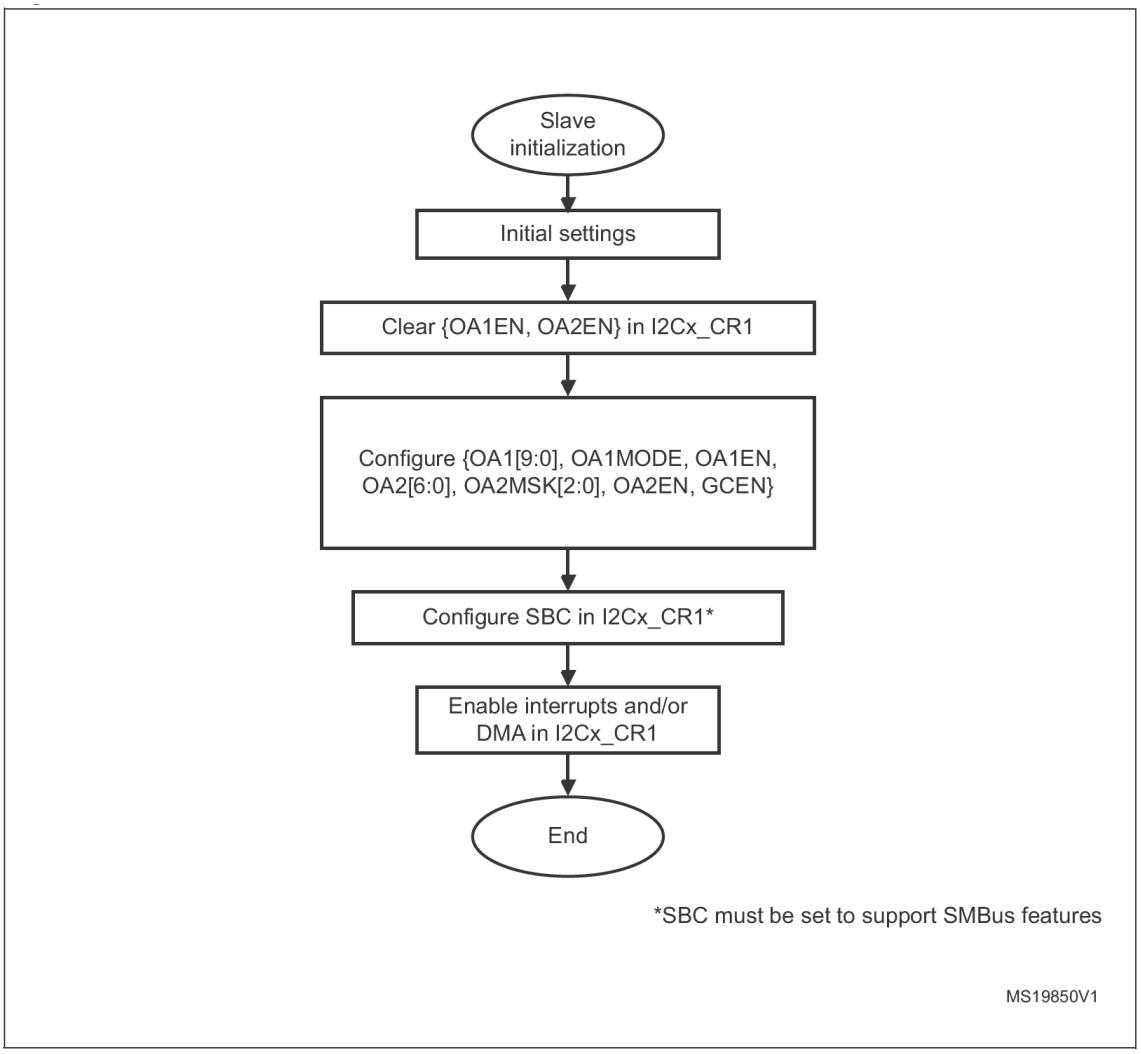
必须选择重载模式，以便允许在从机接收模式下的字节 ACK 控制（RELOAD = 1）。 要获得每个字节的控制，必须在 ADDR 中断服务程序中将 NBYTES 初始化为 0x1，并在每接收一个字节后重新加载为 0x1。 每当收到一个字节，TCR 位置 1，在第 8 和第 9 个 SCL 脉冲间，延长 SCL 信号为低。 可以从 I2Cx_RXDR 寄存器读数据，然后决定要不要通过配置 I2Cx_CR2 寄存器的 ACK 位来发 ACK。 对 NBYTES 写入一个非零值会释放 SCL 延长： ACK 或 NAK 被发送，然后可以接收下一字节。

可以写入一个比 0x1 更大的值到 NBYTES，在这种情况下，会连续接收 NBYTES 个数据。

注： *1 SBC 位必须在 I2C 被禁用时配置，或当从机没有被寻址到时，或者当 ADDR= 1 时。*
2 RELOAD 位的值可以在 ADDR=1 或者当 TCR=1 的时候改变。

警告： 从机字节控制模式不兼容 NOSTRETCH 模式。 不允许在 NOSTRETCH=1 的时候置位 SBC。

图 201. 从机初始化流程图



从机发送

当 I2Cx_TXDR 寄存器为空时会产生发送中断状态（TXIS）。 如果 I2Cx_CR1 寄存器中的 TXIE 位为 1，则会产生中断。

向 I2Cx_TXDR 寄存器写入下一个发送数据时，TXIS 位被清除。

当收到一个 NACK，I2Cx_ISR 寄存器中的 NACKF 位置 1，如果 I2Cx_CR1 寄存器的 NACKIE 位为 1，会产生一个中断。 从机会自动释放 SCL 和 SDA 线，这是为了允许主机执行停止或重新启动条件。 在收到一个 NACK 时 TXIS 位不会被置 1。

当收到一个 STOP 的时候 I2Cx_ISR 寄存器中的 STOPF 标志会被置 1，如果这时 I2Cx_CR1 寄存器的 STOPIE 又为 1，就会产生一个中断。 在大多数应用中，SBC 位通常被设定为 0 “。 在这种情况下，在 TXE 为 0 时收到从机地址（ADDR = 1），可以选择是将 I2Cx_TXDR 寄存器的内容当作第一个数据字节发送出去，还是将 TXE 位置 1，从而清空寄存器 I2Cx_TXDR 以便写一个新的数据字节进去。

在从机字节控制模式（SBC= 1），要发送的字节数必须在地址匹配（ADDR=1）中断服务程序中写入到 NBYTES。 在这种情况下，在传输过程中的 TXIS 事件个数与 NBYTES 中写入的值对应。

警告： 当 NOSTRETCH = 1，SCL 时钟在 ADDR 标志被置位的时候不延长，所以你不能够在 ADDR 子程序中清除 I2Cx_TXDR 寄存器的内容而定义第一个数据字节。 要发送的第一个数据字节，必须预先写到 I2Cx_TXDR 寄存器：

- 这个数据可以是前一次发送消息的时候 TXIS 事件里写入的数据。
- 如果这个数据字节不是要发送的那个，I2Cx_TXDR 寄存器可以随着 TXE 位的置位而被清空，以便写入一个新的数据字节。 STOPF 位必须在这些动作之后被清除，以保证他们在第一个数据传输开始前就执行，跟着地址的 ACK。
  如果 STOPF 在第一个数据传输开始时仍然为 1，会产生欠载错误（OVR 标志被置 1）。
  如果需要 1 个 TXIS 事件，（发送中断或发送 DMA 请求），你除了置 TXE 位以外还要置 TXIS 位，这是为了产生 TXIS 的事件。

图 202. I2C 从机发送的发送顺序流程图，NOSTRETCH = 0

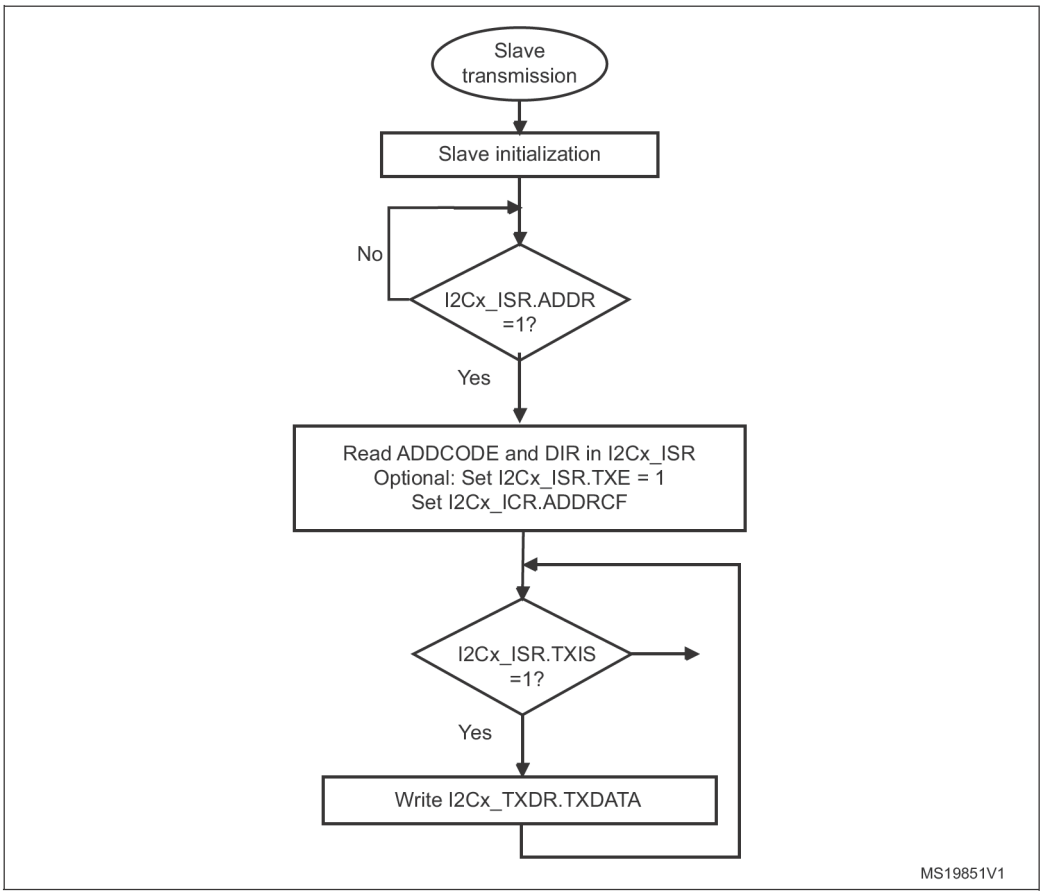


图 203. I2C 从机发送的发送顺序流程图，NOSTRETCH = 1

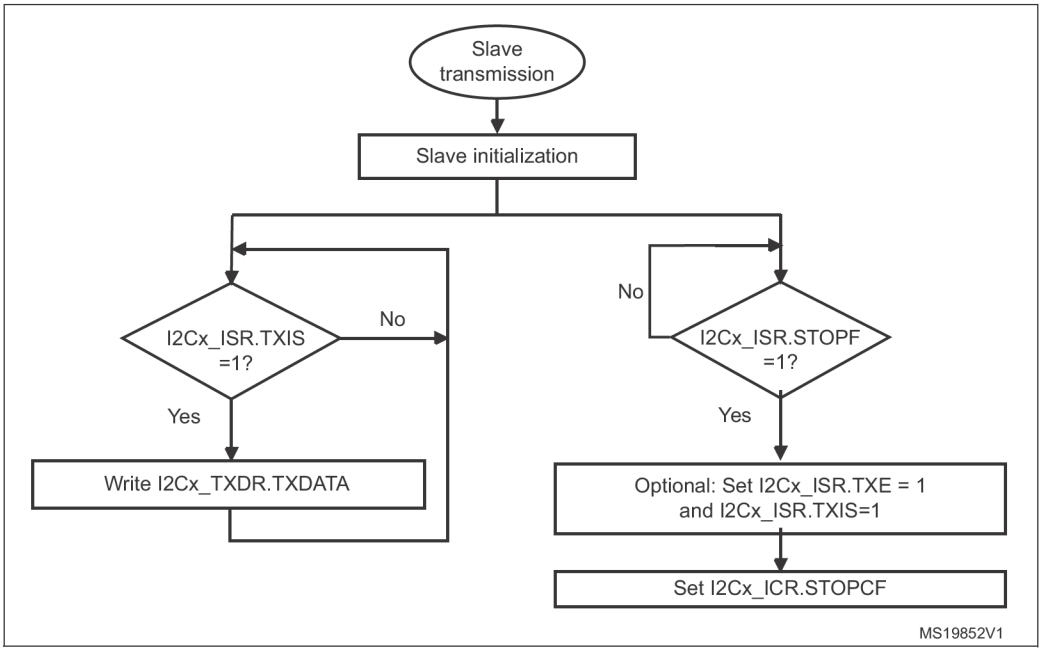
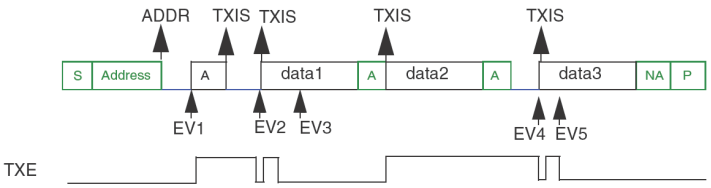


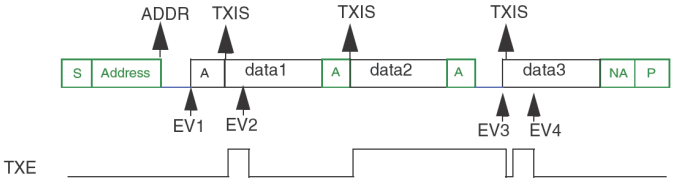
图 204. I2C 从机发送机传输总线图

Example I2C slave transmitter 3 bytes with 1st data flushed, NOSTRETCH=0:



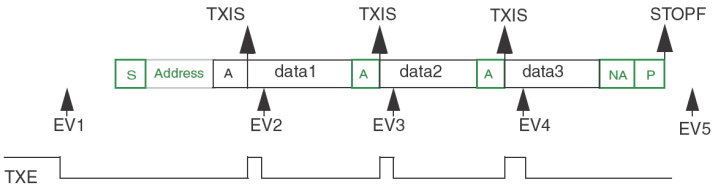
EV1: ADDR ISR: check ADDCODE and DIR, set TXE, set ADDRCF
EV2: TXIS ISR: wr data1
EV3: TXIS ISR: wr data2
EV4: TXIS ISR: wr data3
EV5: TXIS ISR: wr data4 (not sent)

Example I2C slave transmitter 3 bytes without 1st data flush, NOSTRETCH=0:



EV1: ADDR ISR: check ADDCODE and DIR, set ADDRCF
EV2: TXIS ISR: wr data2
EV3: TXIS ISR: wr data3
EV4: TXIS ISR: wr data4 (not sent)

Example I2C slave transmitter 3 bytes, NOSTRETCH=1:



EV1: wr data1
EV2: TXIS ISR: wr data2
EV3: TXIS ISR: wr data3
EV4: TXIS ISR: wr data4 (not sent)
EV5: STOPF ISR: (optional: set TXE and TXIS), set STOPCF

MS19853V1

从机接收器

在 I2Cx_RXDR 收满时 I2Cx_ISR 寄存器的 RXNE 被置 1，如果 I2Cx_CR1 寄存器的 RXIE 为 1，会产生一个中断。 在读取 I2Cx_RXDR 时 RXNE 会被清除。

在收到一个 STOP 条件，并且 I2Cx_CR1 的 STOPIE 被置 1，I2Cx_ISR 寄存器的 STOPF 位会被置 1，并产生一个中断。

图 205. I2C 从机接收器的传输顺序流程图，NOSTRETCH = 0

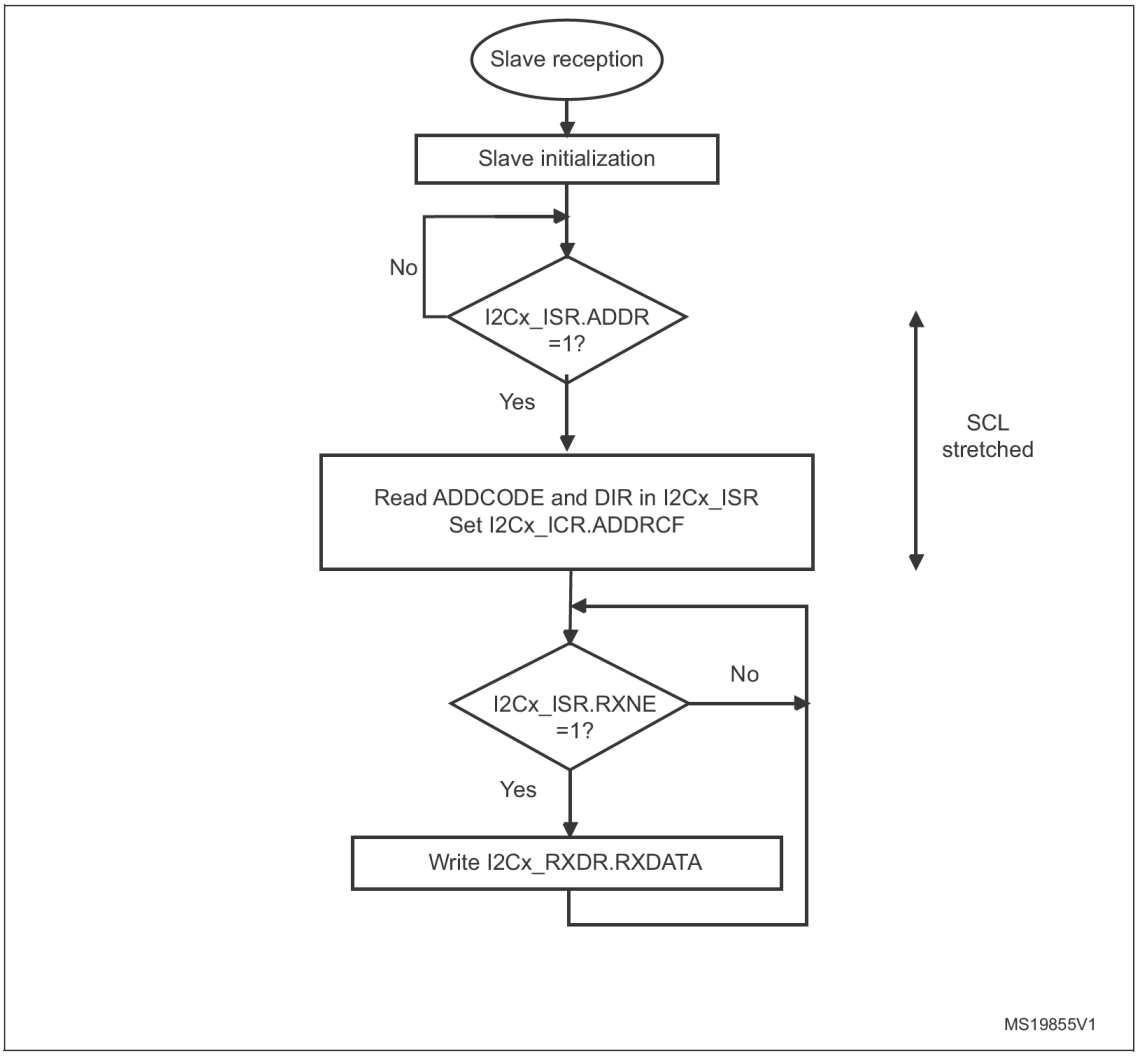


图 206. I2C 从机接收器的传输顺序流程图，NOSTRETCH = 1

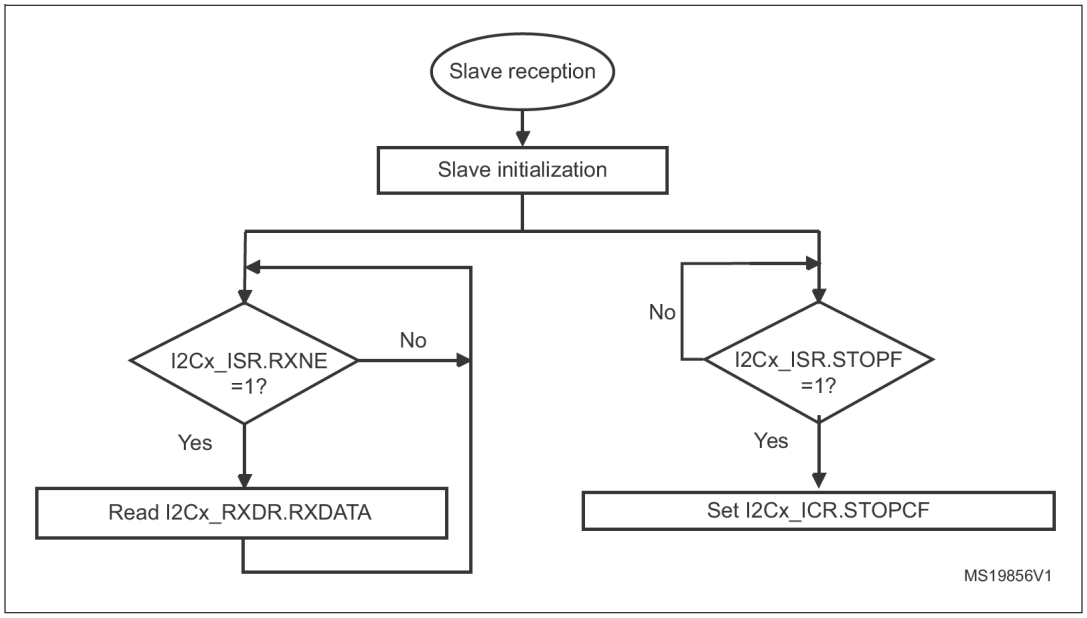
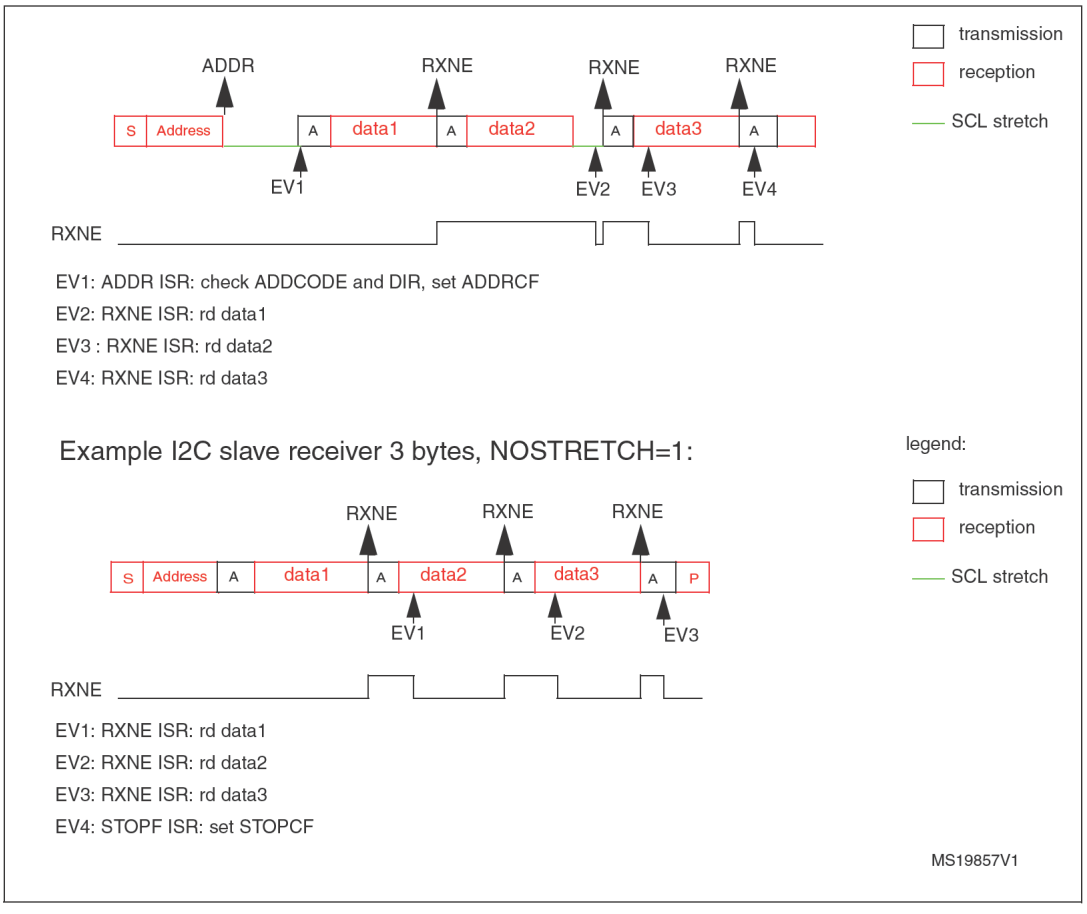


图 207. I2C 从机接收机的传输总线图



23.4.9 I^2C 主模式

I^2C 主机初始化

在启动外设之前，必须设置 I2Cx_TIMINGR 寄存器的 SCLH 和 SCLL 的位来配置 I^2C 主时钟。

这会实施一个时钟同步机制，以支持多主机环境和从机时钟延长。

为了让时钟同步：

- 从 SCL 低电平的内部检测开始用 SCLL 计数器来计数时钟低电平的个数。
- 从 SCL 高电平的内部检测开始用 SCLH 计数器来计数时钟高电平的个数。

I2C 依靠 SCL 下降沿的一个 tSYNC1 延迟和 SCL 输入噪声滤波器（模拟 + 数字）来检测自己的 SCL 低电平，并将 SCL 同步到 I2Cx_CLK 时钟。 一旦 SCLL 计数器达到了 I2Cx_TIMINGr 寄存器的 SCLL[7:0] 位域中的编程值，I2C 释放 SCL 到高电平。

I2C 依靠 SCL 上升沿的一个 tSYNC2 延迟和 SCL 输入噪声滤波器（模拟 + 数字）来检测自己的 SCL 高电平，并将 SCL 同步到 I2Cx_CLK 时钟。 一旦 SCLH 计数器达到了 I2Cx_TIMINGR 寄存器的 SCLH[7:0] 位域中的编程值，I2C 将 SCL 拉低到低电平。

因此，主机时钟周期是：

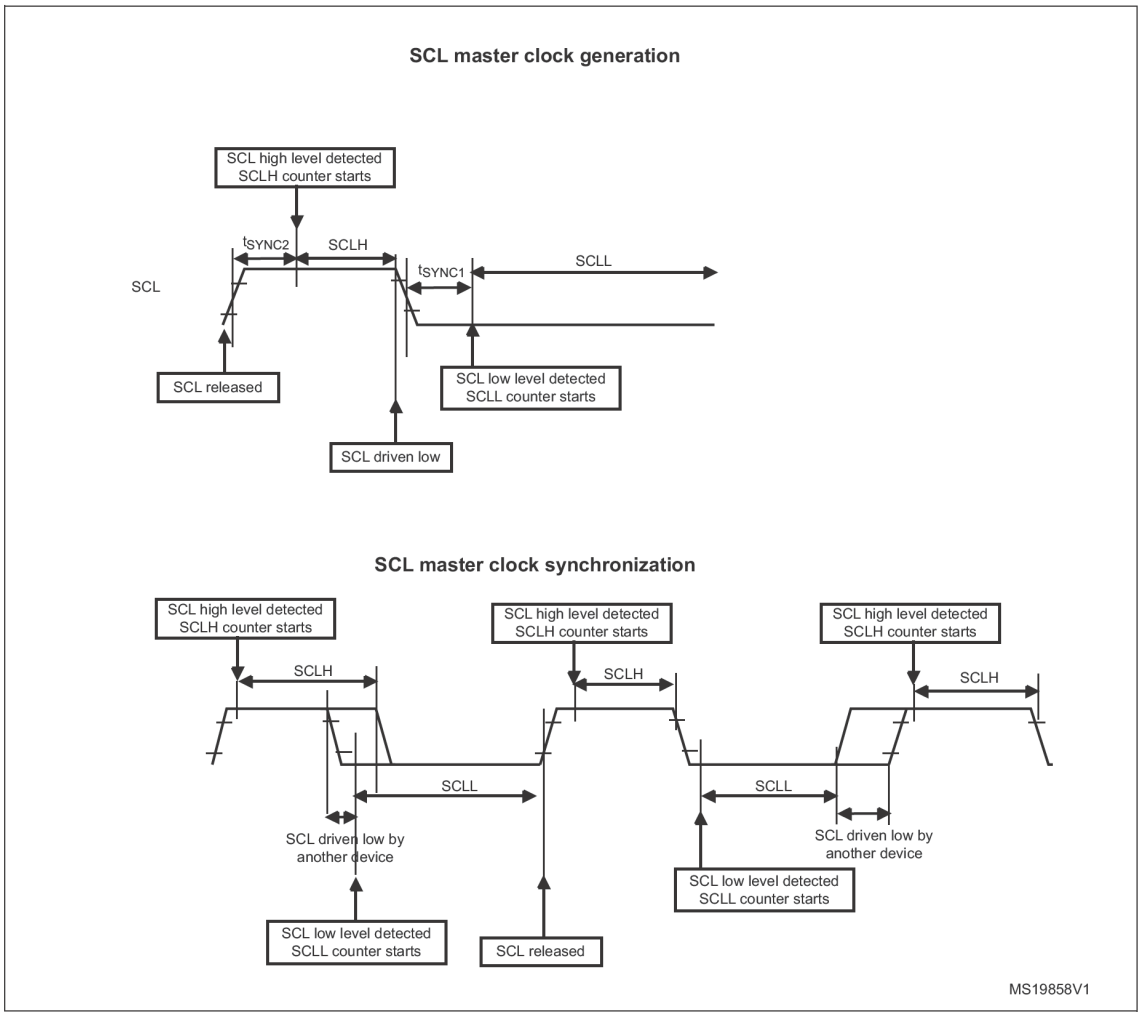
$t_{SCL} = t_{SYNC1} + t_{SYNC2} + \{ [(SCLH+1) + (SCLL+1)] \times (PRESC+1) \times t_{I2C\_CLK} \}$ t_{SYNC1} 时间取决于以下参数：

- SCL 下降斜率
- 启用时，模拟滤波器所带来的输入延迟：
- 启用时，数字滤波器所带来的输入延迟： DNF x T_{I2C_CLK}
- 由于 SCL 的同步 I2C_CLK 时钟（2 至 3 I2C_CLK 周期）造成的延迟

t_{SYNC2} 时间取决于以下参数：

- SCL 的上升斜率
- 启用时，模拟滤波器所带来的输入延迟：
- 启用时，数字滤波器所带来的输入延迟： DNF x T_{I2C_CLK}
- 由于 SCL 的同步 I2C_CLK 时钟（2 至 3 I2C_CLK 周期）造成的延迟

图 208. 主机时钟的产生



警告： 为了与 I2C 或 SMBus 兼容，主机时钟必须保证的时序如下：

表 70. I2C SMBus 规范的时钟时序

Symbol	Parameter	Standard		Fast Mode		Fast Mode Plus		SMBUS		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	
f_{SCL}	SCL clock frequency		100		400		1000		100	kHz
$t_{HD:STA}$	Hold time (repeated) START condition	4.0		0.6		0.26		4.0		µs
$t_{SU:STA}$	Set-up time for a repeated START condition	4.7		0.6		0.26		4.7		µs
$t_{SU:STO}$	Set-up time for STOP condition	4.0		0.6		0.26		4.0		µs

表 70. I^2C SMBus 规范的时钟时序（续）

Symbol	Parameter	Standard		Fast Mode		Fast Mode Plus		SMBUS		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	
t_{BUF}	Bus free time between a STOP and START condition	4.7		1.3		0.5		4.7		µs
t_{LOW}	Low period of the SCL clock	4.7		1.3		0.5		4.7		µs
t_{HIGH}	Period of the SCL clock	4.0		0.6		0.26		4.0	50	µs
t_r	Rise time of both SDA and SCL signals		1000		300		120		1000	ns
t_f	Fall time of both SDA and SCL signals		300		300		120		300	ns

注： *SCLL 也被用来产生 TBUF 和 TSU：STA 时序。*
SCLH 也被用来产生 THD：STA 和 TSU：STO 时序。

详情参见章节 23.4.10： I2Cx_TIMINGR 寄存器配置的例子： I2Cx_TIMINGR 设置与 I2C_CLK 频率的例子。

主机通信初始化（地址阶段）

为了启动通讯，必须在 I2Cx_CR2 寄存器中设置从机地址的下列参数：

- 地址模式（7 位或 10 位）： ADD10
- 要发送的从机地址： SADD [9:0]
- 传输方向： RD_WRN
- 在 10 位地址的情况下读取： HEAD10R 位。 在必须发送完整的地址顺序时，HEAD10R 必须被先置位，以示在方向改变时仅需要帧头的区别。
- 要传输的字节数： NBYTES[7:0]. 如果字节数大于等于 255 个字节，NBYTES[7:0] 最初必须使用 0xff 填充。

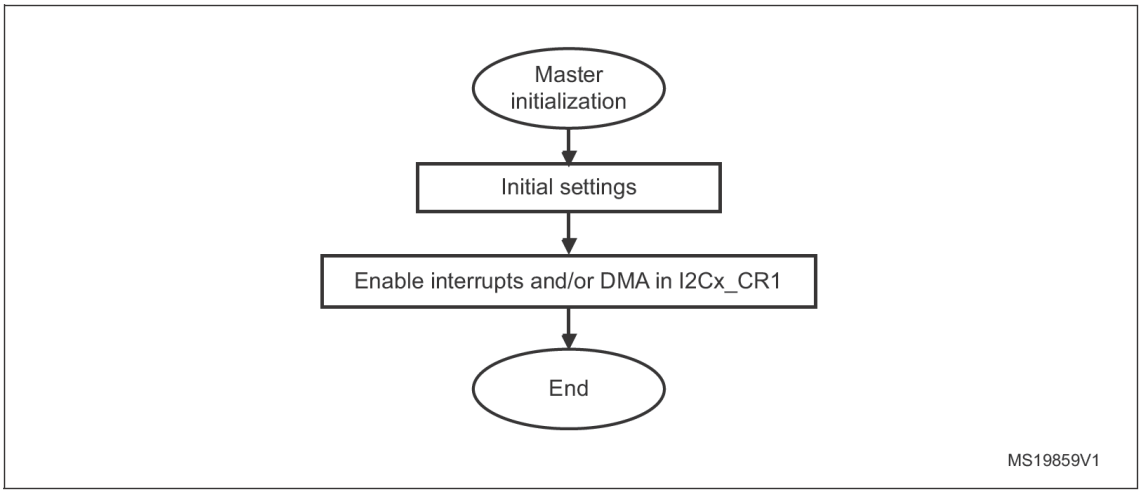
随后必须设置 I2Cx_CR2 寄存器的 START 位。 START 位一旦被置 1，就不再允许更改上述所有位。

只要检测到总线是空闲（BUSY = 0）的并插入一个延时后，主机自动发送 START 条件，随后就是发送从机地址。
主在仲裁丢失的情况下，主机自动切换回从模式，如果从机地址被选中，还将会自动发送 ACK 应答。

注： *在从机地址在总线上面发出后，无论收到 ACK 值的内容是什么，START 位都由硬件复位。 如果发生了仲裁丢失。START 位也由硬件清零。在 START 位为 1 期间，如果 I2C 作为一个从机（ADDR= 1）被选中，I2C 会切换到从机模式，并且在 ADDRCF 位被置 1 时 START 位被清零。*

注： *相同的程序适用于重复开始条件。 在这种情况下，BUSY = 1。*

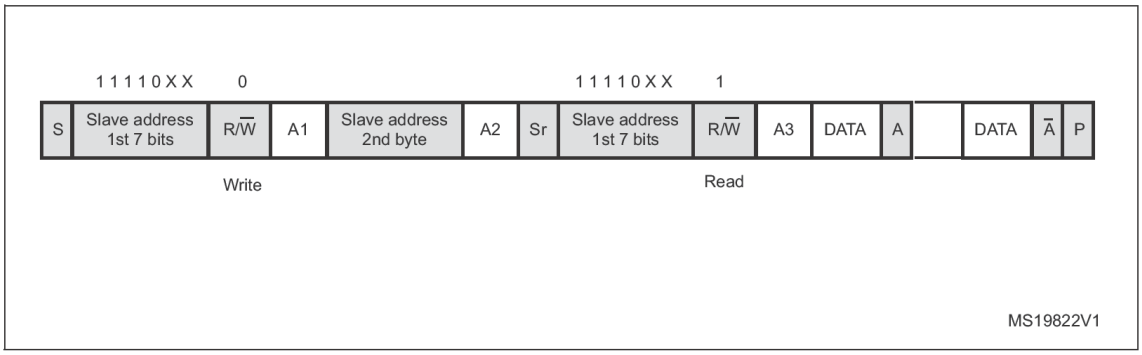
图 209. 主机初始化流程图



主机接收器寻址一个 10 位地址的从机的初始化

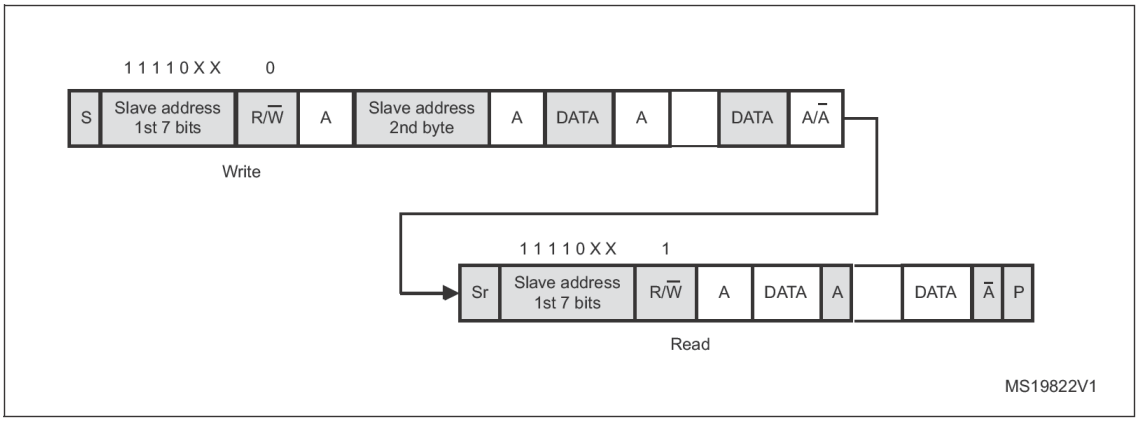
- 如果从机地址是 10 位格式，你可以选择清除 I2Cx_CR2 寄存器的 HEAD10R 位以发送一个完整的读序列。在这种情况下，主机会在 START 位被置 1 后，自动发送下面的完整序列: (Re)Start + 从机地址 10 位 头的写操作 + 从机地址的第二字节 + REStart + 从机地址 10 位头的读操作

Figure 210. 10-bit address read access with HEAD10R=0



- 如果主机要寻址到一个 10 位地址的从机，将数据发送给这个从机，然后从同一个从机读取数据，主机发送流程必须首先完成。 然后以 HEAD10R = 1 的状态重复开始条件和 10 位从机地址。 在这种情况下，主机发送序列如下： ReStart+ 从机地址 10 位头的读操作

图 211. 10 位地址的读访问 HEAD10R = 1



主机发送器

在写传输的情况下，每个字节发送完之后，TXIS 标志会被置 1，也就是在第 9 个 SCL 脉冲的时候收到一个 ACK。

如果 I2Cx_CR1 寄存器的 TXIE 位为 1，则会由 TXIS 事件产生一个中断。 向 I2Cx_TXDR 寄存器写入下一个发送数据时，这个标志位被清除。

在传输过程中的 TXIS 事件个数与 NBYTES 中写入的值对应。 如果数据要发送的字节总数大于 255，必须将 I2Cx_CR2 寄存器的 RELOAD 位置 1 以选择重加载模式。 这种情况下，发送了 NBYTES 个字节之后，TCR 标志被置位，并且 SCL 线被拉低直至 NBYTES[7:0] 中被写入一个非零值。

在收到一个 NACK 时 TXIS 位不会被置 1。

- 当 RELOAD= 0 以及 NBYTES 个数据已传输完：
  - 自动结束模式（AUTOEND = 1）下，会自动发送一个 STOP 条件。
  - 在软件结束模式下（AUTOEND = 0），TC 标志被置 1，并且 SCL 线被拉低，这时要软件执行下一步的动作：
    这时如果已经配置好从机地址和要传送的字节数，就可以将 I2Cx_CR2 中的 START 位置 1，再次发出一个 START 条件。 将 START 位置 1 的操作将会清除 TC 标志，并在总线上发出 START 条件。
    可以将 I2Cx_CR2 寄存器的 STOP 位置 1 来发出停止条件。 将 STOP 位置 1 的操作将会清除 TC 标志，并在总线上发出 STOP 条件。
- 如果收到一个 NACK： TXIS 标志不会被置 1，并在收到 NACK 之后自动发送一个 STOP 条件。I2Cx_ISR 寄存器的 NACKF 标志会被置 1，这时如果 NACKIE 位为 1，就会产生中断。

图 212. N<=255 字节时的 I2C 主机发送器传输顺序流程图

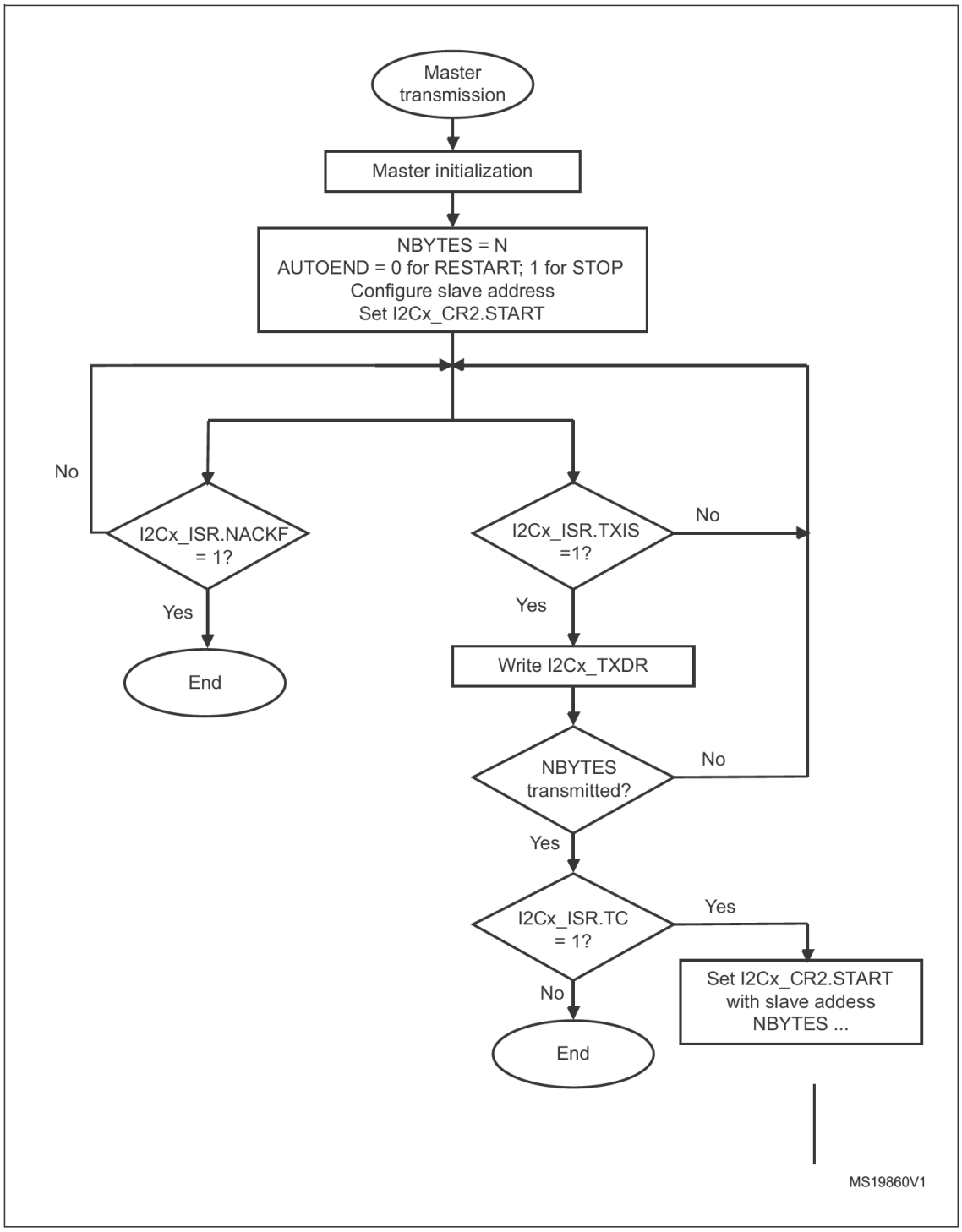


图 213. N>255 字节时的 I2C 主机发送器传输顺序流程图

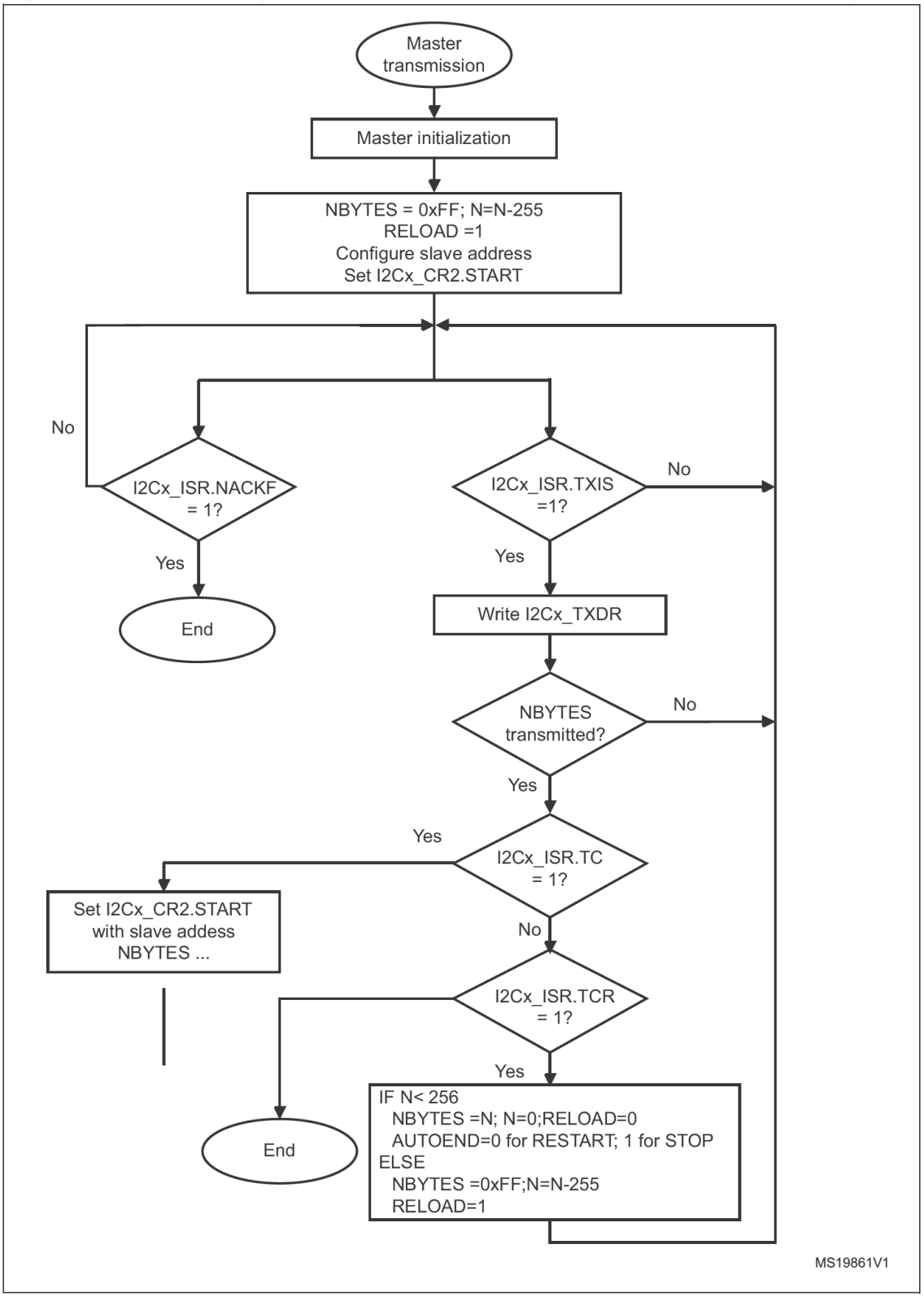
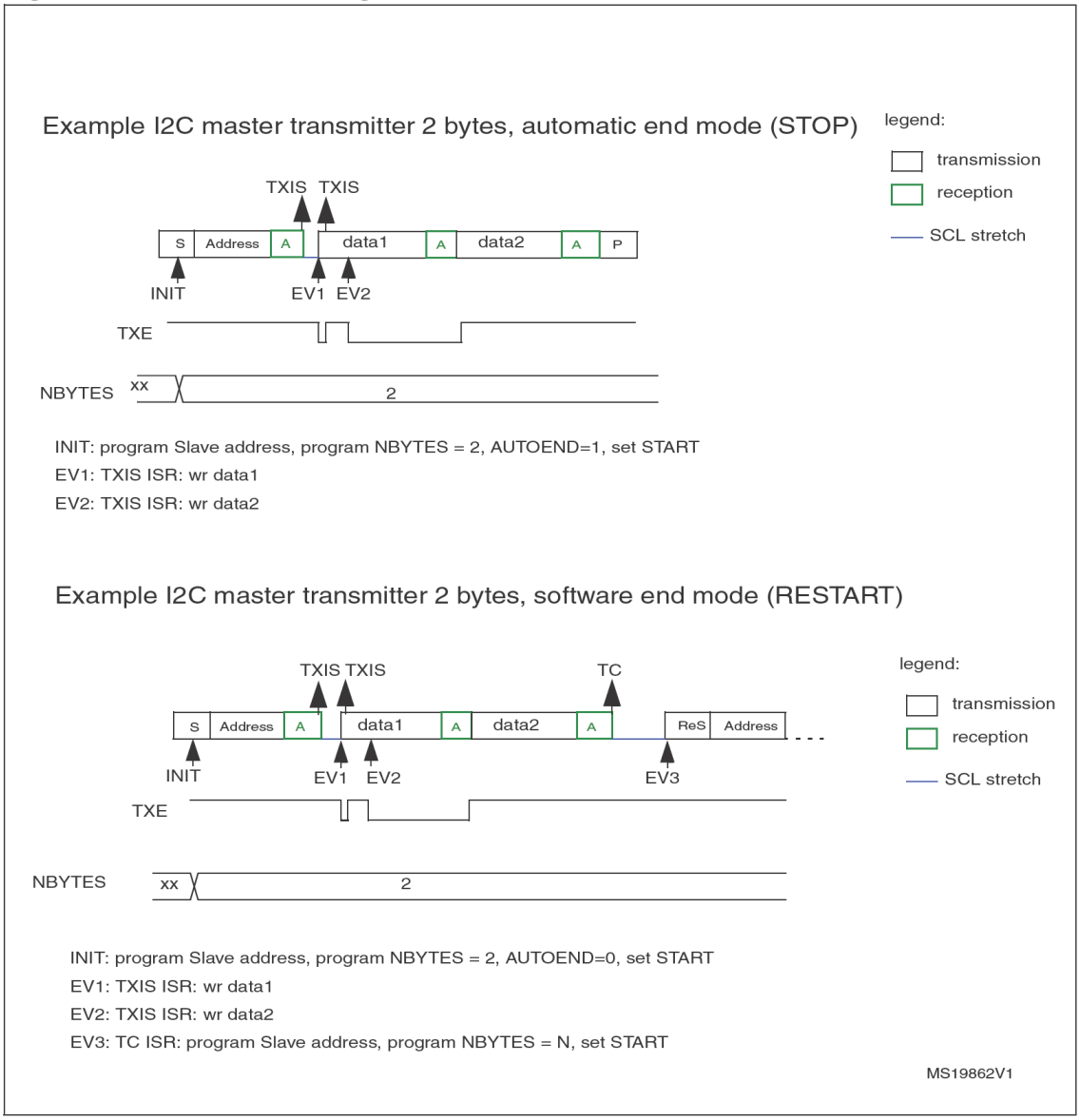


图 214. I²C 主机发送传输时序图



主机接收器

在读传输的时，在每个字节接收完后，第 8 个 SCL 脉冲时，RXNE 标志会置 1。 如果 I2Cx_CR1 寄存器的 RXIE 位为 1，则会由 RXNE 事件产生一个中断。 在读取 I2Cx_RXDR 时 RXNE 会被自动清零。

如果数据要接收的字节总数大于 255，必须将 I2Cx_CR2 寄存器的 RELOAD 位置 1 以选择重加载模式。 这种情况下，发送了 NBYTES 个字节之后，TCR 标志被置位，并且 SCL 线被拉低直至 NBYTES[7:0] 中被写入一个非零值。

- 当 RELOAD= 0 以及 NBYTES 个数据已传输完：
  - 自动结束模式（AUTOEND = 1）下，在收到最后一个字节后会自动发送一个 NACK 和一个 STOP 条件。
  - 在软件结束模式（AUTOEND = 0）下，在收到最后一个字节后会自动发送一个 NACK，然后 TC 标志被置 1，并且 SCL 线被拉低，这时要软件来执行后一步的操作：
    这时如果已经配置好从机地址和要传送的字节数，就可以将 I2Cx_CR2 中的 START 位置 1，再次发出一个 START 条件。 将 START 位置 1 的操作将会清除 TC 标志，并在总线上发出 START 条件及从机地址。
    可以将 I2Cx_CR2 寄存器的 STOP 位置 1 来发出停止条件。 将 STOP 位置 1 的操作将会清除 TC 标志，并在总线上发出 STOP 条件。

图 215. N<=255 字节时的 I2C 主机接收器传输顺序流程图

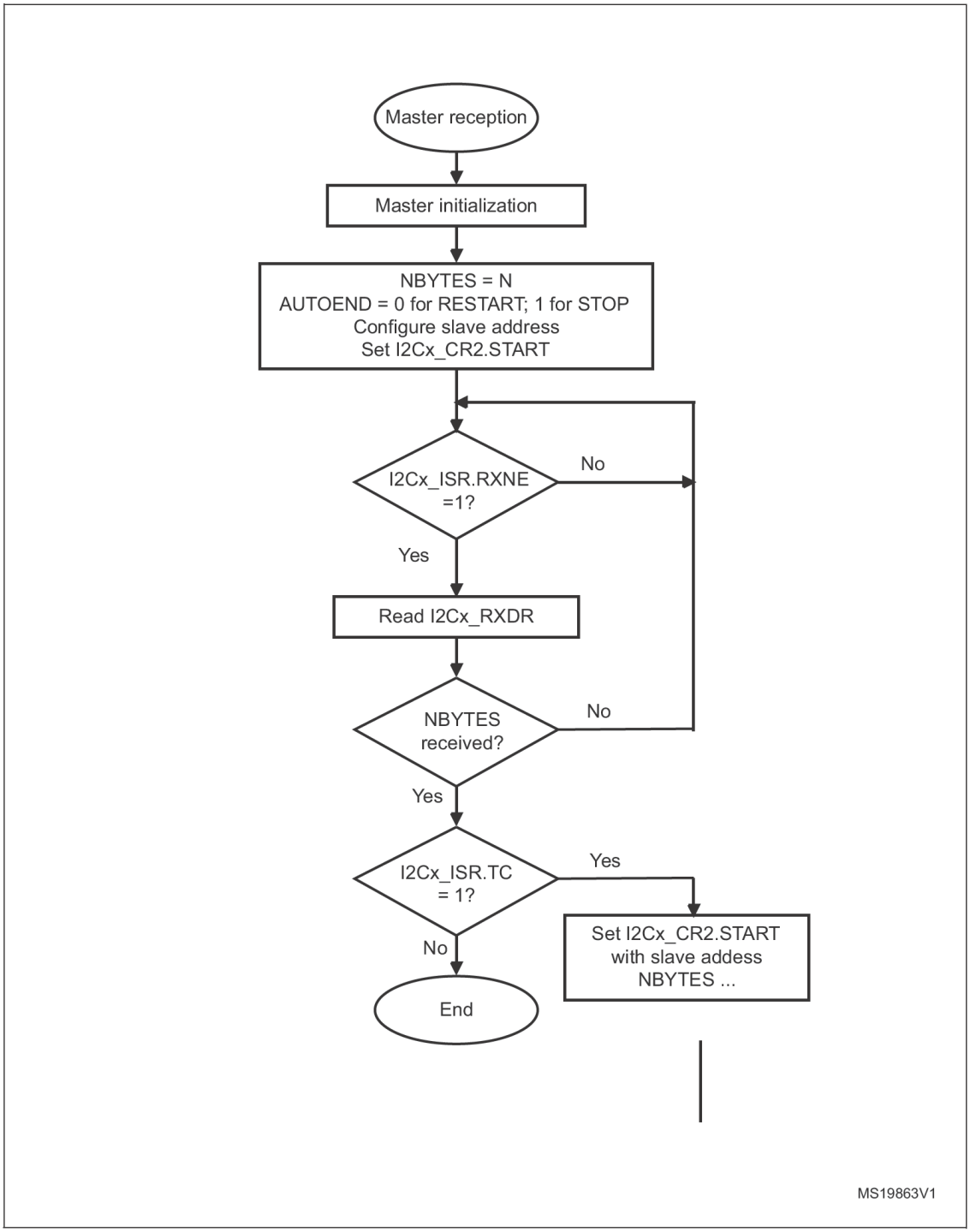


图 216. N>255 字节时的 I2C 主机接收器传输顺序流程图

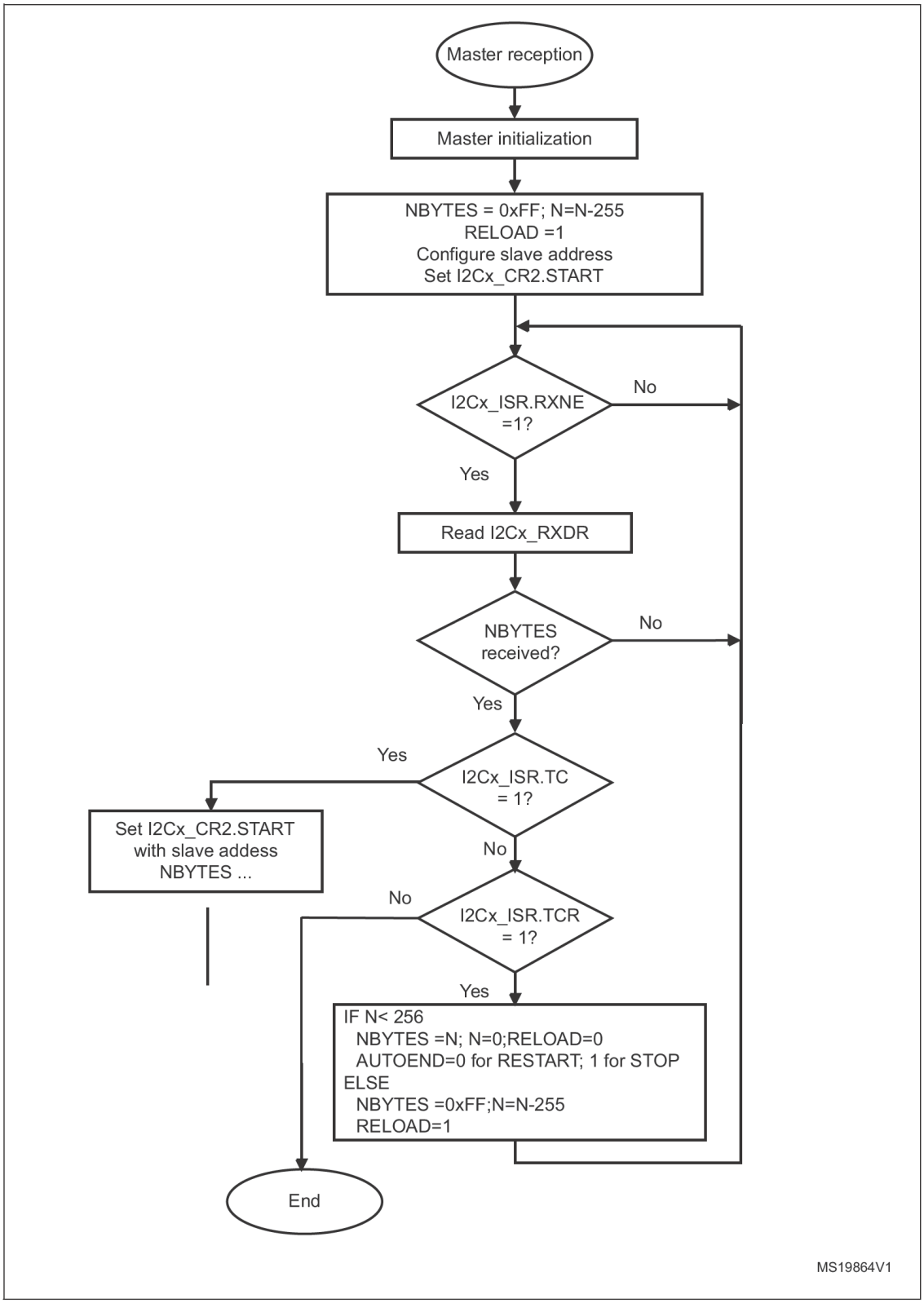
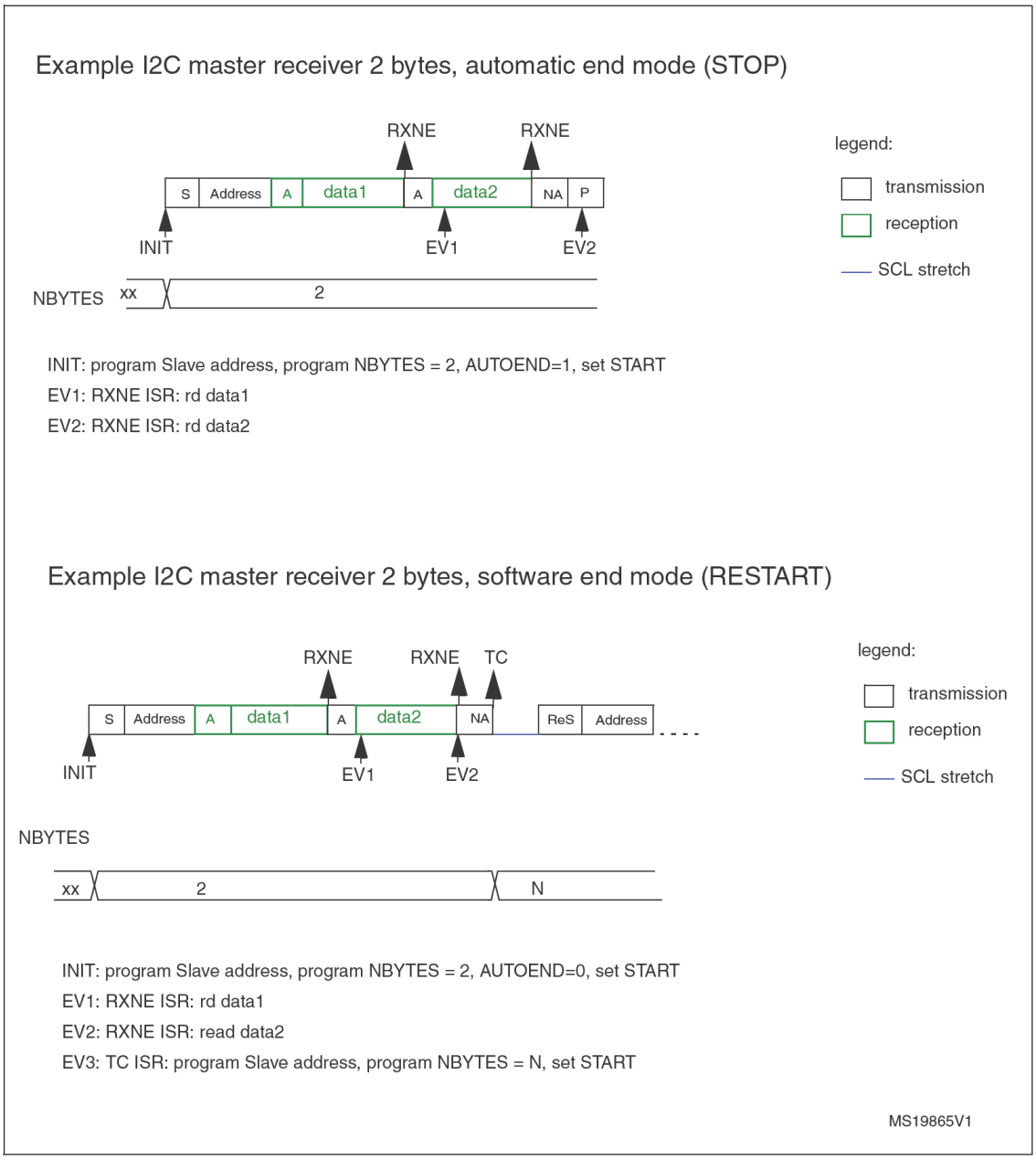


图 217. I^2C 主机接收传输时序图



23.4.10 I2Cx_TIMINGR 寄存器配置举例:

表 71. f_{I2C_CLK} = 8 MHz 的时序设置的例子

Parameter	Standard mode		Fast Mode	Fast Mode Plus
	10 kHz	100 kHz	400 kHz	500 kHz
PRESC	1	1	0	0
SCLL	0xC7	0x13	0x9	0x6
t_{SCLL}	200x250ns = 50µs	20x250ns = 5.0µs	10x125ns = 1250ns	7x125ns = 875ns
SCLH	0xC3	0xF	0x3	0x3
t_{SCLH}	196x250ns = 49µs	16x250ns = 4.0µs	4x125ns = 500ns	4x125ns = 500ns
t_{SCL}(1)	~100 µs(2)	~10 µs(2)	~2500 ns(3)	~2000 ns(4)
SDADEL	0x2	0x2	0x1	0x1
t_{SDADEL}	2x250 ns = 500 ns	2x250 ns = 500 ns	1x125 ns = 125 ns	1x125 ns = 125 ns
SCLDEL	0x4	0x4	0x3	0x1
t_{SCLDEL}	5x250 ns = 1250 ns	5x250 ns = 1250 ns	4x125 ns = 500 ns	2x125 ns = 250 ns

1. 由于SCL内部检测带来的延迟，SCL 周期 tSCL 要大于 $t_{SCLL} + t_{SCLH}$. 提供的 t_{SCL} 值只是举例 .
2. $t_{SYNC1} + t_{SYNC2}$ 的最小值是 4 x t_{I2C_CLK} = 500 ns. 基于例子 $t_{SYNC1} + t_{SYNC2}$ = 1000 ns
3. $t_{SYNC1} + t_{SYNC2}$ 的最小值是 4 x t_{I2C_CLK} = 500 ns. 基于例子 $t_{SYNC1} + t_{SYNC2}$ = 750 ns
4. $t_{SYNC1} + t_{SYNC2}$ 的最小值是 4 x t_{I2C_CLK} = 500 ns. 基于例子 $t_{SYNC1} + t_{SYNC2}$ = 655 ns

表 72. fI2C_CLK = 16 MHz 的时序设置的例子

Parameter	Standard mode		Fast Mode	Fast Mode Plus
	10 kHz	100 kHz	400 kHz	1000 kHz
PRESC	3	3	1	0
SCLL	0xC7	0x13	0x9	0x4
t_{SCLL}	200 x 250 ns = 50 µs	20 x 250 ns = 5.0 µs	10 x 125 ns = 1250 ns	5 x 62.5 ns = 312.5 ns
SCLH	0xC3	0xF	0x3	0x2
t_{SCLH}	196 x 250 ns = 49 µs	16 x 250 ns = 4.0 µs	4 x 125ns = 500 ns	3 x 62.5 ns = 187.5 ns
t_{SCL}(1)	~100 µs(2)	~10 µs(2)	~2500 ns(3)	~1000 ns(4)
SDADEL	0x2	0x2	0x2	0x0
t_{SDADEL}	2 x 250 ns = 500 ns	2 x 250 ns = 500 ns	2 x 125 ns = 250 ns	0 ns
SCLDEL	0x4	0x4	0x3	0x2
t_{SCLDEL}	5 x 250 ns = 1250 ns	5 x 250 ns = 1250 ns	4 x 125 ns = 500 ns	3 x 62.5 ns = 187.5 ns

1. 由于 SCL 内部检测带来的延迟，SCL 周期 t_{SCL} 要大于 $t_{SCLL} + t_{SCLH}$. 提供的 t_{SCL} 值只是举例 .
2. $t_{SYNC1} + t_{SYNC2}$ 的最小值是 4 x t_{I2C_CLK} = 250 ns. 基于例子 $t_{SYNC1} + t_{SYNC2}$ = 1000 ns
3. $t_{SYNC1} + t_{SYNC2}$ 的最小值是 4 x t_{I2C_CLK} = 250 ns. 基于例子 $t_{SYNC1} + t_{SYNC2}$ = 750 ns
4. $t_{SYNC1} + t_{SYNC2}$ 的最小值是 4 x t_{I2C_CLK} = 250 ns. 基于例子 $t_{SYNC1} + t_{SYNC2}$ = 500 ns

表 73. fI2C_CLK = 48 MHz 的时序设置的例子

Parameter	Standard mode		Fast Mode	Fast Mode Plus
	10 kHz	100 kHz	400 kHz	1000 kHz
PRESC	0xB	0xB	5	5
SCLL	0xC7	0x13	0x9	0x3
t_{SCLL}	200 x 250 ns = 50 µs	20 x 250 ns = 5.0 µs	10 x 125 ns = 1250 ns	4 x 125 ns = 500 ns
SCLH	0xC3	0xF	0x3	0x1
t_{SCLH}	196 x 250 ns = 49 µs	16 x 250 ns = 4.0 µs	4 x 125 ns = 500 ns	2 x 125 ns = 250 ns
t_{SCL}(1)	~100 µs(2)	~10 µs(2)	~2500 ns(3)	~875 ns(4)
SDADEL	0x2	0x2	0x3	0x0
t_{SDADEL}	2 x 250 ns = 500 ns	2 x 250 ns = 500 ns	3 x 125 ns = 375 ns	0 ns
SCLDEL	0x4	0x4	0x3	0x1
t_{SCLDEL}	5 x 250 ns = 1250 ns	5 x 250 ns = 1250 ns	4 x 125 ns = 500 ns	2 x 125 ns = 250 ns

1. 由于SCL内部检测带来的延迟，SCL 周期 tSCL 要大于 $t_{SCLL} + t_{SCLH}$. 提供的 t_{SCL} 值只是举例.
2. $t_{SYNC1} + t_{SYNC2}$ 的最小值是 4 x t_{I2C_CLK} = 83.3 ns. 基于例子 $t_{SYNC1} + t_{SYNC2}$ = 1000 ns
3. $t_{SYNC1} + t_{SYNC2}$ 的最小值是 4 x t_{I2C_CLK} = 83.3ns. 基于例子 $t_{SYNC1} + t_{SYNC2}$ = 750 ns
4. $t_{SYNC1} + t_{SYNC2}$ 的最小值是 4 x t_{I2C_CLK} = 83.3ns. 基于例子 $t_{SYNC1} + t_{SYNC2}$ = 250 ns

23.4.11 SMBus 特定功能

本节只针对支持 SMBus 功能的部分。 请参见表 23.3： I2C 具体功能配备

简介

系统管理总线（SMBus）是一个种两线接口，通过它可以与各种设备和系统的其余部分互相通讯。 它基于 I2C 操作原理。 SMBus 提供了一种针对系统和电源管理相关任务的控制总线。

该外设兼容的 SMBus 规范 2.0 版（http://smbus.org/specs/）。

系统管理总线规范是指三种类型的设备。

- 从机是指接收或相应命令的设备。
- 主机是下达命令，产生时钟和终止传输的设备。
- HOST 是一个特殊的主机，它向系统 CPU 提供主接口。 HOST 必须具备主机和从机的双重功能，并且必须支持 SMBus HOST 通知协议。 一个系统中只可以有一个 HOST。

本外设可以配置为主机或从机，当然也可以作为 HOST。 SMBus 是基于 I2C 规范修订版 2.1。

总线协议

对于任何给定的设备有 11 个可能的命令协议。 一个设备可以使用任何或全部的 11 个协议进行通信。 协议是快速命令，发送字节，接收字节，写字节，写字，读字节，读字，过程调用，块读，块写和块写块读过程调用。 这些协议是由用户软件实现的。

对于这些协议的更多细节，请参考 SMBus 规范 版本 . 2.0 (http://smbus.org/specs/).

