基于 BlueNRG-LP ARM Cortex-M0+

引言

本参考手册提供关于如何使用 BlueNRG-LP 微控制器存储器和外设的完整信息。

BlueNRG-LP 是一款功能强大的超低功耗 2.4 GHz 射频收发器，其 Cortex®-M0+ 微控制器的工作频率可达 64 MHz。

BlueNRG-LP 适用于实现符合 Bluetooth®低功耗 SIG 规范的应用。

1 文档约定

1.1 寄存器相关缩写词列表

寄存器说明中使用以下缩写词：

表 1. 寄存器相关缩写词列表

读/写（rw 或 R/W）	软件可以读写这些位
只读（r 或 R）	软件只能读取这些位
只写（w 或 W）	软件只能写入该位。读取该位时将返回复位值
一次性读/写（RWOnce）	软件可读取这些位但只允许写入一次
读/清零（rc_w1 或 RWC1）	软件可以读取该位，也可以通过写入 1 将该位清零。写入“0”对该位的值无影响
读/清零（rc_w0 或 RWC0）	软件可以读取该位，也可以通过写入 0 将该位清零。写入“1”对该位的值无影响
读取/通过读取清零（rc_r 或 RC）	软件可以读取该位。读取该位时，将自动清零。写入“0”对该位的值无影响
读/置位（rs 或 RWS1）	软件可以读取该位，也可将其置 1。写入“0”对该位的值无影响
只读写入触发器（rt_w 或 RWH）	软件可以读取该位。写入“0”或“1”时，将触发事件，但不会影响该位的值
切换（t 或 RWT1）	软件只能通过写入“1”来切换该位。写入“0”无影响
保留（Res.）	保留位，必须保持复位值

1.2 术语表

本节给出了本文档中使用的缩略语的简明定义：

SoC 集成了 SWD 调试端口（SWD-DP），该端口提供基于串行线调试（SWD）协议的 2 引脚（时钟和数据）接口。

- 字：32 位数据/指令
- 半字：16 位数据/指令
- 字节：8 位数据
- 双字：64 位数据
- AHB：高级高性能总线
- APB：高级外围总线
- CPU：请参见 Cortex®-M0+ 内核

1.3 缩略语

表 2. 缩略语

缩略语	说明
ADC	模数转换器
AES	高级加密标准硬件加速器
AHB	高级高性能总线
APB	高级外围总线
ART	自适应实时存储加速器
BLE	低功耗蓝牙（蓝牙标准）
BOR	欠压复位
BPU	断点单元（ARM 调试元件）
COMP	比较器
CPU	中央处理器单元
CRC	循环冗余校验
CTI	交叉触发接口（ARM 调试元件）
DBG	调试
DMA	直接存储器访问
DMAMUX	直接存储器访问路由器
DWT	数据观察点和跟踪（ARM 调试元件）
GPIO	通用输入输出
HSE	高速外部时钟振荡器
HSI	高速内部时钟振荡器
HW	硬件
I2C	内部集成电路（通信标准）
I2S	内部集成（通信标准）
IRQ	中断请求
ITM	仪表跟踪宏单元（ARM 调试元件）
IWDG	独立看门狗
JTAG	联合测试接入组（测试接口标准）
LCD	液晶显示器
LDO	低压差
LP	低功耗
LPUART	低功耗通用异步收发器（通信标准）
LSB	最低有效字节
LSE	低速外部时钟振荡器
LSI	低速内部时钟振荡器
MCU	微控制器单元
MPU	存储器保护单元
MR_BLE	射频子系统
MSB	最高有效字节
NVIC	嵌套向量中断控制器
OBL	选项字节加载

缩略语	说明
OSC	振荡器
OTP	一次性可编程
PDR	下电复位
PDM	脉冲密度调制
PKA	公钥加速器
PLL	锁相环
POR	上电复位
PVD	可编程电压检测器
PVM	外围电压监测
PWR	电源控制器
RC	电阻电容振荡器
RCC	复位和时钟控制器
RF	射频
RNG	真随机数发生器
ROM	只读存储器
RTC	实时时钟
SAI	串行音频接口
SMPS	开关模式电源
SPI	串行外设接口（通信标准）
SRAM	静态随机存取存储器
SW	软件
SWD	单线调试
SWJ-DP	单线联合测试接入组 - 调试端口（ARM 调试元件）
SYSCFG	系统配置
TIM	定时器
USART	通用同步异步收发器（通信标准）
VCO	压控振荡器
VREF	电压参考
WFI	等待指令（ARM 指令进入低功耗模式）
WKUP	唤醒
WRP	写保护
WWDG	窗口看门狗
XIP	本地执行

注意：　如需MR_BLE/射频子系统的相关信息，请参见“BlueNRG-LP 无线电控制器”参考手册（RM0480）。

2 系统和存储器概述

2.1 系统架构

主系统由 32 位多层 AHB 总线矩阵构成，可实现以下部分的互连：

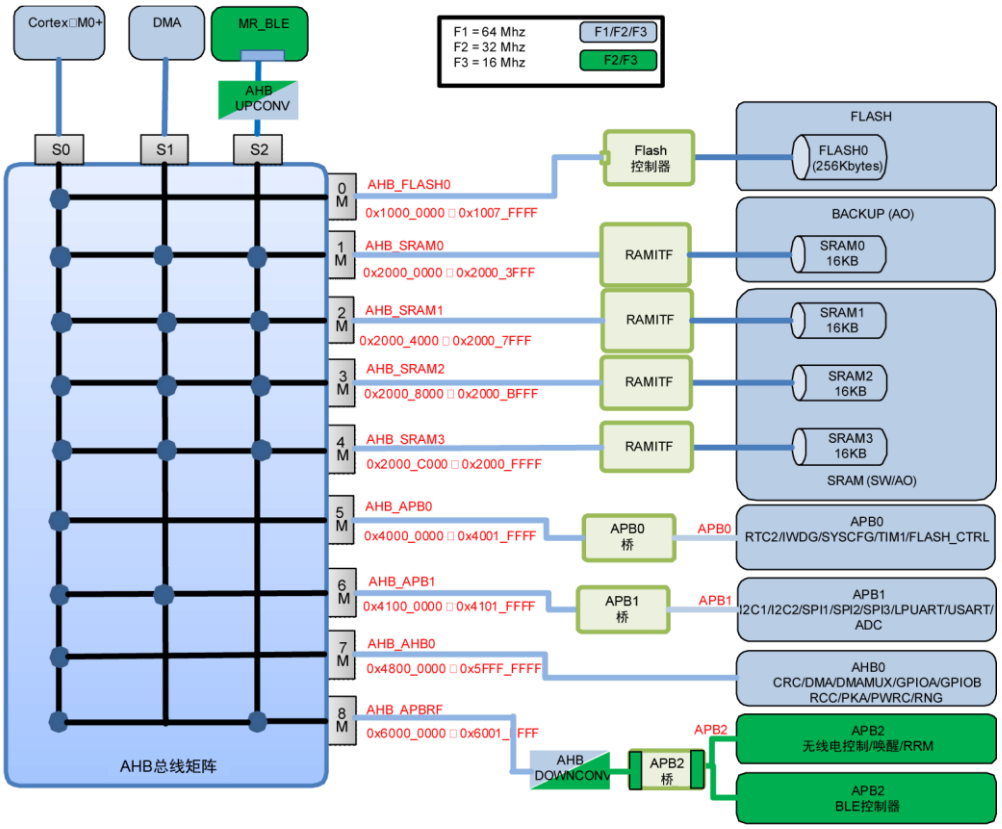
- 3 个主器件：
  - CPU（Cortex®-M0+）内核 S 总线
  - DMA1
  - 射频系统
- 9 个从器件：
  - CPU（Cortex®-M0+）S 总线上的内部 Flash 存储器
  - 内部 SRAM0（16 kB）
  - 内部 SRAM1（16 kB）
  - 内部 SRAM2（16 kB）
  - 内部 SRAM3（16 kB）
  - APB0 外设（通过一个 AHB-to-APB 桥）
  - APB1 外设（通过一个 AHB-to-APB 桥）
  - AHB0 外设
  - AHBRF，包括 AHB-to-APB 桥和无线电外设（连接到 APB2）

借助总线矩阵，可以实现主控总线到被控总线的访问，这样即使在多个高速外设同时运行期间，系统也可以实现并发访问和高效运行。此架构如“图 1. BlueNRG-LP 系统架构”所示。

系统包含 Cortex®-M0+“无线电协议和应用”处理器及其射频子系统。有一个 Flash 存储器会被 CPU 用于蓝牙协议和应用管理。

外设位于不同的系统总线（用于射频系统的 AHB、APB0、APB1 和 APB2）上。有 4 个 SRAM 块，1 个 SRAM0 始终通电，3 个 SRAM（SRAM1、SRAM2 和 SRAM3）可编程为始终通电或可开关。如需更多信息，请参见第 5.6.2 节“控制寄存器 2（PWRC_CR2）”。

图 1. BlueNRG-LP 系统架构



2.1.1 S0：CPU（Cortex®-M0+）S 总线

该总线将 CPU 内核的系统总线连接到总线矩阵。内核使用该总线获取指令，进行字面值加载和调试访问，以及访问位于外设或 SRAM 区中的数据。该总线的访问目标是所有可能的外设（内部 Flash 和 SRAM 存储器，AHB0、APB0、APB1 和 APB2 外设）。

2.1.2 S1：DMA 总线

该总线将 DMA 的 AHB 主接口连接到总线矩阵。该总线的访问目标是 4 块 SRAM 和 APB1 外设。

2.1.3 S2：射频系统总线

该总线将射频系统的 AHB 主接口连接到总线矩阵。该总线的访问目标是 4 块 SRAM 和 APB2 外设（MR_BLE/射频子系统 IP 的内部 APB 模块）。

2.1.4 总线矩阵

总线矩阵管理主设备（CPU、DMA、MR_BLE）之间的访问仲裁。仲裁使用循环轮询（Round Robin）算法。总线矩阵由 3 个主器件（CPU、DMA1 总线和射频系统总线）和 9 个从器件（FLASH、SRAM0、SRAM1、SRAM2、SRAM3、APB0、APB1、AHB0 和 AHBRF）组成。

AHB/APB 桥

AHB-to-APB0 桥和 AHB-to-ABP1 桥在 AHB 和两个 APB 总线之间提供全同步连接。

AHB-to-APB2 桥在 AHB 和 APB 总线之间提供同步连接。在 AHB/APB 路径上添加了两个模块，用于管理 MR_BLE 频率和系统频率之间、潜在的预分频关系：

- AHB 上变频器用于 MR_BLE AHB 主设备访问 SRAM
- AHB 下变频器则用于 CPU AHB 主设备访问 MR_BLE APB 寄存器。

请参见第 2.2.2 节“存储器映射和寄存器边界地址”了解连接到该桥的外设的地址映射。

每次芯片复位后，所有外设时钟都被关闭（SRAM 和 Flash 存储器接口除外）。在使用外设前，用户必须在 RCC_AHBxENR 和 RCC_APBxENR 寄存器中使能其时钟。

注意： *对 APB 寄存器执行 16 位或 8 位访问时，该访问将转换为 32 位访问：总线桥将 16 位或 8 位数据复制后提供给 32 位向量。*

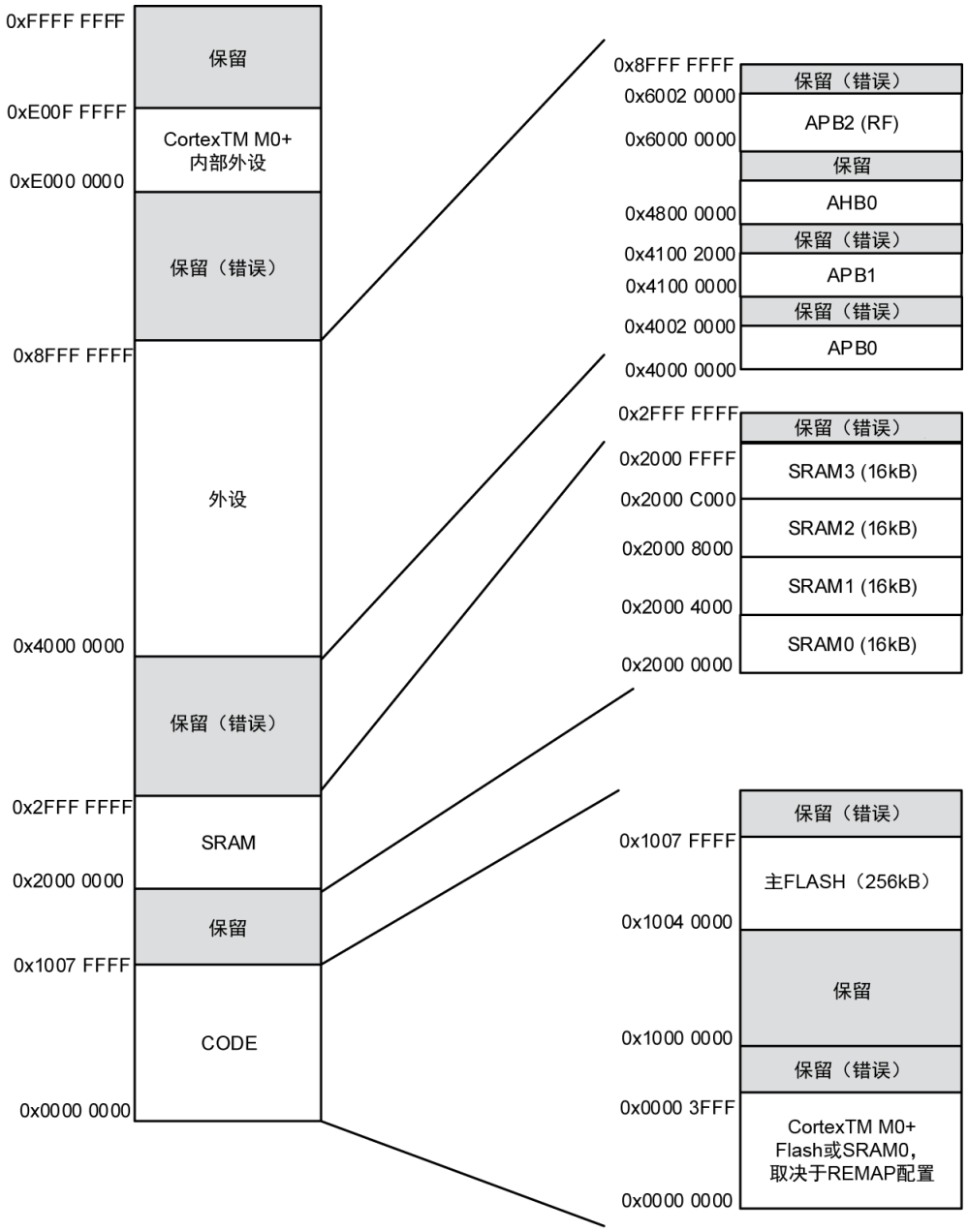
2.2 存储器组织结构

2.2.1 引言

程序存储器、数据存储器和寄存器排列在同一个线性（即地址连续）的 4 GB 地址空间内。

这些字节按小端序格式在存储器中编码。字中编号最低的字节被视为该字的最低有效字节，而编号最高的字节被视为最高有效字节。

图 2. 存储器映射



未分配给片上存储器和外设的所有存储区域均视为“保留区”。关于可用存储区和寄存器区的具体映射，请参见第 2.2.2 节“存储器映射和寄存器边界地址”和各个外设寄存器映射章节。

2.2.2 存储器映射和寄存器边界地址

表 3. BlueNRG-LP 存储器映射和外设寄存器边界地址给出了器件中可用外设的边界地址。

表 3. BlueNRG-LP 存储器映射和外设寄存器边界地址

总线	边界地址	实际大小（字节）	外设	外设寄存器映射
Misc	0xE010 0800 - 0xFFFF FFFF	255 MB	保留	
	0xE000 0000 - 0xE00F FFFF	1 MB	私有外设总线	Cortex-M0+ 寄存器（内部控制器、SysTick 等）
APB2	0x6000 1800 - 0x6000 1BFF	1 kB	WAKEUP	MR_BLE 的唤醒寄存器
	0x6000 1500 - 0x6000 17FF	1 kB	无线电寄存器	通过 APB 直接访问无线电寄存器
	0x6000 1400 - 0x6000 14FF		RRM	MR_BLE 的 RRM 寄存器
	0x6000 1000 - 0x6000 13FF	1 kB	RADIO_CTRL	MR_BLE 的无线电控制器寄存器
	0x6000 0000 - 0x6000 00FF	256 B	BLE	MR_BLE 的 BLE 链路层寄存器
AHB0	0x4880 0000 - 0x4880 03FF	1 kB	DMAMUX	参见表 28. DMAMUX 寄存器映射和复位值
	0x4870 0000 - 0x4870 00FF	256 B	从 DMA	参见表 25. DMA 寄存器映射和复位值
	0x4860 0000 - 0x4860 03FF	4 kB	RNG	参见表 37. RNG 寄存器列表
	0x4850 0000 - 0x485F FFFF	156 B	PWRC	参见表 10. PWRC 寄存器映射
	0x4840 0000 - 0x484F FFFF	156 B	RCC	参见第 6.7 节“RCC 寄存器映射”
	0x4830 0400 - 0x4830 07FF	1 kB	PKA RAM	参见表 41. PKA 寄存器列表
	0x4830 0000 - 0x4830 03FF	1 kB	从 PKA	
	0x4820 0000 - 0x4820 03FF	1 kB	CRC	参见表 47. CRC 寄存器映射和复位值
	0x4810 0000 - 0x4810 03FF	1 kB	GPIOB	参见第 7.5 节“GPIO 寄存器映射”
	0x4800 0000 - 0x4800 03FF	1 kB	GPIOA	参见第 7.5 节“GPIO 寄存器映射”
APB1	0x4100 7000 - 0x4100 73FF	1 kB	SPI3	参见第 22.10 节“SPI/I2S 寄存器映射”
	0x4100 6000 - 0x4100 60FF	256 B	ADC	参见第 12.7 节“ADC 寄存器映射”
	0x4100 5000 - 0x4100 53FF	1 kB	LPUART	参见第 21.5 节“LPUART 寄存器”
	0x4100 4000 - 0x4100 43FF	1 kB	USART	参见第 20.7 节“USART 寄存器”
	0x4100 3000 - 0x4100 33FF	1 kB	SPI2	参见第 22.10 节“SPI/I2S 寄存器映射”
	0x4100 2000 - 0x4100 23FF	1 kB	SPI1	参见第 22.10 节“SPI/I2S 寄存器映射”
	0x4100 1000 - 0x4100 13FF	1 kB	I2C2	参见第 19.7 节“I2C 寄存器映射”
	0x4100 0000 - 0x4100 03FF	1 kB	I2C1	参见第 19.7 节“I2C 寄存器映射”
APB0	0x4000 4000 - 0x4000 43FF	1 kB	RTC	参见第 17.7 节“RTC 寄存器映射”
	0x4000 3000 - 0x4000 33FF	1 kB	IWDG	参见第 18.5 节“IWDG 寄存器映射”
	0x4000 2000 - 0x4000 23FF	1 kB	TIM1	参见第 16.5 节“TIM1 寄存器映射”
	0x4000 1000 - 0x4000 1FFF	4 kB	FLASH_CTRL	参见第 9.3 节“Flash 控制器寄存器映射”
	0x4000 0000 - 0x4000 003F	64 B	SYSTEM_CTRL	参见第 8.3 节“系统控制器寄存器映射”
SRAM	0x2000 C000 - 0x2000 FFFF	16 kB	SRAM3	
	0x2000 8000 - 0x2000 BFFF	16 kB	SRAM2	
	0x2000 4000 - 0x2000 7FFF	16 kB	SRAM1	
	0x2000 0000 - 0x2000 3FFF	16 kB	SRAM0	
Flash	0x1004 0000 - 0x1007 FFFF	256 kB	主 Flash	
OTP	0x1000 1800 - 0x1000 1BFF	1 kB	OTP 区域	
Boot(1)	0x0000 0000 - 0x0000 3FFF	16 kB	CPU 启动区	Flash 或 SRAM0 的镜像

1. *此区域为镜像区。根据 Flash 控制器 CONFIG 寄存器中的 REMAP 和 PREMAP 位，CPU 访问被重定向至其他存储器映射。参见表 4.“地址重映射取决于 REMAP 位”了解重映射详情。*

表 4. 地址重映射取决于 REMAP 位

REMAP	内存映射
0	主 Flash
1	SRAM0

2.3 ARM® Cortex®-M0+

ARM® Cortex®-M0+处理器引脚数少、功耗低。这款处理器是为了满足微控制器应用需求而开发的低成本平台，它具有较少的引脚数和低功耗，同时在计算性能和对中断响应方面表现出色。

嵌入式 Cortex-M0+中嵌入了：

- 4 个断点
- 2 个观察点
- 1 个 MPU（存储器保护单元），提供了 8 个统一的保护区
- SysTick，仅以系统时钟运行（不支持外部时钟选项）

2.3.1 CPU 存储器重映射

在 CPU 启动后，应用软件可以修改地址 0x0000 0000 处的存储器映射。通过 Flash 控制器中 REMAP 位的编程来执行此修改（参见第 9.4.2 节“配置寄存器（CONFIG）”）。

可重映射以下存储器：

- 主 Flash 存储器
- SRAM0 内存

嵌入式自举程序

ST 提供了一个在每次 CPU 重启后执行的自举程序。该自举程序有单独的文档说明。

注意： *BlueNRG-LP 器件在 POR（上电复位）时锁存了 PA8/PA9/PA10/PA11 引脚的值。PWRC_SR2 寄存器中提供了此信息（参见第 5.6.6 节“状态寄存器 2（PWRC_SR2）”）。系统允许四个特定 I/O 引脚（PA8、PA9、PA10 和 PA11）中的一个，其输入值在上电复位时锁存，作为自举程序是执行正常引导还是在串行接口上引导的指示。*

2.3.2 中断

由 Cortex-M0+嵌套向量中断控制器（NVIC）处理中断。NVIC 控制特定的 Cortex-M0 中断（地址 0x00 至 0x3C）和 32 用户中断（地址 0x40 至 0xBC）。在 BlueNRG-LP 器件中，用户中断连接到不同外设（GPIO、Flash 控制器、定时器、UART 等）的中断信号。可使用 ISER、ICER、ISPR 和 ICOR 寄存器控制这些中断。

向量表

复位时，Cortex-M0+向量表固定在地址 0x0000_0000。软件可能通过 Cortex-M0+寄存器区域中的向量表偏移寄存器（VTOR），将向量表地址重新定位到范围 0x0000_0000 至 0xFFFF_FF80（256 字节的倍数）以内的其他存储器位置。

BlueNRG-LP 器件向量表的中断映射如“表 5.中断向量”所述。

表 5. 中断向量

位置	优先级	优先级类型	缩略语	说明	偏移地址
				初始主 SP	0x0000_0000
	-3	固定	复位	Reset_Handler	0x0000_0004
	-2	固定	NmiSR	NMI_Handler	0x0000_0008
	-1	固定	FaultISR	HardFault_handler	0x0000_000C
		RESERVED	RESERVED	RESERVED	0x0000_000C - 0x0000_0038
	6	可设置	SysTick	系统节拍定时器	0x0000_003C
0	Init 0	可设置	NVM	非易失性存储器（Flash）控制器	0x0000_0040
1	Init 0	可设置	RCC	复位和时钟控制器	0x0000_0044
2	Init 0	可设置	BATTERY	PVD	0x0000_0048
3	Init 0	可设置	I2C1	I2C1 中断	0x0000_004C
4	Init 0	可设置	I2C2	I2C2 中断	0x0000_0050
5	Init 0	可设置	SPI1	SPI1 中断	0x0000_0054
6	Init 0	可设置	SPI2	SPI2 中断	0x0000_0058
7	Init 0	可设置	SPI3	SPI3 中断	0x0000_005C
8	Init 0	可设置	USART	USART 中断	0x0000_0060
9	Init 0	可设置	LPUART	LPUART 中断	0x0000_0064
10	Init 0	可设置	TIM1	TIM1 中断	0x0000_0068
11	Init 0	可设置	RTC	RTC 中断	0x0000_006C
12	Init 0	可设置	RESERVED	RESERVED	0x0000_0070
12	Init 0	可设置	ADC	ADC 中断	0x0000_0070
13	Init 0	可设置	PKA	PKA 中断	0x0000_0074
14	Init 0	可设置	-	-	0x0000_0078
15	Init 0	可设置	GPIOA	GPIOA 中断	0x0000_007C
16	Init 0	可设置	GPIOB	GPIOB 中断	0x0000_0080
17	Init 0	可设置	DMA	DMA 中断	0x0000_0084
18	Init 0	可设置	BLE_TXRX (1)	传输中断的 BLE 结束	0x0000_0088
19	Init 0	可设置	-(1)	-	0x0000_008C
20	Init 0	可设置	-(1)	-	0x0000_0090
21	Init 0	可设置	RADIO_CTRL(1)	无线电控制慢速时钟测量中断	0x0000_0094
22	Init 0	可设置	MR_BLE (1)	RRM 和无线电 FSM 中断	0x0000_0098
23	Init 0	可设置	CPU_WKUP(1)	CPU 唤醒中断	0x0000_009C
24	Init 0	可设置	BLE_WKUP (1)	BLE 唤醒中断	0x0000_00A0

位置	优先级	优先级类型	缩略语	说明	偏移地址
25	Init 0	可设置	BLE_SEQ	BLE RX/TX 序列中断	0x0000_00A4
26 - 31	Init 0	可设置	-	RESERVED	0x0000_00A8 - 0x0000_00BC

1. 为了将中断清零，软件层面有一些需要特别注意的事项：参见“关于系统时钟快于 MR_BLE 时钟时中断清零的警告”。

中断激活序列

安全地激活外设中断需要以下步骤：

- 确保在外设端中断被禁用并清零（防止接收到先前事件导致的中断）
- 在 Cortex-M0+端使用 NVIC ICPR 寄存器将该中断的待处理请求清零
- 使用 NVIC IPR0-IPR7 寄存器设置优先级
- 在 Cortex-M0+端使用 NVIC ISER 寄存器激活
- 在外设端激活

请注意，大多数外设在处理中断服务请求时需要清除外设端的中断请求，以防止为同一事件触发连续中断。

如需关于 Cortex-M0+异常模型的更多信息，参见《ARMv6-M 架构参考手册》§B1.5。

如需关于 Cortex-M0+ NVIC 行为和寄存器的更多信息，参见《ARMv6-M 架构参考手册》§B3.4。

3 AHB 上/下变频器

BlueNRG-LP 器件可支持 1 MHz 至 64 MHz 的多个系统时钟频率。

当系统（CPU、DMA 和存储器）可能因为应用而需要更高性能时，MR_BLE/射频系统 IP 无需高于 32 MHz 即可实现无线电传输的处理。

与系统总线矩阵频率相比，为避免无用的过量消耗，添加了 AHB 上/下变频器模块，以便在 MR_BLE（链接到 AHBRF/APB2 桥）的 AHB 和 APB 总线上引入可调节分频器（1、2 或 4 分频）。该模块可将器件的 MR_BLE IP 的系统时钟进行 1、2 或 4 分频。

当系统和 MR_BLE 共享相同频率时，AHB 上/下变频器模块只将 AHB 信号从一个时钟域传输到另一个。

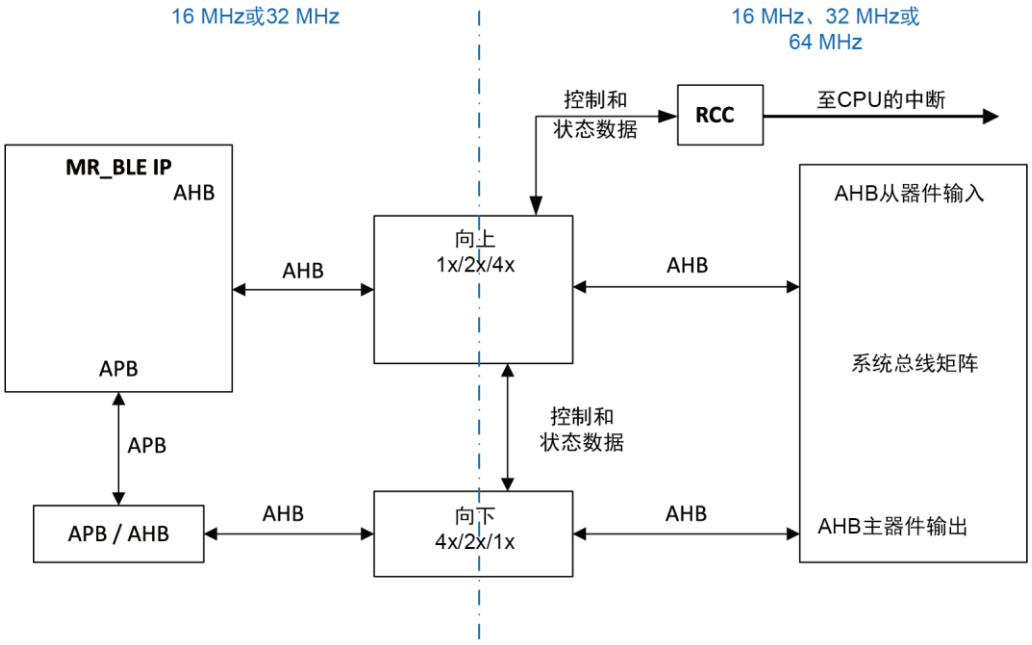
注意： *当启用射频时，系统时钟的频率必须是 16/32/64 MHz，并且总是等于或大于 MR_BLE 时钟（无其他频率）。*

3.1 AHB 上/下变频器说明

AHB 上/下变频器的作用是允许 BlueNRG-LP 器件支持快速系统时钟（最高 64 MHz）。AHBUPCONV 模块管理：

- 通过安全地更新系统频率与 MR_BLE IP 频率的分频比（1、2 或 4），进行合适的实时（上至下和下至上）频率切换。
- 与 MR_BLE 之间的 AHB 和 APB 数据传输，MR_BLE 以与系统其余部分相同或其一半或四分之一的频率运行，系统其余部分（AHB 和 APB）的运行频率可高达 64 MHz。

图 3. AHB 上/下变频器



使用状态机完成数据传输与时钟域以及可能的时钟切换请求的管理：

- 一个用于 AHB 主器件上变频器
- 一个用于 AHB 从器件下变频器

当需要 CPU/系统时钟频率切换（激活或停用系统与 MR_BLE 之间的 2 或 4 分频）时，用户必须在 RCC_CSCMDR.REQUEST 位请求新的系统时钟目标频率（参见第 6.6.5 节“时钟切换指令寄存器（RCC_CSCMDR）”了解详情）。在接收到要应用的新的分频比（来自 RCC）时，AHBUPCONV 模块：

- 通知 AHBDOWNCONV 模块（管理从 CPU 到 MR_BLE APB 寄存器的 APB 传输）
- 检查 MR_BLE 与 CPU 之间的 AHB 总线上的流量：
  - 如果目前没有进行传输，它在总线上使用 HREADY 信号保持时钟频率切换期间可能发生的任何潜在事务

注意： *为了遵守 AHB Lite 协议，HREADY 信号仅在新的传输的地址阶段之后变为低电平，新的传输阶段数据被保存在变频器内部。*

如果正在进行 AHB 传输，它将等待当前 AHB 传输结束，然后如上文所述保持 AHB 流量。

- 同时，AHBDOWNCONV 模块执行相同操作，以便保持数据路径的从路径上的任何新的传输（如 CPU 访问 MR_BLE APB 寄存器）。
- 在 AHBUPCONV 模块暂停其 AHB 线路并获得来自 AHBDOWNCONV 的确认后，其他 AHB2APB 数据路径也被安全暂停；这将便于 RCC 模块将系统时钟频率更改为请求的频率。
- 当新的系统时钟频率稳定时，RCC 将更改完毕的消息通知 AHBUPCONV。然后：
  - AHBUPCONV 释放 AHB 传输
  - AHBUPCONV 通知 AHBDOWNCONV，它还可以释放 AHB/APB 传输。

4 I/O 工作模式

BlueNRG-LP 器件提供至多 32 个可编程 I/O（取决于封装）。

每个 I/O 可编程为数种模式：

- 输入模式
- 通用输出模式
- 复用功能
- 模拟模式（如可用）

BlueNRG-LP 器件支持 8 种复用模式，称为 AFx（x = 0 至 7）。这些复用模式的每个 I/O 的配置详见“表 7.GPIO 复用选项 AF3 - AF4”和“表 8. I/O 模拟特性映射”。

注意：当使用外部慢速时钟 XO（PB12 和 PB13 I/O 上的 LSE）时，在 AF7 中的 PB14/PB15 I/O 上完成的 USART 传输会对慢速时钟产生一些干扰。

注意： *表 7. GPIO 复用选项 AF3 - AF4 和表 8. I/O 模拟特性映射中列出的 I/O 特性，仅在相关 I/O 被设定为复用功能时可用。*

请参见第 7 节“通用 I/O（GPIO）”了解关于所有配置的更多信息。

BlueNRG-LP 器件中可用 I/O 的数量取决于封装：

- QFN32 中 20 个 I/O
- CSP49 中 30 个 I/O
- QFN48 中 32 个 I/O

类型缩略语对应的含义：

- I：输入
- O：输出
- I/O：输入输出
- OD：开漏

请注意，在“表 6. GPIO 复用选项 AF0 - AF2”中，绿色单元格对应的是 QFN32 封装的可用 I/O。蓝色单元格对应的是 QFN32 中 CSP49 封装的额外可用 I/O。粉红色单元格对应的是 CSP49 中 QFN48 封装的额外可用 I/O。粗体的引脚名称表示在 DEEPSTOP 期间，I/O 能够唤醒器件且能够处于输入或输出状态。

表 6. GPIO 复用选项 AF0 - AF2

引脚名称	AF0 模式		AF1 模式		AF2 模式	
	类型	信号	类型	信号	类型	信号
端口 A						
PA0(1)	I/OD	I2C1_SCL	I	USART_CTS	O	SPI2_MCK/ I2S2_MCK
PA1(1)	I/OD	I2C1_SDA	I/O	SPI2_MISO/I2S2_MISO	I/OD	USART_TX
PA2	I/O	TMS-SWDIO	I/O	USART_CK	I	TIM_BKIN
PA3	I	TCK-SWCLK	O	USART_RTS_DE	I	TIM_BKIN2
PA4	O	LCO	I/O	SPI2_NSS/I2S2_WS	-	-
PA5	O	MCO	I/O	SPI2_SCK/ I2S2_SCK	-	-
PA6	I	LPUART_CTS	I/O	SPI2_MOSI/I2S2_SD	-	-
PA7	O	LPUART_RTS_DE	I/O	SPI2_MISO/I2S2_MISO	-	-
PA8	I/O	USART_RX	I/O	SPI1_MOSI	O	RX_SEQUENCE
PA9	I/O	USART_TX	I/O	SPI1_SCK	O	RTC_OUT
PA10	O	LCO	I/O	SPI1_MISO	O	TX_SEQUENCE
PA11	O	MCO	I/O	SPI1_NSS	O	RX_SEQUENCE
PA12 (1)	I/O	I2C1_SMBA	I/O	TMS-SWDIO	I/O	SPI1_NSS
PA13 (2)	I/O	I2C2_SCL	I	TCK-SWCLK	I/O	SPI1_SCK
PA14 (1)	I/O	I2C2_SDA	-	-	I/O	SPI1_MISO
PA15 (1)	I/O	I2C2_SMBA	-	-	I/O	SPI1_MOSI
端口 B						
PB0(1)	I/O	USART_RX	O	LPUART_RTS_DE	-	-
PB1 (1)	I/O	SPI1_NSS	O	PDM_CLK	-	-
PB2(1)	O	USART_RTS_DE	I	PDM_DATA	-	-
PB3(1)	I	USART_CTS	I/O	LPUART_TX	-	-
PB4(1)	I/O	LPUART_TX	I/O	SPI2_MISO /I2S2_MISO	-	-
PB5(1)	I/O	LPUART_RX	I/O	SPI2_MOSI /I2S2_SD	-	-
PB6(1)	OD	I2C2_SCL	I/O	SPI2_NSS /I2S2_WS	-	-
PB7(1)	OD	I2C2_SDA	I/O	SPI2_SCK /IS2S_SCK	-	-
PB8	I/O	USART_CK	I/O	LPUART_RX	-	-
PB9	I/O	USART_TX	I	LPUART_CTS	O	SPI2_MCK/ I2S2_MCK
PB10	I/O	SPI1_NSS	I/O	SPI2_SCK /I2S2_SCK	I/O	I2C1_SDA
PB11	I/O	SPI1_SCK	I/O	SPI2_NSS /I2S2_WS	I/O	I2C1_SCL

引脚名称	AF0 模式		AF1 模式		AF2 模式	
	类型	信号	类型	信号	类型	信号
PB12[1][2]	I/O	SPI1_SCK	O	LCO	I	PDM_DATA
PB13[1][3]	I/O	SPI1_MISO	I/O	I2C2_SCL	O	PDM_CLK
PB14[1]	I/O	SPI1_MOSI	I/O	I2C2_SDA	I	TIM1_ETR
PB15[1]	I/O	I2C1_SMBA	O	TX_SEQUENCE	O	MCO

1. *在 QFN32 封装中，此 I/O 能够配置为模拟（mixedI/O）。*
2. *此 I/O 与 OSC32_OUT（低速时钟振荡器引脚）共享，在 CSP49 的 QFN48 封装中可用。*
3. *此 I/O 与 OSC32_IN（低速时钟振荡器引脚）共享。*

应考虑以下关于表 7. GPIO 复用选项 AF3 - AF4。绿色单元格对应的是 QFN32 封装的可用 I/O。蓝色单元格对应的是 QFN32 中 CSP49 封装的额外可用 I/O。粉红色单元格对应的是 CSP49 中 QFN48 封装的额外可用 I/O。粗体的引脚名称表示在 DEEPSTOP 期间，I/O 能够唤醒器件且能够处于输入或输出状态。

表 7. GPIO 复用选项 AF3 - AF4

引脚名称	AF3 模式		AF4 模式	
	类型	信号	类型	信号
端口 A				
PA0(1)	I		I/O	TIM1_CH3
PA1(1)	O		I/O	TIM1_CH4
PA2	O	SPI3_MCK/ I2S3_MCK	O	TIM1_CH5
PA3	I/O	SPI3_SCK/ I2S3_SCK	O	TIM1_CH6
PA4	I/O	LPUART_TX	I/O	TIM1_CH1
PA5	I/O	LPUART_RX	I/O	TIM1_CH2
PA6	I/O	SPI2_NSS/ I2S2_WS	I/O	TIM1_CH1
PA7	I/O	SPI2_SCK/I2S2_SCK	I/O	TIM1_CH2
PA8	I/O	SPI3_MISO/ I2S3_MISO	I/O	TIM1_CH3
PA9	I/O	SPI3_NSS/ I2S3_WS	I/O	TIM1_CH4
PA10	O	SPI3_MCK/ I2S3_MCK	O	TIM1_CH5
PA11	I/O	SPI3_MOSI/ I2S3_SD	O	TIM1_CH6
PA12(2)	I/O	SPI2_MOSI/ I2S2_SD	I/O	TIM1_CH1
PA13 (2)	I/O	SPI2_MISO/ I2S2_MISO	I	TIM1_ETR
PA14 (2)	-	-	I	TIM1_BKIN
PA15 (2)	-	-	I	TIM1_BKIN2
端口 B				
PB0 (2)	I/O	TIM1_CH2N	-	-
PB1(2)	I	TIM1_ETR	-	-
PB2(2)	I/O	TIM1_CH3	-	-
PB3 (2)	I/O	TIM1_CH4	-	-
PB4 (2)	I	PDM_DATA	-	-
PB5(2)	O	PDM_CLK	-	-
PB6(1)	I/O	LPUART_TX	I/O	TIM1_CH1
PB7(1)	I/O	LPUART_RX	I/O	TIM1_CH2
PB8	I/O	TIM1_CH4	I/O	TIM1_CH1N
PB9	I/O	TIM1_CH1N	I/O	TIM1_CH2N
PB10	I/O	TIM1_CH2	I/O	TIM1_CH3N
PB11	I/O	TIM1_CH1	I/O	TIM1_CH4N

引脚名称	AF3 模式		AF4 模式	
	类型	信号	类型	信号
PB12(2)(3)	I	TIM1_BKIN	I/O	TIM1_CH3
PB13(2)(4)	I	TIM1_BKIN2	I/O	TIM1_CH4
PB14 (2)	I/O	TIM1_CH3N	O	TIM1_CH5
PB15 (2)	I/O	TIM1_CH4N	O	TIM1_CH6

1. *此 I/O 能够配置为开漏模式。*
2. *此 I/O 能够配置为模拟（mixedI/O）。*
3. *此 I/O 与 OSC32_OUT（低速时钟振荡器引脚）共享。*
4. *此 I/O 与 OSC32_IN（低速时钟振荡器引脚）共享。*

注意： *AF5、AF6、AF7 复用功能：*

PA2：TMS_SWDIO（AF5，AF7）

PA3：TCK_SWCLK（AF5，AF7）

PB14：USART_RX（AF7）

PB15：USART_TX（AF7）

注意： *BlueNRG-LP 具备 TX_SEQUENCE 和 RX_SEQUENCE 信号，分别用于发送和接收活动的提示。可通过以下复用功能使能一个信号，用于两个引脚上的发送和接收：*

- *TX_SEQUENCE 在 PA10（AF2）或 PB15（AF1）上可用*
- *RX_SEQUENCE 在 PA8（AF2）或 PA11（AF2）上可用*

当无线电处于发送（或接收）模式时，信号为高电平，否则为低电平。

该信号可用于控制外部天线切换，并支持与其他无线技术共存。

注意： *关于 BlueNRG-LP 器件中的 ADC 模块：*

- *PA12、PA13、PA14、PA15、PB0、PB1、PB2 和 PB3 上有 8 个 ADC 通道可用*
- *PB4 和 PB5 上有 2 个模拟麦克风引脚可用。*

这些引脚上的 ADC 特性在相关引脚被配置为模拟模式（参见第 7.3.1 节“通用 I/O（GPIO）”了解更多信息）时可用

下表所示为与模拟配置（对于支持模拟模式的引脚）相关的映射。

在表 8. I/O 模拟特性映射，绿色单元格对应的是 QFN32 封装的可用 I/O，蓝色单元格对应的是 CSP49 封装的额外可用 I/O，粉红色单元格对应的是 QFN48 封装的额外可用 I/O。

注意： *对复用功能编程以设置引脚的输出值：I2C_SDA、I2C_SCL、I2C_SMBA 置为 1，SPI_SCK、SPI_NSS、SPI_MISO、SPI_MOSI、SPI_MCKO 置为 0，UART_RX 置为 0 且 UART_TX 置为 1。*

表 8. I/O 模拟特性映射

引脚名称	模拟特性	引脚名称	模拟特性	并行模拟特性
端口 A		端口 B		
PA0	N/A(1)	**PB0**	ADC_VINM1	N/A(1)
PA1	N/A(1)	**PB1**	ADC_VINP1	N/A(1)
PA2	N/A(1)	**PB2**	ADC_VINM0	PGA_CAP0(2)
PA3	N/A(1)	**PB3**	ADC_VINP0	PGA_CAP1 (2)
PA4	N/A(1)	**PB4**	PGA_VIN	N/A(1)
PA5	N/A(1)	**PB5**	PGA_VBIAS_MIC(3)	N/A(1)
PA6	N/A(1)	**PB6**	N/A(1)	N/A(1)
PA7	N/A(1)	**PB7**	N/A(1)	N/A(1)
PA8	N/A(1)	**PB8**	N/A(1)	N/A(1)
PA9	N/A(1)	**PB9**	N/A(1)	N/A(1)
PA10	N/A(1)	**PB10**	N/A(1)	N/A(1)
PA11	N/A(1)	**PB11**	N/A(1)	N/A(1)
PA12	ADC_VINM3	**PB12**	N/A(1)	SXTAL0 (4)
PA13	ADC_VINP3	**PB13**	N/A(1)	SXTALI (4)
PA14	ADC_VINM2	**PB14**	N/A(1)	PVD 输入电压
PA15	ADC_VINP2	**PB15**	N/A(1)	N/A

1. *N/A 表示不适用，因为相关 I/O 不支持模拟选项。*
2. *通过 SYSCFG 模块中的寄存器在 ADC_VINx0 和 PGA_CAPx 之间进行选择。参见第 8.2.11 节“I/O 模拟开关控制寄存器（GPIO_SWA_CTRL）”。*
3. *该引脚不是 5 V 容限。其余所有 31 个引脚均为 5 V 容限。*
4. *通过 RCC 寄存器中 RCC_CR.LSEON 位的置位获得并行模拟特性。然后，硬件强制 PB12 和 PB13 通过 SXTAL0/SXTALI 管理 LSE，无论在相关 GPIO_MODERx 寄存器中选择了哪种模式。*

5 电源控制器（PWRC）

电源控制器模块控制模拟供电模块，并管理器件的启动、活动和低功耗阶段，包括从一种状态到另一种状态的转变。

5.1 特性

电源控制器模块支持以下特性：

- 低功耗模式选择和进入/退出序列
- Flash 存储器电源（开/关）和下电时序
- RAM 区块数据保持控制
- 电源监控：
  - VDDIO 电压上升/下降时 POR/PDR 复位
  - 可编程电压检测器（PVD）用可编程阈值监控 VDDIO 或监控外部模拟输入电压（与内部 VBGP 进行比较）
- 低功耗模式下的 I/O 上/下拉

唤醒 I/O 配置。

5.2 电源域

BlueNRG-LP 器件中嵌入了 3 个电源域：

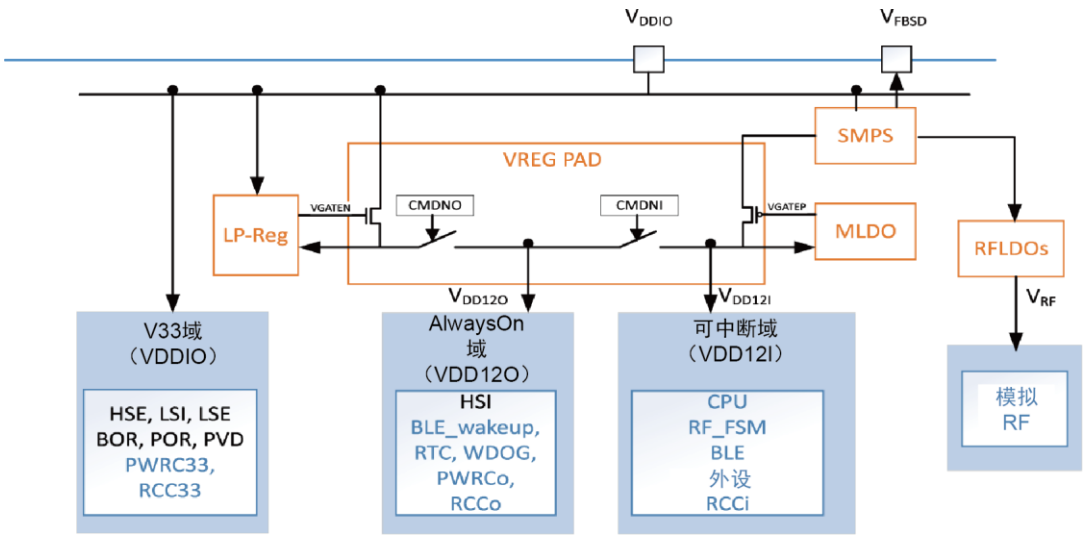
- VDD33 亦即 VDDIO 或 VDD：
  - 电压范围为 1.7 V 至 3.6 V，
  - 它为 I/O 环的一部分、嵌入式稳压器和系统模拟 IP（如电源管理模块和嵌入式振荡器）供电
- VDD12o：
  - 常开数字电源域
  - 在器件活跃阶段，该电源域的供电电压通常为 1.2 V
  - 在低功耗模式（DEEPSTOP）下，该电源域的供电电压通常为 1.0 V
- VDD12i：
  - 可中断数字电源域
  - 在器件活跃阶段，该电源域的供电电压通常为 1.2 V
  - 在低功耗模式（DEEPSTOP）下，该电源域关闭

  通过不同稳压器提供数字电源：
  - 主 LDO（MLDO）：
    - 从 1.4-3.3 V 输入电压提供 1.2 V 电压
    - 在器件活跃时提供 VDD12i 和 VDD12o
    - 在低功耗模式（DEEPSTOP）下被禁用
  - 低功耗 LDO（LPREG）：
    - 在活跃和低功耗阶段皆保持使能
    - 提供 1.0 V 电压
    - 在器件活跃时未连接到数字电源域
    - 在低功耗模式（DEEPSTOP）下连接到 VDD12o 域
  - 专用 LDO（RFLDO），用于向模拟 RF 模块提供 1.2 V 电压

可提供嵌入式 SMPS 同步降压转换器（位于外部电源和 LDO 之间）。

图 4. 电源域总览显示了电源域之间的不同稳压器和连接的总览。

图 4. 电源域总览



5.3 电源电压监控器

BlueNRG-LP 器件中嵌入了多种电源电压监控：

- 上电复位（POR）/下电复位（PDR）/欠压复位（BOR）
- 电源电压检测器（PVD）

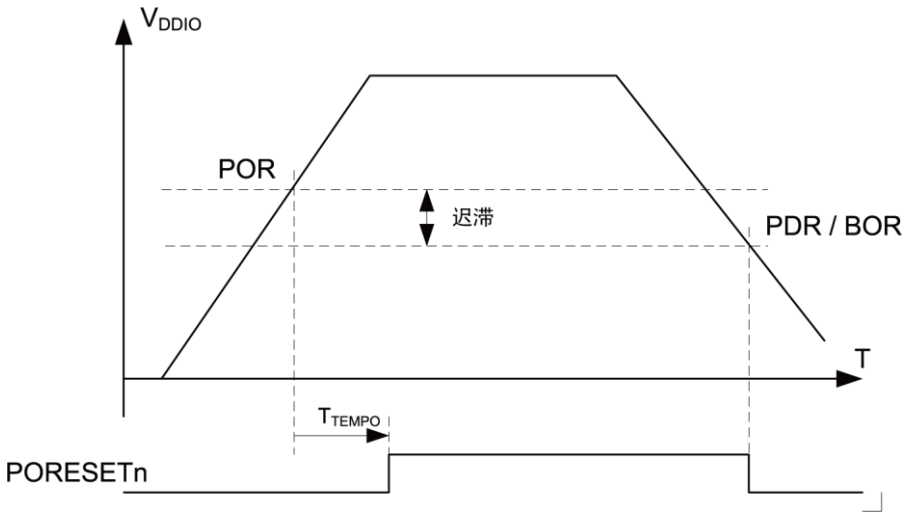
5.3.1 上电复位 POR/下电复位（PDR）/欠压复位（BOR）

该器件具有一个集成的上电复位/下电复位，与欠压复位电路耦合。

上电期间，只要 V_{DDIO} 低于 V_{POR} 阈值（通常为 1.65 V），则器件维持复位模式。

下电期间，PDR 在供电电压（VDD）降至 V_{PDR} 阈值（低于 VPOR 约 20 mV）以下时使器件保持复位。PDR 特性始终使能。

图 5. 上电复位/下电复位波形



典型值如下：

- VPOR：1.65 V

- 迟滞：20 mV（因此，VPDR：1.63 V）
- TTEMPO：250 us

当供电电压（VDD）降至 VPDR 以下时，欠压复位（BOR）生成器件复位。

此特性始终激活，只有在 SHUTDOWN 模式下例外，在该模式下软件可决定是否使能此特性（通过 PWRC_CR1.ENSDNBOR 位）。

5.3.2 电源电压检测（PVD）

PVD 可用于监控：

- VDDIO：
  - 将 VDDIO 与设定阈值（在 2.05 V 和 2.91 V 之间）进行比较
  - 通过 PWRC_CR2.PVDLS[2:0]位字段完成阈值设定
- 外部模拟输入信号：
  - 将外部模拟信号与内部 VBGP（1.2 V）电压进行比较
  - 通过 PWRC_CR2.PVDLS[2:0]位字段选择该特性

PVD 可通过 PWRC_CR2.PVDE 位使能或禁用。

当特性使能且 PVD 实测电压低于比较器时，SYSCFG 模块中的状态标志变为高电平，可生成对 CPU 的中断（如未屏蔽）（参见第 8.2.10 节“电源控制器中断状态和清零寄存器（PWRC_ISCR）”）。

5.4 工作模式

BlueNRG-LP 支持 3 种主要工作模式：

- 运行模式
- DEEPSTOP 模式
- SHUTDOWN 模式

通过 PMU 状态机管理从一种模式到另一种模式的转换。

5.4.1 RUN 模式

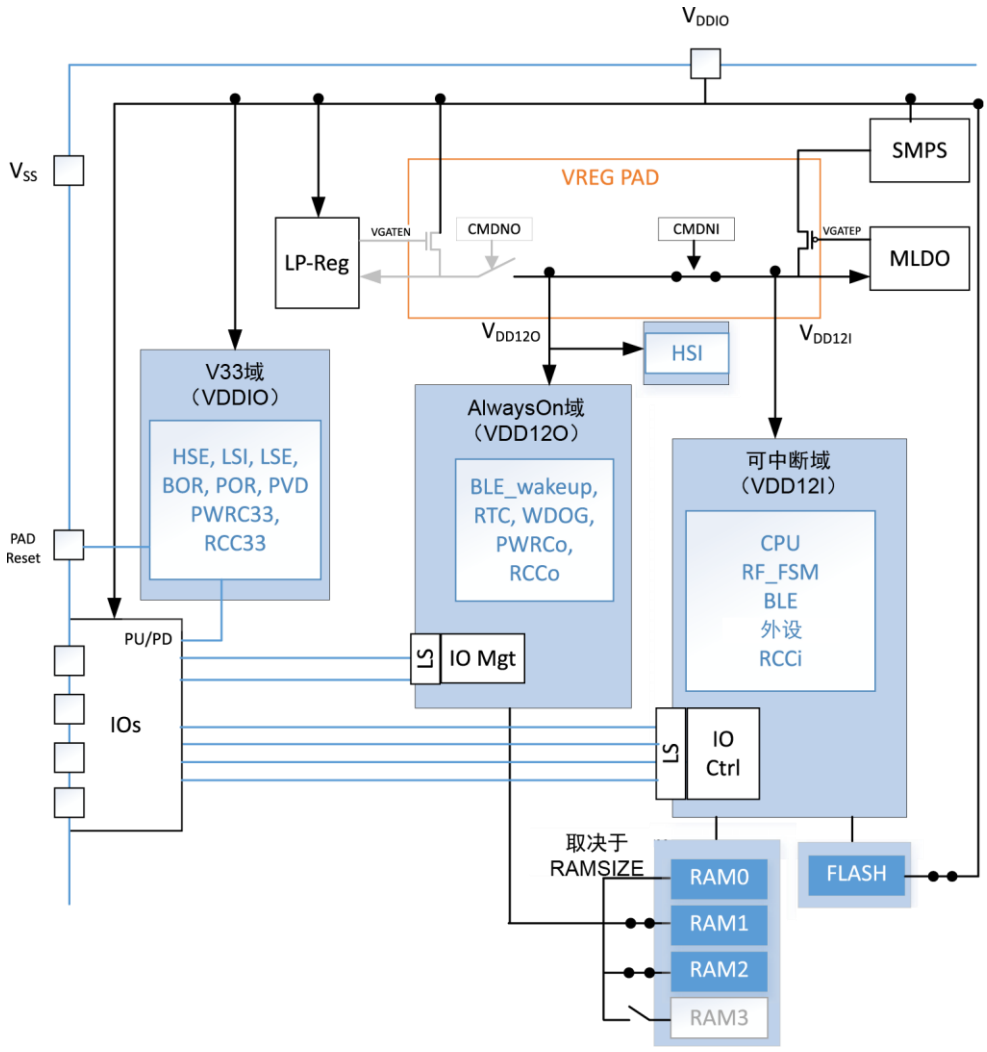
在 RUN 模式下：

- 两个稳压器（MLDO 和 LPREG）使能
- MLDO 为 VDD12i 和 VDD12o 提供电源
- 系统时钟和总线时钟正在运行
- 可使用 CPU 和无线电

借助于 RCC 时钟使能寄存器，可通过对未使用外设的时钟实施门控来降低功耗（参见第 6.7 节“RCC 寄存器映射”）。

图 6. RUN 模式下的功率调节器和 SMPS 配置显示了只有 48 kB RAM 的产品在 RUN 模式下的调节器和 SMPS 配置。

图 6. RUN 模式下的功率调节器和 SMPS 配置



5.4.2 DEEPSTOP 模式

DEEPSTOP 是 BlueNRG-LP 唯一的低功耗模式，能够在唤醒时从保存的上下文环境重启并继续运行应用程序。

进入 DEEPSTOP 模式的条件为：

- 无线电（MR_BLE）正在休眠
- CPU 进入睡眠状态（WFI，且 SLEEPDEEP 信息处于活跃状态）
- 没有活跃的未屏蔽唤醒源（包括软件在唤醒后没有将相关标志清零的先前唤醒序列的唤醒源）
- 系统通过 RC64MPLL（HSI 或 pll 锁定模式）计时
- PWRC_CR1.LPMS 位等于 0

注意： *如果 SoC 根本没有使用 MR_BLE（或尚未启动），在任何一次复位后，需执行以下步骤以便支持低功耗模式（DEEPSTOP 和 SHUTDOWN）：*

- *通过将 RCC_APB2ENR.MRBLEEN 位置 1 使能 MR_BLE 时钟*
- *将 MR_BLE 的唤醒模块内部的 BLUE_SLEEP_REQUEST_MODE.FORCE_SLEEPING 位置 1，以使 MR_BLE IP 向 SoC 请求低功耗模式*
- *通过将 RCC_APB2ENR.MRBLEEN 位清零重新对 MR_BLE 时钟实施门控*

在 DEEPSTOP 模式下：

- 由于 RC64MPLL 模块关闭，系统和总线时钟停止
- VDD12i 电源域关闭
- VDDI2o 电源域开启且供电电压为 1.0 V
- RAM0 区块数据保留
- 如果 PWRC_CR1.APC = 1，在 DEEPSTOP 模式下通过 PWRC_PUCRx/PWRC_PDCRx 控制 I/O 上/下拉
- 其他 RAM 存储区块是否保留取决于 PWRC_CR2 寄存器中的软件选择
- 慢速时钟可以是正在运行或停止，取决于进入 DEEPSTOP 模式之前的软件配置：
  - ON 或 OFF
  - LSE 或 LSI 源
- RTC 和 IWDOG 保持激活状态（如已使能且一个慢速时钟源开启）
- MR_BLE 唤醒模块（包括其定时器）保持激活状态（如已使能且一个慢速时钟源开启）
- 8 个 I/O（PA4/PA5/PA6/PA7/PA8/PA9/PA10/PA11）能够处于驱动静态低或高电平、慢速时钟信息或 RTC_OUT 的输出状态（参见第 5.6.15 节“I/O DEEPSTOP 驱动配置寄存器（WRC_IOxCFG）”了解详细信息）。

已实现了 DEEPSTOP 模式的名为 DEEPSTOP2 的版本，用于在不丢失调试器连接和断点或观察点的情况下模仿 DEEPSTOP 模式。

- 可通过将 PWRC_DBGR.DEEPSTOP2 位置 1 选择该变式
- 在这种情况下，无需关闭 VDD12i 电源域即可执行 DEEPSTOP 模式序列（进入和退出）

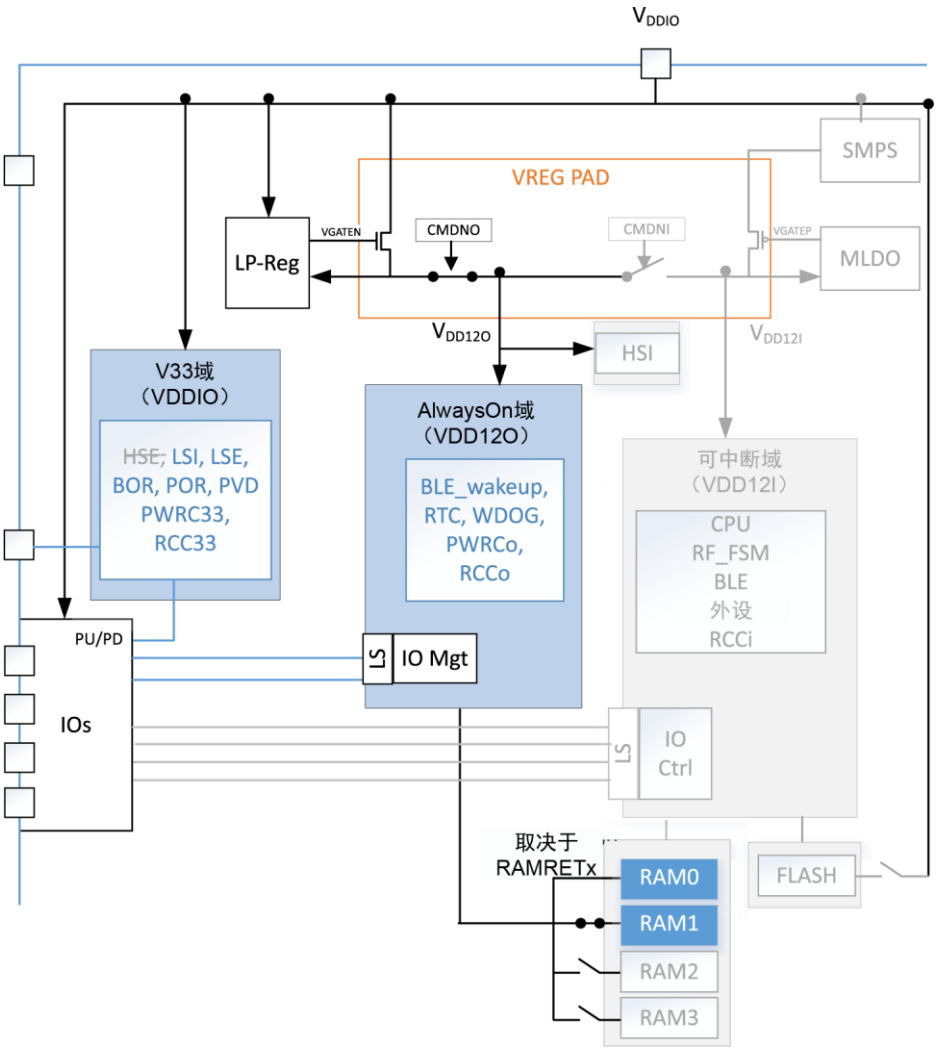
可能的唤醒源：

- MR_BLE 模块能够生成两个事件，以便通过其基于慢速时钟运行的嵌入式唤醒定时器唤醒系统：
  - 到达 BLE IP 唤醒时间
- RTC 能够生成唤醒事件
- IWDG 能够生成复位事件
- 至多 28 个 GPIO 能够唤醒系统（PA0 至 PA15 和 PB0 至 PB11）

唤醒时，VDD12i 电源域中的硬件资源复位，CPU 重启。PWRC 寄存器中可见唤醒原因（参见第 5.6.5 节“状态寄存器 1（PWRC_SR1）”了解详细信息）。

图 7. DEEPSTOP 模式下的功率调节器和 SMPS 配置显示了 DEEPSTOP 模式下的调节器和 SMPS 配置，该配置只请求保留 RAM0 和 RAM1 存储区块。

图 7. DEEPSTOP 模式下的功率调节器和 SMPS 配置



5.4.3 SHUTDOWN 模式

关断模式是功耗最小的模式。

进入 SHUTDOWN 模式的条件与进入 DEEPSTOP 模式所需的条件相同，只是 PWRC_CR1.LPMS 位必须等于 1 且 PWRC_DBGR.DEEPSTOP2 位必须保持复位。

在 SHUTDOWN 模式下：

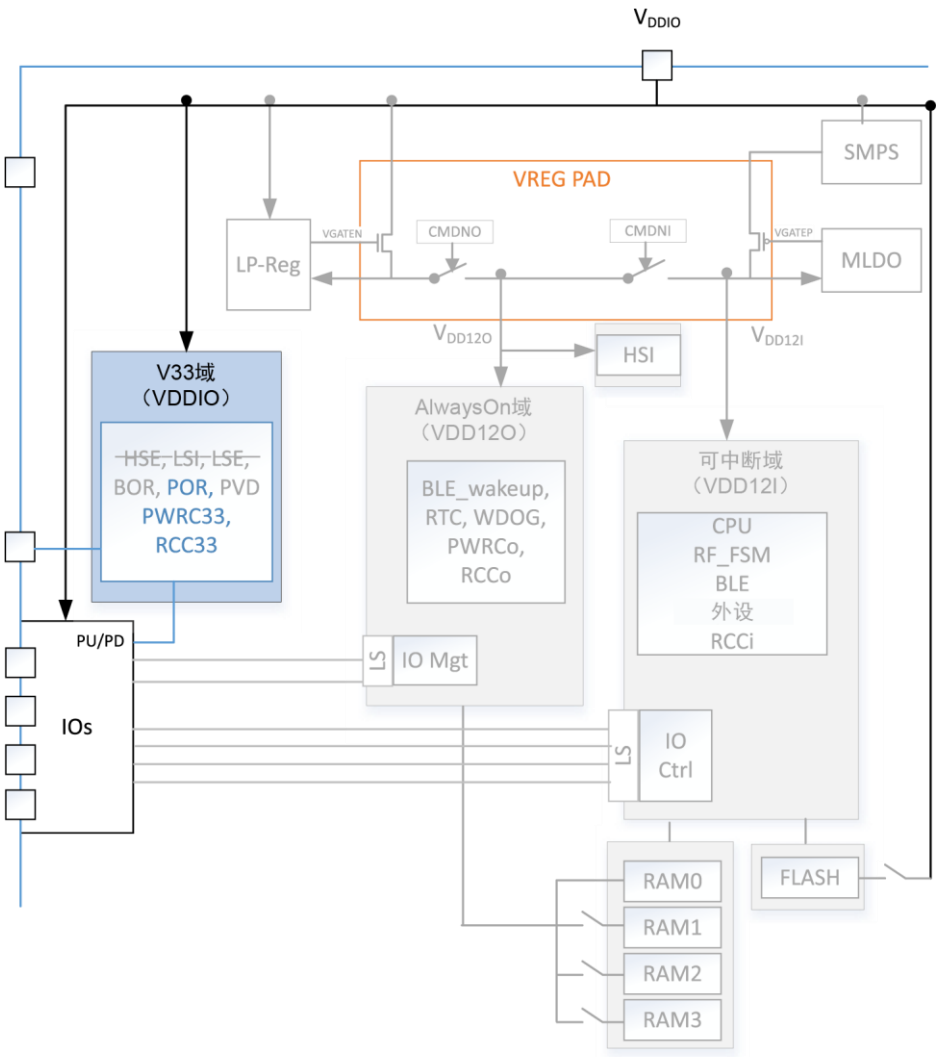
- 由于两个调节器均关闭，因此系统掉电（VDD12i 和 VDD12o 电源域均关闭）
- 只有 VDDIO 电源域开启
- 由于 RC64MPLL、LSI 和 LSE 关闭，所有时钟均关闭（系统和慢速时钟树）。
- 如果 PWRC_CR1.APC = 1，在 DEEPSTOP 模式下通过 PWRC_PUCRx/PWRC_PDCRx 控制 I/O 上/下拉
- 唯一的唤醒源是 RSTN PIN 脚上的低脉冲

关断退出类似于板子的 POR 启动。相关复位原因是 PORRSTF 标志（参见第 6.6.14 节“V33 复位状态寄存器（RCC_CSR）”了解复位原因标志详情）。

在 SHUTDOWN 模式下，可通过 PWRC_CR1.ENSDNBOR 位使能或禁用 BOR 特性。

图 8. SHUTDOWN 模式下的功率调节器和 SMPS 配置显示了 SHUTDOWN 模式下的调节器和 SMPS 配置，该配置禁用 BOR 复位。

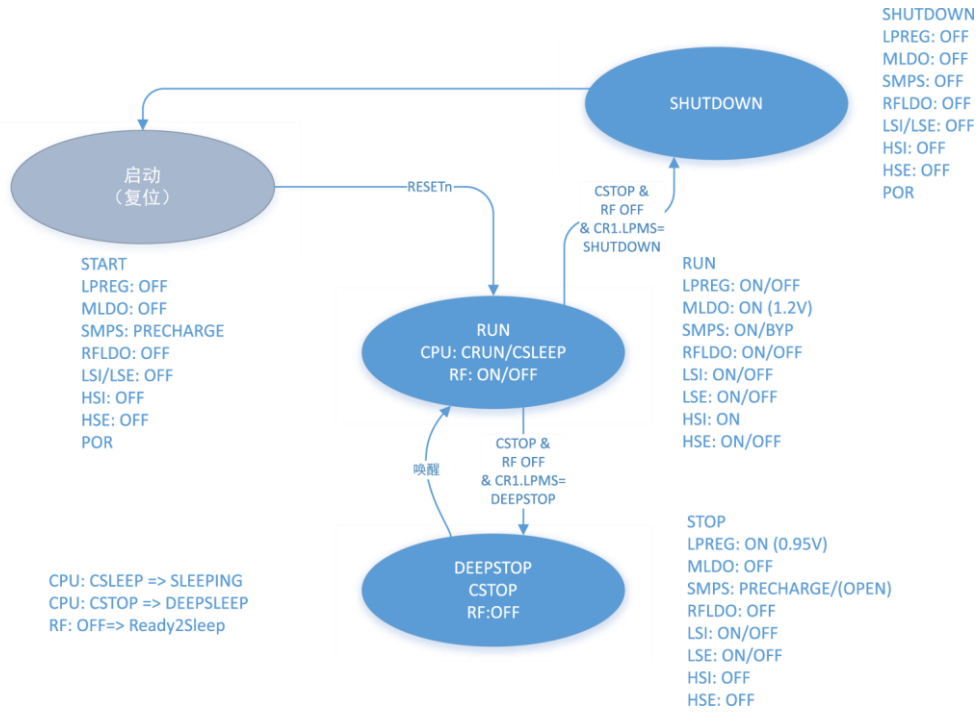
图 8. SHUTDOWN 模式下的功率调节器和 SMPS 配置



5.4.4 工作模式转换管理

PWRC 模块通过状态机管理从一种工作模式到另一种工作模式的切换。

图 9. 用于工作模式转换的 PWRC 状态机



5.5 SMPS 降压稳压器

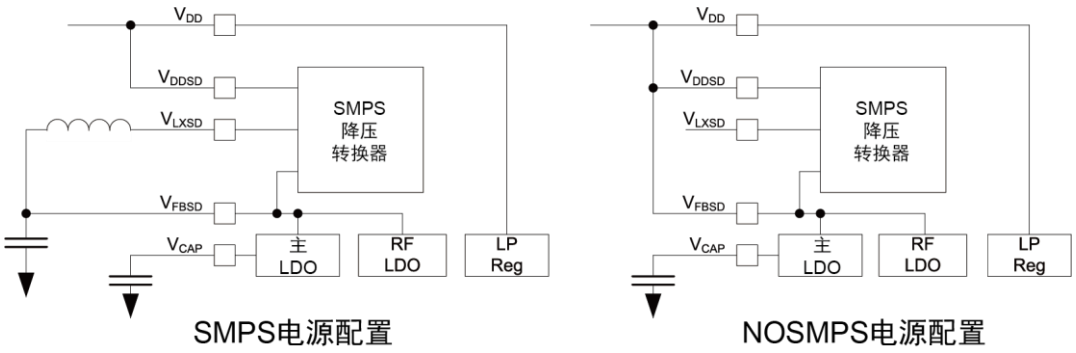
BlueNRG-LP SMPS 是一款 20 mA 输出降压 SMPS（开关模式电源）转换器。

SMPS 输出电压可设定为 1.2 V 至 1.90 V。内部时钟频率为 4 MHz 或 8 MHz。SMPS 可使用不同配置：

- ON：
  - SMPS 的 VFBSD 引脚输出稳定电压（1.2 V 至 1.9 V）
  - SMPS 需要时钟
- OFF：
  - VFBSD 引脚必须用 VDDIO 进行外部强制
  - SMPS 不需要时钟
- PRECHARGE（亦即 BYPASS）：
  - VFBSD 引脚输出未经调节的 VDDIO
  - SMPS 不需要时钟
- OPEN：
  - VFBSD 引脚浮空
  - SMPS 不需要时钟

除非配置 SMPS 关闭，否则板件上必须有 L/C BOM 且其必须连接到 VFBSD 焊盘（参见图 10. 电源配置）。

图 10. 电源配置



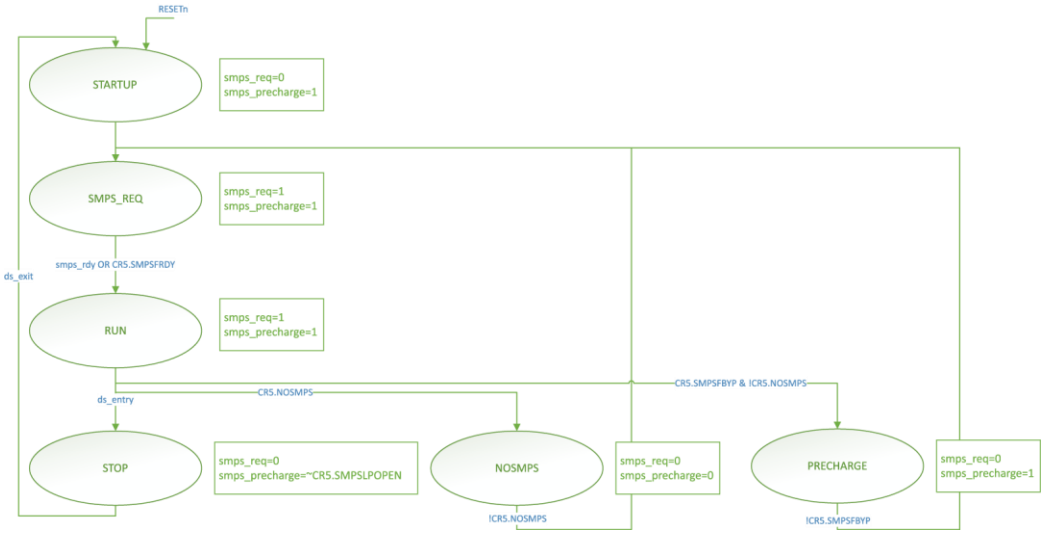
用户必须根据其板件上实现的 BOM 来配置 PWRC_CR5.SMPSBOMSEL[1:0]。要设定的值见“表 9. SMPS BOM 信息”。

表 9. SMPS BOM 信息

BOM	电感（L）	输出电容（C）	SMPS BOMSEL[1:0]
BOM1	1.5 uH	2.2 uF	00
BOM2	2.2 uH	4.7 uF	01
BOM3	10 uH	4.7 uF	10

PWRC 通过“图 11. PWRC SMPS 状态机总览”中所示的状态机管理 SMPS。

图 11. PWRC SMPS 状态机总览



在上电复位序列后，SMPS FSM 总是转至 RUN 状态。此时，SMPS FSM 可停留在三种状态（其他为转换状态）：

- RUN
  - SMPS 开启并处于 RUN 模式
  - SMPS 时钟运行
  - VFBSD 经过调节且电压幅度为在 PWRC_CR5.SMPSLVL 位字段中设定的值
  - 板上有 L/C BOM 并连接到 BlueNRG-LP 的 VFBSD 焊盘（参见“图 10. 电源配置”）
- 无 SMPS（若软件配置 PWRC_CR5.NOSMPS=1）：
  - SMPS 关闭
  - SMPS 时钟停止
  - VFBSD 通过 BlueNRG-LP 的 VFBSD 焊盘直接连接到 VDDIO（参见“图 10. 电源配置”）
  - 在低功耗进入/退出阶段，PWRC 不控制 SMPS 上的任何特定排序
- PRECHARGE，亦即 BYPASS（若软件配置 PWRC_CR5.NOSMPS=1）：
  - SMPS 开启并处于预充电模式
  - SMPS 时钟停止
  - VFBSD 是跨越 SMPS 模块的 VDDIO 电压
  - 在低功耗进入/退出阶段，PWRC 不控制 SMPS 上的任何特定排序
  - 此模式与 BlueNRG-LP 的 VFBSD 焊盘上连接的 L/C BOM 兼容

  当器件进入 DEEPSTOP 模式时，PWRC 自动将 SMPS 从 RUN 切换至 STOP 模式，这可以是：
  - PWRC_CR5.SMPSLPOPEN = 0 是否成立：SMPS 输出为 VDDIO（如同处于 PRECHARGE FSM 状态）
  - 如果 PWRC_CR5.SMPSLPOPEN = 1：SMPS 输出浮空

5.6 PWRC 寄存器说明

所有 PWRC APB 均只是 16 位寄存器。16 MSB 位始终保持为 0。

5.6.1 控制寄存器 1（PWRC_CR1）

此寄存器控制 SHUTDOWN 模式下的 BOR、低功耗模式选择和 IO 控制所有者。

偏移地址：0x00

复位值：0x0000 0010

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.	APC	Res.	Res.	ENSDNBOR	LPMS
											rw			rw	rw

位 31:5	保留，必须保持复位值。
位 4	**APC**：从 PWRC 或 GPIO 寄存器应用上/下拉配置。 • 0：使用 GPIO 模块的 GPIOx_PUPDR 控制 IO 的产品上/下拉 • 1：使用 PWRC 模块的 PWRC_PUCRx 和 PWRC_PDCRx 控制 IO 的产品上/下拉（默认）
位 3:2	保留，必须保持复位值。
位 1	**ENSDNBOR**：使能 SHUTDOWN 模式下的 BOR 复位监督。 • • 0：在 SHUTDOWN 模式下不监控 BOR（默认） • • 1：在 SHUTDOWN 模式下监控 BOR（在 SHUTDOWN 模式下，如果 VDDIO 低于 1.6 V，将发生 POR 复位） 注意：使能此特性可防止在 SHUTDOWN 模式下 VDDIO 低于支持的电压时封锁器件。但是，它增加了额外功耗。
位 0	**LPMS**：低功耗模式选择。 此位定义当 CPU 和 MR_BLE 均请求进入低功耗模式时，器件进入 DEEPSTOP 还是 SHUTDOWN 模式。 • 0：DEEPSTOP 模式（默认） • 1：HUTDOWN 模式（PWRC_DBGR.DEEPSTOP2 位必须保持复位）

5.6.2 控制寄存器 2（PWRC_CR2）

偏移地址：0x04

复位值：0x0000 0100

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.	LSILP MUFE N	ENTS	Res.	RAMR ET3	RAMR ET2	RAMR ET1	Res.	PVDLS[2:0]			PVDE
					rw	rw		rw	rw	rw		rw	rw	rw	rw

位 31:11	保留，必须保持复位值。
位 10	LSILPMUFEN：LSI LPMU 强制使能。 • 0：硬件根据模拟 LPMU 模块需求自动使能/禁用 LSI LPMU（默认） • 1：总是使能 LSI LPMU（32 kHz 自由运行时钟）
位 9	**ENTS**：使能温度传感器。 • 0：温度传感器禁用（默认） • 1：温度传感器使能
位 8	保留，必须保持复位值。
位 7	**RAMRET3**：使能 DEEPSTOP 模式下的 RAM3 存储区保留。 • 0：DEEPSTOP 模式下不保留 RAM3 存储区（默认） • 1：DEEPSTOP 模式下保留 RAM3 存储区
位 6	**RAMRET2**：使能 DEEPSTOP 模式下的 RAM2 存储区保留。 • 0：DEEPSTOP 模式下不保留 RAM2 存储区（默认） • 1：DEEPSTOP 模式下保留 RAM2 存储区
位 5	**RAMRET1**：使能 DEEPSTOP 模式下的 RAM1 存储区保留。 • 0：DEEPSTOP 模式下不保留 RAM1 存储区（默认） • 1：DEEPSTOP 模式下保留 RAM1 存储区
位 4	保留，必须保持复位值。
位 3:1	**PVDLS[2:0]**：可编程电压检测器的电平选择： • 000：2.05 V - 最低电平 • 001：2.20 V • 010：2.36 V • 011：2.52 V • 100：2.64 V • 101：2.81 V • 110：2.91 V - 最高电平 • 111：外部输入模拟电压（与 VBGP 进行内部比较）。在这种情况下，当外部电压低于 VBGP 时，PVDO 信号为高电平
位 0	**PVDE**：可编程电压检测器使能。 • 0：电源电压检测器特性禁用（默认） • 1：电源电压检测器特性使能

5.6.3 控制寄存器 3（PWRC_CR3）

该寄存器管理退出 DEEPSTOP 模式时唤醒源的选择。

注意： *复位后，默认禁用所有唤醒源。*

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.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EIWL	Res.	EWBLE HCPU	EWBLE	EWU11	EWU10	EWU9	EWU8	EWU7	EWU6	EWU5	EWU4	EWU3	EWU2	EWU1	EWU0
rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16	保留，必须保持复位值。
位 15	**EIWL**：使能在发生内部事件（RTC）时唤醒。 • 0：禁用内部线路上的唤醒（默认） • 1：使能内部线路上的唤醒
位 14	保留，必须保持复位值。
位 13	**EWBLEHCPU**：使能在发生 BLE 主机 CPU 事件时唤醒。 • 0：禁用 BLE 主机 CPU 线路上的唤醒（默认） • 1：使能 BLE 主机 CPU 线路上的唤醒
位 12	**EWBLE**：使能在发生 BLE 事件时唤醒。 • 0：禁用 BLE 线路上的唤醒（默认） • 1：使能 BLE 线路上的唤醒
位 11	**EWU11**：使能在发生 PA11 I/O 事件时唤醒。 • 0：禁用 PA11 I/O 线路上的唤醒（默认） • 1：使能 PA11 I/O 线路上的唤醒
位 10	**EWU10**：使能在发生 PA10 I/O 事件时唤醒。 • 0：禁用 PA10 I/O 线路上的唤醒（默认） • 1：使能 PA10 I/O 线路上的唤醒
位 9	**EWU9**：使能在发生 PA9 I/O 事件时唤醒。 • 0：禁用 PA9 I/O 线路上的唤醒（默认） • 1：使能 PA9 I/O 线路上的唤醒
位 8	**EWU8**：使能在发生 PA8 I/O 事件时唤醒。 • 0：禁用 PA8 I/O 线路上的唤醒（默认） • 1：使能 PA8 I/O 线路上的唤醒
位 7	**EWU7**：使能在发生 PB7 I/O 事件时唤醒。 • 0：禁用 PB7 I/O 线路上的唤醒（默认） • 1：使能 PB7 I/O 线路上的唤醒
位 6	**EWU6**：使能在发生 PB6 I/O 事件时唤醒。 • 0：禁用 PB6 I/O 线路上的唤醒（默认） • 1：使能 PB6 I/O 线路上的唤醒

位 5	**EWU5**：使能在发生 PB5 I/O 事件时唤醒。 • 0：禁用 PB5 I/O 线路上的唤醒（默认） • 1：使能 PB5 I/O 线路上的唤醒
位 4	**EWU4**：使能在发生 PB4 I/O 事件时唤醒。 • 0：禁用 PB4 I/O 线路上的唤醒（默认） • 1：使能 PB4 I/O 线路上的唤醒
位 3	**EWU3**：使能在发生 PB3 I/O 事件时唤醒。 • 0：禁用 PB3 I/O 线路上的唤醒（默认） • 1：使能 PB3 I/O 线路上的唤醒
位 2	**EWU2**：使能在发生 PB2 I/O 事件时唤醒。 • 0：禁用 PB2 I/O 线路上的唤醒（默认） • 1：使能 PB2 I/O 线路上的唤醒
位 1	**EWU1**：使能在发生 PB1 I/O 事件时唤醒。 • 0：禁用 PB1 I/O 线路上的唤醒（默认） • 1：使能 PB1 I/O 线路上的唤醒
位 0	**EWU0**：使能在发生 PB0 I/O 事件时唤醒。 • 0：禁用 PB0 I/O 线路上的唤醒（默认） • 1：使能 PB0 I/O 线路上的唤醒

5.6.4 控制寄存器 4（PWRC_CR4）

该寄存器管理退出 DEEPSTOP 模式时 I/O 唤醒源的极性。

注意： *唤醒事件只有边沿检测，无电平检测。*

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.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	WUP11	WUP10	WUP9	WUP8	WUP7	WUP6	WUP5	WUP4	WUP3	WUP2	WUP1	WUP0
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:12	保留，必须保持复位值。
位 11	**WUP11**：PA11 I/O 的唤醒极性。 • 0：在上升沿检测唤醒事件（默认） • 1：在下降沿检测唤醒事件
位 10	**WUP10**：PA10 I/O 的唤醒极性。 • 0：在上升沿检测唤醒事件（默认） • 1：在下降沿检测唤醒事件
位 9	**WUP9**：PA9 IO 事件的唤醒极性。 • 0：在上升沿检测唤醒事件（默认） • 1：在下降沿检测唤醒事件
位 8	**WUP8**：PA8 IO 事件的唤醒极性。 • 0：在上升沿检测唤醒事件（默认） • 1：在下降沿检测唤醒事件
位 7	**WUP7**：PB7 IO 事件的唤醒极性。 • 0：在上升沿检测唤醒事件（默认） • 1：在下降沿检测唤醒事件

位 6	**WUP6**：PB6 IO 事件的唤醒极性。 • 0：在上升沿检测唤醒事件（默认） • 1：在下降沿检测唤醒事件
位 5	**WUP5**：PB5 IO 事件的唤醒极性。 • 0：在上升沿检测唤醒事件（默认） • 1：在下降沿检测唤醒事件
位 4	**WUP4**：PB4 IO 事件的唤醒极性。 • 0：在上升沿检测唤醒事件（默认） • 1：在下降沿检测唤醒事件
位 3	**WUP3**：PB3 IO 事件的唤醒极性。 • 0：在上升沿检测唤醒事件（默认） • 1：在下降沿检测唤醒事件
位 2	**WUP2**：PB2 IO 事件的唤醒极性。 • 0：在上升沿检测唤醒事件（默认） • 1：在下降沿检测唤醒事件
位 1	**WUP1**：PB1 IO 事件的唤醒极性。 • 0：在上升沿检测唤醒事件（默认） • 1：在下降沿检测唤醒事件
位 0	**WUP0**：PB0 IO 事件的唤醒极性。 • 0：在上升沿检测唤醒事件（默认） • 1：在下降沿检测唤醒事件
位 2	**WUP2**：PB2 IO 事件的唤醒极性。 • 0：在上升沿检测唤醒事件（默认） • 1：在下降沿检测唤醒事件
位 1	**WUP1**：PB1 IO 事件的唤醒极性。 • 0：在上升沿检测唤醒事件（默认） • 1：在下降沿检测唤醒事件
位 0	**WUP0**：PB0 IO 事件的唤醒极性。 • 0：在上升沿检测唤醒事件（默认） • 1：在下降沿检测唤醒事件

5.6.5 状态寄存器 1（PWRC_SR1）

此寄存器提供 DEEPSTOP 后唤醒器件的源的相关信息。

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.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IWUF	Res.	WBLE HCPUF	WBLEF	WUF11	WUF10	WUF9	WUF8	WUF7	WUF6	WUF5	WUF4	WUF3	WUF2	WUF1	WUF0
r		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 31:16	保留，必须保持复位值。
位 15	**IWUF**：内部唤醒标志（RTC）。 • 0：最后一次清零后未发生来自 RTC 的唤醒 • 1：最后一次清零后发生了来自 RTC 的唤醒 注意：为了将此位清零，用户必须清除 RTC IP 内部的 RTC 唤醒标志（PWRC 模块上 RTC 唤醒线路的镜像）。
位 14	保留，必须保持复位值。
位 13	**WBLEHCPUF**：BLE 主机 CPU 唤醒标志。 • 0：最后一次清零后未发生来自 BLE 主机 CPU 的唤醒 • 1：最后一次清零后发生了来自 BLE 主机 CPU 的唤醒。通过在此位写入 1 进行清零
位 12	**WBLEF**：BLE 唤醒标志。 • 0：最后一次清零后未发生来自 BLE 的唤醒 • 1：最后一次清零后发生了来自 BLE 的唤醒。通过在此位写入 1 进行清零
位 11	**WUF11**：PA11 I/O 唤醒标志。 • 0：最后一次清零后未发生来自 PA11 I/O 的唤醒 • 1：最后一次清零后发生了来自 PA11 I/O 的唤醒。通过在此位写入 1 进行清零
位 10	**WUF10**：PA10 I/O 唤醒标志。 • 0：最后一次清零后未发生来自 PA10 I/O 的唤醒 • 1：最后一次清零后发生了来自 PA10 I/O 的唤醒。通过在此位写入 1 进行清零
位 9	**WUF9**：PA9 I/O 唤醒标志。 • 0：最后一次清零后未发生来自 PA9 I/O 的唤醒 • 1：最后一次清零后发生了来自 PA9 I/O 的唤醒。通过在此位写入 1 进行清零
位 8	**WUF8**：PA8 I/O 唤醒标志。 • 0：最后一次清零后未发生来自 PA8 I/O 的唤醒 • 1：最后一次清零后发生了来自 PA8 I/O 的唤醒。通过在此位写入 1 进行清零
位 7	**WUF7**：PB7 I/O 唤醒标志。 • 0：最后一次清零后未发生来自 PB7 I/O 的唤醒 • 1：最后一次清零后发生了来自 PB7 I/O 的唤醒。通过在此位写入 1 进行清零
位 6	**WUF6**：PB6 I/O 唤醒标志。 • 0：最后一次清零后未发生来自 PB6 I/O 的唤醒 • 1：最后一次清零后发生了来自 PB6 I/O 的唤醒。通过在此位写入 1 进行清零
位 5	**WUF5**：PB5 I/O 唤醒标志。 • 0：最后一次清零后未发生来自 PB5 I/O 的唤醒 • 1：最后一次清零后发生了来自 PB5 I/O 的唤醒。通过在此位写入 1 进行清零
位 4	**WUF4**：PB4 I/O 唤醒标志。 • 0：最后一次清零后未发生来自 PB4 I/O 的唤醒 • 1：最后一次清零后发生了来自 PB4 I/O 的唤醒。通过在此位写入 1 进行清零
位 3	**WUF3**：PB3 I/O 唤醒标志。 • 0：最后一次清零后未发生来自 PB3 I/O 的唤醒 • 1：最后一次清零后发生了来自 PB3 I/O 的唤醒。通过在此位写入 1 进行清零
位 2	**WUF2**：PB2 I/O 唤醒标志。 • 0：最后一次清零后未发生来自 PB2 I/O 的唤醒 • 1：最后一次清零后发生了来自 PB2 I/O 的唤醒。通过在此位写入 1 进行清零
位 1	**WUF1**：PB1 I/O 唤醒标志。 • 0：最后一次清零后未发生来自 PB1 I/O 的唤醒 • 1：最后一次清零后发生了来自 PB1 I/O 的唤醒。通过在此位写入 1 进行清零
位 0	**WUF0**：PB0 I/O 唤醒标志。 • 0：最后一次清零后未发生来自 PB0 I/O 的唤醒 • 1：最后一次清零后发生了来自 PB0 I/O 的唤醒。通过在此位写入 1 进行清零

5.6.6 状态寄存器 2（PWRC_SR2）

此寄存器提供一些与电源电压检测器和 SMPS 模块相关的状态标志。

偏移地址：0x14

复位值：0x0000 -306

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
IOBOOTVAL[3:0]				PVDO	Res.	REGM S	REGLP S	Res.	Res.	Res.	Res.	Res.	SMPS RDY	SMPSE NR	SMPSB YPR
r	r	r	r	r		r	r						r	r	r

位 31:16	保留，必须保持复位值。
位 15:12	**IOBOOTVAL**：在 POR 时锁存的 I/O 值。 • 位 3：在 POR 时锁存的 PA11 输入值， • 位 2：在 POR 时锁存的 PA10 输入值， • 位 1：在 POR 时锁存的 PA9 输入值， • 位 0：在 POR 时锁存的 PA8 输入值。 注意：举例来说，此信息可能会被自举程序用来管理串行接口上的自举。
位 11	**PVDO**：电源电压检测器输出。 当电源电压检测器使能（PWRC_CR2.PVDE=1）时，此位在 VDDIO 低于选定阈值时置 1（通过 PWRC_CR2.PVDLS 位字段）。 • 0：VDDIO 不低于阈值或 PVD 特性未使能 • 1：VDDIO 低于选定阈值
位 10	保留，必须保持复位值。
位 9	**REGMS**：主稳压器准备就绪状态。 • 0：主稳压器未准备就绪 • 1：主稳压器准备就绪
位 8	**REGLPS**：低功耗稳压器准备就绪状态。 • 0：低功耗稳压器未准备就绪 • 1：低功耗稳压器准备就绪
位 7:3	保留，必须保持复位值。
位 2	**SMPSRDY**：SMPS 准备就绪状态。 • 0：SMPS 稳压器未准备就绪 • 1：SMPS 稳压器准备就绪
位 1	**SMPSENR**：SMPS RUN 模式状态。 此位反映内部 ENABLE_3V3 控制信号，该信号连接到 SMPS 并由硬件驱动。 • 0：SMPS 稳压器不工作（在 PRECHARGE 或 NOSMPS 模式下） • 1：SMPS 稳压器处于 RUN 模式
位 0	**SMPSBYPR**：SMPS PRECHARGE 模式状态。 此位反映 SMPS 的 PRECHARGE 控制状态。 • 0：SMPS 稳压器不处于 PRECHARGE 模式 • 1：SMPS 稳压器处于 PRECHARGE 模式（连接到 VDDIO 的 VSMPS）

5.6.7 控制寄存器 5（PWRC_CR5）

此寄存器用于配置 SMPS。

偏移地址：0x1C

复位值：0x0000 0014

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.	CLKDE TR_DI SABLE	SMPS_ ENA_D CM	NOSM PS	SMPSF BYP	SMPSL POPEN	Res.	Res.	SMPSBOMSEL[1:0]		SMPSLVL[3:0]			
			rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw

位 31:13	保留，必须保持复位值。
位 12	**CLKDETR_DISABLE**：禁用 SMPS 时钟检测。 SMPS 时钟检测能够在发生意外的 SMPS 时钟丢失时实现自动 SMPS 旁路切换。 • 0：使能 SMPS 时钟检测机制（默认） • 1：禁用 SMPS 时钟检测机制
位 11	**SMPS_ENA_DCM**：非连续导通模式使能。 • 0：禁用 SMPS DCM（默认） • 1：使能 SMPS DCM
位 10	**NOSMPS**：无 SMPS 模式。 • 0：使能 SMPS（默认） • 1：禁用 SMPS *注意：　只有在 SMPS_FB 焊盘直接连接到 VBATT（外部电压）（无 L/C BOM）时，才应使用此配置（SMPS 禁用）。*
位 9	**SMPSFBYP**：将 SMPS 强制为 PRECHARGE 模式。 • 0：无操作（默认） • 1：SMPS 禁用并被旁路 *注意：　当此位置 1 时，VSMPS 输出连接到 VDDIO。SMPS 的实际状态显示在 PWRC_SR2 寄存器中的 SMPS 模式状态位。*
位 8	**SMPSLPOPEN**：在 DEEPSTOP 期间，为 SMPS 选择 OPEN 模式而不是 PRECHARGE 模式。 • 0：在 DEEPSTOP 模式下，SMPS 处于 PRECHARGE 模式且输出连接到 VDDIO（默认） • 1：在 DEEPSTOP 模式下，SMPS 被禁用并具有浮动输出
位 7:6	保留，必须保持复位值。
位 5:4	**SMPSBOMSEL[1:0]**：选择 SMPS BOM。 • 00：BOM1 • 01：BOM2（默认） • 10：BOM3 • 11：不适用 注意：BOM 相关信息/详情见“表 9. SMPS BOM 信息”。

位 3:0	**SMPSLVL[3:0]**：选择 SMPS 输出电压值。 此位字段选择 SMPS 电压输出值，粒度为 50 mV。配置的 SMPS 输出电压值（VSMPS）必须使 VBAT-VSMPS ≥ 0.2 V。 [例如，当 VBAT=2 V 时，VSMPS 不得高于 1.8 V] • -0000：1.2 V • -0001：1.25 V • -0010：1.3 V • -0011：1.35 V • -0100：1.4 V • -0101：1.45 V • -0110：1.5 V • -0111：1.55 V • -1000：1.6 V • -1001：1.65 V • -1010：1.7 V • -1011：1.75 V • -1100：1.8 V • -1101：1.85 V • -1110：1.9 V • -1111：1.9 V 警告：当 SMPS 正在使用时，SMPS 输出电压的修改操作不得多于一个步骤。第 5.8.2 节“重新设定 SMPS 输出电压值”中描述了重新设定新的 SMPS 输出电压的操作序列。

5.6.8 I/O 端口 A 上拉控制寄存器（PWRC_PUCRA）

该寄存器用于控制当 PWRC_CR1.APC 位置 1 时 PA0 至 PA15 I/O 的上拉。

注意：如果在 PWRC_PUCRA 和 PWRC_PDCRA 寄存器中使能了 I/O 的上拉和下拉，则应用下拉。

用户在禁用在模拟模式下设定的上/下拉开启的 I/O 时必须谨慎。偏移地址：0x20

复位值：0x0000 FFF7

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
PUA15	PUA14	PUA13	PUA12	PUA11	PUA10	PUA9	PUA8	PUA7	PUA6	PUA5	PUA4	PUA3	PUA2	PUA1	PUA0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16	保留，必须保持复位值。
位 15	**PUA15**：PA15 I/O 的上拉使能。 • 0：不上拉 • 1：上拉使能（默认）
位 14	**PUA14**：PA14 I/O 的上拉使能。 • 0：不上拉 • 1：上拉使能（默认）
位 13	**PUA13**：PA13 I/O 的上拉使能。 • 0：不上拉 • 1：上拉使能（默认）

位 12	**PUA12**：PA12 I/O 的上拉使能。 • 0：不上拉 • 1：上拉使能（默认）
位 11	**PUA11**：PA11 I/O 的上拉使能。 • 0：不上拉 • 1：上拉使能（默认）
位 10	**PUA10**：PA10 I/O 的上拉使能。 • 0：不上拉 • 1：上拉使能（默认）
位 9	**PUA9**：PA9 I/O 的上拉使能。 • 0：不上拉 • 1：上拉使能（默认）
位 8	**PUA8**：PA8 I/O 的上拉使能。 • 0：不上拉 • 1：上拉使能（默认）
位 7	**PUA7**：PA7 I/O 的上拉使能。 • 0：不上拉 • 1：上拉使能（默认）
位 6	**PUA6**：PA6 I/O 的上拉使能。 • 0：不上拉 • 1：上拉使能（默认）
位 5	**PUA5**：PA5 I/O 的上拉使能。 • 0：不上拉 • 1：上拉使能（默认）
位 4	**PUA4**：PA4 I/O 的上拉使能。 • 0：不上拉 • 1：上拉使能（默认）
位 3	**PUA3**：PA3 I/O 的上拉使能。 • 0：不上拉（默认） • 1：上拉使能
位 2	**PUA2**：PA2 I/O 的上拉使能。 • 0：不上拉 • 1：上拉使能（默认）
位 1	**PUA1**：PA1 I/O 的上拉使能。 • 0：不上拉 • 1：上拉使能（默认）
位 0	**PUA0**：PA0 I/O 的上拉使能。 • 0：不上拉 • 1：上拉使能（默认）

5.6.9 I/O 端口 A 下拉控制寄存器（PWRC_PDCRA）

该寄存器用于控制当 PWRC_CR1.APC 位置 1 时 PA0 至 PA15 I/O 的下拉。

注意：如果在 PWRC_PUCRA 和 PWRC_PDCRA 寄存器中使能了 I/O 的上拉和下拉，则应用下拉。

用户在禁用在模拟模式下设定的上/下拉开启的 I/O 时必须谨慎。

偏移地址：0x24

复位值：0x0000 0008

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
PDA15	PDA14	PDA13	PDA12	PDA11	PDA10	PDA9	PDA8	PDA7	PDA6	PDA5	PDA4	PDA3	PDA2	PDA1	PDA0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16	保留，必须保持复位值。
位 15	**PDA15**：PA15 I/O 的下拉使能。 • 0：不下拉（默认） • 1：下拉使能
位 14	**PDA14**：PA14 I/O 的下拉使能。 • 0：不下拉（默认） • 1：下拉使能
位 13	**PDA13**：PA13 I/O 的下拉使能。 • 0：不下拉（默认） • 1：下拉使能
位 12	**PDA12**：PA12 I/O 的下拉使能。 • 0：不下拉（默认） • 1：下拉使能
位 11	**PDA11**：PA11 I/O 的下拉使能。 • 0：不下拉（默认） • 1：下拉使能
位 10	**PDA10**：PA10 I/O 的下拉使能。 • 0：不下拉（默认） • 1：下拉使能
位 9	**PDA9**：PA9 I/O 的下拉使能。 • 0：不下拉（默认） • 1：下拉使能
位 8	**PDA8**：PA8 I/O 的下拉使能。 • 0：不下拉（默认） • 1：下拉使能
位 7	**PDA7**：PA7 I/O 的下拉使能。 • 0：不下拉（默认） • 1：下拉使能
位 6	**PDA6**：PA6 I/O 的下拉使能。 • 0：不下拉（默认） • 1：下拉使能
位 5	**PDA5**：PA5 I/O 的下拉使能。 • 0：不下拉（默认） • 1：下拉使能
位 4	**PDA4**：PA4 I/O 的下拉使能。 • 0：不下拉（默认） • 1：下拉使能
位 3	**PDA3**：PA3 I/O 的下拉使能。 • 0：不下拉 • 1：下拉使能（默认）
位 2	**PDA2**：PA2 I/O 的下拉使能。 • 0：不下拉（默认） • 1：下拉使能
位 1	**PDA1**：PA1 I/O 的下拉使能。 • 0：不下拉（默认） • 1：下拉使能
位 0	**PDA0**：PA0 I/O 的下拉使能。 • 0：不下拉（默认） • 1：下拉使能