地址解析协议 (ARP)

SMBus 从机地址冲突的问题可以通过给每个从设备动态标定一个新的独特的地址的方式解决。为了分配地址，需要一种区分每个设备的机制，每个设备必须拥有一个唯一的设备标识符。这个 128 位的数字是由软件实现的。

本外设支持地址解析协议（ARP）。 将 I2Cx_CR1 寄存器的 SMBDEN 位置 1，会启用 SMBus 设备的默认地址（0b1100 001）。 ARP 命令由用户软件实现。

ARP 支持的仲裁动作也是在从机模式下完成。

SMBus 地址解析协议的更多细节，请参考 SMBus 规范 版本 2.0 (http://smbus.org/specs/).

命令的接收和数据应答的控制

一个 SMBus 的接收器必须能够对每个收到的命令或数据回应 NACK。 为了允许从机模式下的 ACK 控制，必须将 I2Cx_CR1 寄存器的 SBC 位置 1，以启用从机字节控制模式。 更多的细节参见第 509 页 从机字节控制模式。

HOST 通知协议

设置 I2Cx_CR1 寄存器的 SMBHEN 位，使得本外设支持 HOST 通知协议。 在这种情况下，HOST 会应答 SMBus 主机地址（0b0001000）。

使用此协议时，本设备会作为主机，而 HOST 则会作为一个从机。

SMBus 报警

本外设可选 SMBus 提醒信号支持。 一个仅作为从机的设备在想要发起通讯的时候，可以通过 SMBALERT 引脚通知 HOST。 HOST 会处理这个中断，并且随即通过提醒响应地址 (0b0001100) 来访问全部的 SMBALERT 设备。 只有将 SMBALERT 引脚拉低的设备会回应提醒响应地址。

当被配置为从机设备（SMBHEN = 0）时，将 I2Cx_CR1 寄存器的 ALERTEN 位置 1，会导致 SMBA 引脚被拉低。 提醒响应地址则会同时启用。

当被配置为 HOST 设备（SMBHEN = 1）时，只要发现 ALERTEN=1，并且在 SMBA 引脚上检测到一个下降沿，I2Cx_ISR 寄存器中的提醒标志就会被置位。 如果 I2Cx_CR1 寄存器中的 ERRIE 位为 1，则会产生中断。 当 ALERTEN = 0 时，即使是 SMBA 外部引脚为低，提醒线也会被认为是高。

如果不需要使用 SMBus 提醒引脚，只要是果 ALERTEN = 0，SMBA 脚就可以被用来作为一个标准的 GPIO。

包错误检查

SMBus 规范中介绍到的包错误检查机制可以提高通信可靠性。 包错误检查是通过在每个消息后面附带一个包错误检查码来实现的。 PEC 码通过对全部的消息字节（包括地址和读写位）使用多项式 C(x) = x8 + x2 + x + 1 CRC-8 计算得来。

本外设内嵌硬件 PEC 计算单元，在接收数据的时候如果发现 PEC 结果不匹配，会允许自动发送一个 NACK。

超时

本外设内嵌一个硬件定时器，以便和 SMBus 规范 V2.0 中定义的 3 个超时相符。

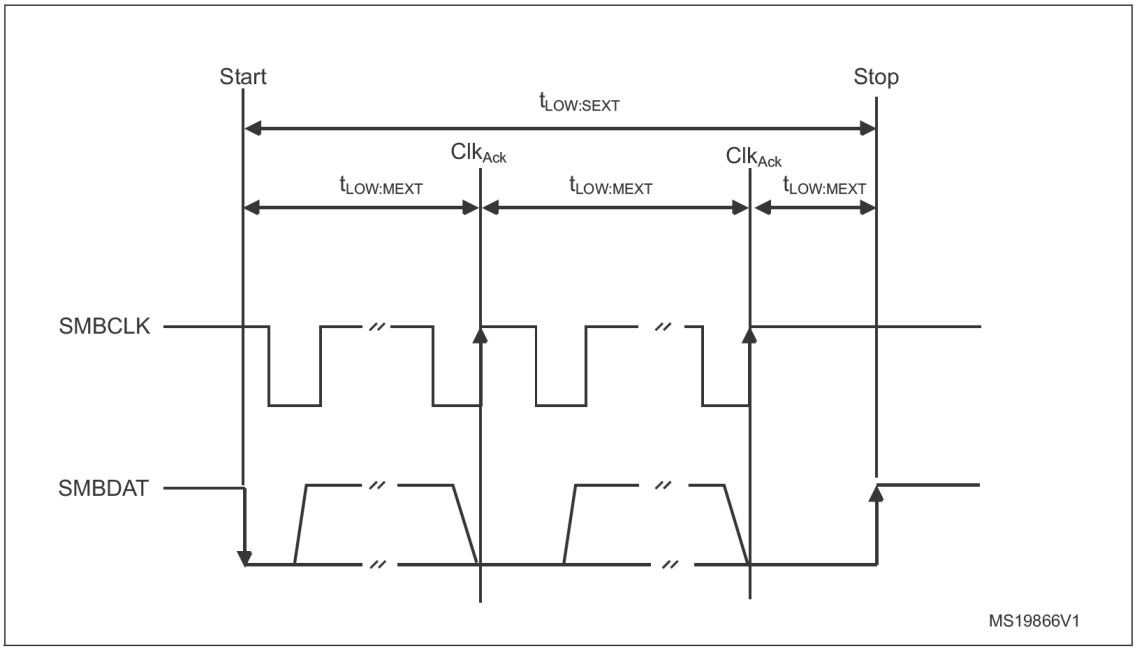
表 74. SMBus 超时规范

Symbol	Parameter	Limits		Unit
		Min	Max	
$t_{TIMEOUT}$	Detect clock low timeout	25	35	ms
$t_{LOW:SEXT}$(1)	Cumulative clock low extend time (slave device)		25	ms
$t_{LOW:MEXT}$(2)	Cumulative clock low extend time (master device)		10	ms

1. $t_{LOW:SEXT}$ 是一个给定的从机设备从 START 到 STOP 之间可以延长的时钟周期的累计。 另一个从机设备或主机也可能对时钟进行占用从而导致总的时钟低的占用时间大于 $t_{LOW:SEXT}$。 因此，这个参数的测量条件是从机作为一个全速的主机的唯一通讯目标。

2. $t_{LOW:MEXT}$ 是一个主机设备按照 START 到 ACK，ACK 到 ACK 或者 ACK 到 STOP 的方式发送一个字节所允许的时钟周期的累计。 另一个从机设备或主机也可能对时钟进行占用从而导致总的时钟低的占用时间大于 $t_{LOW:MEXT}$。 因此，这个参数的测量条件是有只一个全速的从机作为唯一的通讯目标。

图 218. $t_{LOW:SEXT}$, $t_{LOW:MEXT}$ 超时间隔



总线空闲检测

如果主机发现时钟和数据信号保持为高的时间 t_{IDLE} 大于 greater than $t_{HIGH,MAX}$，就可以认为目前总线处于空闲状态。 （参阅表 70： I2C SMBus 规范的时钟时序）

这个时序参数覆盖了那种主机被动态的加入到总线上，不一定检测到了 SMBCLK 或者 SMBDAT 上的状态转换的条件。 在这种情况下，主机必须等待足够长的时间，以确保传输是否是正在进行中。 本外设支持硬件总线空闲检测。

23.4.12 SMBus 初始化

本节只针对支持 SMBus 功能的部分。 请参见章节 23.3： I2C 具体功能配备

为了正确执行 SMBus 通讯，在 I2C 初始化的基础上还有一些其它的初始化动作要做：

命令的接收和数据应答控制

一个 SMBus 的接收器必须能够对每个收到的命令或数据回应 NACK。 为了允许从机模式下的 ACK 控制，必须将 I2Cx_CR1 寄存器的 SBC 位置 1，以启用从机字节控制模式。 更多的细节参见第 509 页 从机字节控制模式。

特定地址（ 从机模式）

如有必要，需要启用特定的 SMBus 地址。 有关详细信息，请参阅第 532 页的总线空闲检测。

- 将 I2Cx_CR1 寄存器的 SMBDEN 位置 1， 会启用 SMBus 设备的默认地址（0b1100 001）。
- 将 I2Cx_CR1 寄存器的 SMBHEN 位置 1 会启用 SMBus 主机地址（0b0001 000）。
- I2Cx_CR1 寄存器的 ALERTEN 位被置 1 时，会启用通知响应地址。

包错误检查

将 I2Cx_CR1 寄存器的 PECEN 位置 1，会启用的 PEC 计算这时的 PEC 传送由硬件计数器位（I2Cx_CR2 寄存器中的 nBytes 个 [7:0]）来配合管理。PECEN 必须在使能 I2C 之前设置好。

既然 PEC 传输是有硬件字节计数器管理的，所以在 SMBus 工作于从机模式时，SBC 位必须先置 1。 PEC 数据会在传送了 NBYTES-1 个数据被传送完毕之后发送，当然这时 PECBYTE 位要求是 1，并且 RELOAD 位要求是 0. 如果 RELOAD 位是 1，那 PECBYTE 位就没有作用了。

警告： 当 I^2C 启用时不允许更改 PECEN 的配置。

表 75. SMBus 带 PEC 的的配置表

Mode	SBC bit	RELOAD bit	AUTOEND bit	PECBYTE bit
Master Tx/Rx NBYTES + PEC+ STOP	x	0	1	1
Master Tx/Rx NBYTES + PEC + ReSTART	x	0	0	1
Slave Tx/Rx with PEC	1	0	x	1

超时检测

启用超时检测功能，须将 I2Cx_TIMEOUTR 寄存器的 TIMOUTEN 位和 TEXTEN 位置 1。 定时器须用某种方式编程，以实现在 SMBus 规范 2.0 给出的最大时间之前检测到超时。 2.0.

- $t_{TIMEOUT}$ 检查

要启用 tTIMEOUT 检查 , 先要将要检查的 $t_{TIMEOUT}$ 参数对应的定时器重载值写入到 12 位的 TIMEOUTA[11:0]。要检查 SCL 低电平超时，还需要保证 TIDLE 位为 0。
在 I2Cx_TIMEOUTR 寄存器的 TIMOUTEN 被置 1 使能了定时器后。 如果 SCL 被拉低的时间大于 (TIMEOUTA+1) x 2048 x t_{I2C_CLK}, I2Cx_ISR 寄存器中的 TIMEOUT 标志会被置 1 。
请参阅表 76： 各种 I2C_CLK 频率下 TIMEOUTA 设置举例

警告： 在 TIMEOUTEN 位为 1 的时候不允许改变 TIMEOUTA [11:0] 位域和 TIDLE 位的配置。

- $t_{LOW:SEXT}$ 和 $t_{LOW:MEXT}$ 检查

必须配置 12 位定时器 TIMEOUTB，从机模式会检查 $t_{LOW:SEXT}$，而主机模式会检查 $t_{LOW:MEXT}$。标准只规定了最大值，可选择与之相同的值。
在 I2Cx_TIMEOUTR 寄存器的 TIMOUTEN 被置 1 使能了定时器后。 如果 SMBus 外设将 SCL 拉低的时间达到了 (TIMEOUTB+1) x 2048 x t_{I2C_CLK}, 并且又没有达到 532 页总线空闲检查中描述的时间，I2Cx_ISR 寄存器中的 TIMEOUT 标志会被置 1 。
请参阅表 77： 各种 I2C_CLK 频率下 TIMEOUTB 设置举例

警告： 当 TEXTEN 位为 1 时，不允许改变 TIMEOUTB 的配置。

总线空闲检测

要使能 t_{IDLE} 检查 , 必须根据要实现的 t_{IDLE} 参数向 TIMEOUTA[11:0] 位域写入预装载值。要同时检测 SCL 和 SDA 上的高电平超时，TIDLE 须写为 1。
将 I2Cx_TIMEOUTR 寄存器中的 TIMOUTEN 位置 1，使能了定时器后。
如果 SCL 和 SDA 线同时保持高的时间大于 (TIMEOUTA+1) x 4 x t_{I2C_CLK}，I2Cx_ISR 寄存器中的 TIMEOUT 标志会被置 1。
请参阅表 78： 各种 I2C_CLK 频率下 TIMEOUTA 设置举例

警告： 在 TIMEOUTEN 位为 1 的时候不允许改变 TIMEOUTA [11:0] 位域和 TIDLE 位的配置。

23.4.13 SMBus： I2Cx_TIMINGR 寄存器配置举例

本节只针对支持 SMBus 功能的部分。 请参见章节 23.3： I2C 具体功能配备

- 配置 TIMEOUT 最长到 25 毫秒：

表 76. 各种 I2C_CLK 频率下 TIMEOUTA 设置举例

f_{I2C_CLK}	TIMEOUTA[11:0] bits	TIDLE bit	TIMEOUTEN bit	$t_{TIMEOUT}$
8 MHz	0x61	0	1	98 x 2048 x 125ns = 25 ms
16 MHz	0xC3	0	1	196 x 2048 x 62.5ns = 25 ms
48 MHz	0x249	0	1	586 x 2048 x 20.08ns = 25 ms

- $t_{LOW:SEXT}$ 和 $t_{LOW:MEXT}$ 最长为 8 ms：

表 77. 各种 I2C_CLK 频率下 TIMEOUTB 设置举例

f_{I2C_CLK}	TIMEOUTB[11:0] bits	TEXTEN bit	$t_{LOW:EXT}$
8 MHz	0x1F	1	32 x 2048 x 125 ns = 8 ms
16 MHz	0x3F	1	64 x 2048 x 62.5 ns = 8 ms
48 MHz	0xBB	1	188 x 2048 x 20.08 ns = 8 ms

- t_{IDLE} 最长为 50 µs

表 78. 各种 I2C_CLK 频率下 TIMEOUTA 设置举例

f_{I2C_CLK}	TIMEOUTA[11:0] bits	TIDLE bit	TIMEOUTEN bit	t_{TIDLE}
8 MHz	0x63	1	1	100 x 4 x 125 ns = 50 µs
16 MHz	0xC7	1	1	200 x 4 x 62.5 ns = 50 µs
48 MHz	0x257	1	1	600 x 4 x 20.08 ns = 50 µs

23.4.14 SMBus 从机模式

本节只针对支持 SMBus 功能的部分。 请参见章节 23.3： I2C 具体功能配备

除了 I2C 的从机传输管理（参见第 23.4.8 I2C 从机模式）还需要执行一些额外的软件流程以支持 SMBus。

SMBus 从机发送器

在 SMBus 模式下，SBC 必须为 1，以允许在发送了足够数量的字节之后跟上 PEC 字节。 当 PECBYTE 位被置 1，NBYTES[7:0] 中编程的字节数中要包括 PEC 字节。 在这种情况下 TXIS 中断总数将为 NBYTES-1 个，如果主机在 NBYTES-1 次传送之后还要求更多的字节，I2Cx_PECR 寄存器的内容就会自动被发送出去。

警告： RELOAD 位被置 1 时，PECBYTE 位没有作用。

图 219. SMBus 从机发送 N 字节 +PEC 时的传输顺序流程图

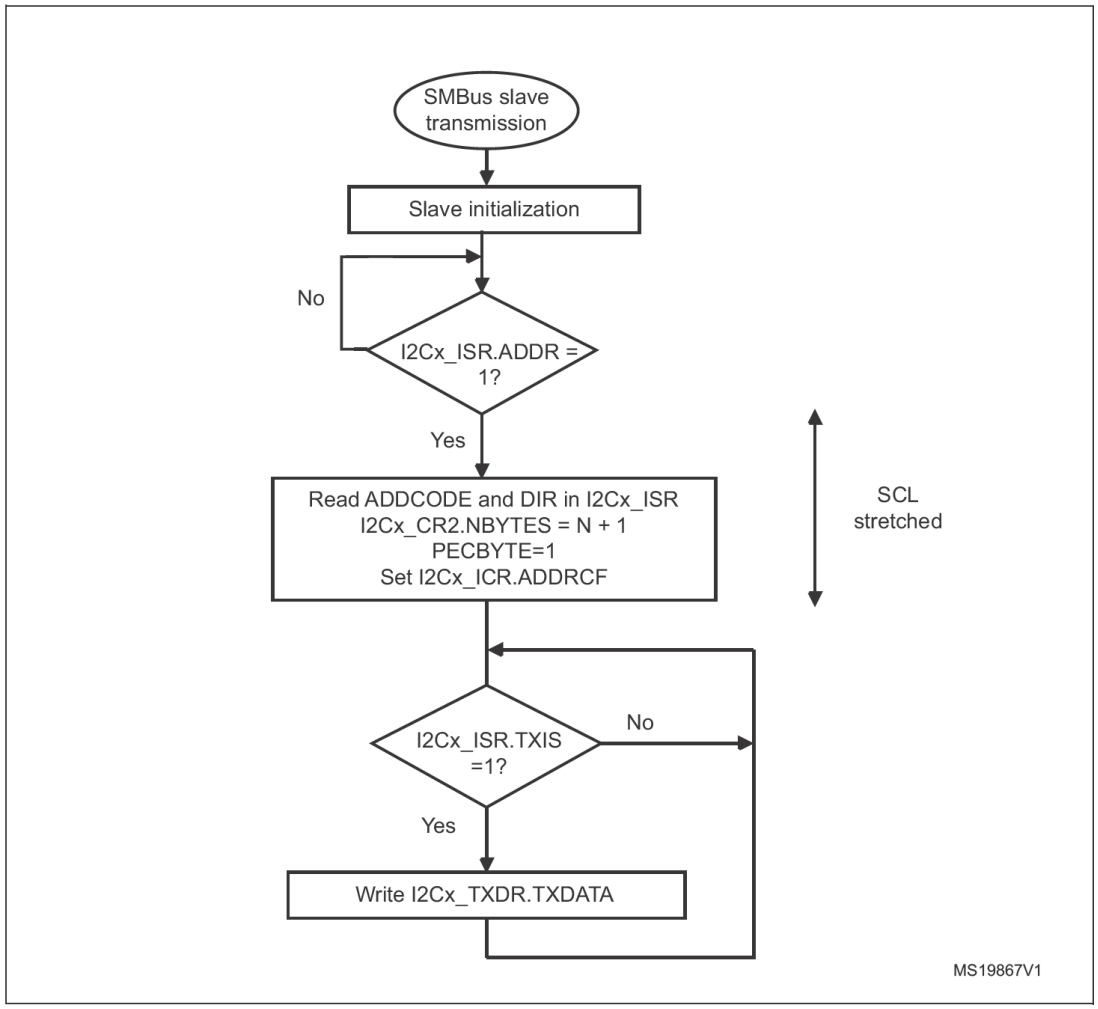
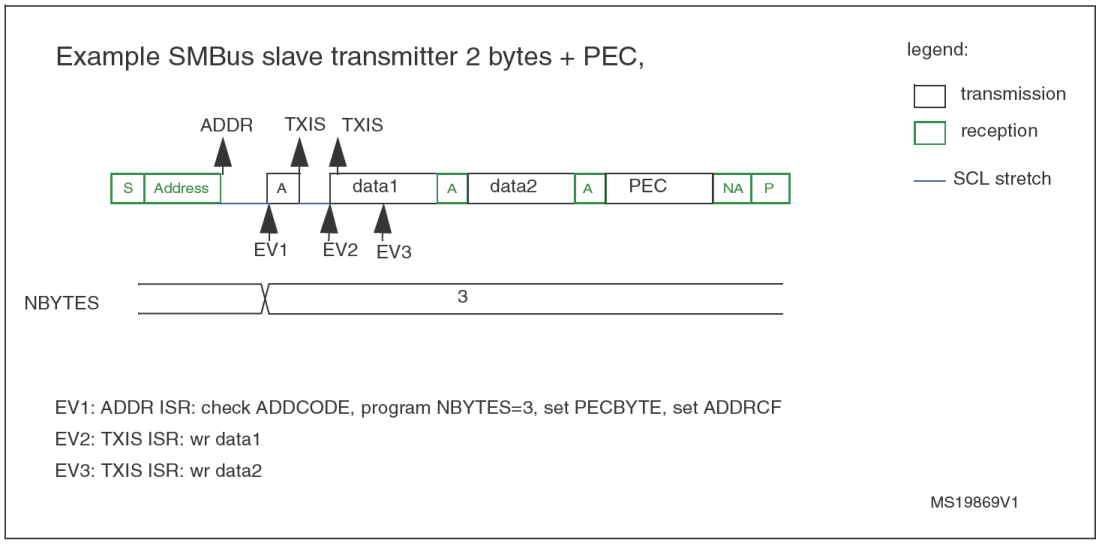


图 220. SMBus 从机发送 (SBC=1) 传输时序图



SMBus 从机接收

在工作于 SMBus 模式时，SBC 必须为 1，以允许在发送了足够数量的字节之后跟上 PEC 字节。 为了实现每个字节的 ACK 控制，必须选择重载模式（RELOAD = 1）。 更多的细节参见第 509 页 从机字节控制模式。

为了检查 PEC 字节，必须清除 RELOAD 位，而且必须置 PECBYTE 位为 1。 在这种情况下，已收到 NBYTES-1 个数据后，下一个收到的字节会与内部 I2Cx_PECR 寄存器的内容进行比较。如果比较不匹配，自动生成一个 NACK，如果比较匹配，自动生成一个 ACK，这时 ACK 位的值是不是 1 都没所谓。 PEC 字节一旦被收到，它像任何其他数据样被复制到 I2Cx_RXDR 寄存器，并且 RXNE 标志被置 1。

在 PEC 不匹配的情况下，PECERR 标志被置 1，如果 I2Cx_CR1 寄存器的 ERRIE 位为 1，会产生一个中断。

如果 ACK 由软件控制，可以令 PECBYTE = 1，并在连续传送中使用同样的写操作，将 NBYTES 的值写成实际要接收的值。 在 NBYTES-1 个字节收到后，下一个收到的字节被当做 PEC 来检查。

警告： RELOAD 位被置 1 时，PECBYTE 位没有作用。

图 221. SMBus 从机接收 N 字节 +PEC 时的传输顺序流程图

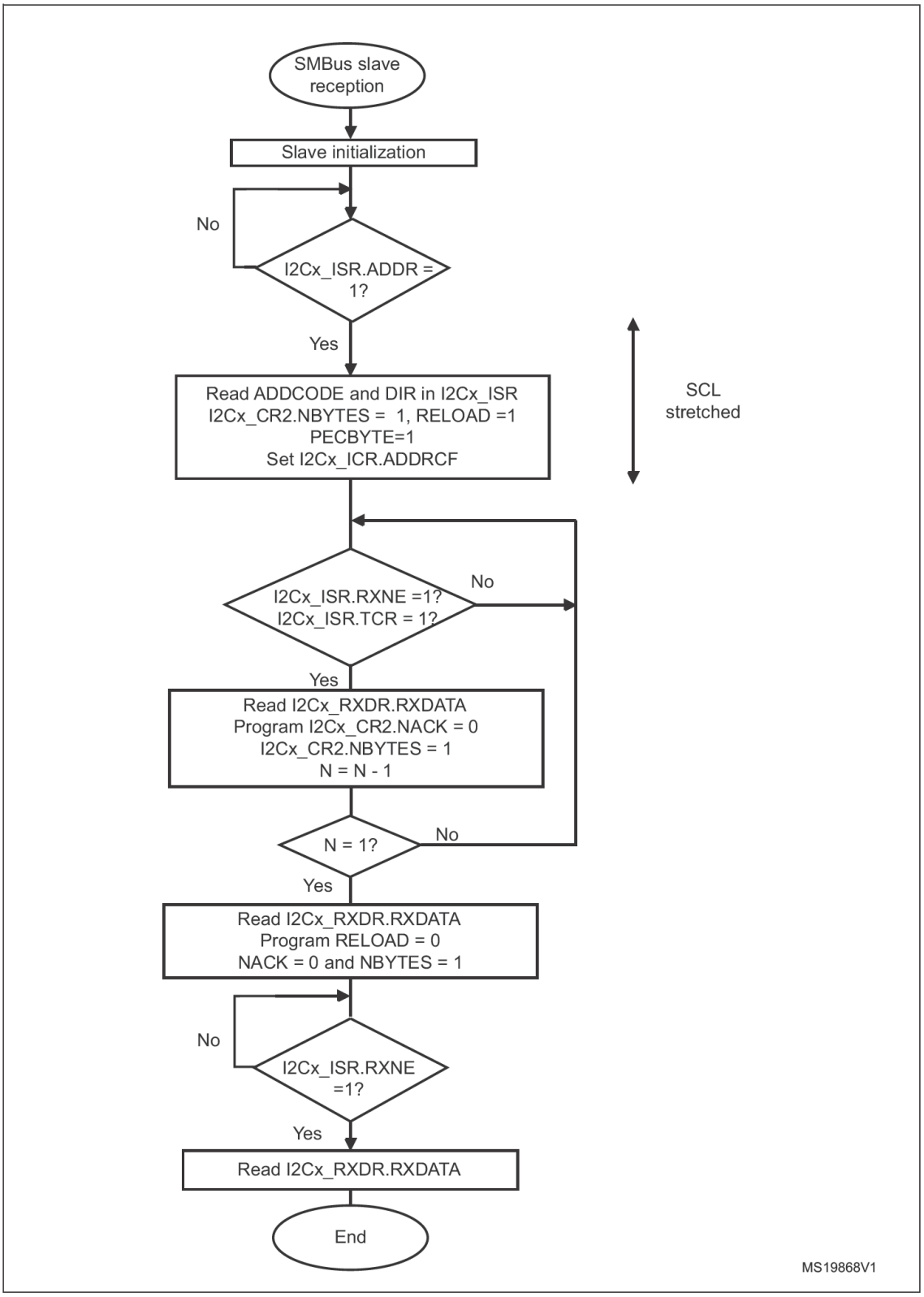
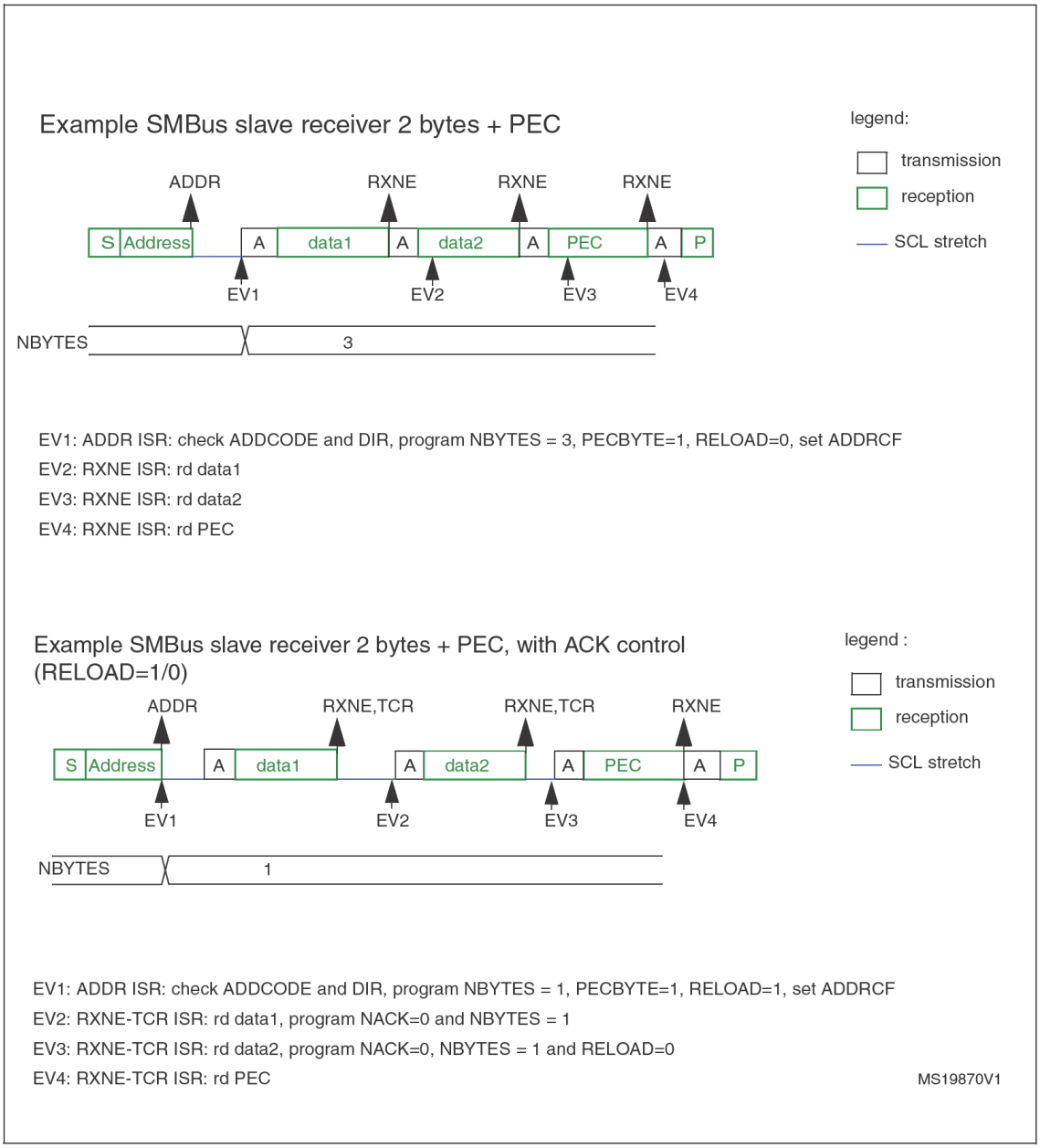


图 222. SMBus 从机接收 (SBC=1) 传输时序图



本节只针对支持 SMBus 功能的部分。 请参见章节 23.3： I2C 具体功能配备

除了 I2C 的从机传输管理（参见第 23.4.9 I2C 主机模式）还需要执行一些额外的软件流程以支持 SMBus。

SMBus 主机发送

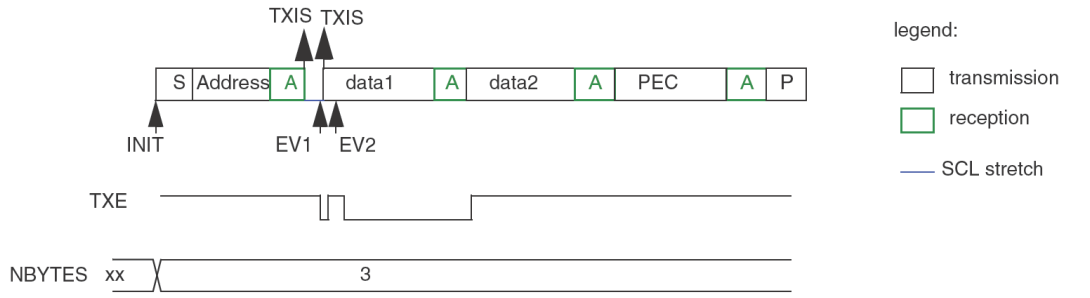
当 SMBus 主机要发送 PEC 时，在将 START 置 1 之前，PECBYTE 位必须置 1，并且要发送的字节数必须写到 NBYTES[7:0] 位域中。 在这种情况下将 TXIS 中断总数为 NBYTES-1。 所以，如果在 NBYTES=0x1 的时候将 PECBYTE 位置 1，I2Cx_PECR 寄存器的内容就会被自动发送出去。

如果 SMBus 主机要在 PEC 数据之后自动发送一个 STOP 条件，则应该选择自动结束方式（AUTOEND = 1）。 在这种情况下，PEC 数据发送过后会自动跟上一个 STOP 条件。

当 SMBus 主机要在 PEC 传输之后发送 RESTART 条件，则必须选择软件结束模式（AUTOEND = 0）。在这种情况下，一旦 NBYTES-1 个字节被传送完，I2Cx_PECR 寄存器的内容会被发送，PEC 发送完后，TC 标志会被置 1，并且 SCL 线会被拉低。 RESTART 条件必须在 TC 中断服务程序中编程实现。

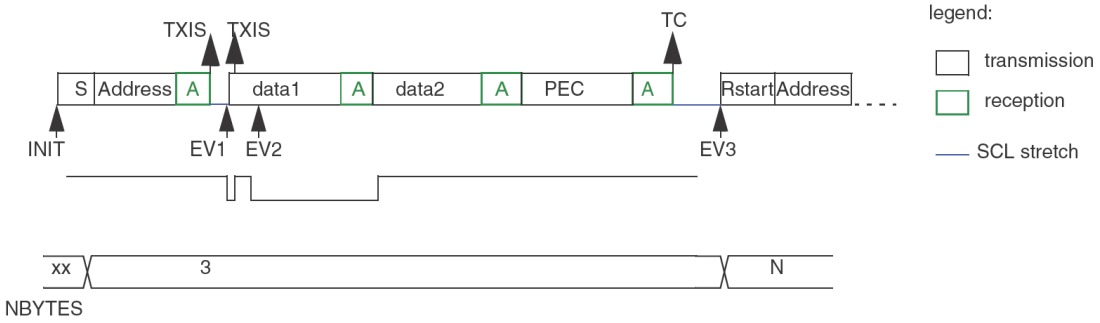
警告： RELOAD 位被置 1 时，PECBYTE 位没有作用。

Example SMBus master transmitter 2 bytes + PEC, automatic end mode (STOP)



INIT: program Slave address, program NBYTES = 3, AUTOEND=1, set PECBYTE, set START
EV1: TXIS ISR: wr data1
EV2: TXIS ISR: wr data2

Example SMBus master transmitter 2 bytes + PEC, software end mode (RESTART)



INIT: program Slave address, program NBYTES = 3, AUTOEND=0, set PECBYTE, set START
EV1: TXIS ISR: wr data1
EV2: TXIS ISR: wr data2
EV3: TC ISR: program Slave address, program NBYTES = N, set START

MS19871V1

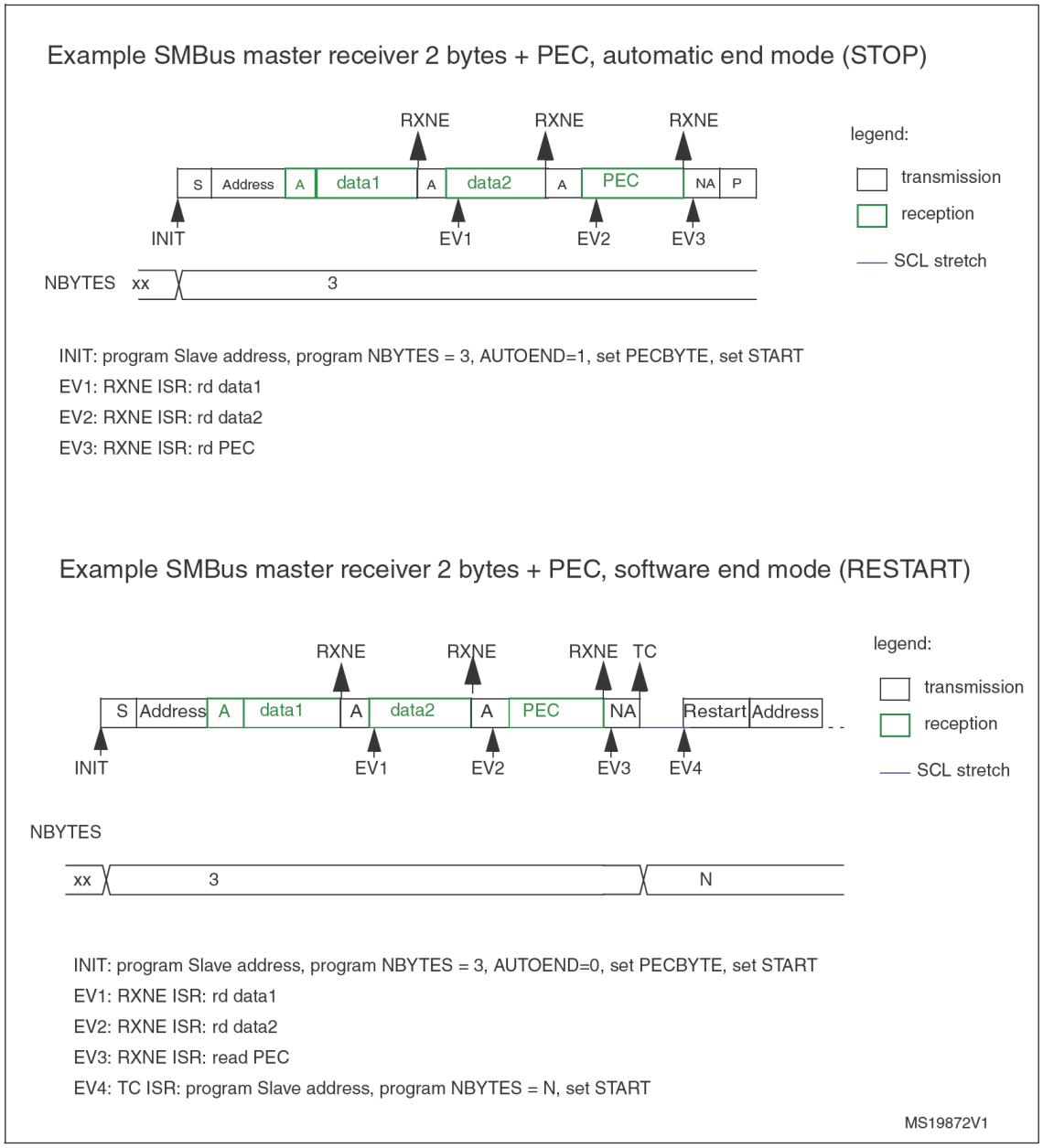
SMBus 主机接收

当 SMBus 主机要在传送末尾接收 PEC 之后自动发出停止条件，须选择自动结束模式（AUTOEND = 1）。 PECBYTE 位必须先置 1，以及从机地址也要在置 START 位之前预先设置。 在这种情况下，已收到 NBYTES-1 个数据后，下一个收到的字节会与内部 I2Cx_PECR 寄存器的内容进行比较。 对 PEC 字节会给出一个 NACK 响应后面再跟着一个 STOP 条件。

当 SMBus 主机要在传送末尾接收 PEC 之后发出 RESTART 条件，须选择软件结束模式（AUTOEND = 0）。PECBYTE 位必须先置 1，以及从机地址也要在置 START 位之前预先设置。在这种情况下，已收到 NBYTES-1 个数据后，下一个收到的字节会与内部 I2Cx_PECR 寄存器的内容进行比较。 在收到 PEC 字节后，TC 标志被置位，同时 SCL 线被拉低。 RESTART 条件可以在 TC 中断服务程序中编程实现。

警告： RELOAD 位被置 1 时，PECBYTE 位没有作用。

图 224. SMBus 主机接收传输时序图



23.4.15 根据地址匹配事件从 STOP 模式唤醒

本节只针对具备从 Stop 模式唤醒功能的情形。 请参见第 23.3 节： I2C 具体功能配备

I2C 能够在被寻址到的时候将 MCU 从 STOP 模式唤醒（APB 时钟关闭时）。支持所有的地址模式。

将 I2Cx_CR1 寄存器的 WUPEN 位置 1，会启用从 STOP 模式唤醒功能。 要从停止模式唤醒，必须选择 HSI 振荡器作为为 I2C_CLK 的时钟源。

在 STOP 模式下，HSI 是关闭的。 当检测到起始条件，I2C 接口将启动 HSI，然后将 SCL 拉低，直到 HSI 启动好。

HSI 随后用于地址接收。

在地址匹配的情况下，在 MCU 唤醒期间 I2C 将 SCL 持续拉低。 当软件清除了 ADDR 标志后，SCL 被释放，传输进入正常状态。

如果地址不匹配，HSI 会再次关闭，MCU 也不会被唤醒。

注： *1 如果 I2C 时钟是系统时钟，或者如果 WUPEN = 0，HSI 振荡器在收到 START 后不会被打开。*
2 只有 ADDR 中断可以唤醒 MCU。 所以，在 I2C 作为主机正在传输数据的时候，或者作为从机在被寻址到（ADDR=1）的时候，不要进入 STOP 模式。 这可以用在 ADDR 中断服务程序中清除 SLEEPDEEP 位，并只在 STOPF 标志置 1 后重新将它置 1 的方式来管理。

警告： 数字滤波器和从 STOP 模式唤醒功能不兼容。 如果 DNF 位不等于 0，设置 WUPEN 位会没有任何效果。

警告： 只有当 I2C 的时钟源是 HSI 振荡器的时候此功能可用。

警告： 必须启用时钟延长功能（NOSTRETCH = 0），以确保正常从停止功能唤醒。

23.4.16 错误条件

以下是可能导致通信失败的错误条件。

总线错误 (BERR)

如果在 9 倍 SCL 时钟脉冲以外检测到一个 START 或者 STOP 条件，会发生一个总线错误。当 SCL 为高的时候在 SDA 上出现上升沿或者下降沿，会检测到 START 或 STOP 条件。

只有在 I2C 处于数据传输状态下（作为主机在发送或作为从机被寻址到）才会有总线错误标志置位，例如在从机模式下的寻址阶段就不会。

对于从机模式下，一个错位的 START 或者 RESTART 条件会被从机当成正常的 START 条件来处理。

当检测到总线错误，I2Cx_ISR 寄存器中的 BEER 标志会被硬件置 1，如果 I2Cx_CR1 寄存器中的 ERRIE 位为 1，则会产生中断。

仲裁丢失（ ARLO）

在往 SDA 线上发高电平，但在 SCL 的上升沿却从 SDA 采样得到低电平时，就会检测到一次仲裁丢失错误。

- 在主机模式下，仲裁丢失在地址阶段、数据阶段以及数据确认阶段进行检测。 在这种情况下，SDA 和 SCL 线被释放，START 控制位由硬件清零，主机自动切换到从机模式。
- 在从机模式下，仲裁丢失在数据阶段和数据确认阶段进行检测。 在这种情况下，传输被终止，SCL 和 SDA 线被释放。

当检测到仲裁丢失错误，I2Cx_ISR 寄存器中的 ARLO 标志会被硬件置 1，如果 I2Cx_CR1 寄存器中的 ERRIE 位为 1，则会产生中断。

溢出 / 欠载错误(OVR)

如果 NOSTRETCH=1，在从机模式下只要下列条件发生，就会检测到欠载或溢出错误：

- 在接收时，尚未读取 RXDR 寄存器的时候又有一个新的字节接收到。
  新收到的字节会丢失，并且会有一个 NACK 自动发送出去，当作是对这个新的字节的响应。
- 在发送时：
  - 当应该发送第一个数据字节时却有 STOPF=1。 如果 TXE=0，那么 I2Cx_TXDR 寄存器的值会发送出去，如果不是 0，那么就会发送 0xFF。
  - 在一个新的字节应该被写入到 I2Cx_TXDR 寄存器，但却没有写，那就会将 0xFF 发送出去。

当检测到欠载 / 溢出错误，I2Cx_ISR 寄存器中的 OVR 标志会被硬件置 1，如果 I2Cx_CR1 寄存器中的 ERRIE 位为 1，则会产生中断。

包错误检查错误(PECERR)

本节只针对支持 SMBus 功能的部分。 请参见第 23.3 节： I2C 具体功能配备

在收到的 PEC 字节与 I2Cx_PECR 寄存器的内容不匹配时会检测到 PEC 错误。 错误的 PEC 接收后，会自动发送一个 NACK。

当检测到 PEC 错误，I2Cx_ISR 寄存器中的 PECERR 标志会被硬件置 1，如果 I2Cx_CR1 寄存器中的 ERRIE 位为 1，则会产生中断。

超时错误(超时)

本节只针对支持 SMBus 功能的部分。 请参见第 23.3 节： I2C 具体功能配备

这些条件中的任何一个都会导致超时错误发生：

- TIDLE = 0 并且 SCL 保持低的时间达到了在 TIMEOUTA [11:0] 位域所定义的时间： 这是用来检测 SMBus 超时。
- TIDLE = 1，SDA 和 SCL 都保持高电平并达到了在 TIMEOUTA [11:0] 位域所定义的时间：这是用来检测总线空闲状态。
- 主机时钟低延长累计时间达到了在 TIMEOUTB [11:0] 位域规定的时间（SMBus 的 $t_{LOW:MEXT}$ 参数）
- 从机时钟低延长累计时间达到了在 TIMEOUTB [11:0] 位域规定的时间（SMBus 的 $t_{LOW:SEXT}$ 参数）

当在主机模式下检测到超时，会自动发送一个 STOP 条件。
当在从机模式下检测到超时，SDA 线和 SCL 线会自动释放。
当检测到超时错误，I2Cx_ISR 寄存器中的 TIMEOUT 标志会被硬件置 1，如果 I2Cx_CR1 寄存器中的 ERRIE 位为 1，则会产生中断。

通知(ALERT)

本节只针对支持 SMBus 功能的部分。 请参见第 23.3 节： I2C 具体功能配备

当 I2C 接口配置为 HOST（SMBHEN = 1），SMBA 引脚检测功能被启用（ALERTEN = 1），并且在 SMBA 脚上检测到一个下降沿时，ALERT 标志会被置 1。 如果 I2Cx_CR1 寄存器中的 ERRIE 位为 1，则会产生中断。

23.4.17 DMA 请求

利用 DMA 发送

通过设置在 I2Cx_CR1 寄存器 TXDMAEN 位，可以启用 DMA（直接存储器存取）发送。 数据被预先放到 DMA 外设所设定的 SRAM 区域（参见章节 10： 直接内存访问控制器（DMA）140 页）发往 I2Cx_TXDR 寄存器，而不管 TXIS 位是不是 1。
只使用 DMA 传输数据。

- 在主机模式下： 初始化，从机地址，方向，字节数和起始位都由软件设置（已经发送了的从机地址，不能使用 DMA 传输）。 当所有数据都使用 DMA 传输，必须在设置 START 位之前初始化 DMA。 传输的结束由 NBYTES 计数器来管理。 参见 520 页主机发送器。
- 在从机模式下：
  - NOSTRETCH = 0 时，所有数据都使用 DMA 传输时，DMA 必须在地址匹配事件之前初始化好，或者中断服务程序中，清除 ADDR 标志之前完成这个任务。
  - 当 NOSTRETCH = 1，DMA 必须在地址匹配事件之前初始化好。
- SMBus 支持的实例： PEC 的传送由 NBYTES 计数器管理。

参见第 535 页的 SMBus 从机发送 和第 539 页的 SMBus 主机发送。

注： 如果使用 DMA 发送，TXIE 位则不需要启用。

利用 DMA 接收

通过设置在 I2Cx_CR1 寄存器 RXDMAEN 位，可以启用 DMA（直接存储器存取）接收。 数据会从 I2Cx_RXDR 寄存器取出来，转移到 DMA 外设中指向的 SRAM 区域（参见章节 10：直接内存访问控制器（DMA）140 页）而不管 RXNE 位是不是 1。 只有数据（包括 PEC）被 DMA 传输。

- 在主机模式下，初始化，从机地址，方向，字节数和起始位，都要通过软件设置。 当所有数据都使用 DMA 传输完后，必须在设置 START 位之前初始化 DMA。 传输的结束由 NBYTES 计数器来管理。 参见 524 页主机接收。
- 在从机模式中如果 NOSTRETCH = 0，所有数据都使用 DMA 传输后，DMA 必须在地址匹配事件之前初始化好，或者在中断服务程序中，清除 ADDR 标志之前完成这个任务。
- 如果 SMBus 支持（见 23.3 节： I2C 具体功能配备）： PEC 的传送由 NBYTES 计数器管理。参阅 537 页的 SMBus 从机接收 和 541 页的 SMBus 主机接收 。

注： 如果使用 DMA 发送，RXIE 位则不需要启用。

23.5 I^2C 中断

下表给出了 I^2C 中断请求列表。

表 79. I^2C 中断请求

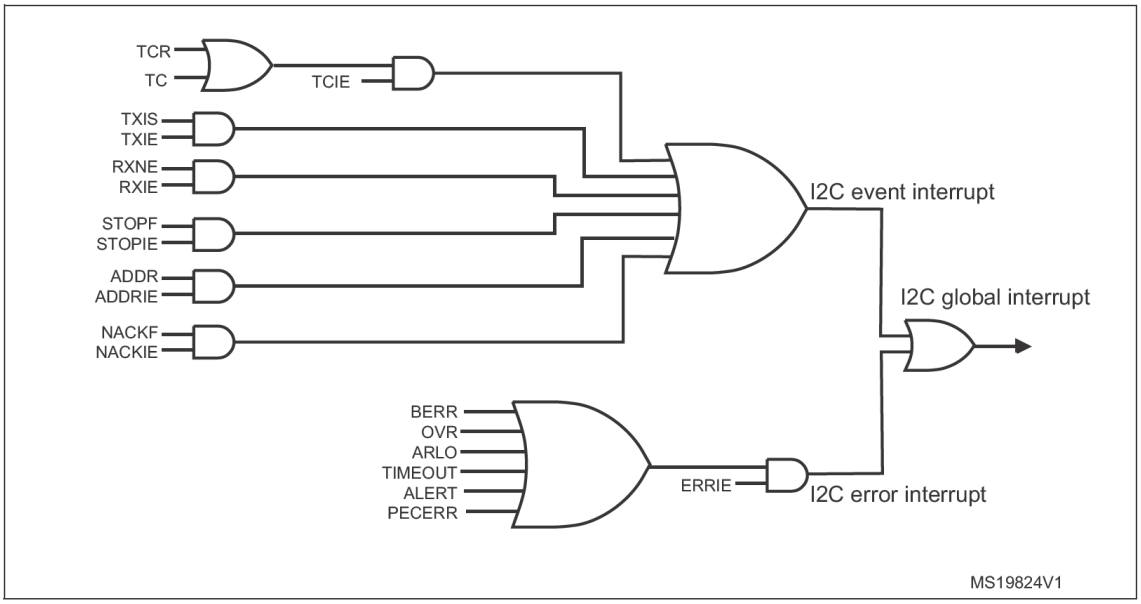
Interrupt event	Event flag	Event flag/Interrupt clearing method	Interrupt enable control bit
Receive buffer not empty	RXNE	Read I2Cx_RXDR register	RXIE
Transmit buffer interrupt status	TXIS	Write I2Cx_TXDR register	TXIE
Stop detection interrupt flag	STOPF	Write STOPCF=1	STOPIE
Transfer Complete Reload	TCR	Write I2Cx_CR2 with NBYTES[7:0] ≠ 0	TCIE
Transfer complete	TC	Write START=1 or STOP=1	
Address matched	ADDR	Write ADDRCF=1	ADDRIE
NACK reception	NACKF	Write NACKCF=1	NACKIE
Bus error	BERR	Write BERRCF=1	ERRIE
Arbitration loss	ARLO	Write ARLOCF=1	
Overrun/Underrun	OVR	Write OVRCF=1	
PEC error	PECERR	Write PECERRCF=1	
Timeout/t_{LOW} error	TIMEOUT	Write TIMEOUTCF=1	
SMBus Alert	ALERT	Write ALERTCF=1	

根据产品的实际功能，所有这些中断事件可以共享同一个中断向量（I^2C 全局中断），或组合成 2 个（I^2C 事件中断和 I2C 错误中断）中断向量。 详情请参阅表 26： 中断向量表 155 页。

要使能 I^2C 中断，有下列顺序要求：
1. 在 NVIC 中配置和启用 I^2C IRQ 通道。
2. 配置 I^2C 以产生中断。

I^2C 唤醒事件连接到的 EXTI 控制器（参见 11.2.6 节： 外部和内部中断 / 事件线映射 第 159 页）。

图 225. I^2C 中断镜像图



23.6 I²C 调试模式

当微控制器进入调试模式时 (内核被暂停)，SMBus 超时检测可以正常运行或者停止，这取决于 DBG 模块中的 DBG_I2Cx_SMBUS_TIMEOUT 配置位的状态。 有关详细信息，请参阅第 23.15.2： 定时器，看门狗和 I²C 的调试支持 551 页。

23.7 I²C 寄存器

参见 31 页的章节 1.1 中对寄存器描述所采用的缩写列表。
本外设寄存器按字（32 位）访问。

23.7.1 控制寄存器 1(I2Cx_CR1)

地址偏移 : 0x00
复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PECEN	ALERT EN	SMBD EN	SMBH EN	GCEN	WUP EN	NOSTR ETCH	SBC
								rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXDMA EN	TXDMA EN	SWRST	ANF OFF	DNF				ERRIE	TCIE	STOP IE	NACK IE	ADDR IE	RXIE	TXIE	PE
rw	rw	w	rw	rw				rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位时的值。

位 23 PECEN： 启用 PEC
0: PEC 计算禁用
1: 启用 PEC 计算
注： 如果 SMBus 模式不被支持，这个位保留并由硬件强制为 0。
请参见章节 23.3： I2C 具体功能配备

位 22 ALERTEN： SMBus 通知使能
设备模式（SMBHEN = 0）：
0: 释放 SMBA 引脚为高，并禁用 NACK 之后的通知响应地址头： 0001100x。
1: 拉低 SMBA 引脚并启用 ACK 之后的通知响应地址头： 0001100x。
HOST 模式（SMBHEN = 1）：
0: SMBus 通知引脚（SMBA）不支持。
1: 支持 SMBus 通知引脚（SMBA）。
注： 当 ALERTEN = 0，SMBA 引脚可以作为一个标准的 GPIO。
如果 SMBus 模式不被支持，这个位保留并由硬件强制为 0。 请参见章节 23.3： I²C 具体功能配备

位 21 SMBDEN： SMBus 器件的默认地址启用

0: 禁用设备的默认地址。 地址 0b1100001x 会被 NACK。

1: 启用设备的默认地址。 地址 0b1100001x 会被 ACK。

注： 如果 SMBus 模式不被支持，这个位保留并由硬件强制为 0。

请参见章节 23.3： I2C 具体功能配备

位 20 SMBHEN： SMBus HOST 地址启用

0: 禁用 HOST 地址。 地址 0b0001000x 会被 NACK。

1: 启用 HOST 地址。 地址 0b0001000x 会被 ACK。

注： 如果 SMBus 模式不被支持，这个位保留并由硬件强制为 0。

请参见章节 23.3： I2C 具体功能配备

位 19 GCEN: 广播呼叫地址使能

0: 禁止广播呼叫。 地址 0b00000000 会被 NACK。

1: 启用广播呼叫。 地址 0b00000000 会被 ACK。

位 18 WUPEN： 从 STOP 模式唤醒

0: 禁止从 STOP 唤醒

1: 允许从 STOP 模式唤醒

注：如果从 STOP 模式唤醒功能不被支持，这个位保留并由硬件强制为 0。

请参见章节 23.3： I2C 具体功能配备

位 17 NOSTRETCH： 禁止时钟延长

该位用于在从机模式中禁止时钟延长。

0: 允许时钟延长

1: 禁止时钟延长

*注： 这个位只能在 I2C 被关闭的时候 (PE=0) 改写。*位 16 SBC： 从机字节控制

该位用于在从机模式使能硬件字节控制。

0: 从机字节控制模式关闭

1: 从机字节控制模式使能

位 15 RXDMAEN： DMA 接收请求使能

0: 关闭 DMA 接收请求

1: 开启 DMA 接收请求

位 14 TXDMAEN： DMA 发送请求使能

0: 关闭 DMA 发送功能

1: 开启 DMA 发送功能

位 13 SWRST: 软件复位

置 1 时，I2C 的 SCL 和 SDA 线都被释放。内部状态机和所有的状态为回到其复位值。控制位的内容被保留。

注： 该位只能写 1，读的话总返回 0。 写 '0' 也不会有作用。

位 12 ANFOFF： 模拟噪声滤波器关闭

0: 模拟噪声滤波器开启

1: 模拟噪声滤波器关闭

注： 这个位只能在 I2C 被关闭的时候 (PE=0) 改写。

位 11:8 DNF[3:0]： 数字噪声滤波器

这些位用来配置 SDA 和 SCL 输入上的数字噪声滤波器。 数字滤波器会用 DNF[3:0]* t_{I2C_CLK} 的长度来工作。

0000: 数字滤波器禁用

0001: 数字滤波器启用，并且滤波能力达到 1 个 t_{I2C_CLK}

……

1111: 数字滤波器启用，其滤波能力达到 15 个 t_{I2C_CLK}

注： 如果模拟滤波器也被启用了，那数字滤波器是加在模拟滤波器上的。

这个位只能在 I2C 被关闭的时候 (PE=0) 改写。

位 7 ERRIE： 错误中断使能

0: 错误检测中断禁止

1: 错误检测中断使能

注： 下列任何错误都可引发中断：

仲裁丢失（ARLO）

总线错误检测（BERR）

溢出 / 欠载（OVR）

超时检测（TIMEOUT）

PEC 错误检测（PECERR）

通知引脚事件检测（ALERT）

位 6 TCIE：传输完成中断使能

0: 传输完成中断禁用

1: 传输完成中断使能

注： 下列任何事件都可引发中断： 传输完成（TC），

传输完成重加载（TCR）

位 5 STOPIE： STOP 检测中断使能

0: 停止检测（STOPF）中断禁止

1: 停止检测（STOPF）中断使能

位 4 NACKIE： 收到 NACK 中断使能

0: （NACKF）收到中断禁用

1: （NACKF）收到中断允许

位 3 ADDRIE： 地址匹配中断使能（仅从机）

0: 地址匹配（ADDR）中断禁用

1: 启用地址匹配（ADDR）中断

位 2 RXIE： 接收中断使能

0: 接收（RXNE）中断禁止

1: 启用接收（RXNE）中断

位 1 TXIE： Tx 中断使能

0: 发送（TXIS）中断禁止

1: 发送（TXIS）中断使能

位 0 PE： 外设使能

0: 外设禁用

1: 外设使能

23.7.2 控制寄存器 2(USART_CR2)

地址偏移 : 0x04

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	PEC BYTE	AUTO END	RE LOAD	NBYTES[7:0]							
					rs	rw	rw	rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NACK	STOP	START	HEAD 10R	ADD10	RD_W RN	SADD[9:0]									
rs	rs	rs	rw	rw	rw	rw									

位 31:27 保留，必须保持复位时的值。

位 26 PECBYTE： 包错误检查字节

这个位由软件置位，在 PEC 传输完后由硬件清零，或者在收到 STOP 条件后，在收到地址匹配事件后以及在 PE=0 或者 SWRST 被写为 1 的时候都会被硬件清零。

0: 没有 PEC 传送。

1: 要求发送 / 接收 PEC

注： 对这个位写 0 是没有用的。

在 RELOAD 位为 1 的时候这个位也没有用了。

在从机模式下，如果 SBC=0，这个位也没有用。

如果 SMBus 模式不被支持，这个位保留并由硬件强制为 0。 请参见章节 23.3： I2C 具体功能配备

位 25 AUTOEND： 自动结束模式（主机模式）此位由软件置 1 和清零。

0: 软件结束模式： 当 NBYTES 个数据传输完毕后，TC 标志被置 1，SCL 被拉低。

1: 自动结束模式： NBYTES 个数据传输完后，会自动发送一个停止条件。

注： 该位在从机模式或 RELOAD 位被置 1 时有没有效果。

位 24 RELOAD： NBYTES 重装载模式

由软件设置和清除。

0: 在传输完 NBYTES 个字节后，传输结束（后面会跟一个 STOP 或者 RESTART）。

1: 传输 NBYTES 个字节后，传输并不结束（NBYTES 将被重装载）。 当 NBYTES 个数据传输完毕后，TC 标志被置 1，SCL 被拉低。

位 23:16 NBYTES[7:0]： 字节数

这里写入要发送 / 接收的字节数。 从机模式并且 SBC=0 的时候，这个地方的值无所谓。

注： 在 START 位被置 1 后，这些位不允许改变。

位 15 NACK： 产生 NACK （从机模式）

这个位由软件置位，在 NACK 发送后由硬件清零，或者在收到 STOP 条件后，在收到地址匹配事件后以及在 PE=0 或者 SWRST 被写为 1 的时候都会被硬件清零。

0: 在当前字节收到后发送 ACK。

1: 当前字节接收到后发送一个 NACK。

注： 对这个位写 0 是没有用的。

该位仅在从机模式下使用。 在主机接收模式，无论 NACK 位的值是什么，都会在最后一个字节之后和 STOP 条件或者 RESTART 条件之前自动发送。

在从机接收，无延长模式下，发生溢出事件时，会自动生成一个 NACK。这时的 NACK 位的值也是不管用的。

当启用了硬件 PEC 检查（PECBYTE = 1），PEC 的回应值也不依赖于 NACK 位的值。

位 14 STOP： 产生停止条件（主机模式）

该位由软件置 1，在检测到 STOP 条件或者 PE=0 或者 SWRST 被置 1 时由硬件清零。

在主机模式下：

0: 不产生 STOP 条件。

1: 当前字节传输完后产生停止条件。

注： 对这个位写 0 是没有用的。

位 13 START： 产生起始条件

该位通过软件设置，在发送完一个起始条件和地址序列之后由硬件清零，或者由于仲裁丢失、超时错误、PE=0 以及 SWRST 被置 1 等事件由硬件清零。 这个位也可以由软件向 I2Cx_ICR 寄存器的 ADDRCF 位写 1 来清零。

0: 没有起始条件产生

1: 产生 START/RESTART 条件：

- 如果 I2C 已经是在主机模式下并且 AUTOEND = 0，RELOAD=0，并且 NBYTES 个字节发送完毕后设置此位会产生重复起始条件。
- 否则只要总线空闲，设置此位将会立即产生一个起始条件。

注： 对这个位写 0 是没有用的。

可以设置 START 位，即使总线是忙的或者 I2C 处于从机模式。 在 RELOAD 位为 1 的时候这个位没有作用。

位 12 HEAD10R： 10 位地址头只读方向（主接收器模式）

0: 主机发送完整的 10 位从机地址读序列： 起始 +2 字节 10 位地址（写方向）+ 重新起始 +10 位地址中的前 7 位（读方向）。

1: 主机只发送 10 位地址的前 7 位，跟着是读方向。

注： 在 START 位被置 1 后，这个位不允许改变。

位 11 ADD10： 10 位地址模式（主机模式）

0: 主机按 7 位地址模式操作，

1: 主机按 10 位地址模式操作

注： 在 START 位被置 1 后，这个位不允许改变。

位 10 RD_WRN： 传输方向（主机模式）

0: 主机请求一个写传输。

1: 主机请求一个读传输。

注： 在 START 位被置 1 后，这个位不允许改变。

位 9:8 SADD [9:8]： 从机地址位 9:8（主机模式）

在 7 位地址模式下（ADD10 = 0）： 不用管这些个位

在 10 位地址模式下（ADD10 = 1）：

这些位应写入要发送的从机地址位的 9:8

注： 在 START 位被置 1 后，这些位不允许改变。

位 7:1 SADD [7:1]： 从机地址位 7:1（主机模式）

在 7 位地址模式下（ADD10 = 0）：

这些位应写入要发送的 7 位从机地址位

在 10 位地址模式下（ADD10 = 1）：

这些位应写入要发送的从机地址位的 7:1

注： 在 START 位被置 1 后，这些位不允许改变。

位 0 SADD0：从机地址的 0 位（主模式）

在 7 位地址模式下（ADD10 = 0）： 不用管这个位

在 10 位地址模式下（ADD10 = 1）：

这些位应写入要发送的从机地址位的位 0

注： 在 START 位被置 1 后，这些位不允许改变。

23.7.3 本机地址 1 寄存器(I2Cx_OAR1)

地址偏移 : 0x08

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OA1EN	Res.	Res.	Res.	Res.	OA1 MODE	OA1[9:8]		OA1[7:1]							OA1[0]
rw					rw	rw		rw							rw

位 31:16 保留，必须保持复位时的值。

位 15 OA1EN： 本机地址 1 启用

0: 禁用本机地址 1。接收到从机地址 OA1 后会用 NACK 回应。

1: 本机地址 1 启用 接收到从机地址 OA1 后会用 ACK 回应。 位 14:11 保留，必须保持复位时的值。

位 10 OA1MODE 本机地址 1 的 10 位模式

0: 本机地址 1 是 7 位地址。

1: 本机地址 1 是一个 10 位的地址。

注： 只有当 OA1EN = 0 该位才可以改写。

位 9:8 OA1[9:8]： 接口地址的 9:8 位

7 位地址模式： 无所谓

10 位地址模式： 地址的位 9:8

注： 这些位只可以在 OA1EN = 0 的时候改写。

位 7:1 OA1[7:1]： 接口地址的 7:1

注： 这些位只可以在 OA1EN = 0 的时候改写。

位 0 OA1 的 [0]： 接口地址的 0 位

7 位地址模式： 无所谓

10 位地址模式： 地址的 0 位

注： 只有当 OA1EN = 0 该位才可以改写。

23.7.4 本机地址 2 寄存器(I2Cx_OAR2)

地址偏移 : 0x0C

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OA2EN	Res.	Res.	Res.	Res.	OA2MSK[2:0]			OA2[7:1]							Res.
rw					rw			rw							

位 31:16 保留，必须保持复位时的值。

位 15 OA2EN： 本机地址 2 启用

0: 禁用本机地址 2。 接收到从机地址 OA2 后会用 NACK 回应。

1: 本机地址 2 启用 接收到从机地址 OA2 后会用 ACK 回应。 位 14:11 保留，必须保持复位时的值。

位 10:8 OA2MSK [2:0]： 本机地址 2 屏蔽

000: 没有屏蔽

001: OA2 [1] 被屏蔽掉，可忽略。 只有 OA2 [7:2] 参与比较。

010: OA2 [2:1] 被屏蔽掉，可忽略。 只有 OA2 [7:3] 参与比较。

011: OA2 [3:1] 被屏蔽掉，可忽略。 只有 OA2 [7:4] 参与比较。

100: OA2 [4:1] 被屏蔽掉，可忽略。 只有 OA2 [7:5] 参与比较。

101: OA2 [5:1] 被屏蔽掉，可忽略。 只有 OA2 [7:6] 参与比较。

110: OA2 [6:1] 被屏蔽掉，可忽略。 只有 OA2 [7] 参与比较。

111: OA2 [7:1] 被屏蔽掉，可忽略。 没有比较，所有的（除保留地址）收到的 7 位地址都会用 ACK 回应。

注： 这些位只可以在 OA2EN = 0 的时候改写。

只要是 OA2MSK 不等于 0，保留的 I2C 地址（0b0000xxx 和 0b1111xxx）也不被响应，即使是比较匹配上了也不行。

位 7:1 OA2 [7:1]： 接口地址位 7:1

注： 这些位只可以在 OA2EN = 0 的时候改写。

位 0 保留，必须保持复位时的值。

23.7.5 时序寄存器(I2Cx_TIMINGR)

地址偏移 : 0x10

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRESC[3:0]				Res.	Res.	Res.	Res.	SCLDEL[3:0]				SDADEL[3:0]			
rw								rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCLH[7:0]								SCLL[7:0]							
rw								rw							

位 31:28 PRESC [3:0]： 时序预分频器

此字段用于分频 I2C_CLK, 为数据的建立和保持时间计数器（参见 502 页 I2C 时序）和 SCL 高电平低电平计数器（参见 516 页 I2C 主机初始化）产生时钟周期 t_{PRESC} 。

t_{PRESC} = (PRESC+1) x t_{I2C_CLK}

位 27:24 保留，必须保持复位时的值。

位 23:20 SCLDEL [3:0]： 数据建立时间

此字段用于生成发送模式中 SDA 的沿和 SCL 上升沿之间的延迟 t_{SCLDEL}。

t_{SCLDEL} = (SCLDEL+1) x t_{PRESC}

注： t_{SCLDEL} 用于产生 $t_{SU:DAT}$ 时序。

位 19:16 SDADEL [3:0]： 数据保持时间

此字段用于生成发送模式中 SCL 的下降沿和 SDA 的沿之间的延迟 t_{SDADEL}。

t_{SDADEL}= SDADEL x t_{PRESC}

Note: SDADEL is used to generate tHD:DAT timing.