5.6.10 I/O 端口 B 上拉控制寄存器（PWRC_PUCRB）

该寄存器用于控制当 PWRC_CR1.APC 位置 1 时 PB0 至 PB15 I/O 的上拉。

注意：如果在 PWRC_PUCRA 和 PWRC_PDCRA 寄存器中使能了 I/O 的上拉和下拉，则应用下拉。

用户在禁用在模拟模式下设定的上/下拉开启的 I/O 时必须谨慎。

偏移地址：0x28

复位值：0x0000 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
PUB15	PUB14	PUB13	PUB12	PUB11	PUB10	PUB9	PUB8	PUB7	PUB6	PUB5	PUB4	PUB3	PUB2	PUB1	PUB0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16	保留，必须保持复位值。
位 15	**PUB15**：PB15 I/O 的上拉使能。 • 0：不上拉 • 1：上拉使能（默认）
位 14	**PUB14**：PB14 I/O 的上拉使能。 • 0：不上拉 • 1：上拉使能（默认）
位 13	**PUB13**：PB13 I/O 的上拉使能。 • 0：不上拉 • 1：上拉使能（默认）
位 12	**PUB12**：PB12 I/O 的上拉使能。 • 0：不上拉 • 1：上拉使能（默认）
位 11	**PUB11**：PB11 I/O 的上拉使能。 • 0：不上拉 • 1：上拉使能（默认）
位 10	**PUB10**：PB10 I/O 的上拉使能。 • 0：不上拉 • 1：上拉使能（默认）
位 9	**PUB9**：PB9 I/O 的上拉使能。 • 0：不上拉 • 1：上拉使能（默认）
位 8	**PUB8**：PB8 I/O 的上拉使能。 • 0：不上拉 • 1：上拉使能（默认）
位 7	**PUB7**：PB7 I/O 的上拉使能。 • 0：不上拉 • 1：上拉使能（默认）
位 6	**PUB6**：PB6 I/O 的上拉使能。 • 0：不上拉 • 1：上拉使能（默认）
位 5	**PUB5**：PB5 I/O 的上拉使能。 • 0：不上拉 • 1：上拉使能（默认）

位 4	**PUB4**：PB4 I/O 的上拉使能。 • 0：不上拉 • 1：上拉使能（默认）
位 3	**PUB3**：PB3 I/O 的上拉使能。 • 0：不上拉 • 1：上拉使能（默认）
位 2	**PUB2**：PB2 I/O 的上拉使能。 • 0：不上拉 • 1：上拉使能（默认）
位 1	**PUB1**：PB1 I/O 的上拉使能。 • 0：不上拉 • 1：上拉使能（默认）
位 0	**PUB0**：PB0 I/O 的上拉使能。 • 0：不上拉 • 1：上拉使能（默认）

5.6.11 I/O 端口 B 下拉控制寄存器（PWRC_PDCRB）

该寄存器用于控制当 PWRC_CR1.APC 位置 1 时 PB0 至 PB15 I/O 的下拉。

注意：如果在 PWRC_PUCRA 和 PWRC_PDCRA 寄存器中使能了 I/O 的上拉和下拉，则应用下拉。

用户在禁用在模拟模式下设定的上/下拉开启的 I/O 时必须谨慎。

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
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
PDB15	PDB14	PDB13	PDB12	PDB11	PDB10	PDB9	PDB8	PDB7	PDB6	PDB5	PDB4	PDB3	PDB2	PDB1	PDB0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16	保留，必须保持复位值。
位 15	**PDB15**：PB15 I/O 的下拉使能。 • 0：不下拉（默认） • 1：下拉使能
位 14	**PDB14**：PB14 I/O 的下拉使能。 • 0：不下拉（默认） • 1：下拉使能
位 13	**PDB13**：PB13 I/O 的下拉使能。 • 0：不下拉（默认） • 1：下拉使能
位 12	**PDB12**：PB12 I/O 的下拉使能。 • 0：不下拉（默认） • 1：下拉使能
位 11	**PDB11**：PB11 I/O 的下拉使能。 • 0：不下拉（默认） • 1：下拉使能

位 10	**PDB10**：PB10 I/O 的下拉使能。 • 0：不下拉（默认） • 1：下拉使能
位 9	**PDB9**：PB9 I/O 的下拉使能。 • 0：不下拉（默认） • 1：下拉使能
位 8	**PDB8**：PB8 I/O 的下拉使能。 • 0：不下拉（默认） • 1：下拉使能
位 7	**PDB7**：PB7 I/O 的下拉使能。 • 0：不下拉（默认） • 1：下拉使能
位 6	**PDB6**：PB6 I/O 的下拉使能。 • 0：不下拉（默认） • 1：下拉使能
位 5	**PDB5**：PB5 I/O 的下拉使能。 • 0：不下拉（默认） • 1：下拉使能
位 4	**PDB4**：PB4 I/O 的下拉使能。 • 0：不下拉（默认） • 1：下拉使能
位 3	**PDB3**：PB3 I/O 的下拉使能。 • 0：不下拉（默认） • 1：下拉使能
位 2	**PDB2**：PB2 I/O 的下拉使能。 • 0：不下拉（默认） • 1：下拉使能
位 1	**PDB1**：PB1 I/O 的下拉使能。 • 0：不下拉（默认） • 1：下拉使能
位 0	**PDB0**：PB0 I/O 的下拉使能。 • 0：不下拉（默认） • 1：下拉使能

5.6.12 控制寄存器 6（PWRC_CR6）

该寄存器管理退出 DEEPSTOP 模式时唤醒源的选择。

注意： *复位后，默认禁用所有唤醒源。*

偏移地址：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.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EWU27	EWU26	EWU25	EWU24	EWU22	EWU22	EWU21	EWU20	EWU19	EWU18	EWU17	EWU16	EWU15	EWU14	EWU13	EWU12
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16	保留，必须保持复位值。
位 15	**EWU27**：使能在发生 PA15 I/O 事件时唤醒。 • 0：禁用 PA15 I/O 线路上的唤醒（默认） • 1：使能 PA15 I/O 线路上的唤醒
位 14	**EWU26**：使能在发生 PA14 I/O 事件时唤醒。 • 0：禁用 PA14 I/O 线路上的唤醒（默认） • 1：使能 PA14 I/O 线路上的唤醒
位 13	**EWU25**：使能在发生 PA13 I/O 事件时唤醒。 • 0：禁用 PA13 I/O 线路上的唤醒（默认） • 1：使能 PA13 I/O 线路上的唤醒
位 12	**EWU24**：使能在发生 PA12 I/O 事件时唤醒。 • 0：禁用 PA12 I/O 线路上的唤醒（默认） • 1：使能 PA12 I/O 线路上的唤醒
位 11	**EWU23**：使能在发生 PB11 I/O 事件时唤醒。 • 0：禁用 PB11 I/O 线路上的唤醒（默认） • 1：使能 PB11 I/O 线路上的唤醒
位 10	**EWU22**：使能在发生 PB10 I/O 事件时唤醒。 • 0：禁用 PB10 I/O 线路上的唤醒（默认）。 • 1：使能 PB10 I/O 线路上的唤醒。
位 9	**EWU21**：使能在发生 PB9 I/O 事件时唤醒。 • 0：禁用 PB9 I/O 线路上的唤醒（默认） • 1：使能 PB9 I/O 线路上的唤醒
位 8	**EWU20**：使能在发生 PB8 I/O 事件时唤醒。 • 0：禁用 PB8 I/O 线路上的唤醒（默认） • 1：使能 PB8 I/O 线路上的唤醒
位 7	**EWU19**：使能在发生 PA7 I/O 事件时唤醒。 • 0：禁用 PA7 I/O 线路上的唤醒（默认） • 1：使能 PA7 I/O 线路上的唤醒
位 6	**EWU18**：使能在发生 PA6 I/O 事件时唤醒。 • 0：禁用 PA6 I/O 线路上的唤醒（默认） • 1：使能 PA6 I/O 线路上的唤醒
位 5	**EWU17**：使能在发生 PA5 I/O 事件时唤醒。 • 0：禁用 PA5 I/O 线路上的唤醒（默认） • 1：使能 PA5 I/O 线路上的唤醒
位 4	**EWU16**：使能在发生 PA4 I/O 事件时唤醒。 • 0：禁用 PA4 I/O 线路上的唤醒（默认） • 1：使能 PA4 I/O 线路上的唤醒
位 3	**EWU15**：使能在发生 PA3 I/O 事件时唤醒。 • 0：禁用 PA3 I/O 线路上的唤醒（默认） • 1：使能 PA3 I/O 线路上的唤醒
位 2	**EWU14**：使能在发生 PA2 I/O 事件时唤醒。 • 0：禁用 PA2 I/O 线路上的唤醒（默认） • 1：使能 PA2 I/O 线路上的唤醒
位 1	**EWU13**：使能在发生 PA1 I/O 事件时唤醒。 • 0：禁用 PA1 I/O 线路上的唤醒（默认） • 1：使能 PA1 I/O 线路上的唤醒
位 0	**EWU12**：使能在发生 PA0 I/O 事件时唤醒。 • 0：禁用 PA0 I/O 线路上的唤醒（默认） • 1：使能 PA0 I/O 线路上的唤醒

5.6.13 控制寄存器 7（PWRC_CR7）

该寄存器管理退出 DEEPSTOP 模式时 I/O 唤醒源的极性。

注意： *唤醒事件只有边沿检测，无电平检测。*

偏移地址：0x34

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
WUP27	WUP26	WUP25	WUP24	WUP23	WUP22	WUP21	WUP20	WUP19	WUP18	WUP17	WUP16	WUP15	WUP14	WUP13	WUP12
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16	保留，必须保持复位值。
位 15	**WUP27**：PA15 I/O 的唤醒极性。 • 0：在上升沿检测唤醒事件（默认） • 1：在下降沿检测唤醒事件
位 14	**WUP26**：PA14 I/O 的唤醒极性。 • 0：在上升沿检测唤醒事件（默认） • 1：在下降沿检测唤醒事件
位 13	**WUP25**：PA13 IO 事件的唤醒极性。 • 0：在上升沿检测唤醒事件（默认） • 1：在下降沿检测唤醒事件
位 12	**WUP24**：PA12 I/O 的唤醒极性。 • 0：在上升沿检测唤醒事件（默认） • 1：在下降沿检测唤醒事件
位 11	**WUP23**：PB11 I/O 的唤醒极性。 • 0：在上升沿检测唤醒事件（默认） • 1：在下降沿检测唤醒事件
位 10	**WUP22**：PB10 I/O 的唤醒极性。 • 0：在上升沿检测唤醒事件（默认） • 1：在下降沿检测唤醒事件
位 9	**WUP21**：PB9 IO 事件的唤醒极性。 • 0：在上升沿检测唤醒事件（默认） • 1：在下降沿检测唤醒事件
位 8	**WUP20**：PB8 IO 事件的唤醒极性。 • 0：在上升沿检测唤醒事件（默认） • 1：在下降沿检测唤醒事件
位 7	**WUP19**：PA7 IO 事件的唤醒极性。 • 0：在上升沿检测唤醒事件（默认） • 1：在下降沿检测唤醒事件
位 6	**WUP18**：PA6 IO 事件的唤醒极性。 • 0：在上升沿检测唤醒事件（默认） • 1：在下降沿检测唤醒事件
位 5	**WUP17**：PA5 IO 事件的唤醒极性。 • 0：在上升沿检测唤醒事件（默认） • 1：在下降沿检测唤醒事件

位 4	**WUP16**：PA4 IO 事件的唤醒极性。 • 0：在上升沿检测唤醒事件（默认） • 1：在下降沿检测唤醒事件
位 3	**WUP15**：PA3 IO 事件的唤醒极性。 • 0：在上升沿检测唤醒事件（默认） • 1：在下降沿检测唤醒事件
位 2	**WUP14**：PA2 IO 事件的唤醒极性。 • 0：在上升沿检测唤醒事件（默认） • 1：在下降沿检测唤醒事件
位 1	**WUP13**：PA1 IO 事件的唤醒极性。 • 0：在上升沿检测唤醒事件（默认） • 1：在下降沿检测唤醒事件
位 0	**WUP12**：PA0 IO 事件的唤醒极性。 • 0：在上升沿检测唤醒事件（默认） • 1：在下降沿检测唤醒事件

5.6.14 状态寄存器 3（PWRC_SR3）

此寄存器提供 DEEPSTOP 后唤醒器件的源的一些相关信息。

偏移地址：0x38

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
WUF27	WUF26	WUF25	WUF24	WUF23	WUF22	WUF21	WUF20	WUF19	WUF18	WUF17	WUF16	WUF15	WUF14	WUF13	WUF12
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 31:16	保留，必须保持复位值。
位 15	**WUF27**：PA15 I/O 唤醒标志。 • 0：最后一次清零后未发生来自 PA15 I/O 的唤醒 • 1：最后一次清零后发生了来自 PA15 I/O 的唤醒。通过在此位写入 1 进行清零
位 14	**WUF26**：PA14 I/O 唤醒标志。 • 0：最后一次清零后未发生来自 PA14 I/O 的唤醒 • 1：最后一次清零后发生了来自 PA14 I/O 的唤醒。通过在此位写入 1 进行清零
位 13	**WUF25**：PA13 I/O 唤醒标志。 • 0：最后一次清零后未发生来自 PA13 I/O 的唤醒 • 1：最后一次清零后发生了来自 PA13 I/O 的唤醒。通过在此位写入 1 进行清零
位 12	**WUF24**：PA12 I/O 唤醒标志。 • 0：最后一次清零后未发生来自 PA12 I/O 的唤醒 • 1：最后一次清零后发生了来自 PA12 I/O 的唤醒。通过在此位写入 1 进行清零
位 11	**WUF23**：PB11 I/O 唤醒标志。 • 0：最后一次清零后未发生来自 PB11 I/O 的唤醒 • 1：最后一次清零后发生了来自 PB11 I/O 的唤醒。通过在此位写入 1 进行清零
位 10	**WUF22**：PB10 I/O 唤醒标志。 • 0：最后一次清零后未发生来自 PB10 I/O 的唤醒 • 1：最后一次清零后发生了来自 PB10 I/O 的唤醒。通过在此位写入 1 进行清零

位 9	**WUF21**：PB9 I/O 唤醒标志。 • 0：最后一次清零后未发生来自 PB9 I/O 的唤醒 • 1：最后一次清零后发生了来自 PB9 I/O 的唤醒。通过在此位写入 1 进行清零
位 8	**WUF20**：PB8 I/O 唤醒标志。 • 0：最后一次清零后未发生来自 PB8 I/O 的唤醒 • 1：最后一次清零后发生了来自 PB8 I/O 的唤醒。通过在此位写入 1 进行清零
位 7	**WUF19**：PA7 I/O 唤醒标志。 • 0：最后一次清零后未发生来自 PA7 I/O 的唤醒 • 1：最后一次清零后发生了来自 PA7 I/O 的唤醒。通过在此位写入 1 进行清零
位 6	**WUF18**：PA6 I/O 唤醒标志。 • 0：最后一次清零后未发生来自 PA6 I/O 的唤醒 • 1：最后一次清零后发生了来自 PA6 I/O 的唤醒。通过在此位写入 1 进行清零
位 5	**WUF17**：PA5 I/O 唤醒标志。 • 0：最后一次清零后未发生来自 PA5 I/O 的唤醒 • 1：最后一次清零后发生了来自 PA5 I/O 的唤醒。通过在此位写入 1 进行清零
位 4	**WUF16**：PA4 I/O 唤醒标志。 • 0：最后一次清零后未发生来自 PA4 I/O 的唤醒 • 1：最后一次清零后发生了来自 PA4 I/O 的唤醒。通过在此位写入 1 进行清零
位 3	**WUF15**：PA3 I/O 唤醒标志。 • 0：最后一次清零后未发生来自 PA3 I/O 的唤醒 • 1：最后一次清零后发生了来自 PA3 I/O 的唤醒。通过在此位写入 1 进行清零
位 2	**WUF14**：PA2 I/O 唤醒标志。 • 0：最后一次清零后未发生来自 PA2 I/O 的唤醒 • 1：最后一次清零后发生了来自 PA2 I/O 的唤醒。通过在此位写入 1 进行清零
位 1	**WUF13**：PA1 I/O 唤醒标志。 • 0：最后一次清零后未发生来自 PA1 I/O 的唤醒 • 1：最后一次清零后发生了来自 PA1 I/O 的唤醒。通过在此位写入 1 进行清零
位 0	**WUF12**：PA0 I/O 唤醒标志。 • 0：最后一次清零后未发生来自 PA0 I/O 的唤醒 • 1：最后一次清零后发生了来自 PA0 I/O 的唤醒。通过在此位写入 1 进行清零

5.6.15 I/O DEEPSTOP 驱动配置寄存器（PWRC_IOxCFG）

此寄存器用于配置 8 个 I/O 的行为，使之能够在 DEEPSTOP 模式下输出信号(PA4/PA5/PA6/PA7/PA8/PA9/PA10/PA11)。

注意： *当不同于 BYPASS 模式时，在 PWRC_IOxCFG 寄存器中定义的配置将重载 GPIO 模块中通过 GPIOx_MODER 和GPIOx_ODR 设定的配置。*

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
IOCFG7[1:0]		IOCFG6[1:0]		IOCFG5[1:0]		IOCFG4[1:0]		IOCFG3[1:0]		IOCFG2[1:0]		IOCFG1[1:0]		IOCFG0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16	保留，必须保持复位值。
位 15:14	**IOCFG7[1:0]**：PA7 的驱动配置。 • 00：BYPASS 模式（默认） 在活动模式下，I/O 模式由 GPIO 模块寄存器控制，并在 DEEPSTOP 状态下切换至输入模式。 • 01：RTC_OUT 信号在活动模式和 DEEPSTOP 模式下均为输出 • 10：I/O 在活动模式和 DEEPSTOP 模式下均驱动低电平 • 11：I/O 在活动模式和 DEEPSTOP 模式下均驱动高电平
位 13:12	**IOCFG6[1:0]**：PA6 的驱动配置。 • 00：BYPASS 模式（默认） 在活动模式下，I/O 模式由 GPIO 模块寄存器控制，并在 DEEPSTOP 状态下切换至输入模式。 • 01：LCO 信号在活动模式和 DEEPSTOP 模式下均为输出 • 10：I/O 在活动模式和 DEEPSTOP 模式下均驱动低电平 • 11：I/O 在活动模式和 DEEPSTOP 模式下均驱动高电平
位 11:10	**IOCFG5[1:0]**：PA5 的驱动配置。 • 00：BYPASS 模式（默认） 在活动模式下，I/O 模式由 GPIO 模块寄存器控制，并在 DEEPSTOP 状态下切换至输入模式。 • 01：LCO 信号在活动模式和 DEEPSTOP 模式下均为输出 • 10：I/O 在活动模式和 DEEPSTOP 模式下均驱动低电平 • 11：I/O 在活动模式和 DEEPSTOP 模式下均驱动高电平
位 9:8	**IOCFG4[1:0]**：PA4 的驱动配置。 • 00：BYPASS 模式（默认） 在活动模式下，I/O 模式由 GPIO 模块寄存器控制，并在 DEEPSTOP 状态下切换至输入模式。 • 01：RTC_OUT 信号在活动模式和 DEEPSTOP 模式下均为输出 • 10：I/O 在活动模式和 DEEPSTOP 模式下均驱动低电平 • 11：I/O 在活动模式和 DEEPSTOP 模式下均驱动高电平
位 7:6	**IOCFG3[1:0]**：PA11 的驱动配置。 • 00：BYPASS 模式（默认） 在活动模式下，I/O 模式由 GPIO 模块寄存器控制，并在 DEEPSTOP 状态下切换至输入模式。 • 01：RTC_OUT 信号在活动模式和 DEEPSTOP 模式下均为输出

	• 10：I/O 在活动模式和 DEEPSTOP 模式下均驱动低电平 • 11：I/O 在活动模式和 DEEPSTOP 模式下均驱动高电平
位 5:4	**IOCFG2[1:0]**：PA10 的驱动配置。 • 00：BYPASS 模式（默认） 在活动模式下，I/O 模式由 GPIO 模块寄存器控制，并在 DEEPSTOP 状态下切换至输入模式。 • 01：LCO 信号在活动模式和 DEEPSTOP 模式下均为输出 • 10：I/O 在活动模式和 DEEPSTOP 模式下均驱动低电平 • 11：I/O 在活动模式和 DEEPSTOP 模式下均驱动高电平
位 3:2	**IOCFG1[1:0]**：PA9 的驱动配置。 • 00：BYPASS 模式（默认） 在活动模式下，I/O 模式由 GPIO 模块寄存器控制，并在 DEEPSTOP 状态下切换至输入模式。 • 01：LCO 信号在活动模式和 DEEPSTOP 模式下均为输出 • 10：I/O 在活动模式和 DEEPSTOP 模式下均驱动低电平 • 11：I/O 在活动模式和 DEEPSTOP 模式下均驱动高电平
位 1:0	**IOCFG0[1:0]**：PA8 的驱动配置。 • 00：BYPASS 模式（默认） 在活动模式下，I/O 模式由 GPIO 模块寄存器控制，并在 DEEPSTOP 状态下切换至输入模式。 • 01：RTC_OUT 信号在活动模式和 DEEPSTOP 模式下均为输出 • 10：I/O 在活动模式和 DEEPSTOP 模式下均驱动低电平 • 11：I/O 在活动模式和 DEEPSTOP 模式下均驱动高电平

5.6.16 调试寄存器（PWRC_DBGR）

该寄存器供调试特性使用。

偏移地址：0x84

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.	DEEPSTOP2
															rw

位 31:1	保留，必须保持复位值。
位 0	**DEEPSTOP2**：DEEPSTOP2 低功耗节能仿真使能。 • 0：应用正常 DEEPSTOP • 1：应用 DEEPSTOP2（调试器特性未丢失）而不是 DEEPSTOP

5.6.17 扩展状态和复位寄存器（PWRC_EXTSRR）

该寄存器提供与蓝牙活动开始和 DEEPSTOP 序列发生与否相关的标志。

偏移地址：0x88

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.	RFPHA SEF	DEEPSTOPF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
					rc_w1	rc_w1									

位 31:11	保留，必须保持复位值。
位 10	**RFPHASEF**：RFPHASE 标志。 • 0：BLE IP 不需要任何关注 • 1：BLE IP 已醒并可能需要系统关注。当发生无线电唤醒事件时，该位由硬件置 1 当 BLE IP 发出信息“准备睡眠”时，硬件将复位此位。 软件可通过写入 1 复位此位。
位 9	**DEEPSTOPF**：系统 DEEPSTOP 标志。 • 0：器件未进入 DEEPSTOP 模式 • 1：器件进入了 DEEPSTOP 模式 当 DEEPSTOP 序列发生时，该位由硬件置 1。软件可通过写入 1 复位此位。
位 8:0	保留，必须保持复位值。

5.7 PWRC 寄存器映射

请参见“表 3. BlueNRG-LP 存储器映射和外设寄存器边界地址”了解 BlueNRG-LP 中的 PWRC 基址位置。

注意： *在 DEEPSTOP 模式下，保留所有 PWRC 寄存器。粉红色单元格表示 VDD33 电源域中的位字段。这意味着即使在 SHUTDOWN 状态下，仍然应用相关特性（但是，在退出 SHUTDOWN 模式时会因为生成了 PORESETn 而丢失）。*

表 10. PWRC 寄存器映射

偏移	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
0x00	**PWRC_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.	Res.	Res.	Res.	APC	Res.	Res.	ENSDNBOR	LPMS
	复位值																												1			0	0
0x04	**PWRC_CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSILPMUFEN	ENTS	Res.	RAMRET3	RAMRET2	RAMRET1	Res.	PVDLS			PVDE
	复位值																						0	0	1	0	0	0		0	0	0	0
0x08	**PWRC_CR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EIWL	Res.	EWBLEHCPU	EWBLE	EWU11	EWU10	EWU9	EWU8	EWU7	EWU6	EWU5	EWU4	EWU3	EWU2	EWU1	EWU0
	复位值																	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	**PWRC_CR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUP11	WUP10	WUP9	WUP8	WUP7	WUP6	WUP5	WUP4	WUP3	WUP2	WUP1	WUP0
	复位值																					0	0	0	0	0	0	0	0	0	0	0	0
0x10	**PWRC_SR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IWUF	Res.	WBLEHCPUF	WBLEF	WUF11	WUF10	WUF9	WUF8	WUF7	WUF6	WUF5	WUF4	WUF3	WUF2	WUF1	WUF0
	复位值																	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	**PWRC_SR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IOBOOTVAL[3:0]				PVDO	Res.	REGMS	REGLPS	Res.	Res.	Res.	Res.	Res.	SMPSRDY	SMPSENR	SMPSBYPR
	复位值	0	1	1	1	1	0																										
0x18	保留																																

偏移	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
0x1C	**PWRC_CR5**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLKDETR_DISABLE	SMPS_ENA_DCM	NOSMPS	SMPSFBYP	SMPSLPOEN	Res.	Res.	SMPSBOMSEL[1:0]		SMPSLVL[3:0]			
	复位值																				0	0	0	0	0			0	1	0	1	0	0
0x20	**PWRC_PUCRA**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PUA15	PUA14	PUA13	PUA12	PUA11	PUA10	PUA9	PUA8	PUA7	PUA6	PUA5	PUA4	PUA3	PUA2	PUA1	PUA0
	复位值																	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1
0x24	**PWRC_PDCRA**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PDA15	PDA14	PDA13	PDA12	PDA11	PDA10	PDA9	PDA8	PDA7	PDA6	PDA5	PDA4	PDA3	PDA2	PDA1	PDA0
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	**PWRC_PUCRB**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PUB15	PUB14	PUB13	PUB12	PUB11	PUB10	PUB9	PUB8	PUB7	PUB6	PUB5	PUB4	PUB3	PUB2	PUB1	PUB0
	复位值																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x2C	**PWRC_PDCRB**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PDB15	PDB14	PDB13	PDB12	PDB11	PDB10	PDB9	PDB8	PDB7	PDB6	PDB5	PDB4	PDB3	PDB2	PDB1	PDB0
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	**PWRC_CR6**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EWU27	EWU26	EWU25	EWU24	EWU23	EWU22	EWU21	EWU20	EWU19	EWU18	EWU17	EWU16	EWU15	EWU14	EWU13	EWU12
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x34	**PWRC_CR7**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUP27	WUP26	WUP25	WUP24	WUP23	WUP22	WUP21	WUP20	WUP19	WUP18	WUP17	WUP16	WUP15	WUP14	WUP13	WUP12
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	**PWRC_SR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUF27	WUF26	WUF25	WUF24	WUF23	WUF22	WUF21	WUF20	WUF19	WUF18	WUF17	WUF16	WUF15	WUF14	WUF13	WUF12
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	保留																																
0x40	**PWRC_IOxCFG**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IOCFG7[1:0]		IOCFG6[1:0]		IOCFG5[1:0]		IOCFG4[1:0]		IOCFG3[1:0]		IOCFG2[1:0]		IOCFG1[1:0]		IOCFG0[1:0]	
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44 0x80	保留																																
0x84	**PWRC_DBGR**	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.	DEEPSTOP2
	复位值																																0

偏移	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
0x88	PWRC_EXTSRR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RFPHASEF	DEEPSTOPF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	复位值																						0	0									

5.8 编程模型

5.8.1 复位原因管理

导致 CPU 复位并执行其复位处理程序的原因有许多。下表提供了可帮助嵌入式软件获悉 CPU 复位的根本原因的标志总览。

表 11. 标志与 CPU 重启原因

	RCC_CSR					PWRC_ISCR（SYSCFG 中）	PWRC_EXTSRR
	LOCKUPRSTF	WDGRSTF	SFTRSTF	PORRSTF	PADRSTF	WAKEUP_ISC	DEEPSTOPF
POR/BOR 复位				1	1		
NRSTn 焊盘复位					1		
看门狗复位		1			1		
系统复位（CPU 请求）			1		1		
LOCKUP 复位	1				1		
发生唤醒事件时退出 DEEPSTOP						1	1
发生看门狗复位时退出 DEEPSTOP		1			1		
发生 NRSTn 焊盘复位时退出 DEEPSTOP					1		
发生 POR/BOR 时退出 DEEPSTOP				1	1		
退出 SHUTDOWN				1	1		

如果重启原因是从 DEEPSTOP 唤醒，则可以从 PWRC_SR1 寄存器读取唤醒源，如“表 12. 唤醒原因标志”所示。

表 12. 唤醒原因标志

	PWRC_SR1			
	IWUF	WBLEHCPUF	WBLEF	WUFx (x=8..11)
在发生 BLE 事件时唤醒			1	
在 MR_BLE 事件中通过主机定时器唤醒		1		
在发生 RTC 事件时唤醒	1			
通过 I/O 唤醒（PA0..PA15, PB0..PB11）				1

注意： *如果发生多个（使能的）唤醒事件，PWRC_SR1 中的多个位置 1。在发生唤醒事件（在 PWRC_CR3 寄存器中使能）后，唤醒标志置 1，PWRC_SR1 寄存器中的相关标志置 1，即使器件处于活动模式或处在睡眠退出序列（由另一个唤醒源发起）中。*

注意：*当系统请求 DEEPSTOP 模式时，如果唤醒标志已激活，则这些标志必须由知晓 DEEPSTOP 进入序列不可能发生的软件来清零。*

5.8.2 SMPS 输出电平重新编程

当一个以上的步骤用到了 SMPS 时，不能实时修改 SMPS 输出电压。当软件需要将 SMPS 输出电压重新设定为其他值时，必须遵循以下序列：

- 将 PWRC_CR5.SMPSBYP 置为 1
- 等待 PWRC_SR2.SMPSRDY = 0
- 在 PWRC_CR5.SMPSLVL[3:0]中设定新的目标值
- 将 PWRC_CR5.SMPSBYP 清零
- 等待 PWRC_SR2.SMPSRDY = 1

注意：当没有无线电活动/只有蓝牙传输正在进行时，必须启动此序列。

6 复位和时钟控制器（RCC）

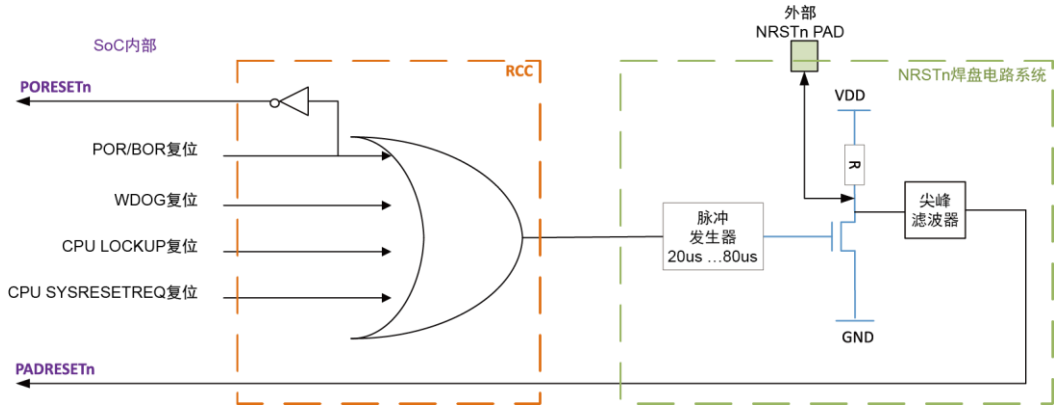
RCC 模块管理 BlueNRG-LP 器件的所有外设的时钟和复位生成。

6.1 复位管理

6.1.1 概述

图 12. 复位生成显示了复位生成的一般原理。

图 12. 复位生成



注意： *在 NRSTn 焊盘上输出系统复位信息，以便通知外部世界并将板件上的其他元件复位（如果需要）。*

设计中有两种不同的复位：

- PORESETn：此复位由 LPMU 模拟模块提供，对应于 POR 或 BOR 根本原因。它链接到电源电压上升或下降。

  PORESETn 复位影响器件的所有资源。

注意： *SHUTDOWN 退出相当于 POR/BOR 场景并生成 PORESETn。*

- PADRESETn（亦即系统复位）：此复位基于多个来源：
  - PORESETn
  - 看门狗复位
  - CPU LOCKUP 复位
  - CPU 软件系统复位
  - NRSTn 外部焊盘

注意： *系统复位被称为 PADRESETn 的原因是，当内部复位源被激活（看门狗、软件等）时，NRSTn 焊盘进行切换以将复位的发生通知外部世界。*

此系统复位将复位器件的所有资源，下面几项除外：

- 调试特性（SWD、测试寄存器等）
- Flash 控制器密钥管理部分
- RTC 定时器
- 电源控制器（PWRC）
- 部分 RCC 寄存器

脉冲发生器可确保每个内部复位源的复位脉冲都至少持续 20 µs。如果从 NRSTn 外部焊盘复位，会在焊盘置为低电平时生成复位脉冲。

6.1.2 电源复位

当器件供电电压低于阈值或当稳压器不提供目标电压时，PORESETn 信号激活。PORESETn 复位器件的所有资源。

6.1.3 看门狗复位

BlueNRG-LP 设备嵌入一个看门狗定时器，可用于软件崩溃后的恢复。参见第 18 节“独立看门狗（IWDG）”了解关于看门狗的使用和编程的详细信息。

6.1.4 LOCKUP 复位

Cortex-M0+生成 LOCKUP，以指示核心处于不可恢复的异常所导致的锁定状态。如果调试器连接到 Cortex-M0+，LOCKUP 复位会被屏蔽。在这种情况下，用户可使用 SWD 来复位或恢复代码。

6.1.5 系统复位请求

通过 Cortex-M0+的调试电路生成系统复位请求。调试器对“应用中断和复位控制寄存器”（AIRCR）的 SYSRESETREQ 位执行写操作。AIRCR 寄存器生成的该系统复位请求也可通过嵌入式软件来执行（例如在硬件故障处理程序中）。如需关于 Cortex-M0+系统控制和 ID 寄存器的更多信息，请参见《ARMv6-M 架构参考手册》§B3.2.2。

6.1.6 退出 DEEPSTOP

低功耗 DEEPSTOP 状态导致 1.2 V（名为 V12i 的电源域）的一部分和 3.3 V（VDDIO）关闭，同时 1.2 V 的其余部分维持 1 V 电压（名为 V12o 的电源域）。

当器件退出 DEEPSTOP 模式时，由于只有 V12i 电源域失去供电，因此只有该电源域复位。

6.2 时钟管理

可使用三个不同时钟源来驱动 BlueNRG-LP 中的系统时钟（CLK_SYS）：

- HSI：高速内部 64 MHz RC 振荡器（由 RC64MPLL 模拟模块提供），
- PLL64M：64 MHz PLL 时钟（由 RC64MPLL 模拟模块提供），
- HSE（高速外部）：
  - 高速 32 MHz 外部晶振
  - 或由单端 32 MHz 输入而非晶振提供。

BlueNRG-LP 器件还有一个被一些定时器（RTC、看门狗和 MR_BLE 无线电定时器）使用的低频时钟树。可对此低频时钟树使用四个不同时钟源：

- LSI：低速低漂移内部 RC，频率固定且范围为 24 kHz 至 49 kHz（取决于采样）。在本文档中，将其简称为 32 kHz 时钟。
- LSE：
  - 32.768 kHz 低速外部晶振。
  - 或由单端 32.768 kHz 输入而非晶振提供。
- 系统时钟 2048 分频。在这种情况下，在 DEEPSTOP 低功耗模式下慢速时钟可用。
- LSI_LPMU：LPMU 模拟模块使用的 32 kHz 时钟。

6.2.1 系统时钟详情

HSI 和 PLL64M 时钟由名为 RC64MPLL 的同一模拟模块提供。此模块输出的 64 MHz 时钟可以是：

- 非精确时钟（目标通常为 1%），当没有外部 XO 为此模块提供输入时钟时
- 精确时钟，当外部 XO 提供 32 MHz 时以及其内部 PLL 锁定后

注意： *对于蓝牙无线电操作，强制使用 PLL64M 或 HSE 作为时钟源（时钟需要高精度）。*

第 6.8 节“程序员模型”中显示了切换系统精确时钟的软件流程。此高频时钟源用于通过分频器生成器件的所有高频时钟。

复位后，CLK_SYS 经过四分频，为整个系统（CPU、DMA、存储器和外设）提供 16 MHz 时钟。

然后，软件可以设定以下列表中的另一个系统时钟频率：

- 1 MHz（当无线电或 ADC 正在使用时禁用）
- 2 MHz（当无线电或 ADC 正在使用时禁用）
- 4 MHz（当无线电或 ADC 正在使用时禁用）
- 8 MHz（当无线电正在使用时禁用）
- 16 MHz
- 32 MHz
- 64 MHz（当 I2S 正在使用时禁用）

注意： *禁用配置表示如果系统时钟以此频率运行，“正在使用”特性将无法工作。在设定 CLK_SYS 时必须特别注意，需符合一些限制条件：CLK_SYS 频率必须高于或等于 CLK_SYS_BLE。*

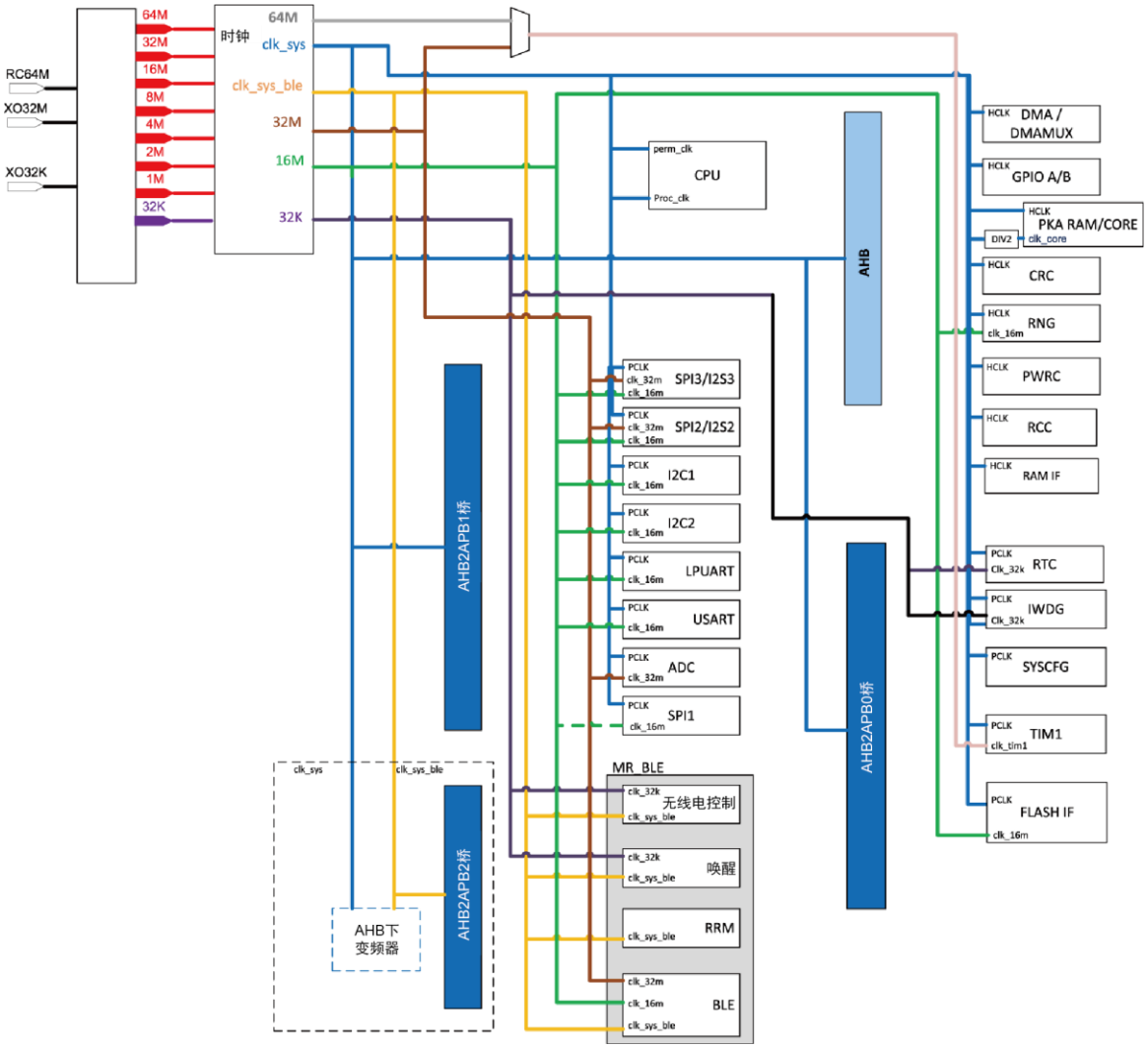
6.2.2 外设时钟详情

此高频时钟源还用于生成系统中的数个内部高频时钟：

- TIM1 内核时钟，是系统可达到的最高频率（在 RC64MPLL 配置中为 64 MHz，在 HSE 中为 32 MHz）。
- 固定的 32 MHz 时钟，由几个外设请求，如 MR_BLE 无线电 IP
- 固定的 16 MHz 时钟，由几个外设请求，如串行接口（在系统时钟从一个频率切换到另一个频率时维持固定波特率）或 Flash 控制器和 MR_BLE 无线电 IP（获得固定的参考时钟以管理延时）。

图 13. 外设时钟树总览显示了外设时钟树的情况总览。

图 13. 外设时钟树总览



大多数外设只使用系统时钟（CLK_SYS），以下外设除外：

- I2C、USART、LPUART：
  - 在使用系统时钟的同时，它们使用固定的 16 MHz 时钟为波特率管理提供固定的参考时钟。目的是允许 CPU 提高或降低系统时钟频率（取决于正在进行的活动），同时不影响外部 I/O 上可能正在进行的串行接口传输。
- SPI：
  - 在使用 I2S 模式时，通过固定的 16 MHz 或 32 MHz 时钟管理波特率

注意： *CPU/系统时钟频率必须等于或低于 I2S 时钟频率。*

- *在除 I2S 外的其他模式下运行时，通过系统时钟管理波特率。这意味着波特率受动态系统时钟频率变化的影响。*
- *RNG：*
  - *在使用系统时钟的同时，RNG 使用固定的 16 MHz 时钟以恒定频率生成随机数，不受系统时钟频率的影响。*
- *Flash 控制器：*
  - *在使用系统时钟的同时，Flash 控制器使用固定的 16 MHz 时钟生成 Flash 存储器需要的特定延时（例如在编程和擦除操作期间）*
- *PKA：*
  - *在使用系统时钟的同时，由于 PKA RAM 是单端口 RAM，PKA 使用频率为系统时钟的一半的时钟*
- *MR_BLE IP*
  - *MR_BLE IP 不直接对其 APB/AHB 接口使用系统时钟，而是使用可能经过分频（1 或 2 或 4）的系统时钟。表 13. CPU 与 MR_BLE 时钟依赖关系*
  - *在使用 CLK_SYS_BLE 的同时，MR_BLE 对调制器和解调器使用固定的 16 MHz 和 32 MHz 时钟，同时为管理特定延时提供了固定的参考时钟*

表 13. CPU 与 MR_BLE 时钟依赖关系

CLK_SYS	CLK_SYS_BLE
1 MHz/2 MHz/4 MHz/8 MHz	不可能使用 MR_BLE IP
16 MHz	16 MHz（CLKBLEDIV = 4）
32 MHz	• 16 MHz（CLKBLEDIV = 4） • 或 32 MHz（CLKBLEDIV = 2）
64 MHz	• 16 MHz（CLKBLEDIV = 4） • 或 32 MHz（CLKBLEDIV = 2）

- ADC
  - 在使用系统时钟的同时，ADC 使用以 16 MHz 频率运行的 64 MHz 预分频时钟（称为 CLKANA_ADC）

注意： *当使用 ADC 时，系统时钟必须以至少 8 MHz 的频率运行，才能在 ADC 采样因新采样而过载之前读取它们。*

- *此 CLKANA_ADC 经 2 或 4 分频后提供给 SMPS 模拟模块。*

注意： *为避免 ADC 的 SNR 劣化，SMPS 和 ADC 时钟必须同步*

6.2.3 慢速时钟频率详情

如时钟管理子章节的开头所述，有四个不同的时钟源可用于此低频时钟树：

- LSI：低速低漂移内部 RC，频率固定且范围为 24 kHz 至 49 kHz（取决于采样）。在本文档中，将其简称为 32 kHz 时钟。
- LSE：32.768 kHz 低速外部晶振（或单端输入频率）。

注意： *如果使用外部振荡器，PB12/PB13 I/O 会在 RCC_CR.LSEON 位置 1（GPIO_MODERX 配置过载）时自动连接此特性。*

***注意：**为了让特性正常工作，用户必须通过软件禁用 PB12/PB13 上的上/下拉。如果使用单端输入选项，则必须将 PB13 I/O 配置为数字输入模式。*

系统时钟 2048 分频。在这种情况下，在 DEEPSTOP 低功耗模式下慢速时钟不可用。

LSI_LPMU：LPMU 模拟模块使用的 32 kHz 时钟。一次只有一个源驱动整个低速时钟树。

注意： *默认在 PORESETn 后，所有低速源关闭。在 PADRESETn 后，慢速时钟配置是在 PADRESETn 之前设定的。*

DEEPSTOP 低功耗模式和唤醒时的慢速时钟激活和选择十分重要，它们为唤醒事件生成过程中涉及的定时器提供计时。

注意： *如果使用 LSI 配置，软件必须测量慢速时钟频率，以了解定时器使用的相关周期。慢速时钟测量特性在 MR_BLE IP 中可用。*

6.3 使用 MR_BLE 时的系统频率切换

CPU/系统时钟频率可以是 1 MHz 至 64 MHz，而 MR_BLE 时钟频率可以是 16 MHz 或 32 MHz。

在器件上使用无线电时，系统时钟频率的选择必须遵守一些规则：

- 系统时钟频率必须是 16 MHz、32 MHz 或 64 MHz，并高于或等于 MR_BLE 频率。其他选项会使无线电失灵。
- 为避免任何与无线电方案相冲突的风险，系统频率的更改必须通过 RCC_CSCMDR 寄存器机制来执行。

通过几个模块的协作来管理适当的系统频率切换：

- RCC（参见第 6.6.5 节“时钟切换指令寄存器（RCC_CSCMDR）”）
- AHBUPCONV 和 AHBDOWNCONV 模块（参见第 3 节“AHB 上/下变频器”）

使用此安全机制时，软件请求系统时钟频率更改，并且会在通过状态位（参见第 6.6.5 节“时钟切换指令寄存器（RCC_CSCMDR）”）和 CPU 上的相关中断线路（参见第 2.3.2 节“中断”）实际应用新频率时收到硬件发出的通知。

第 6.8.3 节“在 MR_BLE 使能时更改系统时钟频率”中描述了软件操作序列。

6.4 外部引脚上的时钟观测

可以在外部焊盘上输出一些内部时钟：

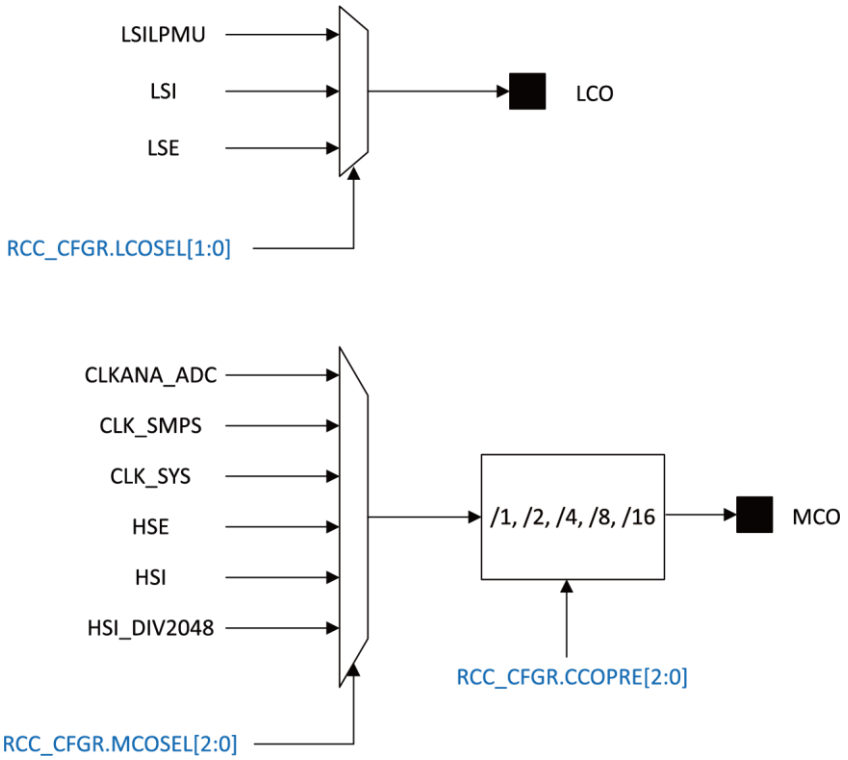
- 可在 LCOI/O 上输出低速时钟
- 可在 MCOI/O 上输出高速时钟

这可以通过在合适的复用功能中设定相关 I/O 来实现（参见表 6. GPIO 复用选项 AF0 - AF2）。

可通过 RCC 寄存器设定每个 I/O 的要输出的时钟选择（参见第 6.6.2 节“时钟配置寄存器（RCC_CFGR）”了解更多信息）。

图 14. LCO/MCO 输出时钟显示用来输出内部时钟的可能配置。

图 14. LCO/MCO 输出时钟



6.5 其他情况

6.5.1 IO BOOSTER

使用了一些模拟交换机，以便选择将由 ADC 使用的模拟 VINM/P 成对输入信号。

添加了 IO BOOSTER 模块，以便在 VBAT 低于阈值（2.7 V）时根据这些模拟交换机的指令提升电压，从而保证这些交换机的正确行为。在需要时，必须由软件通过 RCC_CFGR.IOBOOSTEN 位使能该模块。

6.6 RCC 寄存器说明

请参见“表 3. BlueNRG-LP 存储器映射和外设寄存器边界地址”了解 BlueNRG-LP 中的 RCC 基址位置。

绿色单元格表示寄存器位于 V12o 电源域中，粉红色单元格表示寄存器位于 VBAT（亦即 V33）电源域中。这意味着在 DEEPSTOP 退出时，这些寄存器不会复位。

6.6.1 时钟源控制寄存器（RCC_CR）

此寄存器控制不同时钟源（低速和高速）上的使能。

注意： *连接到高速时钟源的控制位在 PADRESETn 时复位。连接到低速时钟源的控制位只在 PORESETn 时复位（用表格脚注标识）。由于该寄存器位于 V12o 电源域中，其内容在从 DEEPSTOP 唤醒后不会被修改且系统时钟将恢复为进入 DEEPSTOP 模式之前的配置。*

偏移地址：0x00

复位值：0x0000 1400

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.	HSERD Y	HSEO N
														r	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	HSIPLL RDY	HSIPLL ON	HSEPL LBUFO N	Res.	HSIRDY	LOCKDET_NSTOP			LSEBY P	LSERD Y	LSEON	LSIRD Y	LSION	Res.	Res.
	r	rw	rw		r	rw	rw	rw	rw	r	rw	r	rw		

位 31:18	保留，必须保持复位值。
位 17	**HSERDY**：外部高速时钟标志。 此位由硬件置位，以指示 HSE 振荡器（32 MHz XO）稳定。 • 0：HSE 振荡器未准备就绪 • 1：HSE 振荡器准备就绪
位 16	**HSEON**：外部高速时钟使能。 软件必须将此位置 1 才能启动 XO 32 MHz，将其清零可停止 XO 32 MHz。 • 0：HSE 振荡器关闭 • 1：HSE 振荡器开启
位 15	保留，必须保持复位值。
位 14	**HSIPLLRDY**：内部高速时钟 PLL 标志。 此位由硬件置位，以指示 RC64MPLL pll 锁定。 • 0：RC64MPLL 上/下拉解锁 • 1：RC64MPLL 上/下拉锁定
位 13	**HSIPLLON**：内部高速时钟 PLL 使能。 软件必须将此位置 1 才能请求 HSE 上的 RC64MPLL 锁定，将其清零可停止锁定。 • 0：RC64MPLL PLL 关闭 • 1：RC64MPLL PLL 开启
位 12	**HSEPLLBUFON**：PLL RF2G4 的外部高速时钟缓冲区使能。 当使用无线电时，软件必须将此位置 1（以使 2.4 GHz PLL 工作）。 • 0：HSE PLL RF2G4 缓冲区关闭 • 1：HSE PLL RF2G4 缓冲区开启

	警告：当使用无线电时，此位必须置 1。将此位清零的唯一原因将是为该器件上不使用无线电的应用降低功耗。
位 11	保留，必须保持复位值。
位 10	**HSIRDY**：内部高速时钟标志。 此位由硬件置位，以指示内部 64 MHz RC 稳定。 • 0：内部 64 MHz RC 未准备就绪 • 1：内部 64 MHz RC 准备就绪
位 9:7	**LOCKDET_NSTOP**：为负责的锁定检测器模块的计数器定义时间窗目标，以便管理 HSIPLLRDY 信息（如果该时间窗内模拟锁定信号保持高电平且稳定，则 PLL 显示为锁定）。 定义时间窗目标的公式如下： 时间窗目标 = (LOCKDET_NSTOP + 1) x 64。
位 6(1)(2)	**LSEBYP**：外部低速时钟旁路。 当直接通过 SXTALI 引脚提供慢速时钟时，此位需置 1。 • 0：无 LSE 振荡器旁路 • 1：LSE 振荡器旁路使能
位 5(1)	**LSERDY**：外部低速时钟标志。 此位由硬件置位，以指示慢速时钟已启动。 • 0：LSE 振荡器未准备就绪 • 1：LSE 振荡器准备就绪 注意：无论选择哪种配置（外部 32 kHz 振荡器 -= LSEON 或者 SXTALI 上提供的外部时钟 = LSEBYP），此状态位均为真。
位 4(1)(2)	**LSEON**：外部低速时钟使能。 软件必须将此位置 1 才能启动 XO 32 kHz，将其清零可停止 XO 32 MHz。 • 0：LSE 振荡器关闭 • 1：LSE 振荡器开启
位 3(1)	**LSIRDY**：内部低速时钟标志。 此位由硬件置位，以指示内部低速 RC 稳定。 • 0：内部低速 RC 未准备就绪 • 1：内部低速 RC 准备就绪
位 2(1)	**LSION**：内部低速 RC 时钟使能。 软件必须将此位置 1 才能启动内部慢速时钟 RO，将其清零可停止内部慢速时钟 RO。 • 0：LSI RC 关闭 • 1：LSI RC 开启
位 1:0	保留，必须保持复位值。

1. *此位只在 PORESETn 时复位。*
2. *LSEBYP 和 LSEON 位不得同时使用。如果用户决定动态地更改外部 XO 和 SXTALI 上时钟注入之间的慢速时钟源，则为了复位 LSERDY 标志，它们必须确保 LSEON 和 LSEBYP 同时为低电平。*

6.6.2 时钟配置寄存器（RCC_CFGR）

注意： *连接到高速时钟源的控制位在 PADRESETn 时复位。链接到低速时钟源的控制位只在 PORESETn 时复位（用表格脚注标识）。*

偏移地址：0x08

复位值：0x0000 0440

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCOPRE[2:0]			MCOSEL[2:0]			LCOSEL[1:0]		SPI2I2 SCLKS EL	SPI3I2 SCLKS EL	Res.	Res.	Res.	Res.	IOBOO STEN	CLKSL OWSE L[1]
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
CLKSL OWSE L[0]	Res.	Res.	SMPS DIV	ANADIV		Res.	Res.	CLKSYSDIV[2:0]			Res.	Res.	STOPH SI	HSESE L	SMPSI NV
rw			rw	rw	rw			rw	rw	rw			rw	rw	rw

位 31:29	**CCOPRE**：可配置时钟输出预分频器。 • 000：CCO 时钟 1 分频 • 001：CCO 时钟 2 分频 • 010：CCO 时钟 4 分频 • 011：CCO 时钟 8 分频 • 100：CCO 时钟 1 6 分频 • 其他：保留 注意：如果在 IO 上的 MCO 输出使能时修改 CCOPRE[2:0]值，可能发生故障传播。
位 28:26	**MCOSEL**：主要可配置时钟输出选择。 • 000：MCO 输出已禁用。MCO 焊盘上无时钟 • 001：系统时钟 • 010：保留 • 011：HSI_64M = RC64MPLL 模块输出时钟（可以是内部 64 MHz 或 PLL 64 MHz 精确时钟） • 100：HSE（外部 32 MHz 振荡器） • 101：HSI_64M 2048 分频时钟 • 110：SMPS 时钟 • 111：ADC 时钟 注意：如果在 IO 上的 MCO 输出使能时修改 MCOSEL[2:0]值，可能发生故障传播。
位 25:24(1)	**LCOSEL**：低速可配置时钟输出选择。 • 00：LCO 输出已禁用。LCO 焊盘上无时钟 • 01：LSI_LPMU 时钟（内部 LPMU 慢速时钟源） • 10：LSI（内部慢速时钟 RC）时钟 • 11：LSE（外部 32 kHz） 注意：如果在 IO 上的 LCO 输出使能时修改 LCOSEL[1:0]值，可能发生故障传播。

位 23	**SPI2I2SCLKSEL**：为 SPI2 IP 选择 I2S 时钟。 • 0：16 MHz 外设时钟（默认） • 1：32 MHz 外设时钟 注意：I2S 时钟频率必须高于或等于系统时钟（通过 RCC_CFGR.CLKSYSDIV[2:0]位字段进行配置）。
位 22	**SPI2I3SCLKSEL**：为 SPI3 IP 选择 I2S 时钟。 • 0：16 MHz 外设时钟（默认） • 1：32 MHz 外设时钟 注意：I2S 时钟频率必须高于或等于系统时钟（通过 RCC_CFGR.CLKSYSDIV[2:0]位字段进行配置）。
位 21:18	保留，必须保持复位值。
位 17	IOBOOSTEN：IO BOOSTER 使能（参见第 6.5.1 节“IO BOOSTER”了解详情）。 • 0：IO BOOSTER 模块禁用 • 1：IO BOOSTER 模块使能。
位 16:15(1)	**CLKSLOWSEL**：低速时钟源选择。 • 00：LSILPMU 振荡器 • 01：LSE（外部振荡器）。在 DEEPSTOP 模式下可保留此源 • 10：LSI（内部 RC）。在 DEEPSTOP 模式下可保留此源 • 11：HSI_64M 2048 分频 注意：没有添加干扰机制，因此当用户更改慢速时钟源时，慢速时钟可能出现干扰。
位 14:13	保留，必须保持复位值。
位 12	**SMPSDIV**：SMPS 时钟分频因子。 • 0：SMPS 时钟频率为 8 MHz • 1：SMPS 时钟频率为 4 MHz
位 11:8	保留，必须保持复位值。
位 7:5	**CLKSYSDIV**：来自 HSI_64M 的系统时钟分频因子。 • 000：系统时钟频率为 64 MHz**（当 HSESEL=1 时不可用）** • 001：系统时钟 32 MHz • 010：系统时钟 16 MHz • 011：系统时钟 8 MHz* • 100：系统时钟 4 MHz* • 101：系统时钟 2 MHz* • 110：系统时钟 1 MHz* • 111：未使用 *：如果 RCC_APB2ENR.MRBLEEN 位置 1，则在 CLKSYSDIV 中写入的值（这些值当中的一个）会被在硬件层面写入的 010b = 16 MHz 取代。 警告： • 如果在 RCC_CFGR.HSESEL=1 时软件设定了 64 MHz 频率目标，硬件将再次切换 HSI64MPLL 上的系统时钟树（并在 RCC_CFGR.STOPHSI=1 时重启 HSIPLL64M 模拟模块）。 • 在使用 MR_BLE 时，为了在 64/32/16 MHz 之间无风险地切换系统频率，最好使用 RCC_CSCMDR 寄存器更改系统频率。 • 为了让无线电正常工作，MR_BLE 频率必须始终等于或低于 CPU/系统时钟。
位 4:3	保留，必须保持复位值。
位 2	**STOPHSI**：RC64MPLL 时钟源停止请求 • 0：RC64MPLL 使能（默认） • 1：请求了 RC64MPLL 禁用 注意：如果 CLKSYSDIV（从 RCC_CFGR 或 RCC_CSCMDR 寄存器）选择了 64 MHz 的频率，硬件将自动重启 RC64MPLL 模块并开启 RC64MPLL 时钟源。

位 1	**HSESEL**：时钟源选择请求。 • 0：请求了 RC64MPLL 时钟源（默认） 在这种情况下，快速时钟树的源由 RC64MPLL 模块提供。如果 HSI PLL 已锁定，则时钟可以是 HSI 或 PLL64M。 • 1：请求了直接 HSE 时钟源 在这种情况下，不使用 RC64MPLL 模块且系统时钟树的最大可用频率为 32 MHz。
位 0	**SMPSINV**：控制 SMPS 时钟的反相（与 ADC 时钟相比） • 0：SMPS 时钟不反相（默认） • 1：SMPS 时钟反相

1. 此位只在 PORESETn 时复位。

6.6.3 时钟中断使能寄存器（RCC_CIER）

此寄存器控制中断源上的使能。

此寄存器在 PADRESETn 时复位。

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
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.	WDGR STIE	RTCRS TIE	HSIPLL UNLOC KDETI E	HSIPLL RDYIE	HSERD YIE	HSIRD YIE	Res.	LSERD YIE	LSIRD YIE
							rw	rw	rw	rw	rw	rw		rw	rw

位 31:9	保留，必须保持复位值。
位 8	**WDGRSTIE**：看门狗复位释放中断使能。 • 0：禁用看门狗复位释放中断 • 1：使能看门狗复位释放中断
位 7	**RTCRSTIE**：RTC 复位释放中断使能。 • 0：禁用 RTC 复位释放中断 • 1：使能 RTC 复位释放中断
位 6	**HSIPLLUNLOCKDETIE**：HSI PLL 解锁检测中断使能。 • 0：禁用 HSI PLL 解锁检测中断 • 1：使能 HSI PLL 解锁检测
位 5	**HSIPLLRDYIE**：HSI PLL 就绪中断使能。 • 0：禁用 HSI PLL 锁定中断 • 1：使能 HSI PLL 锁定中断
位 4	**HSERDYIE**：HSE 就绪中断使能。 • 0：禁用 HSE 就绪中断 • 1：使能 HSE 就绪中断

位 3	**HSIRDYIE**：HSI 就绪中断使能。 • 0：禁用 HSI 就绪中断 • 1：使能 HSI 就绪中断
位 2	保留，必须保持复位值。
位 1	**LSERDYIE**：LSE 就绪中断使能。 • 0：禁用 LSE 就绪中断 • 1：使能 LSE 就绪中断
位 0	**LSIRDYIE**：LSI 就绪中断使能。 • 0：禁用 LSI 就绪中断 • 1：使能 LSI 就绪中断

6.6.4 时钟中断标志寄存器（RCC_CIFR）

该寄存器提供链接到时钟源是否就绪状态的状态标志。它还用于将标志清零。

此寄存器在 PADRESETn 时复位。

偏移地址：0x1C

复位值：0x0000 0008

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.	WDGR STF	RTCRS TF	HSIPLL UNLOC KDETF	HSIPLL RDYF	HSERD YF	HSIRD YF	Res.	LSERD YF	LSIRD YF.
							rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1		rc_w1	rc_w1

位 31:9	保留，必须保持复位值。
位 8	**WDGRSTF**：看门狗复位释放标志。 • 0：未发生看门狗复位释放事件 • 1：通过在此位写入 1 将发生的看门狗复位释放事件清零。 注意：由于异步慢速时钟/快速时钟管理的缘故，当软件通过写入 RCC_APB0RSTR.WDGRST 位请求释放看门狗复位时，复位释放仅在 APB 写入后的两个慢速时钟周期内有效。此中断允许在复位释放已完成时通知软件。 注意：此标志还在任意 PORESETn 后置位。
位 7	**RTCRSTF**：RTC 复位释放标志。 • 0：未发生 RTC 复位释放事件 • 1：发生了 RTC 复位释放事件。通过在此位写入 1 进行清零 注意：由于异步慢速时钟/快速时钟管理的缘故，当软件通过写入 RCC_APB0RSTR.RTCRST 位请求释放 RTC 复位时，复位释放仅在 APB 写入后的两个慢速时钟周期内有效。此中断允许在复位释放已完成时通知软件。 注意：此标志还在任意 PORESETn 后置位。
位 6	**HSIPLLUNLOCKDETF**：HSI PLL 解锁检测标志。 • 0：未发生 HSI PLL 解锁事件 • 1：发生了 HSI PLL 解锁事件。通过在此位写入 1 进行清零

位 5	**HSIPLLRDYF**：HSI PLL 就绪标志。 • 0：未发生 HSI PLL 锁定事件 • 1：发生了 HSI PLL 锁定事件。通过在此位写入 1 进行清零
位 4	**HSERDYF**：HSE 就绪标志。 • 0：未发生 HSE 就绪事件 • 1：发生了 HSE 就绪事件。通过在此位写入 1 进行清零
位 3	**HSIRDYF**：HSI 就绪标志。 • 0：未发生 LSI 就绪事件 • 1：发生了 LSI 就绪事件。通过在此位写入 1 进行清零
位 2	保留，必须保持复位值。
位 1	**LSERDYF**：LSE 就绪标志。 • 0：未发生 LSE 就绪事件 • 1：发生了 LSE 就绪事件。通过在此位写入 1 进行清零
位 0	**LSIRDYF**：LSI 就绪标志。 • 0：未发生 LSI 就绪事件 • 1：发生了 LSI 就绪事件。通过在此位写入 1 进行清零

6.6.5 时钟切换指令寄存器（RCC_CSCMDR）

该寄存器允许在 MR_BLE 活动时安全地切换 CPU/系统时钟频率。

请求频率时钟切换会使 MR_BLE 与系统其余部件之间的 AHB/APB 传输暂停，以便安全地执行时钟切换，并在新频率就绪后立即释放 AHB/APB 传输。

NVIC 上对 EOFSEQ_IRQ 信息使用专用中断线路（参见“表 5. 中断向量”）。

注意： *无论如何，当使用无线电时，用户必须确保 CPU/系统时钟频率至少为 16 MHz。*

此寄存器在 PADRESETn 时复位。

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
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.	EOFSE Q_IRQ	EOFSE Q_IE	STATUS[1:0]		CLKSYSDIV_REQ[2:0]			REQU EST
								rc_w1	rw	r	r	rw	rw	rw	rw

位 31:8	保留，必须保持复位值。
位 7	**EOFSEQ_IRQ**：序列结束标志。 • 0：未发生序列结束事件 • 1：发生了序列结束事件。通过在此位写入 1 进行清零
位 6	**EOFSEQ_IE**：序列结束中断使能。 • 0：禁用序列结束中断 • 1：使能序列结束中断

位 5:4	**STATUS**：切换序列的状态。 • 00：IDLE = 未请求/没有正在运行的切换序列 • 01：ONGOING = 系统时钟频率切换正在进行 • 10：DONE = 系统时钟频率切换已完成 • 11：保留 当 EOFSEQ_IRQ 位清零时，该位字段清零。
位 3:1	**CLKSYSDIV_REQ**：系统时钟请求/目标频率。 格式和注释/警告与第 6.6.2 节“时钟配置寄存器（RCC_CFGR）”中所述 SYSCLKDIV[2:0]位字段的相同。
位 0	**REQUEST**：请求切换系统时钟频率。 在此位写入 1 可请求系统时钟频率切换（使用 CLKSYSDIV_REQ[2:0]信息）。 当时钟频率切换完成时，此位由硬件清零。 注意：在此位写入 0 会中止频率切换序列（若尚未完成）。在应用程序的正常使用寿命中不得使用此操作，除非序列结束长时间未发生（为了消除这种情况），但这种情况不应发生。

6.6.6 AHB0 宏单元复位寄存器（RCC_AHBRSTR）

该寄存器允许通过软件将 AHB0 映射中的每个 IP 分别复位。此寄存器在 PADRESETn 时复位。

偏移地址：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.	Res.	Res.	Res.	RNGR ST	Res.	PKARS T.
													rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CRCR ST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIOB RST	GPIOA RST	Res.	DMAR ST
			rw									rw	rw		rw

位 31:19	保留，必须保持复位值。
位 18	**RNGRST**：RNG 复位。 • 0：RNG IP 不保持复位 • 1：RNG IP 保持复位
位 17	保留，必须保持复位值。
位 16	**PKARST**：PKA 复位。 • 0：PKA IP 不保持复位 • 1：PKA IP 保持复位 注意：当此位置 1 时，CPU 不再能够访问 PKA RAM。
位 12	**CRCRST**：CRC 复位。 • 0：CRC IP 不保持复位 • 1：CRC IP 保持复位
位 11:4	保留，必须保持复位值。
位 3	**GPIOBRST**：端口 B 的 IO 控制器复位。 • 0：GPIOB IP 不保持复位 • 1：GPIOB IP 保持复位

位 2	**GPIOARST**：端口 A 的 IO 控制器复位。 • 0：GPIOA IP 不保持复位 • 1：GPIOA IP 保持复位
位 1	保留，必须保持复位值。
位 0	**DMARST**：DMA 和 DMAMUX 复位。 • 0：DMA 和 DMAMUX IP 不保持复位 • 1：DMA 和 DMAMUX IP 保持复位

6.6.7 APB0 宏单元复位寄存器（RCC_APB0RSTR）

该寄存器允许通过软件将 APB0 映射中的每个 IP 分别复位。此寄存器在 PADRESETn 时复位。

注意： *每一位均由软件置位和复位。*

偏移地址：0x34

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.	WDGR ST	Res.	RTCRS T	Res.	Res.	Res.	SYSCF GRST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM1R ST
	rw		rw				rw								rw

位 31:15	保留，必须保持复位值。
位 14	**WDGRST**：看门狗复位。 • 0：看门狗 IP 不保持复位 • 1：看门狗 IP 保持复位 注意：由于异步慢速时钟/快速时钟管理的缘故，当软件通过写入 RCC_APB0RSTR.RTCRST 位请求释放 RTC 复位时，复位释放仅在 APB 写入后的两个慢速时钟周期内有效。此中断/状态标志可用于在复位释放已完成时通知软件。
位 13	保留，必须保持复位值。
位 12	**RTCRST**：RTC 复位。 • 0：RTC IP 不保持复位 • 1：RTC IP 保持复位 注意：由于异步慢速时钟/快速时钟管理的缘故，当软件通过写入 RCC_APB0RSTR.RTCRST 位请求释放 RTC 复位时，复位释放仅在 APB 写入后的两个慢速时钟周期内有效。此中断/状态标志可用于在复位释放已完成时通知软件。
位 11:9	保留，必须保持复位值。
位 8	**SYSCFGRST**：系统控制器复位。 • 0：系统控制器 IP 不保持复位 • 1：系统控制器 IP 保持复位
位 7:1	保留，必须保持复位值。
位 0	**TIM1RST**：TIM1 复位。 • 0：TIM1 IP 不保持复位 • 1：TIM1 IP 保持复位

6.6.8 APB1 宏单元复位寄存器（RCC_APB1RSTR）

该寄存器允许通过软件将 APB1 映射中的每个 IP 分别复位。此寄存器在 PADRESETn 时复位。

偏移地址：0x38

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.	I2C2RS T	Res.	I2C1RS T	Res.	Res.	Res.	Res.	Res.
								rw		rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SPI3R ST	Res.	SPI2R ST	Res.	USART RST	Res.	LPUAR TRST	Res.	Res.	Res.	ADCRS T	Res.	Res.	Res.	SPI1R ST
	rw		rw		rw		rw				rw				rw

位 31:24	保留，必须保持复位值。
位 23	**I2C2RST**：I2C2 复位。 • 0：I2C2 IP 不保持复位 • 1：I2C2 IP 保持复位
位 22	保留，必须保持复位值。
位 21	**I2C1RST**：I2C1 复位。 • 0：I2C1 IP 不保持复位 • 1：I2C1 IP 保持复位
位 20:15	保留，必须保持复位值。
位 14	**SPI3RST**：SPI3 复位。 • 0：SPI3 IP 不保持复位 • 1：SPI3 IP 保持复位
位 13	保留，必须保持复位值。
位 12	**SPI2RST**：SPI2 复位。 • 0：SPI2 IP 不保持复位 • 1：SPI2 IP 保持复位
位 11	保留，必须保持复位值。
位 10	**USARTRST**：USART 复位。 • 0：USART IP 不保持复位 • 1：USART IP 保持复位
位 9	保留，必须保持复位值。
位 8	**LPUARTRST**：LPUART 复位。 • 0：LPUART IP 不保持复位 • 1：LPUART IP 保持复位
位 7:5	保留，必须保持复位值。
位 4	**ADCRST**：ADC 复位。 • 0：ADC IP 不保持复位 • 1：ADC IP 保持复位

位 3:1	保留，必须保持复位值。
位 0	**SPI1RST**：SPI1 复位。 • 0：SPI1 IP 不保持复位 • 1：SPI1 IP 保持复位

6.6.9 APB2 宏单元复位寄存器（RCC_APB2RSTR）

该寄存器允许通过软件将 APB2 映射中的每个 IP 分别复位（无线电）。

此寄存器在 PADRESETn 时复位。

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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MRBLE RST
															rw

位 31:1	保留，必须保持复位值。
位 0	**MRBLERST**：MR_BLE（蓝牙无线电）复位。 • 0：MR_BLE IP 不保持复位 • 1：MR_BLE IP 保持复位

6.6.10 AHB0 宏单元时钟使能寄存器（RCC_AHBENR）

该寄存器允许通过软件将 AHB0 映射中的每个 IP 分别复位。此寄存器在 PADRESETn 时复位。

注意： *每个 IP 时钟门控只由 1 位控制，它在 IP 使用 AHB 时钟和内核时钟之一时对两者同时实施门控。*

偏移地址：0x50

复位值：0x0000 000C

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.	RNGREN	Res.	PKAEN
													rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CRCREN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIOBEN	GPIOAEN	Res.	DMAEN
			rw									rw	rw		rw

位 31:19	保留，必须保持复位值。
位 18	**RNGEN**：RNG 时钟使能。 • 0：RNG IP 有时钟门控 • 1：RNG IP 时钟使能

位 17	保留，必须保持复位值。
位 16	**PKAEN**：PKA 使能。 • 0：PKA IP 有时钟门控 • 1：PKA IP 时钟使能 注意：当此位置 1 时，CPU 不再能够访问 PKA RAM。
位 15:13	保留，必须保持复位值。
位 12	**CRCEN**：CRC 使能。 • 0：CRC IP 有时钟门控 • 1：CRC IP 时钟使能
位 11:4	保留，必须保持复位值。
位 3	**GPIOBEN**：端口 B 的 IO 控制器使能。 • 0：GPIOB IP 有时钟门控 • 1：GPIOB IP 时钟使能（默认）
位 2	**GPIOAEN**：端口 A 的 IO 控制器使能。 • 0：GPIOA IP 有时钟门控 • 1：GPIOA IP 时钟使能（默认）
位 1	保留，必须保持复位值。
位 0	**DMAEN**：DMA 和 DMAMUX 使能。 • 0：DMA 和 DMAMUX IP 有时钟门控 • 1：DMA 和 DMAMUX IP 时钟使能

6.6.11 APB0 宏单元时钟使能寄存器（RCC_APB0ENR）

该寄存器允许通过软件对 AHB0 映射中的每个 IP 分别实施门控。

注意： *每个 IP 时钟门控只由 1 位控制，它在 IP 使用 APB 时钟和内核时钟之一时对两者同时实施门控。*

该寄存器在 PADRESETn 时复位（用表格脚注标识的一位除外）。

偏移地址：0x54

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.	WDGE N	Res.	RTCEN	Res.	Res.	Res.	SYSCF GEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM1E N
	rw		rw				rw								rw

位 31:15	保留，必须保持复位值。
位 14	**WDGEN**：看门狗使能。 • 0：看门狗 IP 有时钟门控 • 1：看门狗 IP 时钟使能 警告：在此位置 1 后，由于慢速时钟上的双重再同步，在使用 IWDOG IP 前软件必须等待 2 个慢速时钟周期。
位 13	保留，必须保持复位值。
位 12(1)	**RTCEN**：RTC 使能。 • 0：RTC IP 有时钟门控 • 1：RTC IP 时钟使能 警告：在此位置 1 后，由于慢速时钟上的双重再同步，在使用 RTC IP 前软件必须等待 2 个慢速时钟周期。

位 11:9	保留，必须保持复位值。
位 8	**SYSCFGEN**：系统控制器使能。 • 0：系统控制器 IP 有时钟门控 • 1：系统控制器 IP 时钟使能
位 7:1	保留，必须保持复位值
位 0	**TIM1EN**：TIM1 使能 • 0：TIM1 IP 有时钟门控 • 1：TIM1 IP 时钟使能
位 1	保留，必须保持复位值。
位 0	**DMARST**：DMA 和 DMAMUX 复位。 • 0：DMA 和 DMAMUX IP 不保持复位 • 1：DMA 和 DMAMUX IP 保持复位

1. *此位只在 PORESETn 时复位。*

6.6.12 APB1 宏单元时钟使能寄存器（RCC_APB1ENR）

该寄存器允许通过软件对 APB1 映射中的每个 IP 分别实施门控。

注意： *每个 IP 时钟门控只由 1 位控制，它在 IP 使用 APB 时钟和内核时钟之一时对两者同时实施门控。*

此寄存器在 PADRESETn 时复位。

偏移地址：0x58

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.	I2C2EN	Res.	I2C1EN	Res.	Res.	Res.	Res.	Res.
								rw		rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SPI3E N	Res.	SPI2E N	Res.	USART EN	Res.	LPUAR TEN	Res.	Res.	ADCAN AEN	ADCDI GEN	Res.	Res.	Res.	SPI1E N
	rw		rw		rw		rw			rw	rw				rw

位 31:24	保留，必须保持复位值。
位 23	**I2C2EN**：I2C2 使能。 • 0：I2C2 IP 有时钟门控 • 1：I2C2 IP 时钟使能
位 22	保留，必须保持复位值。
位 21	**I2C1EN**：I2C1 使能。 • 0：I2C1 IP 有时钟门控 • 1：I2C1 IP 时钟使能
位 20:15	保留，必须保持复位值。
位 14	**SPI3EN**：SPI3 使能。 • 0：SPI3 IP 有时钟门控 • 1：SPI3 IP 时钟使能

位 13	保留，必须保持复位值。
位 12	**SPI2EN**：SPI2 使能。 • 0：SPI2 IP 有时钟门控 • 1：SPI2 IP 时钟使能
位 11	保留，必须保持复位值。
位 10	**USARTEN**：使能 USART。 • 0：USART IP 有时钟门控 • 1：USART IP 时钟使能
位 9	保留，必须保持复位值。
位 8	**LPUARTEN** LPUART 使能。 • 0：LPUART IP 有时钟门控 • 1：LPUART IP 时钟使能
位 7:6	保留，必须保持复位值。
位 5	ADC 模块模拟部分的 **ADCANAEN** ADC 时钟使能。 • 0：ADC 模拟 IP 有时钟门控 • 1：ADC 模拟 IP 时钟使能
位 4	ADC 模块数字部分的 **ADCDIGEN** ADC 时钟使能。 • 0：ADC 数字 IP 有时钟门控 • 1：ADC 数字 IP 时钟使能
位 3:1	保留，必须保持复位值。
位 0	**SPI1EN** SPI1 使能。 • 0：SPI1 IP 有时钟门控 • 1：SPI1 IP 时钟使能

6.6.13 APB2 宏单元时钟使能寄存器（RCC_APB2ENR）

该寄存器允许通过软件对 APB2 映射（无线电）中的 MR_BLE 时钟实施门控，并设定 MR_BLE IP 要使用的频率。

注意： *对 MR_BLE 实施门控指的是 MR_BLE 快速和慢速时钟树（因而包括 WAKEUP 模块）。*

此寄存器在 PADRESETn 时复位。

偏移地址：0x60

复位值：0x0000 0002

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.	CLKBLEDIV		MRBLE EN
													rw	rw	rw

位 31:22	保留，必须保持复位值。
位 21	**CLKBLEDIV**：RCC_APB2ENR.MRBLEEN=1 时的 MR_BLE（Bluetooth® Low Energy 无线电）时钟频率选择。 • 00：保留 注意：通过软件写入“00”被在硬件中写入“01”取代。 • 01：32 MHz • 10：16 MHz • 11：保留 警告： • 为了让无线电正常工作，MR_BLE 频率必须始终等于或低于 CPU/系统时钟。 • 在修改系统时钟频率与 MR_BLE 频率之比时，AHBUPCONV 模块必须调整 APB/AHB 总线上的时钟比。只管理动态 CPU 系统时钟切换（参见第 6.6.5 节“时钟切换指令寄存器（RCC_CSCMDR）”）。 • 因此，强烈建议使用静态 MR_BLE 时钟配置。
位 0	**MRBLEEN**：MR_BLE（Bluetooth® Low Energy 无线电）使能。 • 0：MR_BLE IP 有时钟门控 • 1：MR_BLE IP 时钟使能

6.6.14 V33 复位状态寄存器（RCC_CSR）

此寄存器提供复位原因标志。当发生任何新的复位事件时，硬件自动将其置 1 且必须通过软件清零。

表 12. 唤醒原因标志提供了活动标志与复位原因的汇总。

此寄存器在 PORESETn 时复位。

偏移地址：0x94

复位值：0x0C00 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	LOCKU PRSTF	WDGR STF	SFTRS TF	PORR STF	PADRS TF	Res.	Res.	RMVF	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	r	r	r	r	r			rc_w1							
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	保留，必须保持复位值。
位 30	**LOCKUPRSTF**：CPU 锁定复位标志。 由硬件在发生 CPU 锁定复位时置位。通过在 RMVF 位写入 1 进行复位。
位 29	**WDGRSTF**：看门狗复位标志。 由硬件在发生看门狗复位时置位。通过在 RMVF 位写入 1 进行复位。
位 28	**SFTRSTF**：软件复位标志。 由硬件在发生 CPU 系统复位时置位。通过在 RMVF 位写入 1 进行复位。
位 27	**PORRSTF**：上电复位标志。 由硬件在发生 PORESETN 或 BOR 复位时置位。通过在 RMVF 位写入 1 进行复位。
位 26	**PADRSTF**：NRSTn 焊盘复位标志。 在发生从外部 NRSTn 焊盘复位时和在任何复位后由硬件置位。这表示仅当除该标志外的所有标志均为低电平时，复位源是 NRSTn 焊盘。通过在 RMVF 位写入 1 进行复位。
位 25:24	保留，必须保持复位值。
位 23	**RMVF**：去除复位标志。 在此位写入 1 会将该寄存器的所有复位标志清零。此位由硬件自动清零。
位 22:0	保留，必须保持复位值。

6.6.15 RF 软件高速外部寄存器 (RCC_RFSWHSECR)

此寄存器在 PADRESETn 时复位。

偏移地址：0x98

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.	SWXOTUNE[5:0]						SWXO TUNEE N	GMC[2:0]			SATRG	Res.	Res.	Res.
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			

位 31:14	保留，必须保持复位值。
位 13:8	**SWXOTUNE**：由软件设置的 RF-HSE 电容器组调谐值。 如果 SWXOTUNEEN 位置 1，将考虑该值而不是复位时硬件加载的修正值。
位 7	**SWXOTUNEEN**：RF HSE 软件电容器组调谐使能。 • 0：RCC_RFHSECR.XOTUNE[5:0]位字段中可读的修正值被用作 HSE 上的修正值 • 1：写入 RCC_RFSWHSECR.SWXOTUNE[5:0]位字段的修正值被用作 HSE 上的修正值
位 6:4	**GMC**：高速外部 IO 电流控制。 – 000：最大 0.18 mA/V – 001：最大 0.57 mA/V – 010：最大 0.78 mA/V – 011：最大 1.13 mA/V – 100：最大 0.61 mA/V – 101：最大 1.65 mA/V – 110：最大 2.12 mA/V – 111：最大 2.84 mA/V 注意：该值只通过软件设置。
位 3	**SATRG**：感应放大器阈值。 • 0：偏置电流与参考电流之比为 1/2 • 1：偏置电流与参考电流之比为 3/4
位 2:0	保留，必须保持复位值。

6.6.16 Rf 高速外部寄存器（RCC_RFHSECR）

此寄存器在 PADRESETn 时复位。

偏移地址：0x9C

复位值：0x0000 0000（当 BlueNRG-LP Flash 为空时），否则取决于采样中刷写的修正值。

注意： *XOTUNE 值取决于外部 XO 元件的选择。因此，需要在启动 XO 之前设定和选择 RCC_RFSWHSECR.SWXOTUNE 位字段。*

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.	XOTUNE[5:0]					
										r	r	r	r	r	r

位 31:6	保留，必须保持复位值。
位 5:0	**XOTUNE**：RF-HSE 电容器组调谐。 Flash 控制器读取 Flash 存储器中的信息后，由硬件在复位时加载该值。

6.7 RCC 寄存器映射

请参见“表 3. BlueNRG-LP 存储器映射和外设寄存器边界地址”了解 BlueNRG-LP 中的 RCC 基址位置。

绿色单元格表示寄存器位于 V12o 电源域中，粉红色单元格表示寄存器位于 VBAT（亦即 V33）电源域中。这意味着在 DEEPSTOP 退出时，这些寄存器不会复位。

表 14. RCC 寄存器映射和复位值

偏移	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
0x00	RCC_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSERDY	HSEON	Res.	HSIPLLRDY	HSIPLLON	HSEPLLBUFON	Res.	HSIRDY	LOCKDET_NSTOP			LSEBYP	LSERDY	LSEON	LSIRDY	LSION	Res.	Res.
	复位值															0	0		0	0	1		1	0	0	0	0	0	0	0	0		
0x04	保留																																
0x08	RCC_CFGR	CCOPRE			MCOSEL			LCOSEL		SPI2I2SCLKSEL	SPI3I2SCLKSEL	Res.	Res.	Res.	Res.	IOBOOSTEN	CLKSLOWSEL		Res.	Res.	SMPSDIV	ANADIV (11:10)		Res.		CLKSYSDIV			Res.	Res.	STOPHSI	HSESEL	SMPSINV
	复位值	0	0	0	0	0	0	0	0	0	0					0	0	0			0	0	0		0	0	1	0			0	0	0
0x0C-014	保留																																

偏移	0x18		0x1C		0x20		0x24-0x2C
寄存器	RCC_CIER	复位值	RCC_CIFR	复位值	RCC_CSCMDR	复位值	
31	Res.		Res.		Res.		保留
30	Res.		Res.		Res.		
29	Res.		Res.		Res.		
28	Res.		Res.		Res.		
27	Res.		Res.		Res.		
26	Res.		Res.		Res.		
25	Res.		Res.		Res.		
24	Res.		Res.		Res.		
23	Res.		Res.		Res.		
22	Res.		Res.		Res.		
21			Res.		Res.		
20			Res.		Res.		
19			Res.		Res.		
18			Res.		Res.		
17			Res.		Res.		
16			Res.		Res.		
15	Res.		Res.		Res.		
14	Res.		Res.		Res.		
13	Res.		Res.		Res.		
12	Res.		Res.		Res.		
11	Res.		Res.		Res.		
10	Res.		Res.		Res.		
9	Res.		Res.		Res.		
8	WDGRSTIE	0	WDGRSTIF	0	Res.		
7	RTCRSTIE	0	RTCRSTIF	0	EOFSEQ_IRQ	0	
6	HSIPLLUNLOCKDETIE	0	HSIPLLUNLOCKDETF	0	EOFSEQ_IE	0	
5	HSIPLLRDYIE	0	HSIPLLRDYF	0	STATUS[1:0]	0	
4	HSERDYIE	0	HSERDYF	0		0	
3	HSIRDYIE	0	HSIRDYF	1	CLKSYSDIV_REQ[2:0]	0	
2	Res.		Res.				
1	LSERDYIE	0	LSERDYF	0	REQUEST	0	
0	LSIRDYIE	0	LSIRDYF	0		0	

偏移	0x40	0x3C	0x38		0x34		0x30	
寄存器	RCC_APB2RSTR		RCC_APB1RSTR		RCC_APB0RSTR		RCC_AHBRSTR	
	复位值		复位值		复位值		复位值	
31	Res.	保留	Res.		Res.		Res.	
30	Res.		Res.		Res.		Res.	
29	Res.		Res.		Res.		Res.	
28	Res.		Res.		Res.		Res.	
27	Res.		Res.		Res.		Res.	
26	Res.		Res.		Res.		Res.	
25	Res.		Res.		Res.		Res.	
24	Res.		Res.		Res.		Res.	
23	Res.		I2C2RST	0	Res.		Res.	
22	Res.		Res.		Res.		Res.	
21	Res.		I2C1RST	0	Res.		Res.	
20	Res.		Res.		Res.		Res.	
19	Res.		Res.		Res.		Res.	
18	Res.		Res.		Res.		RNGRST	0
17	Res.		Res.		Res.		Res.	
16	Res.		Res.		Res.		PKARST	0
15	Res.		Res.		Res.		Res.	
14	Res.		SPI3RST	0	WDGRST	0	Res.	
13	Res.		Res.		Res.		Res.	
12	Res.		SPI2RST	0	RTCRST	0	CRCRST	0
11	Res.		Res.		Res.		Res.	
10	Res.		USARTRST	0	Res.		Res.	
9	Res.		Res.		Res.		Res.	
8	Res.		LPUARTRST	0	SYSCFGRST	0	Res.	
7	Res.		Res.		Res.		Res.	
6	Res.		Res.		Res.		Res.	
5	Res.		Res.		Res.		Res.	
4	Res.		ADCRST	0	Res.		Res.	
3	Res.		Res.		Res.		GPIOBRST	0
2	Res.		Res.		Res.		GPIOARST	0
1	Res.		Res.		Res.		Res.	0
0	MRBLERST		SPI1RST	0	TIM1RST	0	DMARST	0

偏移	0x0																															
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
0x0	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.
复位值																																
0x4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDGEN	Res.	RTCEN	Res.	Res.	Res.	SYSCFGEN	Res.	Res.	Res.		GPIOBEN	GPIOAEN	Res.	DMAEN
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
0x8	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	I2C2EN	Res.	I2C2EN	Res.	Res.	Res.	Res.	Res.	Res.	SPI3EN	Res.	SPI2EN	Res.	USARTFN	Res.	LPUARTEN	Res.	Res.	ADCANAFN	ADCDIGFN	Res.	Res.	Res.	SPI1EN
复位值									0		0							0		0		0		0			0	0				0
0xC	保留																															
0x10	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.	CLKBLEDIV		MRBLEEN
复位值																														0	1	0
0x14-0x40	保留																															
0x44-0x4C	保留																															
0x50	RCC_AHBENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RNGEN	Res.	PKAEN	Res.	Res.	Res.	CRCEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIOBEN	GPIOAEN	Res.	DMAEN
复位值														0		0				0									1	1	0	0
0x54	RCC_APB0ENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDGEN	Res.	RTCEN	Res.	Res.	Res.	SYSCFGEN	Res.	Res.	Res.					
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x58	RCC_APB1ENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	I2C2EN	Res.	I2C2EN	Res.	Res.	Res.	Res.	Res.	SPI3EN	Res.	SPI2EN	Res.	USARTFN	Res.	LPUARTEN	Res.	Res.	ADCANAFN	ADCDIGFN	Res.	Res.	Res.	SPI1EN
复位值									0		0							0		0		0		0			0	0				0
0x5C	保留																															
0x60	RCC_APB2ENR	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.	CLKBLEDIV		MRBLEEN
复位值																														0	1	0

偏移	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
0x64-0x90		保留																															
0x94	RCC_CSR	Res.	LOCKUPRSTF	WDGRSTF	SFTRSTF	PORRSTF	PADRSTF	Res.	Res.	RMVF	Res.	Res.	Res.	Res.	Res	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	复位值		0	0	0	1	1			0																							
0x98	RCC_RFSWHSEC R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWXOTUNE						SWXOTUNEEN		GMC		SATRG	Res.	Res.	Res.
	复位值																			0	0	0	0	0	0	0	0	1	1	0			
0x9C	RCC_RFHSECR	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.	XOTUNE					
	复位值																											-	-	-	-	-	-

6.8 编程模型

6.8.1 将系统时钟切换到 PLL64M 时钟树

为了将系统从 HSI 时钟源切换到 PLL64M 时钟源，用户必须：

1. 使能 HSE（32 MHz 外部晶振）
2. 等待 HSE 就绪标志信息（通过中断或轮询）
3. 请求使能 PLL
4. 等待 PLL 就绪标志信息（通过中断或轮询）。从这一刻开始，整个快速时钟树的时钟源均为精确 PLL64M 时钟源。

注意：状态标志和相关中断可用于在发生 HSIPLL64M 解锁事件时通知软件。参见 RCC_CIER 和 RCC_CIFR 寄存器。

6.8.2 使用直接的 HSE（高速外部时钟）代替 RC64MPLL 模块

如果应用程序不打算使用 64 MHz 系统时钟频率，可将系统配置为直接使用外部 XO（HSE）提供的 32 MHz。

此配置选择应为静态且应在绝不使用 64 MHz 时使用。在这种情况下，软件必须：

1. 确保 RCC_CR.CLKSYSDIV 位字段设定的系统频率低于 64 MHz
2. 使能外部 XO（通过将 RCC_CR.HSEON 置 1——若尚未执行）
3. 等待 HSE 就绪标志信息（通过中断或轮询）
4. 将 RCC_CFGR.HSESEL 位置 1 可切换 HSE 路径上的快速时钟树。如果有两个时钟（HSI 和 HSE），应需要约 4 个时钟周期完成切换
5. 为了节约电力，软件可通过将 RCC_CFGR.STOPHSI 位置 1 来停止 RC64MPLL 模拟模块。

注意：提供了硬件机制以重启 RC64MPLL，并在 HSERDY 为低电平或 CLKSYSDIV 位字段已设定为请求 64 MHz 时切换回其时钟树。

注意：执行 DEEPSTOP 序列时 HSE 配置未丢失。因此，在唤醒时，系统重启 HSE（借助于 HEON 位）。但是，软件必须将系统时钟切换回 HSI 或 PLL64M 路径，才能进入 DEEPSTOP/SHUTDOWN 模式，因此在唤醒时：

- RC64MPLL 模块上运行时钟树（HSI 或 PLL64M）
- 软件必须重新设定 HSE 模式。但是，当 MR_BLE 已触发唤醒事件并启动了其序列时，必须避免 HSI 与 HSE 之间的时钟树切换。软件必须预估 CPU 唤醒，以确保在无线电开始任何活动前恢复最终时钟源。

6.8.3 在 MR_BLE 使能时更改系统时钟频率

当 MR_BLE 使能（通过将 RCC_APB2ENR.MRBLEEN 置 1）时，应用软件无法保证无线电正在运行或即将启动一个序列以使 MR_BLE IP 执行对 RAM 的 AHB 访问。若管理不当，更改系统时钟并通过此操作更改 MR_BLE 时钟域与系统时钟域之比可能导致崩溃。

因此，实施了一种硬件机制，当 MR_BLE 开启时，必须使用该机制更改系统频率。

执行系统时钟更改的操作序列如下：

1. 确保目标频率高于或等于 MR_BLE 频率（显示在 RCC_APB2ENR.CLKBLEDIV[1:0]中）
2. 在 RCC_CSCMDR.CLKSYDIV 位字段中设定所需频率并将 RCC_CSCMDR.REQUEST 位置 1
3. 等待 RCC_CSCMDR.EOFSEQ_IRQ 标志信息（通过中断或轮询）
4. 当标志（和中断（若使能））置位时，系统正以新的频率运行
5. RCC_CFGR.CLKSYSDIV[1:0]位字段已由硬件更新至新的频率值。

注意：如果软件请求的频率低于 16 MHz，则当前最终频率为 16 MHz（相关值可在 RCC_CFGR.CLKSYSDIV[1:0]中读取）。当 MR_BLE 锁定在 32 MHz 时，如果软件请求的频率为 16 MHz，则使用所需频率但无线电方案不再起作用。

7 通用 I/O（GPIO）

7.1 引言

每个通用 I/O 端口包括 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 和 GPIOx_ODR）和 1 个 32 位置位/复位寄存器 (GPIOx_BSRR)。此外，所有 GPIO 都包括 1 个 32 位锁定寄存器 (GPIOx_LCKR) 和 2 个 32 位复用功能选择寄存器（GPIOx_AFRH 和 GPIOx_AFRL）。

7.2 GPIO 主要特性

- 输出状态：推挽或开漏 + 上拉/下拉
- 从输出数据寄存器（GPIOx_ODR）或外设（复用功能输出）输出数据
- 可为每个 I/O 选择不同的速度
- 输入状态：浮空、上拉/下拉、模拟
- 将数据输入到输入数据寄存器（GPIOx_IDR）或外设（复用功能输入）
- 置位和复位寄存器（GPIOx_BSRR），对 GPIOx_ODR 具有按位写权限
- 锁定机制（GPIOx_LCKR），可冻结 I/O 端口配置
- 模拟功能
- 复用功能选择寄存器
- 快速翻转，每个时钟周期翻转一次
- 引脚复用非常灵活，允许将 I/O 引脚用作 GPIO 或多种外设功能中的一种

7.3 GPIO 功能说明

根据第 4 节“I/O 工作模式”中列出的每个 I/O 端口的特定硬件特性，通用 I/O（GPIO）端口的每个端口位可由软件分别配置为几种模式：

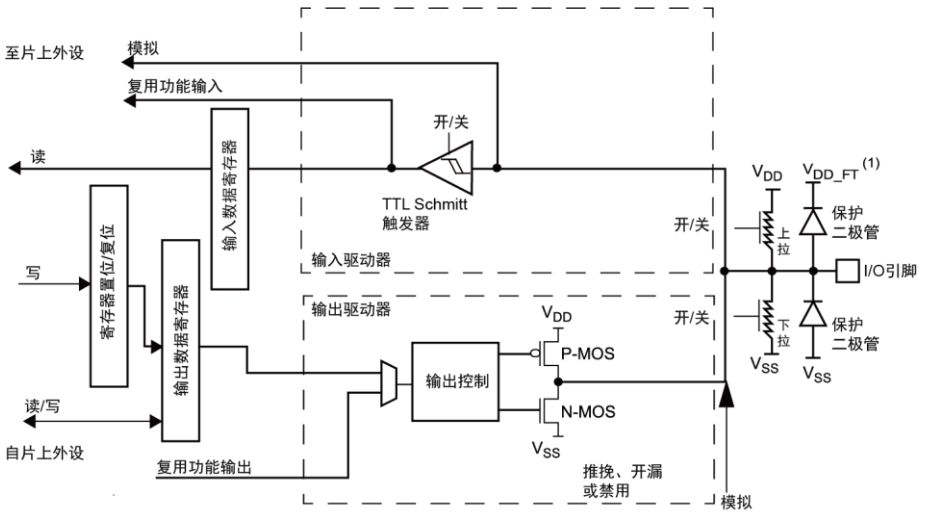
- 输入浮空
- 输入上拉
- 输入下拉
- 模拟
- 具有上拉或下拉功能的开漏输出
- 具有上拉或下拉功能的推挽输出
- 具有上拉或下拉功能的复用功能推挽
- 具有上拉或下拉功能的复用功能开漏

每个 I/O 端口位均可自由编程，但 I/O 端口寄存器必须按 32 位字、半字或字节进行访问。GPIOx_BSRR 寄存器和 GPIOx_BRR 寄存器旨在实现对 GPIOx_ODR 寄存器进行原子读取/修改访问。这样便可确保在读取和修改访问之间发生中断请求也不会有问题。

注意： *开漏和模拟特性在 BlueNRG-LP 的所有 I/O 上均不可用。请参见“表 7. GPIO 复用选项 AF3 - AF4”和“表 8. I/O 模拟特性映射”脚注。*

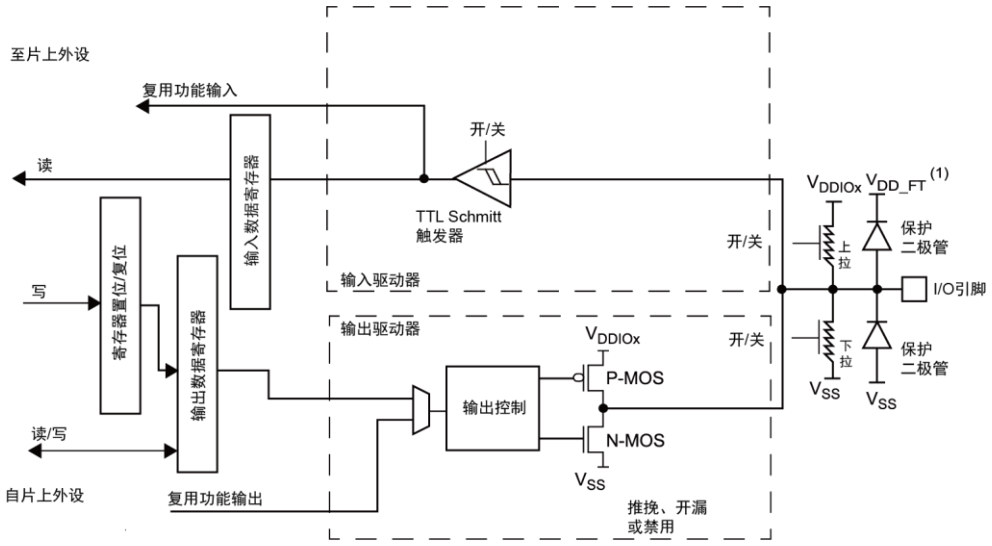
图 15. 混合模拟/数字 5 V 容限 I/O 端口位的基本结构和“图 16. 仅数字 5 V 容限 I/O 端口位的基本结构”分别展示了混合模拟/数字 5 V 容限 I/O 端口位和仅数字 5 V 容限 I/O 端口位的基本结构。表 15. 端口位配置给出了可能的端口位配置。

图 15. 混合模拟/数字 5 V 容限 I/O 端口位的基本结构



1. V_{DD_FT}可能特定于5 V容限I/O且不同于V_{DD}。

图 16. 仅数字 5 V 容限 I/O 端口位的基本结构



1. V_{DD_FT}可能特定于5 V容限I/O且不同于V_{DD}。

注意：　GP = 通用、PP = 推挽、PU = 上拉、PD = 下拉、OD = 开漏、AF = 复用功能

表 15. 端口位配置

MODE(i) [1:0]	OTYPER(i)	OSPEED(i) [1:0]		PUPD(i) [1:0]		I/O 配置	
01	0	SPEED [1:0]		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 [1:0]		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（输入浮空）	
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		

注意： *开漏和模拟特性在 BlueNRG-LP 的所有 I/O 上均不可用。请参见“表 7. GPIO 复用选项 AF3 - AF4”和“表 8. I/O 模拟特性映射”。*