位 15:8 SCLH[7:0]： SCL 高电平时间（主机模式）

此字段用于在主机模式下产生 SCL 的高电平时间。

t_{SCLH} = (SCLH+1) x t_{PRESC}

注： SCLH is 同时用于产生 $t_{SU:STO}$ and $t_{HD:STA}$ 时序。

位 7:0 SCLL [7:0]： SCL 低电平时间（主机模式）

此字段用于在主机模式下产生 SCL 的低电平时间。

t_{SCLL} = (SCLL+1) x t_{PRESC}

注： SCLL 同时用于产生 $t_{BUF\ and\ tSU:STA}$ 时序。

注： 该寄存器必须在 I2C 被禁用（PE= 0）的时候配置。

23.7.6 超时寄存器(I2Cx_TIMEOUTR)

地址偏移 : 0x14

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TEXTEN	Res.	Res.	Res.	TIMEOUTB											
rw				rw											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMOUTEN	Res.	Res.	TIDLE	TIMEOUTA											
rw			rw	rw											

位 31 TEXTEN： 外部时钟超时启用

0: 外部时钟超时检测被禁用

1: 外部时钟超时检测启用。 当 I^2C 接口将 SCL 拉低的累计时间达到了 $t_{TLOW:EXT}$， 会检测到一个超时错误 (TIMEOUT=1).

位 30:29 保留，必须保持复位时的值。

位 27:16 TIMEOUTB [11:0]： 总线超时 B

此字段用于配置累计时钟延长超时：

在主机模式下，要检测的累计主机时钟低延长时间 ($t_{LOW:MEXT}$)。

在从机模式下，要检测的累积从机时钟低延长时间 ($t_{LOW:SEXT}$)。

$t_{TLOW:EXT}$= (TIMEOUTB+1) x 2048 x t_{I2C_CLK}

注： 这些位只可以在 TEXTEN = 0 的时候改写。

位 15 TIMOUTEN： 时钟超时检测启用

0: SCL 超时检测被禁用

1: 启用 SCL 超时检测： 当 SCL 保持低的时间超过 $t_{TIMEOUT}$ (TIDLE=0) 或保持高的时间超过 t_{IDLE} (TIDLE=1), 会检测到一个超时错误 (TIMEOUT=1).

位 14:13 保留，必须保持复位时的值。 位 12 TIDLE： 空闲时钟超时检测

0: TIMEOUTA 用于检测 SCL 低电平超时

1: TIMEOUTA 用于同时检测 SCL 和 SDA 高电平超时（总线空闲条件）

注： 只有当 TIMOUTEN = 0 该位才可以改写。

位 11:0 TIMEOUTA [11:0]：总线超时 A

此字段用于配置：

- 在 TIDLE=0 的时候，SCL 低超时条件 $t_{TIMEOUT}$

  $t_{TIMEOUT}$= (TIMEOUTA+1) x 2048 x t_{I2C_CLK}
- TIDLE=1 的时候，总线空闲条件 (SCL 和 SDA 同时为高)

  t_{IDLE}= (TIMEOUTA+1) x 4 x t_{I2C_CLK}

  注： 这些位只可以在 TIMOUTEN = 0 的时候改写。

注： 如果 SMBus 模式不被支持，这个寄存器保留并由硬件强制为 0x00000000。

请参见章节 23.3： I2C 具体功能配备

23.7.7 中断和状态寄存器(I2Cx_ISR)

地址偏移 : 0x18

复位值 : 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDCODE[6:0]							DIR
								r							r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUSY	Res.	ALERT	TIME OUT	PEC ERR	OVR	ARLO	BERR	TCR	TC	STOPF	NACKF	ADDR	RXNE	TXIS	TXE
r		r	r	r	r	r	r	r	r	r	r	r	r	r_w1	r_w1

位 31:24 保留，必须保持复位时的值。

位 23:17 ADDCODE[6:0]： 匹配的地址码（从机模式）

这些位由地址匹配事件发生时所接收到的地址（ADDR= 1）来更新。

在 10 位地址的情况下，ADDCODE 提供 10 位地址的头 2 位以后的地址。

位 16 DIR： 传输方向（从机模式）

此标志在地址匹配事件发生时（ADDR= 1）更新。

0: 写传输，从机进入接收模式。

1: 读传输，从机进入发送模式。

位 15 BUSY： 总线忙

此标志表示总线上正在通讯。 它在检测到一个起始条件时由硬件设置。 它在检测到一个停止条件时由硬件清零，或者在 PE=0 以及 SWRST 被置 1 的时候清零。

位 14 保留，必须保持复位时的值。 位 13 ALERT： SMBus 通知

该标志在 SMBHEN=1（SMBus HOST 配置）及 ALERTEN=1 的条件下，在 SMBA 脚上检测到 SMBALERT 事件（下降沿）时，由硬件置 1。 通过设置 ALERTCF 位，由软件清零。

注： 此位在 PE = 0 或 SWRST 被置 1 时，由硬件清零。

如果 SMBus 模式不被支持，这个位保留并由硬件强制为 0。 请参见章节 23.3：I2C 具体功能配备

位 12 TIMEOUT： 超时或 tLOW 检测标志

在超时或外部时钟超时发生时，该标志被硬件置 1。 通过设置 TIMEOUTCF 位，由软件清零。

注： 此位在 PE = 0 或 SWRST 被置 1 时，由硬件清零。

如果 SMBus 模式不被支持，这个位保留并由硬件强制为 0。 请参见章节 23.3：I2C 具体功能配备

位 11 PECERR： 接收中的 PEC 错误

该标志在收到的 PEC 值和 PEC 寄存器的内容不匹配时，由硬件置 1。 收到错误的 PEC 后，会自动发送一个 NACK。 这个位可以通过将 PECCF 位置 1，由软件清零。

注： 此位在 PE = 0 或 SWRST 被置 1 时，由硬件清零。

如果 SMBus 模式不被支持，这个位保留并由硬件强制为 0。 请参见章节 23.3：I2C 具体功能配备

位 10 OVR： 溢出 / 欠载 (从机模式)

此标志在从机模式下 NOSTRETCH = 1 时，发生溢出 / 欠载错误的时候，由硬件置 1。通过设置 OVRCF 位，由软件清零。

注： 此位在 PE = 0 或 SWRST 被置 1 时，由硬件清零。

位 9 ARLO： 仲裁丢失

该标志在总线仲裁丢失的情况下由硬件置 1。 通过设置 ARLOCF 位，由软件清零。

注： 此位在 PE = 0 或 SWRST 被置 1 时，由硬件清零。

位 8 BERR： 总线错误

此标志在检测到错位的起始或者停止条件的时候由硬件置 1。通过设置 BERRCF 位，由软件清零。

注： 此位在 PE = 0 或 SWRST 被置 1 时，由硬件清零。

位 7 TCR： 传输完成重加载

该标志在 RELOAD=1，并且 NBYTES 个数据发送完毕后，由硬件置 1。 向 NBYTES 写入是一个非零的值时，TCR 标志由软件清除。

注： 此位在 PE = 0 或 SWRST 被置 1 时，由硬件清零。

此标志仅适用于主机模式，或从机模式中 SBC 位为 1 的情形。

位 6 TC： 发送完毕（主机模式）

该标志在 RELOAD=0，AUTOEND=0，并且 NBYTES 个数据发送完毕后，由硬件置 1。它在软件将 START 或 STOP 位置 1 的时候清零。

注： 此位在 PE = 0 或 SWRST 被置 1 时，由硬件清零。

位 5 STOPF： 停止检测标志

该标志在外设参与传输的下列情况下，在总线上检测到一个停止条件时，由硬件置 1：

- 作为主机，如果由外设生成一个停止条件的时候。
- 或作为从机，如果外设在这次传输之前被正确的寻址到了。

这个位可以通过将 STOPCF 位置 1，由软件清零。

注： 此位在 PE = 0 或 SWRST 被置 1 时，由硬件清零。

位 4 NACKF： 收到 NACK 标志

该标志在一个字节传输后收到一个 NACK 的时候由硬件设置。 这个位可以通过将 NACKCF 位置 1，由软件清零。

注： 此位在 PE = 0 或 SWRST 被置 1 时，由硬件清零。

位 3 ADDR： 地址匹配（从机模式）

这个位在收到的从机地址与其中一个有效的从机地址匹配的时候，由硬件置 1。 通过设置 ADDRCF 位，由软件清零。

注： 此位在 PE = 0 或 SWRST 被置 1 时，由硬件清零。

位 2 RXNE： 接收数据寄存器非空（接收）

此位在当接收到的数据被复制到 I2Cx_RXDR 寄存器，准备好被软件读取的时候由硬件置位。 在读取 I2Cx_RXDR 时 RXNE 会被清除。

注： 此位在 PE = 0 或 SWRST 被置 1 时，由硬件清零。

位 1 TXIS： 发送中断状态（发送）

在 I2Cx_TXDR 寄存器为空的时候由硬件置 1，这时必须把要发的数据写到 I2Cx_TXDR 寄存器。下一个要发送的数据被写到 I2Cx_TXDR 寄存器的时候它会被清除。

该位只在 NOSTRETCH = 1 的时候可以由软件写成 1，以生成一个 TXIS 事件（如果 TXIE = 1 就有中断，如果 TXDMAEN = 1 就有 DMA 请求）。

注： 此位在 PE = 0 或 SWRST 被置 1 时，由硬件清零。

位 0 TXE： 发送数据寄存器空（发送）

在 I2Cx_TXDR 寄存器为空的时候由硬件置 1。 下一个要发送的数据被写到 I2Cx_TXDR 寄存器的时候它会被清除。

该位可通过软件写 1，以清空发送数据寄存器 I2Cx_TXDR。

注： 此位在 PE = 0 或 SWRST 被置 1 时，由硬件清零。

23.7.8 中断清除寄存器(I2Cx_ICR)

地址偏移 : 0x1C

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	ALERT CF	TIM OUTCF	PECCF	OVRCF	ARLO CF	BERR CF	Res.	Res.	STOP CF	NACK CF	ADDR CF	Res.	Res.	Res.
		w	w	w	w	w	w			w	w	w			

位 31:14 保留，必须保持复位时的值。

位 13 ALERTCF： 通知标志清零

对这个位写 1，会清除 I2Cx_ISR 寄存器中的 ALERT 标志位。

注： 如果 SMBus 模式不被支持，这个位保留并由硬件强制为 0。

请参见章节 23.3： I2C 具体功能配备

位 12 TIMOUTCF： 超时检测标志清除

对这个位写 1，会清除 I2Cx_ISR 寄存器中的 TIMEOUT 标志位。

注： 如果 SMBus 模式不被支持，这个位保留并由硬件强制为 0。

请参见章节 23.3： I2C 具体功能配备

位 11 PECCF： PEC 错误标志清除

对这个位写 1，会清除 I2Cx_ISR 寄存器中的 PECERR 标志位。

注： 如果 SMBus 模式不被支持，这个位保留并由硬件强制为 0。

请参见章节 23.3： I2C 具体功能配备

位 10 OVRCF： 溢出 / 欠载标志清除

对这个位写 1，会清除 I2Cx_ISR 寄存器中的 OVR 标志位。

位 9 ARLOCF： 仲裁丢失标志清除

对这个位写 1，会清除 I2Cx_ISR 寄存器中的 ARLO 标志位。

位 8 BERRCF： 总线错误标志清除

对这个位写 1，会清除 I2Cx_ISR 寄存器中的 BERRF 标志位。 位 7:6 保留，必须保持复位时的值。

位 5 STOPCF： 停止检测标志清除

对这个位写 1，会清除 I2Cx_ISR 寄存器中的 STOPF 标志位。

位 4 NACKCF： 收到 NACK 标志清除

对这个位写 1，会清除 I2Cx_ISR 寄存器中的 NACKF 标志位。

位 3 ADDRCF： 地址匹配标志清除

对这个位写 1，会清除 I2Cx_ISR 寄存器中的 ADDR 标志位。 对这个位写 1，还会清除 I2C_CR2 寄存器中的 START 位。

位 2:0 保留，必须保持复位时的值。

23.7.9 PEC 寄存器(I2Cx_PECR)

地址偏移 : 0x20

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PEC[7:0]							
								r							

位 31:8 保留，必须保持复位时的值。

位 7:0 PEC [7:0] 包错误检查寄存器

当 PECEN = 1 时，此字段包含内部 PEC 结果。

PEC 在 PE = 0 或 SWRST 被置 1 时，由硬件清零。

注： *如果 SMBus 模式不被支持，这个寄存器保留并由硬件强制为 0x00000000。*

请参见章节 23.3： I2C 具体功能配备

23.7.10 接收数据寄存器(I2Cx_RXDR)

地址偏移 : 0x24

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXDATA[7:0]							
								r							

位 31:8 保留，必须保持复位时的值。

位 7:0 RXDATA [7:0] 8 位接收数据

从 I²C 总线接收的数据字节。

23.7.11 发送数据寄存器(I2Cx_TXDR)

地址偏移 : 0x28

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXDATA[7:0]							
								rw							

位 31:8 保留，必须保持复位时的值。

位 7:0 TXDATA[7:0] 8 位发送数据

要发送到 I2C 总线上的数据字节。

注： 这些位只可以在 TXE = 1 的时候改写。

23.8 I²C 寄存器映射

下表给出了 UI²C 寄存器镜像和复位值。

表 80. I²C 寄存器镜像和复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0	I2Cx_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PECEN	ALERTEN	SMBDEN	SMBHEN	GCEN	WUPEN	NOSTRETCH	SBC	RXDMAEN	TXDMAEN	SWRST	ANFOFF	DNF				ERRIE	TCIE	STOPIE	NACKIE	ADDRIE	RXIE	TXIE	PE
	Reset Value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x4	I2Cx_CR2	Res.	Res.	Res.	Res.	Res.	PECBYTE	AUTOEND	RELOAD	NBYTES								NACK	STOP	START	HEAD10R	ADD10	RD_WRN	SADD									
	Reset Value						0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x8	I2Cx_OAR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OA1EN	Res.	Res.	Res.	Res.	OA1MODE	OA1									
	Reset Value																	0					0	0	0	0	0	0	0	0	0	0	0
0xC	I2Cx_OAR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OA2EN	Res.	Res.	Res.	Res.	OA2MSK	OA2									Res.
	Reset Value																	0					0	0	0	0	0	0	0	0	0	0	
0x10	I2Cx_TIMINGR	PRESC				Res.	Res.	Res.	Res.	SCLDEL				SDADEL				SCLH						SCLL									
	Reset Value	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	I2Cx_TIMEOUTR	TEXTEN	Res.	Res.	Res.	TIMEOUTB												TIMOUTEN	Res.	Res.	TIDLE	TIMEOUTA											
	Reset Value	0				0	0	0	0	0	0	0	0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	I2Cx_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDCODE							DIR	BUSY	Res.	ALERT	TIMEOUT	PECERR	OVR	ARLO	BERR	TCR	TC	STOPF	NACKF	ADDR	RXNE	TXIS	TXE
	Reset Value									0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	1
0x1C	I2Cx_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ALERTCF	TIMOUTCF	PECCF	OVRCF	ARLOCF	BERRCF	Res.	Res.	STOPCF	NACKCF	ADDRCF	Res.	Res.	Res.
	Reset Value																			0	0	0	0	0	0			0	0	0			
0x20	I2Cx_PECR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PEC							
	Reset Value																									0	0	0	0	0	0	0	0
0x24	I2Cx_RXDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXDATA							
	Reset Value																									0	0	0	0	0	0	0	0
0x28	I2Cx_TXDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXDATA							
	Reset Value																									0	0	0	0	0	0	0	0

寄存器边界地址请参见 35 页的章节 2.2.2

24 通用同步异步串行收发器（USART）

24.1 USART 介绍

通用同步异步收发器（USART）提供了一个灵活的方式，使 MCU 可以与外部设备通过工业标准 NRZ 的形式实现全双工异步串行数据通讯。 USART 可以使用分数波特率发生器，提供了超宽的波特率设置范围。

USART 支持同步通讯模式和半双工单线通讯。 也支持 LIN（本地互联网络），智能卡协议和 IrDA（红外数据协会）SIR ENDEC 规范和 modem 流控操作（CTS/RTS）同时还 支持多机通讯方式。

可以使用 DMA 实现多缓冲区设置，从而能够支持高速数据通讯。

24.2 USART 主要功能

- 全双工，异步通讯
- NRZ 标准格式（mark/space）
- 可配置的 16 倍或 8 倍过采样方法提供速度和时钟容忍度间的灵活选择
- 小数波特率发生器
  - 一个公共的可编程收发波特率高达 6Mbit/s（时钟频率为 48MHz，过采样率为 8 倍时）
- 双时钟驱动支持
  - UART 从 Stop 模式唤醒功能
  - 方便的波特率改变，不依赖对 PCLK 的改变。
- 自动波特率检测
- 数据字长可编程（8 或 9 位）
- 高位在前或低位在前可设置
- 停止位个数可设置 - 支持 1 个或 2 个停止位
- 同步模式下时钟输出功能，实现同步通讯
- 单线半双工通讯
- DMA（直接内存访问）支持下的连续数据通讯
  - 利用 DMA 功能将收 / 发字节缓冲到保留的 SRAM 空间
- 针对接收器和发送器的单独的使能位
- 针对发送和接受的单独的信号极性控制
- 可配置为 Tx/Rx 引脚互换
- 用于 MODEM 的硬件流控制和 RS-485 发送使能控制
- 发送检测标志有：
  - 接收缓冲区满
  - 发送缓冲区空
  - 忙和发送结束标志

- 校验控制：
  - 发送奇偶校验位
  - 接收数据的奇偶检查
- 四种错误检测：
  - 溢出错误
  - 噪声检测错误
  - 帧错误
  - 校验错误
- 十四个中断源和中断标志
  - CTS 切换
  - LIN 断开检测
  - 发送数据寄存器空
  - 发送完成
  - 接收数据寄存器满
  - 检测到线路空闲
  - 溢出错误
  - 帧错误
  - 噪声错误
  - 奇偶错误
  - 地址 / 字符匹配
  - 接收超时中断
  - 块结束中断
  - 从 Stop 模式唤醒
- 多机通讯 - 如果地址不匹配则进入静默模式
- 从静默模式唤醒（检测到线路空闲或检测到地址标记）
- 两个接收唤醒模式： 地址位（MSB，第 9 位），线路空闲

24.3 USART 扩展功能

- LIN 主机的断开信号发送能力和 LIN 从机的断开信号检测能力
  - 将 USART 硬件设置成 LIN 模式时，有 13 位的断开信号发生器和 10/11 位的断开信号检测功能
- IrDA SIR 编解码器
  - 普通模式下支持 3/16 位时长
- 智能卡模式
  - 支持 ISO/IEC7816-3 标准定义的 T=0 和 T=1 智能卡异步协议
  - 智能卡用到的 1.5 个停止位
- 支持 ModBus 通讯
  - 超时检测功能
  - CR/LF 字符识别

24.4 USART 具体功能配备

本文描述了 USART1 所实现的全套功能。 USART2 功能稍有裁剪，但有的功能都和 USART1 一样。 差异如下表所示。

表 81. STM32F0xx USART 具体功能配备

USART 模式 / 功能 (1)	USART1	USART2
MODEM 所需的硬件流控制	X	X
用 DMA 实现连续通讯	X	X
多机通讯	X	X
同步模式	X	X
智能卡模式	X	
单线半双工通讯	X	X
红外 IrDA SIR 编解码	X	
LIN 模式	X	
双时钟驱动和从 Stop 模式唤醒	X	
接收超时中断	X	
Modbus 通讯	X	
自动波特率检测	X	
RS485 用的驱动使能信号	X	X

1. X= 支持

24.5 USART 功能描述

接口与外部设备通过三个引脚相连（见图 226） 任何 USART 双向通讯要求最少有两个引脚：接收数据输入（RX）和发送数据输出（TX）

RX: 接收数据输入是串行数据的输入口。 使用过采样技术来完成数据恢复，以区别输入数据和噪声。

TX: 数据发送输出。 当发送器被禁止，输出脚回到其 I/O 口配置状态。 当发送器被使能，但不发送数据时，TX 脚为高电平输出。 在单线和智能卡模式中，这个口线既用于发送数据也用于接收数据。

通过这些引脚，串行数据用数据帧的形式发送和接收：

- 在发送和接收之前为空闲状态
- 起始位
- 数据字（8 或 9 位）最低有效位在前
- 1,1.5,2 个停止位表明帧的结束
- 采用小数波特率发生器，整数 12 位小数 4 位
- 一个状态寄存器（USART_ISR）
- 分开的接收和发送数据寄存器（USART_RDR，USART_TDR）
- 一个波特率寄存器（USART_BRR）-12 位整数和 4 位小数。
- 一个保护时间寄存器（USART_GTPR）用于智能卡模式。

参见章节 24.7： USART 寄存器每一位的详细描述。

下面的引脚在同步模式和智能卡模式中会用到：

- SCLK: 时钟输出。 该引脚会输出发送数据的同步时钟
  发送功能和 SPI 主模式一致（起始位和停止位上没有时钟脉冲，在最后一位数据上可选择有无时钟脉冲）。 同时，数据可经由 RX 引脚同步接收，这可以被用来连接那种通过移位寄存器相连的外设（例如： LCD 驱动器）。 时钟相位和极性可软件设定。 智能卡模式中，SCLK 引脚会向智能卡提供时钟。

下列引脚用于支持硬件流控制模式：

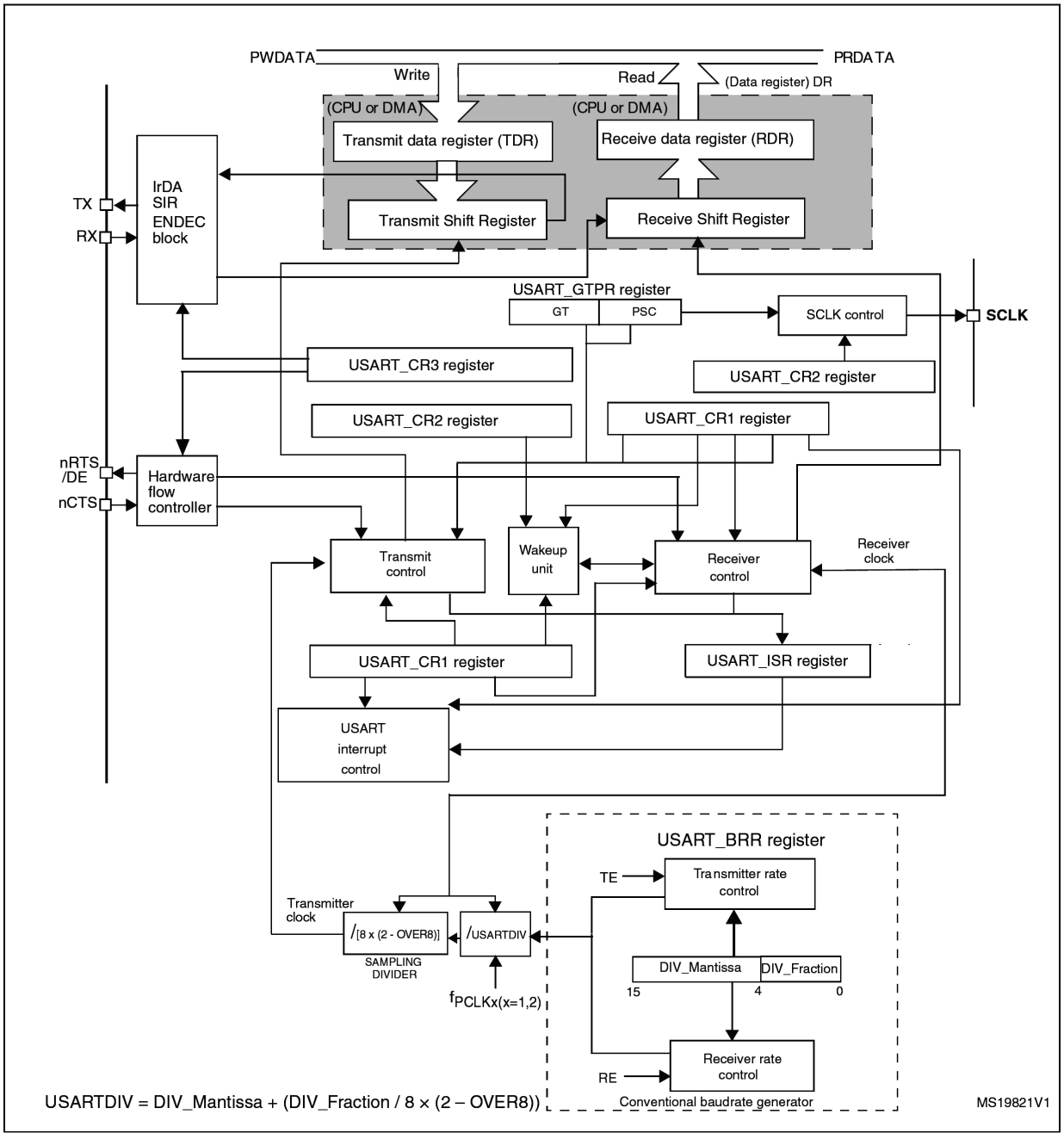
- nCTS: 低发送，当高电平时作为发送阻塞信号。
- nRTS: 请求发送，表明 USART 已经准备好接收数据（低的时候）。

下列引脚在 RS485 驱动使能控制的时候会用到：

- DE: 驱动使能将外部收发器的发送模式激活。

注： DE 和 nRTS 共用同一个外部引脚。

图 226. USART 框图



24.5.1 USART 符号描述

配置 USART_CR1 寄存器（参见图 227）中的 M 位可选择 8 位或 9 位字长。

默认设置中，发送和接收的起始位都是低电平。 而停止位都是高电平。

这个逻辑可以在极性控制中单独的设置为反向。

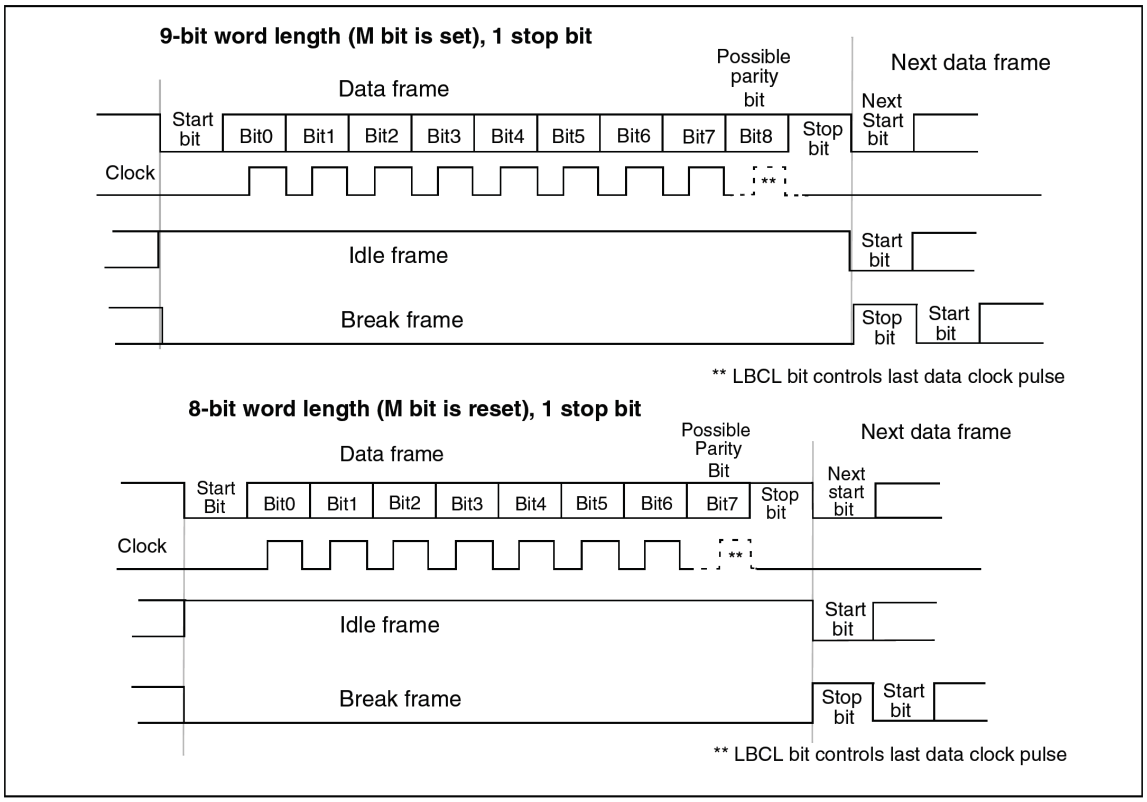
空闲符号被视为完全由 '1' 组成的完整的数据帧，后面跟着包含了数据的下一帧的开始位 ('1' 的位数也包括了停止位的位数)。

断开符号 被视为在一个帧周期内全部收到’0’(包括停止位期间，也是’0’)。 在断开帧结束时，发送器会再插入 2 个停止位。

发送和接收由一个共用的波特率发生器驱动，当发送器和接收器的使能位分别置 1 时，分别为其产生时钟。

下面是每个模块的详细说明。

图 227. 字长设置



24.5.2 发送器

发送器根据 M 位的状态发送 8 位或 9 位的数据字。

发送使能位必须置 1 以打开发送功能。 发送移位寄存器中的数据在 TX 脚上输出，相应的时钟脉冲在 CK 脚上输出。

字符发送

在 USART 发送期间，在 TX 引脚上首先移出数据的最低有效位。 在此模式里，USART_TDR 寄存器充当了一个内部总线和发送移位寄存器之间的缓冲器 (TDR)(见图 226)。

每个字符之前都有一个低电平的起始位 . 之后跟着停止位，停止位的数目是可选择的。

USART 支持多种停止位的选择： ：0.5、1、1.5 和 2 个停止位。

注： *1. 在向 USART_TDR 写数据之前必须先令 TE 位为 1。*

2. 在 TE 位被置 1 后将会发送一个空闲帧。

可配置的停止位

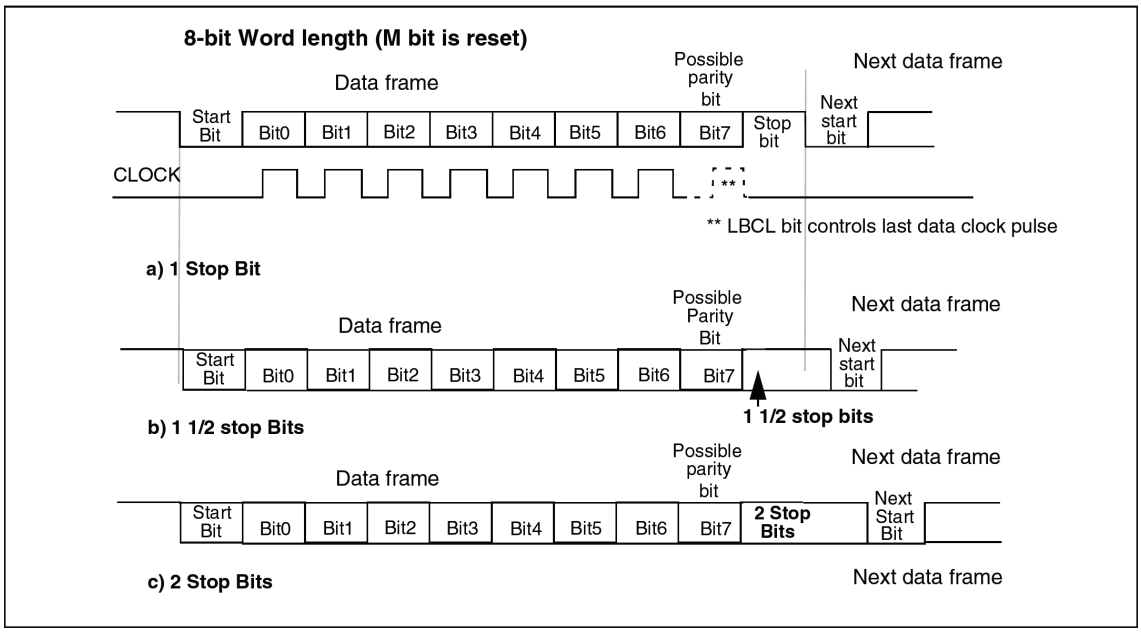
随每个字符发送的停止位的位数可以通过控制寄存器 2 的位 13、12 进行编程。

1. 1 个停止位： 停止位的位数的默认值。
2. 2 个停止位： 可用于常规 USART 模式、单线模式以及调制解调器模式。
3. 1.5 个停止位： 在智能卡模式下发送和接收数据时使用。

空闲帧包括了停止位。

断开帧是 10 位低电平 (当 m=0 时)；或者 11 位低电平 (m=1 时)，后跟 2 个停止位。 没办法传输更长的断开帧 (长度大于 10 或者 11 位的那种)。

图 228. 可配置的停止位



配置步骤：

1. 设置 USART_CR1 的 M 位来定义字长。
2. 利用 USART_BRR 寄存器选择希望的波特率。
3. 在 USART_CR2 中设置停止位的位数。
4. 通过将 USART_CR1 寄存器的 UE 位置 1 来使能 USART。
5. 如果采用多缓冲器通信，配置 USART_CR3 中的 DMA 使能位 (DMAT)。 按多缓冲器通信中的描述配置 DMA 寄存器。
6. 设置 USART_CR1 中的 TE 位，发送一个空闲帧作为第一次数据发送。
7. 把要发送的数据写进 USART_TDR 寄存器 (此动作将清除 TXE 位)。 在只有一个缓冲器的情况下，对每个待发送的数据重复步骤 7。重复步骤 7 之前应等待 TXE 变成 1。
8. 在 USART_TDR 寄存器中写入最后一个数据字后，要等待 TC=1， 它表示最后一个数据帧的传输结束。 当需要关闭 USART 或需要进入停机模式之前，需要确认传输结束，避免破坏最后一次传输。

单字节通信

清零 TXE 位总是通过对数据寄存器的写操作来完成的。TXE 位由硬件来设置，它表明：

- 数据已经从 TDR 移送到移位寄存器，数据发送已经开始。
- USART_TDR 寄存器被清空
- 下一个数据可以被写进 USART_TDR 寄存器而不会覆盖先前的数据。

如果 TXEIE 位被设置，此事件将产生一个中断请求。

如果此时 USART 正在发送数据，对 USART_TDR 寄存器的写操作把数据存进 TDR 寄存器，并在当前传输结束时把该数据复制进移位寄存器。

如果此时 USART 没有在发送数据，处于空闲状态，对 USART_TDR 寄存器的写操作将导致直接把数据放进移位寄存器，数据传输开始，TXE 位则立即被置起。

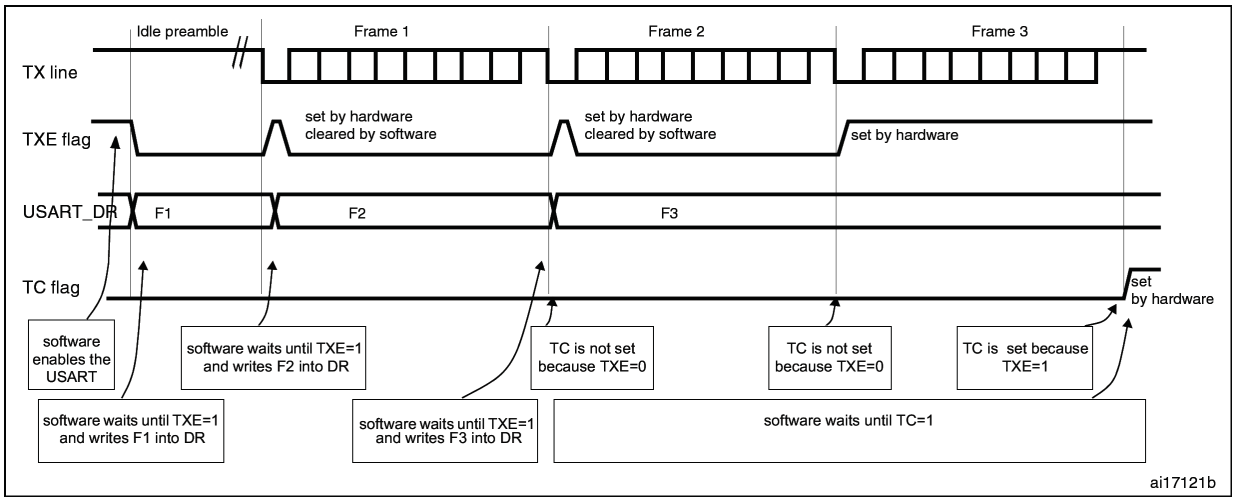
当一个字节发送完成时 (停止位发送后) 并且之前已经置 1 了 TXE 位时，TC 位会被置 1。 如果 USART_CR1 寄存器中的 TCIE 位是 1 时，则会产生中断。

在 USART_TDR 寄存器中写入了最后一个数据字后，在关闭 USART 模块之前或设置微控制器进入低功耗模式之前，必须先等待 TC=1。（见图 229： 发送时 TC/TXE 的变化情况）

注： 关闭 USART 的正确步骤如下：

- *清除 TE 位（如果有数据正在传输或者 USART_ 寄存器中还有待发送的数据，会在关闭动作生效前发送完）*
- *TC 位会在发送完毕后被置 1*
- *在 TC=1 后清除 UE 位*

图 229. 发送时 TC/TXE 的变化情况



断开符号

向 SBKRQ 位写 1 可发送一个断开符号。断开符号的长度取决于 M 位 (见图 227)。

如果设置 SBK=1，在完成当前数据发送后，将在 TX 线上发送一个断开符号。 SBKF 位会在写操作发生时置 1，在断开符号发送完成时 (在断开符号的停止位发出时) 被硬件清零。USART 在最后一个断开帧的结束处插入一个逻辑’1’并保持 2 位时长作为停止，以保证能识别下一帧的起始位。

空闲符号

将 TE 置 1 将使得 USART 在第一个数据前发送一空闲符号。

24.5.3 接收器

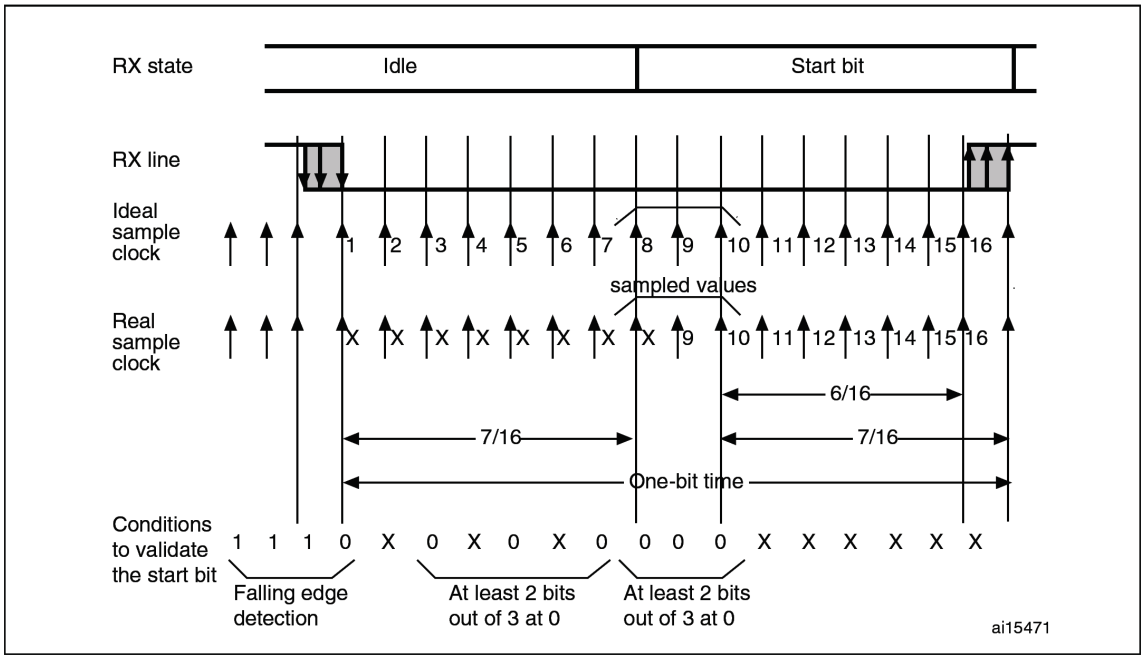
接收器根据 USART_CR1 寄存器中 M 位的状态接收 8 位或 9 位的数据字。

起始位侦测

过采样率设置成 16 或 8，不影响起始位侦测的顺序。

在 USART 中，如果辨认出一个特殊的采样序列，那么就认为侦测到一个起始位。 该序列为：1 1 1 0 X 0 X 0X 0X 0 X 0X 0.

图 230. 过采样率为 16 或 8 时的起始位侦测



注：如果该序列不完整，那么接收端将退出起始位侦测并回到空闲状态 (不设置标志位) 开始等待下降沿。

如果 3 个采样点都为 '0' (在第 3、5、7 位的第一次采样，和在第 8、9、10 的第二次采样都为 '0')，则确认收到起始位，这时 RXNE 标志会由硬件置 1，如果这时 RXNEIE=1，则会产生中断请求。

如果两次 3 个采样点上仅有 2 个是 '0' (第 3、5、7 位的采样点和第 8、9、10 位的采样点)，那么起始位仍然是有效的，但是会设置 NE 噪声标志位。如果不能满足这个条件，则中止起始位的侦测过程，接收器会回到空闲状态 (不设置标志位)。

如果有一次 3 个采样点上仅有 2 个是 '0' (第 3、5、7 位的采样点或第 8、9、10 位的采样点)，那么起始位仍然是有效的，但是会设置 NE 噪声标志位。

字符接收

在 USART 接收期间，数据的最低有效位（默认情况下）首先从 RX 脚移进。在此模式里，USART_RDR 寄存器充当了一个位于内部总线和接收移位寄存器之间的缓冲器。

配置步骤：

1. 设置 USART_CR1 的 M 位来定义字长。
2. 利用波特率寄存器 USART_BRR 选择希望的波特率。
3. 在 USART_CR2 中设置停止位的位数。
4. 通过将 USART_CR1 寄存器的 UE 位置 1 来激活 USART。
5. 如果采用多缓冲器通信，配置 USART_CR3 中的 DMA 使能位 (DMAR)。 按多缓冲器通信中的描述配置 DMA 寄存器。
6. 将 USART_CR1 的 RE 位置 1。 这将激活接收器，使它开始寻找起始位。

当一个字符被接收到时

- RXNE 位被置 1。 它表明移位寄存器的内容被转移到 RDR。 换句话说，数据已经被接收并且可以被读出 (包括与之有关的错误标志)。
- 这时如果 RXNEIE 位是 1，将会引起中断请求。
- 在接收期间如果检测到帧错误，噪音或溢出错误，错误标志将被置起。 PE 标志也会和 RXNE 一起被置 1。
- 在多缓冲器通信时，RXNE 在每个字节接收后被置起，并由 DMA 对数据寄存器的读操作而清零。
- 在单缓冲器模式里，由软件读 USART_RDR 寄存器完成对 RXNE 位清除。 RXNE 标志也可以通过对 USART_RQR 寄存器中的 RXFRQ 位写 1 来清除。 RXNE 位必须在下一字符接收结束前被清零，以避免溢出错误。

断开符号

当接收到一个断开帧时，USART 会像处理帧错误一样处理它。

空闲符号

当一空闲帧被检测到时，其处理步骤和接收到普通数据帧一样，但如果 IDLEIE 位为 1 将产生一个中断请求。

溢出错误

如果 RXNE 还没有被复位，而又接收到了一个字符，则发生溢出错误。 数据只有当 RXNE 位被清零后才能从移位寄存器转移到 RDR 寄存器。

RXNE 标记是接收到每个字节后被置 1 的。 如果下一个数据已被收到或先前 DMA 请求还没被服务时，RXNE 标志仍是置起的，也会产生溢出错误。 当溢出错误产生时：

- ORE 位被置 1。
- RDR 内容将不会丢失。 读 USART_RDR 寄存器仍能得到先前的数据。
- 移位寄存器中以前的内容将被覆盖。 随后接收到的数据都将丢失。
- 如果 RXNEIE 位被置为 1 或 EIE 位被置为 1，会产生中断。
- ORE 位可以通过将 ICR 寄存器中的 ORECF 位置 1 的方式清除。

注： *当 ORE 位置 1 时，表明至少有 1 个数据已经丢失。有两种可能性：*

- *如果 RXNE=1，尚一个有效数据还在接收寄存器 RDR 上，可以被读出。*
- *如果 RXNE=0，这意味着上一个有效数据已经被读走，RDR 已经没有东西可读。 当上一个有效数据在 RDR 中被读取的同时又接收到新的 (也就是丢失的) 数据时，此种情况是可能发生的。*

选择时钟源和适当的过采样率的方法

时钟源的选择要通过时钟控制系统（参见第七章： 复位和时钟控制系统（RCC）81 页）。必须在使能 USART 之前（将 UE 位置 1）从而打开 USART 的时钟源。

时钟源的选择需要依据两个标准：

- 在低功耗模式下使用串口的可能性
- 通讯速度

时钟源的频率是 fCK。

当使用双时钟域和从 Stop 模式唤醒的功能时，时钟源可以是下列时钟源中的一个。 fPCLK (default), fLSE, fHSI or fSYS. 除此之外，USART 的时钟源是 fPCLK。

选择 fLSE, fHSI 作为时钟源，可以使得 USART 在 MCU 处于低功耗状态的时候还能接受数据。基于唤醒模式选择和接收到的数据，USART 会将 MCU 唤醒，并由软件或者 DMA 将收到的数据从 USART_RDR 寄存器中读走。

如果用其它的时钟源，必须先将其打开以保证 USART 通讯。

通讯速度范围（特别是最大通讯速度）是由所选的时钟源来决定的。

接收器根据用户设定的过采样率来执行数据恢复，从而将接收数据和噪声区分开来。如果设成同步模式则不会是这样。 这需要在最大通讯速度和噪声抗扰度及对波特率偏差的敏感度之间做取舍。

通过 USART_CR1 寄存器中的 OVER8 位来选择过采样率是波特率时钟的 8 倍或者是 16 倍（图 231 和图 232）。

根据应用：

- 选择 8 倍过采样以实现较高速度（高至 $f_{CK}/8$）。 这种情况下最大的接收器允许的波特率偏差要小一些（参见章节 24.5.5： USART 对时钟偏差的容忍度）
- 选择 16 倍过采样率（OVER8=0）可提高时钟偏差容忍度。 这种情况下，最高通讯速度就会被限制在 $f_{CK}/16$，其中，f_{CK} 就是时钟源的频率。

USART_CR3 中的 ONEBIT 位用来选择判断逻辑电平的方法。 有两种选择：

- 根据接收数据位中央的三个采样结果进行多数票决。 这种情况下，当发现三个参与票决的采样结果不等时，NF 位会被置 1
- 根据接收数据位中央的单个置采样结果来直接裁决
  根据应用：
  - 在噪声干扰环境中应该选择三个采样多数票决的方式，可以排除检测到噪声的数据，因为那说明在数据采样的时候有干扰。
  - 在不担心噪声的情况下，应选择单采样裁决（ONEBIT=1），可以提高接收器对时钟偏差的容忍度（参见章节 24.5.5：USART 对时钟偏差的容忍度） 这种情况下 NF 位永远都不会置 1。

当一帧数据中检测到噪声时：

- 在 RXNE 位的上升沿，NF 位会被置 1.
- 有问题的数据仍会从移位寄存器转移到 USART_RDR 寄存器。
- 单字节通讯的时候不会产生中断。 然而与这一位同时上升的 RXNE 是会产生中断的。 在多缓冲区通讯的情况下，只要 USART_CR3 中的 EIE 位是高就会产生中断。

将 ICR 寄存器中的 NFCF 位置 1 就可以清除 NF 标志。

注： 在智能卡、IrDA 和 LIN 模式下，不能用 8 倍过采样的方式。 在那些模式下，OVER8 位会由硬件强制为 0。

图 231. 过采样率为 16 的时候的数据采样

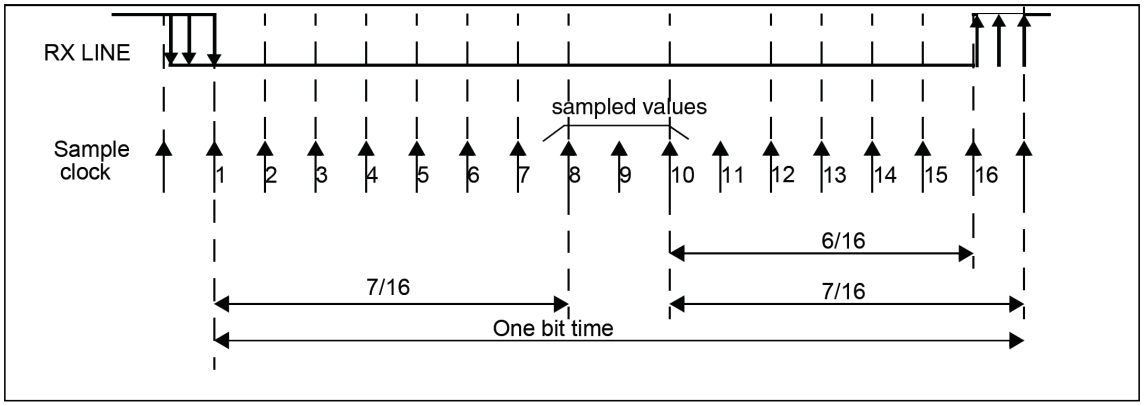


图 232. 过采样率为 8 的时候的数据采样

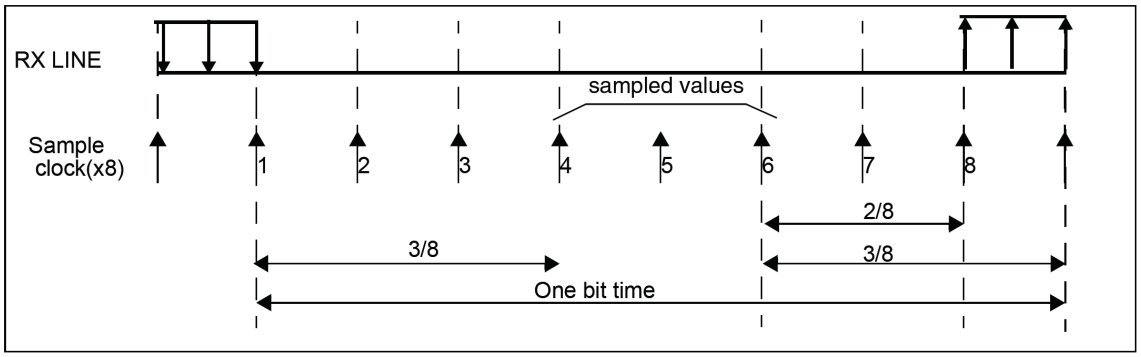


表 82. 从采样数据中检测噪声

采样值	NF 状态	接收到的位值
000	0	0
001	1	0
010	1	0
011	1	1
100	1	0
101	1	1

表 82. 从采样数据中检测噪声

采样值	NF 状态	接收到的位值
110	1	1
111	0	1

帧错误

当以下情况发生时检测到帧错误：

由于没有同步上或大量噪音的原因，停止位没有在预期的时间上接收和识别出来。

当帧错误被检测到时：

- FE 位被硬件置 1
- 有问题的数据仍会从移位寄存器转移到 USART_RDR 寄存器。
- 单字节通讯的时候不会产生中断。 然而与这一位同时上升的 RXNE 是会产生中断的。 在多缓冲区通讯的情况下，只要 USART_CR3 中的 EIE 位是高就会产生中断。

将 USART_ICR 寄存器中的 FECF 置 1 就可以清除 FE 标志。

接收期间的可配置的停止位

被接收的停止位的个数可以通过 USART_CR2 的控制位来配置，在正常模式时，可以是 1 或 2 个，在智能卡模式里可能是 1.5 个。

1. 1 个停止位： 对 1 个停止位的采样在第 8，第 9 和第 10 采样点上进行。
2. 1.5 个停止位 (智能卡模式)： 当以智能卡模式发送时，器件必须检查数据是否被正确的发送出去。 所以接收器功能块必须被激活 (USART_CR1 寄存器中的 RE =1)，并且在停止位的发送期间采样数据线上的信号。 如果出现校验错误，智能卡会在发送方采样 NACK 信号时，即总线上停止位对应的时间内时，拉低数据线，以此表示出现了帧错误。FE 在 1.5 个停止位结束时和 RXNE 一起被置 1。 对 1.5 个停止位的采样是在第 16，第 17 和第 18 采样点进行的。 1.5 个的停止位可以被分成 2 部分： 一个是 0.5 个时钟周期，期间不做任何事情 。随后是 1 个时钟周期的停止位，在这段时间的中点处采样。 参见章节 24.5.13
3. 2 个停止位： 对 2 个停止位的采样是在第一停止位的第 8，第 9 和第 10 个采样点完成的。如果第一个停止位期间检测到一个帧错误，帧错误标志将被设置。 第二个停止位不再检查帧错误。 在第一个停止位结束时 RXNE 标志将被设置。

24.5.4 分数波特率的产生

接收器和发送器的波特率在 USARTDIV 的整数和小数寄存器中的值应设置成相同。

公式 1： 标准 USART（包括 SPI 模式）的波特率

$$\text{Tx/Rx baud} = \frac{f_{CK}}{8 \times (2 - \text{OVER8}) \times \text{USARTDIV}}$$

公式 2： 智能卡模式、LIN 模式和 IrDA 模式的波特率

$$\text{Tx/Rx baud} = \frac{f_{CK}}{16 \times \text{USARTDIV}}$$

USARTDIV 是一个无符号的定点数。这 12 位的值设置在 USART_BRR 寄存器。

- 当 OVER8=0，小数部分按 4 位计算，设置在 USART_BRR 寄存器的 DIV_fraction[3:0] 中
- 当 OVER8=1 时， 小 数 部 分 只 有 3 位 有 效， 设 置 在 USART_BRR 寄 存 器 的 DIV_fraction[2:0] 中，DIV_fraction[3] 必须保持为 0。

注： 在写入 USART_BRR 之后，波特率计数器会立即被波特率寄存器的新值替换。 因此，不要在通信过程中改变波特率寄存器的数值。

OVER8=0 时如何从 USART_BRR 寄存器值得到 USARTDIV

例 1：
如果 DIV_Mantissa = 27 DIV_Fraction = 12 (USART_BRR = 0x1BC), 那么
Mantissa (USARTDIV) = 27
Fraction (USARTDIV) = 12/16 = 0.75
因此 USARTDIV = 27.75

例 2：
要想 USARTDIV = 25.62
那么：
DIV_Fraction = 16*0.62 = 9.92
最接近的整数是 10=0x0A DIV_Mantissa = mantissa (25.620) = 25 = 0x19
所以，USART_BRR = 0x19A 对应的 USARTDIV = 25.625

例 3：
要想 USARTDIV = 50.99
那么：
DIV_Fraction = 16*0.99 = 15.84
最接近的整数是 16=0x10=>DIV_frac[3:0] 溢出 => 进位必须加到整数部分
DIV_Mantissa = mantissa (50.990 + 进位) = 51 = 0x33
于是：USART_BRR = 0x330，USARTDIV=51

OVER8=1 时如何从 USART_BRR 寄存器值得到 USARTDIV

例 1：
如果 DIV_Mantissa = 0x27 and DIV_Fraction[2:0]= 6 (USART_BRR = 0x1B6), 所以
Mantissa (USARTDIV) = 27
Fraction (USARTDIV) = 6/8 = 0.75
因此 USARTDIV = 27.75

例 2：
要想 USARTDIV = 25.62
那么：
DIV_Fraction = 8*0.62 = 4.96
最接近的整数是 5=0x05
DIV_Mantissa = mantissa (25.620) = 25 = 0x19
于是，USART_BRR = 0x195 => USARTDIV = 25.625

例 3：
要想 USARTDIV = 50.99
那么：
DIV_Fraction = 8*0.99 = 7.92
最接近的整数是 8=0x08=>DIV_frac[2:0] 溢出 => 进位必须加到整数部分
DIV_Mantissa = mantissa (50.990 + 进位) = 51 = 0x33
于是，USART_BRR = 0x0330 => USARTDIV = 51.000

表 83. f_{CK}=8MHz 或 f_{PL}=12MHz, 过采样率为 16 时设置波特率时的误差计算

过采样率为 16（OVER8=0）							
波特率		f_{PCLK} = 8 MHz			f_{PCLK} = 12 MHz		
序号	目标	实际	波特率寄存器中的值	误差 %	实际	波特率寄存器中的值	误差 %
1	1.2 KBps	1.2 KBps	416.6875	0	1.2 KBps	625	0
2	2.4 KBps	2.4 KBps	208.3125	0.01	2.4 KBps	312.5	0
3	9.6 KBps	9.604 KBps	52.0625	0.04	9.6 KBps	78.125	0
4	19.2 KBps	19.185 KBps	26.0625	0.08	19.2 KBps	39.0625	0
5	38.4 KBps	38.462 KBps	13	0.16	38.339 KBps	19.5625	0.16
6	57.6 KBps	57.554 KBps	8.6875	0.08	57.692 KBps	13	0.16
7	115.2 KBps	115.942 KBps	4.3125	0.64	115.385 KBps	6.5	0.16
8	230.4 KBps	228.571 KBps	2.1875	0.79	230.769 KBps	3.25	0.16

表 83. f_{CK}=8MHz 或 f_{PL}=12MHz 过采样率为 16 时设置波特率时的误差计算（续）

过采样率为 16（OVER8=0）							
波特率		f_{PCLK} = 8 MHz			f_{PCLK} = 12 MHz		
序号	目标	实际	波特率寄存器中的值	误差 %	实际	波特率寄存器中的值	误差 %
9	460.8 KBps	470.588 KBps	1.0625	2.12	461.538 KBps	1.625	0.16
10	921.6 KBps	NA	NA	NA	NA	NA	NA
11	2 MBps	NA	NA	NA	NA	NA	NA
12	3 MBps	NA	NA	NA	NA	NA	NA

1. CPU 的时钟频率越低，则某一特定波特率的精度也越低。 可以达到的波特率上限可以由这组数据得到。

表 84. f_{CK}=8MHz 或 f_{PL}=12MHz 过采样率为 8 时 设置波特率时的误差计算

过采样率为 8（OVER8=1）							
波特率		f_{PCLK} = 8 MHz			f_{PCLK} = 12 MHz		
序号	目标	实际	波特率寄存器中的值	误差 %	实际	波特率寄存器中的值	误差 %
1	1.2 KBps	1.2 KBps	833.375	0	1.2 KBps	1250	0
2	2.4 KBps	2.4 KBps	416.625	0.01	2.4 KBps	625	0
3	9.6 KBps	9.604 KBps	104.125	0.04	9.6 KBps	156.25	0
4	19.2 KBps	19.185 KBps	52.125	0.08	19.2 KBps	78.125	0
5	38.4 KBps	38.462 KBps	26	0.16	38.339 KBps	39.125	0.16
6	57.6 KBps	57.554 KBps	17.375	0.08	57.692 KBps	26	0.16
7	115.2 KBps	115.942 KBps	8.625	0.64	115.385 KBps	13	0.16
8	230.4 KBps	228.571 KBps	4.375	0.79	230.769 KBps	6.5	0.16
9	460.8 KBps	470.588 KBps	2.125	2.12	461.538 KBps	3.25	0.16
10	921.6 KBps	888.889 KBps	1.125	3.55	923.077 KBps	1.625	0.16
11	2 MBps	NA	NA	NA	NA	NA	NA
12	3 MBps	NA	NA	NA	NA	NA	NA