7.3.1 通用 I/O（GPIO）

在复位过程中及之后，复用功能不活跃且大多数 I/O 端口配置为 GPIO 输入上拉模式，但 SWD 调试引脚除外。复位后，调试引脚处于 AF0 上拉/下拉状态：

- PA2：SWDIO 处于上拉状态
- PA3：SWDCLK 处于下拉状态

当引脚配置为输出后，写入到输出数据寄存器 (GPIOx_ODR) 的值将在 I/O 引脚上输出。可以在推挽模式下或开漏模式下使用输出驱动器（仅驱动低电平，高电平为高阻态）。

输入数据寄存器 (GPIOx_IDR) 每隔 1 个 AHB 时钟周期捕获一次 I/O 引脚的数据。

所有 GPIO 引脚都具有内部弱上拉及下拉电阻，可根据 GPIOx_PUPDR 寄存器中的值来打开/关闭。

7.3.2 I/O 引脚复用功能复用器和映射

器件 I/O 引脚通过一个复用器连接到板载外设/模块，该复用器一次仅允许一个外设的复用功能 (AF) 连接到 I/O 引脚。这可以确保共用同一个 I/O 引脚的外设之间不会发生冲突。

每个 I/O 引脚都有一个复用器，该复用器采用多达 8 路复用功能输入（AF0 到 AF7），可通过 GPIOx_AFRL（针对引脚 0 到 7）和 GPIOx_AFRH（针对引脚 8 到 15）寄存器对这些输入进行配置：

- 复位后，复用器选择为复用功能 0 (AF0)。在复用功能模式下通过 GPIOx_MODER 寄存器配置 I/O。
- 第 4 节“I/O 工作模式”中详述了每个引脚的特定复用功能分配。

除了这种灵活的 I/O 复用架构之外，各外设还可以将复用功能映射到不同 I/O 引脚，这可以优化小型封装中可用外设的数量。

为了以给定配置使用 I/O，用户必须进行如下操作：

- **调试功能：**每个器件复位后，立即将这些引脚分配为可由调试主机使用的复用功能引脚
- **系统功能：**MCO 和 LCO 引脚必须配置为复用功能模式
- **GPIO：**在 GPIOx_MODER 寄存器中将所需 I/O 配置为输出、输入或模拟通道
- **外设复用功能：**
  - 在 GPIOx_AFRL 或 GPIOx_AFRH 寄存器中，将 I/O 连接到所需的 AFx。
  - 通过 GPIOx_OTYPER、GPIOx_PUPDR 和 GPIOx_OSPEEDER 寄存器，分别选择类型、上拉/下拉以及输出速度。
  - 在 GPIOx_MODER 寄存器中将所需 I/O 配置为复用功能。
- **其他功能：**
  - 对于 ADC，在 GPIOx_MODER 寄存器中将所需 I/O 配置为模拟模式，并在 ADC 寄存器中配置需要的功能

注意： *用户必须通过软件禁用配置为模拟模式的 I/O 的上/下拉。*

对于其他功能，如唤醒 I/O 和 LSE 振荡器，在相关的 PWRC 和 RCC 寄存器中配置所需功能。这些功能优先于标准 GPIO 寄存器中的配置。

请参见表 6. GPIO 复用选项 AF0 - AF2 和表 7. GPIO 复用选项 AF3 - AF4 中复用功能 I/O 引脚的具体映射。

7.3.3 I/O 端口控制寄存器

每个 GPIO 端口有 4 个 32 位存储器映射的控制寄存器（GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR），可配置多达 16 个 I/O。GPIOx_MODER 寄存器用于选择 I/O 模式（输入、输出、AF、模拟）。GPIOx_OTYPER 和 GPIOx_OSPEEDR 寄存器用于选择输出类型（推挽或开漏）和速度。无论采用哪种 I/O 方向，GPIOx_PUPDR 寄存器都用于选择上拉/下拉。

7.3.4 I/O 端口数据寄存器

每个 GPIO 都具有 2 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 用于存储待输出数据，可对其进行读/写访问。通过 I/O 输入的数据存储到输入数据寄存器 (GPIOx_IDR) 中，它是一个只读寄存器。参见第 7.4.5 节“GPIO 端口输入数据寄存器（GPIOx_IDR）（x = A, B）”和第 7.4.6 节“GPIO 端口输出数据寄存器（GPIOx_ODR）（x = A, B）”。

7.3.5 I/O 数据位操作

置位复位寄存器（GPIOx_BSRR）是一个 32 位寄存器，它允许应用程序在输出数据寄存器（GPIOx_ODR）中对各个单独的数据位执行置位和复位操作。置位复位寄存器的大小是 GPIOx_ODR 的二倍。

对于 GPIOx_ODR 中的每一位，GPIOx_BSRR 中有两个控制位与之相对应：BS(i)和 BR(i) 当写入 1 时，BS(i) 位会置位对应的 ODR(i) 位。当写入 1 时，BR(i) 位会复位 ODR(i) 对应的位。

在 GPIOx_BSRR 中向任何位写入 0 都不会对 GPIOx_ODR 中的对应位产生任何影响。如果在 GPIOx_BSRR 中同时尝试对某个位执行置位和清零操作，则置位操作优先。

使用 GPIOx_BSRR 寄存器更改 GPIOx_ODR 中各个位的值是一个“单次”操作，不会锁定 GPIOx_ODR 位。随时都可以直接访问 GPIOx_ODR 位。GPIOx_BSRR 寄存器提供了一种执行原子按位处理的方法。

在对 GPIOx_ODR 进行位操作时，软件无需禁止中断：在一次原子 AHB 写访问中，可以修改一个或多个位。

7.3.6 GPIO 锁定机制

通过将特定的写序列应用到 GPIOx_LCKR 寄存器，可以冻结 GPIO 控制寄存器。冻结的寄存器包括 GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR、GPIOx_AFRL 和 GPIOx_AFRH。

要对 GPIOx_LCKR 寄存器执行写操作，必须应用特定的写/读序列。当正确的 LOCK 序列应用到此寄存器的第 16 位后，会使用 LCKR[15:0]的值来锁定 I/O 的配置（在写序列期间，LCKR[15:0]的值必须相同）。如果对端口位应用了 LOCK 序列，则在下一次 MCU 复位或外设复位之前，端口位的值不能再修改。每个 GPIOx_LCKR 位都会冻结控制寄存器（GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR、GPIOx_AFRL 和 GPIOx_AFRH）中的对应位。

只能使用对 GPIOx_LCKR 寄存器的字（32 位长度）访问执行 LOCK 序列（参见第 7.4.8 节“GPIO 端口配置锁定寄存器（GPIOx_LCKR）（x = A, B）”），原因在于 GPIOx_LCKR 位 16 必须与[15:0]位同时置位。

如需更多信息，请参见第 7.4.8 节“GPIO 端口配置锁定寄存器（GPIOx_LCKR）（x = A, B）”中关于 LCKR 寄存器的描述。

7.3.7 I/O 复用功能输入/输出

提供了两个寄存器，用于从每个 I/O 的可用复用功能输入/输出中选择一个。借助于这些寄存器，用户可以将复用功能连接到其应用程序需要的其他一些引脚。

这意味着可使用 GPIOx_AFRL 和 GPIOx_AFRH 复用功能寄存器在每个 GPIO 上复用多个可用的外设功能。这样一来，应用程序可为每个 I/O 选择任何一个可用功能。由于 AF 选择信号由复用功能输入和复用功能输出共用，所以只需为每个 I/O 的复用功能输入/输出选择一个通道即可。

如需了解每个 GPIO 引脚上复用的功能，请参见第 4 节“I/O 工作模式”。

7.3.8 外部中断/唤醒线路

所有端口都具有外部中断功能。要使用外部中断线，必须将 IO 端口配置为输入模式。中断配置（电平/边沿、极性、屏蔽）必须在系统控制器（SYSCFG）中完成。参见第 8 节“系统控制器（SYSCFG）”。

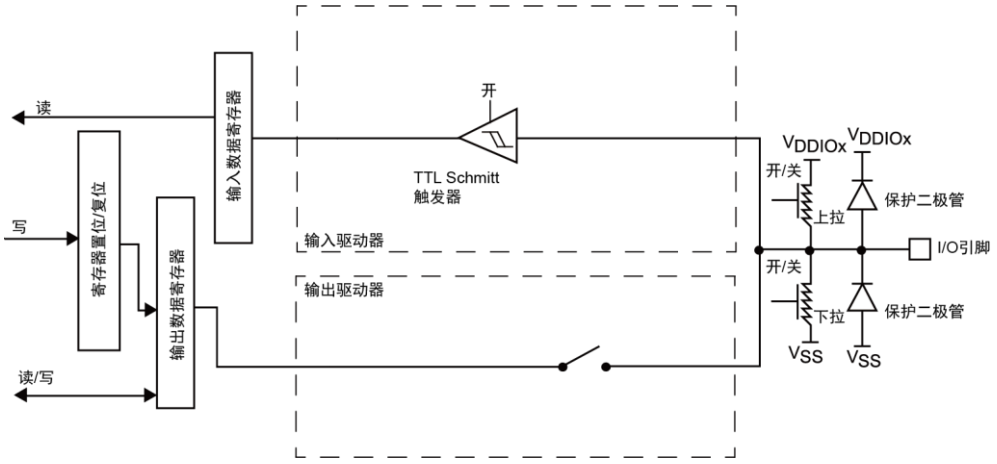
7.3.9 输入配置

对 I/O 端口进行编程作为输入时：

- 输出缓冲器被禁止
- 施密特触发器输入被打开
- 根据 GPIOx_PUPDR 寄存器中的值决定是否打开上拉和下拉电阻
- 输入数据寄存器每隔 1 个 AHB 时钟周期对 I/O 引脚上的数据进行一次采样
- 对输入数据寄存器的读访问可获取 I/O 状态。

图 17. 输入浮动/上拉/下拉配置显示了 I/O 端口位的输入配置。

图 17. 输入浮空/上拉/下拉配置



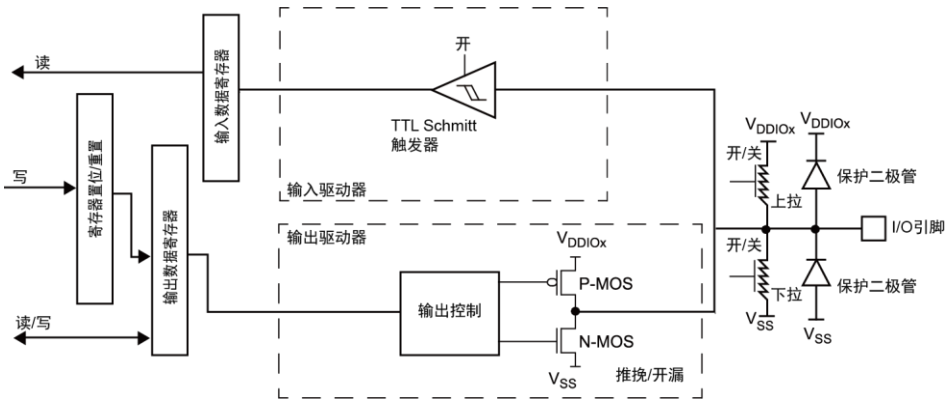
7.3.10 输出配置

对 I/O 端口进行编程作为输出时：

- 输出缓冲器被打开：
  - 开漏模式：输出寄存器中的“0”可激活 N-MOS，而输出寄存器中的“1”会使端口保持高组态（Hi-Z）（P-MOS 始终不激活）
  - 推挽模式：输出寄存器中的“0”可激活 N-MOS，而输出寄存器中的“1”可激活 P-MOS
- 施密特触发器输入被打开
- 根据 GPIOx_PUPDR 寄存器中的值决定是否打开上拉和下拉电阻
- 输入数据寄存器每隔 1 个 AHB 时钟周期对 I/O 引脚上的数据进行一次采样
- 对输入数据寄存器的读访问可获取 I/O 状态
- 对输出数据寄存器的读访问可获取最后的写入值

图 18. 输出配置显示了 I/O 端口位的输出配置。

图 18. 输出配置



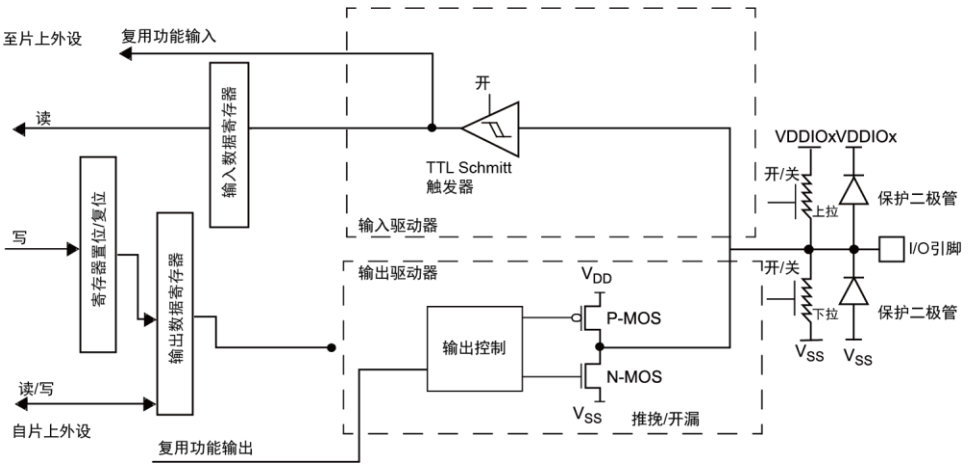
7.3.11 复用功能配置

对 I/O 端口进行编程作为复用功能时：

- 可将输出缓冲器配置为开漏或推挽模式
- 输出缓冲器由来自外设的信号驱动（发送器使能和数据）
- 施密特触发器输入被打开
- 根据 GPIOx_PUPDR 寄存器中的值决定是否打开弱上拉电阻和下拉电阻
- 输入数据寄存器每隔 1 个 AHB 时钟周期对 I/O 引脚上的数据进行一次采样
- 对输入数据寄存器的读访问可获取 I/O 状态。

图 19. 复用功能配置显示了 I/O 端口位的复用功能配置。

图 19. 复用功能配置



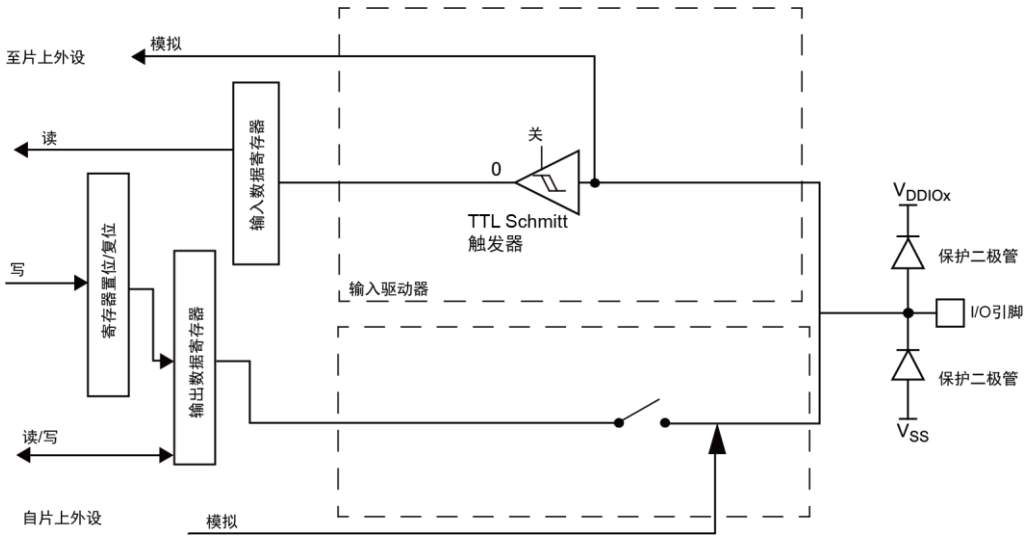
7.3.12 模拟配置

对 I/O 端口进行编程作为模拟配置时：

- 输出缓冲器被禁止
- 施密特触发器输入停用，I/O 引脚的每个模拟输入的功耗变为零。施密特触发器的输出被强制处理为恒定值（0）。
- 上拉和下拉电阻必须由软件禁用，否则相关模拟特性不会按预期正常工作。
- 对输入数据寄存器的读访问值为“0”。

图 20. 高阻抗模拟配置显示了 I/O 端口位的高阻抗模拟输入配置。

图 20. 高阻态模拟配置



7.3.13 使用 LSE 振荡器引脚作为 GPIO

当 LSE 振荡器关闭（复位后的默认状态）时，可将相关的振荡器引脚用作常规 GPIO。

当振荡器配置为用户外部时钟模式时，只保留了 OSC32_IN 引脚用于时钟输入，OSC32_OUT 引脚仍然可以用作正常 GPIO。

注意：当使用外部低速振荡器 (LSE) 时，对于与 OSC32_IN (PB13) 和 OSC32_OUT (PB12) 共享的 I/O，没有硬件机制用于自动隔离软件配置。用户必须小心地将相关 I/O 设定为输入浮动。

注意： *高速振荡器 (HSE) OSC_IN 和 OSC_OUT 引脚是专用振荡器引脚，不能用作 GPIO。*

7.4 GPIO 寄存器

本节对 GPIO 寄存器进行了详细介绍。

如需寄存器位、寄存器偏移地址和复位值的汇总信息，请参见表 16. GPIO 寄存器映射和复位值。可按字、半字或字节模式写入外设寄存器。

7.4.1 GPIO 端口模式寄存器（GPIOx_MODER）（x = A, B）

偏移地址：0x00 复位值：

- 0x0000 00A0（端口 A）
- 0x0000 0000（端口 B）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODE15[1:0]		MODE14[1:0]		MODE13[1:0]		MODE12[1:0]		MODE11[1:0]		MODE10[1:0]		MODE9[1:0]		MODE8[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
MODE7[1:0]		MODE6[1:0]		MODE5[1:0]		MODE4[1:0]		MODE3[1:0]		MODE2[1:0]		MODE1[1:0]		MODE0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 2y+1:2y	**MODEy[1:0]**：端口 x 配置位 (y = 0..15)。 这些位通过软件写入，用于配置 I/O 模式。00：输入模式 01：通用输出模式 10：复用功能模式 11：模拟模式 *注意：在模拟模式下配置焊盘时，为避免对模拟信号的噪声污染，用户必须注意禁用相关的上/下拉。*

7.4.2 GPIO 端口输出类型寄存器（GPIOx_OTYPER）（x = A, B）

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
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，必须保持复位值。
位 15:0	**OTy**：端口 x 配置位（y = 0..15）。 这些位通过软件写入，用于配置 I/O 输出类型。 0：输出推挽（复位状态） 1：输出开漏

7.4.3 GPIO 端口输出速度寄存器（GPIOx_OSPEEDR）（x = A, B）

偏移地址：0x08

复位值：

- 0x0000 0030（端口 A）
- 0x0000 0000（端口 B）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSPEED15 [1:0]		OSPEED14 [1:0]		OSPEED13 [1:0]		OSPEED12 [1:0]		OSPEED11 [1:0]		OSPEED10 [1:0]		OSPEED9 [1:0]		OSPEED8 [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
OSPEED7 [1:0]		OSPEED6 [1:0]		OSPEED5 [1:0]		OSPEED4 [1:0]		OSPEED3 [1:0]		OSPEED2 [1:0]		OSPEED1 [1:0]		OSPEED0 [1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 2y+1:2y	**OSPEEDy[1:0]**：端口 x 配置位 (y = 0..15)。 这些位通过软件写入，用于配置 I/O 输出速度。

7.4.4 GPIO 端口上/下拉寄存器（GPIOx_PUPDR）（x = A, B）

偏移地址：0x0C 复位值：

- 0x5555 5595（端口 A）
- 0x5555 5555（端口 B）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUPD15[1:0]		PUPD14[1:0]		PUPD13[1:0]		PUPD12[1:0]		PUPD11[1:0]		PUPD10[1:0]		PUPD9[1:0]		PUPD8[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
PUPD7[1:0]		PUPD6[1:0]		PUPD5[1:0]		PUPD4[1:0]		PUPD3[1:0]		PUPD2[1:0]		PUPD1[1:0]		PUPD0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 2y+1:2y	PUPDy[1:0]：端口 x 配置位（y = 0..15）。 这些位通过软件写入，用于配置 I/O 上拉或下拉。 00：无上拉或下拉 01：上拉 10：下拉 11：保留

注意： *当 PWRC_CR1[4] = APC 位置位时，GPIOx_PUPDR 不影响行为。当 PWRC_CR1[4] = APC 位未置位时，GPIOx_PUPDR 上/下拉配置在低功耗模式下无效。*

7.4.5 GPIO 端口输入数据寄存器（GPIOx_IDR）（x = A, B）

偏移地址：0x10

复位值：0x0000 XXXX（其中 X 表示未定义）

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
ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16	保留，必须保持复位值。
位 15:0	**IDy**：端口输入数据位（y = 0..15）。 这些位为只读。它们包含相应 I/O 端口的输入值。

7.4.6 GPIO 端口输出数据寄存器（GPIOx_ODR）（x = A, B）

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
OD15	OD14	OD13	OD12	OD11	OD10	OD9	OD8	OD7	OD6	OD5	OD4	OD3	OD2	OD1	OD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16	保留，必须保持复位值。
位 15:0	**ODy**：端口输出数据位 (y = 0..15)。 这些位可通过软件读取和写入。 *注意：　对于原子位置位/复位，通过对 GPIOx_BSRR 或 GPIOx_BRR 寄存器 (x = A, B) 执行写操作，可将 OD 位分别置位和/或复位。*

7.4.7 GPIO 端口位置位/复位寄存器（GPIOx_BSRR）（x = A, B）

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）。 这些位为只写。读取这些位可返回值 0x0000。0：对应 ODx 位上无操作 1：复位对应 ODx 位 *注意：　如果同时对 BSx 和 BRx 置位，则 BSx 的优先级更高。*
位 15:0	**BSy**：端口 x 置位位 y（y= 0..15）。 这些位为只写。读取这些位可返回值 0x0000。0：对应 ODx 位上无操作 1：对应 ODx 位置位

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

当正确的写序列应用到第 16 位（LCKK）时，此寄存器将用于锁定端口位的配置。位[15:0]的值用于锁定 GPIO 的配置。在写序列期间，不能更改 LCKR[15:0] 的值。如果对端口位应用了 LOCK 序列，则在下一次 MCU 复位或外设复位之前，该端口位的值不能再修改。

注意： *可使用特定的写序列对 GPIOx_LCKR 寄存器执行写操作。在此锁定序列期间只允许使用字访问（32 位长）。*

每个锁定位冻结一个特定的配置寄存器（控制寄存器和复用功能寄存器）。

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，必须保持复位值。
位 16	**LCKK**：锁定键。 可随时读取此位。可使用锁定键写序列对其进行修改。0：端口配置锁定键未激活 1：端口配置锁定键已激活。在下一次 MCU 复位或外设复位之前，GPIOx_LCKR 寄存器被锁定。锁定键写序列： WR LCKR[16] = ‘1’ +LCKR[15:0] WR LCKR[16] = ‘0’ + LCKR[15:0] WR LCKR[16] = ‘1’ + LCKR[15:0] RD LCKR RD LCKR[16] = ‘1’（此读操作为可选操作，但它可确认锁定已激活） *注意：* *在锁定键写序列期间，不能更改 LCK[15:0]的值。锁定序列中的任何错误都将中止锁定操作。在端口任意位上的第一个锁定序列之后，在下一次 MCU 复位或外设复位之前，对 LCKK 位的任何读访问都将返回“1”。*
位 15:0	**LCKy**：端口 x 锁定位 y (y= 0..15)。 这些位支持读/写，但在 LCKK 位为“0”时，只能使用 LCKK 位说明中描述的特定序列进行写入。 0：端口配置未锁定 1：端口配置已锁定

7.4.9 GPIO 复用功能低寄存器（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
AFSEL7[3:0]				AFSEL6[3:0]				AFSEL5[3:0]				AFSEL4[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
AFSEL3[3:0]				AFSEL2[3:0]				AFSEL1[3:0]				AFSEL0[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0	**y[3:0]**：端口 x 引脚 y（y = 0..7）的复用功能选择。 这些位通过软件写入，用于配置复用功能 I/O。
	AFSELy 选择：0000：AF0 0001：AF1 0010：AF2 0011：AF3 0100：AF4 其他：保留

7.4.10 GPIO 复用功能高寄存器（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
AFSEL15[3:0]				AFSEL14[3:0]				AFSEL13[3:0]				AFSEL12[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
AFSEL11[3:0]				AFSEL10[3:0]				AFSEL9[3:0]				AFSEL8[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0	**y[3:0]**：端口 x 引脚 y（y = 8..15）的复用功能选择。 这些位通过软件写入，用于配置复用功能 I/O。
	AFSELy 选择：0000：AF0 0001：AF1 0010：AF2 0011：AF3 0100：AF4 其他：保留

7.5 GPIO 寄存器映射

下表提供了 GPIO 寄存器映射和复位值。

表 16. GPIO 寄存器映射和复位值

偏移	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
0x00	GPIOA_MODER	MODE15[1:0]		MODE14[1:0]		MODE13[1:0]		MODE12[1:0]		MODE11[1:0]		MODE10[1:0]		MODE9[1:0]		MODE8[1:0]		MODE7[1:0]		MODE6[1:0]		MODE5[1:0]		MODE4[1:0]		MODE3[1:0]		MODE2[1:0]		MODE1[1:0]		MODE0[1:0]	
	复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0
0x00	GPIOB_MODER	MODE15[1:0]		MODE14[1:0]		MODE13[1:0]		MODE12[1:0]		MODE11[1:0]		MODE10[1:0]		MODE9[1:0]		MODE8[1:0]		MODE7[1:0]		MODE6[1:0]		MODE5[1:0]		MODE4[1:0]		MODE3[1:0]		MODE2[1:0]		MODE1[1:0]		MODE0[1:0]	
	复位值	0	0	0	0	0	0	0	0	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（其中：x = A、B）	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
	复位值																																
0x08	GPIOA_OSPEEDR	OSPEED15[1:0]		OSPEED14[1:0]		OSPEED13[1:0]		OSPEED12[1:0]		OSPEED11[1:0]		OSPEED10[1:0]		OSPEED9[1:0]		OSPEED8[1:0]		OSPEED7[1:0]		OSPEED6[1:0]		OSPEED5[1:0]		OSPEED4[1:0]		OSPEED3[1:0]		OSPEED2[1:0]		OSPEED1[1:0]		OSPEED0[1:0]	
	复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
0x08	GPIOB_OSPEEDR	OSPEED15[1:0]		OSPEED14[1:0]		OSPEED13[1:0]		OSPEED12[1:0]		OSPEED11[1:0]		OSPEED10[1:0]		OSPEED9[1:0]		OSPEED8[1:0]		OSPEED7[1:0]		OSPEED6[1:0]		OSPEED5[1:0]		OSPEED4[1:0]		OSPEED3[1:0]		OSPEED2[1:0]		OSPEED1[1:0]		OSPEED0[1:0]	
	复位值	0	0	0	0	0	0	0	0	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]	
	复位值	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	1	1	0	0	1	0	1	0	1
0x0C	GPIOB_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]	
	复位值	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	1	0	1

偏移	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
0x10	**GPIOx_IDR** （其中：x = A、B）	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
	复位值																	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x14	**GPIOx_ODR** （其中：x = A、B）	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OD15	OD14	OD13	OD12	OD11	OD10	OD9	OD8	OD7	OD6	OD5	OD4	OD3	OD2	OD1	OD0
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	**GPIOx_BSRR** （其中：x = A、B）	BR15	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
	复位值	0	0	0	0	0	0	0	0	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** （其中：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
	复位值																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	**GPIOx_AFRL** （其中：x = A、B）	AFSEL7[3:0]				AFSEL6[3:0]				AFSEL5[3:0]				AFSEL4[3:0]				AFSEL3[3:0]				AFSEL2[3:0]				AFSEL1[3:0]				AFSEL0[3:0]			
	复位值	0	0	0	0	0	0	0	0	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** （其中：x = A、B）	AFSEL15[3:0]				AFSEL14[3:0]				AFSEL13[3:0]				AFSEL12[3:0]				AFSEL11[3:0]				AFSEL10[3:0]				AFSEL9[3:0]				AFSEL8[3:0]			
	复位值	0	0	0	0	0	0	0	0	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** （其中：x = A、B）	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
	复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																

请参见“表 3. BlueNRG-LP 存储器映射和外设寄存器边界地址”，了解寄存器边界地址。

8 系统控制器（SYSCFG）

系统控制器是一组链接到 BlueNRG-LP 器件系统特性的寄存器（配置、控制和状态）。

8.1 SYSCFG 主要特性

系统控制器系列寄存器主要链接到：

- 提供 JTAG_ID、晶片 ID 和切割版本
- 管理关联到 GPIO 特性的中断
- 管理关联到电源控制器 (PWRC) 的中断
- 管理关联到 MR_BLE 接收和发送序列的中断
- 使能/禁用一些 I/O 上的 I^2C 超快速模式加驱动功能

注意： *该外设位于非保留电源域中，因此在 DEEPSTOP 后，在相关寄存器中执行的所有置位都将丢失。*

8.2 系统控制器寄存器说明

8.2.1 晶片 ID 寄存器（DIE_ID）

该寄存器提供器件版本和切割信息。

偏移地址：0x00

复位值：0x0000 0120

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.	PRODUCT[3:0]				VERSION[3:0]				REVISION[3:0]			
				r	r	r	r	r	r	r	r	r	r	r	r

位 31:12	保留，必须保持复位值。
位 11:8	**PRODUCT**：产品版本。
位 7:4	**VERSION**：晶元版本修订号。
位 3:0	**REVISION**：晶元修订版本（金属层修改）。

8.2.2 JTAG ID 寄存器（JTAG_ID）

该寄存器提供 BlueNRG-LP 的 JTAG ID。

偏移地址：0x04

复位值：0x0201E041

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

位 31:28	**VERSION_NUMBER**：版本。
位 27:12	**PART_NUMBER**：产品编号。
位 11:1	**MANUF_ID**：制造商 ID。
位 0	**RESERVED**

8.2.3 I2C 超快速模式加引脚功能控制寄存器（I2C_FMP_CTRL）

该寄存器允许激活 I2C 开漏焊盘上的超快速模式加驱动功能。

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.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.									I2C2_P B7_FM P	I2C2_P B6_FM P	I2C1_P A1_FM P	I2C1_P A0_FM P
												rw	rw	rw	rw

位 31:4	保留，必须保持复位值。
位 3	**I2C2_PB7_FMP**：PB7 I/O 上 I2C2_SDA 的 I2C2 超快速模式加驱动功能。 • 0：PB7 引脚以标准模式工作 • 1：PB7 引脚上使能 FM+ 模式，速度控制被旁路
位 2	**I2C2_PB6_FMP**：PB6 I/O 上 I2C2_SCL 的 I2C2 超快速模式加驱动功能。 • 0：PB6 引脚以标准模式工作 • 1：PB6 引脚上使能 FM+ 模式，速度控制被旁路
位 1	**I2C1_PA1_FMP**：PA1 I/O 上 I2C1_SDA 的 I2C1 超快速模式加驱动功能。 • 0：PA1 引脚以标准模式工作 • 1：PA1 引脚上使能 FM+ 模式，速度控制被旁路
位 0	**I2C1_PA0_FMP**：PA0 I/O 上 I2C1_SCL 的 I2C1 超快速模式加驱动功能。 • 0：PA0 引脚以标准模式工作 • 1：PA0 引脚上使能 FM+ 模式，速度控制被旁路

8.2.4 I/O 中断检测类型寄存器（IO_DTR）

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
PB15_DT	PB14_DT	PB13_DT	PB12_DT	PB11_DT	PB10_DT	PB9_DT	PB8_DT	PB7_DT	PB6_DT	PB5_DT	PB4_DT	PB3_DT	PB2_DT	PB1_DT	PB0_DT
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
PA15_DT	PA14_DT	PA13_DT	PA12_DT	PA11_DT	PA10_DT	PA9_DT	PA8_DT	PA7_DT	PA6_DT	PA5_DT	PA4_DT	PA3_DT	PA2_DT	PA1_DT	PA0_DT
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16	**PBx_DT**（x=15 至 0）：端口 B I/O 的中断检测类型。 • 0：边沿检测 • 1：电平检测
位 15:0	**PAx_DT**（x=15 至 0）：端口 A I/O 的中断检测类型。 • 0：边沿检测 • 1：电平检测

8.2.5 I/O 中断边沿寄存器（IO_IBER）

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
PB15_IBE	PB14_IBE	PB13_IBE	PB12_IBE	PB11_IBE	PB10_IBE	PB9_IBE	PB8_IBE	PB7_IBE	PB6_IBE	PB5_IBE	PB4_IBE	PB3_IBE	PB2_IBE	PB1_IBE	PB0_IBE
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
PA15_IBE	PA14_IBE	PA13_IBE	PA12_IBE	PA11_IBE	PA10_IBE	PA9_IBE	PA8_IBE	PA7_IBE	PA6_IBE	PA5_IBE	PA4_IBE	PA3_IBE	PA2_IBE	PA1_IBE	PA0_IBE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16	**PBx_IBE**（x=15 至 0）：端口 B I/O 的中断边沿选择。 • 0：单边沿检测 • 1：双边沿检测
位 15:0	**PAx_IBE**（x=15 至 0）：端口 A I/O 的中断边沿选择。 • 0：单边沿检测 • 1：双边沿检测

8.2.6 I/O 中断极性事件寄存器 (IO_IEVR)

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
PB15_IEV	PB14_IEV	PB13_IEV	PB12_IEV	PB11_IEV	PB10_IEV	PB9_IEV	PB8_IEV	PB7_IEV	PB6_IEV	PB5_IEV	PB4_IEV	PB3_IEV	PB2_IEV	PB1_IEV	PB0_IEV
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
PA15_IEV	PA14_IEV	PA13_IEV	PA12_IEV	PA11_IEV	PA10_IEV	PA9_IEV	PA8_IEV	PA7_IEV	PA6_IEV	PA5_IEV	PA4_IEV	PA3_IEV	PA2_IEV	PA1_IEV	PA0_IEV
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16	**PBx_IEV**（x=15 至 0）：端口 B I/O 的中断极性事件。 • 0：下降沿/低电平 • 1：上升沿/高电平
位 15:0	**PAx_IEV**（x=15 至 0）：端口 A I/O 的中断极性事件。 • 0：下降沿/低电平 • 1：上升沿/高电平

8.2.7 I/O 中断使能寄存器（IO_IER）

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
PB15_IE	PB14_IE	PB13_IE	PB12_IE	PB11_IE	PB10_IE	PB9_IE	PB8_IE	PB7_IE	PB6_IE	PB5_IE	PB4_IE	PB3_IE	PB2_IE	PB1_IE	PB0_IE
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
PA15_IE	PA14_IE	PA13_IE	PA12_IE	PA11_IE	PA10_IE	PA9_IE	PA8_IE	PA7_IE	PA6_IE	PA5_IE	PA4_IE	PA3_IE	PA2_IE	PA1_IE	PA0_IE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16	**PBx_IE**（x=15 至 0）：端口 B I/O 的中断使能。 • 0：禁用中断 • 1：使能中断
位 15:0	**PAx_IE**（x=15 至 0）：端口 A I/O 的中断使能。 • 0：禁用中断 • 1：使能中断

8.2.8 I/O 中断状态和清零寄存器（IO_ISCR）

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
PB15_ISC	PB14_ISC	PB13_ISC	PB12_ISC	PB11_ISC	PB10_ISC	PB9_ISC	PB8_ISC	PB7_ISC	PB6_ISC	PB5_ISC	PB4_ISC	PB3_ISC	PB2_ISC	PB1_ISC	PB0_ISC
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
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PA15_ISC	PA14_ISC	PA13_ISC	PA12_ISC	PA11_ISC	PA10_ISC	PA9_ISC	PA8_ISC	PA7_ISC	PA6_ISC	PA5_ISC	PA4_ISC	PA3_ISC	PA2_ISC	PA1_ISC	PA0_ISC
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 31:16	**PBx_ISC**（x=15 至 0）：端口 B I/O 的中断状态（屏蔽前）。 • 0：无挂起中断 • 1：对应 I/O 上发生了事件/发生了中断（若已使能）。通过在此位写入 1 进行清零
位 15:0	**PAx_ISC**（x=15 至 0）：端口 A I/O 的中断状态（屏蔽前）。 • 0：无挂起中断 • 1：对应 I/O 上发生了事件/发生了中断（若已使能）。通过在此位写入 1 进行清零

8.2.9 电源控制器中断使能寄存器（PWRC_IER）

该寄存器允许电源控制器（PWRC）模块的中断源上的使能或屏蔽控制。

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
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.										WKUP_IE	PVD_I E	Res.
													rw	rw	

位 31:3	保留，必须保持复位值。
位 2	**WKUP_IE**：电源控制器唤醒事件中断使能。 • 0：禁用 PWRC 所见的发生唤醒事件时中断 • 1：使能 PWRC 所见的发生唤醒事件时中断
位 1	**PVD_IE**：可编程电压检测器中断使能。 • 0：PVD 中断禁用 • 1：中断使能
位 0	保留，必须保持复位值。

8.2.10 电源控制器中断状态和清零寄存器（PWRC_ISCR）

该寄存器允许状态检查并将电源控制器（PWRC）模块的中断源清零。

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.	Res.	Res.	Res.	Res.	Res.	WKUP_ISC	PVD_I SC	Res.
													rc_w1	rc_w1	

位 31:3	保留，必须保持复位值。
位 2	**WKUP_ISC**：指示电源控制器接收到唤醒事件。 • 0：无挂起中断 • 1：PWRC 上发生了唤醒事件/发生了中断（若已使能）。通过在此位写入 1 进行清零。 如果唤醒事件如此接近于低功耗模式进入请求，以至于 PWRC 在关闭系统之前中止，则该标志读为 1。
位 1	**PVD_ISC**：可编程电压检测器状态。 • 0：无挂起中断 • 1：电压低于设定阈值/发生了中断（若已使能）。通过在此位写入 1 进行清零。 参见第 5.3.2 节“电源电压检测（PVD）”了解详情。
位 0	保留，必须保持复位值。

8.2.11 I/O 模拟开关控制寄存器（GPIO_SWA_CTRL）

该寄存器用于选择模拟源，以连接到内置两个特性的模拟焊盘。

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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PGACAP_nVIN 0	ATB1_n PVD
														rw	rw

位 31:2	保留，必须保持复位值。
位 1	**PGACAP_nVIN0**：选择 PB2/PB3 上的模拟特性：在 PGA 上连接外部电容（适用于模拟音频模式）或 VINM/P0，以便在 PB2/3 I/O 设定为模拟模式（在相关 GPIO_MODER 寄存器中）时进行 ADC 测量： • 0：选择 VINM/P0 特性（默认） • 1：选择 PGA 特性的外部电容 WARNING：在通过 VINM/P0 引脚进行的模拟测量中，如果配置选择 PGA 外部电容且软件使能 ADC，不提供特定的硬件保护。
位 0	**ATB1_nPVD**：当 PB14 I/O 设定为模拟模式（在相关的 GPIO_MODER 寄存器中）时，在 ATB1 和 PVD 之间选择 PB14 上的模拟特性： • 0：选择 PVD 外部电压特性（默认） • 1：选择 ATB1 特性

8.2.12 MR_BLE RX 或 TX 序列信息检测类型寄存器（BLERXTX_DTR）

该寄存器用于选择来自 MR_BLE IP 的 RX_SEQUENCE 和 TX_SEQUENCE 上的检测类型（电平或边沿）。

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
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.	RX_DT	TX_DT
														rw	rw

位 31:2	保留，必须保持复位值。
位 1	**RX_DT**：RX_SEQUENCE 信号的检测类型： • 0：检测边沿（默认） • 1：检测电平
位 0	**TX_DT**：TX_SEQUENCE 信号上的检测类型： • 0：检测边沿（默认） • 1：检测电平

8.2.13 MR_BLE RX 或 TX 序列信息检测类型寄存器（BLERXTX_IBER）

当边沿检测类型激活时，该寄存器用于激活单边沿或双边沿的 RX_SEQUENCE 和 TX_SEQUENCE 信号检测。

偏移地址：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.	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.	RX_IBE	TX_IBE
														rw	rw

位 31:2	保留，必须保持复位值。
位 1	**RX_IBE**：RX_SEQUENCE 信号的中断边沿寄存器： • 0：检测单边沿（默认） • 1：检测双边沿
位 0	**TX_IBE**：TX_SEQUENCE 信号的中断边沿寄存器： • 0：检测单边沿（默认） • 1：检测双边沿

8.2.14 MR_BLE RX 或 TX 序列信息检测事件寄存器（BLERXTX_IEVR）

该寄存器定义 RX_SEQUENCE 和 TX_SEQUENCE 信号检测的极性。

偏移地址：0x34

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.	RX_IEV	TX_IEV
														rw	rw

位 31:2	保留，必须保持复位值。
位 1	**RX_IEV**：RX_SEQUENCE 信号上的中断极性事件： • 0：检测下降沿/低电平（默认） • 1：检测上升沿/高电平
位 0	**TX_IEV**：TX_SEQUENCE 信号上的中断极性事件： • 0：检测下降沿/低电平（默认） • 1：检测上升沿/高电平

8.2.15 MR_BLE RX 或 TX 序列信息检测中断使能寄存器（BLERXTX_IER）

该寄存器定义 RX_SEQUENCE 和 TX_SEQUENCE 信号的中断使能。

偏移地址：0x38

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.	Res.

位 31:2	保留，必须保持复位值。
位 1	**RX_IE**：RX_SEQUENCE 信号上的中断使能： • 0：中断禁用（默认） • 1：中断使能
位 0	**TX_IE**：TX_SEQUENCE 信号上的中断使能： • 0：中断禁用（默认） • 1：中断使能

8.2.16 MR_BLE RX 或 TX 序列信息检测状态和清零寄存器（BLERXTX_ISCR）

该寄存器用于检查状态并将链接到 MR_BLE IP 提供的 RX_SEQUENCE 和 TX_SEQUENCE 信息的中断清零。

偏移地址：0x3C

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.	RX_ISC	TX_ISC
														rc_wc1	rc_wr1

位 31:2	保留，必须保持复位值。
位 1	**RX_ISC**：RX_SEQUENCE 信号上的中断状态（可以是上升沿或下降沿，取决于 BLERXTX_IEVR 和 BLERXTX_IBER）： • 0：RX_SEQUENCE 上未检测到活动 • 1：RX_SEQUENCE 上发生活动
位 0	**TX_ISC**：TX_SEQUENCE 信号上的中断状态（可以是上升沿或下降沿，取决于 BLERXTX_IEVR 和 BLERXTX_IBER）： • 0：TX_SEQUENCE 上未检测到活动 • 1：TX_SEQUENCE 上发生活动

8.3 系统控制器寄存器映射

请参见“表 3. BlueNRG-LP 存储器映射和外设寄存器边界地址”了解 BlueNRG-LP 中的系统控制器基址位置。

表 17. SYSCFG 寄存器映射和复位值

偏移	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
0x00	DIE_ID	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRODUCT				VERSION				REVISION			
	复位值																					0	0	0	1	0	0	1	0	0	0	0	0
0x04	JTAG_ID	VERSION_NUMBER				PART_NUMBER																MANUF_ID											Res.
	复位值	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	1
0x08	I2C_FMP_CT RL	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.	I2C2 PB7 FMP	I2C2 PB6 FMP	I2C1 PA1 FMP	I2C1 PA0 FMP
	复位值																													0	0	0	0
0x0C	IO_DTR	PB15 DT	PB14 DT	PB13 DT	PB12 DT	PB11 DT	PB10 DT	PB9 DT	PB8 DT	PB7 DT	PB6 DT	PB5 DT	PB4 DT	PB3 DT	PB2 DT	PB1 DT	PB0 DT	PA15 DT	PA14 DT	PA13 DT	PA12 DT	PA11 DT	PA10 DT	PA9 DT	PA8 DT	PA7 DT	PA6 DT	PA5 DT	PA4 DT	PA3 DT	PA2 DT	PA1 DT	PA0 DT
	复位值	0	0	0	0	0	0	0	0	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	IO_IBER	PB15 IBE	PB14 IBE	PB13 IBE	PB12 IBE	PB11 IBE	PB10 IBE	PB9 IBE	PB8 IBE	PB7 IBE	PB6 IBE	PB5 IBE	PB4 IBE	PB3 IBE	PB2 IBE	PB1 IBE	PB0 IBE	PA15 IBE	PA14 IBE	PA13 IBE	PA12 IBE	PA11 IBE	PA10 IBE	PA9 IBE	PA8 IBE	PA7 IBE	PA6 IBE	PA5 IBE	PA4 IBE	PA3 IBE	PA2 IBE	PA1 IBE	PA0 IBE
	复位值	0	0	0	0	0	0	0	0	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	IO_IEVR	PB15 IEV	PB14 IEV	PB13 IEV	PB12 IEV	PB11 IEV	PB10 IEV	PB9 IEV	PB8 IEV	PB7 IEV	PB6 IEV	PB5 IEV	PB4 IEV	PB3 IEV	PB2 IEV	PB1 IEV	PB0 IEV	PA15 IEV	PA14 IEV	PA13 IEV	PA12 IEV	PA11 IEV	PA10 IEV	PA9 IEV	PA8 IEV	PA7 IEV	PA6 IEV	PA5 IEV	PA4 IEV	PA3 IEV	PA2 IEV	PA1 IEV	PA0 IEV
	复位值	0	0	0	0	0	0	0	0	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	IO_IER	PB15 IE	PB14 IE	PB13 IE	PB12 IE	PB11 IE	PB10 IE	PB9 IE	PB8 IE	PB7 IE	PB6 IE	PB5 IE	PB4 IE	PB3 IE	PB2 IE	PB1 IE	PB0 IE	PA15 IE	PA14 IE	PA13 IE	PA12 IE	PA11 IE	PA10 IE	PA9 IE	PA8 IE	PA7 IE	PA6 IE	PA5 IE	PA4 IE	PA3 IE	PA2 IE	PA1 IE	PA0 IE
	复位值	0	0	0	0	0	0	0	0	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	IO_ISCR	PB15 ISC	PB14 ISC	PB13 ISC	PB12 ISC	PB11 ISC	PB10 ISC	PB9 ISC	PB8 ISC	PB7 ISC	PB6 ISC	PB5 ISC	PB4 ISC	PB3 ISC	PB2 ISC	PB1 ISC	PB0 ISC	PA15 ISC	PA14 ISC	PA13 ISC	PA12 ISC	PA11 ISC	PA10 ISC	PA9 ISC	PA8 ISC	PA7 ISC	PA6 ISC	PA5 ISC	PA4 ISC	PA3 ISC	PA2 ISC	PA1 ISC	PA0 ISC
	复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

偏移	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
0x20	PWRC_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.	WKUP_IE	PVD_IE	Res.
	复位值																														0	0	
0x24	PWRC_ISCR	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.	WKUP_ISC	PVD_ISC	Res.
	复位值																														0	0	
0x28	保留																																
0x2c	BLERXTX_DTR	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.	RX_DT	TX_DT
																																0	0
0x30	BLERXTX_IBER	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.	RX_IBE	TX_IBE
																																0	0
0x34	BLERXTX_IEVR	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.	RX_IEV	TX_IBV
																																0	0
0x38	BLERXTX_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.	RX_IE	TX_IB
																																0	0
0x3C	BLERXTX_ISCR	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.	RX_ISC	TX_ISC
																																0	0

9 嵌入式 Flash 存储器

Flash 控制器实施擦除并设定 Flash 存储器操作。Flash 控制器还实施读和写保护。

9.1 Flash 主要特性

Flash 存储器特性：

- 至多 256 kB Flash 存储器单存储区架构。
- 存储器组织结构：1 个存储区
  - 主存储器：至多 256 kB
  - 页面大小：2 Kb
- 32 位宽数据读取
- 32 位宽数据写入
- 页面擦除（2 kB）和批量擦除。

Flash 存储器特性：

- Flash 存储器读操作
- Flash 存储器写操作：单次数据写入或 4x32 位突发写入（设定时间减少到 1/4）
- Flash 存储器擦除操作
- Flash 读出保护和调试访问禁用
- 页面写保护机制

9.2 说明

Flash 存储器结构如下：

- 主存储器模块包含 128 个页面，每个页面 2 kB。每个页面有 8 行，每行 256 字节（64 字）。

擦除 Flash 的每个位单元中的全部 Flash 结果。

同时，Flash 控制器管理 1 kB OTP（一次性可编程）用户数据。OTP 数据无法擦除。当最后一个字（地址 0x1000_1BFC）不同于 0xFFFF_FFFF 且发生系统复位时，OTP 数据区不能再是只写。

Flash 被映射到 AHB-Lite 总线上，地址范围如下。

表 18. Flash 存储器节区地址

章节	Flash AHB 起始地址	Flash AHB 结束地址
主 Flash	0x1004_0000	0x1007_FFFF(1)
用户 OTP	0x1000_1800	0x1000_1BFF

1. *取决于 Flash 大小。参见表 21. Flash 大小信息。*

9.3 Flash 控制器寄存器映射

请参见“表 3. BlueNRG-LP 存储器映射和外设寄存器边界地址”了解 BlueNRG-LP 中的 Flash 控制器基址位置。

表 19. Flash APB 寄存器

偏移地址	名称	宽度	RW[1]	复位	说明
0x00	COMMAND	8	RW	0x0000_0000	模块指令。参见第 9.4.1 节“指令寄存器（COMMAND）”。
0x04	CONFIG	2	RW	0x000_0018	配置封装。参见第 9.4.2 节“配置寄存器（CONFIG）”。
0x08	IRQSTAT	5	RC	0x0000_0000	Flash 状态中断（已屏蔽）。参见第 9.4.3 节“中断状态寄存器（IRQSTAT）”。
0x0C	IRQMASK	5	RW	0x0000_003F	中断屏蔽。参见第 9.4.4 节“中断屏蔽寄存器（IRQMASK）”。
0x10	IRQRAW	5	RC	0x0000_0001	状态中断（未屏蔽）。参见第 9.4.5 节“原始状态寄存器（IRQRAW）”。
0x14	FLASH_SIZE	16	RO	0x000-_----	指示主 Flash 的最后一个可使用地址和 RAM 大小。参见第 9.4.6 节“SIZE 寄存器”。
0x18	ADDRESS	14	RW	0x0000_0000	Flash 编程地址，自动递增。参见第 9.4.7 节“地址寄存器（ADDRESS）”。
0x24	LFSRVAL	32	RO	0xFFFF_FFFF	MASS READ 指令后 LFSR 寄存器需要检查。参见第 9.4.8 节“线性反馈移位寄存器（LFSRVAL）”。
0x34	PAGEPROT0	32	RW	0x0000_0000	写入/页面擦除保护管理寄存器。参见第 9.4.9 节“主 Flash 页面保护寄存器（PAGEPROTx）”（PAGEPRT0）。
0x38	PAGEPROT1	32	RW	0x0000_0000	写入/页面擦除保护管理寄存器。参见参见第 9.4.9 节“主 Flash 页面保护寄存器（PAGEPROTx）”（PAGEPRT1）。
0x3C	RESERVED	32	RW	0x0000_0000	UNUSED
0x40	DATA0	32	RW	0xFFFF_FFFF	设定周期数据。参见第 9.4.10 节“数据寄存器（DATA0-DATA3）”。
0x44	DATA1	32	RW	0xFFFF_FFFF	设定周期数据。参见第 9.4.10 节“数据寄存器（DATA0-DATA3）”。
0x48	DATA2	32	RW	0xFFFF_FFFF	设定周期数据。参见第 9.4.10 节“数据寄存器（DATA0-DATA3）”。
0x4C	DATA3	32	RW	0xFFFF_FFFF	设定周期数据。参见第 9.4.10 节“数据寄存器（DATA0-DATA3）”。

1. 缩略语含义：RW：读和写 RC：读和写以清零 RO：只读。

9.4 Flash 控制器寄存器说明

9.4.1 指令寄存器（COMMAND）

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.	COMMAND							
								rw							

位 31:8	保留，必须保持复位值。
位 7:0	**COMMAND**：用于启动 Flash 存储器上的任何运算的指令运算码。请参见“表 20. 客户可用的指令列表”。

表 20. 客户可用的指令列表

指令	Flash 扇区	说明	指令运算码
ERASE	主存储器	擦除寄存器 ADDRESS 定义的页。	0x11
WRITE(1)	主存储器	设定一个位置（由寄存器 DATA 和 ADDRESS 定义）。	0x33
MASSREAD	主存储器	读取所有位置并与 DATA 寄存器值进行比较或生成 LFSR 签名。	0x55
SLEEP	整个存储器	使 Flash 进入睡眠模式。 **警告：在该指令启动后，到请求 WAKE 指令（并返回相关的 CMDDONE 状态）之前，不可能对 Flash 元件进行任何访问（读取）或操作（除 WAKE 外的任何指令）**	0xAA
WAKEUP	整个存储器	使 Flash 退出睡眠模式。	0xBB
BURSTWRITE	主存储器	用 DATA0 - DATA3 寄存器中的数据设定 4 个位置（ADDRESS 0 → ADDRESS+3）。	0xCC
OTPWRITE	用户 OTP	客户可进行一次 1 kB 的写入。 不能进行擦除或二次设定。	0xEE
KEYWRITE	主存储器	写入用于保护主 Flash 的客户密钥。	0xFF

1. *每个地址只能设定两次，两次之间无擦除操作。*

状态位行为与指令：

- 对 COMMAND 寄存器的写操作将启动在 Flash 上执行的操作
- CMDSTART 标志置位并保持高电平直至被清零
- 当指令完成时，CMDDONE 标志变为高电平
- 当给出 MASS READ 指令时和 CMDDONE 为高电平时，可以检查 READOK 标志或者可以读取 LFSRVAL 寄存器（包含批量读取的签名）

第 9.5 节“程序员模型”中描述了不同指令的使用顺序。

9.4.2 配置寄存器（CONFIG）

偏移地址：0x04

复位值：0x0000 0018

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.	WAIT_STATES		Res.	DIS_GROUP_WRITE	REMAP	Res.
										rw	rw		rw	rw	

位 31:6	保留，必须保持复位值。
位 5:4	**WAIT_STATES**：要在 Flash 读取（AHB 访问）时插入的等待状态数目。 当系统时钟频率为 64 MHz 时，BlueNRG-LP 器件中嵌入的 Flash 需要 1 个 wait_state。
位 3	保留，必须保持复位值。
位 2	**DIS_GROUP_WRITE**： • 0：允许/使能突发写操作 • 1：阻止突发写操作并导致单次写入 注意：如果在进行突发写操作时此位置位，Flash 控制器将在当前字写入结束时停止写操作，即使仍然有一些字等待写入。
位 1	**REMAP**：用于重定向 SRAM0 上自举区的位。
位 0	保留，必须保持复位值。

当系统时钟为 32 MHz 最大值时，可在一个系统时钟周期内读取 Flash（最优功耗）。

9.4.3 中断状态寄存器（IRQSTAT）

中断状态寄存器显示了中断原始寄存器的屏蔽版本。

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.	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.	READOK_MIS	ILLCMD_ MIS	CMDERR_MIS	CMDSTART_MIS	CMDDONE_MIS
											rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:5	保留，必须保持复位值。
位 4	**READOK_MIS**：批量读取 OK 屏蔽中断状态。 如果在存储器中读取的所有字均与 DATA0 寄存器值相匹配，则在 MASSREAD 操作结束时此位置位。 写入 1 即可将该位清零。
位 3	**ILLCMD_MIS**：非法指令屏蔽中断状态。 当在 COMMAND 寄存器中写入了不良运算码指令时，此位置位。写入 1 即可将该位清零。
位 2	**CMDERR_MIS**：指令错误屏蔽中断状态。 当 Flash 忙碌时，如果在 COMMAND 寄存器中写入了指令运算码，则此位置位。写入 1 即可将该位清零。
位 1	**CMDSTART_MIS**：指令开始屏蔽中断状态。 在请求的指令开始执行后，此位置位。写入 1 即可将该位清零。
位 0	**CMDDONE_MIS**：指令完成屏蔽中断状态。 在请求的指令执行完毕后，此位置位。写入 1 即可将该位清零。

通过写入 COMMAND 寄存器启动请求的指令，几个时钟周期后更新 CMDDONE 和 CMDSTART 位。

注意： *通过写入 IRQSTAT（相应的 IRQRAW）寄存器将位清零，会同时将 IRQRAW（相应的 IRQSTAT）寄存器中的同一位清零，这是因为二者指向一个共同条件/事件。*

9.4.4 中断屏蔽寄存器（IRQMASK）

IRQMASK 中的屏蔽位屏蔽状态寄存器 IRQSTAT 中的状态并防止生成中断。

偏移地址：0x0C

复位值：0x0000 003F

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.	READOK M	ILLCMD M	CMDERRM	CMDSTART M	CMDDONE M
											rw	rw	rw	rw	rw

位 31:5	保留，必须保持复位值。
位 4	**READOKM**：批量读取 OK 屏蔽。 • 0：使能在发生“批量读取 OK”事件时中断 • 1：禁用在发生“批量读取 OK”事件时中断
位 3	**ILLCMDM**：非法指令屏蔽。 • 0：使能在发生“非法指令”事件时中断 • 1：禁用在发生“非法指令”事件时中断
位 2	**CMDERRM**：指令错误屏蔽。 • 0：使能在发生“指令错误”事件时中断 • 1：禁用在发生“指令错误”事件时中断
位 1	**CMDSTARTM**：指令启动屏蔽。 • 0：使能在发生“指令启动”事件时中断 • 1：禁用在发生“指令启动”事件时中断
位 0	**CMDDONEM**：指令完成屏蔽。 • 0：使能在发生“指令完成”事件时中断 • 1：禁用在发生“指令完成”事件时中断

9.4.5 原始状态寄存器（IRQRAW）

原始状态寄存器显示中断事件的未屏蔽状态。

偏移地址：0x10

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.	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.	READOK_RIS	ILLCMD_RIS	CMDERR_RIS	CMDSTART_RIS	CMDDONE_RIS
											rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:5	保留，必须保持复位值。
位 4	**READOK_RIS**：批量读取 OK 原始/未屏蔽中断状态。 如果在存储器中读取的所有字均与 DATA0 寄存器值相匹配，则在 MASSREAD 操作结束时此位置位。 写入 1 即可将该位清零。
位 3	**ILLCMD_RIS**：非法指令原始/未屏蔽中断状态。 当在 COMMAND 寄存器中写入了不良运算码指令时，此位置位。写入 1 即可将该位清零。
位 2	**CMDERR_RIS**：指令错误原始/未屏蔽中断状态。 当 Flash 忙碌时，如果在 COMMAND 寄存器中写入了指令运算码，则此位置位。写入 1 即可将该位清零。
位 1	**CMDSTART_RIS**：指令启动原始/未屏蔽中断状态。在请求的指令开始执行后，此位置位。 写入 1 即可将该位清零。
位 0	**CMDDONE_RIS**：指令完成原始/未屏蔽中断状态。在请求的指令执行完毕后，此位置位。写入 1 即可将该位清零。

通过写入 COMMAND 寄存器启动请求的指令，几个时钟周期后更新 CMDDONE 和 CMDSTART 位。

9.4.6 SIZE 寄存器

偏移地址：0x14

复位值：0x000- ---- （取决于器件）

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.	SWD_DISABLE	FLASH_SECURE	RAM_SIZE[1:0]		Res.
													r		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLASH_SIZE[15:0]															
r															

位 31:21	保留，必须保持复位值。
位 20	**SWD_DISABLE**：Flash+SWD 保护： • 0：无 SWD 保护（参考 FLASH_SECURE） • 1：Flash 和 SWD 保护
位 19	**FLASH_SECURE**：Flash 存储器保护： • 0：主 FLASH 无保护 • 1：通过客户密钥保护主 FLASH
位 18:17	**RAM_SIZE**：指示器件中可用 RAM 的大小： • 00：32 kB RAM 可用（RAM0 和 RAM1 存储库） • 01：32 kB RAM 可用（RAM0 和 RAM1 存储库） • 10：48 kB RAM 可用（RAM0、RAM1 和 RAM2 存储库） • 11：64 kB RAM 可用（RAM0、RAM1、RAM2 和 RAM3 存储库）
位 16	保留，必须保持复位值。
位 15:0	**FLASH_SIZE**：指示使用存储器元件地址格式的 Flash 的最后一个可使用地址。参见表 21. Flash 大小信息，了解 Flash 元件层面的地址与 AHB 地址映射之间的关系。 • 0x7FFF：该器件上可用的主 Flash 为 128 kB • 0xBFFF：该器件上可用的主 Flash 为 192 kB • 0xFFFF：该器件上可用的主 Flash 为 256 kB

表 21. Flash 大小信息

主 Flash 大小	Flash 层面的最高可使用地址(1)	AHB 层面的最高可使用地址
128 kB	0x7FFF	0x1005_FFFC
192 kB	0xBFFF	0x1006_FFFC
256 kB	0xFFFF	0x1007_FFFC

1. *FLASH_SIZE 位字段所示值。*

9.4.7 寄存器地址（ADDRESS）

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
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
XADDR[9:0]										YADDR[5:0]					
rw										rw					

位 31:16	保留，必须保持复位值。
位 15:6	**XADDR[9:0]**： • XADDR[9:3]：页面编号（0 至 127） • XADRR[2:0]：行编号（0 至 7）
位 5:0	**YADDR[5:0]**：选定行中的字编号（0 至 63）

要提供给 Flash 的地址不是 AHB 器件映射地址，而是遵循 Flash 元件格式的地址。

主 Flash 包含 128 个页面，每个页面有 8 行，每行 64 字 = 256 字节。

使用以下公式进行 ADDRESS 寄存器编程：

- XADDR[9:0] = AHB 地址位[17:8]
- YADDR[5:0] = AHB 地址位[7:2]

示例 1：在 AHB 地址 0x1005_0454 处进行一个字（32 位）的编程：

- XADDR[9:0] = AHB 地址位[17:8] = 0x104
- YADDR[5:0] = AHB 地址位[7:2] = 0x15
- ADDRESS 寄存器 = 0x4115

9.4.8 线性反馈移位寄存器（LFSRVAL）

LFSRVAL 寄存器包含 MASSREAD 指令发布的签名。

当 MASS READ 指令被写入 COMMAND 寄存器时，全部用 1 初始化 LFSRVAL 寄存器。然后，使每个读取值通过 LFSR。

在 CMDDONE 信息置位后，可在该寄存器中读取最终签名。

偏移地址：0x24

复位值：0xFFFF FFFF

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

位 31:0	**LSFRVAL**：MASSREAD 指令后签名，通过线性反馈生成。移位寄存器模块。

9.4.9 主 Flash 页面保护寄存器（PAGEPROTx）

PAGEPROTx 寄存器用于防止意外写入一组连续页面，下面称之为“段”。最多可定义 4 段。

用法示例见第 9.5.7 节：页面写保护示例。

PAGEPROT0

偏移地址：0x34

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEG1[15:0]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEG0[15:0]															
rw															

位 31:16	**SEG1**：第二段的定义。参见 SEG0 说明了解关于 SEG1[31:16]内容的详细信息。	
位 15:0	**SEG0**：第一段的定义。 SEGx 段的构成如下： • SEGx[15]：保留 • SEGx[14:8] = OFFSET：要开始写保护的页面编号（0 与 0x7F 之间的值） • SEGx[7]：保留 • SEGx[6:0] = SIZE：要保护的 2 kB 页面的数量，包括起始页（在 SEGx[14:8]中提供）	
	注意：	• *SIZE=0 意味着没有定义任何段，因此如果所有段的 SIZE=0，则不对整个 FLASH 应用写保护。* • *段可能重叠，如果至少有一段覆盖了页面，即可保证对该页面的保护。* • *如果 OFFSET + SIZE > 127d，超过了 FLASH 的最大大小，则段的末尾位于最大允许地址。*

PAGEPROT01

偏移地址：0x38

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEG3[15:0]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEG2[15:0]															
rw															

位 31:16	**SEG3**：第四段的定义。参见 PAGEPROT0 SEG0 说明了解关于 SEG3[15:0]内容的详细信息。
位 15:0	**SEG2**：第三段的定义。参见 PAGEPROT0 SEG0 说明了解关于 SEG2[15:0]内容的详细信息。

9.4.10 数据寄存器（DATA0-DATA3）

DATA0 寄存器需写入以下内容：

- 写入 Flash 位置的所需值（单次写入或批量写入）。
- （批量）读操作的所需比较值，标志 READOK 指示是否有匹配项。对于批量读取，所有读取值必须匹配才能使 READOK 置位。

只有突发写入才需要写入 DATA1-DATA3 寄存器。

DATA0

偏移地址：0x40

复位值：0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA0[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							DATA0[15:0]								
							rw								

位 31:0	**DATA0**：该寄存器有数种使用方式： • 要以单次写入模式写入 Flash 的数据 • 突发写入时要写入 Flash 的第一个数据 • MASSREAD 指令的比较值（仅在 Flash 写满相同字时有用） 注意：在最后这种情况下，标志 READOK 指示批量读取结束时是否有匹配项。

DATA1

偏移地址：0x44

复位值：0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA1[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							DATA1[15:0]								
							rw								

位 31:0	**DATA1**：在 BURSTWRITE 指令期间写入地址 ADDRESS+1 的数据。 注意：该寄存器只在突发写入时使用。

DATA2

偏移地址：0x48

复位值：0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA2[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							DATA2[15:0]								
							rw								

位 31:0	**DATA2**：在 BURSTWRITE 指令期间写入地址 ADDRESS+2 的数据。 注意：该寄存器只在突发写入时使用。

DATA3

偏移地址：0x4C

复位值：0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA3[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							DATA3[15:0]								
							rw								

位 31:0	**DATA3**：在 BURSTWRITE 指令期间写入地址 ADDRESS+3 的数据。 注意：该寄存器只在突发写入时使用。

9.5 编程模型

BlueNRG-LP 嵌入了最大 256 KB（65536 x 32 位）的内部 Flash 存储器。Flash 接口基于 AHB 协议实现指令存取和数据存取。它实现执行 Flash 存储器操作（编程/擦除）所必需的逻辑，这些操作通过 Flash 寄存器进行控制。

9.5.1 概述

对 Flash 的写操作只允许将位清零（从“1”到“0”）。这表示在执行写入前，任何从“0”到“1”的写操作都意味着擦除。

Flash 存储器包含 128 页，每页 8 行，每行 64 字（128 x 8 x 64 = 65536 字）。每个字的长度为 32 位 = 4 字节，即 256 kB Flash。

Flash 控制器 ADDRESS 寄存器中地址的构成如下：ADDRESS[15:0] = XADR[9:0] & YADR[5:0]，其中：

- XADR[9:3] = 页地址
- XADR[2:0] = 行地址
- YADR[5:0] = 字地址（一个字 = 四字节）

注意： *在两次擦除操作之间，一个特定地址只能写入两次，即使每次写入只是将位清零。*

9.5.2 读取功能示例

有两种可能的读访问：

- 读取一个字：简单读取，如同 SRAM 存储器：读取所需 Flash 地址并通过总线获取读取数据
- MASSREAD 指令：读取整个 Flash 存储器并与预期内容进行比较

有两种使用 MASSREAD 的方式：

- 完整 Flash 包含固定的 32 位模式：指示 Flash 控制器 DATA 寄存器中的预期模式（将值与 Flash 中的每个读取值进行比较），并在指令完成后检查 Flash 控制器中断寄存器中的 READOK 标志
- 否则：请求 MASSREAD 指令而不指定任何预期读取值，并在指令完成后检查 LFSRVAL 寄存器。该 LFSRVAL 寄存器包含存储器读取的签名

MASSREAD 序列：

- 在 Flash 控制器 DATA 寄存器中写入预期值（若结合 READOK 标志使用了 MASSREAD）
- 在 Flash 控制器 COMMAND 寄存器中写入 MASSREAD 指令（0x55）
- 等待 IRQSTAT 寄存器（轮询模式或中断模式）中的 CMDSTART 标志，该标志表示指令已生效且正在执行
- 通过在 Flash 控制器 IRQSTAT 寄存器中为 CMDSTART 写入“1”清除 CMDSTART 标志
- 然后等待 IRQSTAT 寄存器（轮询模式或中断模式）中的 CMDDONE 标志，该标志表示指令已完成
- 检查 IRQSTAT 寄存器中的 READOK 标志（预期高电平）或 LFSRVAL 寄存器值以确保 Flash 存储器内容为预期结果
- 通过在 Flash 控制器 IRQSTAT 寄存器中为 CMDDONE 写入“1”清除 CMDDONE 标志。

9.5.3 擦除功能示例

Flash 控制器允许擦除一个页面。

ERASE 序列（擦除一页）：

- 在 Flash 控制器寄存器 ADDRESS 中以下面的方式写入要擦除的页：
  - ADDRESS[15:9] = XADR[9:3] = 要擦除的页地址
  - ADDRESS[8:0] = 9'b0（行和字地址为 0）
- 在 Flash 控制器 COMMAND 寄存器中写入 ERASE 指令（0x11）
- 等待 IRQSTAT 寄存器（轮询模式或中断模式）中的 CMDSTART 标志，该标志表示指令已生效且正在执行
- 通过在 Flash 控制器 IRQSTAT 寄存器中为 CMDSTART 写入“1”清除 CMDSTART 标志
- 等待 IRQSTAT 寄存器（轮询模式或中断模式）中的 CMDDONE 标志，该标志表示指令已完成
- 通过在 Flash 控制器 IRQSTAT 寄存器中为 CMDDONE 写入“1”清除 CMDDONE 标志
- 执行该指令后，擦除的页只包含置为“1”的位。

9.5.4 Write 功能示例

Flash 控制器允许写入 1 个字（WRITE）、最多 4 个字（BURSWRITE）或整个主 Flash 存储器（有 1 个固定字）。

注意： *由于只有已置为“1”的位才能写入“0”，必须擦除页并请求将位置为“1”（而不是“0”），以便重写为“0”。*

WRITE 序列：

- 用目标地址（页、行和字编号）填充 Flash 控制器 ADDRESS 寄存器，以指示要写入的位置
- 将要设定的值写入 Flash 控制器 DATA 寄存器
- 在 WRITE 控制器 COMMAND 寄存器中写入 MASSREAD 指令（0x33）
- 等待 IRQSTAT 寄存器（轮询模式或中断模式）中的 CMDSTART 标志，该标志表示指令已生效且正在执行
- 通过在 Flash 控制器 IRQSTAT 寄存器中为 CMDSTART 写入“1”清除 CMDSTART 标志
- 等待 IRQSTAT 寄存器（轮询模式或中断模式）中的 CMDDONE 标志，该标志表示指令已完成
- 通过在 Flash 控制器 IRQSTAT 寄存器中为 CMDDONE 写入“1”清除 CMDDONE 标志。

BURSTWRITE 序列：

- 用要写入的第一个数据的目标地址（页、行和字编号）填充 Flash 控制器 ADDRESS 寄存器，以指示要写入的位置。在 ADDRESS 写入 DATA0，在 ADDRESS+1 写入 DATA1，以此类推。
- 将要设定的值写入 Flash 控制器 DATA0-3 寄存器。为了写入少于四个字，在未使用的 DATA1-3 寄存器中写入 0xFFFFFFFF
- 在 Flash 控制器 COMMAND 寄存器中写入 BURSTWRITE 指令（0xCC）
- 等待 IRQSTAT 寄存器（轮询模式或中断模式）中的 CMDSTART 标志，该标志表示指令已生效且正在执行
- 通过在 Flash 控制器 IRQSTAT 寄存器中为 CMDSTART 写入“1”清除 CMDSTART 标志
- 等待 IRQSTAT 寄存器（轮询模式或中断模式）中的 CMDDONE 标志，该标志表示指令已完成
- 通过在 Flash 控制器 IRQSTAT 寄存器中为 CMDDONE 写入“1”清除 CMDDONE 标志。

9.5.5 使能保护示例

器件提供三个级别的保护，以防应用程序克隆和/或修改应用程序代码。不同级别的描述见表 22. 系统存储器保护。必须注意的是，禁用 SWD 访问是不可逆操作。

表 22. 系统存储器保护

保护	Userkey[DATA0]	Userkey[DATA1]	禁用保护	说明
无	0xFFFFFFFF	0xFFFFFFFF	不适用	调试器可以访问和修改 Flash 存储器和 RAM 内容。这是默认配置。
读出	0xAAAAAAAA	0xAAAAAAAA	执行批量擦除	调试器不能读取或修改 Flash 存储器和 RAM 内容。
SWD	0xABACABAD	0xABACABAD	该选择不可逆	不能连接调试器。
未指定	任何其他值	任何其他值	不适用	此配置被禁用，可能导致不可恢复的器件损坏。

为了激活需要的保护级别，应使用以下 KEYWRITE 序列：

- 将要设定的值写入 DATA0（LSB 密钥）和 DATA1（MSB 密钥）寄存器
- 将 0xC7EF584D 写入 DATA2 并将 0xB3A21096 写入 DATA3
- 在 Flash 控制器 COMMAND 寄存器中写入 KEYWRITE 指令（0xFF）
- 等待 IRQSTAT 寄存器（轮询模式或中断模式）中的 CMDSTART 标志，该标志表示指令已生效且正在执行
- 通过在 Flash 控制器 IRQSTAT 寄存器中为 CMDSTART 写入“1”清除 CMDSTART 标志
- 等待 IRQSTAT 寄存器（轮询模式或中断模式）中的 CMDDONE 标志，该标志表示指令已完成
- 通过在 Flash 控制器 IRQSTAT 寄存器中为 CMDDONE 写入“1”清除 CMDDONE 标志。

复位后激活密钥（并保护 Flash 和 RAM 存储库）。

9.5.6 OTP 功能示例

OTPWRITE 序列：

- 将要设定的值写入 DATA0 寄存器（由于一次只写入几个字节，因此无突发写入特性可用）
- 按照以下规则写入 ADDRESS 寄存器：
  - ADDRESS[15:9] = 无意义（硬件根据该指令冻结的页面编号）
  - ADDRESS[8:6] = 0, 1, 2 or 3
  - ADDRESS[5:0] = 整个区域皆有可能（自 0x00 至 0x3F）
- 在 Flash 控制器 COMMAND 寄存器中写入 OTPWRITE 指令（0xEE）
- 等待 IRQSTAT 寄存器（轮询模式或中断模式）中的 CMDSTART 标志，该标志表示指令已生效且正在执行
- 通过在 Flash 控制器 IRQSTAT 寄存器中为 CMDSTART 写入“1”清除 CMDSTART 标志
- 等待 IRQSTAT 寄存器（轮询模式或中断模式）中的 CMDDONE 标志，该标志表示指令已完成
- 通过在 Flash 控制器 IRQSTAT 寄存器中为 CMDDONE 写入“1”清除 CMDDONE 标志。

注意： *OTP 位置遵循 Flash 存储器规则，即第二次写入只将位从 1 转换为 0。如果用户希望锁定 OTP 值并阻止 OTP 区域中的任何进一步修改，则必须用不同于 0xFFFFFFFF 的值写入最后一个 OTP 字（地址 0x10001BFC）并执行系统复位。锁定 OTP 区域的操作不可逆。*

9.5.7 页面写保护示例

以防止意外编程的写保护为例，已知 Flash 起始于地址 0x1004_0000 并包含 128 个页面（页面 0 至 127），每个页面 2 kB

- 地址范围 0x1004C000-0x1004FFFF

起始页面：0xC000/0x800 = 0x18

- SEG0[14:8] = 0x18（OFFSET = 0x18）

页面数量：(0x10000 - 0xC000）/0x800 = 0x8

- SEG0[6:0] = 0x8（SIZE = 0x8）
- 和地址范围 0x1005E000-0x1005FFFF

起始页面：0x1E000/0x800 = 0x3C

- SEG1[14:8] = 0x3C（OFFSET = 0x3C）

页面数量：(0x20000 - 0x1E000）/0x800 = 0x4

结论：设定 PAGEPROT0 = 0x3C041808。

10 DMA 控制器（DMA）

10.1 DMA 简介

使用直接存储器访问（DMA）是为了在外设与存储器之间以及存储器与存储器之间提供高速数据传输。可以在无需任何 CPU 操作的情况下通过 DMA 快速移动数据。这样节省的 CPU 资源可供其它操作使用。

DMA 有一个仲裁器，用于处理 DMA 请求之间的优先顺序。

10.2 DMA 主要特性

- 8 个可独立配置的通道（请求）
- 8 个通道中的每一个均连接至专用硬件 DMA 请求，每个通道还支持软件触发。此配置由软件完成
- 来自 DMA 通道的请求之间的优先级可通过软件来设定（4 个级别，包括极高、高、中和低），或者如果优先级相等，则通过硬件设定（请求 1 优先于请求 2 等）
- 独立的源和目标传输大小（字节、半字、字），模拟打包和解包。源/目标地址必须与数据大小相匹配
- 支持循环缓冲区管理
- 3 个事件标志（DMA 半传输、DMA 传输完成和 DMA 传输错误）在每个通道的单个中断请求中构成逻辑或关系
- 存储器至存储器的传输（SRAM0/SRAM1/SRAM2/SRAM3 而非 PKA RAM）
- 外设至存储器、存储器至外设以及外设至外设的传输
- 对作为源和目标的 SRAM 和 APB1 外设的访问
- 可编程的待传输数据量：最多 65536。

10.3 DMA 功能说明

DMA 控制器通过与芯片中其他总线主设备共享系统总线来执行直接存储器传输。当 DMA 与 CPU 访问目标（存储器或外设）相同时，DMA 请求会将 CPU 对系统总线的访问停止数个总线周期。总线矩阵执行循环调度，从而确保 CPU 至少可使用一半的系统总线带宽（无论对于存储器还是外设）。

10.3.1 DMA 传输

在产生事件后，外设会向 DMA 控制器发送请求信号。DMA 控制器根据通道优先级处理该请求。只要 DMA 控制器访问外设，DMA 控制器就会向外设发送确认信号。外设获得 DMA 控制器的确认信号后，便会立即释放其请求。在外设取消请求后，DMA 控制器将释放确认。如果有更多请求，外设可以启动下一个事务。概括起来，每个 DMA 传输包含三项操作：

- 从外设数据寄存器或通过内部当前外设/存储器地址寄存器寻址的存储器单元中加载数据。用于首次传输的起始地址是在 DMA_CPARx 寄存器或 DMA_CMARx 寄存器中编程的外设/存储器基址
- 将加载的数据存储到外设数据寄存器或通过内部当前外设/存储器地址寄存器寻址的存储器单元。用于首次传输的起始地址是在 DMA_CPARx 寄存器或 DMA_CMARx 寄存器中编程的外设/存储器基址
- DMA_CNDTRx 寄存器向后递减，该寄存器中包含仍需执行的事务数。

10.3.2 仲裁器

仲裁器根据通道请求的优先级管理通道请求，并启动外设/存储器访问序列。

优先级管理分为两个阶段：

- 软件：每个通道优先级都可以在 DMA_CCRx 寄存器中配置。分为四个级别：
  - 非常高优先级
  - 高优先级
  - 中优先级
  - 低优先级
- 硬件：如果两个请求具有相同的软件优先级，则编号低的通道优先于编号高的通道。例如，通道 2 的优先级高于通道 4

10.3.3 DMA 通道

各个通道均能处理外设寄存器（位于固定地址中）与存储器单元之间的 DMA 传输。待传输数据的数目可编程（最大 65535）。每个事务完成后，包含要传输的数据项总量的寄存器都会递减。

数据大小可编程

外设和存储器的传输数据大小可通过 DMA_CCRx 寄存器的 PSIZE 位和 MSIZE 位完全进行编程。

指针累加

根据 DMA_CCRx 寄存器的 PINC 位和 MINC 位的状态，可以选择使外设和存储器指针在每次传输后自动向后递增。如果使能了递增模式，则根据所选的数据大小，下次传输的地址即为前一次传输的地址加上 1、2 或 4。首次传输的地址可在 DMA_CPARx/DMA_CMARx 寄存器中进行编程。在传输过程中，这些寄存器将保持初始编程的值。软件无法获得当前传输的地址（位于当前内部外设或存储器地址寄存器中）。

如果将通道配置为非循环模式，则在最后一次传输完成后（即待传输的数据项数目达到零后），将不处理任何 DMA 请求。若需要将待传输数据项的新数目重新加载至 DMA_CNDTRx 寄存器中，则必须禁止 DMA 通道。

注意： *如果禁止 DMA 通道，DMA 寄存器不会被复位。在通道配置阶段，DMA 通道寄存器（DMA_CCRx、DMA_CPARx 和 DMA_CMARx）仍保留初始编程值。*

在循环模式下，最后一次传输完成后，DMA_CNDTRx 寄存器将自动重新加载初始编程值。当前的内部地址寄存器重新加载 DMA_CPARx/DMA_CMARx 寄存器中的基址值。

通道配置流程

配置 DMA 通道 x（其中 x 为通道编号）时应遵循下列顺序。

1. 在 DMA_CPARx 寄存器中设置外设寄存器地址。外设事件发生后，数据会从此地址移动到存储器或从存储器移动到此地址。
2. 设置 DMA_CMARx 寄存器中的存储器地址。外设事件发生后，将从此存储器读取数据或将数据写入此存储器。
3. 在 DMA_CNDTRx 寄存器中配置要传输的总数据数。每次发生外设事件后，该值都会递减。
4. 使用 DMA_CCRx 寄存器中的 PL[1:0] 位来配置通道优先级。
5. 在 DMA_CCRx 寄存器中配置数据传输方向、循环模式、外设和存储器递增模式、外设和存储器数据大小以及中断方式（数据传输一半产生中断还是数据完全传输后产生中断）。
6. 将 DMA_CCRx 寄存器中的 ENABLE 位置 1 以激活通道。

通道使能后，即可响应与该通道相连的外设所发出的所有 DMA 请求。

在数据传输一半后，传输一半标志（HTIF）置位，如果传输一半中断使能位（HTIE）置位，还会生成中断。传输结束时，传输完成标志（TCIF）便会置 1，如果传输完成中断使能位（TCIE）置 1，还会生成中断。

循环模式（Circular mode）

循环模式可用于处理循环缓冲区和连续数据流（例如 ADC 扫描模式）。可以使用 DMA_CCRx 寄存器中的 CIRC 位使能此特性。如果循环模式已激活，待传输数据的数目将自动加载为在通道配置阶段设置的初始值，并继续响应 DMA 请求。

存储器到存储器模式

DMA 通道在没有外设请求触发的情况下同样可以工作。此模式称为“存储器到存储器”模式。

如果 DMA_CCRx 寄存器中的 MEM2MEM 位置 1，则只要通过软件将 DMA_CCRx 寄存器中的使能位（EN）置 1，通道就会启动传输。DMA_CNDTRx 寄存器达到零后，传输停止。存储器到存储器模式不能与循环模式同时使用。

10.3.4 可编程数据宽度、数据对齐和字节序

当 PSIZE 与 MSIZE 不相等时，DMA 执行一些如“表 23. 可编程数据宽度和字节序行为（当 PINC=MINC=1 且 NDT=4 时）”所述的数据对齐。请注意，NDT 表示要传输的数据项的数量。

表 23. 可编程数据宽度和字节序行为（当 PINC=MINC=1 且 NDT=4 时）

端口宽度	源内容	传输操作	目标内容
源 => 目标	地址/数据		地址/数据
8 => 8	@0x0/B0 @0x1/B1 @0x2/B2 @0x3/B3	读取 B0[7:0] @0x0，然后写入 B0[7:0] @0x0 读取 B1[7:0] @0x1，然后写入 B1[7:0] @0x1 读取 B2[7:0] @0x2，然后写入 B2[7:0] @0x2 读取 B3[7:0] @0x3，然后写入 B3[7:0] @0x3	@0x0/B0 @0x1/B1 @0x2/B2 @0x3/B3
8 => 16	@0x0/B0 @0x1/B1 @0x2/B2 @0x3/B3	读取 B0[7:0] @0x0，然后写入 00B0[15:0] @0x0 读取 B1[7:0] @0x1，然后写入 00B1[15:0] @0x2 读取 B2[7:0] @0x2，然后写入 00B2[15:0] @0x4 读取 B3[7:0] @0x3，然后写入 00B3[15:0] @0x6	@0x0/00B0 @0x2/00B1 @0x4/00B2 @0x6/00B3
8 => 32	@0x0/B0 @0x1/B1 @0x2/B2 @0x3/B3	读取 B0[7:0] @0x0，然后写入 000000B0[31:0] @0x0 读取 B1[7:0] @0x1，然后写入 000000B1[31:0] @0x4 读取 B2[7:0] @0x2，然后写入 000000B2[31:0] @0x8 读取 B3[7:0] @0x3，然后写入 000000B3[31:0] @0xC	@0x0/000000B0 @0x4/000000B1 @0x8/000000B2 @0xC/000000B3
16 => 8	@0x0/B1B0 @0x2/B3B2 @0x4/B5B4/ @0x6/B7B6	读取 B1B0[15:0] @0x0，然后写入 B0[7:0] @0x0 读取 B3B2[15:0] @0x2，然后写入 B2[7:0] @0x1 读取 B5B4[15:0] @0x4，然后写入 B4[7:0] @0x2 读取 B7B6[15:0] @0x6，然后写入 B6[7:0] @0x3	@0x0/B0 @0x1/B2 @0x2/B4 @0x3/B6
16 => 16	@0x0/B1B0 @0x2/B3B2 @0x4/B5B4/ @0x6/B7B6	读取 B1B0[15:0] @0x0，然后写入 B1B0[15:0] @0x0 读取 B3B2[15:0] @0x2，然后写入 B3B2[15:0] @0x2 读取 B5B4[15:0] @0x4，然后写入 B5B4[15:0] @0x4 读取 B7B6[15:0] @0x6，然后写入 B7B6[15:0] @0x6	@0x0/B1B0 @0x2/B3B2 @0x4/B5B4 @0x6/B7B6
16 => 32	@0x0/B1B0 @0x2/B3B2 @0x4/B5B4/ @0x6/B7B6	读取 B1B0[15:0] @0x0，然后写入 0000B1B0[31:0] @0x0 读取 B3B2[15:0] @0x2，然后写入 0000B3B2[31:0] @0x4 读取 B5B4[15:0] @0x4，然后写入 0000B5B4[31:0] @0x8 读取 B7B6[15:0] @0x6，然后写入 0000B7B6[31:0] @0xC	@0x0/0000B1B0 @0x4/0000B3B2 @0x8/0000B5B4 @0xC/0000B7B6
32 => 8	@0x0/B3B2B1B0 @0x4/B7B6B5B4 @0x8/BBBAB9B8 @0xC/BFBEBDBC	读取 B3B2B1B0[31:0] @0x0，然后写入 B0[7:0] @0x0 读取 B7B6B5B4[31:0] @0x4，然后写入 B4[7:0] @0x1 读取 BBBAB9B8[31:0] @0x8，然后写入 B8[7:0] @0x2 读取 BFBEBDBC[31:0] @0xC，然后写入 BC[7:0] @0x3	@0x0/B0 @0x1/B4 @0x2/B8 @0x3/BC

端口宽度	源内容	传输操作	目标内容
源 => 目标	地址/数据		地址/数据
32 => 16	@0x0/B3B2B1B0 @0x4/B7B6B5B4 @0x8/BBBAB9B8 @0xC/BFBEBDBC	读取 B3B2B1B0[31:0] @0x0，然后写入 B1B0[15:0] @0x0 读取 B7B6B5B4[31:0] @0x4，然后写入 B5B4[15:0] @0x2 读取 BBBAB9B8[31:0] @0x8，然后写入 B9B8[15:0] @0x4 读取 BFBEBDBC[31:0] @0xC，然后写入 BDBC[15:0] @0x6	@0x0/B1B0 @0x2/B5B4 @0x4/B9B8 @0x6/BDBC
32 => 32	@0x0/B3B2B1B0 @0x4/B7B6B5B4 @0x8/BBBAB9B8 @0xC/BFBEBDBC	读取 B3B2B1B0[31:0] @0x0，然后写入 B3B2B1B0[31:0] @0x0 读取 B7B6B5B4[31:0] @0x4，然后写入 B7B6B5B4[31:0] @0x4 读取 BBBAB9B8[31:0] @0x8，然后写入 BBBAB9B8[31:0] @0x8 读取 BFBEBDBC[31:0] @0xC，然后写入 BFBEBDBC[31:0] @0xC	@0x0/B3B2B1B0 @0x4/B7B6B5B4 @0x8/BBBAB9B8 @0xC/BFBEBDBC

不支持字节或半字写操作的 AHB 外设的寻址

当 DMA 发起 AHB 字节或半字写操作时，在 HWDATA[31:0]总线的未使用通道上复制数据。因此，当使用的 AHB 从外设不支持字节或半字写操作（当外设不使用 HSIZE 时）且不生成任何错误时，DMA 写入 32 个 HWDATA 位，如下面的两个示例所示：

- 要写入半字“0xABCD”，则 DMA 会将 HWDATA 总线设为“0xABCDABCD”，同时将 HSIZE 设为 HalfWord
- 要写入字节“0xAB”，则 DMA 会将 HWDATA 总线设为“0xABABABAB”，同时将 HSIZE 设为 Byte

假设 AHB/APB 桥为 AHB 32 位从外设，且该外设未设置数据的 HSIZE，则任何 AHB 字节或半字操作都将转换为 32 位 APB 操作，转换方式如下所示：

- 将数据“0xB0”向 0x0（或 0x1、0x2、0x3）的 AHB 字节写操作转化为数据“0xB0B0B0B0”向 0x0 的 APB 字写操作
- 将数据“0xB1B0”向 0x0（或 0x2）的 AHB 半字写操作转化为数据“0xB1B0B1B0”向 0x0 的 APB 字写操作

例如，如需写入 APB 备份寄存器（分配给 32 位地址边界的 16 位寄存器），您必须将存储器源大小（MSIZE）配置为“16 位”，并将外设目标大小（PSIZE）配置为“32 位”。

10.3.5 错误管理

当对保留的地址空间执行读取或写入操作时，将生成 DMA 传输错误。若在 DMA 读或写访问过程中生成了 DMA 传输错误，则会将相应的通道配置寄存器（DMA_CCRx）的 EN 位进行硬件清零，从而自动禁止出错的通道。如果 DMA_CCRx 寄存器中的传输错误中断使能位（TEIE）置位，则 DMA_IFR 寄存器中该通道的传输错误中断标志（TEIF）将置位并生成中断。

10.3.6 中断

对于每个 DMA 通道，在发生“半传输”、“传输完成”或“传输错误”时都可以产生中断。可以使用单独的中断使能位以提高灵活性。

表 24. DMA 中断请求

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

10.3.7 DMA 请求映射

BlueNRG-LP 器件中有 DMAMUX，用于选择将哪个请求者连接到哪个 DMA 通道。参见表 27. DMAMUX 映射了解详细信息。

10.4 DMA 寄存器

请参见第 1.3 节“缩略语”获取寄存器说明中所用缩略语的列表。外设寄存器必须按字（32 位）进行访问。

10.4.1 DMA 中断状态寄存器（DMA_ISR）

偏移地址：0x000

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TEIF8	*HTIF8*	*TCIF8*	*GIF8*	*TEIF7*	*HTIF7*	*TCIF7*	*GIF7*	*TEIF6*	*HTIF6*	*TCIF6*	*GIF6*	*TEIF5*	*HTIF5*	*TCIF5*	*GIF5*
r	r	r	r	r	r	r	r	r	r	r	r	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，必须保持复位值。
位 31、27、23、19、15、11、7、3	**TEIFx**：通道 x 传输错误标志（x = 1..8）。 此位将由硬件置 1。由软件清零，软件只需将 1 写入 DMA_IFCR 寄存器的相应位。 0：通道 x 上无传输错误（TE） 1：通道 x 上出现传输错误（TE）
位 30、26、22、18、14、10、6、2	**HTIFx**：通道 x 半传输标志（x = 1..8）。 此位将由硬件置 1。由软件清零，软件只需将 1 写入 DMA_IFCR 寄存器的相应位。 0：通道 x 上无半传输（HT）事件 1：通道 x 上发生半传输（HT）事件
位 29、25、21、17、13、9、5、1	**TCIFx**：通道 x 传输完成标志（x = 1..8）。 此位将由硬件置 1。由软件清零，软件只需将 1 写入 DMA_IFCR 寄存器的相应位。 0：通道 x 上无传输完成（TC）事件 1：通道 x 上发生了传输完成（TC）事件
位 28、24、20、16、12、8、4、0	**GIFx**：通道 x 全局中断标志（x = 1..8）。 此位将由硬件置 1。由软件清零，软件只需将 1 写入 DMA_IFCR 寄存器的相应位。 0：通道 x 上无 TE、HT 或 TC 事件 1：通道 x 上发生 TE、HT 或 TC 事件

10.4.2 DMA 中断标志清零寄存器（DMA_IFCR）

偏移地址：0x004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CTEIF8	CHTIF8	CTCIF8	CGIF8	CTEIF7	CHTIF7	CTCIF7	CGIF7	CTEIF6	CHTIF6	CTCIF6	CGIF6	CTEIF5	CHTIF5	CTCIF5	CGIF5
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
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，必须保持复位值。
位 31、27、23、19、15、11、7、3	**CTEIFx**：通道 x 传输错误清零（x = 1..8）。此位由软件置 1 和清零。 0：无操作 1：清除 DMA_ISR 寄存器中相应的 TEIF 标志
位 30、26、22、18、14、10、6、2	**CHTIFx**：通道 x 半传输清零（x = 1..8）。此位由软件置 1 和清零。 0：无操作 1：清除 DMA_ISR 寄存器中相应的 HTIF 标志
位 29、25、21、17、13、9、5、1	**CTCIFx**：通道 x 传输完成清零（x = 1..8）。此位由软件置 1 和清零。 0：无操作 1：清除 DMA_ISR 寄存器中相应的 TCIF 标志
位 28、24、20、16、12、8、4、0	**CGIFx**：通道 x 全局中断清零（x = 1..8）。此位由软件置 1 和清零。 0：无操作 1：清除 DMA_ISR 寄存器中的 GIF、TEIF、HTIF 和 TCIF 标志

10.4.3 DMA 通道 x 配置寄存器（DMA_CCRx）（x = 1..8，其中的 x = 通道编号）

偏移地址：0x008 + 0d20 ×（通道编号 - 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.	MEM2 MEM	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，必须保持复位值。
位 14	**MEM2MEM**：存储器到存储器模式。此位由软件置 1 和清零。 0：禁止存储器到存储器模式 1：使能存储器到存储器模式

位 13:12	**PL[1:0]**：通道优先级。 这些位将由软件置 1 和清零。 00：低 01：中 10：高 11：极高
位 11:10	**MSIZE[1:0]**：存储器容量。 这些位将由软件置 1 和清零。 00：8 位 01：16 位 10：32 位 11：保留
位 9:8	**PSIZE[1:0]**：外设容量。 这些位将由软件置 1 和清零。 00：8 位 01：16 位 10：32 位 11：保留
位 7	**MINC**：存储器递增模式。 此位由软件置 1 和清零。 0：存储器递增模式禁用 1：存储器递增模式使能
位 6	**PINC**：外设递增模式。此位由软件置 1 和清零。 0：外设递增模式禁用 1：外设递增模式使能
位 5	**CIRC**：循环模式。 此位由软件置 1 和清零。 0：禁止循环模式 1：使能循环模式
位 4	**DIR**：数据传输方向。 此位由软件置 1 和清零。 0：从外设读取 1：从存储器读取
位 3	**TEIE**：传输错误中断使能。此位由软件置 1 和清零。 0：禁止 TE 中断 1：TE 中断启用
位 2	**HTIE**：半传输中断使能。此位由软件置 1 和清零。 0：禁止 HT 中断 1：HT 中断启用
位 1	**TCIE**：传输完成中断使能。此位由软件置 1 和清零。 0：禁止 TC 中断 1：TC 中断启用
位 0	**EN**：通道使能。 此位由软件置 1 和清零。 0：禁止通道 1：使能通道

10.4.4 DMA 通道 x 数据数量寄存器（DMA_CNDTRx）（x = 1..8，其中的 x = 通道编号）

偏移地址：0x00C + 0d20 ×（通道编号 - 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
NDT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16	保留，必须保持复位值。
位 15:0	**NDT[15:0]**：要传输的数据的数量。 待传输的数据数（0 到 65535）。只有在通道禁止时才能写入此寄存器。通道使能后，该寄存器为只读，用于指示待传输的剩余字节数。每次 DMA 传输后，此寄存器将递减。 传输完成后，该寄存器的值可以保持为零，若已将通道配置为自动重载模式，则自动重新加载先前编程的值。 若该寄存器的值为零，则无论通道是否使能，都将不会处理任何事务。

10.4.5 DMA 通道 x 外设地址寄存器（DMA_CPARx）（x = 1..8，其中的 x = 通道编号）

偏移地址：0x010 + 0d20 ×（通道编号 - 1）

复位值：0x0000 0000

当通道使能时，不得写入此寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PA [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
PA [15:0]															
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。 读/写数据的外设数据寄存器的基址。若 PSIZE 为 01（16 位），则忽略 PA[0] 位。访问将自动对齐到半字地址。 若 PSIZE 为 10（32 位），则忽略 PA[1:0]。访问将自动对齐到字地址。

10.4.6 DMA 通道 x 存储器地址寄存器（DMA_CMARx）（x = 1..8，其中的 x = 通道编号）

偏移地址：0x014 + 0d20 ×（通道编号 - 1）

复位值：0x0000 0000

当通道使能时，不得写入此寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MA [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
MA [15:0]															
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。 读/写数据的存储区基址。 若 MSIZE 为 01（16 位），则忽略 MA[0] 位。访问将自动对齐到半字地址。 若 MSIZE 为 10（32 位），则忽略 MA[1:0]。访问将自动对齐到字地址。

10.5 DMA 寄存器映射

下表列出了 DMA 寄存器映射和复位值。

表 25. DMA 寄存器映射和复位值

偏移	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
0x000	**DMA_ISR**	TEIF8	HTIF8	TCIF8	GIF8	TEIF7	HTIF7	TCIF7	GIF7	TEIF6	HTIF6	TCIF6	GIF6	TEIF5	HTIF5	TCIF5	GIF5	TEIF4	HTIF4	TCIF4	GIF4	TEIF3	HTIF3	TCIF3	GIF3	TEIF2	HTIF2	TCIF2	GIF2	TEIF1	HTIF1	TCIF1	GIF1
	复位值	0	0	0	0	0	0	0	0	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**	CTEIF8	CHTIF8	CTCIF8	CGIF8	CTEIF7	CHTIF7	CTCIF7	CGIF7	CTEIF6	CHTIF6	CTCIF6	CGIF6	CTEIF5	CHTIF5	CTCIF5	CGIF5	CTEIF4	CHTIF4	CTCIF4	CGIF4	CTEIF3	CHTIF3	CTCIF3	CGIF3	CTEIF2	CHTIF2	CTCIF2	CGIF2	CTEIF1	CHTIF1	CTCIF1	CGIF1
	复位值	0	0	0	0	0	0	0	0	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
	复位值																		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]															
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x010	**DMA_CPAR1**	PA[31:0]																															
	复位值	0	0	0	0	0	0	0	0	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]																															
	复位值	0	0	0	0	0	0	0	0	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
	复位值															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]															
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x024	**DMA_CPAR2**	PA[31:0]																															
	复位值	0	0	0	0	0	0	0	0	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]																															
	复位值	0	0	0	0	0	0	0	0	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
	复位值															0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

偏移	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
0x034	**DMA_CNDTR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDT[15:0]															
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x038	**DMA_CPAR3**	PA[31:0]																															
	复位值	0	0	0	0	0	0	0	0	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]																															
	复位值	0	0	0	0	0	0	0	0	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
	复位值															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]															
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04C	**DMA_CPAR4**	PA[31:0]																															
	复位值	0	0	0	0	0	0	0	0	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]																															
	复位值	0	0	0	0	0	0	0	0	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	保留																																
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
	复位值																		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]															
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x060	**DMA_CPAR5**	PA[31:0]																															
	复位值	0	0	0	0	0	0	0	0	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]																															
	复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x068	保留																																
0x06C	**DMA_CCR6**	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
	复位值															0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

偏移	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
0x070	**DMA_CNDTR6**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDT[15:0]															
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x074	**DMA_CPAR6**	PA[31:0]																															
	复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x078	**DMA_CMAR6**	MA[31:0]																															
	复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x07C	保留																																
0x080	**DMA_CCR7**	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
	复位值															0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x084	**DMA_CNDTR7**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDT[15:0]															
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x088	**DMA_CPAR7**	PA[31:0]																															
	复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08C	**DMA_CMAR7**	MA[31:0]																															
	复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x090	保留																																
0x094	**DMA_CCR8**	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
	复位值															0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x088	**DMA_CNDTR8**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDT[15:0]															
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x09C	**DMA_CPAR8**	PA[31:0]																															
	复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0A0	**DMA_CMAR8**	MA[31:0]																															
	复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
x0A4 至 0x31C	保留																																

请参见第 2.2.2 节“存储器映射和寄存器边界地址”了解寄存器边界地址。

11 DMA 请求复用器（DMAMUX）

11.1 引言

外设通过设置其 DMA 请求信号来指示 DMA 传输请求。DMA 控制器会处理 DMA 请求并生成 DMA 确认信号，而且相应的 DMA 请求信号也将变为无效，但在此之前 DMA 请求一直处于挂起状态。

为简单起见，本文档中提取了 DMA 请求/确认协议的功能描述及其相关控制信号，并将它们统称为 DMA 请求线路。未在图中显示或在文本中描述 DMA 控制器响应信号。

DMAMUX 请求复用器可将外设的 DMA 请求线路由到产品 DMA 控制器。该路由功能通过可编程的多通道 DMA 请求线复用器来确保实现。每个通道选择一个唯一的 DMA 请求线路。

11.2 DMAMUX 主要特性

- 8 通道可编程 DMA 请求线路复用器输出
- 每个 DMA 请求线路复用器通道输出：
  - 来自外设的 16 个输入 DMA 请求线路
  - 1 个 DMA 请求线路输出

11.3 DMAMUX 实现

11.3.1 DMAMUX 实例化

用以下硬件配置参数实例化 DMAMUX。

表 26. DMAMUX 实例化

功能	DMAMUX
DMAMUX 输出请求通道的数量	8
DMAMUX 请求生成器通道的数量	16
DMAMUX 请求触发信号输入的数量	2
DMAMUX 同步输入的数量	2
DMAMUX 外设请求输入的数量	1