1. CPU 的时钟频率越低，则某一特定波特率的精度也越低。 可以达到的波特率上限可以由这组数据得到。

表 85. f_{CK}=16MHz 或 f_{PL}=24MHz, 过采样率为 16 时设置波特率时的误差计算

过采样率为 16（OVER8=0）							
波特率		f_{PCLK} = 16 MHz			f_{PCLK} = 24 MHz		
序号	目标	实际	波特率寄存器中的值	误差 %	实际	波特率寄存器中的值	误差 %
1	1.2 KBps	1.2 KBps	833.3125	0	1.2	1250	0
2	2.4 KBps	2.4 KBps	416.6875	0	2.4	625	0
3	9.6 KBps	9.598 KBps	104.1875	0.02	9.6	156.25	0
4	19.2 KBps	19.208 KBps	52.0625	0.04	19.2	78.125	0
5	38.4 KBps	38.369 KBps	26.0625	0.08	38.4	39.0625	0
6	57.6 KBps	57.554 KBps	17.375	0.08	57.554	26.0625	0.08
7	115.2 KBps	115.108 KBps	8.6875	0.08	115.385	13	0.16
8	230.4 KBps	231.884 KBps	4.3125	0.64	230.769	6.5	0.16
9	460.8 KBps	457.143 KBps	2.1875	0.79	461.538	3.25	0.16
10	921.6 KBps	941.176 KBps	1.0625	2.12	923.077	1.625	0.16
11	2 MBps	NA	NA	NA	NA	NA	NA
12	3 MBps	NA	NA	NA	NA	NA	NA

1. CPU 的时钟频率越低，则某一特定波特率的精度也越低。 可以达到的波特率上限可以由这组数据得到。

表 86. f_{CK}=16MHz 或 f_{PL}=24MHz, 过采样率为 8 时设置波特率时的误差计算

过采样率为 8（OVER8=1）							
波特率		f_{PCLK} = 16 MHz			f_{PCLK} = 24 MHz		
序号	目标	实际	波特率寄存器中的值	误差 %	实际	波特率寄存器中的值	误差 %
1	1.2 KBps	1.2 KBps	1666.625	0	1.2 KBps	2500	0
2	2.4 KBps	2.4 KBps	833.375	0	2.4 KBps	1250	0
3	9.6 KBps	9.598 KBps	208.375	0.02	9.6 KBps	312.5	0
4	19.2 KBps	19.208 KBps	104.125	0.04	19.2 KBps	156.25	0
5	38.4 KBps	38.369 KBps	52.125	0.08	38.4 KBps	78.125	0
6	57.6 KBps	57.554 KBps	34.75	0.08	57.554 KBps	52.125	0.08
7	115.2 KBps	115.108 KBps	17.375	0.08	115.385 KBps	26	0.16
8	230.4 KBps	231.884 KBps	8.625	0.64	230.769 KBps	13	0.16

表 86. f_{CK}=16MHz 或 f_{PL}=24MHz, 过采样率为 8 时设置波特率时的误差计算（续）

过采样率为 8（OVER8=1）							
波特率		f_{PCLK} = 16 MHz			f_{PCLK} = 24 MHz		
序号	目标	实际	波特率寄存器中的值	误差 %	实际	波特率寄存器中的值	误差
9	460.8 KBps	457.143 KBps	4.375	0.79	461.538 KBps	6.5	0.16
10	921.6 KBps	941.176 KBps	2.125	2.12	923.077 KBps	3.25	0.16
11	2 MBps	2000 KBps	1	0	2000 KBps	1.5	0
12	3 MBps	NA	NA	NA	3000 KBps	1	0

1. CPU 的时钟频率越低，则某一特定波特率的精度也越低。 可以达到的波特率上限可以由这组数据得到。

表 87. f_{CK}=1MHz 或 f_{PL}=8MHz, 过采样率为 16 时设置波特率时的误差计算

过采样率为 16（OVER8=0）							
波特率		f_{PCLK} = 1 MHz			f_{PCLK} = 8 MHz		
序号	目标	实际	波特率寄存器中的值	误差 %	实际	波特率寄存器中的值	误差 %
1	1.2 KBps	1.2 KBps	52.0625	0.04	1.2 KBps	416.6875	0
2	2.4 KBps	2.398 KBps	26.0625	0.08	2.4 KBps	208.3125	0.01
3	9.6 KBps	9.615 KBps	6.5	0.16	9.604 KBps	52.0625	0.04
4	19.2 KBps	19.231 KBps	3.25	0.16	19.185 KBps	26.0625	0.08
5	38.4 KBps	38.462 KBps	1.625	0.16	38.462 KBps	13	0.16
6	57.6 KBps	58.824 KBps	1.0625	2.12	57.554 KBps	8.6875	0.08
7	115.2 KBps	NA	NA	NA	115.942 KBps	4.3125	0.64
8	230.4 KBps	NA	NA	NA	228.571 KBps	2.1875	0.79
9	460.8 KBps	NA	NA	NA	470.588 KBps	1.0625	2.12
10	921.6 KBps	NA	NA	NA	NA	NA	NA
11	2 MBps	NA	NA	NA	NA	NA	NA
12	4 MBps	NA	NA	NA	NA	NA	NA

1. CPU 的时钟频率越低，则某一特定波特率的精度也越低。 可以达到的波特率上限可以由这组数据得到。

表 88. f_{CK}=1MHz 或 f_{PL}=8MHz, 过采样率为 8 时 设置波特率时的误差计算

过采样率为 8（OVER8=1）							
波特率		f_{PCLK} = 1 MHz			f_{PCLK} = 8 MHz		
序号	目标	实际	波特率寄存器中的值	误差 %	实际	波特率寄存器中的值	误差 %
1	1.2 KBps	1.2 KBps	104.125	0.04	1.2 KBps	833.375	0
2	2.4 KBps	2.398 KBps	52.125	0.08	2.4 KBps	416.625	0.01
3	9.6 KBps	9.615 KBps	13	-0.16	9.604 KBps	104.125	0.04
4	19.2 KBps	19.231 KBps	6.5	0.16	19.185 KBps	52.125	0.08
5	38.4 KBps	38.462 KBps	3.25	0.16	38.462 KBps	26	0.16
6	57.6 KBps	58.824 KBps	2.125	2.12	57.554 KBps	17.375	0.08
7	115.2 KBps	111.111 KBps	1.125	3.55	115.942 KBps	8.625	0.64
8	230.4 KBps	NA	NA	NA	228.571 KBps	4.375	0.79
9	460.8 KBps	NA	NA	NA	470.588 KBps	2.125	2.12
10	921.6 KBps	NA	NA	NA	888.889 KBps	1.125	3.55
11	2 MBps	NA	NA	NA	NA	NA	NA
12	4 MBps	NA	NA	NA	NA	NA	NA

1. CPU 的时钟频率越低，则某一特定波特率的精度也越低。 可以达到的波特率上限可以由这组数据得到。

表 89. f_{CK}=16MHz 或 f_{PL}=32MHz, 过采样率为 16 时 设置波特率时的误差计算

过采样率为 16（OVER8=0）							
波特率		f_{PCLK} = 16 MHz			f_{PCLK} = 32 MHz		
序号	目标	实际	波特率寄存器中的值	误差 %	实际	波特率寄存器中的值	误差 %
1	1.2 KBps	1.2 KBps	833.3125	0	1.2 KBps	1666.6875	0
2	2.4 KBps	2.4 KBps	416.6875	0	2.4 KBps	833.3125	0
3	9.6 KBps	9.598 KBps	104.1875	0.02	9.601 KBps	208.3125	0.01
4	19.2 KBps	19.208 KBps	52.0625	0.04	19.196 KBps	104.1875	0.02
5	38.4 KBps	38.369 KBps	26.0625	0.08	38.415 KBps	52.0625	0.04
6	57.6 KBps	57.554 KBps	17.375	0.08	57.554 KBps	34.75	0.08
7	115.2 KBps	115.108 KBps	8.6875	0.08	115.108 KBps	17.375	0.08
8	230.4 KBps	231.884 KBps	4.3125	0.64	230.216 KBps	8.6875	0.08

表 89. f_{CK}=16MHz 或 f_{PL}=32MHz, 过采样率为 16 时 设置波特率时的误差计算（续）

过采样率为 16（OVER8=0）							
波特率		f_{PCLK} = 16 MHz			f_{PCLK} = 32 MHz		
序号	目标	实际	波特率寄存器中的值	误差 %	实际	波特率寄存器中的值	误差 %
9	460.8 KBps	457.143 KBps	2.1875	0.79	463.768 KBps	4.3125	0.64
10	921.6 KBps	941.176 KBps	1.0625	2.12	914.286 KBps	2.1875	0.79
11	2 MBps	NA	NA	NA	2000 KBps	1	0
12	4 MBps	NA	NA	NA	NA	NA	NA

1. CPU 的时钟频率越低，则某一特定波特率的精度也越低。 可以达到的波特率上限可以由这组数据得到。

表 90. f_{CK}=16MHz 或 f_{PL}=32MHz, 过采样率为 8 时 设置波特率时的误差计算

过采样率为 8（OVER8=1）							
波特率		f_{PCLK} = 16 MHz			f_{PCLK} = 32 MHz		
序号	目标	实际	波特率寄存器中的值	误差 %	实际	波特率寄存器中的值	误差 %
1	1.2 KBps	1.2 KBps	1666.625	0	1.2 KBps	3333.375	0
2	2.4 KBps	2.4 KBps	833.375	0	2.4 KBps	1666.625	0
3	9.6 KBps	9.598 KBps	208.375	0.02	9.601 KBps	416.625	0.01
4	19.2 KBps	19.208 KBps	104.125	0.04	19.196 KBps	208.375	0.02
5	38.4 KBps	38.369 KBps	52.125	0.08	38.415 KBps	104.125	0.04
6	57.6 KBps	57.554 KBps	34.75	0.08	57.554 KBps	69.5	0.08
7	115.2 KBps	115.108 KBps	17.375	0.08	115.108 KBps	34.75	0.08
8	230.4 KBps	231.884 KBps	8.625	0.64	230.216 KBps	17.375	0.08
9	460.8 KBps	457.143 KBps	4.375	0.79	463.768 KBps	8.625	0.64
10	921.6 KBps	941.176 KBps	2.125	2.12	914.286 KBps	4.375	0.79
11	2 MBps	2000 KBps	1	0	2000 KBps	2	0
12	4 MBps	NA	NA	NA	4000 KBps	1	0

1. CPU 的时钟频率越低，则某一特定波特率的精度也越低。 可以达到的波特率上限可以由这组数据得到。

24.5.5 USART 接收器对时钟的变化容忍度

只有当整体的时钟系统的变化小于 USART 异步接收器能够容忍的范围，USART 异步接收器才能正常地工作。 影响这些变化的因素有：

- DTRA: 由于发送器误差而产生的变化 (包括发送器端振荡器的变化)
- DQUANT: 接收器端波特率取整所产生的误差
- DREC: 接收器端振荡器的变化
- DTCL: 由于传输线路产生的变化 (通常是由于收发器在由低变高的转换时序，与由高变低转换时序之间的不一致性所造成)。

需要满足：DTRA + DQUANT + DREC + DTCL < USART 接收器的容忍度

对于正常接收数据，USART 接收器的容忍度等于最大能容忍的变化，它依赖于下述选择：

- 由 USART_CR1 寄存器的 M 位定义的 10 或 11 位字符长度
- USART_CR1 寄存器的 OVER8 位定义的，过采样率 8 位 16 位的选择
- 小数波特率的使用
- USART_CR3 寄存器的 ONEBIT 位定义的，是 1 位采样还是 3 位采样

表 91. 当 DIV_Fraction 为 0 时串口接收容忍度

M 位	OVER8 bit = 0		OVER8 bit = 1	
	ONEBIT=0	ONEBIT=1	ONEBIT=0	ONEBIT=1
0	3.75%	4.375%	2.50%	3.75%
1	3.41%	3.97%	2.27%	3.41%

表 92. 当 DIV_Fraction 为非 0 时串口接收容忍度

M 位	OVER8 bit = 0		OVER8 bit = 1	
	ONEBIT=0	ONEBIT=1	ONEBIT=0	ONEBIT=1
0	3.33%	3.88%	2%	3%
1	3.03%	3.53%	1.82%	2.73%

注： 在接收数据格式为 10 位时长（M=0 时）或 11 位时长（M=1 时）时，空闲帧表 91 和表 92 中的数据规范可能有一些轻微的偏差。

24.5.6 自动波特率检测

USART 可以根据接收到的一个字符来检测和自动设置 USART_BRR 寄存器的值。 自动波特率检测在两种情况下很有用：

- 通讯速度不可知的情况下
- 使用低精度时钟源，需要在不测量时钟偏差的条件下纠正波特率的时候。

时钟源的频率必须和预期的波特率保持相对的稳定
（过采样率为 16，并且波特率处于 $f_{CK}/65535$ 和 $f_{CK}/16$ 之间）。

在打开自动波特率检测之前，字符的内容必须先确认。 有两个可能的字符内容，能够通过 USART_CR2 寄存器的 ABRMOD[1:0] 域进行选择。 具体是：

1. 以 1 开头的任何字符。 这种情况下 USART 将测量起始位的长度（下降沿到上升沿）。
2. 以 10xx 开头的任何字符。 这种情况下 USART 将测量第一个数据位的长度。 测量下降沿到下降沿的时长，在小信号摆率的时候可以确保更好的测量精度。

在打开波特率自动检测之前，USART_BRR 寄存器必须先初始化为一个不为零的波特率值。

将 USART_CR2 寄存器中的 ABREN 位置 1，就打开自动波特率检测功能了。 然后 USART 就在 RX 线上等待第一个字符过来。 自动波特率操作结束后，USART_ISR 寄存器中的 ABRF 标志会被硬件置 1.

如果线路噪声严重，不能保证得到的波特率是准确的。 这时 BRR 值可能是错的或者 ABRE 错误标志会被置 1。 在通讯速度超出自动波特率检测范围（位长度不在 16 到 65536 个时钟周期之间）时也会发生这种情况。

RXNE 的中断也会在操作结束的时候产生。

随后，可以将 ABRF 标志清除掉，以重启自动波特率检测。

注： 如果在自动波特率操作期间将 USART 关掉（UE=0），那 BRR 的检测结果可能就不可信了。

24.5.7 多机通讯

可以将多个 USART 连接成一个网络来实现多机通讯。 例如某个 USART 设备可以是主，它的 TX 输出和其他 USART 从设备的 RX 输入相连接； USART 从设备各自的 TX 输出逻辑地与在一起，并且和主设备的 RX 输入相连接。

在多处理器配置中，我们通常希望只有被寻址的接收者才被激活，来接收随后的数据，这样就可以减少由未被寻址的接收器的参与带来的多余的 USART 服务开销。

未被寻址的设备可启用其静默功能进入静默模式。 要使用静默模式功能，USART_CR1 寄存器的 MME 位必须被置 1。

在静默模式里：

- 任何接收状态位都不会被置 1。
- 所有接收中断被禁止。
- USART_CR1 寄存器中的 RWU 位被置 1。 RWU 可以被硬件自动控制或在某个条件下由软件通过 USART_RQR 寄存器的 MMRQ 位写入。

根据 USART_CR1 寄存器中的 WAKE 位状态，USART 可以用二种方法进入或退出静默模式。

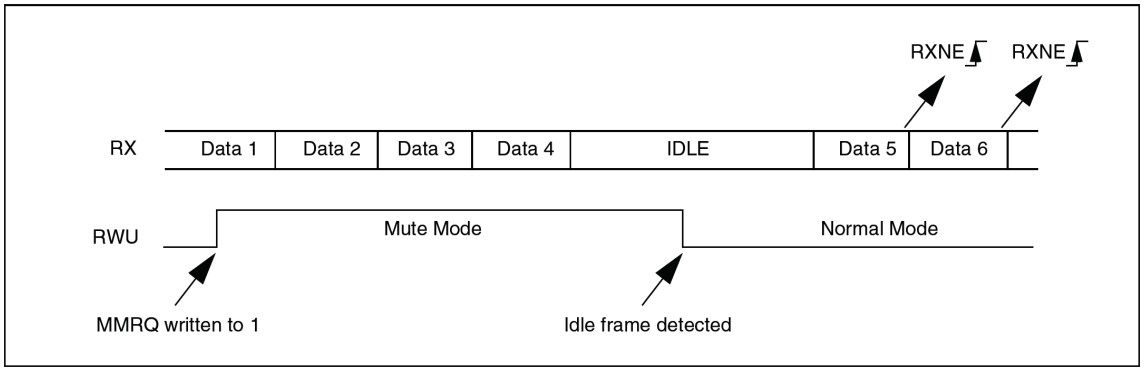
- 如果 WAKE 位为 0：进行空闲总线检测。
- 如果 WAKE 位为 1：进行地址标记检测。

空闲总线检测 (WAKE=0)

当 MMRQ 位被置 1，并且 RWU 被自动置 1 时，USART 进入静默模式。

检测到一个空闲帧的时候会唤醒。 当时 RWU 位被硬件清零但 USART_ISR 寄存器中的 IDEL 位没有被置 1. 图 233 给出了利用空闲总线检测来唤醒和进入静默模式的一个例子。

图 233. 用线路空闲检测从静默模式唤醒



注： 1. 如果在空闲字符已经过去之后才置 1MMRQ，将不会退出静默模式（RWU 没被置 1）。

2. 如果在线路空闲期间激活 USART，在一个空闲帧时长之后才会检测到空闲状态（不仅仅是在收到一个数据帧之后）。

4-bit/7-bit 地址标记检测 (WAKE=1)

在这个模式里，如果 MSB 是 1，该字节被认为是地址，否则被认为是数据。 在一个地址字节中，目标接收器的地址被放在 4 个或 7 个位的位域中。 用 ADDM7 位来选择是用 4 位地址还是用 7 位地址。 这个 4 位或 7 位地址被接收器同它自己地址做比较，接收器的地址被设置在 USART_CR2 寄存器的 ADD 域。

注： 在 7 位和 9 位数据模式下，地址检测分别按 6 位和 8 位地址 (ADD[5:0] 和 ADD[7:0]) 操作。

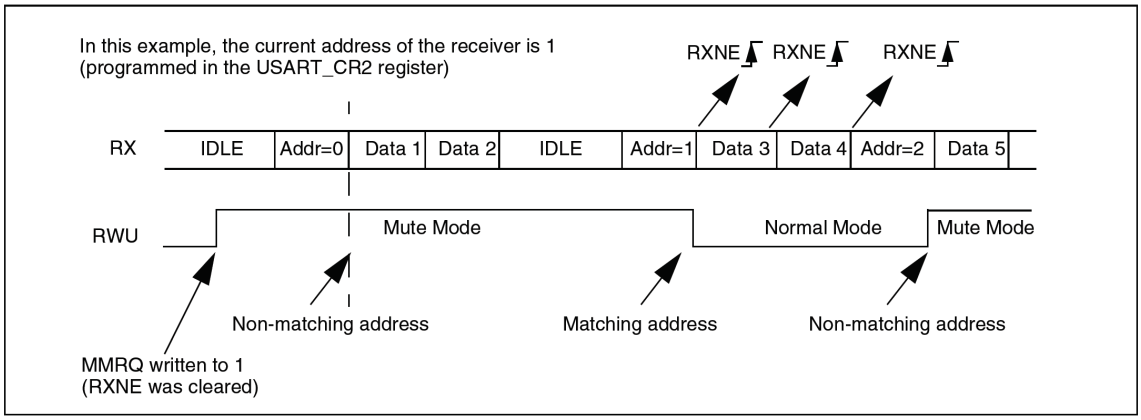
如果接收到的字节与它的编程地址不匹配时，USART 进入静默模式。 此时，硬件将 RWU 位置 1。 当 USART 进到静默模式后，接收字节时既不会动 RXNE 标志也不会产生中断或发出 DMA 请求。

将 MMRQ 置 1 也会令 USART 进入静默模式。 这时 RWU 位也被自动置 1.

如果接收到的字节与它的编程地址匹配，USART 将退出静默模式。 然后 RWU 位被清零，后续的字节会被正常接收。 从 RWU 位被清零开始，RXNE 位会因为地址字节的接收而被置 1.

图 234 给出了利用地址标记检测来退出静默模式的一个例子。

图 234. 利用地址标记检测退出静默模式



24.5.8 Modbus 通讯

USART 提供对 Modbus/RTU 和 Modbus/ASCII 协议实现的基本支持。 Modbus/RTU 是一个半双工，块式传输协议。 协议的控制部分（地址识别，块完整性控制和通信解释）必须由软件来完成。

USART 提供对块尾检测的基本支持，无需软件的经常性介入。

Modbus/RTU

这个模式下，块尾一般公认为一个超过 2 个字符长度的静默阶段。 通过一个可设置的超时长度功能来实现。

超时功能和相应的中断必须通过 USART_CR2 寄存器中的 RTOEN 位和 USART_CR1 寄存器中的 RTOIE 位来打开。 RTO 寄存器中要填入一个与超时长度相当的数字（例如 2 个字符长度为 22 个位长）。当接收线路保持空闲阶段达到这个长度时，在最后一个停止位被收到之后，会产生一个中断，表示当前的块接收已经完毕。

Modbus/ASCII

在这个模式，块尾被公认为回车字符（CR/LF）串。 USART 用字符匹配功能实现这个机制。

将 LF 的 ASCII 码写到 ADD[7:0] 区域，然后打开字符匹配中断（CMIE=1），那么软件就会在收到 LF 字符后或者能够在 DMA 缓冲区中找到 CR/LF 字符时得到提示。

24.5.9 校验控制

设置 USART_CR1 寄存器上的 PCE 位，可以使能校验控制 (发送时生成一个校验位，接收时进行校验检查)。 根据 M 位定义的帧长度，可能的 USART 帧格式列在下表中。

表 93. 帧格式

M 位	PCE 位	USART 帧 (1)
0	0	| SB | 8 bit data | STB |
0	1	| SB | 7-bit data | PB | STB |
1	0	| SB | 9-bit data | STB |
1	1	| SB | 8-bit data PB | STB |

1. SB：起始位，STB： 停止位，PB： 校验位。
2. 在数据寄存器中，PB 总是在最高位（第 8 或者第 7 位，根据 M 位的状态不同）。

偶校验

校验位的内容使得一帧中的 7 或 8 个 LSB 数据以及校验位中’1’的个数为偶数。
例如： 数据 =00110101，有 4 个’1’，如果选择偶校验 (在 USART_CR1 中的 PS ＝ 0)，校验位将是’0’。

奇校验

此校验位的内容使得一帧中的 7 或 8 个 LSB 数据以及校验位中’1’的个数为奇数。
例如： 数据 =00110101，有 4 个’1’，如果选择奇校验 (在 USART_CR1 中的 PS ＝ 1)，校验位将是’1’。

接收时的校验检查

如果校验检查失败，USART_ISR 寄存器中的 PE 标志会被置 1，如果 USART_CR1 寄存器中的 PEIE 为 1，将引发相应中断。 由软件向 USART_ICR 寄存器的 PECF 位写 1，可以清除 PE 标志。

发送时的校验生成

如果 USART_CR1 的 PCE 位被置 1，写进数据寄存器的数据的 MSB 位被校验位替换后发送出去 (如果选择偶校验偶数个’1’，如果选择奇校验奇数个’1’)。

24.5.10 LIN（本地互联网络）模式

本节只针对支持 LIN 模式的部分。 请参见章节 24.4： USART 具体功能配备。

LIN 模式是通过设置 USART_CR2 寄存器的 LINEN 位选择的。 在 LIN 模式下，下列位必须保持为 0：

- USART_CR2 寄存器的 CLKEN 位
- USART_CR3 寄存器的 STOP[1:0]，SCEN，HDSEL 和 IREN

LIN 发送

第 24.5.2 章节的过程解释也适用于 LIN 的主机发送。 和常规的 USART 发送相同，但包含下列区别：

- 清零 M 位以配置 8 位字长
- 置 1LINEN 位以进入 LIN 模式。 这时，置 1SBK 将发送 13 位’0’作为断开符号。 然后发一位’1’，以允许对下一个开始位的检测。

LIN 接收

当 LIN 模式被使能时，断开符号检测电路被激活。 该检测完全独立于 USART 接收器。 不管是在总线空闲时还是在发送某数据帧期间，断开符号只要一出现就能检测到。

一旦接收器被激活 (USART_CR1 的 RE=1)，电路就开始监测 RX 上的起始信号。 监测起始位的方法同检测断开符号或数据是一样的。 当起始位被检测到后，电路对每个接下来的位，在每个位的第 8，9，10 个过采样时钟点上进行采样，就像针对数据一样。 如果 10 个 (当 USART_CR2 的 LBDL = 0) 或 11 个 (当 USART_CR2 的 LBDL = 1) 连续位都是’0’，并且又跟着一个定界符，USART_SR 的 LBD 标志就会被置 1。 如果 LBDIE 位为 1，还会产生中断。在确认断开符号前，要检查定界符，因为它表示 RX 线已经回到高电平。

如果在第 10 或 11 个采样点之前采样到了’1’，检测电路取消当前检测并重新寻找起始位。

如果 LIN 模式被禁止，接收器继续如正常 USART 那样工作，不再考虑检测断开符号。

如果 LIN 模式被激活 (LINEN=1)，只要一发生帧错误 (例如： 停止位检测到’0’，这种情况出现在断开符号被接收到的时候)，接收器就停止，直到断开符号检测电路接收到一个’1’(这种情况发生于断开符号没有完整的发出来)，或一个定界符 (这种情况发生于已经检测到一个完整的断开符号)。

图 235 LIN 模式下断开信号检测（11 位断开长度 -LBDL 位为 1 时， 说明了断开符号检测器状态机的行为和断开符号标志的关系。。

图 236 给出了断开帧的例子： LIN 模式下的断开检测 vs 帧错误检测。

图 235. LIN 模式下断开信号检测（11 位断开长度 -LBDL 位为 1 时）

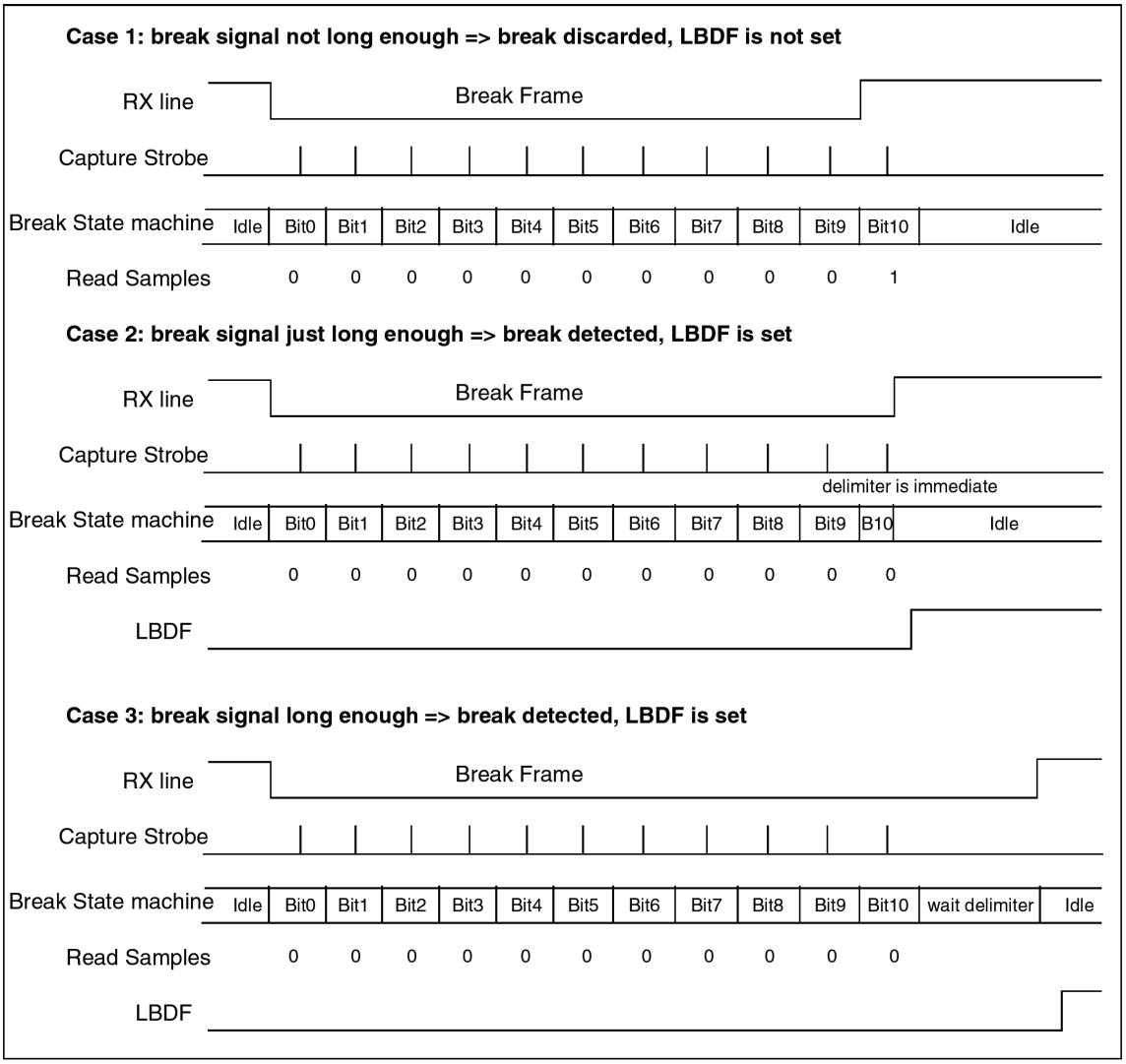
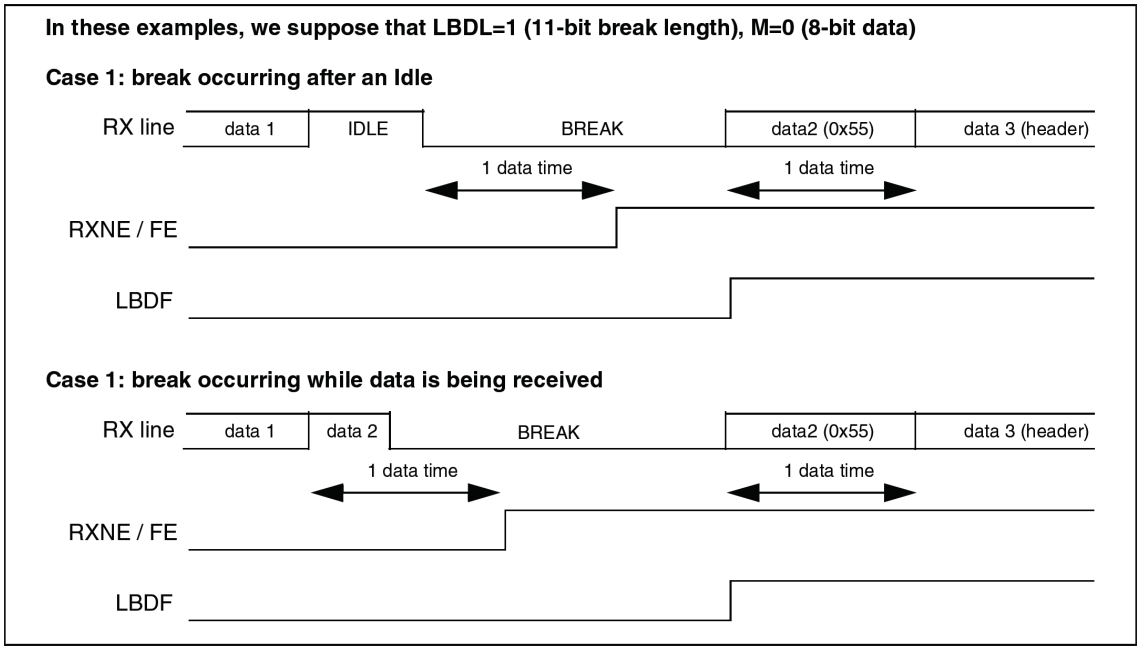


图 236. LIN 模式下的断开检测与帧错误的检测



24.5.11 USART 同步模式

通过在 USART_CR2 寄存器的 CLKEN 位上写 1 来选择同步模式 在同步模式下，下列位必须保持为 0：

- USART_CR2 寄存器中的 LINEN 位
- USART_CR3 寄存器中的 SCEN,HDSEL 和 IREN 位

USART 允许用户以主模式方式控制双向同步串行通信。 SCLK 脚是 USART 发送器时钟的输出。 在起始位和停止位期间，SCLK 脚上没有时钟脉冲。 USART_CR2 寄存器中 LBCL 位的状态决定了在最后一个有效数据位期间产生或不产生时钟脉冲。 USART_CR2 寄存器的 CPOL 位允许用户选择时钟极性，USART_CR2 寄存器上的 CPHA 位允许用户选择外部时钟的相位 (见图 237 图 238 和图 239)。

在总线空闲期间，实际数据到来之前以及发送断开符号的时候，外部 SCLK 时钟不被激活。同步模式时，USART 发送器和异步模式里工作一模一样。 但是因为 SCLK 与 TX 同步 (根据 CPOL 和 CPHA)，所以 TX 上的数据是随 SCLK 同步发出的。

同步模式的 USART 接收器工作方式与异步模式不同。 如果 RE=1，数据在 SCLK 上采样 (根据 CPOL 和 CPHA 决定在上升沿还是下降沿)，不需要任何的过采样。 但必须考虑建立时间和持续时间 (取决于波特率，1/16 位时间)。

注：1. SCLK 脚同 TX 脚是协同工作的。因而，只有在使能了发送器 (TE = 1)，并且发送数据时 (写入数据至 USART_DR 寄存器) 才提供时钟。 这意味着在没有发送数据时是不可能接收一个同步数据的。
2. 应该在发送器和接收器都被禁止时（UE=0）去改变 LBCL,CPOL 和 CPHA 位的配置 . 这能保证时钟脉冲功能的正确性。

图 237. USART 同步发送的例子

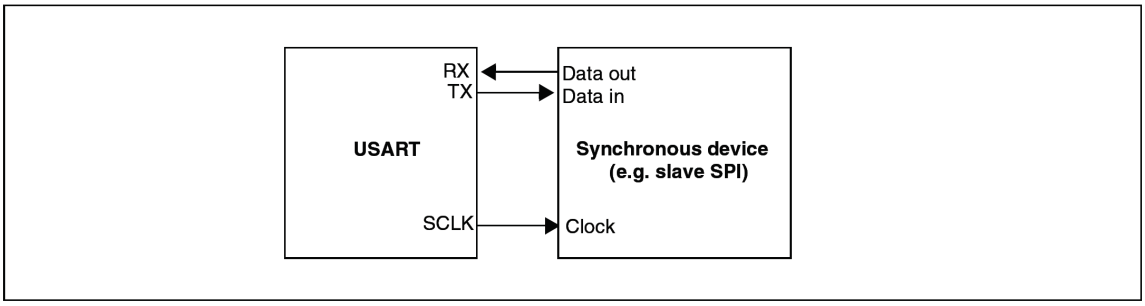


图 238. USART 数据时钟时序示例（M=0）

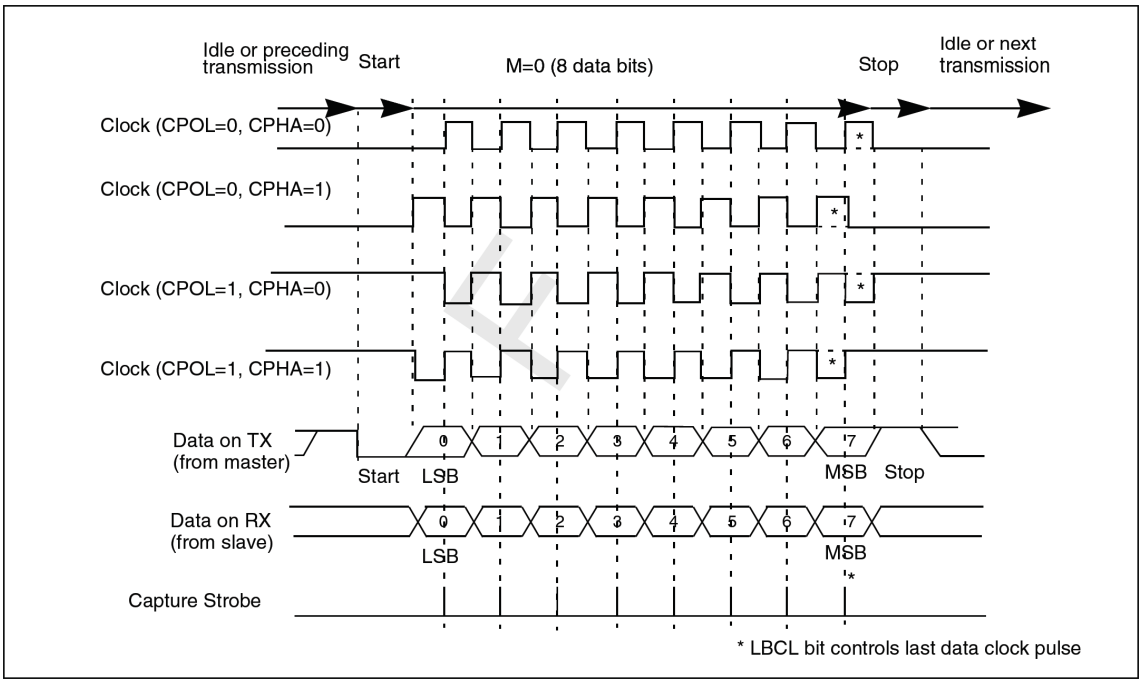


图 239. USART 数据时钟时序示例（M=1）

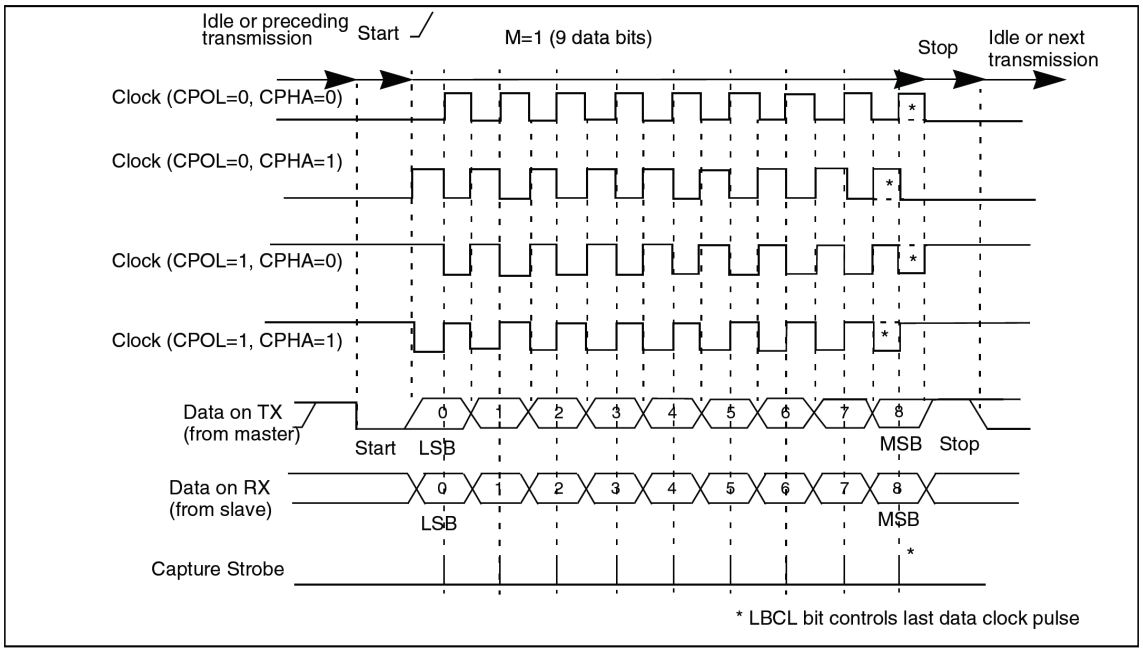
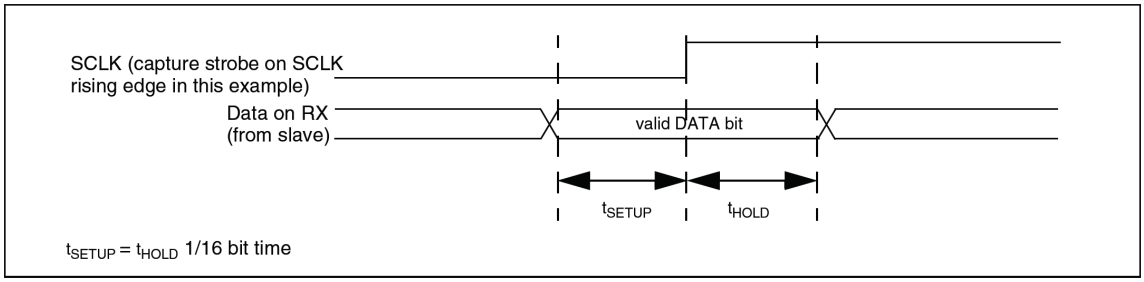


图 240. RX 数据建立 / 保持时间



注： SCLK 的功能和智能卡模式不同。 详情参见章节 24.5.13： 智能卡模式。

24.5.12 单线半双工通讯

单线半双方模式通过设置 USART_CR3 寄存器的 HDSEL 位选择。 在本模式下，下列位必须保持为 0：

- USART_CR2 寄存器的 LINEN 和 CLKEN 位
- USART_CR3 寄存器的 SCEN 和 IREN 位

USART 可以配置成遵循单线半双工协议。在单线半双工模式下，TX 和 RX 引脚在芯片在内部是连在一起的。 使用控制位”HALF DUPLEX SEL”(USART_CR3 中的 HDSEL 位) 选择半双工和全双工通信。

当 HDSEL 为’1’时：

- TX 和 RX 引脚在芯片在内部是连在一起的
- RX 不再被使用
- 当没有数据传输时，TX 总是被释放。 因此，它在空闲状态的或接收状态时表现为一个标准 I/O 口。这就意味该 I/O 在不被 USART 驱动时，必须配置成悬空输入 (或开漏的输出高)。

除此以外，通信与正常 USART 模式类似。 由软件来管理线上的冲突 (例如通过使用一个中央仲裁器)。 特别的是，发送从不会被硬件所阻碍。当 TE 位被设置时，只要数据一写到数据寄存器上，发送就会开始。

24.5.13 智能卡模式

本节只针对支持智能卡模式的部分。 请参见
章节 24.4： USART 具体功能配备

设置 USART_CR3 寄存器的 SCEN 位选择智能卡模式。在智能卡模式下，下列位必须保持为 0:

- USART_CR2 寄存器中的 LINEN 位
- USART_CR3 寄存器的 HDSEL 和 IREN 位

此外，CLKEN 位应该被置 1，以便提供时钟给智能卡。
该接口符合 ISO7816-3 标准，支持智能卡异步协议。T=0 (字符模式) 和 T=1 (块模式) 都支持。

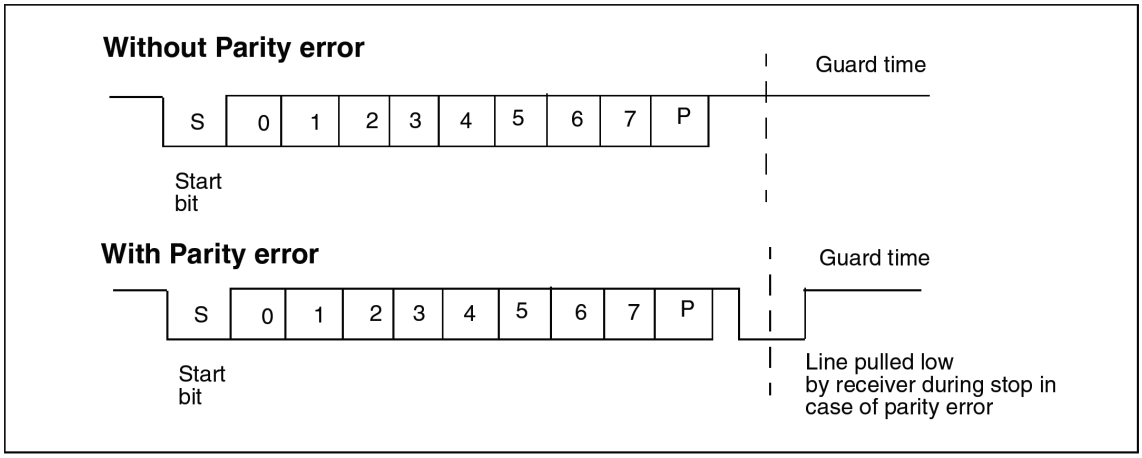
USART 应该被设置为：

- 8 位数据位加校验位： 此时 USART_CR1 寄存器中 M=1、PCE=1
- 1.5 个停止位： 即 USART_CR2 寄存器的 STOP=11

在 T=0（字符）模式中，保护时间内，校验错误在字符发送完毕后被提出。

图 241 所示为在数据线上有校验错误和没有校验错误时的情形。

图 241. ISO 7816-3 异步协议



当连接到智能卡时，USART 的 TX 输出脚和智能卡通过同一根双向数据线进行通讯。 所以 TX 引脚必须配置成开漏状态。

智能卡模式实现一套单线半双工通讯协议。

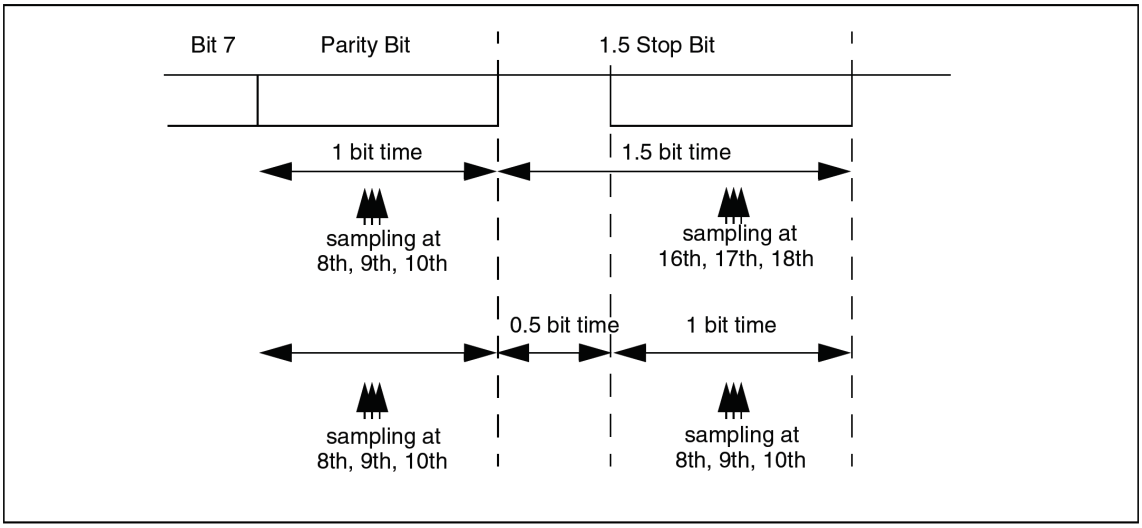
- 发送数据在确认经过最少 1/2 个波特率时钟周期后从发送移位寄存器中移出。 正常操作模式下一个满的发送移位寄存器的内容在下一个波特时钟沿开始移出。 在智能卡模式下则会准确的加入一个 1/2 波特时钟的延迟。

- 发送时，如果智能卡检测到一个校验错误，它会用拉低数据线的方式（NACK）将这个情况提示给 USART。 这个 NACK 信号（拉低传输线 1 个波特时钟）将在发送方（配置为 1.5 个停止位）引起一个帧错误。 USART 能够捕获这个情况并依照协议自动进行重发。 重试的次数可以在 SCARCNT 区域进行设置。 如果 USART 连续收到 NACK 信号，并重试达到设定的次数，USART 会停止发送并提示帧错误。
- 智能卡发送自动重试： USART 在收到 NACK 信号后，会等待 2.5 个波特周期，然后再给出重发字符的起始位。 最后一个重发字符被顺利送达后，TC 位就立即置 1 了，这里没有保护时间的。 如果软件想要再重发，这里必须确保最少 2 个波特周期的延时，这是标准规定的。
- 如果使用 1.5 个停止位，并且在一帧数据的接收过程中检出校验错误，传输线会在一帧传输完毕后被拉低 1 个波特周期。 这是在向智能卡表示，传给 USART 的数据不完全正确。 如果 NACK 控制位为 1，校验错误会导致接收方（USART）发一个 NACK 出来，否则就不发（在 T=1 的模式下）。 如果接收字符不正确，也就不会引起 RXNE 的变化以及接收 DMA 请求。 按照协议规定，智能卡必须再发一个相同的字符。 如果按照 SCARCNT 域的设定重试次数达到了而收到的字符还是错的，USART 就不会再发 NACK 了，而是提示一个校验错误。
- 智能卡接收的自动重试： 如果 USART 向智能卡发了 NACK 但卡却没有重发字符回应，就会使 BUSY 标志残留下来。
- 在发送过程中，USART 会在两次成功的发送字符之间插入一个保护时间（具体长度在保护时间寄存器中设定）。 保护时间长度是从前一个字符的停止位开始计算的，GT[7:0] 区域必须按照所须的 CGT（7816-3 标准要求的保护时间长度）设置一个值，最少是 12（相当于 1 个字符的时间）。
- 可以调整保护时间的设置来延迟 TC 标志的置 1。
  正常操作中，TC 标志是在发送移位寄存器为空，并且也没有进一步的发送要求的是后被置 1 的。 而在智能卡模式下，发送移位寄存器为空的事件只会触发保护时间计数器，令其计数直到所设定的保护时间过完。 在这个过程中，TC 会强制为 0。 当保护时间计数器达到所设定的上限，TC 才会被置 1。
- 对 TC 标志的撤销不受智能卡模式的影响。
- 如果在发送过程的最后检出帧错误（就是从接收方传来一个 NACK），这个 NACK 是不会被当作起始位来检测的。 根据 ISO 协议，接收到的 NACK 的持续时间可以是 1 或 2 个波特时钟周期。
- 在接收器这边，如果检测到一个校验错误，并且已经发送 NACK，接收器也不对把 NACK 当作起始位。

注： *1. 断开字符在智能卡模式中没有意义。一个带帧错误的 0x00 数据将被当成数据而不是断开符号。*
2. 当改变 TE 位时，也不会有空闲帧发出。空闲帧（在其它模式中有定义）在 ISO 协议中是没有的。

图 242 显示 USART 如何采样 NACK 信号 本例中 USART 配置成 1.5 个停止位来发送数据。USART 的接收部分也被使能，用来检查数据完整性以及 NACK 信号。

图 242. 用 1.5 位停止位时检测校验错误



USART 可以通过 SCLK 脚向智能卡提供时钟。 智能卡模式中，SCLK 和通讯没有关系，只是通过一个 5 位的预分频器从内部外设时钟源得到时钟信号。 这个分频系数在 USART_GTPR 寄存器中设置。 SCLK 频率可以设置在 $f_{CK}/2$ 到 $f_{CK}/62$ 之间 ,f_{CK} 指外设输入时钟 .

块模式 (T=1)

在块模式（T=1）中，校验错误是要通过将 USART_CR3 寄存器的 NACK 清零来关闭的。

当要从智能卡读取数据时，在块模式下，软件必须将 RTOR 寄存器设为 BWT（块等待时间）-11 的值。 如果这个时间到期了，还没有收到卡的回应，将会引起超时中断。 如果超时之前收到了第一个字节，则会引起 RXNE 中断。

注： 块模式下，即便是用 USART 的 DMA 模式从智能卡读取数据，也必须使能 RXEN 中断。 同时，也只能在第一个字节收好之后再去使能 DMA。

在收到第一个字节之后（RXNE 中断了），必须将 RTO 寄存器设置成 CWT（字符等待时间）-11 的值，这是为了允许在两个连续的字符之间自动检测最大等待时间。 这个时间以波特时间作为单位。 如果智能卡在前一个字符发送结束后到设定的 CWT 周期之间没有发送新的字符，USART 会通过 RTOF 标志提示给软件，如果 RTOIE 位为 1，则会引起中断。

注： 在 STOP=00，10 的情况下，RTO 计数器从最后一个字符的第一个停止位开始计数。 在 STOP=11 的情况下，RTO 计数器从停止位的开始处就计为 1，并继续计数。 按照智能卡协议的定义，BWT/CWT 值是从最后一个字符的起始位开始计算的。 所以 RTO 寄存器必须为 BWT-11 或者 CWT-11，从而将各自的字符长度也算在内。

有一个块长度计数器可以统计 USART 收到的全部字符的个数。 这个计数器在 USART 开始发送的时候（TXE=0）自动清零。 块长度信息位于智能卡发出数据块的第三字节（序言部分）。这个值必须写入到 USART_RTOR 寄存器的 BLEN 域。 当使用 DMA 模式，从块的开始前，这个寄存器必须设定为最小值（0x0）。 为了得到这个值，在收到第 4 个字节的时候会引起一个中断。 软件必须从接收缓冲区中读取到这个长度域（第三字节）。

在中断驱动接收模式，块的长度可以由软件提取出来并作检查。 在块开始前，BLEN 可以设为最大值（0xFF）。 实际值则要在收到第三字节之后被写到寄存器中。

如果块格式为使用 LRC 纵向冗余校验（1 个校验自己），BLEN 就等于 LEN。 如果块格式使用 CRC 循环冗余校验（2 个校验字节），BLEN 就等于 LEN+1。 整个块长度（包括序言部分，校验和信息区域）等译 BLEN+4。 块尾将通过 EOBF 标志以及相应中断（EOBIE 位被置 1 时）提示给软件。

在块长度出错的情况下，块尾则会引起 RTO 中断（字符等待时间溢出）。

注： （LRC/CRC）校验码的验算需要由软件来完成。

直接和反向转换

智能卡协议定义了两种转换方式： 直接方式和反向方式。

直接转换方式定义如下： 低位在前，逻辑位为 1 相当于传输线高电平，采用偶校验。 要使用这个转换方式，下列控制位必须设置： MSBFIRST=0, DATAINV=0 (默认值).

反向转换方式定义如下： 高位在前，逻辑位为 1 相当于传输线低电平，采用奇校验。 要使用这个转换方式，下列控制位必须设置： MSBFIRST=1, DATAINV=1.

注： 当逻辑数据为反转状态时（0=H，1=L），校验位也是采用同样的逻辑（0 是高，1 是低）。

为了确认卡的转换模式，卡会将初始化字符 TS 作为对 ATR（应答复位）帧的回应。 TS 的两种可能的模式为： LHHL LLL LLH 和 LHHL HHH LLH。

- (H) LHHL LLL LLH 将启用反向转换模式：状态低对应数据 1，第二个时刻包含的信息为数据的最高位。 用反向方式解码时，传达的字节等于 '3F'。
- (H) LHHL HHH LLH 将启用直接转换模式： 状态高对应数据 1，第二个时刻包含的信息为数据的最低位。 用直接方式解码时，传达的字节等于 '3B'。

当在第 2 到第 10 个时刻之间的 9 个时间段中有偶数个 1 时，字符校验正确。

由于 USART 并不知道用该用哪种转换模式，所以它要能够确认转换的样式并且照着执行。这个确认工作不是由硬件完成，而是由软件去判断。 此外，假设 USART 被配置为直接转换模式（默认），而卡的答复是按反向转换的，TS= LHHL LLL LLH => USART 收到的字符会是‘03’ 并且校验规则是奇校验。

因此，有两个方法可以用来确认 TS 的转换样式：

方法 1： USART 被设置为标准智能卡模式，直接转换。 这时，TS 样式确认产生出一个校验错误中断并发送一个错误信号给卡。

- 校验错误中断提示软件说卡在直接转换模式下没有给出正确的应答。 软件来重新设置 USART 为反向转换模式。
- 收到错误信号后，卡会再次发送一个同样 TS 字符，由于 USART 已经被重新设置过，这次就会被正确的接收了。

这样，在对校验错误中断的回复中，软件可以对 USART 重新设置，从而向卡发出新的复位命令，然后重新等待 TS。

方法 2： USART 被设置为 9 位字长 / 无校验的模式，无反向转换。 在这个情况下能够收到的两种 TS 样式如下：

(H) LHHL LLL LLH = 0x103 -> 选为反向转换模式
(H) LHHL HHH LLH = 0x13B -> 选为直接转换模式

软件检查接收到的字符，来判断该用哪一种转换模式，然后将 USART 设置成正确的模式来适应下一个字符的接收。

如果两个都不符合，那就向卡发送一个复位命令以重启新一轮会话。

24.5.14 IrDA SIR ENDEC 功能模块

本节只针对支持 IrDA 模式的部分。 请参见章节 24.4： USART 具体功能配备。

通过设置 USART_CR3 寄存器的 IREN 位选择 IrDA 模式。在 IrDA 模式下，下列位必须保持为 0:

- USART_CR2 寄存器的 LINEN, STOP 和 CLKEN 位，
- USART_CR3 寄存器的 SCEN 和 HDSEL 位。

IrDA SIR 物理层规定使用反向归零调制方案（RZI），该方案用一个红外光脉冲代表逻辑‘0’（见图 243）。

SIR 发送编码器对 USART 给出的 NRZ 比特流进行调制。 输出脉冲流用来驱动一个外部的输出驱动器和红外 LED。 USART 的 SIR ENDEC 最高只支持到 115.2Kbps 速率。 在正常模式下，脉冲宽度规定为一个位周期的 3/16。

SIR 接收解码器对来自红外接收器的归零位比特流进行解调，并将接收到的 NRZ 串行比特流输出到 USART。 在空闲状态里，解码器输入通常是高 (标记状态)。 发送编码器输出的极性和解码器的输入相反。 当解码器输入低时，检测到一个起始位。

- IrDA 是一个半双工通信协议。如果发送器忙 (也就是 USART 正在送数据给 IrDA 编码器)，IrDA 接收线上的任何数据将被 IrDA 解码器忽视。如果接收器忙 (也就是 USART 正在接收从 IrDA 解码器来的解码数据)，从 USART 到 IrDA 的 TX 上的数据将不会被 IrDA 编码。当接收数据时，应该避免发送，因为将被发送的数据可能被破坏。
- SIR 发送逻辑把’0’作为高脉冲发送，把’1’作为低电平发送。 脉冲的宽度规定为正常模式时位周期的 3/16(见图 244)。
- SIR 解码器把 IrDA 兼容的接收信号转变成给 USART 的比特流。
- SIR 接收逻辑把高电平状态解释为’1’，把低脉冲解释为’0’。
- 发送编码器输出与解码器输入有着相反的极性。 当空闲时，SIR 输出处于低状态。
- IrDA 规范要求脉冲要宽于 1.41us。 脉冲宽度是可编程的。 接收器端的尖峰脉冲检测逻辑滤除宽度小于 2 个 PSC 周期的脉冲 (PSC 是在 IrDA 低功耗波特率寄存器 USART_GTPR 中编程的预分频值)。 宽度小于 1 个 PSC 周期的脉冲一定被滤除掉，但是那些宽度大于 1 个而小于 2 个 PSC 周期的脉冲可能被接收或滤除，那些宽度大于 2 个周期的将被视为一个有效的脉冲。 当 PSC=0 时，IrDA 编码器 / 解码器不工作。
- 接收器可以与一个低功耗发送器通信。
- 在 IrDA 模式里，USART_CR2 寄存器上的 STOP 位必须配置成 1 个停止位。

IrDA 低功耗模式

发送器：

在低功耗模式，脉冲宽度不再持续 3/16 个位周期。 而是原来的三分之一，他的频率最小可以是 1.42MHz。通常这个值是 1.8432MHz(1.42 MHz < PSC < 2.12 MHz)。 一个低功耗模式可编程分频器把系统时钟进行分频以达到这个值。

接收器：

低功耗模式的接收类似于正常模式的接收。 为了滤除尖峰干扰脉冲，USART 应该滤除宽度短于 1 个 PSC 的脉冲。 只有持续时间大于 2 个周期的 IrDA 低功耗波特率时钟 (USART_GTPR 中的 PSC) 的低电平信号才被接受为有效的信号。

注： *1. 宽度小于 2 个大于 1 个 PSC 周期的脉冲可能会也可能不会被滤除。*

2. 接收器的建立时间应该由软件管理。 IrDA 物理层技术规范规定了在发送和接收之间最小要有 10ms 的延时 (IrDA 是一个半双工协议)。

图 243. IrDA SIR ENDEC - 框图

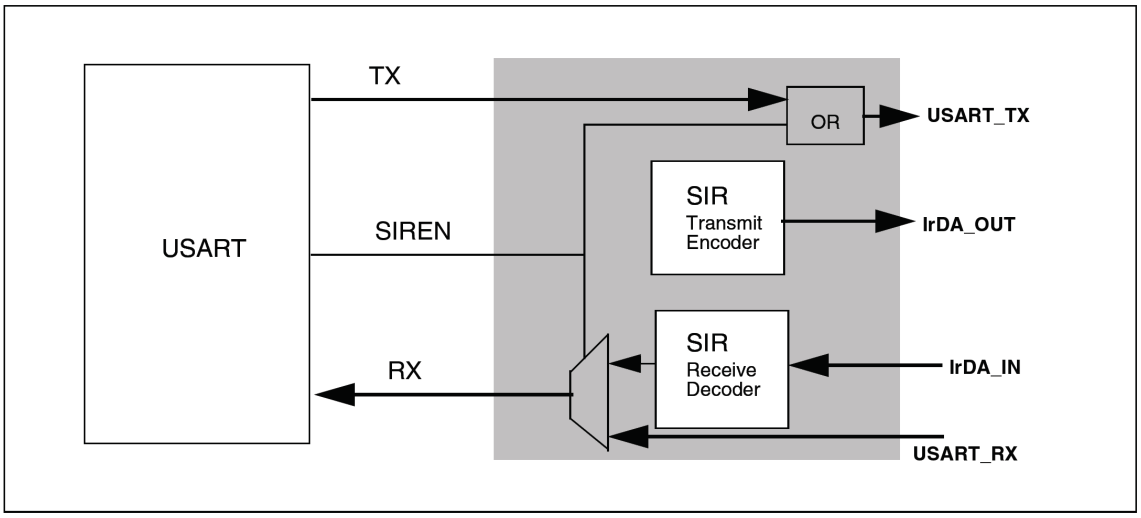
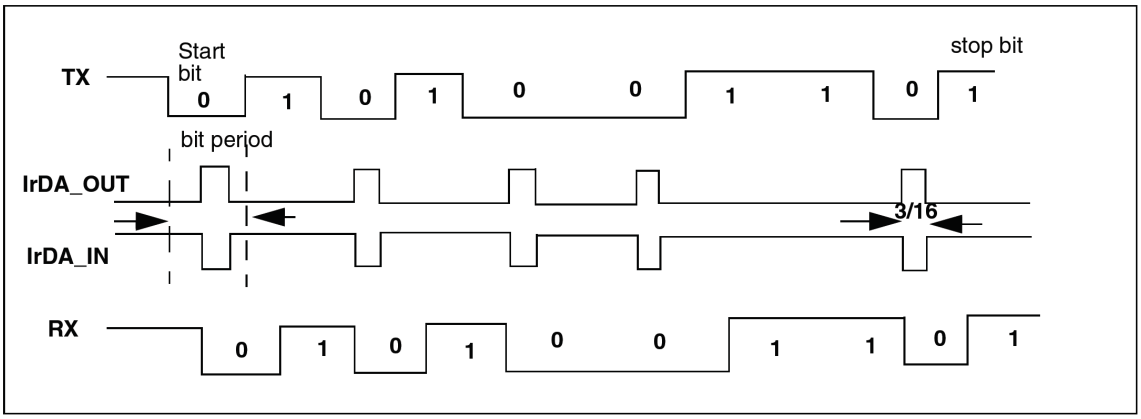


图 244. IrDA 数据调制 (3/16) – 普通模式



24.5.15用 DMA 实现连续通讯

USART 可以利用 DMA 连续通信。Rx 缓冲器和 Tx 缓冲器的 DMA 请求是分别产生的。

注： 请参见章节 24.4：USART 具体功能配备，以确定是否支持 DMA 模式。如果所用产品无 DMA 功能，应按 24.5.2 节或 24.5.3 节里所描述的方法使用 USART。在 USART2_SR 寄存器里，可以清零 TXE/RXNE 标志来实现连续通信。

利用 DMA 发送

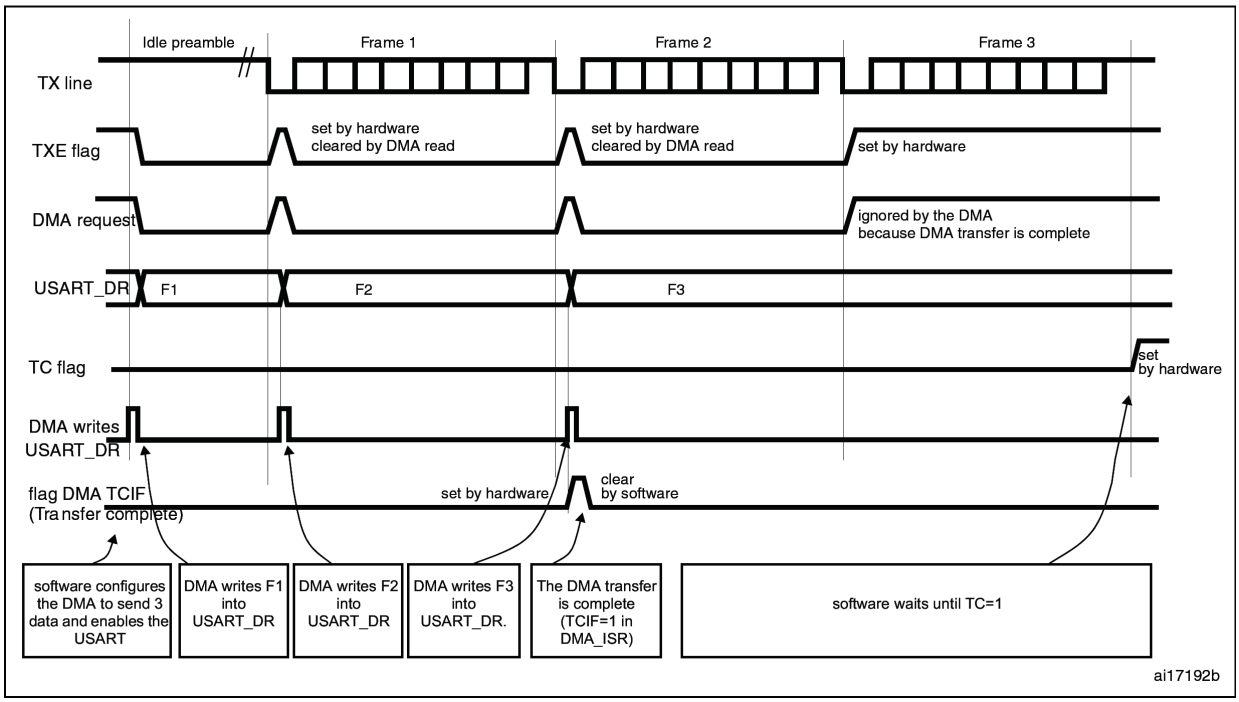
使用 DMA 进行发送，可以通过设置 USART_CR3 寄存器上的 DMAT 位激活。 在设置 TXE 位之前，数据被预先放到 DMA 外设所设定的 SRAM 区域（参见章节 10：直接内存访问控制器）设置一个 DMA 通道给 USART 发送，要使用下列步骤（x 指通道号）：