11.3.2 DMAMUX 映射

资源到 DMAMUX 的映射为硬接线。

表 27. DMAMUX 映射

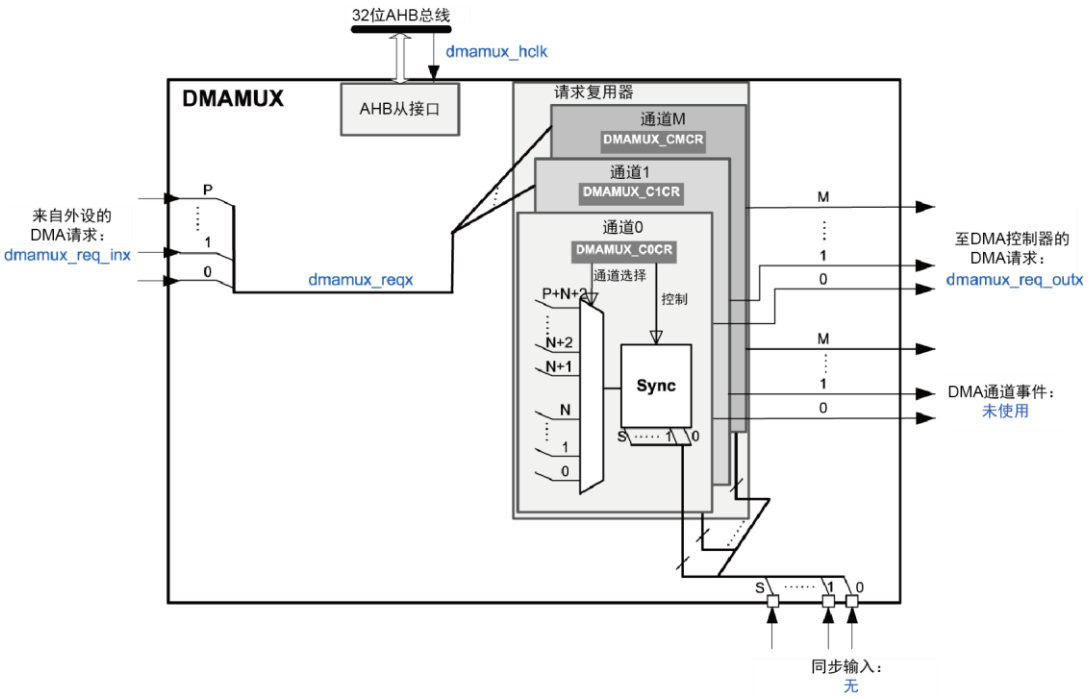
DMA 请求 MUX 输入	资源	DMA 请求 MUX 输入	资源
1	保留	10	I2C2_RX
2	SPI3_RX	11	I2C2_TX
3	SPI3_TX	12	USART_RX
4	SPI1_RX	13	USART_TX
5	SPI1_TX	14	LPUART_RX
6	SPI2_RX	15	LPUART_TX
7	SPI2_TX	16	ADC_CH0（DS 通道）
8	I2C1_RX	17	ADC_CH1（DF 通道）
9	I2C1_TX		

11.4 DMAMUX 功能说明

11.4.1 DMAMUX 框图

图 21. DMAMUX 框图所示为 DMAMUX 框图。

图 21. DMAMUX 框图



该实现分配：

- 来自外设（dmamux_req_inx）的 DMAMUX 请求复用器子模块输入（dmamux_reqx）
- 至 DMA 控制器通道的 DMAMUX 请求输出（dmamux_req_outx）

11.4.2 DMAMUX 通道

DMAMUX 通道是可能包含的 DMAMUX 请求复用器通道，具体取决于选定的请求复用器输入。

一个 DMAMUX 请求复用器通道连接到并专用于一个 DMA 控制器通道。

通道配置流程

配置 DMAMUX x 通道和相关 DMA 通道 y 时应遵循下列顺序：

1. 完整地设置和配置 DMA 通道 y，使能通道 y 除外。
2. 完整地设置和配置相关 DMAMUX y 通道。
3. 将 DMA y 通道寄存器中的 EN 位置 1 以激活 DMA 通道。

11.4.3 DMAMUX 请求线路复用器

DMAMUX 请求复用器及其多个通道确保 DMA 请求/确认控制信号的实际路由（称为 DMA 请求线路）。

每条 DMA 请求线路都与 DMAMUX 请求线路复用器的所有通道并联。DMA 请求源自外设。

DMAMUX 请求线路复用器通道 x 按照 DMAMUX_CxCR 寄存器的 8 位 DMAREQ_ID 字段中的配置选择 DMA 请求线路编号。

注意： *DMAREQ_ID 字段中的空值表示没有选择任何 DMA 请求线路。不应将同一个非空 DMA_REQ_ID 值设定为不同的 x 和 y DMAMUX 请求复用器通道（通过 DMAMUX1_CxCR 和 DMAMUX CyCR）。不允许将同一个非空 DMAREQ_ID 配置给 DMAMUX 请求线路复用器的两个不同通道。*

除了 DMA 请求选择，如有必要，还可以配置和使能同步模式和/或事件生成。

11.5 DMAMUX 寄存器

请参见寄存器边界地址的相关表格了解 DMAMUX 基址。寄存器只能按字（32 位）访问。

11.5.1 DMAMUX 请求线路复用器通道 x 配置寄存器（DMAMUX_CxCR）

偏移地址：0x04 * x（x = 0 至）

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
RESERVED											DMAREQ_ID[4:0]				
											rw				

位 31:5	保留，必须保持复位值。
位 4:0	**DMAREQ_ID[4:0]**：DMA 请求 ID。 选择输入 DMA 请求。请参见 DMAMUX 表格中关于将复用器输入分配给资源的信息。

11.6 DMAMUX 寄存器映射

下表汇总了 DMAMUX 寄存器和复位值。有关 DMAMUX 寄存器的基本地址，请参见寄存器边界地址表。

表 28. DMAMUX 寄存器映射和复位值

偏移	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
0x000	DMAMUX_C0CR	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	DMAREQ_ID[4:0]				
	复位值																												0	0	0	0	0
0x004	DMAMUX_C1CR	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	DMAREQ_ID[4:0]				
	复位值																												0	0	0	0	0
0x008	DMAMUX_C2CR	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	DMAREQ_ID[4:0]				
	复位值																												0	0	0	0	0
0x00C	DMAMUX_C3CR	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	DMAREQ_ID[4:0]				
	复位值																												0	0	0	0	0
0x010	DMAMUX_C4CR	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	DMAREQ_ID[4:0]				
	复位值																												0	0	0	0	0
0x014	DMAMUX_C5CR	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	DMAREQ_ID[4:0]				
	复位值																												0	0	0	0	0

偏移	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
0x018	DMAMUX_C6CR	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	DMAREQ_ID[4:0]				
	复位值																												0	0	0	0	0
0x01C	DMAMUX_C7CR	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	DMAREQ_ID[4:0]				
	复位值																												0	0	0	0	0
0x020 0x3E8	保留	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

12 模数转换器（ADC）

BlueNRG-LP 中嵌入了 12 位 ADC。ADC 包含一个 12 位连续近似模数转换器（SAR），该 SAR 具有 2 x 8 个复用通道，可测量至多 8 个外部源和至多 2 个内部源。

12.1 特性

- 转换频率最高 1 Msps
- 支持 3 个输入电压范围（0 → 1.2 V, 0 → 2.4 V, 0 → 3.6 V）
- 至多 8 个模拟单端通道或 4 个模拟差分输入或二者兼有
- 通过在模拟模式下配置的两个 GPIO 支持一个模拟麦克风（一个模拟麦克风输入和一个模拟麦克风 Vbias 输出）
- 温度传感器转换
- 电池电压转换（最高 3.6 V）
- 连续或单次采集
- 数字抽取滤波器，用于处理 2 个 GPIO 提供的数字音频 PDM 流，并用于 ADC 后处理，特别是对于模拟音频流
- 可以有 5 种转换模式：
  - ADC 连续或单脉冲模式
  - 模拟连续音频模式
  - 偶发转换
  - 数字连续音频模式
  - 完全模式
- 适用于多用途应用的 ADC 下采样器，在减轻 CPU 负载的同时改善模拟性能（比率可调节，范围为 1 至 128）
- 看门狗特性，在数据超出阈值时发出通知（对除数字音频模式外的所有模式可用）
- 与 DMA 相连
- 具有标志位的中断源。

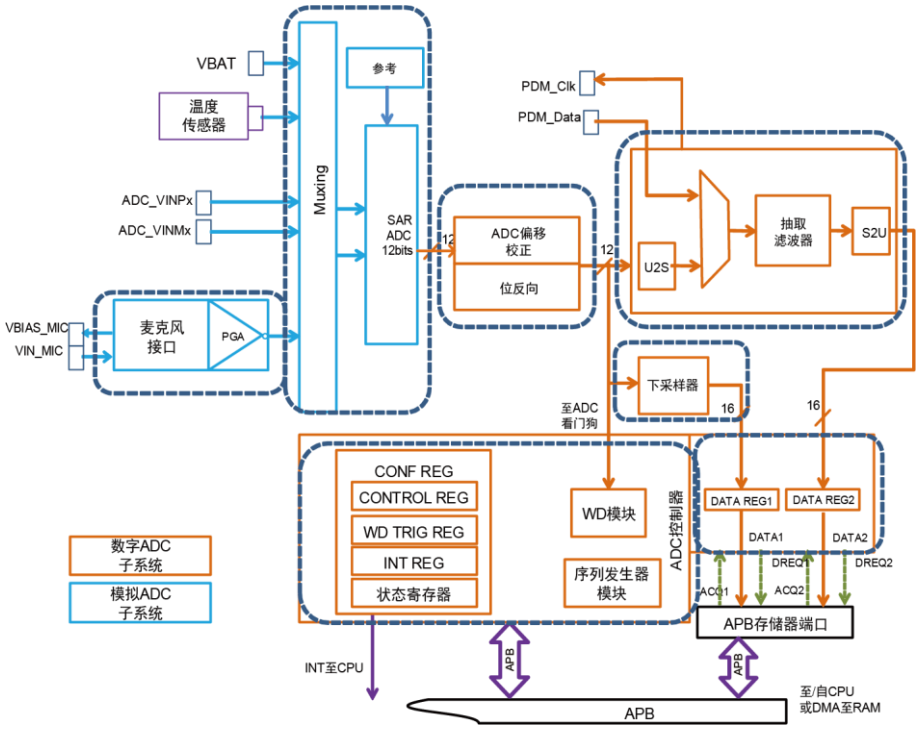
12.2 ADC 简介

图 22. ADC 顶层图所示为 ADC 的顶层图。

模拟 ADC 可配置为与以下输入相连：

- 外部信号，通过 ADC_VINPx 和 ADC_VINMx，其中的 x=0、1、2 或 3
  - 至多 4 个差分输入
  - 至多 8 个单端输入
- 模拟麦克风接口
- 温度传感器电池电压检测器（最高 3.6 V）
- 电池电压检测器（最高 3.6 V）

图 22. ADC 顶层图



除了模拟 ADC，还可支持数字麦克风（PDM 接口，通过 2 个 GPIO）。

数据路径的输入可能来自：

- 模拟 ADC
- 数字麦克风信号

转换数据路径可经过：

- 抽取滤波器
- 下采样器（用于静态或低频输入信号）

这两条不同数据路径的存在使一些并发转换（特定组合）成为可能。

注意：不要更改与正在使用的功能相关的配置寄存器。用户对不同的位执行的任何更改都会立即应用，并立即作用于正在进行的处理（转换、抽取滤波器或下采样器）。此操作可能导致意外结果。

如果 VBAT < 2.7 V，则需要激活 IO BOOSTER 以维持线性。

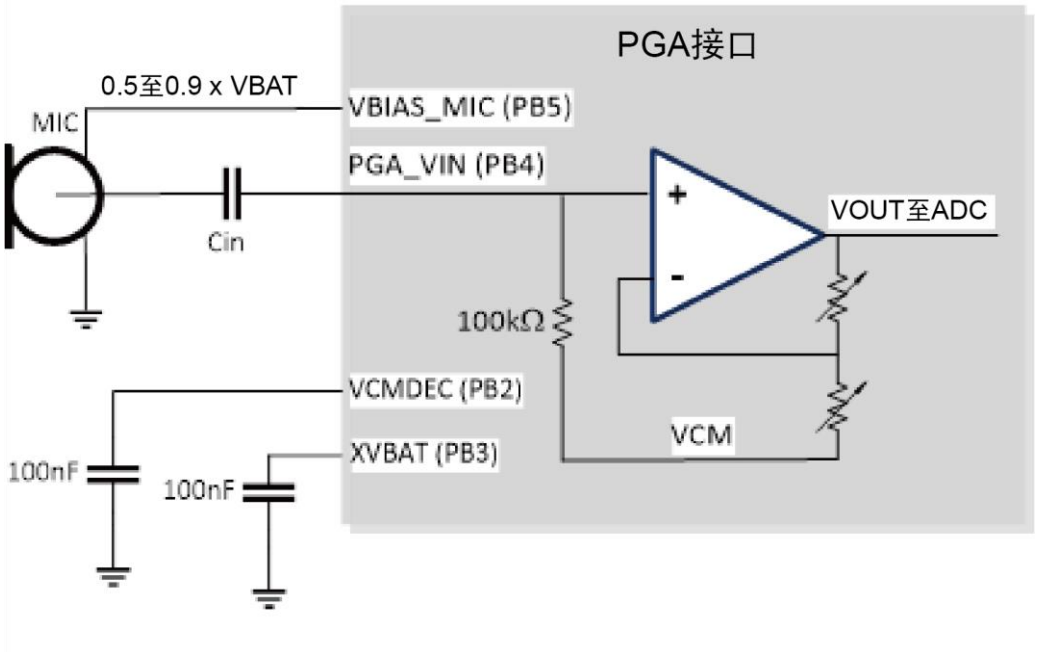
12.2.1 可编程增益放大器（PGA）

来自模拟麦克风的输入信号由可编程增益放大器（PGA）（参见“图 23. 麦克风设置”）以 3 dB 的步长从 0 dB 放大至 30 dB。

然后，低通滤波器（在 20khz 衰减 1dB）对信号进行滤波。

PGA 输出电压为 1.2 V，与 VIN_MIC 输入信号反相。

图 23. 麦克风设置



VBIAS_MIC，其功能独立于 PGA_VIN，用作麦克风电源。为了适应不同的麦克风供电规格，电池电压可在 1.7 V 至 3.6 V 之间变化，VBIAS MIC 电压可按与 VBAT 之比（0.5 至 0.9）进行调节。请参见关于 PGA_CONF 寄存器的章节了解详细信息。

需将 100 nF 电容分别连接到 VCMDEC 和 XVBAT，以便过滤掉高于 1 kHz 的噪声。

PGA_VIN 具有典型的 750 mV 输入共模电压（VCM）。为避免可能导致 PGA 中发生饱和的 DC 冲突，需在麦克风和 PGA_VIN 之间连接耦合电容 Cin。推荐值为 1 µF，但可以使用更小的值（参见表 29. PGA 参数）。电容越小，启动时间越短，但缺点是输入信号会衰减。例如，当 Cin 为 10 nF 时，1 kHz 信号会衰减约 1 dB。

PGA 增益可在范围 0 dB 至 30 dB 以内以 3 dB 的步长进行调节（参见第 12.6.5 节“ADC PGA 配置寄存器（PGA_CONF）”）。为避免 PGA 中发生失真，需根据麦克风输出峰值电压调节增益设置。下面是需要遵守的 PGA 最大输出峰值电压值，以及估算关于麦克风灵敏度 S0 的 PGA 增益公式。

表 29. PGA 参数

参数	符号	最小值	典型值	单位	备注
Cin	Cin	0.1	1	µF	推荐值为 1 µF
PGA Max Vout	MxVout	-6.8	-5.8	dBVpk	25°C 时为典型值；105°C 时为最小值

注意： *最大 PGA 增益（单位：dB）= MxVout – So – 3 [注意模拟麦克风的灵敏度单位：dBVrms]*

注意： *例如，当麦克风灵敏度 = -38 dB，采用典型设置且输入信号无衰减时：最大 PGA 增益 = -5.8 + 38 – 3 = 29.2 dB。*

注意： *在模拟音频模式下使用 PGA 意味着在 PB2/PB3/PB4 I/O 上连接 3 个外部电容。用户必须确保模拟开关被适当地配置为选择 PGA 外部电容模式（参见第 8.2.11 节“I/O 模拟开关控制寄存器（GPIO_SWA_CTRL）”了解详细信息），然后这两个焊盘按此顺序被设定为模拟模式（参见第 7.4.1 节“GPIO 端口模式寄存器（GPIOx_MODER）（x = A, B）”）。为了过滤 DC 电平，需要为抽取滤波器的输出添加滤波器。*

12.2.2 温度传感器子系统

温度传感器可用于测量器件的结温（T_j）。温度传感器内部连接到 ADC 输入通道，该通道用于将传感器输出电压转换为数字值。

温度传感器需通过 PWRC_CR2.ENTS 位使能。

温度传感器测量值转换在输入范围 0 至 1.2 V 以内进行，温度范围为：

- 对于最低温度 = -40°C，Vmin = 0.585 V
- 对于最高温度 = +125°C，Vmax = 0.999 V

温度传感器的输出电压随温度线性变化。由于制程变异，不同芯片间该线路的偏移存在差异。未校准的内部温度传感器更适用于对温度变量而非绝对温度进行测量的应用情况。为了提高温度传感器的测量精度，ST 于生产过程中在系统内存中存储了每个器件的校准值。在制造过程中，温度传感器的校准数据和内部电压参考信息被存储在系统存储区。然后，用户应用程序可以读取数据并使用它们来提高温度传感器或内部参考的精度。这样一来，就可以使用以下公式计算温度：

$$\text{温度（摄氏度）} = \frac{55}{C85 - C30} \cdot [Cmeas - C30] + 30 \quad (1)$$

其中：

C85 是在 85°C 时获得的温度传感器校准值，可从地址@0x10001E68 读取。

C30 是在 30°C 时获得的温度传感器校准值，可从地址@0x10001E60 读取。

Cmeas 是 ADC 转换的实际温度传感器输出值。

注意： *必须应用仅适用于 VINPx 范围 1.2 V 的 ADC 增益校准。偏移校准必须为 0（未使用）。*

12.2.3 电池传感器

电池传感器可用于测量器件的内部电池电压。电池输入内部连接到 ADC 输入通道，该通道用于将传感器输出电压转换为数字值。

电池传感器范围为不超过 3.6 V。

校准后 ADC 转换数据的公式如下：

代码 = Integer（4096/3.6 * VIN）[钳制速度 4095]

作为校准点，VBAT 被视为单个负输入，范围为 3.6 V。

12.2.4 ADC 输入模式转换

ADC 被设计为提供数字值，对应于模拟电源与转换通道上施加的电压之比。对于大多数应用程序用例，必须将该比例转换为独立于 VDDA 的电压。

校准和偏移后 ADC 数字转换数据的公式如下：

- 单端输入模式

  代码 = Integer（Slope* VIN）[钳制速度 4095]

  其中单端输入模式的 Slope 具有以下值：

  3.6 V 模式： Slope = 4096/3.6 [校准增益 = 1/3]

  2.4 V 模式： Slope = 4096/2.4 [校准增益 = 1/2]

  1.2 V 模式： Slope = 4096/1.25 [校准增益 = 0.96，增益箝制速度 1]

- 差分输入模式

  代码 = Integer(Slope *（VINP - VINN））+ 2048 [箝制速度 4095]

  其中差分输入模式的斜率具有以下值：

  3.6 V 模式： Slope = 2048/3.6 [校准增益 = 1/3]

  2.4 V 模式： Slope = 2048/2.4 [校准增益 = 1/2]

  1.2 V 模式： Slope = 4096/1.25 [校准增益 = 0.96，增益箝制速度 1]

12.2.5 校准点

ST 于生产过程中在系统内存中存储了每个器件的校准值。每个值包含如下 12 位无符号增益值和 8 位有符号偏移值：

OFFSET[18:12] | GAIN[11:0]

这些值可写入寄存器 COMP_x（x=1，2，3，4），以便将一点应用于特定的 ADC 输入。COMP_SEL 寄存器允许将特定的校准点关联到一个 ADC 输入。

仅当 7 位寄存器字段能够容纳偏移值时，偏移值才能写入 COMP_x 寄存器，这意味着偏移位于[-64，63]以内。否则，它可能会作为 raw_value + 偏移被输出原始数据手动删除。

负偏移值需进行如下转换：

偏移 | 0x80，如果位字段 BIT_INVERT_SN=1（默认值）。

下面是校准点及其在系统内存中的位置的列表。

表 30. 校准点

校准点	地址位置
VINPx - VINMx 范围 1.2 V	0x10001E00
VINMx 范围 1.2 V	0x10001E04
VINPx 范围 1.2 V	0x10001E08
VINPx - VINMx 范围 2.4 V	0x10001E0C
VINMx 范围 2.4 V	0x10001E10
VINPx 范围 2.4 V	0x10001E14
VINPx - VINMx 范围 3.6 V	0x10001E18
VINMx 范围 3.6 V	0x10001E1C
VINPx 范围 3.6 V	0x10001E20

注意： *校准点的前一个版本具有 7 位有符号偏移。该版本可被视为用户可以读取地址 0x10001EFC 的值 0。*

12.2.6 稳定状态输入阻抗

由于 ADC 的输入性质是切换电容，其稳定状态输入阻抗被定义 DC 中所见阻抗。它只取决于模拟采样频率（Fs）和输入电容（Cin）：Zin = 1/(Cin*Fs）。

12.2.7 输入信号采样瞬态响应

如“图 25. 模拟源电阻的作用”所示，模拟信号路径包含源和引脚之间的串联电阻（Rext）、内部开关电阻（Rin）和内部采样电容（Cin）。电容的充电由 Rin 控制。如果串联了 Rext，Cin 充电的有效值取决于 Rin+Rext。因此，充电时间常量为（Rin+Rext）*Cin，达到给定精度所必需的时间更长。ADC 具有固定的采样时间 Tsw，它取决于 ADC 频率 1/Ts，如“图 24. ADC 采样时间 Tsw 和采样周期 Ts”所示。

图 24. ADC 采样时间 Tsw 和采样周期 Ts

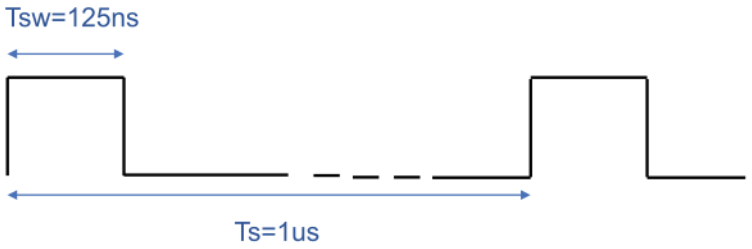
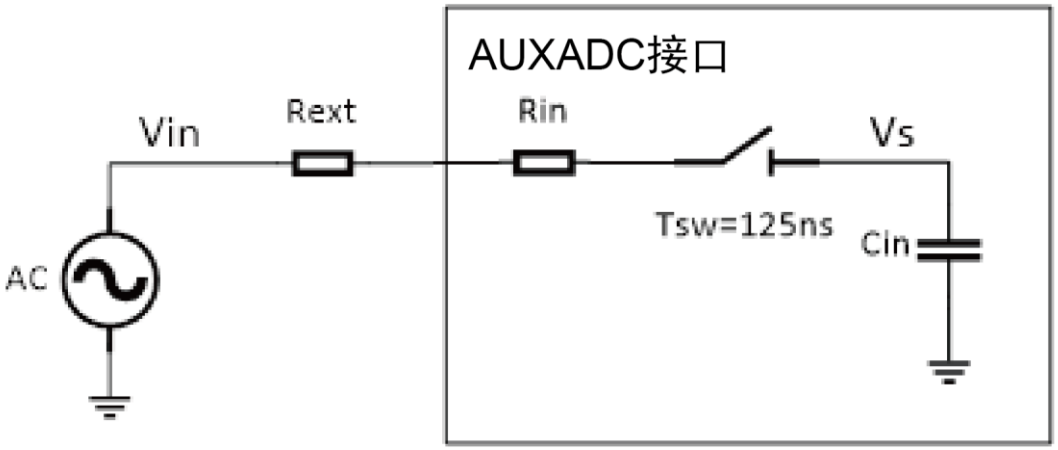


图 25. 模拟源电阻的作用



已知：Rin=550 Ω，Cin=4 pF，并强加 1/2 位的最大采样误差，我们可以确定以下最大输入电阻：

$\varepsilon = (Vs-Vin)/Vin = -e^{(-t/RC)}$

$\ln|\varepsilon| \leq t/(Rext+Rin)*Cin$

$(Rext+Rin)*Cin \leq t/\ln|\varepsilon|$

$(Rext+Rin)*Cin \leq 125e\text{-}9/\ln(122e\text{-}6)$

$(Rext+Rin)*Cin \leq 14\ ns$

$Rext \leq 14\ ns/4\ pF - 550\ \Omega$

$Rext \leq 2950\ \Omega$

其中：

$|\varepsilon| \leq 1/2^{13}$

$|\varepsilon| \leq 122e\text{-}6$

12.2.8 抽取滤波器（DF）

抽取滤波器（DF）的用途是在改善整体噪声性能（信噪比）的同时，提供一种减少传入的固定频率采样以获得某些固定的输出数据速率的方式。

抽取滤波器用于处理来自数字 MEMS 麦克风或来自 12 位 ADC 的输出的 PDM 数据流，主要面向信号的音频源。

抽取滤波器的输入数据速率为：

- 在模拟音频/完全模式下：1 Mpbs

- 在数字音频模式下：PDM 时钟频率本身（通过 DF_CONF 寄存器的 PDM_RATE 位字段进行配置）。限制是满足 MEMS 麦克风的可能频率范围（通常为 1 至 3 MHz）。

  抽取滤波器由两个链接块组成：

  – CIC 滤波器能够执行各种抽取因子

  – FIR 滤波器执行抽取的最后一个阶段并补偿 CIC 滤波器的衰减效应。

FIR 滤波器具有固定的抽取率 3

图 26. 简化的抽取滤波器框图

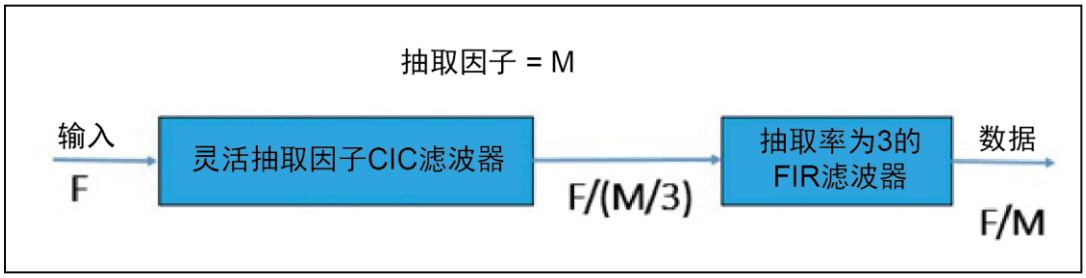


表 31. 模拟模式下 ADC 输入频率为 1 MHz 时的输出数据速率和表 32. 有数字麦克风输入的 CIC 滤波器输出频率根据支持的输入（ADC 或数字麦克风）列出了抽取因子和输出频率。

输入数据速率与输出数据速率之间的关系是输入数据速率：

- 除以 DF_CIC_DEC_FACTOR
- 然后除以 3
- 除以 2（若 DF_CIC_DHF 置位）

注意： *对于数字音频场景，此 2 分频的噪声过大；仅适用于模拟场景。*

- 乘以 1.2（若 DF_IT1P2 置位）

表 31. 模拟模式下 ADC 输入频率为 1 MHz 时的输出数据速率

目标频率	DF_CIC_DEC_FACTOR	DF_CIC_DHF (freq/2)	DF_ITP1P2 (freq x 1.2)	输出频率	误差
8 kHz	42	0	0	7.936 kHz	0.79 %
16 kHz	21	0	0	15.873 kHz	0.79 %
200 kHz(1)	2	0	1	200 kHz	0 %
44.1 kHz(2)	14	1	0	44.444 kHz	0.78 %
48 kHz(2)	7	0	0	47.619 kHz	0.79 %
22.05 kHz(2)	15	0	0	22.222 kHz	0.78 %

1. *不应对模拟音频场景使用 200 kHz 用例。它主要面向传感器测量。*
2. *这是一种可能的配置，但未经确认亦无保证（要在实验室中验证）。*

注意： *必须遵守对 APB 系统时钟（FPCLK）与输出数据速率（DR_{out}）之比的限制：*

- *在模拟音频模式和完全模式下，要遵循的比率为 FPCLK/DRout≥6。*

示例：为使 DRout = 200 kHz，F_{PCLK} 必须至少为 2 MHz。

如果不使用 DMA 通过抽取滤波器路径获取数据输出，为避免丢失采样，CPU 需以足够高的频率比计时（考虑到总线矩阵延时）。

表 32. 有数字麦克风输入的 CIC 滤波器输出频率

32 MHz 时钟的分频比	数字麦克风频率	MCIC	目标频率	输出频率	误差
16	2 MHz	84	8 kHz	7.936 kHz	0.79 %
16	2 MHz	42	16 kHz	15.873 kHz	0.79 %
16	2 MHz	30	22.05 kHz	22.22 kHz	0.78 %
16	2 MHz	15	44.1 kHz	44.44 kHz	0.78 %
16	2 MHz	14	48 kHz	47.619 kHz	0.79 %

表 33. 不同 pdm_rate 的 CIC/完全链的最小抽取因子提供了 CIC 滤波器和完全链的允许最小抽取因子值和 pdm_rate 输入参数。

表 33. 不同 pdm_rate 的 CIC/完全链的最小抽取因子

pdm_rate	抽取滤波器输入速率	CIC/完全链的最小允许抽取因子
0	3.2 MHz	5/15
1	2.91 MHz	5/15
2	2.66 MHz	5/15
3	2.46 MHz	4/12
4	2.28 MHz	4/12
5	2.13 MHz	4/12
6	2.00 MHz	4/12
7	1.88 MHz	3/9
8	1.78 MHz	3/9
9	1.68 MHz	3/9
10	1.60 MHz	3/9
11	1.52 MHz	3/9
12	1.46 MHz	3/9
13	1.40 MHz	3/9
14	1.33 MHz	3/9
15	1.28 MHz	2/6

当抽取滤波器的输入来自 12 位 ADC 时，可通过配置位（DF_CONF 寄存器中的 DF_I_U2S 位）进行无符号数据至有符号数据的转换。

抽取滤波器的输出也可通过配置位（DF_CONF 寄存器中的 DF_O_S2U 位）从有符号转换为无符号，以便兼容总是有符号的下采样器输出。

在 PDM 时钟的下降沿或上升沿进行输入 PDM 数据采样。这可以通过 DF_CONF 寄存器中的 DF_MICRO_L_RN 位进行配置。PDM 时钟是根据目标输出频率，通过抽取滤波器时钟（总是 32 MHz）的分频来生成。

12.2.9 下采样器（DS）

该下采样器是简单的平均滤波器，可将 ADC 频率进行 1 至 128 分频（2 的幂次）。

目的是处理多个 ADC 采样并进行平均，从而得到一个数据宽度更大（12 至 16 位）的数据。

下采样器提高了数据精度但降低了输出数据速率。

注意： *必须遵守对 APB 系统时钟（F_{PCLK}）与输出数据速率（DRout）之比的限制：*

- 在 ADC 模式下，要遵循的比率为 F_{PCLK}/DRout ≥ 4。

示例：为使 DRout = 500 kHz，F_{PCLK} 必须至少为 2 MHz。

如果不使用 DMA 通过下采样滤波器路径获取数据输出，为避免丢失采样，CPU 需以足够高的频率比计时（考虑到总线矩阵延时）。

12.3 中断

ADC 模块生成的可屏蔽中断有 6 个。这些中断组合后形成一个中断输出，是从 ADC 至 CPU 的唯一中断线路。

表 34. ADC 中断请求

中断事件	事件标志	中断/标志清零方法	中断使能控制位
下采样器转换结束	EODS_IRQ	在 EODS_IRQ 位写入 1	EODS_IRQ_ENA
抽取滤波器转换结束	EODF_IRQ	在 EODF_IRQ 位写入 1	EODF_IRQ_ENA
转换结束序列	EOS_IRQ	在 EOS_IRQ 位写入 1	EOS_IRQ_ENA
模拟看门狗事件	AWD_IRQ	在 AWD_IRQ 位写入 1	AWD_IRQ_ENA
下采样器上溢	OVR_DS_IRQ	在 OVR_DS_IRQ 位写入 1	OVR_DS_IRQ_ENA
抽取滤波器上溢	OVR_DF_IRQ	在 OVR_DF_IRQ 位写入 1	OVR_DF_IRQ_ENA

12.4 DMA 接口

ADC 有两个 DMA 通道接口：

- 一个用于获取下采样器数据输出值（DS_DATAOUT 寄存器），
- 一个用于获取抽取滤波器数据输出值（DF_DATAOUT 寄存器）

DMA 特性由软件通过 CONF 寄存器使能，下采样器数据输出通过 DMA_DS_ENA 位，抽取滤波器数据输出通过 DMA_DF_ENA 位。

如果为一个或两个输出禁用了 DMA 特性，CPU 可通过对应的 APB 寄存器读取数据。

12.5 ADC 模式

表 35. ADC 模式总结提供了支持的不同模式的信息总览。

表 35. ADC 模式总结

模式	输入信号	DF 或 DS	连续或单一	可能的并发模式
模拟音频	模拟麦克风	DF	连续	偶发
数字音频	数字 GPIO（PDM_DATA）	DF	连续	ADC
ADC	• 8 个单一外部通道（当耦合为差分通道时，为 4 个） • VBAT • 温度传感器	DS	连续或单一	数字音频
满	• 8 个单一外部通道（当耦合为差分通道时，为 4 个） • VBAT • 温度传感器	DF	连续	偶发
偶发	• VBAT • 温度传感器	DS	单	• 满 • 模拟音频

12.5.1 模拟音频模式

描述

模拟音频模式有以下特征：

- 模拟音频模式下的输入是模拟麦克风信号
- 输入信号由可编程增益放大器（PGA）从 0 dB 放大至 30 dB
- 数据路径始于模拟音频 PGA 接口、ADC 和抽取滤波器（在该模式下不使用下采样器）
- 在 DF_DATAOUT 寄存器中输出转换的数据
- 模拟音频模式转换序列总是采用连续模式。

模拟音频模式的使用

本段描述使用模拟音频模式的过程：

- 设定 PB2 和 PB3 I/O，以便将两个外部电容连接到 PGA 模拟模块。这是分两个步骤完成的（按顺序执行）：
  - 首先，设定 PB2 和 PB3 模拟开关，通过 SYSCFG 模块中 GPIO_SWA_CTRL[1] 位的置位来选择 PGA 外部电容模式，
  - 然后，通过 GPIOB 模块的 GPIOB_MODER[5:4] 和 GPIOB_MODER[7:6] 位字段将 PB2 和 PB3 I/O 设定为模拟模式。
  - 通过将 CTRL 寄存器中的 ADC_LDO_ENA 位置 1 来使能 ADC 的 LDO。

注意：当使用 QFN32 器件时，此 LDO 使能位不得置位，这是因为该封装上为 ADC LDO 供电的 VDDA 引脚不可用。

- 通过将 CTRL 寄存器中的 ADC_ON_OFF 位置 1 为 ADC 上电（若尚未执行）
- 通过 PGA_CONF 寄存器中的 PGA_GAIN 位字段设定所需 PGA 增益
- 通过 PGA_CONF 寄存器中的 PGA_BIAS 位字段设定模拟偏置电压
- 通过 CONF 寄存器中的 OP_MODE 位字段设定模拟音频模式
- 如有需要，修改抽取滤波器的数据速率
- 通过将 CONF 寄存器中的 CONT 位置 1 来设定连续模式（在该模式下为强制）
- 通过将 CTRL 寄存器中的 START_CONV 位置 1 来启动转换
- 每当抽取滤波器的输出有数据可用时，数据会被存储在 DF_DATAOUT 寄存器中且 EODF 标志置位（以模拟音频模式通过抽取滤波器）

- 为了获得转换值：
  - 要么 DF 数据路径上使能了 DMA（通过 CONF 寄存器中的 DMA_DF_ENA 位）且 DMA 在每次数据转换结束时复制 RAM 中的数据（通过 APB 总线）
  - 要么软件使能了 EODF_IRQ 中断且能够在生成新的转换数据之前从 DF_DATAOUT 寄存器获取数据。

注意： *如果 CPU 不设法在生成新的转换数据之前获取转换数据，OVR_DF_IRQ 标志将变为高电平以发出数据已丢失的通知。软件可通过 CONF 寄存器中的 OVR_DF_CFG 位设定发生上溢时的硬件行为：*

- *如果是 0，保留之前的数据，新数据丢失。*
- *如果是 1，之前的数据丢失，保留新数据。*
- 数据转换继续进行，直至软件通过将 CTRL 寄存器中的 STOP_OP_MODE 位置位使其停止：在这种情况下，转换立即停止且发布正在进行的转换的数据。

12.5.2 ADC 模式

描述

ADC 模式有以下特征：

- ADC 模式下的输入可以是 8 个外部通道和 2 个内部源（VBAT 和温度传感器）
- 从 ADC 至下采样器的数据路径（在该模式下不使用抽取滤波器）
- 在 DS_DATAOUT 寄存器中输出转换的数据
- 输出数据速率处于范围 4.4 ksps 至 1 Msps 以内
- 借助于下采样器，12 位转换数据可扩展至至多 16 位数据。但是，在这种情况下，输出数据速率将下降
- 常规转换序列可以单发或连续模式执行
  - 常规序列包含任意 ADC 输入通道（模拟麦克风除外）上和任何顺序的连续 ADC 转换。
  - 常规序列可链接至多 16 次转换。
  - 序列的每次转换的输入源均通过 SEQ_1 和 SEQ_2 寄存器的 SEQx 位字段进行选择。
  - 通过将 CONF 寄存器中的 CONT 位置 1，可以运行一次或连续重复运行此常规序列。

ADC 模式的使用

本段描述使用 ADC 模式的过程：

- 通过将 CTRL 寄存器中的 ADC_LDO_ENA 位置 1 来使能 ADC 的 LDO

注意： *当使用 QFN32 器件时，此 LDO 使能位不得置位，这是因为该封装上为 ADC LDO 供电的 VDDA 引脚不可用。*

- 通过将 CTRL 寄存器中的 ADC_ON_OFF 位置 1 为 ADC 上电（若尚未执行）
- 通过 SAMPLE_RATE 和 DS_CONF 寄存器设定目标数据速率
- 通过 SWITCH 寄存器设定输入电压选择
- 设定 COMP_1 至 COMP_4 和 COMP_SEL 寄存器
- 通过 CONF 寄存器中的 OP_MODE 位字段设定 ADC 模式
- 通过 SEQ_1 和 SEQ_2 寄存器设定目标常规序列（至多 16 次连环转换）
- 在 CONF 中的 SEQ_LEN 位字段指定序列长度（0（一次转换）至 0xF（16 次转换））。

如需执行一次以上的转换，确保 CONF 寄存器中的 SEQUENCE 位置 1。

- 根据所需序列，设定 CONF 寄存器中的 CONT 位和 SEQ_LEN 位字段（考虑 SEQUENCE 位总是置位）：
  - CONT = 0 且 SEQ_LEN = 0，在单一通道上执行一次转换。
  - CONT = 0 且 SEQ_LEN > 0，在不同通道/源上运行一次包含连续多次转换的序列。
  - CONT = 1 且 SEQ_LEN = 0，在单一通道/源上执行连续转换。
  - CONT = 1 且 SEQ_LEN > 0，在不同通道/源上连续运行包含连续多次转换的序列。
- 通过将 CTRL 寄存器中的 START_CONV 位置位来启动设定的常规序列
- 每当下采样器的输出有数据可用时，数据会被存储在 DS_DATAOUT 寄存器中且 EODS 标志置位（以模拟模式通过下采样器）
- 为了获得转换值：
  - 要么 DS 数据路径上使能了 DMA（通过 CONF 寄存器中的 DMA_DS_ENA 位）且 DMA 在每次数据转换结束时复制 RAM 中的转换数据
  - 要么软件使能了 EODS_IRQ 中断且能够在生成新的转换数据之前从 DS_DATAOUT 寄存器获取数据。

注意： *如果 CPU 不设法在生成新的转换数据之前获取转换数据，OVR_DS_IRQ 标志将变为高电平以发出数据已丢失的通知。软件可通过 CONF 寄存器中的 OVR_DS_CFG 位设定发生上溢时的硬件行为：*

- *如果是 0，保留之前的数据，新数据丢失。*
- *如果是 1，之前的数据丢失，保留新数据。*
- 每当常规序列完成时，EOS_IRQ 标志变为高电平（如使能，还可能生成中断）
- 如果序列是单一序列（SEQ_LEN=0），ADC 将在序列结束时停止且不重新开始，直至 START_CONV 位不再置位
- 数据转换继续进行，直至软件通过将 CTRL 寄存器中的 STOP_OP_MODE 位置位使其停止：在这种情况下，转换立即停止且不发布正在进行的转换的数据。

12.5.3 数字音频模式

描述

数字音频模式旨在与外部数字 MEMS 麦克风互连。数字音频模式有以下特征：

- 只使用抽取滤波器和数字控制资源
- 抽取滤波器通过 2 个 GPIO 与外部麦克风相连：
  - PDM_CLK：时钟输出，频率可通过 DF_CONF 寄存器的 PDM_RATE[3:0] 进行设定（参见第 12.6.2 节“ADC 配置寄存器（CONF）”），
  - PDM_DATA：脉冲密度调制输入数据。
- 在 DF_DATAOUT 寄存器中输出转换的数据
- 数字音频模式转换序列总是采用连续模式。

数字音频模式的使用

本段描述使用数字音频模式的过程：

- 设定目标 IO 上的 PDM_CLK 和 PDM_DATA 复用功能，以便将数字麦克风连接到抽取滤波器
- 设定 DF_CONF 寄存器中的 DF_MICRO_L_RN 位，以便选择麦克风的右或左通道
- 通过 DF_CONF 寄存器的 DF_ITP1P2、DF_CIC_DHF 和 DF_CIC_DEC_FACTOR 位字段设定抽取率
- 确保 CONF 寄存器中的 OP_MODE 位字段等于 2
- 在软件将 CTRL 寄存器中的 DIG_AUD_MODE 位置位后，抽取滤波器开始发送 PDM_CLK 并连续处理从 PDM_DATA 输入接收到的数据
- 每当有滤波数据可用时，数据会被存储在 DF_DATAOUT 寄存器中且 EODF_IRQ 标志置位（以数字音频模式通过抽取滤波器）。

注意： *为保证发布的第一个值的正确性，在得到第一个滤波数据之前有一个延时，该延时取决于滤波器延时（依据抽取因子）。*

- 为了获得转换值：
  - 要么 DF 数据路径上使能了 DMA（通过 CONF 寄存器中的 DMA_DF_ENA 位）且 DMA 在每次数据转换结束时复制 RAM 中的转换数据
  - 要么软件使能了 EODF_IRQ 中断且能够在生成新的转换数据之前从 DF_DATAOUT 寄存器获取数据。

注意： *如果 CPU 不设法在生成新的转换数据之前获取转换数据，OVR_DF_IRQ 标志将变为高电平以发出数据已丢失的通知。*

注意： *软件可通过 CONF 寄存器中的 OVR_DF_CFG 位设定发生上溢时的硬件行为：*

- *如果是 0，保留之前的数据，新数据丢失。*
- *如果是 1，之前的数据丢失，保留新数据。*

- 在软件将 CTRL 寄存器中的 DIG_AUD_MODE 位清零后，转换立即停止且正在进行的转换不再继续。

12.5.4 完全模式

描述

完全模式与模拟音频模式相同，但具有除音频 PGA 接口外的其他模拟通道。

完全模式有以下特征：

- 完全模式下的输入可以是 8 个外部通道和 2 个内部源（VBAT 和温度传感器）
- 数据路径始于 ADC 并经过抽取滤波器（在该模式下不使用下采样器）
- 在 DF_DATAOUT 寄存器中输出转换的数据
- 完全模式转换序列总是采用连续模式
- CONF.SEQ_LEN[3:0]=0x0 是该模式的唯一受支持的配置。

完全模式的使用

本段描述使用完全模式的过程：

- 通过将 CTRL 寄存器中的 ADC_LDO_ENA 位置 1 来使能 ADC 的 LDO

注意： *当使用 QFN32 器件时，此 LDO 使能位不得置位，这是因为该封装上为 ADC LDO 供电的 VDDA 引脚不可用。*

- 通过将 CTRL 寄存器中的 ADC_ON_OFF 位置 1 为 ADC 上电（若尚未执行）
- 通过 CONF 寄存器中的 OP_MODE 位字段设定完全模式
- 设定目标数据速率
- 通过将 CONF 寄存器中的 CONT 位置 1 来设定连续模式（在该模式下为强制）
- 确保 CONF.SEQ_LEN[3:0]=0x0
- 通过将 CTRL 寄存器中的 START_CONV 位置 1 来启动转换
- 每当转换完成时，转换数据会被存储在 DF_DATAOUT 寄存器中且 EODF 标志置位（以模拟音频模式通过抽取滤波器）
- 为了获得转换值：
  - 要么 DF 数据路径上使能了 DMA（通过 CONF 寄存器中的 DMA_DF_ENA 位）且 DMA 在每次数据转换结束时复制 RAM 中的转换数据
  - 要么软件使能了 EODF_IRQ 中断且能够在生成新的转换数据之前从 DF_DATAOUT 寄存器获取数据。

注意： *如果 CPU 不设法在生成新的转换数据之前获取转换数据，OVR_DF_IRQ 标志将变为高电平以发出数据已丢失的通知。*

注意： *软件可通过 CONF 寄存器中的 OVR_DF_CFG 位设定发生上溢时的硬件行为：*

- *如果是 0，保留之前的数据，新数据丢失。*
- *如果是 1，之前的数据丢失，保留新数据。*
- 数据转换继续进行，直至软件通过将 CTRL 寄存器中的 STOP_OP_MODE 位置位使其停止：在这种情况下，转换立即停止且不发布正在进行的转换的数据。

12.5.5 偶发模式

描述

偶发模式有以下特征：

- 偶发模式输入为 VBAT 或温度传感器
- 数据路径始于 ADC 并经过下采样器
- 在 DS_DATAOUT 寄存器中输出转换的数据
- 偶发模式总是单次转换

- 偶发模式只是 AUDIO 模式或完全模式的补充模式，不能单独激活
- 为该模式应用的下采样器比率始终是 1。不考虑在 DS_RATIO 寄存器的 DS_RATIO 位字段中设定的 DS 比率。

偶发模式的使用

本段描述使用偶发模式的过程：

- 通过 OCM_CTRL 寄存器中的 OCM_SRC 位字段选择要在突发模式下转换的源（VBAT 或温度传感器）
- 将 OCM_CTRL 寄存器中的 OCM_ENA 位置位以启动偶发转换
- 单次转换完成后，转换数据会被存储在 DS_DATAOUT 寄存器中且 EODS 标志置位（以偶发模式通过下采样器）
- 为了获得转换值：
  - 要么 DS 数据路径上使能了 DMA（通过 CONF 寄存器中的 DMA_DS_ENA 位）且 DMA 在每次数据转换结束时复制 RAM 中的转换数据
  - 要么软件使能了 EODS_IRQ 中断且能够在生成新的转换数据之前从 DS_DATAOUT 寄存器获取数据。

12.5.6 并发功能

一些模式支持使用并发转换：

- ADC 模式和数字音频模式
- 模拟音频模式和偶发模式
- 完全模式和偶发模式。

这种情况下，在 DF_DATAOUT 和/或 DS_DATAOUT 寄存器中提供数据，具体取决于选择的并发模式。

注意： *音频模拟模式、完全模式和偶发模式使用共用的硬件资源（12 位 ADC 模块）。此外，音频模拟模式和完全模式是连续模式。因此，当在并发模式下使用偶发模式时，会丢失一次转换。在该配置下，硬件保留丢失转换的先前数据。*

12.6 ADC 寄存器说明

12.6.1 版本寄存器（VERSION_ID）

偏移地址：0x00

复位值：0x0000 0020

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.	VERSION_ID[7:0]							
								r							

位 31:8	保留，必须保持复位值。
位 7:0	**VERSION_ID[7:0]**：嵌入式 IP 的版本。

12.6.2 ADC 配置寄存器（CONF）

偏移地址：0x04

复位值：0x0002 0002

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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VBIAS_PRECH_FORCE	ADC_CONT_1V2	BIT_INVERT_DIFF	BIT_INVERT_SN	OVR_DF_CFG	OVR_DS_CFG	DMA_DF_ENA	DMA_DS_ENA	SAMPLE_RATE[1:0]		Res.	Res.	OP_MODE[1:0]		SMPS_SYNCHRO_ENA	SEQ_LEN[3:0]			SEQUENCE	CONT
											rw	rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw					

位 31:21	保留，必须保持复位值。
位 20	**VBIAS_PRECH_FORCE**：保持 VBIAS_PRECH 使能以停用滤波器（当电源足够干净时）的可能性： • 0：VBIAS_PRECH 信号受 ADC 数字状态机控制（默认） • 1：VBIAS_PRECH 信号永久置 1
位 19	**ADC_CONT_1V2**：选择输入采样方法： • 0：只在转换开始时采样（默认） • 1：在转换结束时开始采样
位 18	**BIT_INVERT_DIFF**：当 ADC 连接了差分输入时，将 ADC 数据输出按位取反（1 的补码）： • 0：不取反（默认） • 1：使能取反
位 17	**BIT_INVERT_SN**：当 ADC 连接了单次负输入时，将 ADC 数据输出按位取反（1 的补码）： • 0：不取反 • 1：使能取反（默认）

位 16	**OVR_DF_CFG**：抽取滤波器上溢配置： • 0：保留之前的数据，新数据丢失（默认） • 1：之前的数据丢失，保留新数据
位 15	**OVR_DS_CFG**：下采样器上溢配置： • 0：保留之前的数据，新数据丢失（默认） • 1：之前的数据丢失，保留新数据
位 14	**DMA_DF_EN**：为抽取滤波器数据路径使能 DMA 模式： • 0：DMA 模式禁用 • 1：DMA 模式使能
位 13	**DMA_DS_EN**：为下采样器数据路径使能 DMA 模式： • 0：DMA 模式禁用 • 1：DMA 模式使能
位 12:11	**SAMPLE_RATE[1:0]**：ADC 的转换速率： – 00：16（= 1 Msp/s） – 01：20（= 800 ksp/s） – 10：24（= 667 ksp/s） – 11：28（= 571 ksp/s）
位 10:9	保留，必须保持复位值。
位 8:7	**OP_MODE[1:0]**：ADC 模式选择（= 数据路径选择）： • 00：保留供将来使用 • 01：模拟音频模式（PGAON） • 10：ADC 模式 • 11：完全模式
位 6	**SMPS_SYNCHRO_ENA**：同步 ADC 开始转换和 SMPS 生成的脉冲： • 0：为所有 ADC 时钟频率禁用 SMPS 同步 • 1：SMPS 同步使能 注意：当 PWRC_CR5.NOSMPS=1 时，SMPS_SYNCHRO_ENA 必须为 0。
位 5:2	**SEQ_LEN[3:0]**：常规序列中的转换次数： • 0000：1 次转换，从 SEQ 0 开始 • 0001：2 次转换，从 SEQ 0 开始 – ... • 1111：16 次转换，从 SEQ 0 开始
位 1	**SEQUENCE**：使能序列模式（默认激活）： • 0：序列模式禁用，只选择 SEQ0 • 1：序列模式使能，转换序列为 SEQ0 至 SEQx，其中的 x=SEQ_LEN（默认） 注意：将此位清零相当于 SEQUENCE=1 且 SEQ_LEN=0000。理想情况下，此位可保持高电平，作为冗余，保持高电平并设置 SEQ_LEN=0000。
位 0	**CONT**：当 ADC 模式使能时，常规序列连续运行： • 0：使能单次转换：当序列结束时，转换停止 • 1：使能连续转换：当序列结束时，序列再次启动，直至软件将 CTRL.STOP_OP_MODE 位置位