1. 在 DMA 控制寄存器上将 USART_TDR 寄存器地址配置成 DMA 传输的目的地址。 在每个 TXE 事件后，数据将被传送到这个地址。
2. 在 DMA 控制寄存器上将内存地址配置成 DMA 传输的源地址。 在每个 TXE 事件后，将从此存储器区读出数据并传送到 USART_TDR 寄存器。
3. 在 DMA 控制寄存器中配置要传输的总的字节数。
4. 在 DMA 寄存器上配置通道优先级。
5. 根据应用程序的要求，配置在传输完成一半还是全部完成时产生 DMA 中断。
6. 将 USART_ICR 寄存器的 TCCF 位置 1 以清除 USART_ISR 寄存器的 TC 标志。
7. 在 DMA 寄存器上激活该通道。

当传输完成 DMA 控制器指定的数据量时，DMA 控制器在该 DMA 通道的中断向量上产生一中断。

在发送模式下，当 DMA 传输完所有要发送的数据时，DMA 控制器设置 DMA_ISR 寄存器的 TCIF 标志；监视 USART_SR 寄存器的 TC 标志可以确认 USART 通信是否结束。 这样可以在关闭 USART 或进入停机模式之前避免破坏最后一次传输的数据。 软件必须等待 TC 被置 1。TC 标志在全部数据发送期间会是零，并且在最后一帧数据发送出去之后会由硬件置 1。

图 245. 利用 DMA 发送



利用 DMA 接收

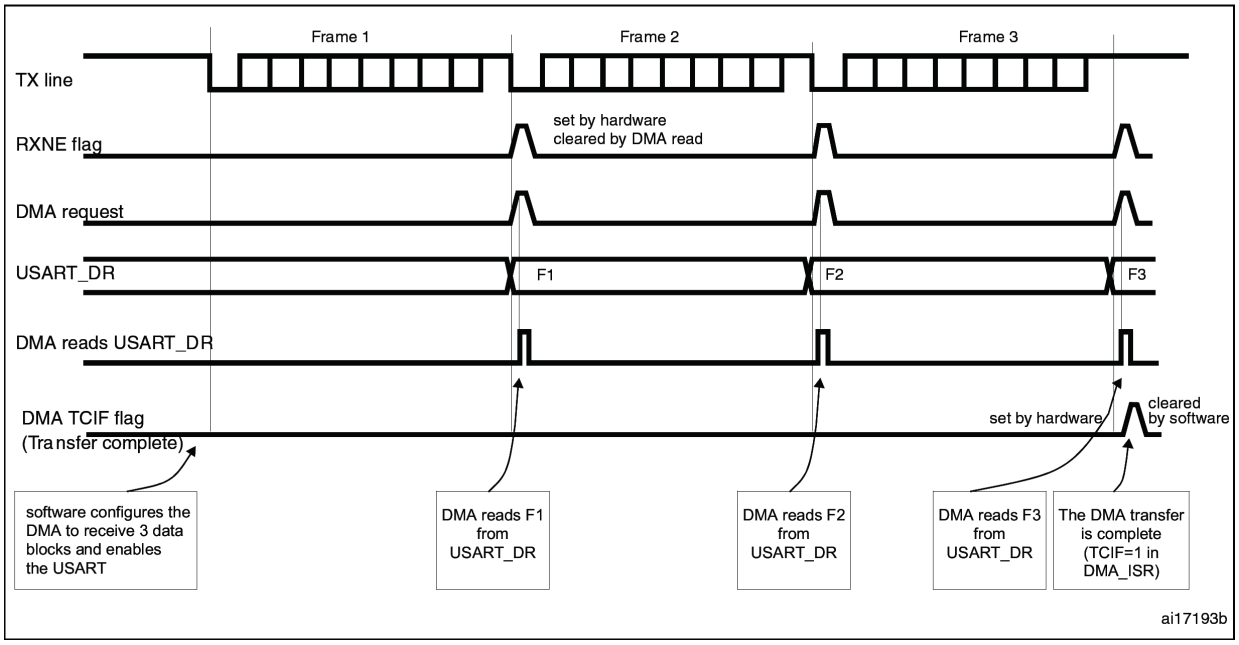
使用 DMA 进行接收，可以通过设置 USART_CR3 寄存器上的 DMAR 位激活。 当收到一个字节数据时，从 USART_RDR 寄存器取出来的数据会被转移到 DMA 外设中指向的 SRAM 区域（参见章节 10： 直接内存访问控制器（DMA））。

将一个 DMA 通道设置给 USART 接收，要按照下列步骤：

1. 在 DMA 控制寄存器上将 USART_RDR 配置成 DMA 传输的源地址。 在每个 RXNE 事件后，数据将从这个地址取走。
2. 在 DMA 控制寄存器上将内存地址配置成 DMA 传输的目标地址。 在每个 RXNE 事件后，数据将从 USART_RDR 取往这个目标地址。
3. 在 DMA 控制寄存器中配置要传输的总的字节数。
4. 在 DMA 寄存器上配置通道优先级。
5. 根据应用程序的要求，配置在传输完成一半还是全部完成时产生 DMA 中断。
6. 在 DMA 寄存器上激活该通道。

当传输完成 DMA 控制器指定的数据量时，DMA 控制器在该 DMA 通道的中断向量上产生一中断。

图 246. 利用 DMA 接收



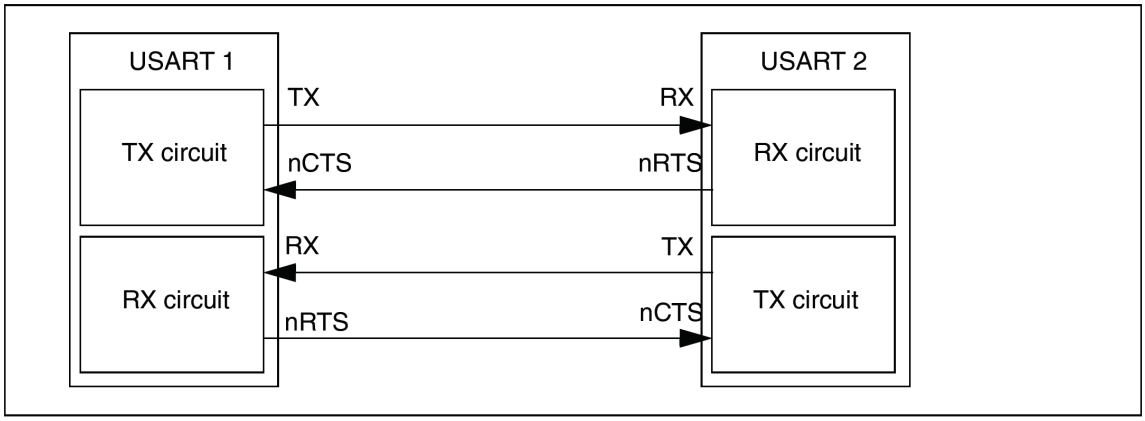
多缓冲器通信中的错误标志和中断产生

在多缓冲器通信的情况下，通信期间如果发生任何错误，会在当前字节传输后将错误标志置 1。如果中断使能位被置 1，将产生中断。 在单个字节接收的情况下，和 RXNE 一起被置起的帧错误、溢出错误和噪音标志，有单独的错误标志中断使能位；如果被置 1 了，会在当前字节传输结束后，产生中断。

24.5.16 硬件流控制和 RS485 驱动使能

利用 nCTS 输入和 nRTS 输出可以控制 2 个设备间的串行数据流。 图 247 显示了这种模式下如何连接两个设备：

图 247. 2 个 USART 之间的硬件流控制

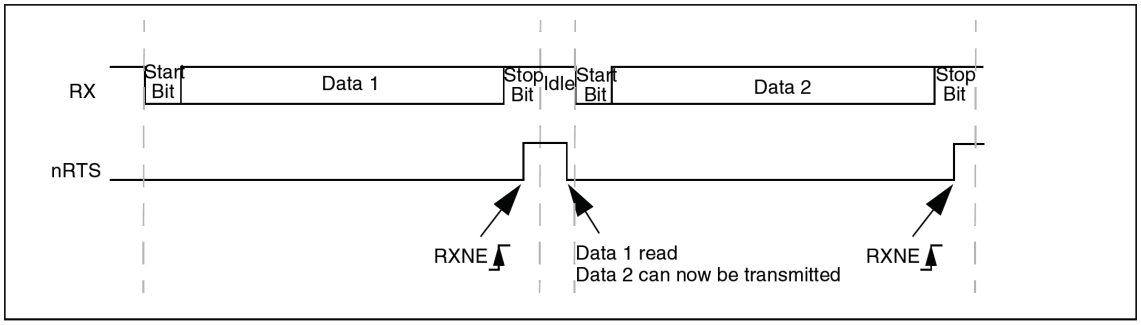


通过将 UASRT_CR3 中的 RTSE 和 CTSE 置 1，可以分别独立地使能 RTS 和 CTS 流控制。

RTS 流控制

如果 RTS 流控制被使能 (RTSE=1)，只要 USART 接收器准备好接收新的数据，nRTS 就变成有效 (接低电平)。 当接收寄存器内的数据未被取走时，nRTS 被释放，由此表明希望在当前帧结束时停止数据传输。图 248 是一个启用 RTS 流控制的通信的例子。

图 248. RTS 流控制

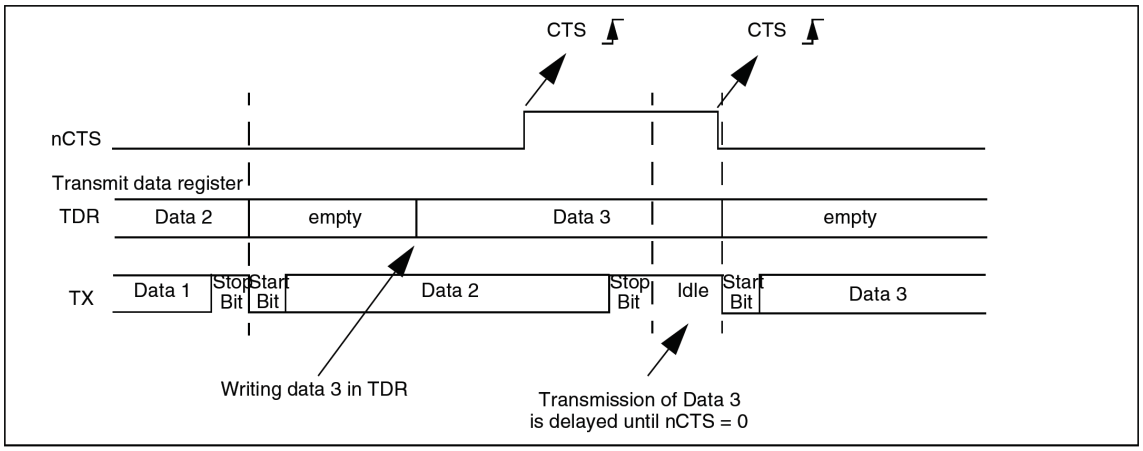


CTS 流控制

如果 CTS 流控制被使能 (CTSE=1)，发送器在发送下一帧前检查 nCTS 输入。 如果 nCTS 有效 (被拉成低电平)，则下一个数据被发送 (假设那个数据是准备发送的，也就是如果 TXE=0)，否则下一帧数据不被发出去。 若 nCTS 在传输期间被变成无效，当前的传输完成后才停止发送。

当 CTSE=1 时，只要 nCTS 输入一变换状态，硬件就自动设置 CTSIF 状态位。 它表明接收器是否准备好进行通信。 如果设置了 USART_CT3 寄存器的 CTSIE 位，则产生中断。
图 249 是一个启用 CTS 流控制的通信的例子。

图 249. CTS 流控制



注： nCTS 必须较当前字符的结束提前至少 3 个 USART 时钟周期做动作。 另外需要注意的是，对于较 2 个 PCLK 周期短的脉冲，CTSCF 标志不一定能够置 1。

RS485 驱动使能

驱动使能功能用 USART_CR3 控制寄存器的 DEM 位置 1 来打开。 它允许用户通过 DE（驱动使能）信号来激活外部收发器的控制端。提前时间的意思是驱动使能信号和第一个字节的起始位之间的时间间隔。 这个时间可以在 USART_CR1 控制寄存器的 DEAT[4:0] 域中设置。 滞后时间是一个发送消息的最后一个字节的停止位和释放 DE 信号之间的时间间隔。 这个时间可以在 USART_CR1 控制寄存器的 DEDT[4:0] 域中设置。 DE 信号的极性则可以通过 USART_CR3 控制寄存器中的 DEP 位进行选择。

24.5.17从 Stop 模式唤醒

本节只针对具备从 Stop 模式唤醒功能的情形。 请参见表 81： STM32F0xx USART 具体功能配备。

在 USART 时钟被设置为 HSI 或者 LSE 的时候，设置了 UESM 位就可以使用 USART 将 MCU 从 Stop 模式唤醒。 第 7 章： 复位和时钟控制系统（RCC）81 页。

可以用标准的 RXNE 中断来干这个事。这时，在进入 Stop 模式之前 RXNEIE 位必须先设置为 1。另外需要在 WUS 位域指定这个中断。
必须在进入 Stop 模式之前先要置 1USART_CR1 中的 UESM 位，否则唤不醒。
当检测到唤醒时间，WUF 表示会被硬件置 1，同时，只要 WUFIE 位是 1，就会立即产生中断。

注： 1. 在进入 Stop 模式之前，软件必须检查 USART 没有正在发数据，这可以通过 USART_SR 寄存器中的 BUSY 标志来判断。
2. 在收到唤醒事件时，WUF 标志肯定会被置 1，这不受 MCU 所处的工作模式影响，无论是在 Stop 模式还是在活动模式，都可以。
3. 当刚刚初始化好接收器的时候就进入 Stop 模式时，需要检查 REACK 位以确定 USART 确实已经被打开了。
4. 当用 DMA 来作接收时，在进入 Stop 模式前必须先禁止掉，在唤醒之后在重新使能它都行。

警告： 不是所有的模式都支持从 Stop 模式唤醒。 例如，在 SPI 模式下就不行，因为 SPI 模式只能工作在主机状态下。

Stop 模式下使用静默功能

如果在进入 Stop 模式之前，USART 已经处于静默状态下：

- 不可以使用基于空闲检测退出静默的方式，因为空闲检测在 Stop 模式下不好使。
- 如果使用地址匹配的方式退出静默状态，唤醒源也就只能是地址匹配事件。
- 如果 USART 被设置为起始位检测就唤醒 MCU，那么 WUF 表示会置 1，但 RXNE 标志就不会。

24.6 USART 中断

表 94. USART 中断请求

中断事件	事件标志	使能控制位
发送数据寄存器空	TXE	TXEIE
CTS 中断	CTSIF	CTSIE
发送完成	TC	TCIE
接收数据寄存器非空（有数据可读）	RXNE	RXNEIE
溢出错误检测	ORE	
空闲线检测	空闲	IDLEIE
奇偶错误	PE	PEIE
LIN 断开	LBDF	LBDIE
噪声标志，溢出错误和多缓冲区通讯中的帧错误	NF or ORE or FE	EIE
字符匹配	CMF	CMIE
接收超时错误	RTOF	RTOIE
发现块尾	EOBF	EOBIE
从 Stop 模式唤醒	WUF(1)	WUFIE

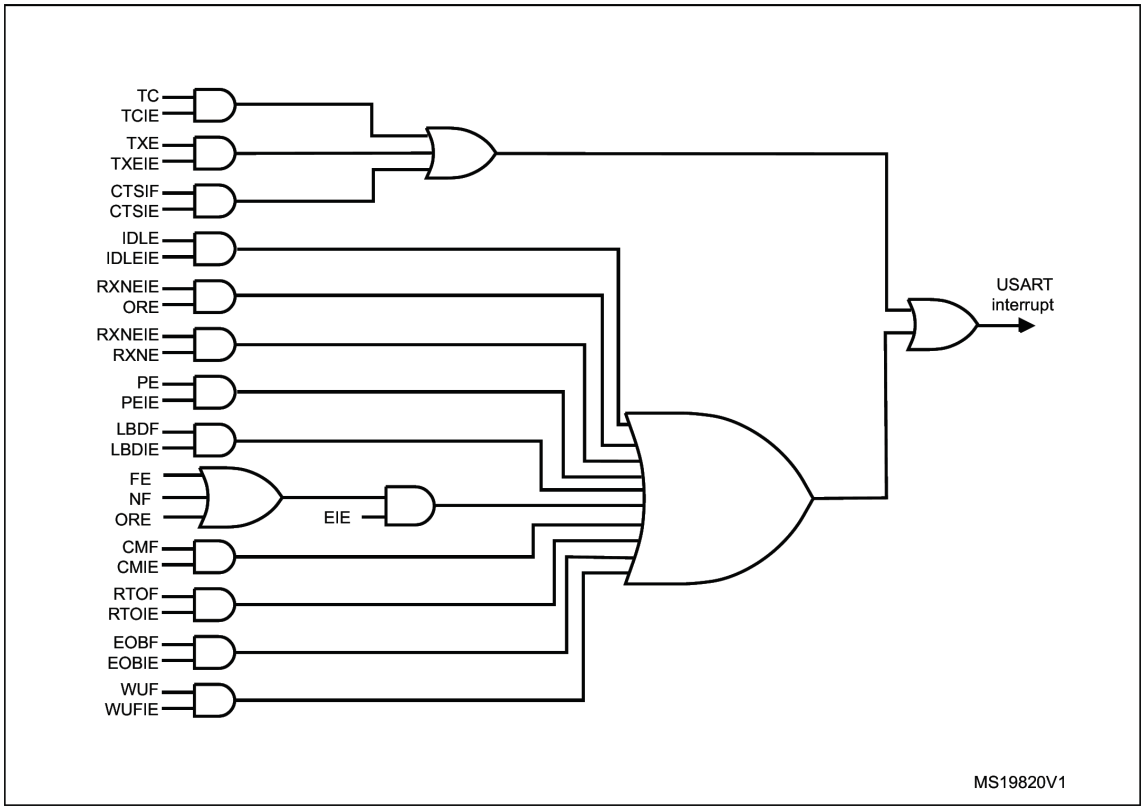
1. WUF 中断只在 Stop 模式中有用。

USART 中断时间全部连接到同一个中断向量 (见图 250)

- 发送期间： 发送完成，CTS，发送数据寄存器空或帧错误（智能卡模式下）中断。
- 接收期间： 空闲线检测，溢出错误，接收数据寄存器非空，校验错误，LIN 断开检测，噪声标志（仅在多缓冲区通讯时），帧错误（仅在多缓冲区通讯），字符匹配，等等。

如果设置了相应的使能控制位，这些事件都可以引起中断。

图 250. USART 中断镜像图



24.7 USART 寄存器

参见 31 页的章节 1.1 中对寄存器描述所采用的缩写列表

24.7.1 控制寄存器 1（USART_CR1）

地址偏移 : 0x00

复位值 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	EOBIE	RTOIE	DEAT[4:0]					DEDT[4:0]				
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVER8	CMIE	MME	M	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	UESM	UE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位时的值

位 27 EOBIE: 块尾中断使能
由软件置 1 和清零。
0: 中断禁止
1: 当 USART_ISR 寄存器中的 EOBF 标志被置 1 时引发 USART 中断
注： 如果 USART 不支持智能卡模式，该位保留并由硬件强制为零。 请参见表 81：STM32F0xx USART 具体功能配备。

位 26 RTOIE: 接收超时中断接收超时中断使能
由软件置 1 和清零。
0: 中断禁止
1: 当 USART_ISR 寄存器中的 RTOF 标志被置 1 时引发 USART 中断。
注： 如果 USART 不支持接收超时功能，该位保留并由硬件强制为零。 表 81. STM32F0xx USART 具体功能配备。

位 25:21 DEAT[4:0]: 驱动使能提前时间
这个 5 位数字定义 DE（驱动器使能）信号激活和第一个发送字节的起始位的时间间隔。 它以采样时间为单位（1/8 或者 1/16 位时间，由过采样率决定）
注： 如果串口不支持驱动器使能功能，这个位保留，且必须保持为零。 请参见表 81：STM32F0xx USART 具体功能配备。
这个位域只能在 USART 未被使能的时候（UE=0）改写。

位 20:16 DEDT[4:0]: 驱动使能滞后时间
这个 5 位数字是一个发送消息的最后一个字节的停止位和释放 DE 信号之间的时间间隔。 它以采样时间为单位（1/8 或者 1/16 位时间，由过采样率决定）
如果 USART_TDR 寄存器在 DEDT 时间内被改写，新的数据只会在 DEDT 和 DEAT 时间都过去之后才会被发送。
注： 如果串口不支持驱动器使能功能，这个位保留，且必须保持为零。 请参见表 81：STM32F0xx USART 具体功能配备。
这个位域只能在 USART 未被使能的时候（UE=0）改写。

位 15 OVER8: 过采样模式
0: 16 倍过采样
1: 8 倍过采样
注： 在 LIN，IrDA 和智能卡模式中，这个位必须保持为零。
这个位域只能在 USART 未被使能的时候（UE=0）改写。

位 14 CMIE: 字符匹配中断使能
由软件置 1 和清零。
0: 中断禁止
1: 当 USART_ISR 寄存器中的 CMF 标志被置 1 时引发 USART 中断。

位 13 MME: 静默模式使能
这个位开启 USART 的静默模式功能。当置为 1 时，USART 可以在活动模式和静默模式之间切换，和 WAKE 位的定义一样， 由软件置 1 和清零。
0: 接收器长期处于活动模式
1: 接收器可以在活动模式和静默模式间切换。

位 12 M: 字长
这个位决定串口字长。 由软件置 1 和清零。
0: 1 个起始位，8 位数据位，n 个停止位
1: 1 个起始位，9 个数据位，n 个停止位
这个位域只能在 USART 未被使能的时候（UE=0）改写。

位 11 WAKE: 接收器唤醒方式
这个位决定 USART 从静默模式唤醒的方式。 由软件置 1 和清零。
0: 空闲线
1: 地址标记
这个位域只能在 USART 未被使能的时候（UE=0）改写。

位 10 PCE: 校验控制使能
这个位选择硬件校验控制（产生和检测）功能。 当校验控制被打开，计算好的校验位被插入到最高位（M=1 时是第九位，M=0 时是第八位），并检测接收数据的校验位。由软件置 1 和清零。一旦这个位被置 1，在当前字节之后就激活了校验控制（在收发的时候都有）。
0: 校验控制禁止
1: 校验控制使能
这个位域只能在 USART 未被使能的时候（UE=0）改写。

位 9 PS: 校验选择
这个位选择在校验生成和检测功能被打开的时候（PCE=1）使用奇校验还是使用偶校验。 由软件置 1 和清零。 校验方式会在当前字节结束后生效。
0: 偶校验
1: 奇校验
这个位域只能在 USART 未被使能的时候（UE=0）改写。

位 8 PEIE: 校验错误中断使能
由软件置 1 和清零。
0: 中断禁止
1: 在 USART_ISR 寄存器中的 PE 被置 1 的时候会产生 USART 中断。

位 7 TXEIE: 发送寄存器空中断使能
由软件置 1 和清零。
0: 中断禁止
1: 在 USART_ISR 寄存器中的 TXE 被置 1 的时候会产生 USART 中断

位 6 TCIE: 发送完毕中断使能
由软件置 1 和清零。
0: 中断禁止
1: 在 USART_ISR 寄存器中的 TC 位被置 1 的时候会产生 USART 中断

位 5 RXNEIE: 接收寄存器非空中断使能
由软件置 1 和清零。
0: 中断禁止
1: 在 USART_ISR 寄存器中的 ORE 或者 RXNE 被置 1 的时候会产生 USART 中断。

位 4 IDLEIE: 空闲中断使能
由软件置 1 和清零。
0: 中断禁止
1: 在 USART_ISR 寄存器中的 IDLE 位被置 1 的时候会产生 USART 中断

位 3 TE: 发送器使能

这个位打开发送器。 由软件置 1 和清零。

0: 发送器关闭

1: 发送器打开

注： 1: 在发送期间，TE 位上的一个 '0' 脉冲 ('0' 后面跟一个 '1') 会引起当前字发送完后跟着再发送一个线路空闲信息出去。但在智能卡模式中则没有这个功能。 要产生一个空闲字符，TE 位不可以立即被写 1。 为了确保所需的持续时间，软件可以等一下 USART_ISR 寄存器中的 TEACK 位。

2: 当 TE 被置为 1 后，和发送开始之间有 1 个位的延迟时间。

位 2 RE: 接收器使能

这个位打开接收器。 由软件置 1 和清零。

0: 接收器被关闭

1: 接收器被打开并开始等待起始位

位 1 UESM: USART 在 Stop 模式下使能

当这个位为零，USART 不能够将 MCU 从 Stop 模式下唤醒。 当这个位为 1 时，USART 可以将 MCU 从 Stop 模式唤醒，条件是 USART 的时钟选择位 HSI 或 LSE。由软件置 1 和清零。

0: USART 不能从 Stop 模式中唤醒 MCU。

1: USART 可以从 Stop 模式中唤醒 MCU。 当这个功能被打开，USART 的时钟源必须为 HSI 或者是 LSE（参见 RCC 章节）

注： 1: 推荐在进入 Stop 模式前将这个 UESM 位置 1，并在推出 Stop 模式后将它清零。

2: 如果 USART 不支持从 Stop 模式唤醒功能，该位为保留位并由硬件强制为零。 请参见表 81： STM32F0xx USART 具体功能配备。

位 0 UE: USART 使能

当这个位被清零，USART 的预分频器和输出都立即停止，并且当前的操作也被取消。对 USART 的设置都不会丢，但 USART_ISR 中所有的状态标志都会被复位。 由软件置 1 和清零。

0: USART 预分频器和输出关闭，低功耗模式

1: USART 开启

注： 1: 为了进入低功耗模式而不在线路上产生错误，需要在清零 UE 位之前先清零 TE 位，并等待 USART_ISR 中的 TC 位被置 1。

2: 在 UE=0 的时候 DMA 请求也同时被复位，所有在清零 UE 位之前也必须先禁止掉 DMA 通道。

24.7.2 控制寄存器 2(USART_CR2)

地址偏移 : 0x04

复位值 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADD[7:4]				ADD[3:0]				RTOEN	ABRMOD[1:0]		ABREN	MSBFIRST	DATAINV	TXINV	RXINV
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWAP	LINEN	STOP[1:0]		CLKEN	CPOL	CPHA	LBCL	SSM	LBDIE	LBDL	ADDM7	Res	Res	Res	Res
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				

位 31:28 ADD[7:4]: USART 的节点地址

这个位域给出 USART 节点的地址或等待确认的字符码。

这用在多机通讯并且进入静默状态或者 Stop 模式的时候，用于 7 位地址标记的检测。发送器发出的字符的最高位应该为 1。 也用在正常接收过程中的字符检测中，这时不打开静默状态（例如，在 ModBus 协议下对块尾的检测）。 这个时候，整个收到的 8 位字节与 ADD[7:0] 进行全面比较，如果匹配，将会引起 CMF 标志被硬件置起。

这个位域只能在接收器被关闭（RE=0）或者在 USART 被关闭的时候（UE=0）才能改写。

位 27:24 ADD[3:0]: USART 的节点地址

这个位域给出 USART 节点的地址或等待确认的字符码。

这用在多机通讯并且进入静默状态或者 Stop 模式的时候，用于 7 位地址标记的检测。

这个位域只能在接收器被关闭（RE=0）或者在 USART 被关闭的时候（UE=0）才能改写。

位 23 RTOEN: 接收器超时检测功能使能

由软件置 1 和清零。

0: 接收器超时检测功能关闭

1: 接收器超时检测功能开启

当这个功能开启，只要 RX 线发现空闲（不是接收）达到 RTOR 寄存器（接收超时寄存器）中设置的时间长度后，USART_ISR 寄存器中的 RTOF 标志会被硬件置 1。

注： 如果 USART 不支持接收超时功能，该位为保留位并由硬件强制为零。 请参见表 81： STM32F0xx USART 具体功能配备。

位 22:21 ABRMOD[1:0]: 自动波特率检测模式

由软件设置或清零

00: 对起始位的测量被用来检测波特率。

01: 使用下降沿对下降沿的测量。 （接收数据必须以单个的 1 作为开头，帧格式为 Start 1 0 xxxxxx）

10: 保留

11: 保留

注： 如果 DATAINV=1 和 / 或 MSBFIRST=1，要求和线路上使用相同的样式，例如 0xAA 对于 MSBFIRST）

注： 如果 USART 不支持自动波特率功能，该位为保留位并由硬件强制为零。 请参见表 81： STM32F0xx USART 具体功能配备。

这个位域只能在 ABREN=0 或者 USART 未被使能的时候（UE=0）改写。

位 20 ABREN: 自动波特率检测使能
由软件置 1 和清零。
0: 自动波特率检测被禁止。
1: 自动波特率检测被打开
注： 如果 USART 不支持自动波特率功能，该位为保留位并由硬件强制为零。 请参见表 81： STM32F0xx USART 具体功能配备。

位 19 MSBFIRST: 高位在前
由软件置 1 和清零。
0: 数据在发送和接收的时候，采用起始位在前，后面跟着第 0 位的顺序。
1: 数据在发送和接收的时候，采用起始位在前，后面跟着最高位（位 7 或者 8）的顺序。
这个位域只能在 USART 未被使能的时候（UE=0）改写。

位 18 DATAINV: 二进制数反向
由软件置 1 和清零。
0: 数据寄存器中的逻辑数据在发送和接收的时候，采用正 / 直接逻辑。 (1=H, 0=L)
1: 数据寄存器中的逻辑数据在发送和接收的时候，采用负 / 反向逻辑。 (1=L, 0=H). 校验位也一样反向。
这个位域只能在 USART 未被使能的时候（UE=0）改写。

位 17 TXINV: TX 脚有效电平反向
由软件置 1 和清零。
0: TX 脚信号工作于标准逻辑电平 (VDD =1/idle, Gnd=0/mark)
1: TX 脚信号被反向。 ((VDD =0/mark, Gnd=1/idle). 这可以用于 TX 线上带有外部反相器的时候。
这个位域只能在 USART 未被使能的时候（UE=0）改写。

位 16 RXINV: RX 脚有效电平反向
由软件置 1 和清零。
0: RX 脚信号工作于标准逻辑电平 (VDD =1/idle, Gnd=0/mark)
1: RX 脚信号被反向。 ((VDD =0/mark, Gnd=1/idle). 这可以用于 RX 线上带有外部反相器的时候。
这个位域只能在 USART 未被使能的时候（UE=0）改写。

位 15 SWAP: 交换 TX/RX 引脚
由软件置 1 和清零。
0: TX/RX 引脚按照标准引脚分配来使用
1: TX 和 RX 的引脚功能交换使用。 这用于和其它 UART 口进行交叉互联的时候。
这个位域只能在 USART 未被使能的时候（UE=0）改写。

位 14 LINEN: LIN 模式使能
由软件置 1 和清零。
0: LIN 模式禁止
1: LIN 模式使能
LIN 模式打开后，可以具备发送 LIN 同步断开（13 个低位）的功能，用 USART_CR1 寄存器的 SBKRQ 位实现，同时还具备 LIN 同步断开信号的检测功能。
这个位域只能在 USART 未被使能的时候（UE=0）改写。
注： 如果 USART 不支持 LIN 模式，该位为保留位并由硬件强制为零。请参见表 81： STM32F0xx USART 具体功能配备。

位 13:12 STOP[1:0]: 停止位
这些位用来定制停止位的个数。
00: 1 个停止位：
01: 保留
10: 2 个停止位：
11: 1.5 个停止位
这个位域只能在 USART 未被使能的时候（UE=0）改写。

位 11 CLKEN: 时钟使能
这个位用来打开 SCLK 引脚的功能
0: SCLK 引脚被禁止
1: SCLK 引脚被使能
这个位域只能在 USART 未被使能的时候（UE=0）改写。
注： 如果同步模式和智能卡模式都不被支持，这个位保留并由硬件强制为 0。 请参见表 81： STM32F0xx USART 具体功能配备。

位 10 CPOL: 时钟极性
这个位允许用户选择同步模式下 SCLK 引脚上的时钟输出的极性。 它连同 CPHA 位一起决定所需要的时钟 / 数据时序关系。
0: 在没有数据传输的时候保持低电平
1: 在没有数据传输的时候保持高电平
这个位域只能在 USART 未被使能的时候（UE=0）改写。
注： 如果同步模式不被支持，这个位保留并由硬件强制为 0。 请参见表 81： STM32F0xx USART 具体功能配备。

位 9 CPHA: 时钟相位
这个位允许用户选择同步模式下 SCLK 引脚上的时钟输出的相位。 它连同 CPOL 位一起决定所需要的时钟 / 数据时序关系。（见图 238 和图 239）
0: 第一个时钟沿对准第一位数据
1: 第二和时钟沿对准第一位数据
这个位域只能在 USART 未被使能的时候（UE=0）改写。
注： 如果同步模式不被支持，这个位保留并由硬件强制为 0。 请参见表 81： STM32F0xx USART 具体功能配备。

位 8 LBCL: 末位时钟脉冲
这个位用来选择在同步模式下，SCLK 脚上传输最后一个位（MSB）的时候是否必须输出时钟脉冲。
0: SCLK 脚上在传输末位数据的时候不输出时钟脉冲。
1: SCLK 脚上在传输末位数据的时候输出时钟脉冲。
警告：末位是第 8 位还是第 9 位，取决于 USART_CR1 寄存器中的 M 位的设置。
这个位域只能在 USART 未被使能的时候（UE=0）改写。
注： 如果同步模式不被支持，这个位保留并由硬件强制为 0。 请参见表 81： STM32F0xx USART 具体功能配备。

位 7 保留，必须保持复位时的值。

位 6 LBDIE: LIN 断开信号检测中断使能
断开中断屏蔽（利用断开分隔符来检测）
0: 中断禁止
1: 在 USART_ISR 寄存器中的 LBDF 被置 1 的时候会产生 USART 中断
注： 如果 LIN 模式不被支持，这个位保留并由硬件强制为 0。请参见表 81：STM32F0xx USART 具体功能配备。

位 5 LBDL: LIN 断开检测长度
这个位用来选择使用 11 位还是 10 位断开检测。
0: 10 位断开检测
1: 11 位断开检测
这个位域只能在 USART 未被使能的时候（UE=0）改写。
注： 如果 LIN 模式不被支持，这个位保留并由硬件强制为 0。请参见表 81：STM32F0xx USART 具体功能配备。

位 4 ADDM7:7 位地址检测或 4 位地址检测选择
这个位用来选择使用 4 位地址检测还是 7 位地址检测。
0: 4 位地址检测
1: 7 位地址检测（8 位数据模式下）
这个位域只能在 USART 未被使能的时候（UE=0）改写。
注： 在 7 位和 9 位数据模式下，地址检测分别按 6 位和 8 位地址 (ADD[5:0] 和 ADD[7:0]) 操作。

位 3:0 保留，必须保持复位时的值。

注： CPOL, CPHA, LBCL 这三个位在发送器被使能期间不能去改写。

24.7.3 控制寄存器 3（USART_CR3）

地址偏移 : 0x08
复位值 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	WUFIE	WUS[2:0]		SCARCNT2:0]			Res
									rw	rw	rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEP	DEM	DDRE	OVR DIS	ONE BIT	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HD SEL	IRLP	IREN	EIE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:23 保留，必须保持复位时的值。

位 22 WUFIE: 从 Stop 模式唤醒中断使能
由软件置 1 和清零。
0: 中断禁止
1: 在 USART_ISR 寄存器中的 WUF 被置 1 的时候会产生 USART 中断
注： 1. WUFIE 位必须在进入 Stop 模式之前设置。
2. WUF 中断只在 Stop 模式中有用。
3. 如果 USART 不支持从 Stop 模式唤醒功能，该位为保留位并由硬件强制为零。

位 21:20 WUS[1:0]: 从 Stop 模式唤醒中断标志选择
这个位域指定激活 WUF 标志的事件。
00: 在发生地址匹配事件的时候激活 WUF
01: 保留
10: WUF 在检测到起始位的时候激活
11: WUF 在得到接受数据寄存器非空事件时激活
这个位域只能在 USART 未被使能的时候（UE=0）改写。
注： 如果 USART 不支持从 Stop 模式唤醒功能，该位为保留位并由硬件强制为零。

位 19:17 SCARCNT[2:0]: 智能卡模式重试计数器
这个位域指定智能卡模式中接收和发送的重试次数。 在发送模式下，它指定的是在产生发送错误（FE=1）之前自动重试发送的次数。
在接收模式下，它指定的是在产生接收错误事件前（RXNE 和 PE=1）所作的错误接收的尝试的次数。
这个位域只能在 USART 未被使能的时候（UE=0）改写。
当 USART 被使能，这个位域只允许写成 0x0，这是为了避免盲目的自动重发数据。
0x0: 重发功能关闭 - 在发送模式下不进行自动重发操作。
0x1 to 0x7: 自动重传的尝试次数（在提示错误之前）
注： 如果智能卡模式不被支持，这个位保留并由硬件强制为 0。
请参见表 81： STM32F0xx USART 具体功能配备。

位 16 保留，必须保持复位时的值。

位 15 DEP: 驱动使能输出脚的极性选择
0: DE 信号高有效
1: DE 信号低有效
注： 如果串口不支持驱动器使能功能，这个位保留，且必须保持为零。 请参见表 81： STM32F0xx USART 具体功能配备。
这个位域只能在 USART 未被使能的时候（UE=0）改写。

位 14 DEM: 驱动器使能模式
它允许用户通过 DE（驱动使能）信号来激活外部收发器的控制端。
0: DE 功能被禁止
1: DE 功能被打开。 DE 信号在 RTS 脚输出。
注： 如果串口不支持驱动器使能功能，这个位保留，且必须保持为零。 表 81. STM32F0xx USART 具体功能配备。
这个位域只能在 USART 未被使能的时候（UE=0）改写。

位 13 DDRE: 在接收错误的时候禁止 DMA

0: 在发生接收错误的时候不禁止 DMA。 相应的错误标志被置 1，但 RXNE 仍保持零以阻止数据溢出覆盖。 作为结果，不会发起 DMA 请求，所以错误的数据不会被传输（因为没有 DMA 请求），但下一个正确的数据会被传送。（用于智能卡模式）

1: 在发生接收错误之后，DMA 被关闭。相应的错误标志会随着 RXNE 的置 1 而被置 1。DMA 请求会被屏蔽，直到相关的错误标志被清零。 这意味着软件必须先禁止 DMA 请求（DMAR=0）或者在清零错误标志前先清零 RXNE 标志。

这个位域只能在 USART 未被使能的时候（UE=0）改写。

注： 接收错误指的是： 校验错，帧错误或噪声错误。

位 12 OVRDIS: 溢出检测禁止

这个位用于禁止对接收溢出现象的检测。

0: 当之前接收到的数据没有在新接收到数据之前读走时，会引起溢出错误标志 ORE 被硬件置 1。

1: 溢出检测功能关闭。 如果在新的接收数据到来时，RXNE 标志仍然是 1，但 ORE 标志还不是 1 时，新的数据会将 USART_RDR 中以前的内容覆盖掉。

这个位域只能在 USART 未被使能的时候（UE=0）改写。

注： 这个控制位允许读取数据的时候检查通讯数据流。

位 11 ONEBIT: 单次采样方式使能

这个位允许用户选择采样方式。 当选择单次采样方式的时候，噪声监测标志（NF）就被禁止了。

0: 三次采样方式

1: 单次采样方式

这个位域只能在 USART 未被使能的时候（UE=0）改写。

位 10 CTSIE: CTS 中断使能

0: 中断禁止

1: 在 USART_ISR 寄存器中的 CTSIF 被置 1 的时候会产生 USART 中断

注： 如果硬件流控制不被支持，这个位保留并由硬件强制为 0。 请参见表 81： STM32F0xx USART 具体功能配备。

位 9 CTSE: CTS 功能使能

0: 关闭 CTS 硬件流控

1: 打开 CTS 硬件流控，只有在 nCTS 输入上收到有效信号（被拉低）时才会发送数据。如果在数据传送时 nCTS 输入变为无效，会在数据发送完成后再停。如果写数据到发送寄存器的时候，nCTS 处于无效状态，那么这个数据的发送会被推迟直到 nCTS 信号变成有效。

这个位域只能在 USART 未被使能的时候（UE=0）改写。

注： 如果硬件流控制不被支持，这个位保留并由硬件强制为 0。 请参见表 81： STM32F0xx USART 具体功能配备。

位 8 RTSE: RTS 使能

0: 关闭 RTS 硬件流控

1: RTS 输出使能，只会在有接收空间的时候才请求下一个数据。当前数据发送完成后，发送操作就需要暂停下来。如果可以接收数据了，将 nRTS 输出置为有效（拉低）。

这个位域只能在 USART 未被使能的时候（UE=0）改写。

注： 如果硬件流控制不被支持，这个位保留并由硬件强制为 0。 请参见表 81： STM32F0xx USART 具体功能配备。

位 7 DMAT: DMA 发送使能
该位由软件置 1 和清零
1: 为发送数据使能 DMA 模式
0: 关闭发送 DMA 模式

位 6 DMAR: DMA 接收使能
该位由软件置 1 和清零
1: 为接收数据使能 DMA 模式
0: 关闭接收 DMA 模式

位 5 SCEN: 智能卡模式使能
该位用于打开智能卡模式。
0: 关闭智能卡模式
1: 打开智能卡模式
这个位域只能在 USART 未被使能的时候（UE=0）改写。
注： 如果 USART 不支持智能卡模式，该位为保留为并由硬件强制为零。 请参见表 81： STM32F0xx USART 具体功能配备。

位 4 NACK: 智能卡 NACK 发送使能
0: 出现校验错误的时候不发送 NACK
1: 出现校验错误的时候，发送 NACK
这个位域只能在 USART 未被使能的时候（UE=0）改写。
注： 如果 USART 不支持智能卡模式，该位为保留位并由硬件强制为零。 请参见表 81： STM32F0xx USART 具体功能配备。

位 3 HDSEL: 半双工选择
选择单线半双工模式
0: 不选择半双工模式
1: 选择半双工模式
这个位域只能在 USART 未被使能的时候（UE=0）改写。

位 2 IRLP: IrDA 低功耗模式选择
用来选择 IrDA 的普通模式还是低功耗模式
0: 正常模式
1: 低功耗模式
这个位域只能在 USART 未被使能的时候（UE=0）改写。
注： 如果 IrDA 模式不被支持，这个位保留并由硬件强制为 0。 请参见表 81： STM32F0xx USART 具体功能配备。

位 1 IREN: 红外模式使能
由软件置 1 和清零。
0: 不使能红外模式
1: 使能红外模式
这个位域只能在 USART 未被使能的时候（UE=0）改写。
注： 如果 IrDA 模式不被支持，这个位保留并由硬件强制为 0。 请参见表 81： STM32F0xx USART 具体功能配备。

位 0 EIE: 错误中断使能
在允许帧错误，溢出错误或噪声错误产生中断请求时要打开这个开关。
0: 中断禁止
1: 当 USART_ISR 寄存器中的 FE=1 或 ORE=1 或 NF=1 时，会产生中断。

24.7.4 波特率寄存器（ USART_BRR）

这个寄存器只能在 USART 未被使能的时候（UE=0）改写。在自动波特率检测模式下，可能会被硬件自动改写。

地址偏移 : 0x0C

复位值 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV_Mantissa[11:0]												DIV_Fraction[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位时的值。

位 15:4 DIV_Mantissa[11:0]: USARTDIV 的整数部分

这 12 位定义 USART 分频器除法因子的整数部分

位 3:0 DIV_Fraction[3:0]: USARTDIV 的小数部分

这 4 位定义 USART 分频器除法因子的小数部分 当 OVER8=1 时，DIV_Fraction3 位是没有用的，并且必须保持为 0。

24.7.5 保护时间和预分频器寄存器（ USART_GTPR）

地址偏移 : 0x10

复位值 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GT[7:0]								PSC[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位时的值。

位 15:8 GT[7:0]: 保护时间值

这个位域用来设置保护时间长度，以波特时钟为单位。

在智能卡模式下要用到。 在保护时间过去之后才会设置发送完成标志。

这个位域只能在 USART 未被使能的时候（UE=0）改写。

注： 如果智能卡模式不被支持，这个位保留并由硬件强制为 0。

请参见表 81： STM32F0xx USART 具体功能配备。

位 7:0 PSC[7:0]: 预分频器值

- 在红外低功耗和正常模式下： PSC[7:0] = IrDA 正常和低功耗模式波特率 对 USART 时钟源进行分频以获得低功耗模式下的频率：

时钟源按寄存器给定的值来分频（8 位有效）：

00000000: 保留 - 不要写这种无聊的值

00000001: 1 分频

00000010: 2 分频

...

- 智能卡模式： PSC[4:0]: 预分频器值

用于设定对 USART 时钟源的分频系数，得到智能卡时钟。按寄存器中的值（低 5 位有效）乘以 2 得到的值作为分频系数来分频：

00000: 保留 - 不要写这种无聊的值

00001: 2 分频

00010: 4 分频

00011: 6 分频

...

这个位域只能在 USART 未被使能的时候（UE=0）改写。

注： 如果使用智能卡模式，Bits [7:5] 必须保持为 0。

当智能卡和红外模式不被支持时，这个位域是保留的，并且由硬件强制为 '0'。 请参见表 81： STM32F0xx USART 具体功能配备。

24.7.6 接收超时寄存器(USART_RTOR)

地址偏移 : 0x14

复位值 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BLEN[7:0]								RTO[23:16]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTO[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 BLEN[7:0]: 块长度

这个位域给出了智能卡模式 T=1 的接收时的块长度。 这个值等于 信息块字符数 + 结束部分（1-LEC/2-CRC）-1 。

例如：

BLEN = 0 -> 0 个信息字符 + LEC

BLEN = 1 -> 0 个信息字符 + CRC

BLEN = 255 -> 254 个信息字符 +CRC（总共 256 个字符）智能卡模式中，当 TXE=0，会导致块长度计数器清零。

这个位域也可以在其它模式中使用。 这时，块长度计数器在 RE=0 的时候清零 和 / 或 在 EOBCF 位被写 1 的时候清零。

注： 这个值可以在块接收开始之后再去设置（用于需要从块的序言部分提取块长度的情形）。 每个块接收的时候只能设置一次这个。

位 23:0 RTO[23:0]: 接收超时值

这个位域给出了接收超时的设置，单位是波特时钟的时长。

标准模式下，最后一个字节接收后，在整个 RTO 值规定的时长内，再也没有检测到新的起始位的时候，RTOF 标志会被硬件置 1。

在智能卡模式中，这个值被用来实现 CWT 和 BWT。 详细信息参见智能卡相关章节。这里，超时测量时从最后一个字节的其实位开始算的。

注： 对于每个接收字符只能改写一次这个值。

注： RTOR 可以边运行边改写，如果新值小于等于已经产生的超时计数，无非就是马上导致 RTOF 标志被置 1。

注： 当接收超时功能未被支持时，这个寄存器保留，并有硬件强制为 0。 请参见表 81： STM32F0xx USART 具体功能配备。

24.7.7 请求寄存器(USART_RQR)

地址偏移 : 0x18

复位值 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TXFRQ	RXFRQ	MMRQ	SBKRQ	ABRRQ
											w_r0	w_r0	w_r0	w_r0	w_r0

位 31:5 保留，必须保持复位时的值。

位 4 TXFRQ: 发送数据清空请求

对这个位写 1 会直接令 TX 标志被硬件置 1

这里允许取消数据发送。 这个位只能在智能卡模式下使用，当数据由于所发生的错误导致还没发出去，USART_ISR 寄存器的 FE 标志是 1 的时候用。

如果 USART 不支持智能卡模式，该位为保留位并由硬件强制为零。 请参见表 81：STM32F0xx USART 具体功能配备。

位 3 RXFRQ: 接收数据清空请求

对这个位写 1 会直接令 RXNE 标志被硬件清零

这里允许直接丢弃还没有读的接收数据，免得被提示溢出错误。

位 2 MMRQ: 静默模式请求

对这个位写 1 将导致 USART 进入静默模式，同时清除 RWU 标志。

位 1 SBKRQ: 请求发送断开字符

对这个位写 1 会令 SBKF 标志置 1，并在发送状态机可用的时候向线路发出一个断开字符。

位 0 ABRRQ: 自动波特率检测请求

向这个位写 1 会清除 USART_ISR 中的 ABRF 标志，并在下一次数据接收的时候开始一次自动波特率测量。

注： 如果 USART 不支持自动波特率功能，该位为保留位并由硬件强制为零。 请参见表 81： STM32F0xx USART 具体功能配备。

24.7.8 中断和状态寄存器(USART_ISR)

地址偏移 : 0x1C

复位值 : 0x00C0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	REACK	TEACK	WUF	RWU	SBKF	CMF	BUSY
									r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABRF	ABRE	ABRE	EOBF	RTOF	CTS	CTSIF	LBDF	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:23 保留，必须保持复位时的值。

位 22 REACK: 接收使能通知标志

这个位有硬件控制，当接收使能信号被 USART 读到的时候，会给出一个回执。

这可以在进入 Stop 模式之前，用来确认 USART 是不是已经准备好接收了。

注： 如果 USART 不支持从 Stop 模式唤醒功能，该位为保留位并由硬件强制为零。

位 21 TEACK: 发送使能通知标志

这个位有硬件控制，当发送使能信号被 USART 读到的时候，会给出一个回执。

这可以在写 USART_CR1 寄存器的 TE=0 以产生一个空闲帧请求，跟着又将 TE 写成 1 的时候，用于保证 TE=0 的最小周期的时候用。

位 20 WUF: 从 Stop 模式唤醒标志

当检测到唤醒事件时由硬件置 1。 具体时间有 WUS 位域来定义。 由软件向 USART_ICR 寄存器的 PECF 位写 1，可以清除这个标志。 如果 USART_CR3 寄存器的 WUF=1，会产生中断请求。

注： 1. 当 UESM 被清零，WUF 标志也会被清零。

2. WUF 中断只在 Stop 模式中有用。

3. 如果 USART 不支持从 Stop 模式唤醒功能，该位为保留位并由硬件强制为零。

位 19 RWU: 接收器从静默模式唤醒

这个位表示 USART 处于静默模式时，当唤醒和静默模式切换时，由硬件清零和置 1。静默模式控制顺序（地址还是空闲）用 USART_CR1 寄存器的 WAKE 位来选择。

如果选择由空闲信号唤醒，那这个位就只能由软件置 1 了，方法是向 USART_RQR 寄存器置 1

0: 接收器处于活动模式

1: 接收器处于静默模式

注： 如果 USART 不支持从 Stop 模式唤醒功能，该位为保留位并由硬件强制为零。

位 18 SBKF: 断开信号发送标志

这个位表示当要求发送一个断开字符时， 由软件置 1，方法是向 USART_CR3 寄存器的 SBKRQ 位写 1。 当断开字符的停止位传出后，由硬件自动清零。

0: 没发送断开字符

1: 断开字符将会被发送

位 17 CMF: 字符匹配标志

当收到一个字符和 ADD[7:0] 中设置的内容相同时，由硬件置 1。 由软件向 USART_ICR 寄存器的 CMCF 位写 1，可以清除这个标志。

如果 USART_CR1 寄存器中的 CMIE 位是 1，就会产生中断请求。

0: 没发现字符匹配

1: 有发现字符匹配

位 16 BUSY: 忙标志

由硬件置 1 和清零。 当 RX 线在通讯时（成功检测到起始位），这个位被硬件置 1。接收结束后（不管成功与否）会由硬件清零。

0: USART 处于空闲（没接收）

1: 正在接收数据

位 15 ABRF: 自动波特率检测标志

当自动波特率功能打开时（RXNE 也被置 1，如果 RXNEIE=1 产生中断）或者当自动波特率操作不成功时（ABRE，RXNE 和 FE 也都被置 1 的时候），这个位被硬件置 1。

开始一轮新的波特率检测（向 USART_RQR 寄存器的 ABRQ 写 1）的时候会被清除。

注： 如果 USART 不支持自动波特率功能，该位为保留位并由硬件强制为零。

位 14 ABRE: 自动波特率检测错误

在波特率测量失败的时候（超量程或者字符比较失败）由硬件置 1。

由软件清零，方法是向 USART_CR3 寄存器的 ABRRQ 位写 1。

注： 如果 USART 不支持自动波特率功能，该位为保留位并由硬件强制为零。

位 13 保留，必须保持复位时的值。

位 12 EOBF: 块结束标志

当一个完整的块被接收时由硬件置 1（例如 T=1 智能卡模式中）。当收到的字节数（从块开始，包括序言部分）大于等于 BLEN+4 的时候完成检测。

如果 USART_CR2 寄存器的 EOBIE=1，会产生中断请求。

由软件向 USART_ICR 寄存器的 EOBCF 位写 1，可以清除这个标志。

0: 还没到块结束

1: 块结束（足够字节数）已经到了

注： 如果智能卡模式不被支持，这个位保留并由硬件强制为 0。

请参见表 81： STM32F0xx USART 具体功能配备。

位 11 RTOF: 接收超时标志

如果没有通讯的时间长度达到了 RTOR 寄存器中设定的超时值，这个位会被硬件置 1。 由软件向 USART_ICR 寄存器的 RTOCF 位写 1，可以清除这个标志。

如果 USART_CR2 寄存器的 RTOIE=1，会产生中断请求。

在智能卡模式中，这个超时相当于 CWT 或 BWT 计时。

0: 尚未超时

1: 已经达到超时

注： 1. 如果 RTOR 寄存器中设定的时间点正好位于两个字符中，RTOF 不会被置 1。而如果这个时间达到了设定值 +2 个采样周期（2/16 或 2/8，取决于过采样方式的设定），RTOF 标志会被置 1。

2. 在 RE=0 的时候还是会进行超时计数，但 RTOF 只会在 RE=1 的时候置 1。 如果在 RE 被置 1 的时候，超时早就过了，RTOF 会马上被置 1。

3. 如果 USART 不支持接收超时功能，该位为保留位并由硬件强制为零。

位 10 CTS: CTS 标志

该位由硬件置 1 和清零 这是 nCTS 输入脚状态的反向拷贝。

0: nCTS 线是高

1: nCTS 线是低

注： 如果硬件流控制不被支持，这个位保留并由硬件强制为 0。

位 9 CTSIF: CTS 中断标志

如果 CTSE 位为 1，那么当 nCTS 输入状态变化的时候，由硬件置 1。 由软件向 USART_ICR 寄存器的 CTSCF 位写 1，可以清除这个标志。

如果 USART_CR3 寄存器的 CTSIE=1，会产生中断请求。

0: nCTS 线的状态无变化

1: nCTS 线的状态有变化

注： 如果硬件流控制不被支持，这个位保留并由硬件强制为 0。

位 8 LBDF: LIN 断开检测

当检测到 LIN 断开字符时由硬件置 1。由软件向 USART_ICR 寄存器的 LBDCF 位写 1，可以清除这个标志。

如果 USART_CR2 寄存器的 LBDIE=1，会产生中断请求。

0: 没检测到 LIN 断开字符

1: 检测到 LIN 断开字符

注： 如果 USART 不支持 LIN 模式，该位为保留位并由硬件强制为零。 请参见表 81： STM32F0xx USART 具体功能配备。

位 7 TXE: 发送数据寄存器空

当 USART_TDR 寄存器中的值被取到移位寄存器的同时，这个位被硬件置 1。 再向 USART_TDR 寄存器写数据就会同时清掉这个位。

TXE 标志还可以用其它方式清除，例如向 USART_RQR 寄存器的 TXFRQ 位写 1，为了丢弃数据（仅于智能卡模式 T=0，传送失败的情形）。

如果 USART_CR1 寄存器中的 TXEIE 位被置起时，则会产生中断。

0: 没有数据被传到移位寄存器

1: 有数据被传到移位寄存器，发送数据寄存器为空。

注： 这个位在单缓冲区发送的时候会用到。

位 6 TC: 发送完成

在 TXE 为 1 的条件下，当数据发送完成的时候，这个位会被硬件置 1。 如果 USART_CR1 寄存器中的 TCIE 位是 1，就会产生中断请求。 向 USART_TDR 寄存器再次写入数据，或者向 USART_ICR 寄存器的 TCCF 位写 1，都可以清除这个标志。

如果 USART_CR1 寄存器中的 TCIE 位是 1，就会产生中断请求。

0: 发送未完成

1: 发送完成

注： 如果 TE 位被清零，并且没有在发送数据，那 TC 位会立即被置 1。

位 5 RXNE: 接收数据寄存器非空

当接收移位寄存器的内容被传递到 USART_RDR 寄存器中时，这个位被硬件置 1。读取 USART_TDR 寄存器的数据就会同时清掉这个位。 RXNE 标志也可以通过对 USART_RQR 寄存器中的 RXFRQ 位写 1 来清除。

如果 USART_CR1 寄存器中的 RXNEIE 位是 1，就会产生中断请求。

0: 没收到数据

1: 收到的数据已经可读

位 4 IDLE: 空闲线检测到

当检测到线路空闲时由硬件置 1。 如果 USART_CR1 寄存器中的 IDLEIE 位是 1，就会产生中断请求。 由软件向 USART_ICR 寄存器的 IDLECF 位写 1，可以清除这个标志。

0: 没有检测到线路空闲

1: 检测到线路空闲

注： 1. 除非 RXNE 位被置 1，否则 IDLE 位不会再次置 1（例如：一个新的线路空闲信号来临）。

2. 如果静默模式被使能了（MME=1），只要 USART 没有处于静默状态（RWU=0），那么 IDLE 会被置 1，而无论在 WAKE 位上用什么方式设置的静默功能。 如果 RWU=1，那 IDLE 就不会被置 1 了。

位 3 ORE: 溢出错误

在 RXNE=1 的的条件下（也就是上次数据还没有读走），串口接收寄存器又接收好了一个字节的数据并准备往 RDR 寄存器去转移的时候，会由硬件将这个位置 1。由软件向 USART_ICR 寄存器的 ORECF 位写 1，可以清除这个标志。

如果 USART_CR1 寄存器中的 RXNEIE 位或 EIE 位是 1，就会产生中断请求。

0: 没有溢出错误

1: 检测到溢出错误

注： 1. 当这个位被置 1，RDR 寄存器中的数据不会丢，但移位寄存器中的（那个新的）数据就会蒸发掉了。 如果在多缓冲区通讯时 EIE 位是 1，并且 ORE 标志被置 1 的话，就会同步引起一个中断请求。

2. 如果 USART_CR3 寄存器中的 OVRDIS 位是 1，那么这个位就会被长期的强制为零（没有了溢出检测功能）。

位 2 NF: 噪声检测标志

当收帧的时候检测到噪声，这一位会由硬件置 1。 由软件向 USART_ICR 寄存器的 NFCF 位写 1，可以清除这个标志。

0: 没有检测到噪声

1: 检测到噪声

注： 1. 这一位不会产生中断，因为和它同时置 1 的 RXNE 能够产生中断。 如果在多缓冲区通讯时 EIE 位是 1，并且 NF 标志被置 1 的话，也会同步引起一个中断请求。

2. 当线路不会受到噪声干扰，可以将这个噪声检测功能关闭，从而降低 USART 对时钟偏差的敏感度，具体做法是将 ONEBIT 位写成 1。（参见章节 24.5.5）： USART 对时钟偏差的容忍度）

位 1 FE: 帧错误

当一个不同步现象、强噪声或一个断开符号被检测到的时候，这个位有硬件置 1。由软件向 USART_ICR 寄存器的 FECF 位写 1，可以清除这个标志。 在智能卡模式中发送数据时，当重发尝试的次数达到上限，由没有收到成功的回应（卡一直响应 NACK）的时候，这个位也会被硬件置 1。

如果 USART_CR1 寄存器中的 EIE 位是 1，会产生中断请求。

0: 没有检测到帧错误

1: 有检测到帧错误或者有收到断开字符

位 0 PE: 校验错误标志

当在接收数据的时候发现校验错误，这个位会由硬件置 1。 由软件向 USART_ICR 寄存器的 PECF 位写 1，可以清除这个标志。

如果 USART_CR1 寄存器中的 PEIE 位是 1，会产生中断请求。

0: 没有校验错误

1: 有校验错误

24.7.9 中断标志清除寄存器(USART_ICR)

地址偏移 : 0x20

复位值 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	WUCF	Res	Res	CMCF	Res
											w_r0			w_r0	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	EOBCF	RTOCF	Res	CTSCF	LBDCF	Res	TCCF	Res	IDLECF	ORECF	NCF	FECF	PECF
			w_r0	w_r0		w_r0	w_r0		w_r0		w_r0	w_r0	w_r0	w_r0	w_r0

位 31:21 保留，必须保持复位时的值。

位 20 WUCF: 从 Stop 模式唤醒标志的清除
对这个位写 1，会清除 USART_ISR 寄存器中的 WUF 标志位。
注： 如果 USART 不支持从 Stop 模式唤醒功能，该位为保留位并由硬件强制为零。

位 19:18 保留，必须保持复位时的值。

位 17 CMCF: 字符匹配标志的清除
对这个位写 1，会清除 USART_ISR 寄存器中的 CMF 标志位。

位 16:13 保留，必须保持复位时的值。

位 12 EOBCF: 块结束标志的清除
对这个位写 1，会清除 USART_ISR 寄存器中的 EOBF 标志位。
注： 如果 USART 不支持智能卡模式，该位为保留位并由硬件强制为零。 请参见表 81： STM32F0xx USART 具体功能配备。

位 11 RTOCF: 接收超时标志的清除
对这个位写 1，会清除 USART_ISR 寄存器中的 RTOF 标志位。
注： 如果 USART 不支持接收超时功能，该位为保留位并由硬件强制为零。 请参见表 81： STM32F0xx USART 具体功能配备。

位 10 保留，必须保持复位时的值。

位 9 CTSCF: CTS 标志的清除
对这个位写 1，会清除 USART_ISR 寄存器中的 CTSIF 标志位。
注： 如果硬件流控制不被支持，这个位保留并由硬件强制为 0。 请参见表 81： STM32F0xx USART 具体功能配备。

位 8 LBDCF: LIN 断开检测标志的清除
对这个位写 1，会清除 USART_ISR 寄存器中的 LBDF 标志位。
注： 如果 LIN 模式不被支持，这个位保留并由硬件强制为 0。 请参见表 81： STM32F0xx USART 具体功能配备。

位 7 保留，必须保持复位时的值。

位 6 TCCF: 发送完成标志的清除
对这个位写 1，会清除 USART_ISR 寄存器中的 TC 标志位。

位 5 保留，必须保持复位时的值。

位 4 IDLECF: 线路空闲检测标志的清除
对这个位写 1，会清除 USART_ISR 寄存器中的 IDLE 标志位。

位 3 ORECF: 溢出错误标志的清除
对这个位写 1，会清除 USART_ISR 寄存器中的 ORE 标志位。

位 2 NCF: 噪声检测标志的清除
对这个位写 1，会清除 USART_ISR 寄存器中的 NF 标志位。

位 1 FECF: 帧错误标志的清除
对这个位写 1，会清除 USART_ISR 寄存器中的 FE 标志位。

位 0 PECF: 校验错误标志的清除
对这个位写 1，会清除 USART_ISR 寄存器中的 PE 标志位。

24.7.10 数据接收寄存器（ USART_RDR）

Address offset: 0x24

Reset value: Undefined

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	Res	RDR[8:0]								
							r	r	r	r	r	r	r	r	r

位 31:9 保留，必须保持复位时的值。

位 8:0 RDR[8:0]: 接收数据的值

包含所收到的字节。

RDR 寄存器提供输入移位寄存器和内部总线间的并行接口（见图 226）。

当接收数据时打开了校验位，读这个寄存器得到的最高位是校验位。

24.7.11 数据发送寄存器（ USART_TDR）

Address offset: 0x28

Reset value: Undefined

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	Res	TDR[8:0]								
							rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:9 保留，必须保持复位时的值。

位 8:0 TDR[8:0]: 发送数据的值

用于写入要发送的数据字节。

TDR 寄存器提供发送移位寄存器和内部总线间的并行接口（见图 226）。

当发送的时候设置了校验功能（USART_CR1 中的 PCE=1），向最高位（位 7 还是位 8 取决于设置的字长）写入的信息是无效的，因为它总是要被校验位代替了之后再去发送的。

注： 这个寄存器只能在 TXE=1 的时候才能做写操作。

24.7.12 USART 寄存器镜像

下表给出了 USART 寄存器镜像和复位值。

表 95. USART 寄存器镜像和复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**USART_CR1**	Res.	Res.	Res.	Res.	EOBIE	RTOIE	DEAT4	DEAT3	DEAT2	DEAT1	DEAT0	DEDT4	DEDT3	DEDT2	DEDT1	DEDT0	OVER8	CMIE	MME	M	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	UESM	UE
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	**USART_CR2**	ADD[7:4]				ADD[3:0]				RTOEN	ABRMOD1	ABRMOD0	ABREN	MSBFIRST	DATAINV	TXINV	RXINV	SWAP	LINEN	STOP [1:0]		CLKEN	CPOL	CPHA	LBCL	Reserved	LBDIE	LBDL	ADDM7	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x08	**USART_CR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUFIE	WUS [1:0]		SCAR CNT2:0]			Res.	DEP	DEM	DDRE	OVRDIS	ONEBIT	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSEL	IRLP	IREN	EIE
	Reset value									0	0	0	0	0	0	0		00	00	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	**USART_BRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIV_Mantissa[15:4]												DIV_Fraction [3:0]			
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	**USART_GTPR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GT[7:0]								PSC[7:0]							
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	**USART_RTOR**	BLEN[7:0]								RTO[23:0]																							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	**USART_RQR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXFRQ	RXFRQ	MMRQ	SBKRQ	ABRRQ
	Reset value																												0	0	0	0	0
0x1C	**USART_ISR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REACK	TEACK	WUF	RWU	SBKF	CMF	BUSY	ABRF	ABRE	Res.	EOBF	RTOF	CTS	CTSIF	LBDF	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE
	Reset value										0	0	0	0	0	0	0	0			0	0	0	0	0	1	0	0	0	0	0	0	0
0x20	**USART_ICR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUCF	Res.	Res.	CMCF	Res.	Res.	Res.	Res.	EOBCF	RTOCF	Res.	CTSCF	LBDCF	Res.	TCCF	Res.	IDLECF	ORECF	NCF	FECF	PECF
	Reset value												0			0					0	0		0	0		0		0	0	0	0	0
0x24	**USART_RDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDR[8:0]								
	Reset value																								0	0	0	0	0	0	0	0	0
0x28	**USART_TDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDR[8:0]								
	Reset value																								0	0	0	0	0	0	0	0	0

寄存器边界地址请参见 35 页的章节 2.2.2

25 串行外设接口 / I2S 音频（SPI/I2S）

25.1 简介

可以使用的 SPI / I2S 接口和外部设备进行基于 SPI 协议和 I2S 音频协议的通信。 SPI 或 I2S 模式可通过软件选择。 器件复位后默认选中 SPI 模式。

串行外设接口（SPI）协议，支持半双工，全双工和简单同步等方式与外部设备的串行通信。 该接口可配置为主机模式，在这种情况下，它向外部的从属设备提供通信时钟（SCK）。 界面也能够以多重配置的方式操作。

I2S 音频协议，也是一个同步串行通信接口，使用 3 个外部信号。 它可以解决四个不同的音频标准，包括飞利浦的 I2S 标准的 MSB 和 LSB 对齐的标准和 PCM 标准。 它可以实现主从模式的半双工通信。 I2S 作为主设备的时候可以向外部从设备的提供通信时钟。

25.1.1 SPI 主要特点

- 主设备或从设备模式
- 三线全双工同步传输
- 两条线的半双工同步传输（双向数据线）
- 两条线的简单同步传输（单向数据线）
- 4 位到 16 位数据的大小选择
- 多重模式的能力
- 8 个主模式波特率分频器，波特率高达 fPCLK/ 2。
- 从模式频率高达 fPCLK/ 2。
- 主机和从机模式下都可以由硬件或软件管理 NSS： 主 / 从模式操作的动态变化
- 可编程时钟极性和相位
- 高位在前或低位在前可设置
- 专用的发送和接收状态标志，全部支持中断触发
- SPI 总线忙状态标志
- SPI 摩托罗拉方式支持
- 硬件 CRC 功能实现可靠的通信：
  - CRC 值可以作为 TX 模式的最后一个字节传送
  - 自动 CRC 错误检查上次接收到的字节
- 主模式故障，溢出等标志具备中断触发能力
- CRC 错误标志
- 两个 32 位嵌入式 Rx 和 Tx FIFO 带 DMA 功能

25.1.2 SPI 扩展功能

- SPI TI 模式支持

25.1.3 I2S 特性

- 单工通信（仅发送或接收）
- 主或从操作
- 8 位线性可编程分频器，以达到精确的音频采样频率（从 8 kHz 到 96 kHz）
- 数据格式可以是 16 位，24 位或 32 位
- 音频通道固定数据帧为 16 位（16 位数据帧）或 32 位（16 位，24 位，32 位数据帧）
- 可编程的时钟极性（稳态）
- 从发送模式的下溢标志和接收功能（主或从）的溢出标志
- 16 位的发送和接收寄存器，在通道两端各有一个寄存器
- 支持的 I2S 协议：
  - I2S 飞利浦标准
  - MSB 对齐标准（左对齐）
  - LSB 对齐标准（右对齐）
  - PCM 标准（16 位通道帧上带长的或短的帧同步或者 16 位数据帧扩展为 32 位通道帧）
- 数据方向始终是 MSB 在先
- 发送和接收的 DMA 支持（16 位宽）
- 主时钟可以向一个外部的音频组件输出。 速率固定为 256 倍的 FS（其中 FS 是音频采样频率）

25.2 SPI/I2S 具体功能配备

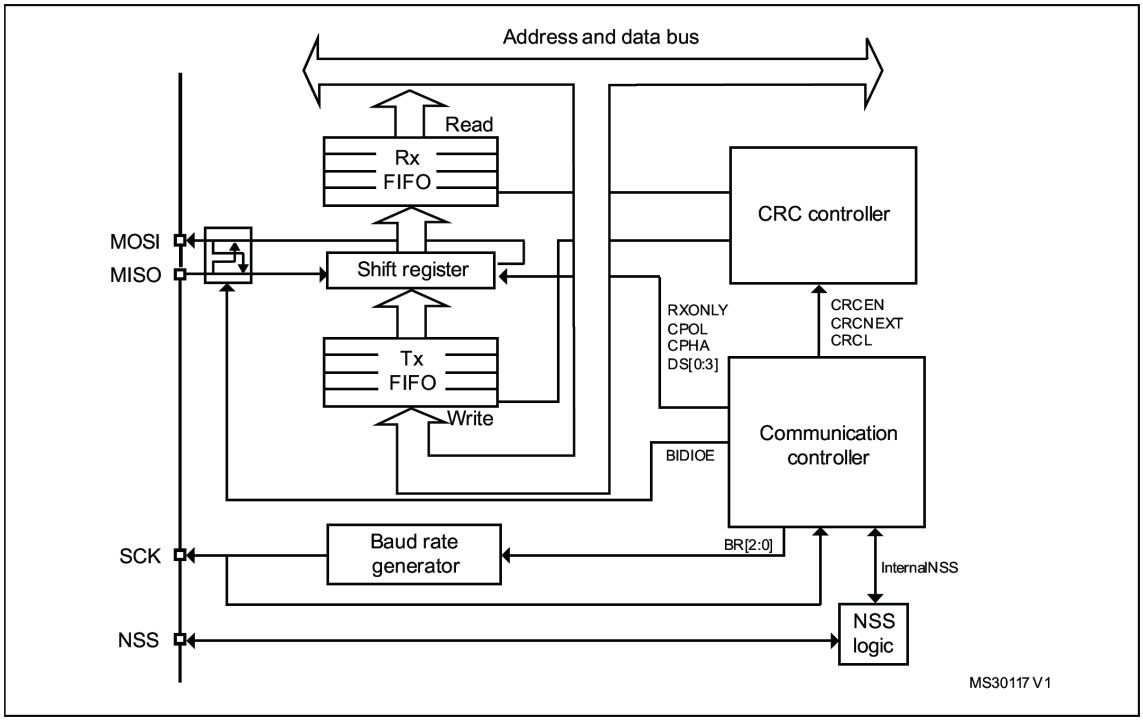
这里描述了 SPI1 所实现的全套功能。 SPI2 的支持除了 I2S 模式外的所有功能。

25.3 SPI 功能描述

25.3.1 一般说明

SPI 允许 MCU 和外部设备之间的同步串行通信。 应用软件可以通过状态标志，或使用专用的 SPI 中断来管理通信过程。 SPI 和它们之间的相互作用的主要内容见下面的框图，图 251 所示。

图 251. SPI 框图



通常 SPI 通过 4 个引脚与外部设备相连。

- MISO： 主设备输入 / 从设备输出引脚。 一般情况下，此引脚用于在从模式下的数据发送和主模式下的数据接收。
- MOSI： 主设备输出 / 从设备输入引脚。 一般情况下，此引脚用于在从模式下的数据接收和主模式下的数据发送。
- SCK： SPI 串行时钟输出引脚，作为主设备输入，从设备的输出。
- NSS： 从机片选脚。 根据 SPI 和 NSS 设置，此引脚可用于：
  - 选择一个从机来通信
  - 同步数据帧或者
  - 检测多个主机之间的冲突

  参见第 25.3.4 节： 从机片选（NSS）引脚管理。

SPI 总线允许一个主设备和一个或多个从设备之间的通信。 总线由至少有两条线 - 时钟信号和同步数据传输。 其他信号基于 SPI 节点和主机间的数据交换目的来添加。

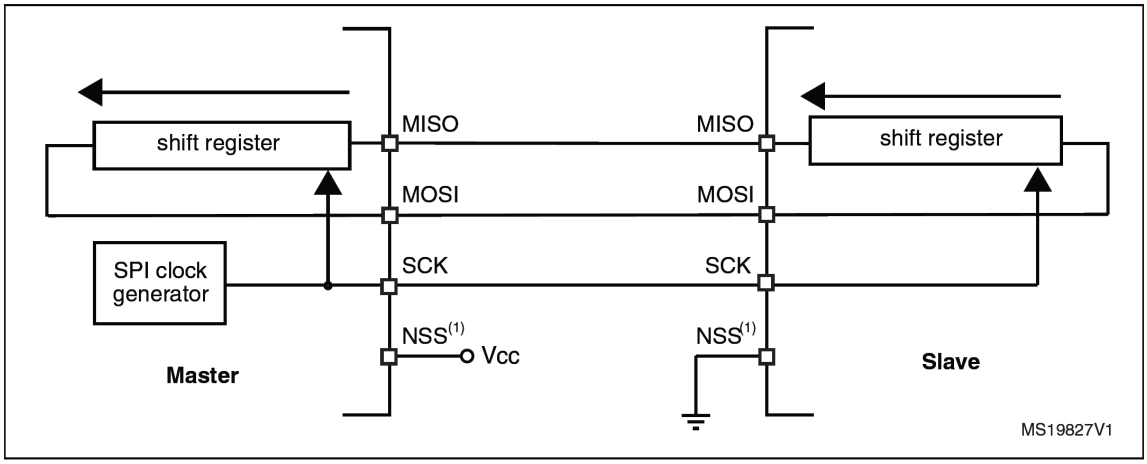
25.3.2 一个主设备和一个从机之间的通信

SPI 允许 MCU 根据目标的设备和应用程序的要求，使用不同的配置进行通信。 这些配置使用 2 或 3 线（软件 NSS 管理）或 3 或 4 线（硬件 NSS 管理）。 通信总是由主机发起的。

全双工通信

默认情况下，SPI 被配置为全双工通信。 在此配置中，主机和从机的移位寄存器通过两个单向线，MOSI 和 MISO 引脚进行连接。 在 SPI 通信时，按照主机提供的 SCK 时钟沿进行同步数据传输。 主机的数据经由 MOSI 发送到从机，从机的数据经由 MISO 线发送到主机。 当数据帧传输完成（所有位转移）时，主机和从机间的信息交换就完成了。

图 252. 全双工单主 / 单从应用

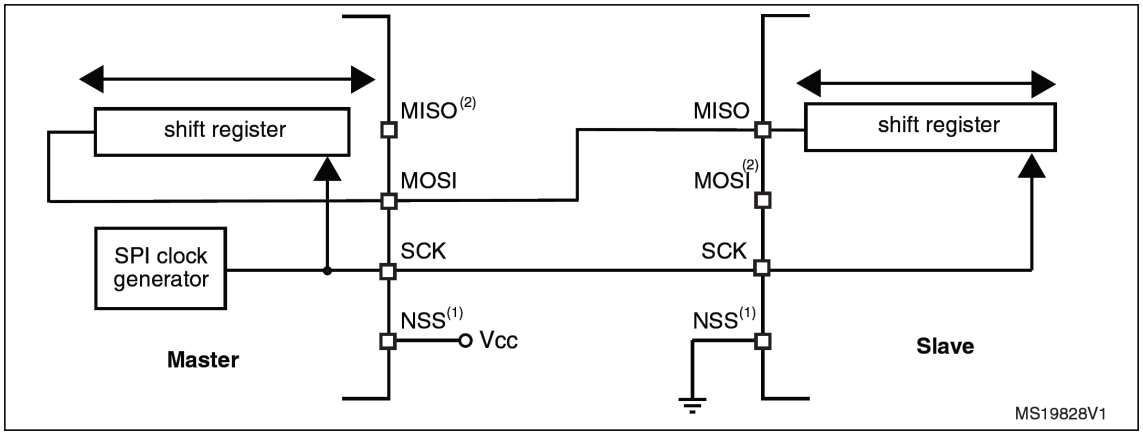


1. 在这种情况下，NSS 引脚被配置为输入。

半双工通信

通过设置在 SPIx_CR1 寄存器 BIDIMODE 位，可以令的 SPI 工作在 half-duplex 模式下。在此配置中性由一个单一的跨接线连接处主机寄存器和从机。在通信期间，数据按照 SPIx_CR1 寄存器的 BDIOE 位的设定，由主机移位寄存器同步于 SCK 时钟沿传输到从机。 在此配置中，主机的 MISO 引脚和从机的 MOSI 引脚都是自由的，可作为 GPIO 供其他应用程序使用。

图 253. 半双工单主 / 单从应用



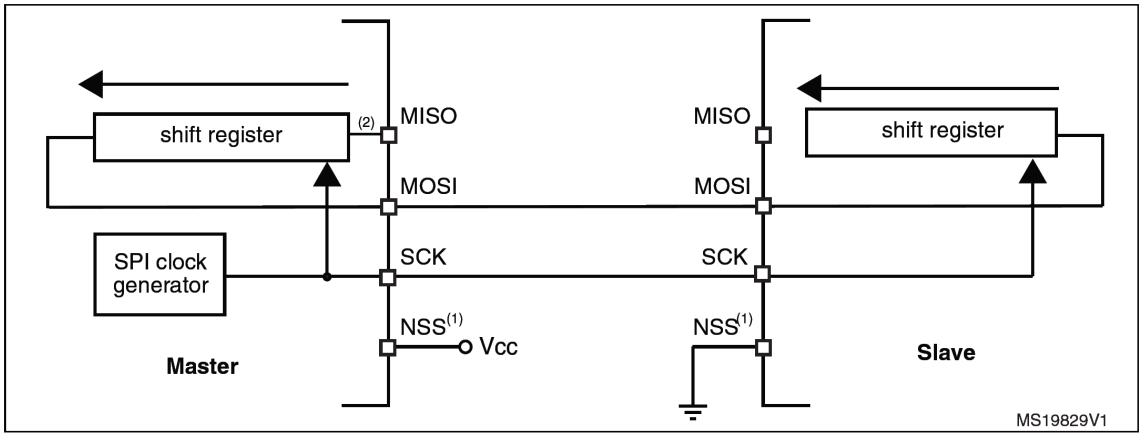
1. 在这种情况下，NSS 引脚被配置为输入。
2. 在此配置中，主机的 MISO 引脚和从机的 MOSI 引脚可以作为 GPIO 用。

简单通信

可以通过 SPIx_CR2 寄存器 RXONLY 位选择 SPI 工作在单工模式下，实现单发送或单接收通讯。 在此配置中，由一个单一的跨接线连接主机寄存器和从机。 其余 MISO 和 MOSI 引脚不用于通信，并可以作为标准的 GPIO 使用。

- 单发送模式（RXONLY = 0）： 配置设置和全双工相同。 应用程序必须忽略在未使用的输入引脚上捕获的信息。 这个脚可以作为一个标准的 GPIO 引脚。
- 单接收模式（RXONLY = 1）： 应用程序可以设置 RXONLY 位以禁用 SPI 输出功能。在配置为从机时，MISO 输出被禁用，并且可以当作一个 GPIO 引脚来使用。 从机将从 MOSI 引脚连续接收数据，在从机选择信号处于激活状态时，它的 BSY 标志总是为 1（见 25.3.4： 从机选择（NSS）的引脚管理）。根据数据缓冲区的配置，会出现数据接收事件。在配置为主机时，MOSI 输出被禁用，并且可以当作一个 GPIO 引脚来使用。 SPI 使能时，时钟信号会不断产生。 要停止时钟输出，只有清除 RXONLY 位或 SPE 位，并等待直到从 MISO 引脚的输入样式的完成并填充数据缓冲区结构，这取决于其具体配置。

图 254. 简单的单主 / 单从应用（主机仅发送 / 从机仅接收）



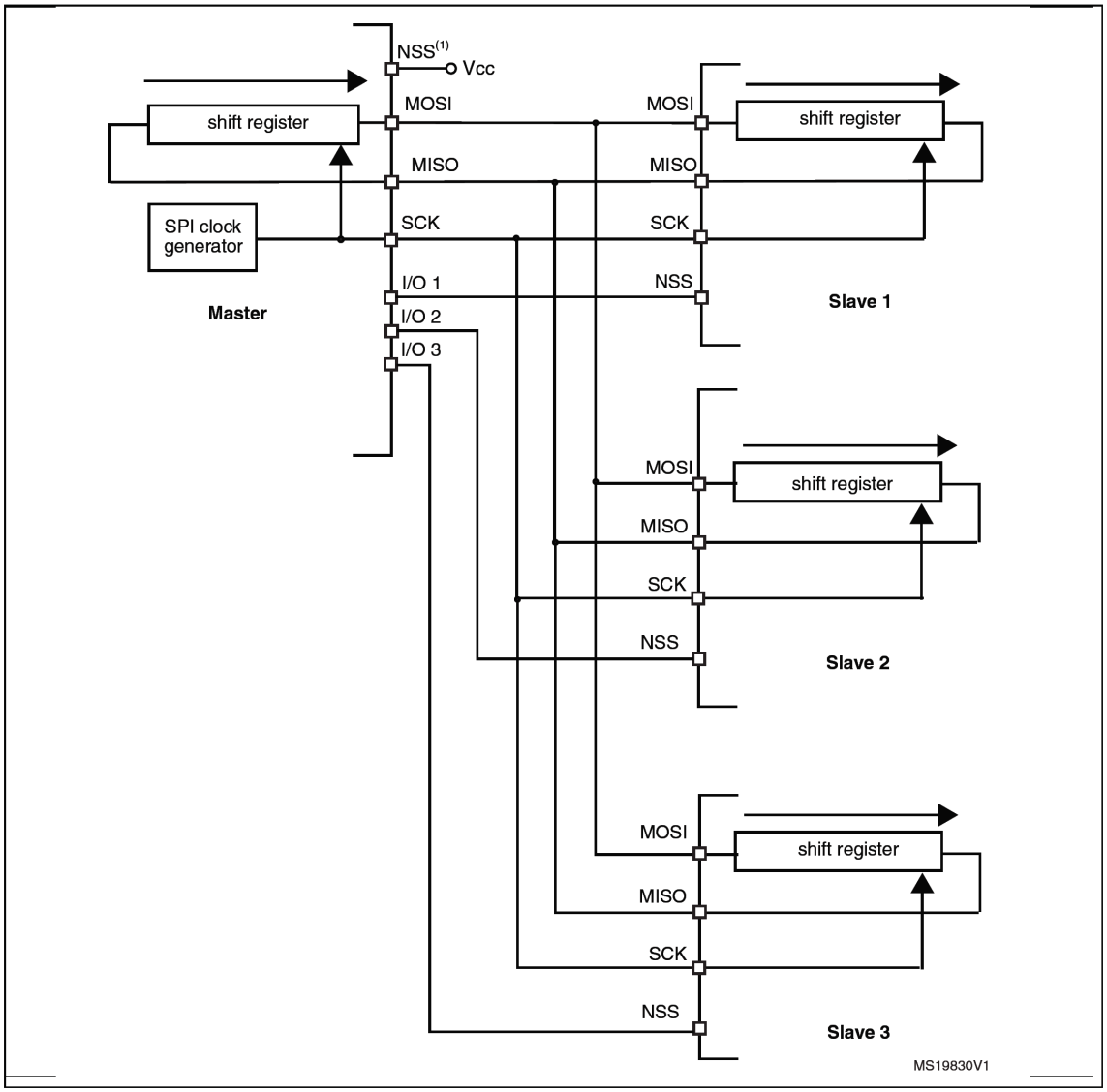
1. 在这种情况下，NSS 引脚被配置为输入。
2. 在移位寄存器中输入的信息被捕获，但在标准单发送模式下必须丢弃。（例如，OVF 格式标志）
3. 在此配置中，只要是 MISO 引脚都可用作 GPIO。

注：　简单通信模式可以被一个固定传输方向的半双工通信模式替代。

25.3.3 标准多从机通讯

在有两个或两个以上独立的从机的配置时，主机用 GPIO 引脚来管理每个从机的片选线（见图 255）。 主机必须通过 GPIO 拉低其中一个从机的 NSS 引脚。 一旦做到这一步，就会建立一个标准的主机和专用的通信渠道。

图 255. 主机和 3 个独立的从机



注：　由于从机的 MISO 引脚连接在一起，所有的从机都必须将他们的 MISO 引脚设置为备用功能漏极开路状态（见 8.3.7 节：I / O 的备用功能输入 / 输出 122 页。

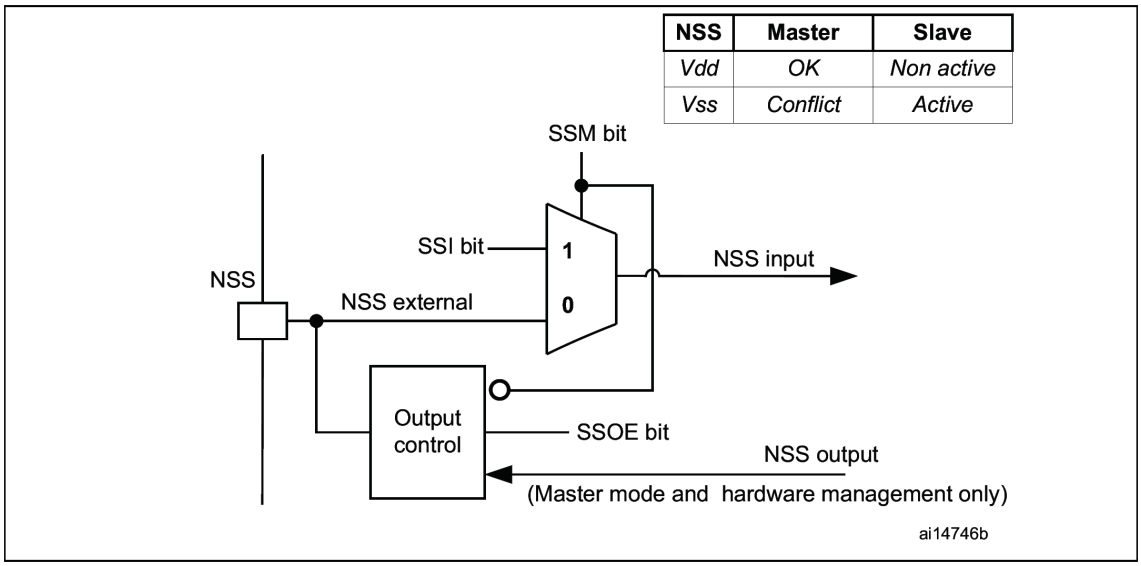
25.3.4 从机选择（ NSS）的引脚管理）

在从机模式下时，NSS 作为一个标准的“片选”输入工作，并允许主从通信。 在主机模式下，NSS 可以用来作为输入或输出。 作为输入，它可以防止多主总线冲突，作为输出，它可以驱动一个单一的从机片选信号。

可以设置 SPIx_CR1 寄存器中的 SSM 位，来选择使用硬件或软件的从机选择管理：

- 软件 NSS 管理（SSM = 1）： 在此配置中，从机选择信息由内部寄存器 SPIx_CR1 的 SSI 位值来驱动。 外部 NSS 引脚可以由其他应用程序自由支配。
- 硬件 NSS 管理（SSM = 0）： 在这种情况下，有两种可能的配置。 这种设置由（寄存器 SPIx_CR1 SSOE 位）来决定 NSS 的输出。
  - NSS 输出使能（SSM= 0，SSOE = 1）： 此配置仅用于当 MCU 作为主机时。 NSS 引脚由硬件管理。 在 SPI 作为主模式（SPE 的 = 1）的同时 NSS 信号被拉低，并保持低直到 SPI 被禁止（SPE = 0）。 如果 NSS 脉冲模式被打开（NSSP = 1），在连续通信间隔期间可以产生一个 NSS 脉冲。 在这种 NSS 设置中，SPI 不能支持多重主机工作状态。
  - NSS 输出禁止（SSM = 0，SSOE = 0）： 如果在总线上 MCU 作为主机，那么此配置就允许多主机通讯。如果在这种模式下 NSS 引脚被拉低，SPI 进入主模式故障状态，并自动重新配置为从机模式。在从机模式下，NSS 引脚作为标准的片选输入来工作，当 NSS 线被拉低时，从机被选中。

图 256. 硬件 / 软件从机选择管理



25.3.5 通讯格式

在 SPI 通信中，接收和发送操作同时进行。 串行时钟（SCK）同步发送并同时采样数据线上的信息。 通讯格式取决于时钟相位，时钟极性和数据帧格式。 为了能够正常通讯，主机和从机必须遵循相同的通信格式。

时钟相位和极性控制

可以通过 SPIx_CR1 的 CPOL 和 CPHA 位用软件选择四种可能的时序关系
CPOL（时钟极性）位控制着在没有数据在发送时的空闲状态的时钟输出电平。 该位既针对主机模式也针对从机模式。 如果 CPOL 被清零，SCK 引脚在闲置状态输出低电平。 如果 CPOL 置 1，SCK 引脚在闲置状态输出高电平。

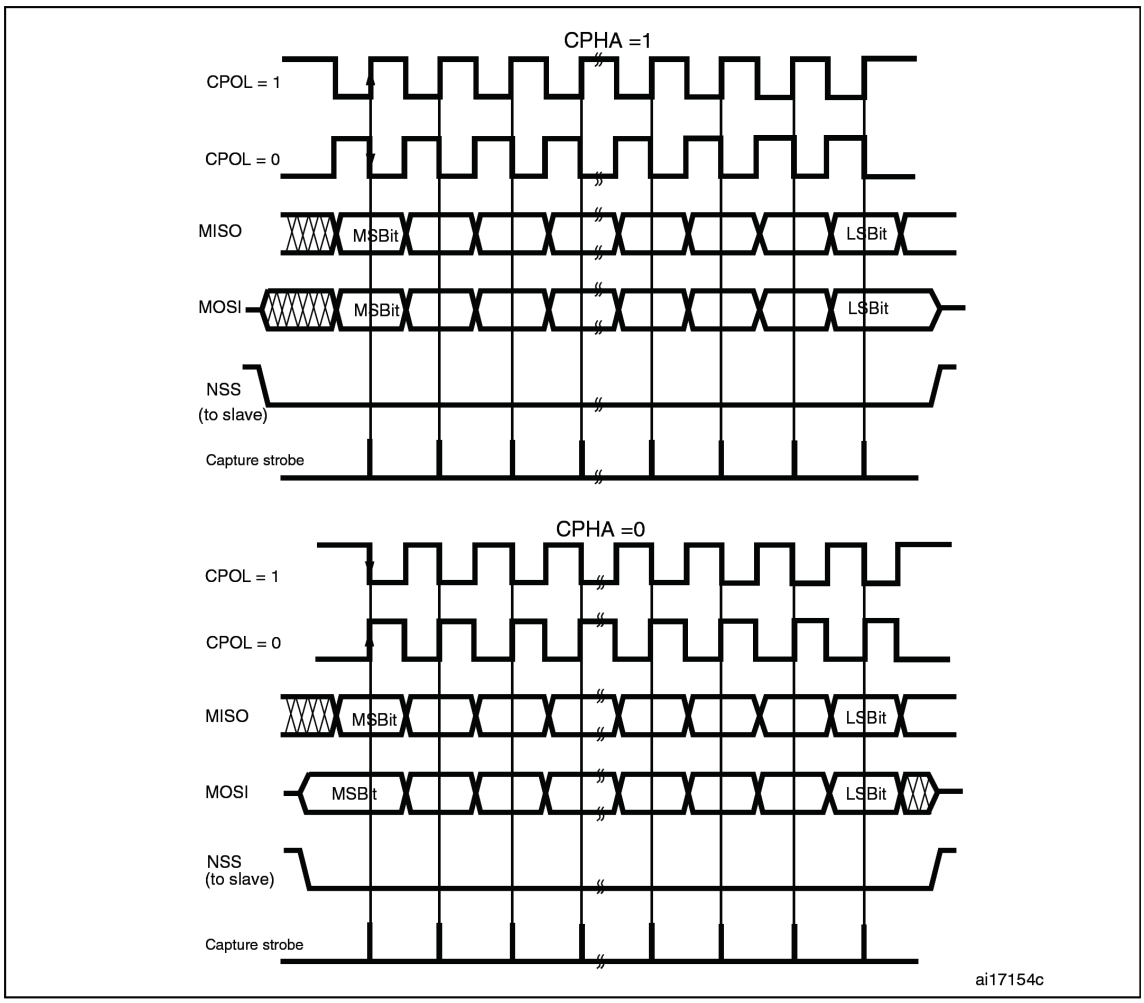
如果 CPHA 位被置 1，SCK 引脚上的第二沿对准的是第一位的捕获时机（如果 CPOL 位是 0 则为下降沿，如果 CPOL 位为 1 则为上升沿）。 每次发生这个时钟切换时，数据被锁存。 如果 CPHA 位为 0，SCK 引脚上的第个沿对准第一位的捕获时机（如果 CPOL 位为 1，则为下降沿，如果 CPOL 位为 0，则为上升沿）。 每次发生这个时钟切换时，数据被锁存。

CPOL（时钟极性）和 CPHA（时钟相位）的选择共同决定数据采集时的时钟边沿。

图 257，显示 CPHA 和 CPOL 的四种组合的 SPI 全双工传输

注： *1. 改变的 CPOL / CPHA 位之前，SPI 必须通过清零 SPE 位先关闭。*
2. 在 SCK 的空闲状态，必须符合在 SPIx_CR1 寄存器中对极性的选择（如果 CPOL = 1 上拉 SCK，如果 CPOL = 0，下拉 SCK）。

图 257. 数据时钟时序图

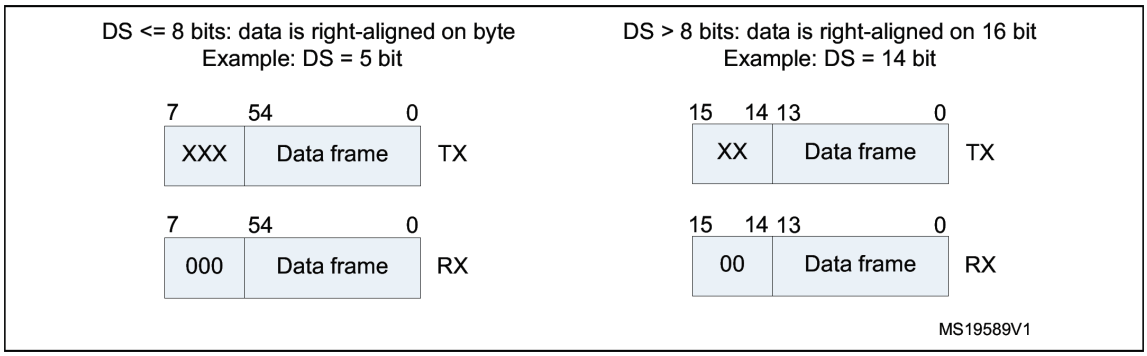


1. 这些时序显示的是 SPIx_CR1 寄存器的 LSBFIRST 位为 0 且 8 位字长（DS [3:0] = 0111）时的情形。

数据帧格式

SPI 移位寄存器可以设置 MSB 在前或 LSB 在前，取决于的 LSBFIRST 位的值。 数据字长使用 DS 位选择。它可以设定从 4 位到 16 位的长度，同时适用于发送和接收。无论选定多大字长，对 FIFO 的读访问必须与 FRXTH 水平对齐。 当访问 SPIx_DR 寄存器时，无论是一个字节（如果数据能够在一个字节中）还是一个字，数据帧总是右对齐（见图 258）。 通讯时，只有数据字长范围内的位会随时钟输出。

图 258. 当数据长度不等于 8 位或 16 位时的数据对齐



注： 最小的数据长度为 4 位。 如果选择数据长度小于 4 位，会强制按照 8 位的数据字长。

25.3.6 SPI 的初始化

主机和从机的初始化过程几乎是相同的。 都是设置位配置寄存器 SPIx_CR1 和 SPIx_CR2：

1. 使用 BR [2:0] 位选择串行时钟的波特率（见注 1）
2. 设置 CPOL 和 CPHA 位的组合，定义数据发送和串行时钟之间的四个关系（见图 257 注 2）
3. 配置 RXONLY，BIDIOE 和 BIDIMODE 选择传输模式（见注 3）。
4. 设置 DS 位选择用于传送的数据长度。
5. 配置 LSBFIRST 位，定义帧格式（见注 2）。
6. 根据应用的需要，设置 SSM 的，SSI 和 SSOE。 在主机模式下，内部 NSS 信号必须在完整序列期间保持为高（见第 25.3.4： 从机选择引脚管理（NSS）637 页）。 在从机模式下，内部 NSS 信号必须在完整序列期间保持为低（见注 2）。
7. 如果需要按 TI 协议，设置 FRF 位（见第 25.4.2： TI 模式，第 647 页）。
8. 如果两个数据单位之间需要 NSS 脉冲，那么设置 NSSP 位以打开 NSS 脉冲模式。 此配置下 CHPA 位必须设置为 1（见注 2）。
9. 设置 FRXTH 位。 RXFIFO 的阈值必须和对 SPIx_DR 寄存器的读访问的字长对齐。
10. 如果使用 DMA，初始化 LDMA_TX 和 LDMA_RX 位。
11. 如果需要的 CRC，将 CRC 多项式设置为“输入”并设置 CRCEN 位。
12. 在 NSS 内部信号为高的时候设置 MSTR 位（见注 1 和第 25.3.4： 从机选择（NSS）引脚的管理）
13. 设置 SPE 位，启用 SPI（见注 3）。

注： 1. 这个步骤在从模式下不需要。

2. 这个步骤在 TI 模式中不要求

3. 在任何主机单接收模式（RXONLY = 1 或 BIDIMODE 的 = 1 & BIDIOE = 0），时钟在 SPI 使能后立即开始运行。

25.3.7 数据发送和接收流程

RXFIFO 和 TXFIFO

所有 SPI 数据换都通过 32 位的嵌入式 FIFO。 这使 SPI 可以连续工作，防止短数据帧时的数据断流。 每个方向都有它自己的 FIFO 称为 TXFIFO 和 RXFIFO。 这些 FIFO 被用于除了单接收 +CRC 模式外的所有的 SPI 模式（主从）（见第 25.4.3： CRC 计算）。

对 FIFO 的处理会根据数据交换模式（双工，单工），数据帧格式（字长），对 FIFO 数据寄存器访问的大小（8 位或 16 位），以及是否按照数据包访问 FIFO（见第 25.4.2： TI 的模式）。

对 SPIx_DR 寄存器的读访问，会返回存储在 RXFIFO 中但还未读的最老的数据。 对 SPIx_DR 的写访问会将新的数据放在发送队列的尾部。 读访问必须总是和 SPIx_CR2 寄存器中的 FRXTH 位设置的 RXFIFO 门限对齐。 FTVL [1:0] 和 FRLVL [1:0] 位，表明两个 FIFO 目前的缓冲存储程度。

对 SPIx_DR 寄存器的读访问必须由 RXNE 事件引发。 这个时间在数据被存入 RXFIFO 并达到门槛（由 FRXTH 位定义）的时候被触发。 在 RXNE 被清除时，认为是空的 RXFIFO 已经清空。 类似的，写访问要由 TXE 事件来引发。 当 TXFIFO 的存储状况少于或等于满额的一半的时候，将触发这个事件。 否则 TXE 被清零，并且 TXFIFO 被认为是有数据。 用这种方式，RXFIFO 可以存储多达 4 个数据帧，而 TXFIFO 在字长不大于 8 的时候也只可以存储多达三个数据帧。 这种差异可以防止在已经有 3 个 8 位数据存在 TXFIFO 中的时候，软件试图在 16 位模式下向 TXFIFO 写入更多的数据而造成数据破坏。 TXE 和 RXNE 事件都可以通过中断查询或处理。

管理数据交换的另一种方法是使用 DMA（见第 25.6.8： DMA 功能）。

如果接收到下一个数据时 RXFIFO 的是满的，将导致发生溢出事件（见第 25.3.8 关于 OVR 标志的描述： 状态标志）。 溢出事件可以查询处理或中断处理。

正在执行数据传输的时候，硬件会将 BSY 位置 1 来指示这个状态。 当时钟信号连续运行时，如果是主机模式，BSY 标志在帧与帧之间一直保持为 1，而在从机模式时，BSY 信号在帧与帧之间会有一小段时间（1 个 SPI 时钟周期）为 0。

序列处理

多个数据字节可以顺序发送来组成一个消息。 启用发送后，在主机 TXFIFO 中的数据会开始发送并连续发完。 主机会连续输出时钟信号，直至 TXFIFO 变为空，然后停下来等待新增的数据。

在单接收模式，半双工（BIDIMODE = 1，BIDIOE = 0），或简单发送（BIDIMODE = 0，RXONLY = 1）模式下，只要 SPI 和单接收模式被使能，主机会立即开始接收序列。 主机连续提供时钟信号，直到主机关闭 SPI 或者关闭单接收模式之前都不会停下来。 这时主机会连续接收数据。

数据帧开始后，从机无法控制或延迟数据序列。 出于这个原因，从机必须在开始传输之前准备好数据，并总是一直保持有数据待传（存在 TXFIFO 中）。 主机必须在每个序列之间给从机保留足够的时间以准备数据。 如果可能的话，序列中的字节的数量应得到限制，以便从机完成自动的数据处理（通过使用 DMA 或 FIFO）。 主机必须提供额外的时间给从机处理数据内容。

在多从机并行的系统中，每个序列应该由 NSS 脉冲来分隔，以将每个序列对应到不同的从机。 在单从机系统中通过 NSS 控制从机就显得没那么有必要了，但这里有个脉冲还是更好，这可以令从机和每个数据序列完成同步。 NSS 既可以由软件管理也可以由硬件管理（见第 25.3.4： 从机选择（NSS）的引脚管理）。

BSY 位被置 1 时，它标志着一个持续的传输。 这一点，结合 FTLVL [1:0] 位，可用于检查传输是否完成。 在系统进入 HALT 模式前这是很有必要的，过早的进入 Halt 模式可能导致数据破坏。 判断 BSY 位的另外一个目的是可以用关键来管理 NSS 信号。 当 RXNE 标志被置 1，它意味着对当前的传输结束了。 即最后一位刚刚采样完毕，以及完整的数据帧存储在 RXFIFO 中。

当停止提供传输数据后，主机会完成当前数据传输。 在这种情况下，最后一个数据传输结束后时钟输出就会停止。 在数据包模式中，奇数个数据传输完毕后要特别注意防止出现空的字节（见第 25.4.2： TI 模式）。 当主机处于单接收模式下，只有禁用 SPI 或单接收模式，才能停止时钟。为了收到正确数量的字节并且阻止任何无效的空数据，就得在最后一个字节正在传输的时候，把握正确的关闭接收的时机。 关闭动作必须发生在首位的采样时间和下一个字节（不要的空数据）的首位之间。

禁用 SPI 的步骤

在一帧数据正在传输的时候，或者 TXFIFO 当中有数据的时候，主机不可以通过操作 SPE 位来关闭 SPI。 如果发生这种情况，时钟信号将持续发送，直到外设被重新启用使得传输可全面完成。 如果收到了数据，但是仍然遗留在 RXFIFO 中未及读取时关闭了 SPI 的话，那么在下一次打开 SPI 开始新的传输序列之前一定要先处理历史数据。 为了防止这种情况，请确保 RXFIFO 的是空的时候再去禁用 SPI。 这个过程可以通过正确的流程来实现，或者通过专门的外设复位控制寄存器来执行软件复位命令，从而全面初始化 SPI 的寄存器（见 RCC_APBiRSTR 中的 SPIiRST 位）。

正确的关闭流程是（除了使用单接收模式时）：

1. 等到 FTLVL [1:0] = 00（没有更多的数据传输）
2. 等待，直到的 BSY = 0（最后一个数据帧处理完毕）
3. 读取数据直到 FRLVL [1:0] = 00（读取所有接收到的数据）
4. 禁止 SPI（SPE 的 = 0）。

某些单接收模式的正确关闭步骤是：

1. 在最后一个数据的传输期间的特定时间窗口中关闭单接受模式（RXONLY = 0 或 BIDIOE= 1）
2. 等待直到 BSY = 0（最后一个数据帧处理完毕）
3. 读取数据，直到 FRLVL [1:0] = 00（读取完所有接收到的数据）
4. 禁止 SPI（SPE= 0）。

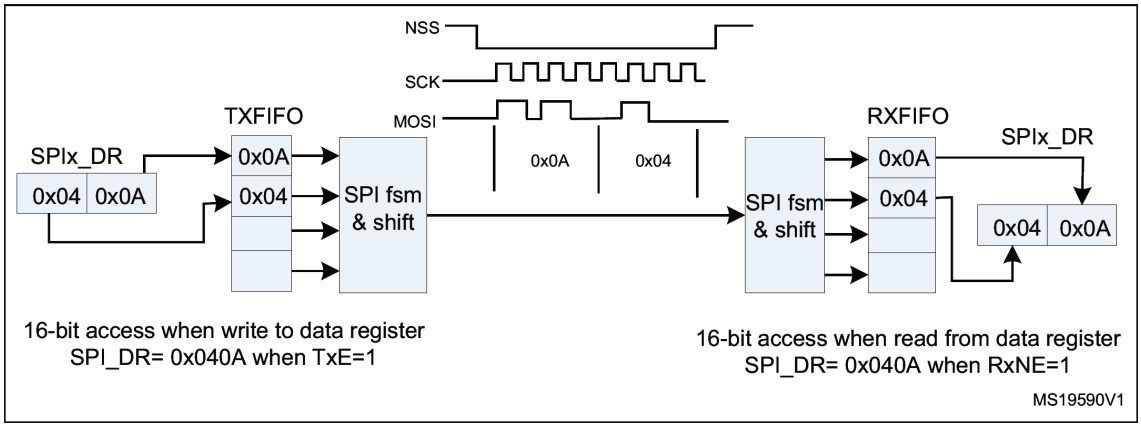
注： 如果用了打包模式，并且收到一个格式为小于或等于 8 位的奇数个数据的数据帧，当 FRLVL [1:0] = 1 时 FRXTH 必然被置 1，为的是产生 RXNE 事件以便去读最后的奇数数据。

数据打包

当数据帧的大小适合一个字节（小于或等于 8 位），并且对 SPIx_DR 寄存器执行任何 16 位的读写访问时，数据会自动打包在一起。 在这种情况下，可以并行处理双数据。 SPI 先操作低 8 位，然后操作高 8 位。 图 259 提供了打包方式顺序处理数据的一个例子。 在一次对 SPIx_DR 的 16 位写访问后，就会有 2 个字节的数据被发送出去。 如果 RXFIFO 的阈值设置为 16 位（FRXTH = 0），该序列则只会生成一个接收 RXNE 事件，而不是两个。 针对这种单个的 RXNE 事件的响应，接受器必须对 SPIx_DR 寄存器作一次 16 位的读访问才能够把数据全都取到。 RxFIFO 的阀值和跟进的数据访问的位宽必须保持一致，否则就会丢数据。

如果出现奇数个字节数据，那就会出现特别的问题，这是一定要解决的。 在发送端，用 8 位方式访问 SPIx_DR 将最后一个字节发出来就够了。 在接收端必须改变 Rx_FIFO 门限，以便在传送技术字节的数据帧的最后字节时能够产生 RXNE 事件。

图 259. 传输和接收 FIFO 中的数据打包



使用 DMA（直接内存访问）通信

为了使 SPI 运行在其最高通讯速度，并且在方便数据寄存器读 / 写过程的同时避免溢出，SPI 需要用 DMA 支持，它实现了一个简单的请求 / 应答协议。

当在 SPIx_CR2 寄存器的 TXE 或 RXNE 位被置 1 时，可以产生一个 DMA 访问请求。 Tx 和 Rx 缓冲区的 DMA 请求是单独的。

- 在发送中，每次的 TXE 被置为 1，发出 DMA 请求。 然后 DMA 向 SPIx_DR 寄存器写数据。
- 在接收中，每次的 RXNE 被置为 1 时，发出 DMA 请求。然后 DMA 从 SPIx_DR 寄存器读数据。

当 SPI 仅用于发送数据，可以只打开 SPI TX DMA 通道。在这种情况下，溢出标志会不断置 1，因为收到的数据不会被读取。 当 SPI 仅用于接收数据，可以只打开 SPI Rx DMA 通道。

在发送模式下，当 DMA 传输完所有要发送的数据时，DMA 控制器中 DMA_ISR 寄存器的 TCIF 标志会被置 1；监视 USART_SR 寄存器的 BSY 标志可以确认 SPI 通信是否结束。 这样可以在关闭 SPI 或进入停机模式之前避免破坏最后一次传输的数据。 软件必须先等待，直到 FTLVL [1:0] = 00，然后等 BSY = 0。

DMA 与打包传输

如果传输由 DMA 来管理（TXDMAEN 和 RXDMAEN 在 SPIx_CR2 寄存器设置），会根据 SPI TX 和 RX DMA 通道的 SPI 配置状态来自动将包装模式启用 / 禁用。 如果 DMA 通道设置为按 16 位访问，而 SPI 数据的大小是小于或等于 8 位，那么打包模式会被启用。 DMA 会自动管理对 SPIx_DR 寄存器的写操作。

如果启用了打包模式，而传输的数据个数又不一定是偶数，则 LDMA_TX / LDMA_RX 位必须置 1。 而 SPI 则会认为在最后的 DMA 传输中只有一个有效的传输或接收字节（更多详情，请参阅第 643 页数据打包。）

25.3.8 状态标志

应用程序可通过 3 个状态标志来了解 SPI 总线的全部状态。

TX 缓冲器空标志（TXE）

当 TXFIFO 中有了足够的空间来存储发送数据时，TXE 标志被置 1。 TXE 标志是连到 TXFIFO Level 的。在 TXFIFO 的的存储水平低于或等于整个 FIFO 的深度的一半的时候，这个标志会置高并且保持高。如果 SPIx_CR2 中的 TXEIE 位是 1，还会产生一个中断。如果 TXFIFO 的存储水平又高于 FIFO 深度的一半了，那这个位会自动的清零。

Rx 缓冲非空（RXNE）

RXNE 标志的设置取决于对在 SPIx_CR2 寄存器 FRXTH 位值：

- 如果 FRXTH 为 1，RXNE 会在 RXFIFO 的存储水平大于或等于 1/4（8 位）的时候被置高，并且保持住。
- 如果 FRXTH 为 0，RXNE 会在 RXFIFO 的存储水平大于或等于 1/2（16 位）的时候被置高，并且保持住。

如果这时 SPIx_CR2 寄存器中的 RXNEIE 位是 1，那就会产生一个中断请求。
当上述条件不再成立时 RXNE 由硬件自动清零。

忙标志（BSY 变）

BSY 标志由硬件设置和清零（软件改写这个标志是没有用的）。
当 BSY 为 1 时，它表示 SPI 正处于数据传输过程中（SPI 总线忙）。

在某些模式下可以使用的 BSY 标志来检测传输结束，从而软件可以在进入 HALT 模式之前禁用 SPI 及其外设时钟。 这就避免了破坏最后的传输字节。

BSY 标志在防止多主机系统中的写碰撞也是很有用的。 BSY 标志在下列条件之一达成时被清除：

- 当 SPI 被正确禁用时
- 当在主模式（MODF 位设置为 1）中检测到故障时
- 在主模式下，当完成了数据发送，并且没有新的数据要发送时
- 在从模式下，当两次数据传输之间间隔达到至少一个 SPI 时钟周期时

注：当下次传输可立即由主机来处理时（例如： 如果主机是单接收模式或它的发送 FIFO 不为空），在主机这边的发送数据之间通讯不会中断，并且 BSY 标志仍然被保留为 1。 尽管从机不会有这个情况，ST 总是建议使用 TXE 和 RXNE 标志（而不是标志的 BSY）来处理数据发送或接收操作。

25.3.9 错误标志

在将错误中断使能位 ERRIE 置 1 后，如果下列错误标志被置 1，就会产生 SPI 中断。

溢出标志（OVR）

当主机或从机在收到数据之后不去清除 RXNE 位，然后收到了新的数据之后，会引发溢出错误。 当接收数据时 RXFIFO 中有没有足够的空间存储收到的数据时，溢出的情况也会发生。如果软件或者 DMA 没有足够时间去读取 RXFIFO 中的前面收到数据，来为后面的数据腾出足够的空间的时候就会发生这种事。在 CRC 功能打开时，接收缓冲区会被认为是一个单缓冲区，这导致 RXFIFO 不好使了，这也会直接导致溢出事件。（见第 25.4.3： CRC 计算）。

溢出的情况发生时，新收到的值不会覆盖在 RXFIFO 的前一个数据。 新收到的值将被丢弃，随后传输的所有数据都将丢失。 在读了 SPIx_SR 寄存器后，再去读一下 SPIx_DR 寄存器，就会清除 OVR 位。

模式故障位（MODF）

主机得知其内部 NSS 信号（NSS 引脚为硬件模式，或在 NSS 软件模式 SSI 位）被拉低时，发生模式故障位。 这将自动设置 MODF 位。 主机模式故障对 SPI 接口的影响包括以下方面：

- MODF 位被置 1，另外如果 ERRIE 位被置 1，会产生 SPI 中断。
- SPE 位被清零。 这将阻止主机的数据输出，同时禁用 SPI 接口。
- MSTR 位被清除，从而迫使设备转到从机模式。

使用下面的软件序列，以清除 MODF 位：

1. 在 MODF 位被设置后，对 SPIx_SR 寄存器做一次读或写访问。
2. 然后写一下 SPIx_CR1 寄存器。

为了避免系统中的多个从机发生冲突，NSS 引脚必须在 MODF 位清零过程中拉高。 SPE 和 MSTR 位可以在这个清零过程后恢复到原来的状态。 作为一种安全措施，硬件不允许在 MODF 位为 1 期间将 SPE 的 MSTR 位置 1。 在从模式下，MODF 位永远不可能被置 1，除非是以前的多主机冲突的结果。

CRC 错误（CRCERR）

在 SPIx_CR1 寄存器中的 CRCEN 位为 1 时，这个标志用来确认收到的数据的有效性。 如果接收移位寄存器中的值和接收 SPIx_RXCRCR 值不匹配，SPIx_SR 寄存器中的 CRCERR 标志会被置 1。 该标志由软件清除。

TI 模式帧格式错误（FRE）

当 SPI 工作在从模式并且配置方式符合 TI 模式协议的时，如果数据通讯期间，在 NSS 上出现一个脉冲时，会引起一次 TI 模式帧格式错误。 发生此错误时，SPIx_SR 寄存器中的 FRE 的标志会被置 1。 在发生错误时 SPI 不会被禁止，NSS 脉冲会被忽略，SPI 会在开始新的传输前等待下一个 NSS 脉冲。因为错误检测可能会导致两个字节的数据丢失，数据可能会损坏。

读 SPIx_SR 寄存器，FRE 的标志会被清除。 如果 ERRIE 位为 1，会在 NSS 错误发生后产生一个中断。 在这种情况下，SPI 应该被禁用，因为数据的完整性将不能保证了，通信应该重新开始。先重新使能从机，再重新初始化主机就行了。

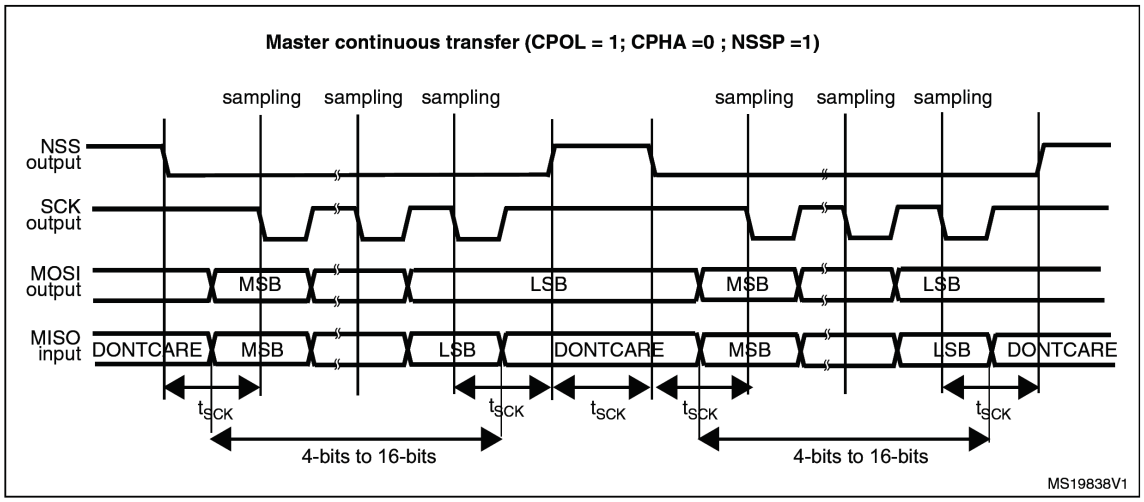
25.4 SPI 特殊功能

25.4.1 NSS 脉冲模式

在 SPIx_CR1 寄存器的 NSSP 位被置 1 时打开这个模式，只有在 SPI 接口被配置为摩托罗拉主模式，而且捕获第一个沿时（SPIx_CR1 CPHA = 0）才会生效。 这时发送时，NSS 脉冲会在两个连续的数据帧之间产生，并且 NSS 至少会在高电平保持一个时钟周期的时间。 这个模式允许从机锁存数据。 NSSP 脉冲模式为单一的主从应用而设计。

图 260 说明 NSSP 脉冲模式时启用时的 NSS 引脚管理。

图 260. 摩托罗拉 SPI 主模式下 NSSP 脉冲的产生



注： 当 CPOL = 0 时动作会类似。这时，采样边沿是在 SCK 的上升沿，对 NSS 的采样也发生在这些沿。

25.4.2 TI 模式

TI 协议主模式

SPI 接口兼容 TI 协议。 SPIx_CR2 寄存器中的 FRF 位可以用来令 SPI 兼容这个协议。

无论 SPIx_CR1 寄存器中怎么设置，时钟极性和相位都会强制为符合 TI 协议要求。 NSS 管理也是按照 TI 协议，这使得在这种情况下通过 SPIx_CR1 和 SPIx_CR2 寄存器（SSM 的 SSOE，SSI），不可能配置 NSS 管理。

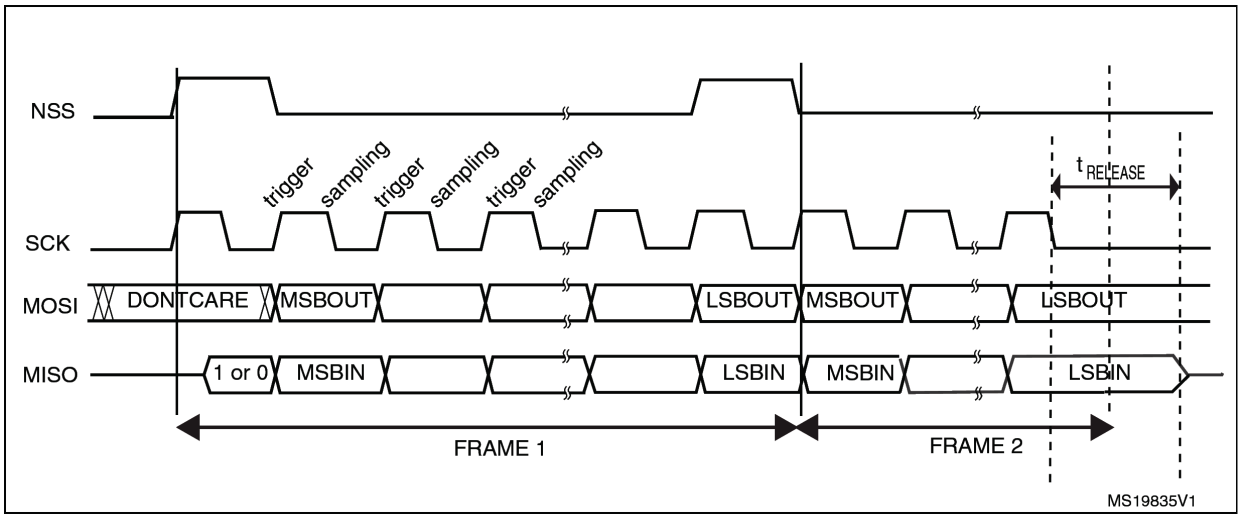
在从模式下，SPI 波特率分频器用于控制 MISO 引脚状态变化为高阻的时刻。 任何波特率都可用，这使其能够确定最佳的灵活性。 然而，波特率一般设置为外部主时钟的波特率。 The delay for the MISO signal to become HiZ (trelease) depends on internal resynchronization and on the baud rate value set in through the BR[2:0] bits in the SPIx_CR1 register. It is given by the formula:

$$\frac{t_{baud\_rate}}{2} + 4 \times t_{pclk} < t_{release} < \frac{t_{baud\_rate}}{2} + 6 \times t_{pclk}$$

此功能并不适用于摩托罗拉 SPI 通信模式（FRF 位设置为 0）。

图 261： TI 模式传输 显示选择 TI 模式时 SPI 的通信波形。

图 261. TI 模式传输



25.4.3 CRC 计算

有两个独立的 CRC 计算器，分别用以检查发送和接收的数据的可靠性。 SPI 提供 CRC8 或 CRC16 计算，独立于帧的数据的位宽，它可以固定设置为 8 位或 16 位。 对于其他所有的数据位宽，CRC 无效。

CRC 原则

在使能 SPI 之前（SPE=0）通过设置 SPIx_CR1 寄存器中的 CRCEN 位，使能 CRC 计算。 CRC 值由一个奇次可编程多项式按每位计算。 计算处理从采样时钟源开始执行，这个时钟沿由 SPIx_CR1 寄存器的 CPHA 和 CPOL 位定义。 在数据块传输结束的同时，计算出的 CRC 结果自动参与检查，由 CPU 或 DMA 控制传输的时候都一样。当发现 CRC 不匹配现象的时候，CRCERR 标志被置 1，表示 CRC 错误。 对 CRC 计算的正确处理过程取决于 SPI 配置和所选择的传输管理。

注： 多项式的值应该是奇数。 不支持偶数值。

由 CPU 管理传输时的 CRC

在最后一个数据发送或接收完之前，SPI 的启动和传输都没什么特别的。

要在当前传输的数据之后跟上 CRC 数据，必须将 SPIx_CR1 中的 CRCNEXT 位设置为 1。 设置 CRCNEXT 位的操作必须在最后一个数据帧交易结束之前完成。 在 CRC 数据传输期间，CRC 计算会被冻结。

接收到的 CRC 数据像正常的数据字节或字存储在 RXFIFO 中。 这就是在 CRC 模式下，任何时候接受缓冲区都必须被认为是单个 16 位的缓冲区，并被用来接收单个数据的原因。

CRC 数据交换，通常需要在数据序列结束后，再占用一个或多个数据通信的时间。 当设置为 8 位的数据宽度，并做 16 位 CRC 检查时，发送完整的 CRC 数据就要两帧。

当收到最后的 CRC 数据后，会自动将收到的值和在 SPIx_RXCRC 寄存器的值进行比较。软件必须坚持 SPIx_SR 寄存器中的 CRCERR 标志以判断传输过程中数据是否被破坏。 软件对 CRCERR 标志写 0 就可清除它。

CRC 数据收到后，被存储在 RXFIFO 中，必须读 SPIx_DR 寄存器以清除 RXNE 标志。

由 DMA 管理传输时的 CRC

当 SPI 通讯功能中的 CRC 功能和 DMA 功能同时打开后，通讯尾声的 CRC 数据的发送和接收是全自动的。 CRCNEXT 位不再必须由软件处理。 SPI 发送 DMA 通道计数器必须设置为不包括 CRC 数据的数量。而接受 DMA 通道计数器则要多包含一个 CRC 数据的长度，例如在 8 位数据宽度传输并使用 16 位 CRC 计算时：

DMA_RX = 数据字节数 + 2
在接收侧，接收到的 CRC 值在数据交换后依然存储在缓冲区中。 全部传输完后，如果发现有错误，SPIx_SR 寄存器中的 CRCERR 标志会被置 1。

SPIx_TXCRC 和 SPIx_RXCRC 值的重置

在 CRC 校验环节读取过 CRC 数据之后，SPIx_TXCRC 和 SPIx_RXCRC 值会自动清零。 这就允许使用 DMA 循环模式（单接收模式除外）来实现没有任何间断的传输数据（中间有几个数据块涉及 CRC 检查阶段）。

如果要在 SPI 通信过程中禁用它，必须遵循以下顺序：

- 禁止 SPI
- 清除 CRCEN 位
- 使能 CRCEN 位
- 使能 SPI

注： 当 SPI 为从模式的时候，一旦设置的 CRCEN 位，CRC 计算器对 SCK 从输入时钟敏感，无论 SPE 位的值是几，都会这样。 为了避免任何错误的 CRC 计算，软件必须在时钟稳定的条件下（稳定状态）启用 CRC 计算。 当 SPI 配置为从机接口时，数据阶段和 CRC 阶段之间，NSS 内部信号需要保持低。

25.5 SPI 中断

在 SPI 通信期间，中断可以由以下事件来产生：

- 发送 TXFIFO 待装填
- 在接收 RXFIFO 中有收到的数据
- 主模式故障
- 溢出错误
- CRC 错误
- TI 帧格式错误

中断可以分别的被启用和禁用。

表 96. SPI 中断请求

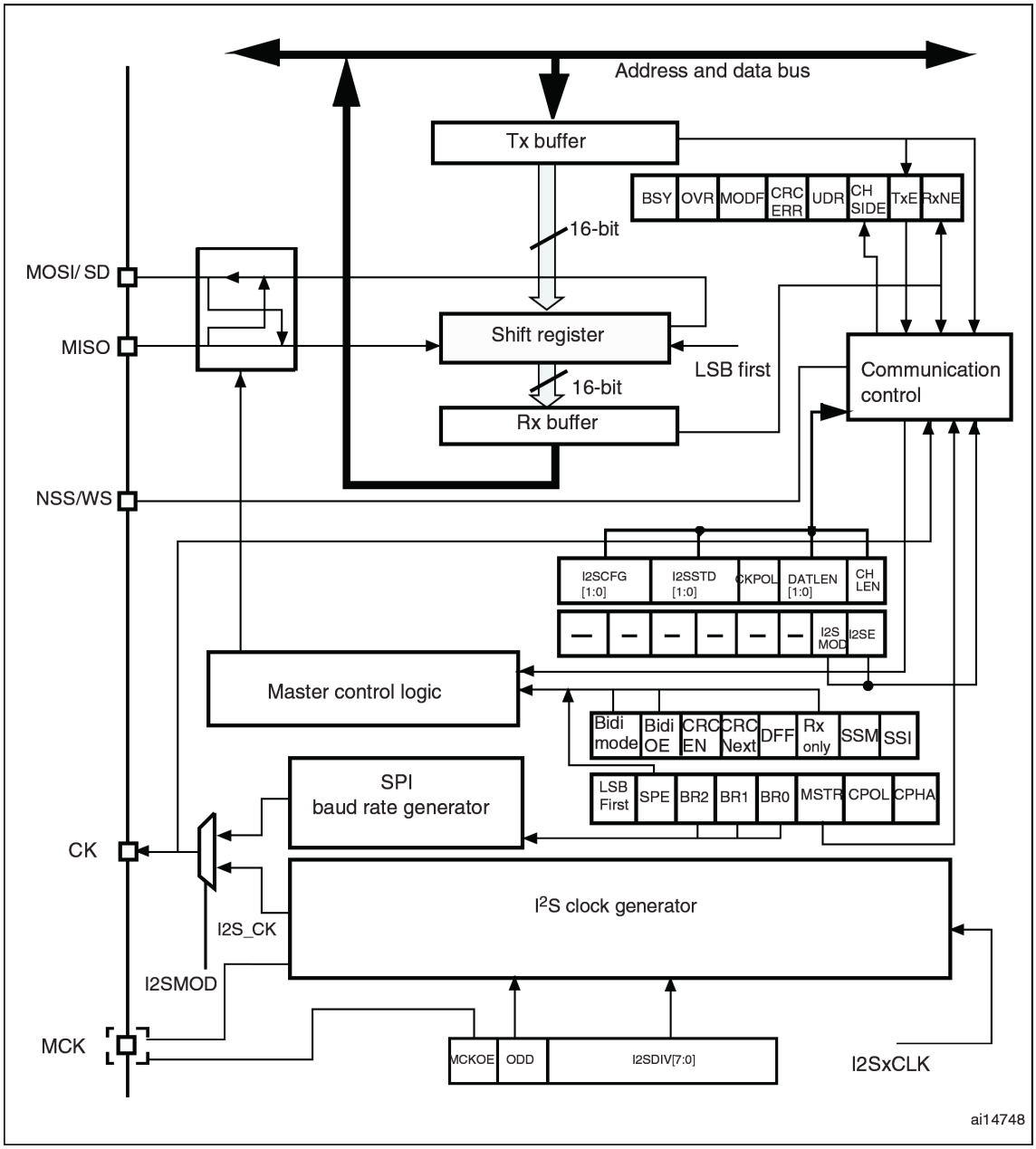
中断事件	事件标志	使能控制位
发送 TXFIFO 待装填	TXE	TXEIE
在接收 RXFIFO 中有收到的数据	RXNE	RXNEIE
主模式故障	MODF	ERRIE
溢出错误	OVR	
CRC 错误	CRCERR	
TI 帧格式错误	FRE	

25.6 I²S 的功能说明

25.6.1 I²S 的一般描述

I²S 框图如图 262 所示。

图 262. I²S 框图



I²S 功能打开时（通过设置在 SPIx_I2SCFGR 寄存器 I2SMOD 位）SPI 可以当作音频 I²S 接口使用。 这个接口使用的引脚，标志和中断和 SPI 基本相同。

SPI 的 I^2S 共用三个引脚：

- SD： 串行数据（MOSI 引脚）发送或接收的两个时分复用数据通道（仅在单工模式下）。
- WS： 字选择（NSS 引脚）主模式下作为数据控制信号输出，从动模式下作为输入。
- CK： 串行时钟（SCK 引脚）主模式下作为串行时钟输出，从模式下为输入。

在某些外部音频设备需要主时钟时，可以另有一个附加引脚输出时钟：

- 主时钟 (独立映射)，在 I^2S 配置为主模式，寄存器 SPI_I2SPR 的 MCKOE 位为’1’时，作为输出额外的时钟信号引脚使用。输出时钟信号的频率预先设置为 256 × Fs，其中 Fs 是音频信号的采样频率。

设置成主模式时，I^2S 使用自身的时钟发生器来产生通信用的时钟信号 这个时钟发生器也是主时钟输出的时钟源。 I^2S 模式下有 2 个额外的寄存器， 一个是与时钟发生器配置相关的寄存器 SPI_I2SPR，另一个是 I^2S 通用配置寄存器 SPI_I2SCFGR(可设置音频标准、从 / 主模式、数据格式、数据包帧、时钟极性等参数)。

在 I^2S 模式下不使用寄存器 SPI_CR1 和所有的 CRC 寄存器。 同样，I^2S 模式下也不使用寄存器 SPI_CR2 的 SSOE 位，和寄存器 SPI_SR 的 MODF 位和 CRCERR 位。

I^2S 使用与 SPI 相同的寄存器 SPI_DR 用作 16 位宽模式数据传输。

25.6.2 支持的音频协议

三线总线支持 2 个声道上音频数据的时分复用： 左声道和右声道。 但是只有一个 16 位寄存器用作发送或接收。 因此，软件必须在对数据寄存器写入数据时，根据当前传输中的声道写入相应的数据；同样，在读取寄存器数据时，通过检查寄存器 SPI_SR 的 CHSIDE 位来判明接收到的数据属于哪个声道。 左声道总是先于右声道发送数据 (CHSIDE 位在 PCM 协议下无意义)。

有四种可用的数据和包帧组合。 可以通过以下四种数据格式发送数据：

- 16 位数据打包进 16 位帧
- 16 位数据打包进 32 位帧
- 24 位数据打包进 32 位帧
- 32 位数据打包进 32 位帧

在使用 16 位数据扩展到 32 位帧时，前 16 位 (MSB) 是有意义的数据，后 16 位 (LSB) 被强制为 0，该操作不需要软件干预，也不需要有 DMA 请求 (仅需要一次读 / 写操作)。

24 位和 32 位数据帧需要 CPU 对寄存器 SPI_DR 进行 2 次读或写操作，在使用 DMA 时，需要 2 次 DMA 传输。 对于 24 位数据，扩展到 32 位后，最低 8 位由硬件置 0。

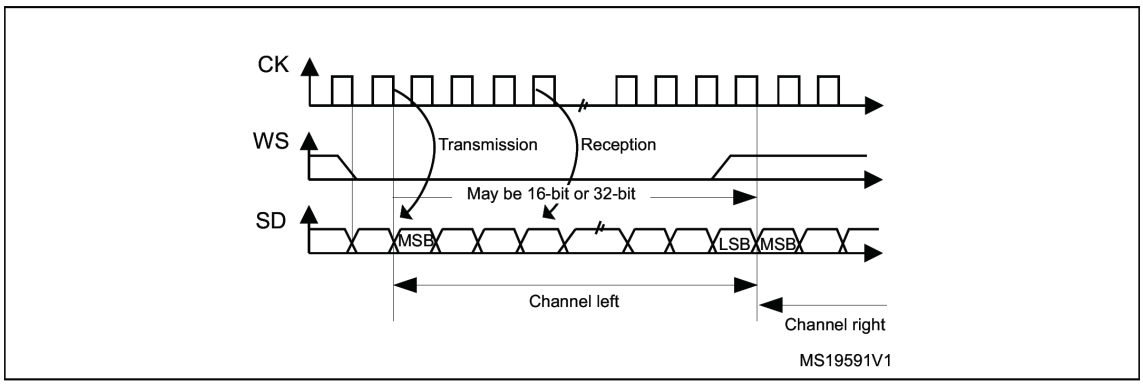
对于所有的数据格式和通讯标准，总是先发送最高位 (MSB)。

I^2S 接口支持四种音频标准，可以通过设置寄存器 SPI_I2SCFGR 的 I2SSTD[1:0] 位和 PCMSY 位来选择。

I²S 飞利浦标准

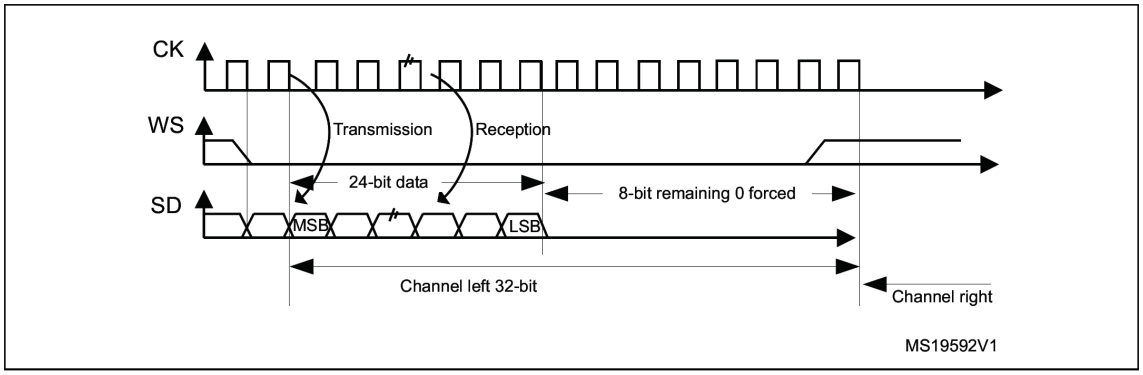
在此标准下，引脚 WS 用来指示正在发送的数据属于哪个声道。 在发送第一位数据 (MSB) 前 1 个时钟周期，该引脚即为有效。

图 263. I²S 飞利浦协议波形 (16/32 位全精度，CPOL = 0)



发送方在时钟信号 (CK) 的下降沿改变数据，接收方在上升沿读取数据。 WS 信号也在时钟信号的下降沿锁存。

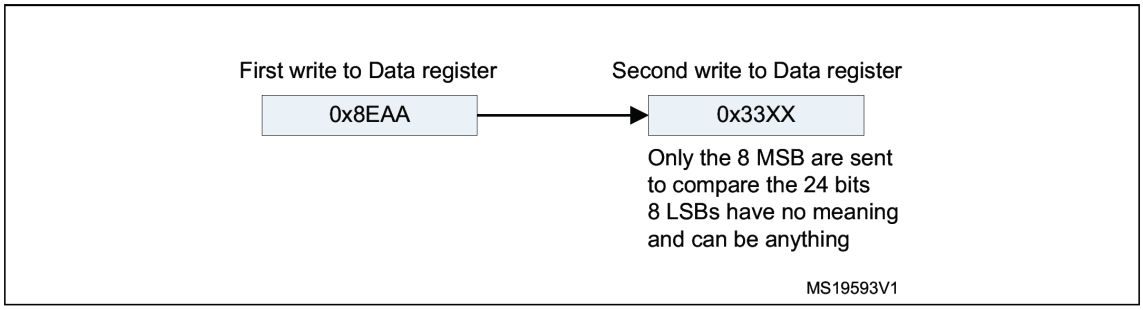
图 264. I²S 飞利浦协议标准波形 (24 位帧，CPOL = 0)



此模式需要对寄存器 SPI_DR 进行 2 次读或写操作。

- 在发送模式下：
  如果要发送 0x8EAA33（24 位）：

图 265. 发送 0x8EAA33



- 在接收模式下：
  如果接收 0x8EAA33：

图 266. 接收 0x8EAA33

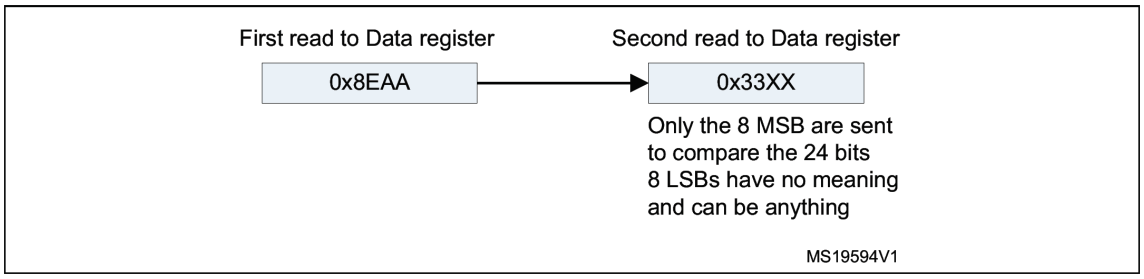
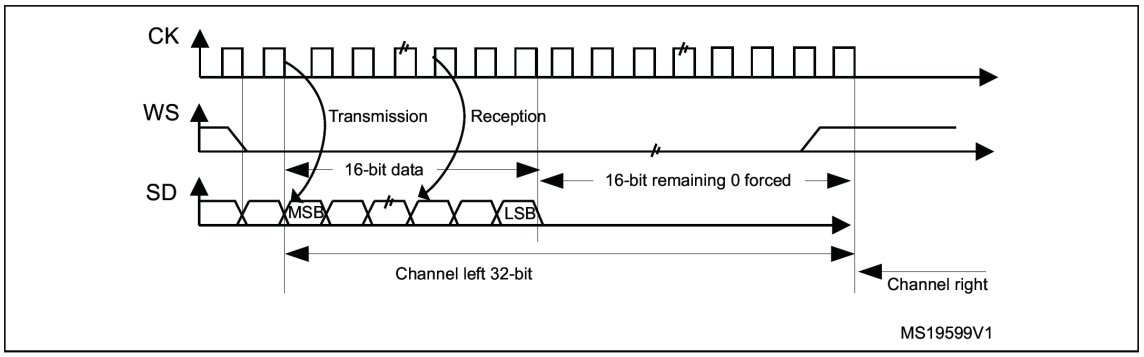


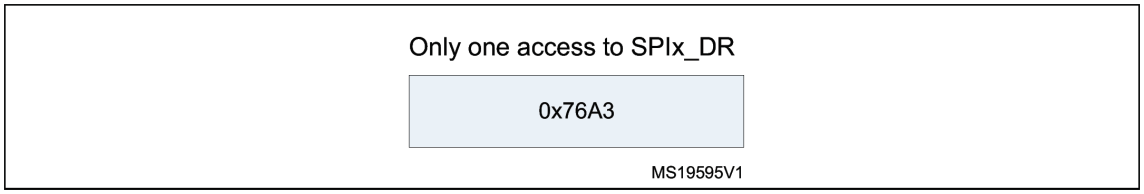
图 267. I²S 飞利浦协议标准波形 (16 位扩展至 32 位包帧，CPOL = 0)



在 I²S 配置阶段，如果选择将 16 位数据扩展到 32 位声道帧，只需要访问一次寄存器 SPI_DR。 用来扩展到 32 位的低 16 位被硬件置为 0x0000。

如果待传输或者接收的数据是 0x76A3(扩展到 32 位是 0x76A30000)，需要的操作如下图所示。

图 268. 16 位扩展至 32 位包帧的例子



在发送时需要将 MSB 写入寄存器 SPI_DR；标志位 TXE 为’1’表示可以写入新的数据，如果允许了相应的中断，则可以产生中断。 发送是由硬件完成的，即使还未发送出后 16 位的 0x0000，也会设置 TXE 并产生相应的中断。

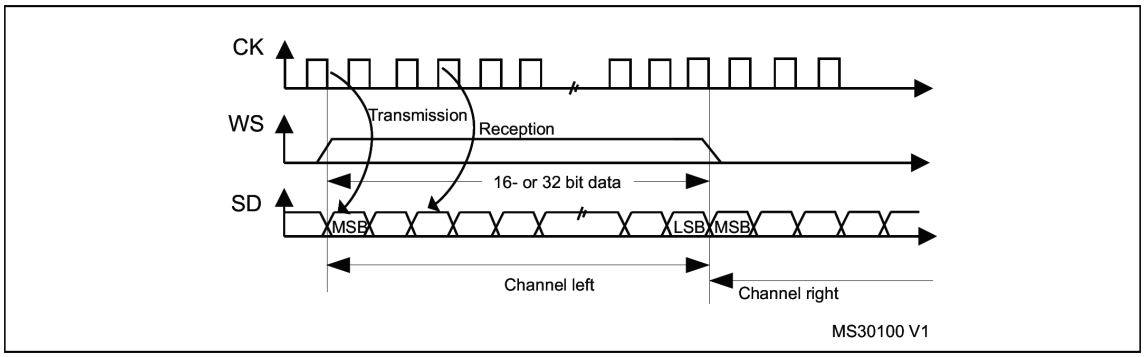
接收时，每次收到高 16 位半字 (MSB) 后，标志位 RXNE 置’1’，如果允许了相应的中断，则可以产生中断。

这样，在 2 次读和写之间有更多的时间，可以防止下溢或者上溢的情况发生。

MSB 对齐标准

在此标准下，WS 信号和第一个数据位，即最高位 (MSB) 同时产生。

图 269. MSB 对齐 16 位或 32 位全精度，CPOL = 0



发送方在时钟信号的下降沿改变数据；接收方是在上升沿读取数据。

图 270. MSB 对齐 24 位数据，CPOL = 0

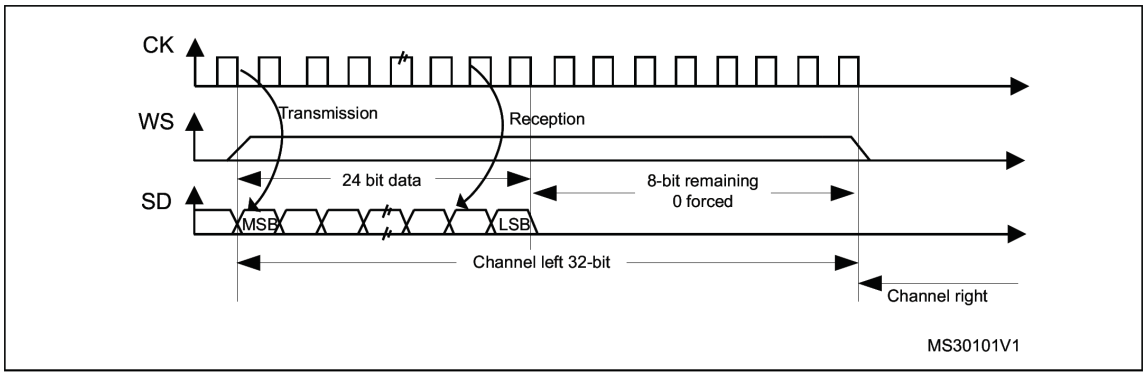
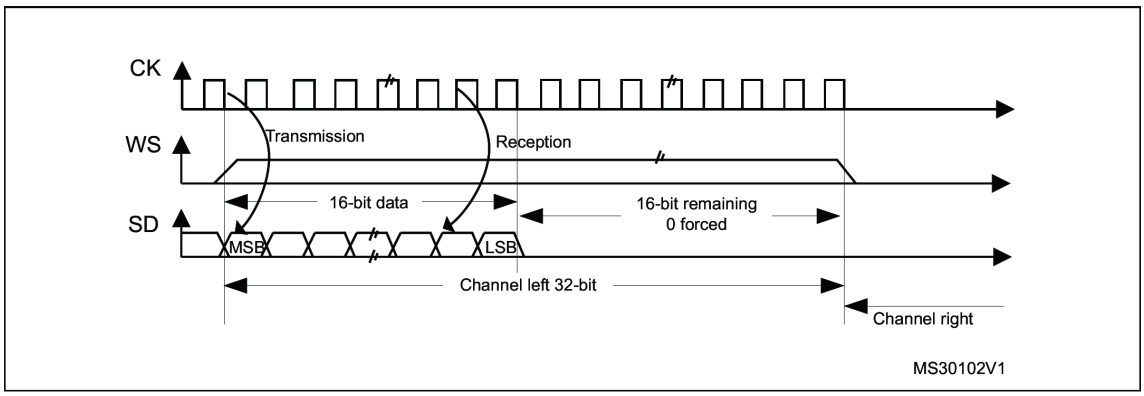


图 271. MSB 对齐 16 位数据扩展到 32 位包帧，CPOL = 0



LSB 对齐标准

此标准与 MSB 对齐标准类似 (在 16 位或 32 位全精度帧格式下无区别)。

图 272. LSB 对齐 16 位或 32 位全精度，CPOL = 0

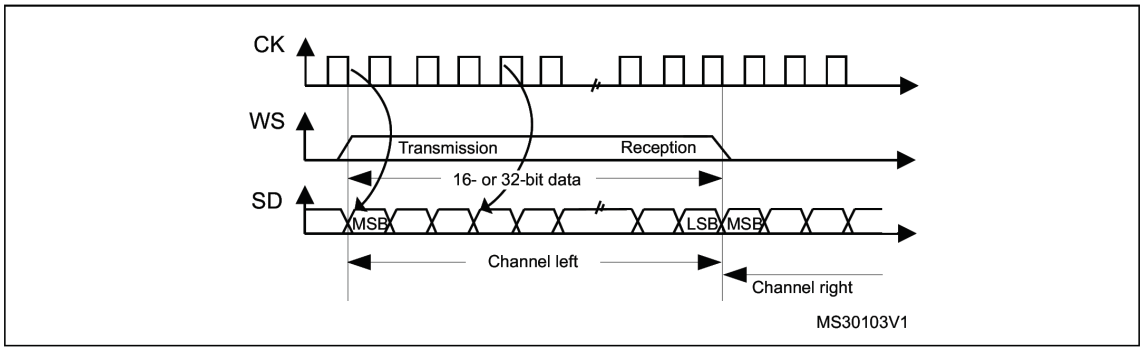
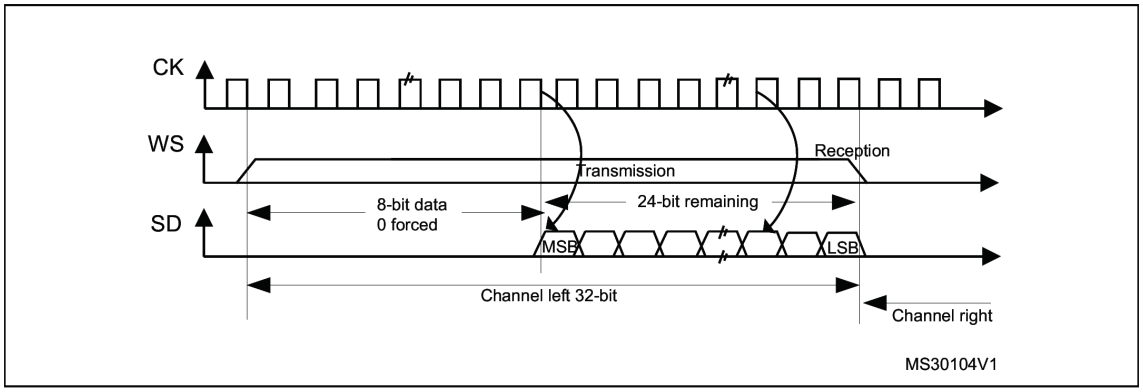
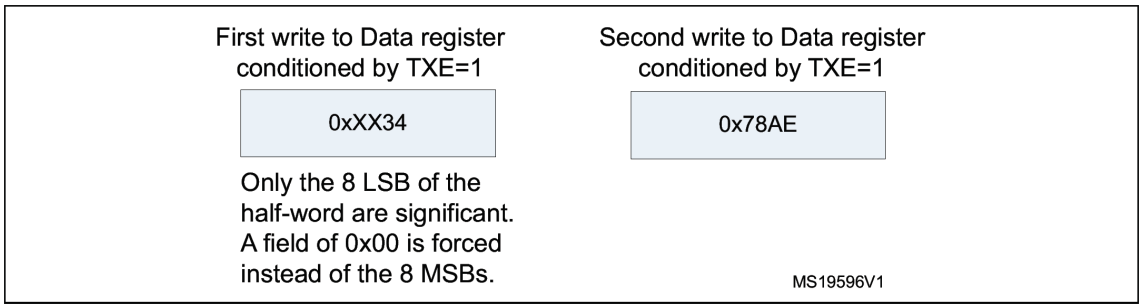


图 273. LSB 对齐 24 位数据，CPOL = 0



- 在发送模式下：
  如果要发送数据 0x3478AE，需要通过软件或者 DMA 对寄存器 SPI_DR 进行 2 次写操作。操作流程如下图所示。

图 274. 发送 0x3478AE 要求的操作



- 在接收模式下：
  如果要接收数据 0x3478AE，需要在 2 个连续的 RXNE 事件发生时，分别对寄存器 SPI_DR 进行 2 次读操作。

图 275。 接收 0x3478AE 要求的操作

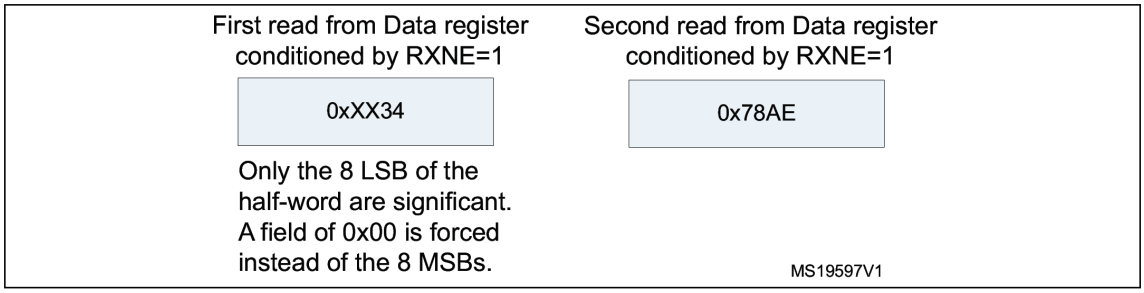
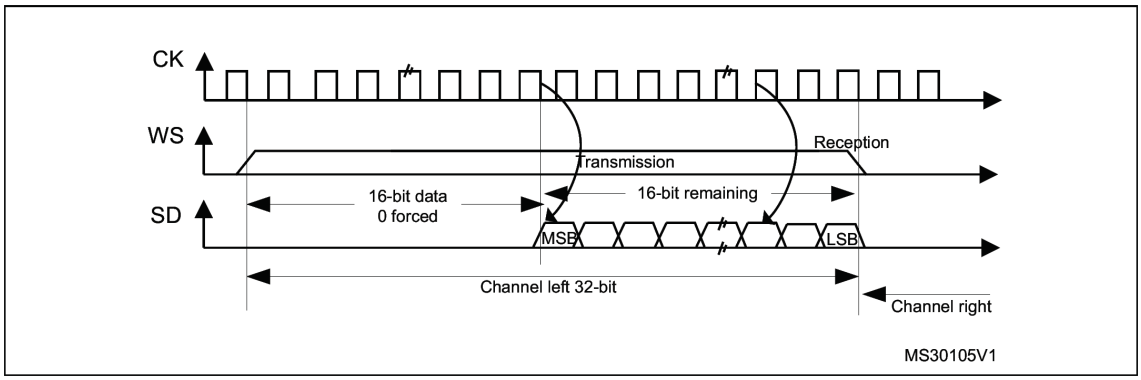


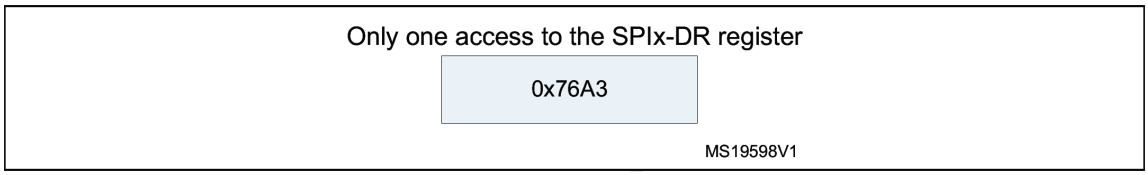
图 276. LSB 对齐 16 位数据扩展到 32 位包帧，CPOL = 0



在 I²S 配置阶段，如果选择将 16 位数据扩展到 32 位声道帧，只需要访问一次寄存器 SPI_DR。 用来扩展到 32 位的低 16 位被硬件置为 0x0000。 在这种情况下，它对应的半字的 MSB。

如果待传输或者接收的数据是 0x76A3(扩展到 32 位是 0x000076A3)，需要的操作如下图所示。

图 277. 16 位扩展至 32 位包帧的例子



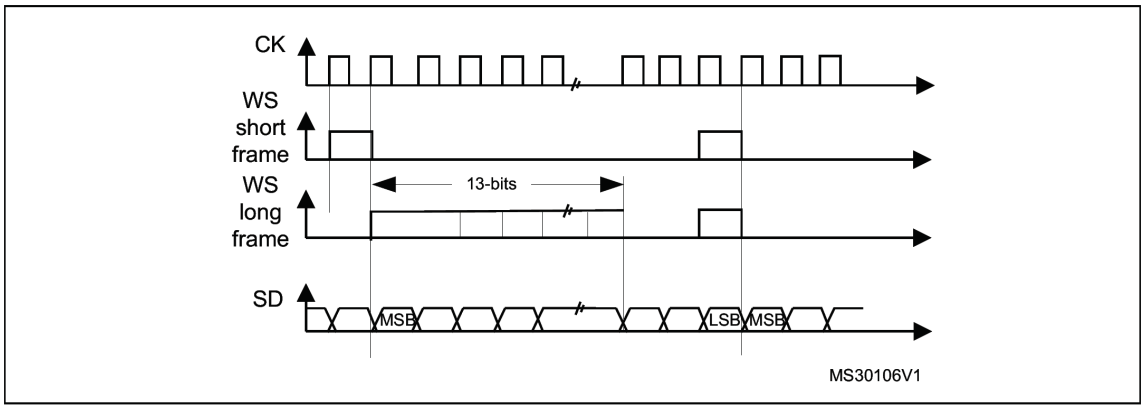
在发送时，如果 TXE 为’1’，用户需要写入待发送的数据 (即 0x76A3)。 用来扩展到 32 位的 0x0000 部分由硬件首先发送出去， 一旦有效数据开始从 SD 引脚送出，即发生下一次 TXE 事件。

在接收时，一旦接收到有效数据 (而不是 0x0000 部分)，即发生 RXNE 事件。
这样，在 2 次读和写之间有更多的时间，可以防止下溢或者上溢的情况发生。

PCM 标准

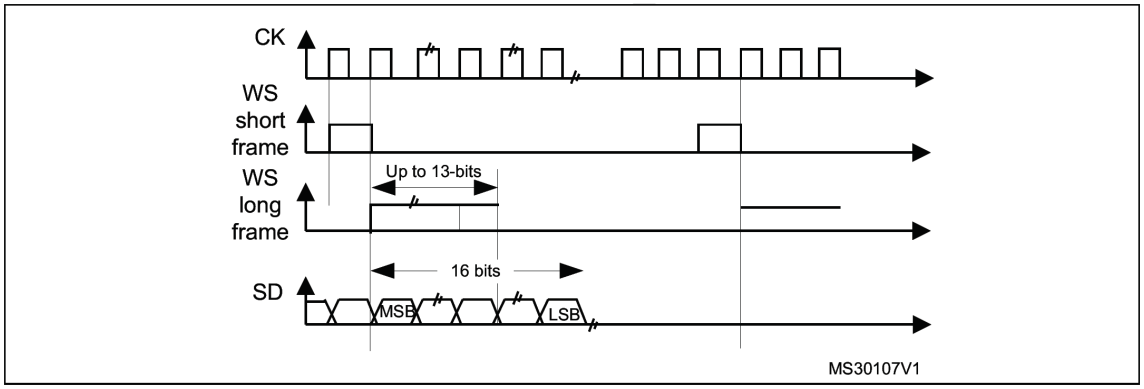
在 PCM 标准下，不存在声道选择的信息。 PCM 标准有 2 种可用的帧结构，短帧或者长帧，可以通过设置寄存器 SPI_I2SCFGR 的 PCMSYNC 位来选择。

图 278. PCM 标准波形 (16 位)



对于长帧，主模式下，用来同步的 WS 信号有效的时间固定为 13 位。
对于短帧，用来同步的 WS 信号长度只有 1 位。

图 279. PCM 标准波形 (16 位扩展到 32 位包帧)



注： 无论哪种模式 (主或从)、哪种同步方式 (短帧或长帧)，连续的 2 帧数据之间和 2 个同步信号之间的时间差，(即使是从模式) 需要通过设置 SPI_I2SCFGR 寄存器的 DATLEN 位和 CHLEN 位来确定。

25.6.3 时钟发生器

I^2S 的比特率即确定了在 I^2S 数据线上的数据流和 I^2S 的时钟信号频率。

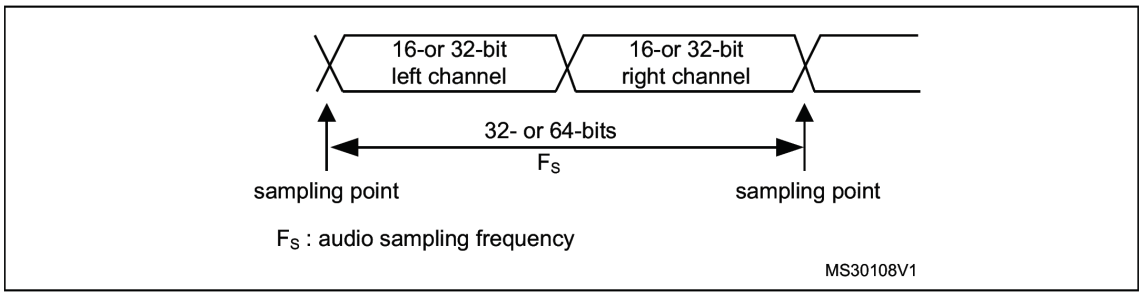
I^2S 比特率 = 每个声道的比特数 × 声道数目 × 音频采样频率

对于一个具有左右声道和 16 位音频信号，I^2S 比特率计算如下：

I^2S 的比特率 = $16 \times 2 \times f_S$

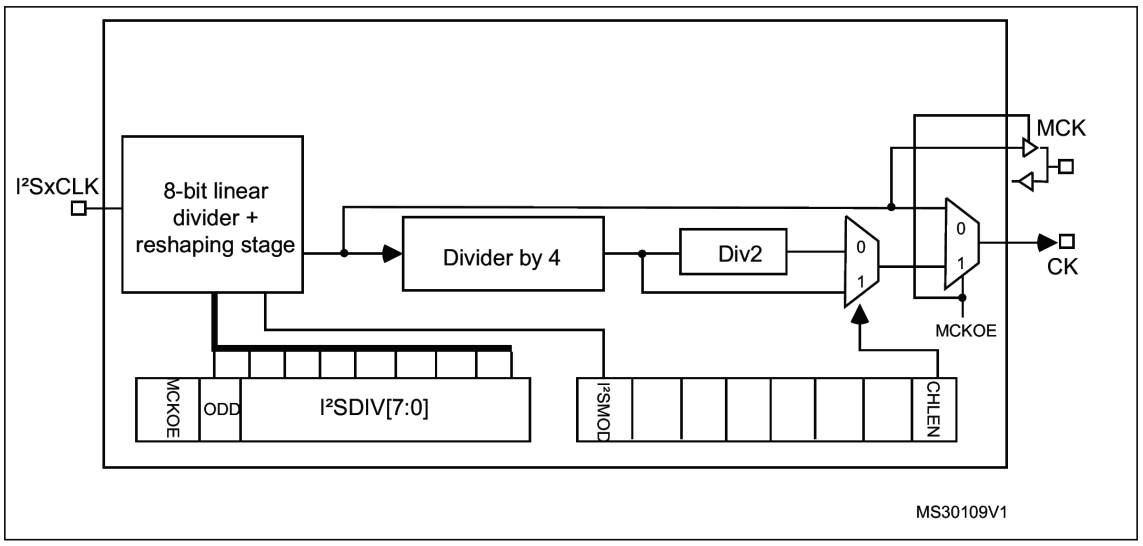
如果包长为 32 位，则有：I^2S 比特率 = $32 \times 2 \times f_S$

图 280. 音频采样频率定义



在主模式下，为了获得需要的音频频率，需要正确地对线性分频器进行设置。

图 281. I²S 时钟发生器结构



1. 其中 X 可以是 2 或 3。

图 280 给出了通信时钟结构。 为了实现高品质的音频性能，的 I2SxCLK 时钟源可以是 PLLI2S 的输出（通过 R 分频因子）或外部时钟（映射到 I2S_CKIN 脚）

音频的采样频率可以是 6kHz、48kHz、44.1kHz、32kHz、22.05kHz、16kHz、11.025kHz 或者 8kHz(或任何此范围内的数值)。 为了获得需要的频率，需按照以下公式设置线性分频器：

当需要生成主时钟时 (寄存器 SPI_I2SPR 的 MCKOE 位为’1’)：
声道的帧长为 16 位时，Fs = I2SxCLK / [(16*2) * ((2*I2SDIV) + ODD)*8
声道的帧长为 32 位时，Fs = I2SxCLK / [(32*2) * ((2*I2SDIV) + ODD)*4
当关闭主时钟时 (MCKOE 位为’0’)：
声道的帧长为 16 位时，Fs = I2SxCLK / [(16*2) * ((2*I2SDIV) + ODD)]
声道的帧长为 32 位时，Fs = I2SxCLK / [(32*2) * ((2*I2SDIV) + ODD)] fS =
下面 2 张表给出了不同时钟配置时，精确参数的例子。

注： 可以使用其它配置以达到优化时钟精确度的目的。

表 97. 音频频率精度（PLLM 的 VCO = 1MHz 时）(1)

Master clock	Target f_S (Hz)	Data format	PLLI2SN	PLLI2SR	I2SDIV	I2SODD	Real f_S (Hz)	Error
Disabled	8000	16-bit	192	2	187	1	8000	0.0000%
		32-bit	192	3	62	1	8000	0.0000%
	16000	16-bit	192	3	62	1	16000	0.0000%
		32-bit	256	2	62	1	16000	0.0000%
	32000	16-bit	256	2	62	1	32000	0.0000%
		32-bit	256	5	12	1	32000	0.0000%
	48000	16-bit	192	5	12	1	48000	0.0000%
		32-bit	384	5	12	1	48000	0.0000%
	96000	16-bit	384	5	12	1	96000	0.0000%
		32-bit	424	3	11	1	96014.49219	0.0151%
	22050	16-bit	290	3	68	1	22049.87695	0.0006%
		32-bit	302	2	53	1	22050.23438	0.0011%
	44100	16-bit	302	2	53	1	44100.46875	0.0011%
		32-bit	429	4	19	0	44099.50781	0.0011%
	192000	16-bit	424	3	11	1	192028.9844	0.0151%
		32-bit	258	3	3	1	191964.2813	0.0186%
Enabled	8000	don't care	256	5	12	1	8000	0.0000%
	16000	don't care	213	2	13	0	16000.60059	0.0038%
	32000	don't care	213	2	6	1	32001.20117	0.0038%
	48000	don't care	258	3	3	1	47991.07031	0.0186%
	96000	don't care	344	2	3	1	95982.14063	0.0186%
	22050	don't care	429	4	9	1	22049.75391	0.0011%
	44100	don't care	271	2	6	0	44108.07422	0.0183%

1. 此表仅提供不同的时钟配置示例值。 允许最佳的时钟精度的其他配置是可能的。

表 98. 音频频率精度（PLLM 的 VCO = 2MHz 时）(1)

Master clock	Target f_S (Hz)	Data format	PLLI2SN	PLLI2SR	I2SDIV	I2SODD	Real f_S (Hz)	Error
Disabled	8000	16-bit	192	3	250	0	8000	0.0000%
		32-bit	192	2	187	1	8000	0.0000%
	16000	16-bit	192	2	187	1	16000	0.0000%
		32-bit	192	3	62	1	16000	0.0000%
	32000	16-bit	192	3	62	1	32000	0.0000%
		32-bit	256	5	25	0	32000	0.0000%
	48000	16-bit	192	2	62	1	48000	0.0000%
		32-bit	192	5	12	1	48000	0.0000%
	96000	16-bit	192	5	12	1	96000	0.0000%
		32-bit	384	5	12	1	96000	0.0000%
	22050	16-bit	290	6	68	1	22049.87695	0.0006%
		32-bit	302	4	53	1	22050.23438	0.0011%
	44100	16-bit	302	4	53	1	44100.46875	0.0011%
		32-bit	405	7	20	1	44098.43359	0.0036%
	192000	16-bit	384	5	12	1	192000	0.0000%
		32-bit	258	7	3	0	191964.2969	0.0186%
Enabled	8000	don't care	256	5	25	0	8000	0.0000%
	16000	don't care	256	5	12	1	16000	0.0000%
	32000	don't care	213	4	6	1	32001.20117	0.0038%
	48000	don't care	258	7	3	0	47991.07422	0.0186%
	96000	don't care	258	3	3	1	95982.14063	0.0186%
	22050	don't care	254	3	15	0	22048.61133	0.0063%
	44100	don't care	254	3	7	1	44097.22266	0.0063%

1. 此表仅提供不同的时钟配置示例值。 允许最佳的时钟精度的其他配置是可能的。

表 99. 音频频率精度，使用标准的 8 兆赫的 HSE（只针对大容量和超大容量的产品）

SYSCLK (MHz)	I2S_DIV		I2S_ODD		MCLK	Target f_S (Hz)	Real f_S (KHz)		Error	
	16-bit	32-bit	16-bit	32-bit			16-bit	32-bit	16-bit	32-bit
72	11	6	1	0	No	96000	97826.09	93750	1.90%	2.34%
72	23	11	1	1	No	48000	47872.34	48913.04	0.27%	1.90%
72	25	13	1	0	No	44100	44117.65	43269.23	0.04%	1.88%
72	35	17	0	1	No	32000	32142.86	32142.86	0.44%	0.44%
72	51	25	0	1	No	22050	22058.82	22058.82	0.04%	0.04%

表 99. 音频频率精度，使用标准的 8 兆赫的 HSE（只针对大容量和超大容量的产品）（续）

SYSCLK (MHz)	I2S_DIV		I2S_ODD		MCLK	Target f_S (Hz)	Real f_S (KHz)		Error	
	16-bit	32-bit	16-bit	32-bit			16-bit	32-bit	16-bit	32-bit
72	70	35	1	0	No	16000	15675.75	16071.43	0.27%	0.45%
72	102	51	0	0	No	11025	11029.41	11029.41	0.04%	0.04%
72	140	70	1	1	No	8000	8007.11	7978.72	0.09%	0.27%
72	2	2	0	0	Yes	96000	70312.15	70312.15	26.76%	26.76%
72	3	3	0	0	Yes	48000	46875	46875	2.34%	2.34%
72	3	3	0	0	Yes	44100	46875	46875	6.29%	6.29%
72	9	9	0	0	Yes	32000	31250	31250	2.34%	2.34%
72	6	6	1	1	Yes	22050	21634.61	21634.61	1.88%	1.88%
72	9	9	0	0	Yes	16000	15625	15625	2.34%	2.34%
72	13	13	0	0	Yes	11025	10817.30	10817.30	1.88%	1.88%
72	17	17	1	1	Yes	8000	8035.71	8035.71	0.45%	0.45%

表 100. 音频频率精度，使用标准的 25MHz 和 PLL3（只针对互联型产品）

PREDIV2		PLL3MUL		I2SDIV		I2SODD		MCLK	Target f_S (Hz)	Real f_S (kHz)		Error	
16-bit	32-bit	16-bit	32-bit	16-bit	32-bit	16-bit	32-bit			16-bit	32-bit	16-bit	32-bit
6	6	14	14	19	9	0	1	No	96000	95942.98	95942.98	**0.0594%**	**0.0594%**
7	12	20	14	46	9	1	1	No	48000	48003.07	47971.49	**0.0064%**	**0.0594%**
8	8	14	14	31	15	0	1	No	44100	44102.82	44102.82	**0.0064%**	**0.0064%**
11	4	16	10	35	30	1	1	No	32000	32010.24	32018.44	**0.0320%**	**0.0576%**
8	8	14	14	62	31	0	0	No	22050	22051.41	22051.41	**0.0064%**	**0.0064%**
7	11	20	16	139	35	1	1	No	16000	16001.02	16005.12	**0.0064%**	**0.0320%**
8	8	14	14	124	62	0	0	No	11025	11025.71	11025.71	**0.0064%**	**0.0064%**
9	9	20	20	217	108	0	1	No	8000	8000.512	8000.512	**0.0064%**	**0.0064%**
4	4	8	8	2	2	0	0	Yes	96000	97656.25	97656.25	**1.7253%**	**1.7253%**
13	13	16	16	2	2	1	1	Yes	48000	48076.92	48076.92	**0.1603%**	**0.1603%**
5	5	9	9	4	4	0	0	Yes	44100	43945.31	43945.31	**0.3508%**	**0.3508%**
5	5	9	9	5	5	1	1	Yes	32000	31960.22	31960.22	**0.1243%**	**0.1243%**
5	5	13	13	11	11	1	1	Yes	22050	22078.80	22078.80	**0.1306%**	**0.1306%**
9	9	14	14	9	9	1	1	Yes	16000	15990.49	15990.49	**0.0594%**	**0.0594%**
8	8	14	14	15	15	1	1	Yes	11025	11025.70	11025.70	**0.0064%**	**0.0064%**
4	4	10	10	30	30	1	1	Yes	8000	8004.611	8004.611	**0.0576%**	**0.0576%**

表 101. 音频频率精度，使用标准的 14.7456MHz 和 PLL3（只针对互联型产品）

PREDIV2		PLL3MUL		I2SDIV		I2SODD		MCLK	Target f_S (Hz)	Real f_S (kHz)		Error	
16-bit	32-bit	16-bit	32-bit	16-bit	32-bit	16-bit	32-bit			16-bit	32-bit	16-bit	32-bit
3	3	10	10	16	8	0	0	No	96000	96000	96000	**0%**	**0%**
6	6	20	20	32	16	0	0	No	48000	48000	48000	**0%**	**0%**
11	11	20	20	19	9	0	1	No	44100	44095.69	44095.69	**0.0098%**	**0.0098%**
2	2	10	10	72	36	0	0	No	32000	32000	32000	**0%**	**0%**
11	11	10	10	19	9	0	1	No	22050	22047.84	22047.84	**0.0098%**	**0.0098%**
4	4	20	20	144	72	0	0	No	16000	16000	16000	**0%**	**0%**
2	2	10	10	209	104	0	1	No	11025	11023.92	11023.92	**0.0098%**	**0.0098%**
12	12	20	20	96	48	0	0	No	8000	8000	8000	**0%**	**0%**
2	2	10	10	3	3	0	0	Yes	96000	96000	96000	**0%**	**0%**
6	6	20	20	4	4	0	0	Yes	48000	48000	48000	**0%**	**0%**
2	2	10	10	6	6	1	1	Yes	44100	44307.69	44307.69	**0.4710**	**0.4710%**
2	2	10	10	9	9	0	0	Yes	32000	32000	32000	**0%**	**0%**

表 101. 音频频率精度，使用标准的 14.7456MHz 和 PLL3（只针对互联型产品）（续）

PREDIV2		PLL3MUL		I2SDIV		I2SODD		MCLK	Target f_S (Hz)	Real f_S (kHz)		Error	
16-bit	32-bit	16-bit	32-bit	16-bit	32-bit	16-bit	32-bit			16-bit	32-bit	16-bit	32-bit
4	4	13	13	8	8	1	1	Yes	22050	22023.52	22023.52	**0.1200%**	**0.1200%**
4	4	20	20	18	18	0	0	Yes	16000	16000	16000	**0%**	**0%**
11	11	20	20	9	9	1	1	Yes	11025	11023.92	11023.92	**0.0098%**	**0.0098%**
6	6	20	20	24	24	0	0	Yes	8000	8000	8000	**0%**	**0%**

25.6.4 I^2S 主模式

设置 I2S 工作在主模式，串行时钟由引脚 CK 输出，字选信号由引脚 WS 产生。 可以通过设置寄存器 SPI_I2SPR 的 MCKOE 位来选择输出或者不输出主时钟 (MCK)。

流程

1. 设置寄存器 SPI_I2SPR 的 I2SDIV[7:0] 定义与音频采样频率相符的串行时钟波特率。 同时也要定义寄存器 SPI_I2SPR 的 ODD 位。
2. 设置 CKPOL 位定义通信用时钟在空闲时的电平状态。 如果需要向外部的 DAC/ADC 音频器件提供主时钟 MCK，将寄存器 SPI_I2SPR 的 MCKOE 位置为’1’。(按照不同的 MCK 输出状态，计算 I2SDIV 和 ODD 的值，详见 23.4.3 节)。
3. 设置寄存器 SPI_I2SCFGR 的 I2SMOD 位为’1’激活 I2S 功能，设置 I2SSTD[1:0] 和 PCMSYNC 位选择所用的 I2S 标准，设置 CHLEN 选择每个声道的数据位数。 还要设置寄存器 SPI_I2SCFGR 的 I2SCFG[1:0] 选择 I2S 主模式和方向 (发送端还是接收端)。
4. 如果需要，可以通过设置寄存器 SPI_CR2 来打开所需的中断功能和 DMA 功能。
5. 必须将寄存器 SPI_I2SCFGR 的 I2SE 位置为’1’。

引脚 WS 和 CK 需要配置为输出模式。 如果寄存器 SPI_I2SPR 的 MCKOE 位为’1’，引脚 MCK 也要配置成输出模式。

发送流程

当写入 1 个半字 (16 位) 的数据至发送缓存，发送流程开始。

假设第一个写入发送缓存的数据对应的是左声道数据。当数据从发送缓存移到移位寄存器时，标志位 TXE 置’1’，这时，要把对应右声道的数据写入发送缓存。 标志位 CHSIDE 提示了目前待传输的数据对应哪个声道。 标志位 CHSIDE 的值在 TXE 为’1’时更新，因此它在 TXE 为’1’时有意义。

在先左声道后右声道的数据都传输完成后，才能被认为是一个完整的数据帧。 不可以只传输部分数据帧，如仅有左声道的数据。

当发出第一位数据的同时，半字数据被并行地传送至 16 位移位寄存器，然后后面的位依次按高位在先的顺序从引脚 MOSI/SD 发出。 每次数据从发送缓存移至移位寄存器时，标志位 TXE 置为’1’，如果寄存器 SPI_CR2 的 TXEIE 位为’1’，则产生中断。
写入数据的操作取决于所选择的 I^2S 标准，详见 25.6.2 节。

为了保证连续的音频数据传输，建议在当前传输完成之前，对寄存器 SPI_DR 写入下一个要传输的数据。

建议在要关闭 I^2S 功能时，等待标志位 TXE=1 及 BSY=0，再将 I2SE 位清’0’。

接收流程

接收流程的配置步骤除了第 3 点外，与发送流程的一致 (参见前述的”发送流程”)，需要通过配置 I2SCFG[1:0] 来选择主接收模式。

无论何种数据和声道长度，音频数据总是以 16 位包的形式接收。 即每次填满接收缓存后，标志位 RXNE 置’1’，如果寄存器 SPI_CR2 的 RXNEIE 位为’1’，则产生中断。 根据配置的数据和声道长度，收到左声道或右声道的数据会需要 1 次或者 2 次把数据传送到接收缓存的过程。

对寄存器 SPI_DR 进行读操作即可清除 RXNE 标志位。

每次接收以后即更新 CHSIDE。 它的值取决于 I^2S 单元产生的 WS 信号。
读取数据的操作取决于所选择的 I^2S 标准，详见 26.6.2 节。

如果前一个接收到的数据还没有被读取，又接收到新数据，即发生上溢，标志位 OVR 被置为’1’， 如果寄存器 SPI_CR2 的 ERRIE 位为’1’，则产生中断，表示发生了错误。

若要关闭 I²S 功能，需要执行特别的操作，以保证 I²S 模块可以正常地完成传输周期而不会开始新的数据传输。 操作过程与数据配置和通道长度、以及音频协议的模式相关：

- 16 位数据扩展到 32 位通道长度 (DATLEN=00 并且 CHLEN=1)，使用 LSB(低位) 对齐模式 (I2SSTD=10)
  a) 等待倒数第二个 (n-1)RXNE=1
  b) 等待 17 个 I²S 时钟周期 (使用软件延迟)
  c) 关闭 I²S(I2SE=0)。
- 16 位数据扩展到 32 位通道长度 (DATLEN=00 并且 CHLEN=1)，使用 MSB(高位) 对齐、I²S 或 PCM 模式 (分别为 I2SSTD=00，I2SSTD=01 或 I2SSTD=11)
  a) 等待最后一个 RXNE=1；
  b) 等待 1 个 I²S 时钟周期 (使用软件延迟)；
  c) 关闭 I²S(I2SE=0)。
- 所有其它 DATLEN 和 CHLEN 的组合，I2SSTD 选择的任意音频模式，使用下述方式关闭 I²S：
  a) 等待倒数第二个 (n-1)RXNE=1；
  b) 等待一个 I²S 时钟周期 (使用软件延迟)；
  c) 关闭 I²S(I2SE=0)。

注： 在传输期间 BSY 标志始终为低。

25.6.5 I²S 从模式

在从模式下，I²S 可以设置成发送和接收模式。 从模式的配置方式基本遵循和配置主模式一样的流程。 在从模式下，不需要 I²S 接口提供时钟。 时钟信号和 WS 信号都由外部主 I²S 设备提供，连接到相应的引脚上。 因此用户无需配置时钟。

配置步骤列举如下：

1. 设置寄存器 SPI_I2SCFGR 的 I2SMOD 位激活 I2S 功能；设置 I2SSTD[1:0] 来选择所用的 I²S 标准；设置 DATLEN[1:0] 选择数据的比特数；设置 CHLEN 选择每个声道的数据位数。设置寄存器 SPI_I2SCFGR 的 I2SCFG[1:0] 选择 I²S 从模式的数据方向 (发送端还是接收端)。
2. 根据需要，设置寄存器 SPI_CR2 打开所需的中断功能和 DMA 功能。
3. 必须设置寄存器 SPI_I2SCFGR 的 I2SE 位为’1’。

发送流程

当外部主设备发送时钟信号，并且当 NSS_WS 信号请求传输数据时，发送流程开始。 必须先使能从设备，并且写入 I²S 数据寄存器之后，外部主设备才能开始通信。

对于 I²S 的 MSB 对齐和 LSB 对齐模式，第一个写入数据寄存器的数据项对应左声道的数据。当开始通信时，数据从发送缓冲器传送到移位寄存器， 然后标志位 TXE 置为’1’；这时，要把对应右声道的数据项写入 I²S 数据寄存器。

标志位 CHSIDE 提示了目前待传输的数据对应哪个声道。 与主模式的发送流程相比，在从模式中，CHSIDE 取决于来自外部主 I^2S 的 WS 信号。 这意味着从 I2S 在接收到主端生成的时钟信号之前，就要准备好第一个要发送的数据。 WS 信号为’1’表示先发送左声道。

注： 设置 I2SE 位为’1’的时间，应当比 CK 引脚上的主 I^2S 时钟信号早至少 2 个 PCLK 时钟周期。

当发出第一位数据的时候，半字数据并行地通过 I^2S 内部总线传输至 16 位移位寄存器，然后其它位依次按高位在先的顺序从引脚 MOSI/SD 发出。 每次数据从发送缓冲器传送至移位寄存器时，标志位 TXE 置’1’，如果寄存器 SPI_CR2 的 TXEIE 位为’1’，则产生中断。
注意，在对发送缓冲器写入数据前，要确认标志位 TXE 为’1’。 写入数据的操作取决于所选中的 I^2S 标准，详见 25.6.2 节。

为了保证连续的音频数据传输，建议在当前传输完成之前，对寄存器 SPI_DR 写入下一个要传输的数据。 如果在代表下一个数据传输的第一个时钟边沿到达之前，新的数据仍然没有写入寄存器 SPI_DR，下溢标志位会置’1’，并可能产生中断； 它指示软件发送数据错误。
如果寄存器 SPI_CR2 的 ERRIE 位为’1’，在寄存器 SPI_SR 的标志位 UDR 为高时，就会产生中断。 建议在这时关闭 I2S，然后重新从左声道开始发送数据。
建议在清除 I2SE 位关闭 I^2S 之前，先等待 TXE=1 并且 BSY=0。

接收流程

配置步骤除了第 1 点（参见 25.6.5 小结）外，与发送流程一致。需要通过配置 SPIx_I2SCFGR 寄存器中的 I2SCFG[1:0] 来选择主接收模式。

无论何种数据和声道长度，音频数据总是以 16 位包的形式接收， 即每次填满接收缓存，标志位 RXNE 置’1’，如果寄存器 SPI_CR2 的 RXNEIE 位为’1’，则产生中断。 按照不同的数据和声道长度设置，收到左声道或者右声道数据会需要 1 次或者 2 次传输数据至接收缓冲器的过程。

每次接收到数据 (将要从 SPI_DR 读出) 以后即更新 CHSIDE， 它对应 I2S 单元产生的 WS 信号。 读取 SPI_DR 寄存器，将清除 RXNE 位。
读取数据的操作取决于所选中的 I2S 标准，详见 25.6.2 节。

在还没有读出前一个接收到的数据，又接收到新数据时，即产生上溢，并设置标志位 OVR 为’1’； 如果寄存器 SPI_CR2 的 ERRIE 位为’1’，则产生中断，指示发生了错误。
要关闭 I2S 功能时，需要在接收到最后一次 RXNE=1 时将 I2SE 位清’0’。

注：　外部主 I2S 器件需要有通过音频声道发送 / 接收 16 位或 32 位数据包的功能。

25.6.6 状态标志位

应用程序可通过 3 个状态标志来了解 I2S 总线的全部状态。

忙标志（BSY）

BSY 标志由硬件设置和清零（软件改写这个标志是没有用的）。 它指示 I^2S 通信层的状态。 该位为’1’时表明 I^2S 通讯正在进行中， 但有一个例外：主接收模式 (I2SCFG=11) 下，在接收期间 BSY 标志始终为低。

在软件要关闭 I^2S 模块之前，可以使用 BSY 标志检测传输是否结束，这样可以避免破坏最后一次传输， 因此需要严格按照下述过程执行。

当传输开始时，BSY 标志被置为’1’，除非 I^2S 模块处于主接收模式。 下述情况时，该标志位被清除：

- 当传输结束时 (除了主发送模式，这种模式下通信是连续的)；
- 当关闭 I^2S 模块时。

当通信是连续的时候：

- 在主发送模式时，整个传输期间，BSY 标志始终为高；
- 在从模式时，每个数据项传输之间，BSY 标志在 1 个 I^2S 时钟周期内变低。

注：　不要使用 BSY 标志处理每一个数据项的发送和接收， 最好使用 TXE 和 RXNE 标志。

TX 缓冲器空标志（TXE）

该标志位为’1’表示发送缓冲器为空，可以对发送缓冲器写入新的待发送数据。 在发送缓冲器中已有数据时，标志位清’0’。 在 I2S 被关闭时 (I2SE 位为’0’)，该标志位也为’0’。

Rx 缓冲非空（RXNE）

该标志位置’1’表示在接收缓存里有接收到的有效数据。在读取寄存器 SPI_DR 时，该位清’0’。

声道标志位 (CHSIDE)

在发送模式下，该标志位在 TXE 为高时刷新， 指示从 SD 引脚上发送的数据所在的声道。 如果在从发送模式下发生了下溢错误，该标志位的值无效，在重新开始通讯前需要把 I2S 关闭再打开。

在接收模式下，该标志位在寄存器 SPI_DR 接收到数据时刷新， 这指示接收到的数据所在的声道。 注意，如果发生错误 (如上溢 OVR)，该标志位无意义，需要将 I2S 关闭再打开 (同时，如果必要修改 I2S 的配置)。

在 PCM 标准下，无论短帧格式还是长帧格式，这个标志位都没有意义。

如果寄存器 SPI_SR 的标志位 OVR 或 UDR 为’1’，且寄存器 SPI_CR2 的 ERRIE 位为’1’，则会产生中断。 (中断源已经被清除后) 可以通过读寄存器 SPI_SR 来清除中断标志。

错误标志

I^2S 单元有 2 个错误标志位。

下溢标志位 (UDR)

在从发送模式下，如果数据传输的第一个时钟边沿到达时，新的数据仍然没有写入 SPI_DR 寄存器，该标志位会被置’1’。 在寄存器 SPI_I2SCFGR 的 I2SMOD 位置’1’后，该标志位才有效。 如果寄存器 SPI_CR2 的 ERRIE 位为’1’，就会产生中断。
通过对寄存器 SPI_SR 进行读操作来清除该标志位。

上溢标志位 (OVR)

如果还没有读出前一个接收到的数据时，又接收到新的数据，即产生上溢，该标志位置’1’，因此，输入的数据都将丢失。 如果寄存器 SPI_CR2 的 ERRIE 位为’1’，则产生中断指示发生了错误。

这时，接收缓存的内容，不会刷新为从发送设备送来的新数据。 对寄存器 SPI_DR 的读操作返回最后一个正确接收到的数据。 其他所有在上溢发生后由发送设备发出的 16 位数据都会丢失。

通过先读寄存器 SPI_SR 再读寄存器 SPI_DR，来清除该标志位。

25.6.7 I^2S 中断

表 102 提供 I^2S 中断的列表。

表 102. I^2S 的中断请求

中断事件	事件标志	使能控制位
发送缓冲区空	TXE	TXEIE
接收缓冲区不空标志	RXNE	RXNEIE
溢出错误	OVR	ERRIE
欠载错误	UDR	

25.6.8 DMA 功能

DMA 的工作方式在 I^2S 模式除了 CRC 功能不可用以外，与在 SPI 模式完全相同。 因为在 I^2S 模式下没有数据传输保护系统。

25.7 SPI 和 I2S 寄存器描述

可以用半字（16 位）或字（32 位）的方式操作这些外设寄存器。此外 SPI_DR 可以用 8 位的方式访问。

25.7.1 SPI 控制寄存器 1(SPIx_CR1)

地址偏移 : 0x00

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIDI MODE	BIDI OE	CRC EN	CRC NEXT	CRCL	RX ONLY	SSM	SSI	LSB FIRST	SPE	BR [2:0]			MSTR	CPOL	CPHA
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15 BIDIMODE： 双向数据模式使能

0: 选择 2 线的单向数据模式

1: 选择单线双向数据模式

注： 该位在 I2S 模式下没有用

位 14 BIDIOE： 双向模式输出使能

和 BIDIMODE 位一起决定在“单线双向”模式下数据的输出方向

0: 输出禁止 (只收模式)；

1: 输出使能 (只发模式)。

注： 1. 在主模式下，用 MOSI 引脚。在从模式下，用 MISO 引脚。

2. 该位在 I2S 模式下没有用

位 13 CRCEN： 硬件 CRC 计算使能

0: CRC 计算禁用

1: CRC 计算使能

注： 1. 只有当 SPI 被禁止，该位才可以被写入。

2. 该位在 I2S 模式下没有用

位 12 CRCNEXT： 下一个发送 CRC

0: 下一个发送的值来自发送缓冲区。

1: 下一个发送的值来自发送 CRC 寄存器。

注： 1. 在 SPI_DR 寄存器写入最后一个数据后应马上设置该位。

2. 该位在 I2S 模式下没有用

位 11 CRCL： CRC 长度

由软件设置和清零，用于选择 CRC 长度。

0: 8 位 CRC 长度

1: 16 位 CRC 长度

注： 1. 只有当 SPI 被禁止，该位才可以被写入。

2. 该位在 I2S 模式下没有用

位 10 RXONLY： 只接收

和 BIDIMODE 位一起决定在“2 线单向”模式下数据的输出方向。 在多个从设备的配置中，在未被访问的从设备上该位被置 1，使得只有被访问的从设备有输出，从而不会造成数据线上数据冲突。

0: 全双工（发送和接收）

1: 输出禁止 (只收模式)；

注： 该位在 I2S 模式下没有用

位 9 SSM： 软件从机管理
当 SSM 被置位时，NSS 引脚上的电平由 SSI 位的值决定。
0: 禁止软件从设备管理；
1: 启用软件从设备管理
注： 在 I2S 模式和 SPI TI 模式下，这个位不可用。

位 8 SSI： 内部从设备选择
此位只有当 SSM 位为 1 的时候才有效果。它决定了 NSS 上的电平，在 NSS 引脚上的 I/O 操作无效。
注： 在 I2S 模式和 SPI TI 模式下，这个位不可用。

位 7 LSBFIRST： 帧格式
0: 先发送 MSB；
1: 先发送 LSB。
注： 1. 当通信在进行时不能改变该位的值。
2. 在 I2S 模式和 SPI TI 模式下，这个位不可用。

位 6 SPE： SPI 使能
0: 禁止 SPI 设备；
1: 开启 SPI 设备。
注： 1. 该位在 I2S 模式下没有用
2. 当关闭 SPI 设备时，请按照第 642 页的过程操作。

位 5:3 BR [2:0]： 波特率控制
000: fPCLK/2
001: fPCLK/4
010: fPCLK/8
011: fPCLK/16
100: fPCLK/32
101: fPCLK/64
110: fPCLK/128
111: fPCLK/256
注： 当通信在进行时不能改变该位的值。
该位在 I2S 模式下没有用

位 2 MSTR： 主设备选择
0: 配置为从设备
1: 配置为主设备
注： 1. 当通信在进行时不能改变该位的值。
2. 该位在 I2S 模式下没有用

位 1 CPOL： 时钟极性
0: 空闲状态时，SCK 保持低电平；
1: 空闲状态时，SCK 保持高电平。
注： 1. 当通信在进行时不能改变该位的值。
2. 在 I2S 模式和 SPI TI 模式下，这个位不可用。

位 0 CPHA： 时钟相位
0: 第一个时钟沿对准第一位数据
1: 第二和时钟沿对准第一位数据
注： 1. 当通信在进行时不能改变该位的值。
2. 在 I2S 模式和 SPI TI 模式下，这个位不可用。

25.7.2 SPI 控制寄存器 2（ SPIx_CR2）

地址偏移 : 0x04

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	LDMA_TX	LDMA_RX	FRXTH	DS [3:0]				TXEIE	RXNEIE	ERRIE	FRF	NSSP	SSOE	TXDMAEN	RXDMAEN
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15 保留，必须保持复位时的值。

位 14 LDMA_TX：最后一次 DMA 发送

此位是在数据打包模式中，用来定义 DMA 数据传输的总数是奇数还是偶数。 它只有在 SPIx_CR2 寄存器的 TXDMAEN 位被设置，并且打包模式已开启（数据长度小于等于 8 位和写入访问 SPIx_DR 是 16 位宽）时有意义。它必须在 SPI 被禁止（SPIx_CR1 寄存器的 SPE = 0）时写入。

0: 传输的数据数量为偶数

1: 传输的数据数量为奇数

注： 1. 请按照 642 页步骤，关闭 SPI。

2. 该位在 I2S 模式下没有用

位 13 LDMA_RX： 最后一次 DMA 接收

此位是在数据打包模式中，用来定义 DMA 数据接收的总数是奇数还是偶数。 它只有在 SPIx_CR2 寄存器的 RXDMAEN 位被设置，并且打包模式已开启（数据长度小于等于 8 位和写入访问 SPIx_DR 是 16 位宽）时有意义。它必须在 SPI 被禁止（SPIx_CR1 寄存器的 SPE = 0）时写入。

0: 传输的数据数量为偶数

1: 传输的数据数量为奇数

注： 1. 请按照 642 页步骤，关闭 SPI。

2. 该位在 I2S 模式下没有用

位 12 FRXTH： FIFO 接收门限

此位用来设置触发 RXNE 事件时的 RXFIFO 的阈值。

0: 如果 FIFO 的存储水平大于或等于 1/2（16 位），产生 RXNE 事件。

1: 如果 FIFO 的存储水平大于或等于 1/4（8 位），产生 RXNE 事件。

注： 该位在 I2S 模式下没有用

位 11:8 DS[3:0] 数据位宽

这些位配置 SPI 传输数据的位宽：

0000: 不使用

0001: 不使用

0010: 不使用

0011: 4 位

0100: 5 位

0101: 6 位

0110: 7 位

0111: 8 位

1000: 9 位

1001: 10 位

1010: 11 位

1011: 12 位

1100: 13 位

1101: 14 位

1110: 15 位

1111: 16 位

如果软件试图写一个“不使用”的值，他们会被迫赋值“0111”（8 位）。

注： 这些位在 I2S 模式下没有用

位 7 TXEIE： TX 缓冲器空中断使能

0: TXE 中断屏蔽

1: TXE 中断没有被屏蔽。 用于在 TXE 标志置 1 的时候产生一个中断请求。

位 6 RXNEIE： RX 缓冲区非空中断使能

0: RXNE 中断屏蔽

1: TXE 中断没有被屏蔽。 用于在 RXNE 标志置 1 的时候产生一个中断请求。

位 5 ERRIE： 错误中断使能

这个位控制在出现错误事件（CRCERR，OVR，SPI 模式中的 MODF，TI 模式中的 FRE 和 UDR，I2S 模式中的 OVR）时是否产生中断。

0: 错误中断屏蔽

1: 错误中断使能

位 4 FRF： 帧格式

0: SPI Motorola 模式

1：SPI TI 模式

注： 1. 该位必须在 SPI 被禁止（SPE = 0）的时候写入

2. 该位在 I2S 模式下没有用

位 3 NSSP： NSS 脉冲管理

该位仅在主模式下使用。 它允许 SPI 在连续传输时，两个数据传输之间产生一个 NSS 脉冲。在单个数据传输的情况下，它会在传输结束后将 NSS 脚强制为高电平。在 CPHA=1 或 FRF=1 的时候，这个位没有什么意义。

0: 没有 NSS 脉冲

1: 产生 NSS 脉冲

注： 1. 该位必须在 SPI 被禁止（SPE = 0）的时候写入

2. 在 I2S 模式和 SPI TI 模式下，这个位不可用。

位 2 SSOE： SS 输出使能
0: 在主模式下 SS 输出被禁用，SPI 接口可以工作在多主机的配置下。
1: SPI 接口启用的同时在主模式下启用 SS 输出。 SPI 接口不能在多主环境下工作。
注： 在 I2S 模式和 SPI TI 模式下，这个位不可用。

位 1 TXDMAEN： Tx 缓冲 DMA 使能
当此位被设置，TXE 标志被设置时，产生一个 DMA 请求。
0: Tx 缓冲 DMA 禁止
1: Tx 缓冲 DMA 使能

位 0 RXDMAEN： Rx 缓冲 DMA 使能
当此位被设置，RXNE 标志被设置时，产生一个 DMA 请求。
0: Rx 缓冲 DMA 禁止
1: Rx 缓冲 DMA 使能

25.7.3 SPI 状态寄存器(SPIx_SR)

地址偏移 : 0x08
复位值 : 0x0002

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			FTLVL[1:0]		FRLVL[2:0]		FRE	BSY	OVR	MODF	CRC ERR	UDR	CHSIDE	TXE	RXNE
			r	r	r	r	r	r	r	r	rc_w0	r	r	r	r

位 15:13 保留，必须保持复位时的值。

位 12:11 FTLVL [1:0]： FIFO 发送存储水平
由硬件设置或清零
00: FIFO 空
01: 1/4 FIFO
10: 1/2 FIFO
11: FIFO 满（当 FIFO 门限大于 1/2 时认为是满）
注： 这些位在 I2S 模式下没有用

位 10:9 FRLVL [1:0]： FIFO 接收存储水平
由硬件设置或清零
00: FIFO 空
01: 1/4 FIFO
10: 1/2 FIFO
11: FIFO 满
注：这些位在 I2S 模式和打开了 CRC 计算功能时的 SPI 单接受模式下，不使用。

位 8 FRE： TI 帧格式错误
0: 没发生帧格式错误
1: 发生了一个帧格式错误

位 7 BSY： 忙标志
0: SPI（或 I2S）不忙
1: （SPI 或 I2S）通信忙或发送缓冲区不为空
由硬件设置和清除。
注： 必须谨慎使用的 BSY 标志： 参见章节 25.3.8： 642 页 禁用 SPI 的步骤和状态标志。

位 6 OVR： 溢出标志
0: 没有发生溢出
1: 发生溢出
此标志由硬件置位，由软件序列复位。 软件序列，请参阅第 669 页 关于错误标志。

位 5 MODF： 模式故障
0: 无模式故障发生
1: 模式故障发生
此标志由硬件置位，由软件序列复位。 软件序列，请参阅第 647 页 25.4 章。
注： I2S 模式不使用

位 4 CRCERR： CRC 错误标志
0: 收到的 CRC 值和 SPIx_RXCRCR 的值是匹配的
1: 收到的 CRC 值和 SPIx_RXCRCR 值不匹配
此标志由硬件置位，由软件清零。
注： I2S 模式下不使用

位 3 UDR： 欠载标志
0: 没有欠载发生
1: 欠载发生
此标志由硬件置位，由软件序列复位。 软件序列，请参阅第 669 页 关于错误标志。
注： 该位在 SPI 模式下没有用

位 2 CHSIDE： 通道标志
0: 需要传输或者接收左声道；
1: 需要传输或者接收右声道；
注： 该位在 SPI 模式下没有用 在 PCM 模式下无意义

位 1 TXE： 发送缓冲区为空标志
0: Tx 缓冲区非空
1: TX 缓冲器空

位 0 RXNE： 接收缓冲区非标志
0: RX 缓冲区空
1: Rx 缓冲非空

25.7.4 SPI 数据寄存器(SPIx_DR)

地址偏移 : 0x0C
复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 DR[15:0]: 数据寄存器

待发送或者已经收到的数据

数据寄存器作为 Rx 和 Tx FIFOs 的接口。 当读取数据寄存器时，RXFIFO 会被访问，而写入数据寄存器则会访问 TXFIFO（见 25.3.7 小结： 数据发送和接收流程）。

注： 数据始终是右对齐。 当写寄存器的时候，没有用到的位会被忽略，读的时候，没有用到的位会是 0. 接收门限的设置必须始终符合目前使用的读访问方式。

25.7.5 SPI 的 CRC 多项式寄存器(SPIx_CRCPR)

地址偏移 : 0x10

复位值 : 0x0007

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCPOLY[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 CRCPOLY [15:0]： CRC 多项式寄存器

该寄存器包含 CRC 计算多项式。

该寄存器的复位值是（0007H）。 根据需要，可以配置成另一个多项式。

注： 多项式的值应该是奇数。 不支持偶数值。

25.7.6 SPI 接收 CRC 寄存器(SPIx_RXCRCR)

地址偏移 : 0x14

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RxCRC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 15:0 RXCRC [15:0]： RX CRC 寄存器

当启用 CRC 计算功能，RxCRC [15:0] 位包含根据收到的字节计算出来的 CRC 值。当 SPIx_CR1 寄存器的 CRCEN 位被写为 1 的时候，这个寄存器被复位。 CRC 计算使用 SPI_CRCPR 中的多项式。

当数据帧格式被设置为 8 位（SPIx_CR1 的 CRCL 位为 0）时，仅低 8 位参与计算，并且按照 CRC8 的方法进行；

当数据帧格式为 16 位（SPIx_CR1 的 CRCL 位为 1）时，寄存器中的所有 16 位都参与计算，并且按照 CRC16 的方法进行；

注： 在 BSY 标志为 1 的期间读这个寄存器的值，可能得到一个不正确结果。

I2S 模式下不使用。

25.7.7 SPI 发送 CRC 寄存器(SPIx_TXCRCR)

地址偏移 : 0x18

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TxCRC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 15:0 TxCRC[15:0]: Tx CRC 寄存器

当启用 CRC 计算功能，TxCRC [7:0] 位包含根据发送的字节计算出来的 CRC 值。当 SPIx_CR1 寄存器的 CRCEN 位被写为 1 的时候，这个寄存器被复位。 CRC 计算使用 SPI_CRCPR 中的多项式。

当数据帧格式被设置为 8 位（SPIx_CR1 中的 CRCL 位为 0）时，仅低 8 位参与计算，并且按照 CRC8 的方法进行；

当数据帧格式为 16 位（SPIx_CR1 的 CRCL 位为 1）时，寄存器中的所有 16 位都参与计算， 并且按照 CRC16 的方法进行；

注： 在 BSY 标志为 1 的期间读这个寄存器的值，可能得到一个不正确结果。

注： I2S 模式下不使用。

25.7.8 SPIx_I²S 配置寄存器(SPIx_I2SCFGR)

地址偏移 : 0x1C

复位值 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				I2SMOD	I2SE	I2SCFG		PCMSYNC	Reserved	I2SSTD		CKPOL	DATLEN		CHLEN
				rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw

位 15:12 保留： 由硬件强制为 0

位 11 I2SMOD： I2S 模式选择

0: 选中 SPI 模式

1: 选中 I2S 模式

注： 应该在 SPI 或 I2S 被禁用时，改变这个位。

位 10 I2SE： I2S 使能

0: I2S 的外设被禁用

1: I2S 外设被使能

注： SPI 模式不使用

位 9:8 I2SCFG： I2S 配置模式

00: 从机 - 发送

01: 从机 - 接收

10: 主机 - 发送

11: 主机 - 接收

注： 应该在 SPI 或 I2S 被禁用时，改变这个位。

SPI 模式下不使用。

位 7 PCMSYNC： PCM 帧同步

0: 短帧同步

1: 长帧同步

注： 此位有只有在 I2SSTD = 11（PCM 标准）的时候有意义，SPI 模式下不使用。

位 6 保留： 由硬件强制为 0

位 5:4 I2SSTD： I2S 标准选择

00: I2S 飞利浦标准

01: 高字节对齐标准（左对齐）

10: 低字节对齐标准（右对齐）

11: PCM 标准

I2S 标准的更多细节，请参阅 652 页的第 25.6.2 小结

注： 为了正确的操作，这些位应该在 I2S 被禁用的时候配置。

SPI 模式不使用

位 3 CKPOL： 静止态时钟极性

0: I2S 时钟静止态为低电平；

1: I2S 时钟静止态为高电平；

注： 为了正确的操作，这些位应该在 I2S 被禁用的时候配置。

SPI 模式不使用

位 2:1 DATLEN： 待传输数据长度
00: 16 位数据长度
01: 24 位数据长度
10: 32 位数据长度
11: 不允许
注： 为了正确的操作，这些位应该在 I2S 被禁用的时候配置。
SPI 模式不使用

位 0 CHLEN： 声道长度（每个音频通道的数据位数）
0: 16 位宽
1: 32 位宽
只有在 DATLEN = 00 时该位的写操作才有意义，否则声道长度都由硬件固定为 32 位。
注： 为了正确的操作，这些位应该在 I2S 被禁用的时候配置。
SPI 模式不使用

25.7.9 SPIx_I²S 预分频寄存器(SPIx_I2SPR)

地址偏移 : 0x20
复位值 : 0000 0010（0x0002）

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						MCKOE	ODD	I2SDIV							
						rw	rw	rw							

位 15:10 保留： 由硬件强制为 0
位 9 MCKOE： 主时钟输出使能
0: 主时钟输出被禁用
1: 主时钟输出启用
注： 应该在 I2S 被禁用时，改变这个位。 它仅用于 I2S 为主模式的状态。
SPI 模式不使用

位 8 ODD： 奇系数预分频
0: 实际分频系数 = I2SDIV *2；
1: 实际分频系数 = （I2SDIV *2）+1；
请参阅 658 页第 25.6.3 小结
注： 应该在 I2S 被禁用时，改变这个位。 它仅用于 I2S 为主模式的状态。
SPI 模式不使用

位 7:0 I2SDIV： I2S 线性预分频器
不允许设置 I2SDIV [7:0] =0 或者 I2SDIV [7:0] =1 的值。请参阅第 658 页 25.6.3 小结
注： 应该在 I2S 被禁用时，改变这个位。 它仅用于 I2S 为主模式的状态。
SPI 模式不使用

25.7.10 SPI/I2S 寄存器地址映象

表 103 给出了 SPI/I2S 寄存器镜像和复位值。

表 103. SPI 寄存器镜像和复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**SPIx_CR1**	Reserved																BIDIMODE	BIDIOE	CRCEN	CRCNEXT	CRCL	RXONLY	SSM	SSI	LSBFIRST	SPE	BR [2:0]			MSTR	CPOL	CPHA
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	**SPIx_CR2**	Reserved																	LDMA_TX	LDMA_RX	FRXTH	DS[3:0]				TXEIE	RXNEIE	ERRIE	FRF	NSSP	SSOE	TXDMAEN	RXDMAEN
	Reset value																		0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
0x08	**SPIx_SR**	Reserved																				FTLVL[1:0]		FRLVL[1:0]		FRE	BSY	OVR	MODF	CRCERR	UDR	CHSIDE	TXE	RXNE
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	1	0
0x0C	**SPIx_DR**	Reserved																DR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	**SPIx_CRCPR**	Reserved																CRCPOLY[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0x14	**SPIx_RXCRCR**	Reserved																RxCRC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	**SPIx_TXCRCR**	Reserved																TxCRC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	**SPIx_I2SCFGR**	Reserved																				I2SMOD	I2SE	I2SCFG		PCMSYNC	Reserved	I2SSTD		CKPOL	DATLEN		CHLEN
	Reset value																					0	0	0	0	0		0	0	0	0	0	0
0x20	**SPIx_I2SPR**	Reserved																						MCKOE	ODD	I2SDIV							
	Reset value																							0	0	0	0	0	0	0	0	1	0

26 触摸传感控制器 (TSC)

26.1 简介

触摸传感控制器提供了给任何应用增加电容式触摸传感功能的一个简单的解决方案。 电容式触摸解决方案可以检测与电极接近的手指，从而避免了触摸按键设计中出现的电气直接接触现象。通过检测有手指(或者其它导体)所引入的电容变化的方法被来判断是否存在触摸按键。

触摸传感控制器全面支持 STMTouch 触摸传感固件库，该固件库可以免费使用。通过该功能库可以轻松完成可靠的触摸式按键的设计。

26.2 TSC 主要功能

触摸传感控制器包含下列主要功能：

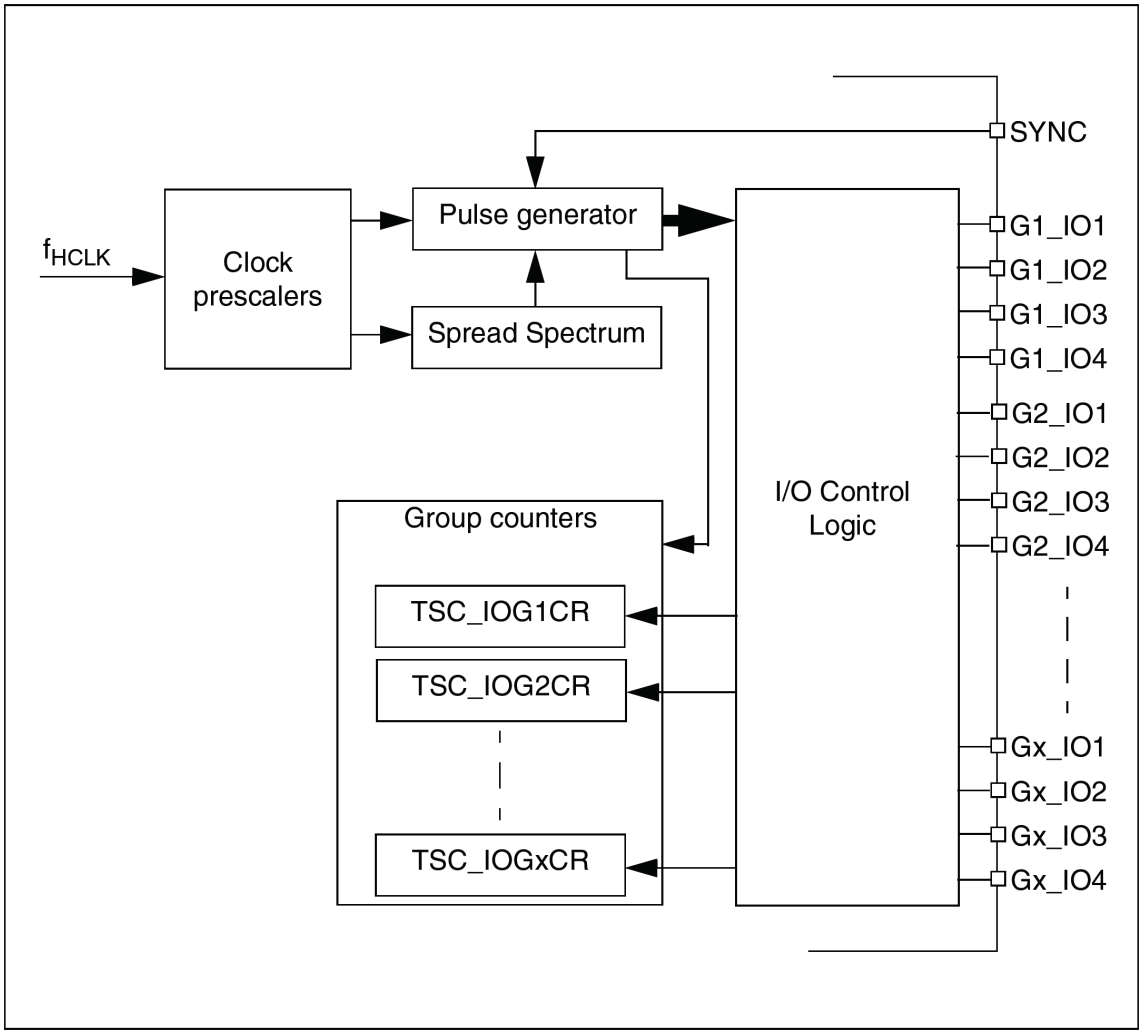
- 可靠的电荷迁移采集检测原理
- 支持多达 18 个电容传感通道
- 多达 6 个模拟 I/O 口组，并发检测，提供非常好的响应时间。
- 扩展频谱功能可提高噪声环境中的鲁棒性
- 全硬件管理充放电次序
- 充放电频率可设定
- I/O 引脚采样功能可设置
- 通道 I/O 可选择
- 可设置的最大计数值用以防止通道失效
- 结果和最大计数错误可引起中断
- 单个采样电容服务 3 个电容传感通道以限制零件个数
- 兼容滑条，触摸键，线性和旋转的触摸传感
- 有 STMTouch 触摸传感固件库直接支持

26.3 TSC 功能描述

26.3.1 TSC 框图

图 282 显示触摸传感控制器框图 TSC 框图

图 282. TSC 框图



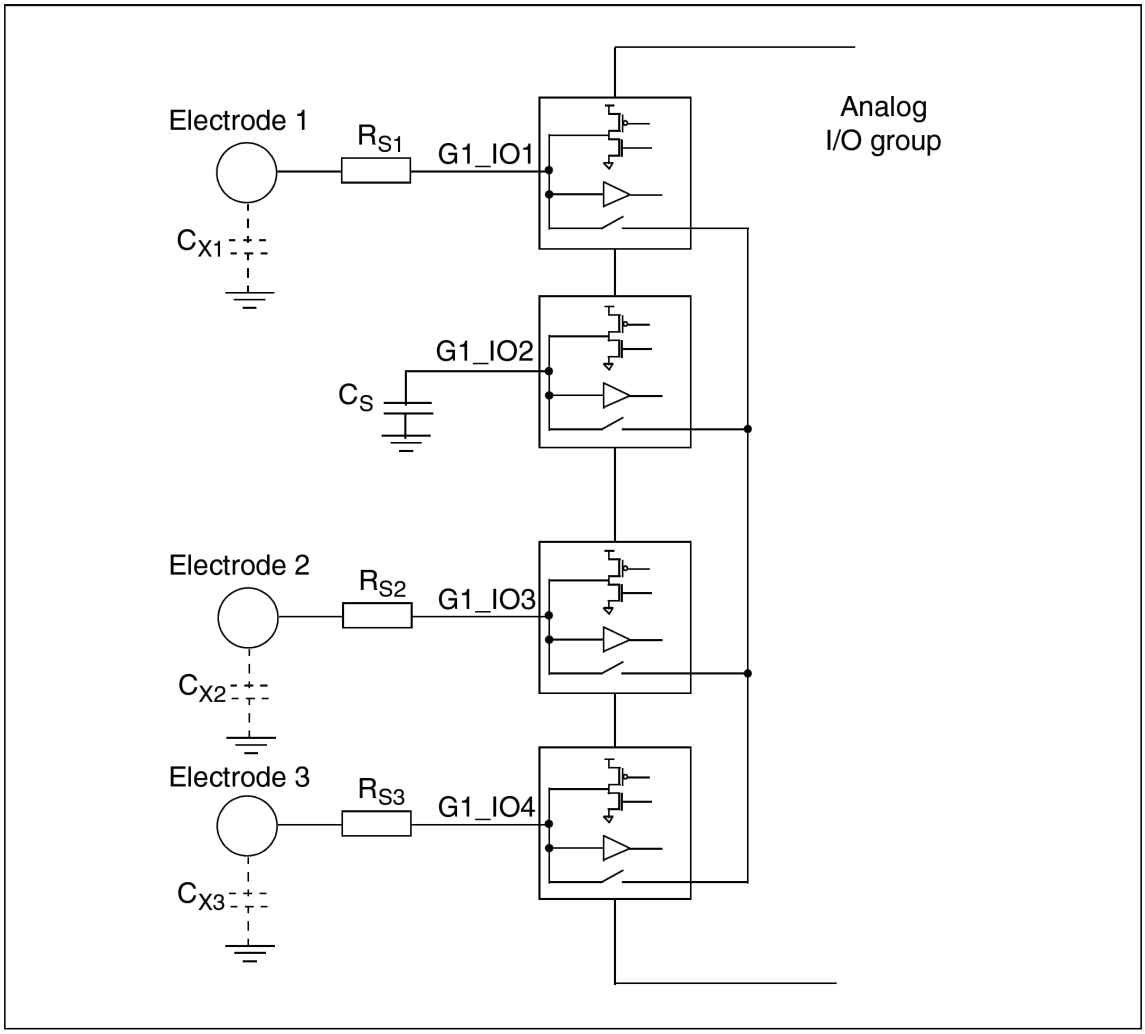
26.3.2 表面电荷迁移采样概述

表面电荷迁移采样是一套经过验证的，可靠有效的测量电容量的方法。 利用单端电极和最小数量的外围器件。 该方法利用带有模拟功能的 I/O 口来设计完成（见图 283）. 若干模拟 I/O 口组可以同时为多个电容传感通道服务从而能够支持更多的电容传感通道。 对于同一个模拟 I/O 口组，各个电容传感通道是按顺序采集信号。

每个采样电容 CS 占用一个 GPIO 口。 每个模拟 I/O 口组只能使用一个采样电容 I/O。

其余的 GPIOs 被用来连接电极，就算是一个通道。作为一些特殊需求设计时（例如接近检测），每个模拟 I/O 口组还可以同时连接多个通道。

图 283. 表面电荷迁移模拟 I/O 口组的结构



注： *Gx_IOy x 指模拟 I/O 口组的编号，y 指所属的组别中的 GPIO 口编号。*

表面电荷迁移检测原理由一个电极电容和一个向采样电容转移电荷的通道组成。 当采样电容 CS 上的电压达到一个预设的门限后顺序执行下一个通道的转移过程。 充电达到门限所需的电荷量和电极电容量直接相关。

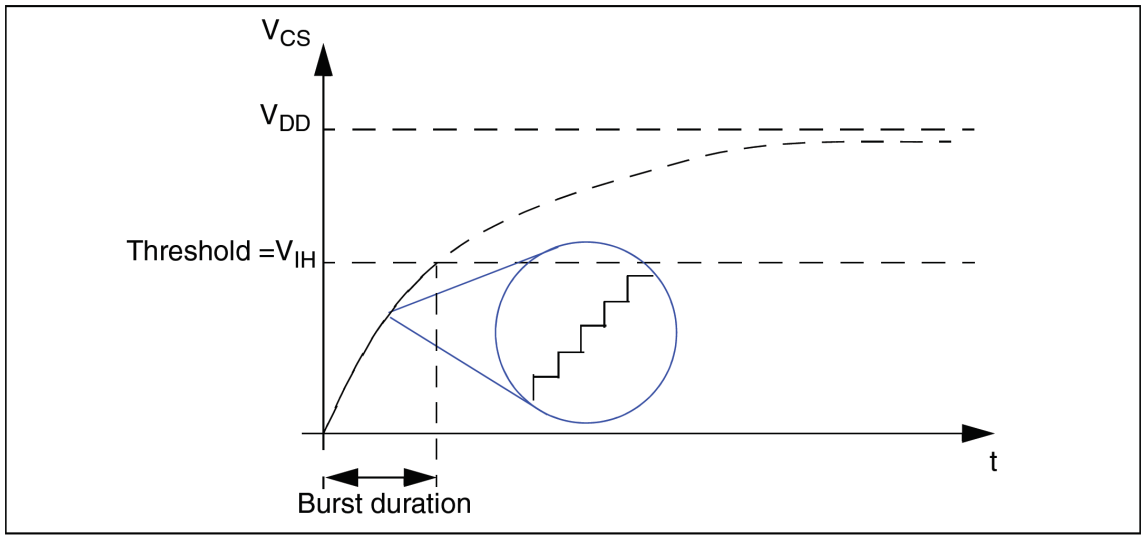
表 104 详细列出了电容传感通道 1 的电荷迁移收集顺序。 重复执行状态 3 到状态 7 的步骤直到 CS 上的电压达到指定门限。 在其它通道上也执行相同的采集顺序。 电极串联电阻可以提高该方案的 ESD 免疫力。

表 104. 采集顺序一览

状态	G1_IO1 (Electrode)	G1_IO2 (Sampling)	G1_IO3 (Electrode)	G1_IO4 (Electrode)	描述
#1	输入悬空 模拟开关关闭	输出开漏低 模拟开关关闭	输入悬空 模拟开关关闭	输入悬空 模拟开关关闭	放空全部的 C_x 和 C_s
#2	输入悬空	输入悬空	输入悬空	输入悬空	死区时间
#3	输出推拉高	输入悬空	输入悬空	输入悬空	充电 C_{x1}
#4	输入悬空	输入悬空	输入悬空	输入悬空	死区时间
#5	输入悬空 模拟开关关闭	输入悬空 模拟开关关闭	输入悬空	输入悬空	电荷转移从 C_{x1} to C_s
#6	输入悬空	输入悬空	输入悬空	输入悬空	死区时间
#7	输入悬空	输入悬空	输入悬空	输入悬空	测量 C_s 电压

时间段内采样电容上的电压变化情况如下：

图 284. 采样电容电压变化



26.3.3 复位和时钟

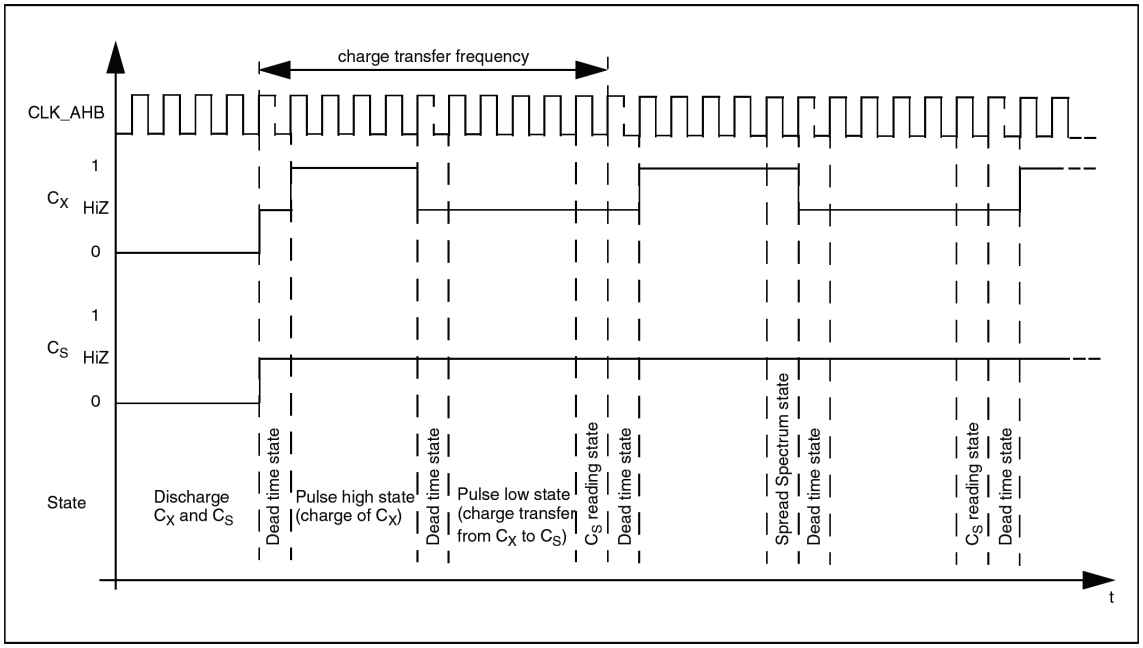
TSC 部分的时钟源是 AHB 时钟（fHCLK）。脉冲发生器和扩展频谱的内部时钟由两个可编程预分频器产生：

- 脉冲发生器时钟（fPGCLK）由 TSC_CR 寄存器中的 PGPSC[2:0] 位来决定。
- 扩展频谱时钟（fSSCLK）由 TSC_CR 寄存器中的 SSPSC 位决定。

复位和时钟控制器提供专门的控制位来使能触摸控制器时钟和复位整个外设。 更多信息，请参考第 7 章： 复位和时钟控制（RCC）

26.3.4 电荷转移采集顺序

图 285 显示了电荷迁移采集的顺序的例子。



为了更高的灵活性，电荷迁移的频率是可设置的。 脉冲的高电平（C_x 的充电）和脉冲的低电平（从 C_x 转移电荷到 C_s）的持续时间都可以在 TSC_CR 寄存器中的 CTPH[3:0] 和 CTPL[3:0] 位单独设置。 脉冲的高电平和低电平的时长通常为 500ns 到 2uS。 为了确保对电极电容的测量准确度，脉冲高电平持续时间必须确保能够将 C_x 充满。

死区时间是插在充电和电荷采集过程之间的一段时间。这段时间内充电开关和电荷迁移开关都处于断开状态，用来保证稳定正确的电荷迁移采集顺序。这个阶段的持续时间为 f_{HCLK} 的 2 个周期。

如果打开了扩展频谱功能，那么会在脉冲高电平状态的尾部，再加上若干个 f_{SSCLK} 的时钟周期。

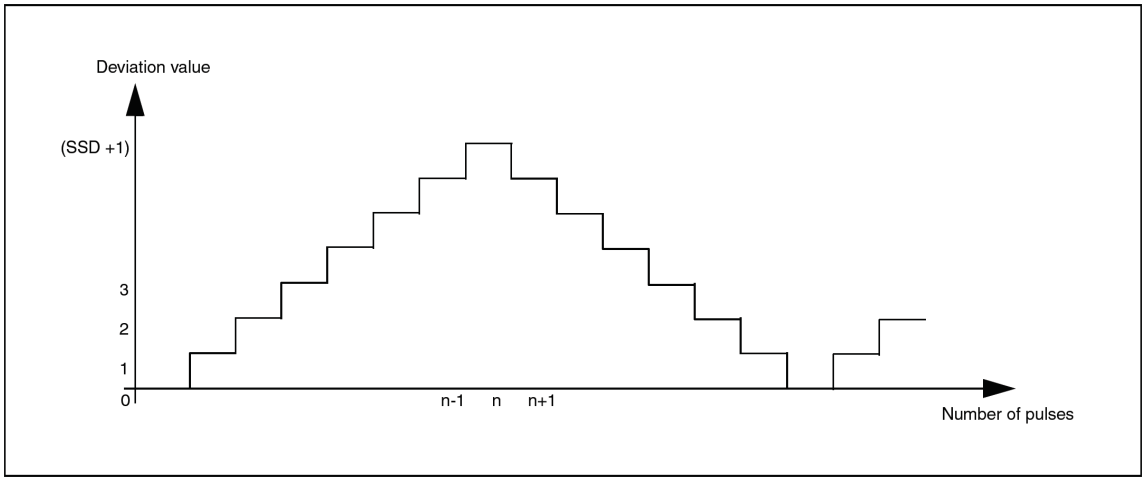
在脉冲低点品快要结束时，通过读取采样电容的 I/O 口状态，可以知道 C_s 电压是否已经达到预定门限，这个过程是一个 f_{HCLK} 时钟周期。

26.3.5 扩展频谱功能

扩展频谱功能允许产生可变的充放电频率。这可以用来提高在噪声环境下电荷迁移采集检测的鲁棒性，同时可以限制感应信号扩撒。在正常的充放电周期的基础上可以有 10% 到 50% 的调整空间。例如，正常的充放电频率是 250KHz（4us），典型的扩展频谱误差是 10%（400ns）导致最小充放电频率变成大约 227KHz。

实际上，使用下列原理可以在脉冲高电平状态中插入若干个 f_{SSCLK} 时钟周期：

图 286. 扩展频谱可变原理



下表给出了不同的 f_{HCLK} 设置下，最大的频率调整范围。

表 105. 扩展频率范围 VSAHB 时钟频率

f_{HCLK}	Spread spectrum step	Maximum spread spectrum deviation
24 MHz	41.6 ns	10666.6 ns
48 MHz	20.8 ns	5333.3 ns

可以在 TSC_CR 寄存器中使用 SSE 位来使能或禁止扩展频谱功能。 通过 TSC_CR 寄存器中的 SSD[6:0] 还可以设置为基于 HCLK 进行频率调整。

26.3.6 最大计数误差

最大计数误差机制可以用来防范针对有缺陷的通道进行采样时的超时问题。 包括基于每个模拟 I/O 口组的计数器的一个最大计数值限制。 这个限制具体在 TSC_CR 寄存器中的 MCV[2:0] 指定。当采集组计数器达到设定值的时候，正在进行的采集操作会被立即停止。同时，采集结束标志（EOAF 位）和超计数错误标志（MCEF 位）会同时由硬件置 1. 如果相关的中断使能位（EOAIE 和 MCEIE）的状态是 1，那么就会由此产生相关的中断请求。

26.3.7 采样电容 I/O 和通道 I/O 模式选择

要令 GPIO 供触摸传感控制器来驱策，必须通过标准的 GPIO 寄存器和 GPIOxAFR 寄存器打开对应的特殊功能。

GPIO 口由 TSC 控制的模式由 TSC_IOSCR 和 TSC_IOCCR 寄存器来决定。

当没有采集过程发生时，所有触摸传感控制器管辖的 I/O 口都处于默认状态。 当采集过程发生时，就只有未使用的 I/O 口（无论是定义为采样电容 I/O 还是通道 I/O）处于默认状态。TSC_CR 寄存器中的 IODEF 位决定了默认状态下的 I/O 口的设置。 下表介绍了 I/O 口在不同状态下的工作模式。

表 106. I/O 模式，工作状态和 IODEF 位的值的关系

IODEF bit	采集状态	Unused I/Omode	Electrode I/Omode	Sampling capacitor I/O mode
0（推拉输出 低）	No	推拉输出低	推拉输出低	推拉输出低
0（推拉输出 低）	过程中	推拉输出低	-	-
1 输入 高阻	No	输入 高阻	输入 高阻	输入 高阻
1 输入 高阻	过程中	输入 高阻	-	-

未使用的 I/O 模式

未使用的 I/O 指的是由 TSC 控制的 GPIO 但既未指定为电极 I/O 又未指定为采样电容 I/O 的。

采样电容 I/O 模式

对于允许 TSC 控制的采样电容 I/O 的脚，首先要将 I/O 口设置为特殊功能开漏输出的模式，同时对应的 YSC_IOSCR 寄存器中的 GxIOy 位必须置 1.

任何时候都只有一个采样电容 I/O 和一个模拟 I/O 组相关联。

通道 I/O 模式

对于允许 TSC 控制的通道 I/O 的脚，首先要将 I/O 口设置为特殊功能推拉输出的模式，同时对应的 YSC_IOSCR 寄存器中的 GxIOy 位必须置 1.

做接近检测时，需要一个更大等效面积的电极以便加速电荷采集的过程，也可以在同一个模拟 I/O 组下面打开多个通道同时检测。

26.3.8 采集模式

触摸传感控制器提供两种采集模式：

- 常规采集模式：在 TSC_CR 寄存器中的 START 位被置 1 的时候立即开始采集。
- 同步采集模式： 采集由 TSC_CR 寄存器中的 START 位来使能，但并不马上开始，而是在同步输入引脚上检测到的下降沿或上升沿和高电平来触发开始。 这个模式在需要用外部信号同步各个电容传感通道采集的时候很有用，也不增加 CPU 的负担。

TSC_IOGCSR 寄存器中的 GxE 位决定所使能的模拟 I/O 组的组别（对应的计数器开始计数）。未被使能的模拟 I/O 组的 Cs 电压不会被检测，也不参与对采集结束标志的触发动作。 然而，这并不影响对这些被禁止 I/O 组所包含的通道的充电动作。

当工作态模拟 I/O 组的 Cs 电压达到指定门限时，TSC_IOGCSR 寄存器中对应的 GxS 位会由硬件置 1。 当全部的活动的模拟 I/O 组的采集工作完成后（全部的 GxS 位被置 1），TSC_ISR 寄存器中的 EOAF 标志会被置 1。 如果这时 TSC_IER 寄存器中的 EOAIE 位是 1，那就会产生一个中断请求。

如果有最大计数错误发生，那么进行中的采集动作会被停止，同时 TSC_ISR 寄存器中的 EOAF 和 MCEF 标志会被同时置 1。只要相关的中断使能位为 1（TSC_IER 寄存器中的 EOAIE 和 MCEIE），任何这种事件都可以触发中断请求。注意当最大计数错误被检测到的时候，剩余的 GxS 位不会被置位。

要清除中断标志，必须对 TSC_ICR 寄存器中的 EOAIC 和 MCEIC 位写 1.

当新的一轮采集开始时，模拟 I/O 组计数器会自动清零。 它们会不断的被对应通道的采集周期更新，直到采集过程结束。

26.3.9 I/O 滞回和模拟开关控制

为了提供更高的灵活性，触摸感应控制器也提供针对每个模拟 I/O 组的施密特迟滞控制。 该控制功能和 I/O 的控制模式没有关系（GPIO 控制寄存器或者其它外设都可以），假定是由触摸感应控制器来控制。 这一点对于某些应用下需要差别化的采样时很有用。

为了提高抗干扰能力，TSC 控制下的 GPIO 口的施密特滞后触发功能必须关闭，方法是将 TSC_IOHCR 寄存器下的对应 Gx_IOy 位清零。

26.3.10 电容传感 GPIO

下表所示为 STM32F05xx 系列器件中可以用作电容传感的 GPIO 一览。

表 107. STM32F05xx 器件可作电容传感的 GPIO

引脚名	电容传感组名	引脚名	电容传感组名
PA0	G1_IO1	PA9	G4_IO1
PA1	G1_IO2	PA10	G4_IO2
PA2	G1_IO3	PA11	G4_IO3
PA3	G1_IO4	PA12	G4_IO4
PA4	G2_IO1	PB3	G5_IO1
PA5	G2_IO2	PB4	G5_IO2
PA6	G2_IO3	PB6	G5_IO3
PA7	G2_IO4	PB7	G5_IO4
PC5	G3_IO1	PB11	G6_IO1
PB0	G3_IO2	PB12	G6_IO2
PB1	G3_IO3	PB13	G6_IO3
PB2	G3_IO4	PB14	G6_IO4

26.4 TSC 低功耗模式

表 108. 低功耗模式对 TSC 的影响

模式	描述
Sleep	无影响 TSC 中断可将 MCU 从 Sleep 模式唤醒。
Stop	TSC 寄存器处于冻结状态 TSC 停止操作直到 MCU 退出 Stop 或 Standby 模式。
Standby	

26.5 TSC 中断

表 109. 中断控制位

中断事件	使能控制位	事件标志	清标志位	退出 Sleep 模式	退出 Stop 模式	退出 Standby 模式
结束采集	EOAIE	EOAIF	EOAIC	是	否	否
最大计数误差	MCEIE	MCEIF	MCEIC	是	否	否

26.6 TSC 寄存器

参见 31 页的章节 1.1 中对寄存器描述所采用的缩写列表

本外设寄存器可以按字（32 位）访问

26.6.1 TSC 控制寄存器(TSC_CR)

地址偏移 : 0x00

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CTPH[3:0]				CTPL[3:0]				SSD[6:0]							SSE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSPSC	PGPSC[2:0]			Res.	Res.	Res.	Res.	MCV[2:0]			IODEF	SYNC POL	AM	START	TSCE
rw	rw	rw	rw					rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 CTPH[3:0]: 电荷迁移脉冲高

由软件设置或清零 定义电荷迁移脉冲高电平的时长（为 C_x 充电）

0000: 1x t_{PGCLK}

0001: 2x t_{PGCLK}

...

1111: 16x t_{PGCLK}

注： 在采集过程中不可以改变这些值。

位 27:24 CTPL[3:0]: 电荷迁移脉冲低

由软件设置或清零 定义电荷迁移脉冲低电平的时长（由 C_x 迁移电荷到 C_s）。

0000: 1x t_{PGCLK}

0001: 2x t_{PGCLK}

...

1111: 16x t_{PGCLK}

注： 在采集过程中不可以改变这些值。

位 23:17 SSD[6:0]: 扩频误差

由软件设置或清零 定义扩频误差值，相应的 f_{SSCLK} 周期会被插入到充电脉冲高电平的周期中。

0000000: 1x t_{SSCLK}

0000001: 2x t_{SSCLK}

...

1111111: 128x t_{SSCLK}

注： 在采集过程中不可以改变这些值。

位 16 SSE: 扩展频谱使能

必须由软件设置和清除，用于使能 / 禁止扩频功能。

0: 扩频功能禁止

1: 扩频功能使能

注： 在采集过程中不可以改变这些值。

位 15 SSPSC: 扩频预分频器

由软件设置和清除。 用于选择 AHB 时钟的分频系数，来产生扩频时钟（f_{SSCLK}）。

0: f_{HCLK}

1: f_{HCLK} /2

注： 在采集过程中不可以改变这些值。

位 14:12 PGPSC[2:0]: 脉冲发生器预分频器

由软件设置和清零，用于选择 AHB 时钟分频系数，来产生脉冲发生器时钟（f_{PGCLK}）。

000: f_{HCLK}

001: f_{HCLK} /2

010: f_{HCLK} /4

011: f_{HCLK} /8

100: f_{HCLK} /16

101: f_{HCLK} /32

110: f_{HCLK} /64

111: f_{HCLK} /128

注： 在采集过程中不可以改变这些值。

位 11:8 保留，必须保持复位时的值。

位 7:5 MCV[2:0]: 最大计数错误

由软件设置或清零 用来定义在产生最大计数错误前的最大电荷迁移脉冲的个数。

000: 255

001: 511

010: 1023

011: 2047

100: 4095

101: 8191

110: 16383

111: 保留

注： 在采集过程中不可以改变这些值。

位 4 IODEF: I/O 默认模式

由软件设置和清除。 用于定义没有处于采样状态时的全部触摸传感 I/O 的设置。 当有采样动作时，定义所有未使用的 IO 口的设置（未被用于采样电容 IO 或者通道 IO 的）。

0: I/O 口被强制设为推拉输出低电平

1: I/O 口处于输入高阻状态

注： 在采集过程中不可以改变这些值。

位 3 SYNCPOL: 同步引脚极性

由软件设置和清零，用于选则同步输入引脚的信号极性。

0: 仅下降沿

1: 上升沿和高电平

位 2 AM: 采集模式
由软件设置和清零，用于选择采集模式。
0: 普通采集模式（START 位置 1 时立即开始采集）
1: 同步采集模式（当 START 位被置 1 的情况下收到同步触发信号才开始采集）
注： 在采集过程中不可以改变这些值。

位 1 START: 启动新的采集过程
由软件置 1 以启动新的采集过程。 由硬件在采集结束或者软件取消一次采集过程时清零。
0: 采集未启动
1: 启动新的采集过程

位 0 TSCE: 触摸传感控制器 (TSC) 使能
必须由软件设置和清除，用于使能 / 禁止触摸感应控制器。
0: 触摸传感控制器禁止
1: 触摸传感控制器使能
注： 当触摸感应控制器被禁止时，对 TSC 寄存器的设置将不会起作用。

26.6.2 TSC 中断使能寄存器(TSC_IER)

地址偏移 : 0x04
上电复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCEIE	EOAIE
														rw	rw

位 31:2 保留，必须保持复位时的值。

位 1 MCEIE: 最大计数错误中断使能位
必须由软件设置和清除，用于使能 / 禁止最大计数错误中断。
0: 最大计数错误中断禁止
1: 最大计数错误中断使能

位 0 EOAIE: 采集结束中断使能位
由软件设置和清除，用于使能 / 禁止采集结束中断。
0: 采集结束中断禁止
1: 采集结束中断使能

26.6.3 TSC 中断清除寄存器(TSC_ICR)

地址偏移 : 0x08

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCEIC	EOAIC
														rw	rw

位 31:2 保留，必须保持复位时的值。

位 1 MCEIC: 最大计数错误中断清除位

由软件置位和清零，用于清除最大计数错误标志，当标志位被清零之后，本位自动由硬件清零。 写 '0' 不会有作用。

0: 无影响

1: 清除 TSC_ISR 寄存器中的对应 MCEF 标志

位 0 EOAIC: 采集结束中断清除位

由软件置位和清零，用于清除采集结束标志，当标志位被清零之后，本位自动由硬件清零。 写 '0' 不会有作用。

0: 无影响

1: 清除 TSC_ISR 寄存器中的对应 EOAF 标志

26.6.4 TSC 中断状态寄存器(TSC_ISR)

地址偏移 : 0x0C

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCEF	EOAF
														rw	rw

位 31:2 保留，必须保持复位时的值。

位 1 MCEF: 最大计数错误标志

在一个模拟 I/O 口组的计数器达到设定的最大计数值时，由硬件置 1。 在 TSC_ICR 的 MCEIC 位上写 1 可清除该位。

0: 没有发现最大计数错误（MCE）

1: 发现最大计数错误（MCE）

位 0 EOAF: 结束采集标志

在全部的有效的模拟 I/O 口组都完成采集过程（全部的 GxS 位为 1）或者发生最大计数错误事件后，由硬件置 1。 在 TSC_ICR 的 EOAIC 位上写 1 可清除该位。

0: 采集过程未完成

1: 采集过程已完成

26.6.5 TSC I/O 口迟滞控制寄存器(TSC_IOHCR)

地址偏移 : 0x10

复位值 : 0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位时的值。

位 23:0 Gx_IOy: Gx_IOy 施密特触发迟滞模式

由软件设置和清零，用于使能 / 禁止 Gx_IOy 施密特迟滞触发。

0: 禁止 Gx_IOy 施密特触发延迟

1: 使能 Gx_IOy 施密特迟滞延迟

注： 该位用于控制 I/O 口施密特触发延迟功能，无论 I/O 控制模式怎样（甚至由标准的 GPIO 寄存器控制时）。

26.6.6 TSC I/O 模拟开关控制寄存器(TSC_IOASCR)

地址偏移 : 0x18

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位时的值。

位 23:0 Gx_IOy: Gx_IOy 模拟开关使能

由软件设置和清零，用于使能 / 禁止 Gx_IOy 模拟开关

0: Gx_IOy 模拟开关禁止（开路状态）

1: Gx_IOy 模拟开关使能（闭合状态）

注： 该位用于控制 I/O 口模拟开关，无论 I/O 控制模式怎样（甚至由标准的 GPIO 寄存器控制时）。

26.6.7 TSC I/O 口采样控制寄存器(TSC_IOSCR)

地址偏移 : 0x20

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位时的值。

位 23:0 Gx_IOy: GxIOy 采样模式

由软件设置和清零，用于将 Gx_IOy 配置为采样电容接口 . 一个模拟 I/O 口组只能定义一个采样电容。

0: Gx_IOy 不用做采样电容

1: GxIOy 用作采样电容

注： 在采集过程中不可以改变这些值。

26.6.8 TSC I/O 通道控制寄存器 (TSC_IOCCR)

地址偏移 : 0x28

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位时的值。

位 23:0 Gx_IOy: GxIOy 通道模式

由软件设置和清零，用于将 Gx_IOy 配置为通道 I/O。

0: Gx_IOy 不用做通道 I/O

1: Gx_IOy 用作通道 I/O

注： 在采集过程中不可以改变这些值。

26.6.9 TSC I/O 组控制状态寄存器(TSC_IOGCSR)

地址偏移 : 0x30

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	G6S	G5S	G4S	G3S	G2S	G1S
										r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	G6E	G5E	G4E	G3E	G2E	G1E
										rw	rw	rw	rw	rw	rw

位 31:22 保留，必须保持复位时的值。

位 21:16 GxS: 模拟 I/O 口组 x 的状态

当对应的模拟 I/O 口组被使能，并且采集周期结束时由硬件置位。 当新的采集过程启动时由硬件清零。

0: 模拟 I/O 口组 x 的采集动作未启动或处于过程中

1: 模拟 I/O 口组 x 的采集动作已经完成

注： 注意当最大计数错误被检测到的时候，剩余的 GxS 位不会被置位。

位 15:6 保留，必须保持复位时的值。

位 5:0 GxE: 模拟 I/O 口组 x 的状态

由软件置位和清零，用于使能 / 禁止对应模拟 I/O 口组 x 上的采集过程。

0: 禁止模拟 I/O 口组 x 的采集动作

1: 使能模拟 I/O 口组 x 的采集动作

26.6.10 TSC I/O 组 x 计数寄存器（ TSC_IOGxCR）（ x=1..6）

地址偏移 : 0x30 + + 0x04 x 模拟 I/O 组号

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	CNT[13:0]													
		r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:14 保留，必须保持复位时的值。

位 13:0 CNT[13:0]: 计数器值

表示对应模拟 I/O 口组完成它们的采集 (Cs 电压达到设定门限) 所经过的电荷迁移周期数。

26.6.11 TSC 寄存器镜像

表 110. TSC 寄存器镜像和复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0000	TSC_CR	CTPH[3:0]				CTPL[3:0]				SSD[6:0]							SSE	SSPSC	PGPSC[2:0]			Res.	Res.	Res.	Res.	MCV[2:0]			IODEF	SYNCPOL	AM	START	TSCE
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0
0x0004	TSC_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCEIE	EOAIE
	Reset value																															0	0
0x0008	TSC_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCEIC	EOAIC
	Reset value																															0	0
0x000C	TSC_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCEF	EOAF
	Reset value																															0	0
0x0010	TSC_IOHCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1	G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1
	Reset value									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x0014	Reserved																																
0x0018	TSC_IOASCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1	G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x001C	Reserved																																
0x0020	TSC_IOSCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1	G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0024	Reserved																																
0x0028	TSC_IOCCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1	G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 110. TSC 寄存器镜像和复位值（续）

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x002C	Reserved																																
0x0030	TSC_IOGCSR	G16S	G15S	G14S	G13S	G12S	G11S	G10S	G9S	G8S	G7S	G6S	G5S	G4S	G3S	G2S	G1S	G16E	G15E	G14E	G13E	G12E	G11E	G10E	G9E	G8E	G7E	G6E	G5E	G4E	G3E	G2E	G1E
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0034	TSC_IOG1CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[13:0]													
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0038	TSC_IOG2CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[13:0]													
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x003C	TSC_IOG3CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[13:0]													
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0040	TSC_IOG4CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[13:0]													
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0044	TSC_IOG5CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[13:0]													
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0048	TSC_IOG6CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[13:0]													
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0

寄存器边界地址请参见 35 页的章节 2.2.2

27 HDMI-CEC 控制器（HDMI-CEC）

27.1 简介

消费电子控制（CEC）HDMI（高清多媒体接口）标准见附录补充 1。

它包括了一个提供了所有的用户环境中的各种音像制品之间的高层次的控制功能的协议。 它已被指定在低速运行，以得到最小的处理和内存开销。

HDMI-CEC 控制器提供针对这个协议的硬件支持。

27.2 HDMI-CEC 控制器的主要功能

- 符合 HDMI-CEC 的 v1.3a 规范
- 32KHz CEC 的内核具有 2 个时钟源选择
  - HSI RC 振荡器的固定分频器（HSI/255）
  - LSE 振荡器
- 超低功耗应用中可工作在 Stop 模式
- 可配置的信号传输开始前的空闲时间
  - 根据 CEC 状态和传输历史由硬件自动控制
  - 由软件修正（7 个定时选项）
- 可配置的外设地址（OAR）
- 支持侦听模式
  - 可以接收不仅限于与 OAR 相同目标地址的数据帧，而不会干扰 CEC 线。
- 可配置的 RX 容忍裕量
  - 标准容忍度
  - 扩展容忍度
- 接收错误检测
  - 位上升错误（BRE），可选的接收停止位（BRESTP）
  - 短位错误（SBPE）
  - 长位错误（LBPE）
- 可配置的错误位生成
  - BRE 的检测（BREGEN）
  - LBPE 的检测（LBPEGEN）
  - SBPE 检测时总是哦生成
- 传输错误检测（TXERR）
- 仲裁丢失检测（ARBLST）
  - 带自动发送重试
- 传输欠载检测（TXUDR）
- 接待过载检测（RXOVR）

27.3 HDMI-CEC 描述

27.3.1 HDMI-CEC 引脚

CEC 总线用单根双向线来发送和接收数据。 它通过一个 27kΩ 的上拉电阻连接到 +3.3 V 电源电压。 该设备的输出级，必须用漏极开路或集电极开路方式，以允许线路连接的逻辑与。

HDMI-CEC 控制器将 CEC 双向线作为一个标准的 GPIO 附加功能来用，GPIO 选择的是附加功能的开漏输出状态。 27kΩ 的上拉电阻必须在 STM32 外部追加。

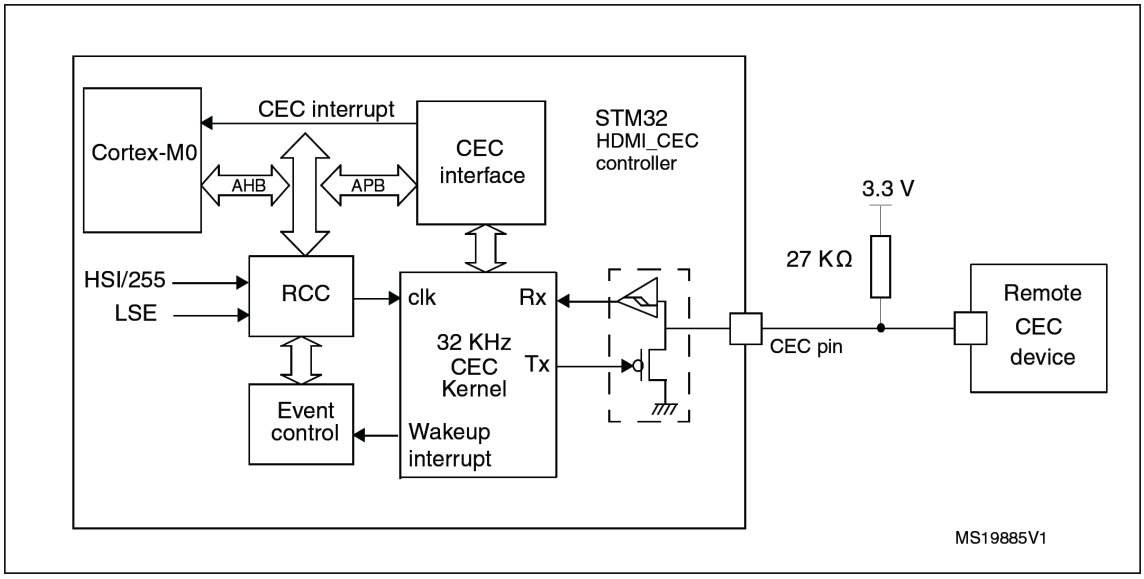
表 111. HDMI-CEC 引脚

名称	信号类型	备注
CEC	双向	两种状态：1 = 高阻 0 = 低阻外部必须添加一个 27kΩ 的上拉电阻。

27.4 功能描述

27.4.1 框图

图 287. 框图



1. GPIO 配置成附加功能开漏输出
2. 当配置为输出开漏附加功能，施密特触发器仍然激活。

27.5 HDMI-CEC 的寄存器

参见 31 页的章节 1.1 中对寄存器描述所采用的缩写列表

27.5.1 CEC 控制寄存器（CEC_CR 中）

地址偏移 : 0x00

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TX EOM	TX SOM	CEC ON
													rs	rs	rw

位 31:3 保留，必须保持复位时的值。

位 2 TXEOM: TX 消息结束

TXEOM 位在软件发送一个 CEC 消息的最后一个字节时置 1。 TXEOM 在同时和同条件下和 TXSOM 一起由硬件清零。

0: TXDR 数据字节按 EOM=0 的方式发送

1: TXDR 数据字节按 EOM=1 的方式发送

注： 在 CECEN=1 的时候 TXEOM 必须为 1

TXEOM 必须在将发送数据写到 TXDR 之前置 1。

如果在 TXSOM=0 的时候将 TXEOM 置 1，传输的消息将只会包含 1 个字节（头）（PING 消息）

位 1 TXSOM： TX 消息起始

TXSOM 位由软件置 1，以发送一个 CEC 消息的首字节。 如果 CEC 消息只包含一个字节，就必须在置 1 TXSOM 之前将 TXEOM 置 1。

Start-Bit is effectively started on the CEC line after SFT is counted. 如果 TXSOM 在消息的接收过程中被置 1，发送过程将在接收结束之后开始。

在消息的最后一个字节被发送完毕，又有正的确认（TXEND=1）时，TXSOM 会被硬件清零。如果发生了发送欠载（TXUDR=1），有负的确认（TXACKE=1），那么传输出错（TXERR=1）。 它在 CECEN=0 的时候也会被清零。 它在传输遇到仲裁丢失的时候自动重发时不会被清零。

它也可以被用作状态位，来向应用程序表明传输请求被挂起还是正在执行。 应用程序可以用清除 CECEN 位的方式随时终止一次传输。

0: 没有在进行 CEC 传输

1: CEC 传输命令

注： 在 CECEN=1 的时候 TXSOM 必须为 1

在发送数据给到 TXDR 时 TXSOM 必须为 1。

在仅用作接收时，帧头的前 4 位包含的自有外设地址取自 TXDR [7:4] 而不是 CEC_CFGR.OAR。

位 0 CECEN： CEC 使能

CECEN 位由软件设置和清除。 将 CECEN 置 1 将开始消息的接收和使能 TXSOM 控制。 将其清零则禁止 CEC 外设，并清除 CEC_CR 寄存器的所有位，同时终止任何正在执行的接收或者发送动作。

0: CEC 外设关闭

1: CEC 外设打开

27.5.2 CEC 配置寄存器(CEC_CFGR)

此寄存器用于配置 HDMI-CEC 控制器。

地址偏移 : 0x04

复位值 : 0x0000 0000

警告： 强制要求只能在 *CECEN=0* 的时候改写 *CEC_CFGR*。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
Res.	Res.	Res.	Res.	LBPE GEN	BRE GEN	BRE STP	RX TOL	SFT[2:0]			LSTN	OAR[3:0]			
				rw	rw	rw	rw	rw			rw	rw			

位 31:12 保留，必须保持复位时的值。

位 11 LBPEGEN： 在长位周期错误时产生错误位

LBPEGEN 位由软件设置和清除。

0: 在 CEC 线上检测到 LBPE 不产生错误位。

1: 在 CEC 线上检测到 LBPE 时产生错误位。

位 10 BREGEN： 在位上升错误时产生错误位

BREGEN 位由软件设置和清除。

0: 在 CEC 线上检测到 BRE 时不产生错误位。

1: 在 CEC 线上检测到 BRE 时产生错误位。

位 9 BRESTP： 在位上升错误时 Rx-Stop

BRESTP 位由软件设置和清除。

0: 在检测到 BRE 的时候不停止 CEC 消息接收。 数据位按 1.05ms 采样。

1: 在检测到 BRE 后停止消息接收。

位 8 RXTOL： Rx 容忍

RXTOL 位由软件设置和清除。

0: 标准容忍裕度：

- 起始位，+/- 200us 上升，+/- 200us 下降。
- 数据位： +/- 200us 上升。 +/- 350us 下降。

1: 扩展容忍度

- 起始位： +/- 400us 上升，+/- 400us 下降
- 数据位： +/- 300us 上升，+/- 500us 下降

位 7:5 SFT： 信号空闲时间

SFT 由软件设置。 在 SFT=0x0 的配置下，名义上的发送前的数据位等待周期由硬件按照传输历史来裁决。 在其它配置下则由软件决定。

- 0x0
  - 如果 CEC 是最后一个不成功传输的总线初始化方（ARBLST=1，TXERR=1，或 TXUDR=1），那么是 3 个标称的数据位周期。
  - 如果 CEC 是新的总线初始化方，那么是 5 个标称的数据位周期。
  - 如果 CEC 是最后的成功传输的总线初始化方（TXEOM=1），那么就是 7 个标称数据位周期。
- 0x1: 1.5 个标称数据位周期
- 0x2: 2.5 个标称数据位周期
- 0x3: 3.5 个标称数据位周期
- 0x4: 4.5 个标称数据位周期
- 0x5: 5.5 个标称数据位周期
- 0x6: 6.5 个标称数据位周期
- 0x7: 7.5 个标称数据位周期

位 4 LSTN： 侦听模式

LSTN 位由软件设置和清除。

0: CEC 外设只接收针对自己地址（OAR）的消息。 不同目标地址的消息会被忽略。广播消息总是会被接收。

1: CEC 外设接收到针对自己地址的消息时给出正的确认。不同地址的消息会被收到，但不妨碍 CEC 总线： 不发送确认。

位 3:0 OAR：本机地址

OAR 位由软件配置，用于设置 CEC 设备的本机地址。 仅用于接收模式。 在收到帧头之后，OAR 会用来和收到帧的目标地址进行比较。 地址匹配的消息会被接收。地址不匹配的消息只会在侦听模式（LSTN=1）下被收到，但不会发送确认。 广播消息总是会被接收。

注： 可以配置 OAR 为 0xF： 这时如果 LSTN=0 就只会收到广播消息，如果 LSTN=1 则会收到所有消息但全都不会发送确认。

27.5.3 CEC Tx 数据寄存器(CEC_TXDR)

地址偏移 : 0x8

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXD[7:0]							
								w	w	w	w	w	w	w	w

位 31:8 保留，必须保持复位时的值。

位 7:0 TXD[7:0]： Tx 数据寄存器。

TXDR 是一个只写寄存器，包含要发送的数据字节。

注： TXDR 只能在 TXSTART=1 的时候被写入。

27.5.4 CEC Rx 数据寄存器(CEC_RXDR)

地址偏移 : 0xC

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXD[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位时的值。

位 7:0 RXD[7:0]： Rx 数据寄存器。

RXD 是一个只读寄存器，包含了从 CEC 线上接收到的最后一个数据字节。

27.5.5 CEC 中断和状态寄存器(CEC_ISR)

地址偏移 : 0x10

复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	TX ACKE	TX ERR	TX UDR	TX END	TXBR	ARB LST	RX ACKE	LBPE	SBPE	BRE	RX OVR	RX END	RXBR
			rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:13 保留，必须保持复位时的值。

位 12 TXACKE： Tx-Missing 确认错误

在发送模式，TXACKE 会被硬件置位，用来通知应用程序在发送完后没有收到确认。在广播发送中，TXACKE 会通知应用程序说收到了一个负的确认。 TXACKE 位会终止消息的发送，并清除 TXSOM 和 TXEOM 控制位。

由软件写 1 时可以清除 TXACKE 位。

位 11 TXERR： Tx 错误

在发送模式中，如果 CEC 的初始化方在释放 CEC 线之后检测到 CEC 线还是低阻的，TXERR 会被硬件置 1。 TXERR 位会终止消息的发送，并清除 TXSOM 和 TXEOM 控制位。

由软件写 1 时可以清除 TXERR 位。

位 10 TXUDR： Tx 缓冲区欠载

在发送模式，如果应用程序没有及时将下一个要发送的字节加载到 TXDR，那么 TXUDR 会被硬件置 1。 TXUDR 位会终止消息的发送，并清除 TXSOM 和 TXEOM 控制位。

由软件写 1 时可以清除 TXUDR 位。

位 9 TXEND： 发送结束

TXEND 会被硬件置 1，用于通知应用程序 CEC 消息的最后一个自己已经被成功的发送出去了。 TXEND 位置 1 的同时会将 TXSOM 和 TXEOM 控制位清除掉。

由软件写 1 时可以清除 TXEND 位。

位 8 TXBR： 字节发送请求

TXBR 会被硬件置 1，以通知应用程序下一个要发送的数据必须马上被写到 TXDR 中。当前发送的字节的前 4 位被发送出去之后 TXBR 位就会被硬件置 1。应用程序必须在 6 个标称的数据位周期内，将下一个字节写到 TXDR，否则将出现发送欠载错误（TXUDR）。

由软件写 1 时可以清除 TXBR 位。

位 7 ARBLST： 仲裁丢失

ARBLST 由硬件置 1，用于通知应用程序，CEC 设备由于在 TXSOM 命令之后遇到了仲裁丢失现象，从而已经切换到了接收模式。 ARBLST 可能由于一个有竞争的 CEC 设备先启动，或者是由于一个具有更改的帧头优先级的消息在同时发送。 在遇到 ARBLST 事件后 TXSOM 会被保持，并尝试下一次的发送。

由软件写 1 时可以清除 ARBLST 位。

位 6 RXACKE： 接收中的确认错误

接收模式中，RXACKE 会被硬件置 1，以用来通知应用程序说没有在 CEC 线上收到确认。 RXACKE 只针对广播消息，以及在侦听模式下对于发给非本机地址的消息。RXACKE 终止消息的接收。

由软件写 1 时可以清除 RXACKE 位。

位 5 LBPE： Rx 长位周期错误

在数据位波形中检测到长位周期错误时，LBPE 位由硬件置 1。 在下降沿达到了 RXTOL 规定的最大的位扩展容忍度之后，LBPE 被硬件置 1。 LBPE 事件总会停止 CEC 消息的接收。 如果 LBPEGEN=1，LBPE 事件会在 CEC 线上产生一个位错误。在广播时，即便是 LBPEGEN=0 时也会产生位错误。

由软件写 1 时可以清除 LBPE 位。

位 4 SBPE： Rx 短位周期错误

在数据位波形中检测到短位周期错误时，SBPE 位由硬件置 1。 在预期的下降沿发生的同时，SBPE 被置 1。 在 CEC 线上检测到 SBPE 时会产生错误位。

由软件写 1 时可以清除 SBPE 位。

位 3 BRE： Rx 位上升错误

在数据位波形中检测到位上升错误时，BRE 位由硬件置 1。 BRE 被置 1 的原因既可能是发生错位的上升沿，或者是在上升沿还在持续的时间达到了 RXTOL 规定的最大 BRE 容忍度。 如果 BRESTP=1，BRE 事件会终止消息的接收。 如果 BREGEN=1，BRE 事件会在 CEC 线上产生一个位错误。

由软件写 1 时可以清除 BRE 位。

位 2 RXOVR： Rx 溢出

如果 RXBR 尚未清除的同时一个新的字节从 CEC 线上被收下来，并被存到 RXD 中，RXOVR 会被硬件置 1。 RXOVR 事件会终止消息的接收，所以不会发送确认。 在广播时，会发送一个负的确认。由软件写 1 时可以清除 RXOVR 位。

位 1 RXEND： 接收结束

RXEND 会被硬件置 1，用于通知应用程序：CEC 消息的最后一个字节已经被接收下来，并且被存储到 RXD 缓冲区。 RXEND 和 RXBR 会同时被置 1。

由软件写 1 时可以清除 RXEND 位。

位 0 RXBR： Rx 字节接收
RXBR 位被硬件置 1，用于通知应用程序从 CEC 线上收到了一个新的字节，并被存储到 RXD 缓冲区中。
由软件写 1 时可以清除 RXBR 位。

27.5.6 CEC 中断使能寄存器(CEC_IER) *

地址偏移 : 0x14
复位值 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	TXACKIE	TXERRIE	TXUDRIE	TXENDIE	TXBRIE	ARBLSTIE	RXACKIE	LBPEIE	SBPEIE	BREIE	RXOVRIE	RXENDIE	RXBRIE
			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:13 保留，必须保持复位时的值。
位 12 TXACKIE： Tx 缺少确认错误中断使能
TXACKEIE 位由软件设置和清除。
0: TXACKE 中断禁止
1: TXACKE 中断使能
位 11 TXERRIE： Tx 错误中断使能
TXERRIE 位由软件设置和清除。
0: TXERR 中断禁止
1: TXERR 中断使能
位 10 TXUDRIE： Tx 欠载中断使能
TXUDRIE 位由软件设置和清除。
0: TXUDR 中断禁止
1: TXUDR 中断使能
位 9 TXENDIE： Tx 消息结束中断使能
TXENDIE 位由软件设置和清除。
0: TXEND 中断禁止
1: TXEND 中断使能
位 8 TXBRIE： Tx 字节请求中断使能
TXBRIE 位由软件设置和清除。
0: TXBR 中断禁止
1: TXBR 中断使能
位 7 ARBLSTIE： 仲裁丢失中断使能
ARBLSTIE 位由软件设置和清除。
0: ARBLST 中断禁止
1: ARBLST 中断使能

位 6 RXACKIE： Rx 缺少确认错误中断使能
RXACKIE 位由软件设置和清除。
0: RXACKE 中断禁止
1: RXACKE 中断使能

位 5 LBPEIE： 长位周期错误中断使能
LBPEIE 位由软件设置和清除。
0: LBPE 中断禁止
1: LBPE 中断使能

位 4 SBPEIE： 短位周期错误中断使能
SBPEIE 位由软件设置和清除。
0: SBPE 中断禁止
1: SBPE 中断使能

位 3 BREIE： 位上升错误中断使能
BREIE 位由软件设置和清除。
0: BRE 中断禁止
1: BRE 中断使能

位 2 RXOVRIE： Rx 缓冲溢出中断使能
RXOVRIE 位由软件设置和清除。
0: RXOVR 中断禁止
1: RXOVR 中断使能

位 1 RXENDIE： 接收结束中断使能
RXENDIE 位由软件设置和清除。
0: RXEND 中断禁止
1: RXEND 中断使能

位 0 RXBRIE： Rx 字节接收中断使能
RXBRIE 位由软件设置和清除。
0: RXBR 中断禁止
1: RXBR 中断使能

警告： (*) 强制要求只能在 *CECEN=0* 的时候改写 *CEC_IER*。

27.5.7 HDMI-CEC 寄存器镜像

下表对应 HDMI-CEC 寄存器

表 112. HDMI-CEC 寄存器镜像和复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**CEC_CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXEOM	TXSOM	CECEN
	Reset value																														0	0	0
0x04	**CEC_CFGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LBPEGEN	BREGEN	BRESTP	RXTOL	SFT[2:0]			LSTN	OAR[3:0]			
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x08	**CEC_TXDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXD[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x0C	**CEC_RXDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXD[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x10	**CEC_ISR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXACKE	TXERR	TXUDR	TXEND	TXBR	ARBLST	RXACKE	LBPE	SBPE	BRE	RXOVR	RXEND	RXBR
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	**CEC_IER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXACKIE	TXERRIE	TXUDRIE	TXENDIE	TXBRIE	ARBLSTIE	RXACKIE	LBPEIE	SBPEIE	BREIE	RXOVRIE	RXENDIE	RXBRIE
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0

寄存器边界地址请参见 35 页的章节 2.2.2

Index

28 修订历史

表 113. 文档的修订历史

日期	版本	变化
28-Nov-2011	1	初稿发布