12.6.3 ADC 控制寄存器（CTRL）

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	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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADC_LDO_ENA	Res.	DIG_AUD_MODE	STOP_OP_MODE	START_CONV	ADC_ON_OFF
																										rw	rw	rw	t	t	rw

位 31:6	保留，必须保持复位值。
位 5	**ADC_LDO_ENA**：使能与 ADC 模块相关的 LDO： • 0：禁用 ADCLDO • 1：使能 ADCLDO 警告：在 QFN32 封装中，此位不得置位。
位 4	保留，必须保持复位值。
位 3	**DIG_AUD_MODE**：使能数字音频模式（数据路径使用抽取滤波器）： • 0：停止数字音频模式 • 1：启动数字音频模式
位 2	**STOP_OP_MODE**(1)：停止正在运行的 OP_MODE（ADC 模式、模拟音频模式和完全模式）： • 0：无操作 • 1：停止正在运行的 ADC 模式 注意：此位通过软件置 1，通过硬件清零。
位 1	**START_CONV**(1)：生成启动脉冲开始 ADC 转换： • 0：无操作 • 1：开始 ADC 转换 注意：此位通过软件置 1，通过硬件清零。
位 0	**ADC_ON_OFF**： • 0：关闭 ADC • 1：开启 ADC

1. *在将 STOP_MODE_OP 置位时，在通过将 START_CONV 位置位开始新的 ADC 转换之前，用户必须等待约 10 us。*

12.6.4 ADC 偶发模式控制寄存器（OCM_CTRL）

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	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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OCM_ENA	OCM_SRC
																														t	rw

位 31:2	保留，必须保持复位值。
位 1	**OCM_ENA**：在模拟音频模式和完全模式下开始偶发转换： • 0：无操作 • 1：开始偶发转换 注意：此位通过软件置 1，通过硬件清零。
位 0	**OCM_SRC**：选择偶发转换源 • 0：VBAT 偶发转换 • 1：温度传感器偶发转换

12.6.5 ADC PGA 配置寄存器（PGA_CONF）

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	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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PGA_BIAS[2:0]			PGA_GAIN[3:0]			
																									rw			rw			

位 31:7	保留，必须保持复位值。
位 6:4	**PGA_BIAS[2:0]**：设置麦克风偏置电压： – 000：0.5 x VBAT – 001：0.55 x VBAT – 010：0.6 x VBAT – 011：0.65 x VBAT – 100：0.7 x VBAT – 101：0.75 x VBAT – 110：0.8 x VBAT – 111：0.9 x VBAT
位 3:0	**PGA_GAIN[3:0]**：从 0 至 30 dB。 – 0000：0 dB PGA 最小增益，相当于 1.2 V ADC 满量程 – 0001：3 dB – 0010：6 dB – 0011：9 dB – 0100：12 dB – 0101：15 dB – 0110：18 dB – 0111：21 dB – 1000：24 dB – 1001：27 dB – 1010 至 1111：30 dB

12.6.6 ADC 输入电压开关选择寄存器（SWITCH）

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	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.	SE_VIN_7[1:0]		SE_VIN_6[1:0]		SE_VIN_5[1:0]		SE_VIN_4[1:0]		SE_VIN_3[1:0]		SE_VIN_2[1:0]		SE_VIN_1[1:0]		SE_VIN_0[1:0]	
																rw		rw		rw		rw			rw			rw			

位 31:16	保留，必须保持复位值。
位 15:14	**SE_VIN_7[1:0]**：VINP[3] 的输入电压。 • 00：Vinput = 1.2 V • 01：保留（不用于此晶元） • 10：Vinput = 2.4 V • 11：Vinput = 3.6 V
位 13:12	**SE_VIN_6[1:0]**：VINP[2]的输入电压。 • 00：Vinput = 1.2 V • 01：保留（不用于此晶元） • 10：Vinput = 2.4 V • 11：Vinput = 3.6 V
位 11:10	**SE_VIN_5[1:0]**：VINP[1]的输入电压。 • 00：Vinput = 1.2 V • 01：保留（不用于此晶元） • 10：Vinput = 2.4 V • 11：Vinput = 3.6 V
位 9:8	**SE_VIN_4[1:0]**：VINP[0]的输入电压。 • 00：Vinput = 1.2 V • 01：保留（不用于此晶元） • 10：Vinput = 2.4 V • 11：Vinput = 3.6 V
位 7:6	**SE_VIN_3[1:0]**：VINM[3]/VINP[3]-VINM[3]的输入电压。 • 00：Vinput = 1.2 V • 01：保留（不用于此晶元） • 10：Vinput = 2.4 V • 11：Vinput = 3.6 V
位 5:4	**SE_VIN_2[1:0]**：VINM[2]/VINP[2]-VINM[2]的输入电压。 • 00：Vinput = 1.2 V • 01：保留（不用于此晶元） • 10：Vinput = 2.4 V • 11：Vinput = 3.6 V
位 3:2	**SE_VIN_1[1:0]**：VINM[1]/VINP[1]-VINM[1]的输入电压。 • 00：Vinput = 1.2 V • 01：保留（不用于此晶元） • 10：Vinput = 2.4 V • 11：Vinput = 3.6 V

位 1:0	**SE_VIN_0[1:0]**：VINM[0]/VINP[0]-VINM[0]的输入电压。 • 00：Vinput = 1.2 V • 01：保留（不用于此晶元） • 10：Vinput = 2.4 V • 11：Vinput = 3.6 V

12.6.7 抽取滤波器配置寄存器（DF_CONF）

偏移地址：0x18

复位值：0x0000 3015

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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DF_HALF_D_EN	DF_HPF_EN	DF_MICROL_RN	PDM_RATE[3:0]				DF_O_S2U	DF_I_U2S	DF_ITP1P2	DF_CIC_DHF	DF_CIC_DEC_FACTOR[6:0]						
														rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:18	保留，必须保持复位值。
位 17	**DF_HALF_D_EN**：半动态使能。 • 0：关：全动态（默认） • 1：开：在抽取滤波器的输入处将动态频率除以 2。
位 16	**DF_HPF_EN**：高通滤波器使能。 • 0：关（默认） • 1：开，截止频率为 40 Hz。
位 15	**DF_MICROL_RN**：数字麦克风上的左/右通道选择： • 0：右通道选择 • 1：左通道选择
位 14:11	**PDM_RATE[3:0]**：选择 PDM 时钟频率。 PDM 周期 =（PDM_RATE[3:0] + 10）x 32mhz_period。 • 0000：PDM 频率 = 3.2 MHz – ... • 0110：PDM 频率 = 2 MHz（默认） – ... • 1111：PDM 频率 = 1.28 MHz
位 10	**DF_O_S2U**：为数据输出选择有符号/无符号格式 • 0：有符号（默认） • 1：无符号
位 9	**DF_I_U2S**：为输入选择有符号/无符号格式 • 0：无符号（默认） • 1：有符号
位 8	**DF_ITP1P2**：1.2 分数插值器使能 • 0：1.2 插值器被旁路（默认）， • 1：1.2 插值器开启。 注意：只能为以下配置将此位置 1：以 200 kps 的数据速率、1 MHz 的频率从 ADC 数据生成。如果为其他配置将其置位，则可能发生不可预测的结果。
位 7	**DF_CIC_DHF**：CIC 抽取滤波器半因子 • 0：整数因子（默认）， • 1：半因子。 注意：只能为以下配置将此位置 1：以 44.1 kps 的数据速率、1 MHz 的频率从 ADC 数据生成。如果为其他配置将其置位，则可能发生不可预测的结果。

位 6:0	**DF_CIC_DEC_FACTOR[6:0]：** 数字麦克风的 MCIC（PDM 频率 = 2 MHz） • 0x0E：输出频率 47.619 kHz（MCIC = 14） • 0x0F：输出频率 44.44 kHz（MCIC = 15） • 0x1E：输出频率 22.22 kHz（MCIC = 30） • 0x2A：输出频率 15.873 kHz（MCIC = 42） • 0x54：输出频率 7.936 kHz（MCIC = 84） 模拟麦克风的 MCIC（ADC 频率 = 1 MHz）： • 0x02：输出频率 200 kHz（DF_ITP1P2 = 1） • 0x15：输出频率 15.873 kHz • 0x2A：输出频率 7.936 kHz

12.6.8 下采样器配置寄存器（DS_CONF）

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	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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DS_WIDTH[2:0]			DS_RATIO[2:0]		
																										rw			rw		

位 31:6	保留，必须保持复位值。
位 5:3	**DS_WIDTH[2:0]**：设定下采样器数据输出宽度（DSDTATA）。 • 000：12 位上的 DS_DATA 输出（默认） • 001：13 位上的 DS_DATA 输出 • 010：14 位上的 DS_DATA 输出 • 011：15 位上的 DS_DATA 输出 • 100：16 位上的 DS_DATA 输出 • 1xx：保留
位 2:0	**DS_RATIO[2:0]**：设定下采样器比率（N 因子）。 – 000：比率 = 1，无下采样（默认） – 001：比率 = 2 – 010：比率 = 4 – 011：比率 = 8 – 100：比率 = 16 – 101：比率 = 32 – 110：比率 = 64 – 111：比率 = 128

12.6.9 ADC 序列编程 1 寄存器（SEQ_1）

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	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEQ7[3:0]				SEQ6[3:0]				SEQ5[3:0]				SEQ4[3:0]				SEQ3[3:0]				SEQ2[3:0]				SEQ1[3:0]				SEQ0[3:0]			
rw				rw				rw				rw				rw				rw				rw				rw			

位 31:28	**SEQ7[3:0]**：序列的第 8 次转换的通道数字代码。 参见 SEQ0 了解代码详情。
位 27:24	**SEQ6[3:0]**：序列的第 7 次转换的通道数字代码。参见 SEQ0 了解代码详情。
位 23:20	**SEQ5[3:0]**：序列的第 6 次转换的通道数字代码。参见 SEQ0 了解代码详情。
位 19:16	**SEQ4[3:0]**：序列的第 5 次转换的通道数字代码。参见 SEQ0 了解代码详情。
位 15:12	**SEQ3[3:0]**：序列的第 4 次转换的通道数字代码。参见 SEQ0 了解代码详情。
位 11:8	**SEQ2[3:0]**：序列的第 3 次转换的通道数字代码。参见 SEQ0 了解代码详情。
位 7:4	**SEQ1[3:0]**：序列的第 2 次转换的通道数字代码。参见 SEQ0 了解代码详情。
位 3:0	**SEQ0[3:0]**：序列的第 1 次转换的通道数字代码 • 0000：VINM[0]至 ADC 单次负输入 • 0001：VINM[1]至 ADC 单次负输入 • 0010：VINM[2]至 ADC 单次负输入 • 0011：VINM[3]至 ADC 单次负输入 • 0100：VINP[0]至 ADC 单次正输入 • 0101：VINP[1]至 ADC 单次正输入 • 0110：VINP[2]至 ADC 单次正输入 • 0111：VINP[3]至 ADC 单次正输入 • 1000：VINP[0]-VINM[0]至 ADC 差分输入 • 1001：VINP[1]-VINM[1]至 ADC 差分输入 • 1010：VINP[2]-VINM[2]至 ADC 差分输入 • 1011：VINP[3]-VINM[3]至 ADC 差分输入 • 1100：VBAT - 电池电压检测器 • 1101：温度传感器 • 111x：保留

12.6.10 ADC 序列编程 2 寄存器（SEQ_2）

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	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEQ15[3:0]				SEQ14[3:0]				SEQ13[3:0]				SEQ12[3:0]				SEQ11[3:0]				SEQ10[3:0]				SEQ9[3:0]				SEQ8[3:0]			
rw				rw				rw				rw				rw				rw				rw				rw			

位 31:28	**SEQ15[3:0]**：序列的第 16 次转换的通道数字代码。参见 SEQ8 了解代码详情。
位 27:24	**SEQ14[3:0]**：序列的第 15 次转换的通道数字代码。参见 SEQ8 了解代码详情。
位 23:20	**SEQ13[3:0]**：序列的第 14 次转换的通道数字代码。参见 SEQ8 了解代码详情。
位 19:16	**SEQ12[3:0]**：序列的第 13 次转换的通道数字代码。参见 SEQ8 了解代码详情。
位 15:12	**SEQ11[3:0]**：序列的第 12 次转换的通道数字代码。参见 SEQ8 了解代码详情。
位 11:8	**SEQ10[3:0]**：序列的第 11 次转换的通道数字代码。参见 SEQ8 了解代码详情。
位 7:4	**SEQ9[3:0]**：序列的第 10 次转换的通道数字代码。参见 SEQ8 了解代码详情。
位 3:0	**SEQ8[3:0]**：序列的第 9 次转换的通道数字代码。 • 0000：VINM[0]至 ADC 单次负输入 • 0001：VINM[1]至 ADC 单次负输入 • 0010：VINM[2]至 ADC 单次负输入 • 0011：VINM[3]至 ADC 单次负输入 • 0100：VINP[0]至 ADC 单次正输入 • 0101：VINP[1]至 ADC 单次正输入 • 0110：VINP[2]至 ADC 单次正输入 • 0111：VINP[3]至 ADC 单次正输入 • 1000：VINP[0]-VINM[0]至 ADC 差分输入 • 1001：VINP[1]-VINM[1]至 ADC 差分输入 • 1010：VINP[2]-VINM[2]至 ADC 差分输入 • 1011：VINP[3]-VINM[3]至 ADC 差分输入 • 1100：VBAT - 电池电压检测器 • 1101：温度传感器 • 111x：保留

12.6.11 ADC 增益和偏移校正 1 寄存器（COMP_1）

偏移地址：0x28

复位值：0x0000 0555

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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OSFFSET1[6:0]							GAIN1[11:0]											
													rw							rw											

位 31:19	保留，必须保持复位值。
位 18:12	**OFFSET1[6:0]**：第一个校准点：有符号偏移补偿[6:0]。
位 11:0	**GAIN1[11:0]**：第一个校准点：增益 AUXADC_GAIN_1V2[11:0]。

12.6.12 ADC 增益和偏移校正 2 寄存器（COMP_2）

偏移地址：0x2C

复位值：0x0000 0555

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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OSFFSET2[6:0]							GAIN2[11:0]											
													rw							rw											

位 31:19	保留，必须保持复位值。
位 18:12	**OFFSET2[6:0]**：第二个校准点：有符号偏移补偿[6:0]。
位 11:0	**GAIN2[11:0]**：第二个校准点：增益 AUXADC_GAIN_1V2[11:0]。

12.6.13 ADC 增益和偏移校正 3 寄存器（COMP_3）

偏移地址：0x30

复位值：0x0000 0555

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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OSFFSET3[6:0]							GAIN3[11:0]											
													rw							rw											

位 31:19	保留，必须保持复位值。
位 18:12	**OFFSET3[6:0]**：第三个校准点：有符号偏移补偿[6:0]。
位 11:0	**GAIN3[11:0]**：第三个校准点：增益 AUXADC_GAIN_1V2[11:0]。

12.6.14 ADC 增益和偏移校正 4 寄存器（COMP_4）

偏移地址：0x34

复位值：0x0000 0555

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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OSFFSET4[6:0]							GAIN4[11:0]											
													rw							rw											

位 31:19	保留，必须保持复位值。
位 18:12	**OFFSET4[6:0]**：第四个校准点：有符号偏移补偿[6:0]。
位 11:0	**GAIN4[11:0]**：第四个校准点：增益 AUXADC_GAIN_1V2[11:0]。

12.6.15 ADC 增益和偏移选择寄存器（COMP_SEL）

偏移地址：0x38

复位值：0x0000 0000

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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OFFSET_GAIN8[1:0]		OFFSET_GAIN7[1:0]		OFFSET_GAIN6[1:0]		OFFSET_GAIN5[1:0]		OFFSET_GAIN4[1:0]		OFFSET_GAIN3[1:0]		OFFSET_GAIN2[1:0]		OFFSET_GAIN1[1:0]		OFFSET_GAIN0[1:0]	
														rw		rw		rw		rw		rw		rw		rw		rw		rw	

位 31:18	保留，必须保持复位值。
位 17:16	**OFFSET_GAIN8[1:0]**：ADC 差分模式下使用的增益/偏移，Vinput 范围 = 3.6 V： • 00：来自 COMP_1 的 OFFSET1 和 GAIN1 • 01：来自 COMP_2 的 OFFSET2 和 GAIN2 • 10：来自 COMP_3 的 OFFSET3 和 GAIN3 • 11：来自 COMP_4 的 OFFSET4 和 GAIN4
位 15:14	**OFFSET_GAIN7[1:0]**：ADC 单次正模式下使用的增益/偏移，Vinput 范围 = 3.6 V： • 00：来自 COMP_1 的 OFFSET1 和 GAIN1 • 01：来自 COMP_2 的 OFFSET2 和 GAIN2 • 10：来自 COMP_3 的 OFFSET3 和 GAIN3 • 11：来自 COMP_4 的 OFFSET4 和 GAIN4
位 13:12	**OFFSET_GAIN6[1:0]**：ADC 单次负模式下使用的增益/偏移，Vinput 范围 = 3.6 V： • 00：来自 COMP_1 的 OFFSET1 和 GAIN1 • 01：来自 COMP_2 的 OFFSET2 和 GAIN2 • 10：来自 COMP_3 的 OFFSET3 和 GAIN3 • 11：来自 COMP_4 的 OFFSET4 和 GAIN4
位 11:10	**OFFSET_GAIN5[1:0]**：ADC 差分模式下使用的增益/偏移，Vinput 范围 = 2.4 V： • 00：来自 COMP_1 的 OFFSET1 和 GAIN1 • 01：来自 COMP_2 的 OFFSET2 和 GAIN2 • 10：来自 COMP_3 的 OFFSET3 和 GAIN3 • 11：来自 COMP_4 的 OFFSET4 和 GAIN4
位 9:8	**OFFSET_GAIN4[1:0]**：ADC 单次正模式下使用的增益/偏移，Vinput 范围 = 2.4 V： • 00：来自 COMP_1 的 OFFSET1 和 GAIN1 • 01：来自 COMP_2 的 OFFSET2 和 GAIN2 • 10：来自 COMP_3 的 OFFSET3 和 GAIN3 • 11：来自 COMP_4 的 OFFSET4 和 GAIN4
位 7:6	**OFFSET_GAIN3[1:0]**：ADC 单次负模式下使用的增益/偏移，Vinput 范围 = 2.4 V： • 00：来自 COMP_1 的 OFFSET1 和 GAIN1 • 01：来自 COMP_2 的 OFFSET2 和 GAIN2 • 10：来自 COMP_3 的 OFFSET3 和 GAIN3 • 11：来自 COMP_4 的 OFFSET4 和 GAIN4
位 5:4	**OFFSET_GAIN2[1:0]**：ADC 差分模式下使用的增益/偏移，Vinput 范围 = 1.2 V： • 00：来自 COMP_1 的 OFFSET1 和 GAIN1 • 01：来自 COMP_2 的 OFFSET2 和 GAIN2 • 10：来自 COMP_3 的 OFFSET3 和 GAIN3 • 11：来自 COMP_4 的 OFFSET4 和 GAIN4
位 3:2	**OFFSET_GAIN1[1:0]**：ADC 单次正模式下使用的增益/偏移，Vinput 范围 = 1.2 V： • 00：来自 COMP_1 的 OFFSET1 和 GAIN1 • 01：来自 COMP_2 的 OFFSET2 和 GAIN2 • 10：来自 COMP_3 的 OFFSET3 和 GAIN3 • 11：来自 COMP_4 的 OFFSET4 和 GAIN4

位 1:0	**OFFSET_GAIN0[1:0]**：ADC 单次负模式下使用的增益/偏移，Vinput 范围 = 1.2 V： • 00：来自 COMP_1 的 OFFSET1 和 GAIN1 • 01：来自 COMP_2 的 OFFSET2 和 GAIN2 • 10：来自 COMP_3 的 OFFSET3 和 GAIN3 • 11：来自 COMP_4 的 OFFSET4 和 GAIN4

12.6.16 ADC 看门狗阈值寄存器（WD_TH）

偏移地址：0x3C

复位值：0x0FFF 0000

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
Res.	Res.	Res.	Res.	WD_HT[11:0]												Res.	Res.	Res.	Res.	WD_LT[11:0]											
				rw																rw											

位 31:28	保留，必须保持复位值。
位 27:16	**WD_HT[11:0]**：模拟看门狗高电平阈值。
位 15:12	保留，必须保持复位值。
位 11:0	**WD_LT[11:0]**：模拟看门狗低电平阈值。

12.6.17 ADC 看门狗配置寄存器（WD_CONF）

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	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.	AWD_CHX[15:0]															
																rw															

位 31:16	保留，必须保持复位值。
位 15:0	**AWD_CHX[15:0]**：模拟看门狗通道选择，用于定义需由看门狗保护的输入通道。 • Bit0：VINM[0]至 ADC 负输入 • Bit1：VINM[1]至 ADC 负输入 • Bit2：VINM[2]至 ADC 负输入 • Bit3：VINM[3]至 ADC 负输入 • Bit4：MICROM 至 ADC 负输入 • Bit5：VBAT 至 ADC 负输入 • Bit6：GND 至 ADC 负输入 • Bit7：VDDA 至 ADC 负输入 • Bit8：VINP[0]至 ADC 正输入 • Bit9：VINP[1]至 ADC 正输入 • Bit10：VINP[2]至 ADC 正输入 • Bit11：VINP[3]至 ADC 正输入 • Bit12：MICROP 至 ADC 正输入 • Bit13：TEMP 至 ADC 正输入 • Bit14：GND 至 ADC 正输入 • Bit15：VDDA 至 ADC 正输入

12.6.18 下采样器数据输出寄存器（DS_DATAOUT）

偏移地址：0x44

复位值：0x0000 0000

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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DS_DATA[15:0]															
																r															

位 31:16	保留，必须保持复位值。
位 15:0	**DS_DATA[15:0]**：在下采样器输出处包含转换数据。

12.6.19 抽取滤波器数据输出寄存器（DF_DATAOUT）

偏移地址：0x48

复位值：0x0000 0000

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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DF_DATA[15:0]															
																r															

位 31:16	保留，必须保持复位值。
位 15:0	**DF_DATA[15:0]**：在下采样器输出处包含转换数据。

12.6.20 ADC 中断状态寄存器（IRQ_STATUS）

偏移地址：0x4C

复位值：0x0000 0000

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
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.	AWD_IRQ	EOS_IRQ	EODF_IRQ	EODS_IRQ	Res.
																								rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	

位 31:8	保留，必须保持复位值。
位 7	**DF_OVRFL_IRQ**：置位表示抽取滤波器饱和。读取时，提供中断状态： • 0：抽取滤波器未饱和 • 1：抽取滤波器饱和。 写入此位会将中断状态清零： • 0：无操作 • 1：将中断清零

位 6	**OVR_DF_IRQ**：置位表示抽取滤波器上溢（数据丢失）。读取时，提供中断状态： • 0：未发生上溢 • 1：发生上溢 写入此位会将中断状态清零： • 0：无操作 • 1：将中断清零
位 5	**OVR_DS_IRQ**：置位表示下采样器上溢（至少一个数据丢失）。读取时，提供中断状态： • 0：未发生上溢 • 1：发生上溢 写入此位会将中断状态清零： • 0：无操作 • 1：将中断清零
位 4	**AWD_IRQ**：在发生模拟看门狗事件时置位。读取时，提供中断状态： • 0：未发生模拟看门狗事件 • 1：发生了模拟看门狗事件。 写入此位会将中断状态清零： • 0：无操作 • 1：将中断清零
位 3	**EOS_IRQ**：在转换序列完成时置位。读取时，提供中断状态： • 0：转换序列未完成 • 1：转换序列完成。 写入此位会将中断状态清零： • 0：无操作 • 1：将中断清零
位 2	**EODF_IRQ**：在抽取滤波器转换完成时置位。读取时，提供中断状态： • 0：抽取滤波器转换未完成 • 1：抽取滤波器转换完成。 写入此位会将中断状态清零： • 0：无操作 • 1：将中断清零
位 1	**EODS_IRQ**：在下采样器转换完成时置位。读取时，提供中断状态： • 0：下采样器转换未完成 • 1：下采样器转换完成。 写入此位会将中断状态清零： • 0：无操作 • 1：将中断清零
位 0	保留，必须保持复位值。 • 0：ADC 转换未完成 • 1：ADC 转换完成 写入此位会将中断状态清零： • 0：无操作 • 1：将中断清零

12.6.21 ADC 中断使能寄存器（IRQ_ENABLE）

偏移地址：0x50

复位值：0x0000 0000

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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DF_OVRFL_IRQ_ENA	OVR_DF_IRQ_ENA	OVR_DS_IRQ_ENA	AWD_IRQ_ENA	EOS_IRQ_ENA	EODF_IRQ_ENA	EODS_IRQ_ENA	EOC_IRQ_ENA
																								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8	保留，必须保持复位值。
位 7	**DF_OVRFL_IRQ_ENA**：抽取滤波器饱和中断使能： • 0：DF_OVRFL 中断禁用 • 1：DF_OVRFL 中断使能
位 6	**OVR_DF_IRQ_ENA**：抽取滤波器上溢中断使能： • 0：抽取滤波器中断禁用 • 1：抽取滤波器中断使能
位 5	**OVR_DS_IRQ_ENA**：下采样器上溢中断使能： • 0：下采样器中断禁用 • 1：下采样器中断使能
位 4	**AWD_IRQ_ENA**：模拟看门狗中断使能： • 0：模拟看门狗中断禁用 • 1：模拟看门狗中断使能
位 3	**EOS_IRQ_ENA**：常规序列结束中断使能： • 0：EOS 中断禁用 • 1：EOS 中断使能
位 2	**EODF_IRQ_ENA**：抽取滤波器输出的转换结束中断使能： • 0：EODF 中断禁用 • 1：EODF 中断使能
位 1	**EODS_IRQ_ENA**：下采样器输出的转换结束中断使能： • 0：EODF 中断禁用 • 1：EODF 中断使能
位 0	保留，必须保持复位值。

12.6.22 ADC 定时器配置寄存器（TIMER_CONF）

偏移地址：0x54

复位值：0x0000 9628

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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRECH_DELAY_SEL	VBIAS_PRECH_DELAY[7:0]								ADC_LDO_DELAY[7:0]							
															rw	rw								rw							

位 31:17	保留，必须保持复位值。
位 16	**PRECH_DELAY_SEL**：为 VBIAS_PRECH_DELAY 定时器选择时间步长 PD_STEP。 – 0：PD_STEP = 4 us =（32 x 4）/32 MHz – 1：PD_STEP = 4.096 ms =（32 x 4 x 1024）/32 MHz
位 15:8	**VBIAS_PRECH_DELAY[7:0]**：定义始于 PGA_EN 信号上升沿的等待时间并对应于 VBIAS 预充电脉冲持续时间的时间长度。延时表示为 PD_STEP 的倍数，已知 PD_STEP 由 PRECH_DELAY_SEL 位的值定义。 时间单位为 PD_STEP（4 us 或 4.096 ms）。 当 PRECH_DELAY_SEL=0 时，最长延时为 1.02 ms（255 x 4 us）。 当 PRECH_DELAY_SEL=1 时，最长延时为 1044.48 ms（255 x 4.096 ms）。默认值为 600 us（150 x 4 us）。
位 7:0	**ADC_LDO_DELAY[7:0]**：定义要在 ADC_LDO 使能与 ADC 开启之间插入的等待时间的时间长度，以使 LDO 在开始转换前有时间稳定下来。 时间单位为 4 us。 最长延时为 1.02 ms（255 x 4 us）。默认值为 40 = 160 us。

12.7 ADC 寄存器映射

表 36. ADC 寄存器映射和复位值

偏移	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
0x00	VERSION_ID	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VERSION_ID[7:0]							
	复位值																									0	0	1	0	0	0	0	0
0x04	CONF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VBIAS_PRECH_FORCE	ADC_CONT_1V2	BIT_INVERT_DIFF	BIT_INVERT_SN	OVR_DF_CFG	OVR_DS_CFG	DMA_DF_ENA	DMA_DS_ENA	SAMPLE_RATE[1:0]			Res.	Res.	OP_MODE[1:0]		SMPS_SYNCHRO_ENA	SEQ_LEN[3:0]			SEQUENCE	CONT.
	复位值												0	0	0	1	0	0	0	0	0	0			0	0	0	0	0	0	0	1	0
0x08	CTRL	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.	ADC_LDO_ENA	Res.	DIG_AUD_MODE	STOP_OP_MODE	START_CONV	ADC_ON_OFF
	复位值																											0		0	0	0	0
0x0C	OCM_CTRL	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.	OCM_EN	OCMA_SRC
	复位值																															0	0
0x10	PGA_CONF	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.	PGA_BIAS[2:0]			PGA_GAIN[3:0]			
	复位值																										0	0	0	0	0	0	0
0x14	SWITCH	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SE_VIN_7[1:0]		SE_VIN_6[1:0]		SE_VIN_5[1:0]		SE_VIN_4[1:0]		SE_VIN_3[1:0]		SE_VIN_2[1:0]		SE_VIN_1[1:0]		SE_VIN_0[1:0]	
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

偏移	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
0x18	DF_CONF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DF_HALF_D_EN	DF_HPF_EN	DF_MICROL_RN	PDM_RATE[3:0]				DF_O_S2U	DF_I_U2S	DF_ITP1P2	DF_CIC_DHF	DF_CIC_DEC_FACTOR[6:0]						
	复位值															0	0	0	0	1	1	0		0	0	0	0	0	1	0	1	0	1
0x1C	DS_CONF	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.	DS_WIDTH[2:0]			DS_RATIO[2:0]		
	复位值																											0	0	0	0	0	0
0x20	SEQ_1	SEQ7[3:0]				SEQ6[3:0]				SEQ5[3:0]				SEQ4[3:0]				SEQ3[3:0]				SEQ2[3:0]				SEQ1[3:0]				SEQ0[3:0]			
	复位值	0	0	0	0	0	0	0	0	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	SEQ_2	SEQ15[3:0]				SEQ14[3:0]				SEQ13[3:0]				SEQ12[3:0]				SEQ11[3:0]				SEQ10[3:0]				SEQ9[3:0]				SE80[3:0]			
	复位值	0	0	0	0	0	0	0	0	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	COMP_1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OFFSET1[6:0]							GAIN1[11:0]											
	复位值														0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1
0x2C	COMP_2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OFFSET2[6:0]							GAIN2[11:0]											
	复位值														0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1
0x30	COMP_3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OFFSET3[6:0]							GAIN3[11:0]											
	复位值														0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1

偏移	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
0x34	COMP_4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OFFSET4[6:0]							GAIN4[11:0]											
	复位值														0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1
0x38	COMP_SEL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OFFSET_GAIN8[1:0]		OFFSET_GAIN7[1:0]		OFFSET_GAIN6[1:0]		OFFSET_GAIN5[1:0]		OFFSET_GAIN4[1:0]		OFFSET_GAIN3[1:0]		OFFSET_GAIN2[1:0]		OFFSET_GAIN01[1:0]		OFFSET_GAIN0[1:0]	
	复位值															0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	WD_TH	Res.	Res.	Res.	Res.	WD_HT[11:0]												Res.	Res.	Res.	Res.	WD_LT[11:0]											
	复位值					1	1	1	1	1	1	1	1	1	1	1	1					0	0	0	0	0	0	0	0	0	0	0	0
0x40	WD_CONF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD_CHX[15:0]															
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	DS_DATAOUT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DS_DATA[15:0]															
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	DF_DATAOUT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DF_DATA[15:0]															
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

偏移	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
0x4C	IRQ_STATUS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DF_OVRFL_IRQ	OVR_DF_IRQ	OVR_DS_IRQ	AWD_IRQ	EOS_IRQ	EODF_IRQ	EODS_IRQ	EOC_IRQ
	复位值																									0	0	0	0	0	0	0	0
0x50	IRQ_ENABLE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DF_OVRFL_IRQ_ENA	OVR_DF_IRQ_ENA	OVR_DS_IRQ_ENA	AWD_IRQ_ENA	EOS_IRQ_ENA	EODF_IRQ_ENA	EODS_IRQ_ENA	EOC_IRQ_ENA
	复位值																									0	0	0	0	0	0	0	0
0x54	TIMER_CONF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRECH_DELAY_SEL	VBIAS_PRECH_DELAY[7:0]								ADC_LDO_DELAY[7:0]							
	复位值																0	1	0	0	1	0	1	1	0	0	0	1	0	1	0	0	0

请参见“表 3. BlueNRG-LP 存储器映射和外设寄存器边界地址”，了解寄存器边界地址。

13 随机数发生器（RNG）

RNG 是一个以连续模拟噪声为基础的随机数发生器，在主机读数时提供一个 16 位的值。

13.1 特性

- AHB 从外设
- 提供由模拟量发生器产生的 16 位随机数
- 两个连续随机数之间的最短周期为 1.25 us（对应于 20 个 RNGCLK 周期）。当读取的发生时间与上一次过于接近时，该计算器在 AHB 总线上添加等待状态，这由一个拉间隔计数器自动管理
- 监控 RNG 的熵来标记异常行为（产生稳定值，或产生稳定的值序列）
- 2 个时钟域：
  - RNGCLK：16 MHz，用于规范化和移位器（特定的串行至 16 位并行转换）
  - HCLK：AHB 接口的 AHB 时钟（16 MHz、32 MHz 或 64 MHz）
- 可以禁用以便降低功耗。

功耗和 RNG：

内部自激振荡器的功耗相当高。在不使用 RNG 时可以停用这些振荡器

- 在 PORESETn 后，默认停用内部自激振荡器以限制功耗
- 在 PORESETn 后通过 RCC 使能 RNG 时钟树时，振荡器自动重启
- 如果软件通过 RCC 使能 RNG 时钟树，则其不停止内部振荡器。为了停止振荡器，软件必须首先将 RNG_CR[2] = RNG_DIS 置位。
- 另一方面，在重新使能 RNG 时钟树后，软件必须通过将 RNG_DIS 位清零来重启振荡器。

13.2 RNG 寄存器映射

请参见第 2.2.2 节“存储器映射和寄存器边界地址”了解寄存器边界地址。

注意： *尽管 RNG 寄存器通过 AHB 寻址，但只允许 32 位访问。任何 8 位或 16 位访问都会生成 AHB 错误，导致 Cortex-M0+上发生硬性故障。*

表 37. RNG 寄存器列表

偏移地址	名称	RW(1)	复位	说明
0x00	RNG_CR	RW	0x00000000	RNG 配置寄存器。参见表 38. RNG_CR 寄存器说明
0x04	RNG_SR	RO	0x00000000	RNG 状态寄存器。参见表 39. RNG_SR 寄存器说明
0x08	RNG_VAL	RO	0x00000000	RNG 16 位随机值。参见表 40. RNG_VAL 寄存器说明

1. *这些缩略语的含义如下：RW = 读和写；RO = 只读。*

13.3 RNG 寄存器说明

RNG_CR

该寄存器配置 RNG。

表 38. RNG_CR 寄存器说明

地址：RNG_BASE + 0x00				
位	字段名称	复位	R/W	说明
1:0	RESERVED	2'h0	RW	RESERVED
2	RNG_DIS	1'h0	RW	RNG 禁用位。 此位使能或禁用随机数发生器。 • 0：使能 RNG（默认） • 1：RNG 禁用。内部自由运行的振荡器进入下电模式，输入块时 RNG 时钟停止。
3	TST_CLK	1'h0	RW	RNG 测试时钟位。 将 1b 写入此位将启动检测 RNG_CLK 存在与否的逻辑。然后，等待（至少 4 个 RNGCLK 周期的超时）RNG_SR 寄存器中的 REVCLK = 1b。如果超时结束后 REVCLK = 0b，则表示 RNGCLK 不存在且读取 RNG_VAL 寄存器会触发 AHB 错误响应。 出于安全原因，软件应在读取随机值之前检查 RNGCLK 是否存在。 此位自动清零且始终读为 0。
31:4	RESERVED	28'h0000000	R	RESERVED

RNG_SR

该寄存器提供 RNG 的状态标志。

表 39. RNG_SR 寄存器说明

地址：RNG_BASE + 0x04				
位	字段名称	复位	R/W	说明
0	RNGRDY	1'h0	R	新的随机值就绪。 • 0：RNG_VAL 寄存器值尚未生效。如果执行对 RNG_VAL 的读访问，主机将暂停（通过在 AHB 总线上插入等待状态），直至有随机值可用。 • 1：RNG_VAL 寄存器包含有效的随机数。当 RNG 禁用（RNG_CR 中的 RNGDIS 位 = 1b）时，此位保持为 0b
1	REVCLK	1'h0	R	RNGCLK 时钟显示位。 将 1b 写入 RNG_CR 寄存器中的 TSTCLK 位会使此位复位。如果 RNGCLK 存在，在对 RNG_CR 的写操作结束后的 4 个 RNGCLK 周期后，此位为 1b。如果这段时间结束后 REVCLK = 0b，则表示 RNGCLK 不存在且读取 RNG_VAL 寄存器会触发 AHB 错误响应。
2	FAULT	1'h0	RC_W1	故障显示位。 当发生位序列错误时，此位由硬件置位。错误序列为： • 超过 32 个连续的相同值（0b 或 1b）位的序列 • 超过 16 个连续的 0band 1b 交替的序列（010101...01b）。 将 1b 写入此位以将其清零。写入 0b 无作用。
31:3	RESERVED	29'h00000000	R	RESERVED

RNG_VAL

在读取时，该寄存器提供 16 位随机值。在读取后，该寄存器只在 20 个 RNGCLK 周期后提供新的随机值。如果主机在这段时间结束前执行新的读取，RNG 将在 AHB 总线上插入等待状态。

表 40. RNG_VAL 寄存器说明

地址：RNG_BASE + 0x08				
位	字段名称	复位	R/W	说明
15:0	RANDOM_VALUE	16’h----	R	随机值
31:16	RESERVED	16’h0000	R	RESERVED

14 公钥加速器（PKA）

公钥加速器是一个 AHB 从模块，专门使用预定义的素数模数和预定义的曲线计算与 ECC（椭圆曲线加密）相关的加密公钥原语。

系统时钟 2 分频为 PKA 内核提供时钟，系统时钟直接为 PKA 存储器提供时钟。

14.1 特性

PKA 主要特性如下：

- 椭圆曲线 Diffie-Hellman（ECDH）公钥-私钥对计算加速器
- 基于 Montgomery 方法实现快速模乘运算
- 内置 Montgomery 域向内和向外变换
- AMBA AHB 精简版从接口具有更小的指令集
- 单端口内部存储器可供系统使用（当 BlueNRG-LPPKA 不使用它时）。

14.2 PKA 寄存器映射

器件通过 32 位宽控制寄存器（可通过 AMBA™ 2.0 版 AHB 总线进行访问）与 PKA 通信。“表 41. PKA 寄存器列表”中列出了这些寄存器，后面几页还提供了它们的详细说明。请参见“表 3. BlueNRG-LP 存储器映射和外设寄存器边界地址”。

表 41. PKA 寄存器列表

偏移地址	名称	RW[1]	复位	说明
0x00	PKA_CSR	RW	0x00000002	指令和状态寄存器。参见表 42. PKA_CSR 寄存器说明。
0x04	PKA_ISR	RW	0x00000000	中断寄存器。参见表 43. PKA_ISR 寄存器说明。
0x08	PKA_IEN	RW	0x00000000	中断使能寄存器。参见表 44. PKA_ISR 寄存器说明。

1. 这些缩略语的含义如下：RW = 读和写 RC = 读和写以清零。

14.3 PKA 寄存器说明

指令和状态寄存器（PKA_CSR）

表 42. PKA_CSR 寄存器说明

地址：PKA_BASE + 0x00				
位	字段名称	复位	R/W	说明
0	GO	1’h0	RW	PKA 开始处理指令。 • 写入 “0” 无效 • 写入 “1” 将启动加密引擎 在计算结束前，此位必须重新写回 0。
1	READY	1’h1	R	PKA 就绪状态。 • 0：PKA 仍在计算 • 1：PKA 可以开始新的计算 注意：如果 READY 位为高电平，则不能通过 AHB 接口访问 PKA。 READY 位的上升沿在 PKA_ISR 寄存器中置位 PROC_END 标志。
6:2	RESERVED	5’h00	-	保留，必须保持复位值。
7	SFT_RST	1’h0	RW	PKA 软件复位。 • 写入 0 会将位清零并释放 PKA 模块复位。 • 写入 1 将复位 PKA 模块。PKA RAM 内容不变。 *注意：当 SFT_RST 置位时，不阻止对 PKA 寄存器的访问，只有内核保持复位。*
31:8	RESERVED	5’h00	-	保留，必须保持复位值。

PKA 中断状态寄存器 PKA_ISR

PKA_ISR 寄存器给出 PKA 模块的中断状态。

如需将挂起中断清零，必须在相应的位中连续两次写入：写入 1’b1，然后是 1’b0。

表 43. PKA_ISR 寄存器说明

地址：PKA_BASE + 0x04				
位	字段名称	复位	R/W	说明
0	PROC_END	1’h0	RW	PKA 处理结束中断。读取时： • 0：未检测到新事件 • 1：PKA 处理结束（当 PKA_CSR.READY 位变为高电平时，此位置 1。） 写入时： 如需将挂起中断清零，用户必须在此位写入 1，随后通过写 0 操作将其清零。如果没有写入 0，会在 CPU 端发生下一个事件时生成中断（若已在 PKA_IER 中使能），但在中断处理程序在该寄存器中读取它时，标志显示为 0（因为清零操作仍然有效）。
1	RESERVED	1’h0	-	保留，必须保持复位值。
2	RAM_ERR	1’h0	RW	RAM 读/写访问错误中断。读取时： • 0：在 PKA 停止时，对 PKA RAM 的所有 AHB 读或写访问 • 1：在 PKA 正在工作和使用内部 RAM 时，对 PKA RAM 的所有 AHB 读或写访问。这些读或写访问无法成功，因为当 PKA 正在工作时，PKA 内部 RAM 已从 AHB 总线上断开（READY 位为低）。 写入时：

地址：PKA_BASE + 0x04				
位	字段名称	复位	R/W	说明
				如需将挂起中断清零，用户必须在此位写入 1，随后通过写入 0 将其清零。如果没有写入 0，会在 CPU 端发生下一个事件时生成中断（若已在 PKA_IER 中使能），但在中断处理程序在该寄存器中读取它时，标志显示为 0（因为清零操作仍然有效）。

表 44. PKA_ISR 寄存器说明

地址：PKA_BASE + 0x04				
位	字段名称	复位	R/W	说明
3	ADD_ERR	1'h0	RW	AHB 地址错误中断。读取时： • 0：在已映射地址范围内发生的对 PKA RAM 的所有 AHB 读或写访问 • 1：在未映射地址范围内发生的对 PKA RAM 的所有 AHB 读或写访问 写入时： 如需将挂起中断清零，用户必须在此位写入 1，随后通过写入 0 将其清零。如果没有写入 0，会在 CPU 端发生下一个事件时生成中断（若已在 PKA_IER 中使能），但在中断处理程序在该寄存器中读取它时，标志显示为 0（因为清零操作仍然有效）。
31:4	RESERVED	28'h0000000		保留，必须保持复位值。

PKA 中断使能寄存器 PKA_IEN

PKA_IEN 寄存器用于使能 PKA 中断。

表 45. KA_IEN 寄存器说明

地址：PKA_BASE + 0x08				
位	字段名称	复位	R/W	说明
0	READY_EN	1'h0	RW	READY 中断使能。 • 0：READY 中断禁用 • 1：READY 中断使能
1	RESERVED	1'h0	-	保留，必须保持复位值。
2	RAMERR_EN	1'h0	RW	RAM 访问错误中断使能。 • 0：RAM_ERR 中断禁用 • 1：RAM_ERR 中断使能
3	ADDERR_EN	1'h0	RW	AHB 地址错误中断使能。 • 0：ADD_ERR 中断禁用 • 1：ADD_ERR 中断使能
31:4	RESERVED	28'h0000000	-	保留，必须保持复位值。

14.4 编程模型

14.4.1 基本操作序列

使用 PKA 模块的典型序列如下：

1. 将输入数据加载到 PKA 内存（PKA_RAM）中。
2. 通过将 PKA_CSR 寄存器中的 GO 位置 1 来生成 GO 指令。
3. 等待 READY 置位（通过轮询 PKA_CSR 寄存器中的 READY 位或通过 PROC_END 中断）。
4. 从 PKA 内部存储器中拷贝出计算结果。

14.4.2 PKA_RAM 中的数据位置

PKA_RAM 中的输入和输出数据有特定位置。位置说明见“表 46. ECC 标量乘法数据位置”。

表 46. ECC 标量乘法数据位置

参数说明	助记符	地址（十进制）	大小（字）	PKA_RAM 偏移地址
输入				
kP 的‘k’	ecc_addr_k	27	EOS(1)	0x6C
起点 P，坐标 X、Y	ecc_addr_px ecc_addr_py	36 45	2*EOS(1)	0x90 0xB4
输出				
结果的坐标 X、Y	ecc_addr_px ecc_addr_py	36 45	2*EOS(1)	0x90 0xB4
误差	ecc_addr_kp_error	0	1	0x00

1. *EOS：ECC 操作数大小。*

如果输入点不是有效点因而不满足曲线等式，则错误字段返回 1。在这种情况下，会很快完成计算。如果计算结束时错误字段为 0，则结果应视为有效。用以下公式计算数据的最大长度：

EOS =（max_ecc_size/word_size）+ 1。

示例 1

如果使用 ECC P256，则操作数需要（256/32 + 1）字，PKA 内核需要 9 字。在加载 256 位（8 字）输入时，将需要一个额外的字且必须用零填充。

15 循环冗余校验计算单元（CRC）

15.1 引言

CRC（循环冗余校验）计算单元使用多项式发生器从一个 8 位/16 位/32 位的数据字中产生 CRC 码。

在众多的应用中，基于 CRC 的技术还常用来验证数据传输或存储的完整性。根据功能安全标准的规定，这些技术提供了验证 Flash 完整性的方法。CRC 计算单元有助于在运行期间计算软件的签名，并将该签名与链路时生成并存储在指定存储单元的参考签名加以比较。

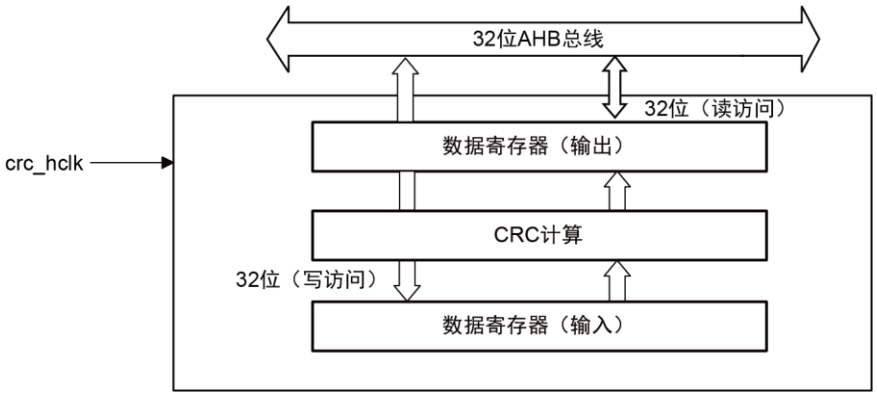
15.2 CRC 主要特性

- 位数可编程的（7 位、8 位、16 位和 32 位）的完全可编程多项式。
- 处理 8 位、16 位、32 位数据大小
- 可编程 CRC 初始值
- 单输入/输出 32 位数据寄存器
- 输入缓冲器可避免计算期间发生总线阻塞
- 对于 32 位数据大小，CRC 计算在 4 个 AHB 时钟周期(HCLK）内完成
- 8 位通用寄存器（可用于临时存储）
- I/O 数据的可逆性选项。

15.3 CRC 功能说明

15.3.1 CRC 框图

图 27. **CRC 计算单元框图**



15.3.2 CRC 操作

CRC 计算单元具有单个 32 位读/写数据寄存器（CRC_DR）。它用于输入新数据（写访问）和保存之前 CRC 计算的结果（读访问）。

对数据寄存器的每个写操作都会对之前的 CRC 值（存储在 CRC_DR 中）和新值再做一次 CRC 计算。CRC 计算针对整个 32 位数据字或逐个字节完成，具体取决于数据的写入格式。

CRC_DR 寄存器可按字、右对齐半字和右对齐字节进行访问。对于其它寄存器，只允许进行 32 位访问。

计算时间取决于数据宽度：

- 32 位数据需要 4 个 AHB 时钟周期
- 16 位数据需要 2 个 AHB 时钟周期
- 8 位数据需要 1 个 AHB 时钟周期。

输入缓冲器中可立即写入第二个数据，无需因之前的 CRC 计算而等待任何等待状态。

可动态调整数据大小，从而能最大程度地减少给定字节数的写访问次数。例如，对 5 个字节进行 CRC 计算时，可先写入字，然后写入字节。

输入数据的顺序可反转，以管理各种数据存放方式（双字/单字/字节，大端/小端等等）。可对 8 位、16 位和 32 位数据执行反转操作，具体取决于 CRC_CR 寄存器中的 REV_IN[1:0] 位。

例如，输入数据 0x1A2B3C4D 在 CRC 计算中用作：

- 按字节执行位反转的 0x58D43CB2
- 按半字执行位反转的 0xD458B23C
- 按全字执行位反转的 0xB23CD458。

通过将 CRC_CR 寄存器中 REV_OUT 位置 1 也可以将输出数据反转。

该操作按位进行：例如，输出数据 0x11223344 将转换为 0x22CC4488。

使用 CRC_CR 寄存器中的 RESET 控制位可将 CRC 计算器初始化为可编程值（默认值为 0xFFFFFFFF）。

可使用 CRC_INIT 寄存器对 CRC 初始值进行编程。对 CRC_INIT 寄存器进行写访问时会自动初始化 CRC_DR 寄存器。

CRC_IDR 寄存器可用于保存与 CRC 计算相关的临时值。它不受 CRC_CR 寄存器中的 RESET 位影响。

多项式可编程

多项式系数可完全通过 CRC_POL 寄存器进行编程，通过编程 CRC_CR 寄存器中的 POLYSIZE[1:0] 位可将多项式大小配置为 7 位、8 位、16 位或 32 位。不支持偶次多项式。

如果 CRC 数据小于 32 位，可从 CRC_DR 寄存器的最低有效位读取 CRC 的值。

为实现可靠的 CRC 计算，CRC 计算期间不能实时更改多项式的值或大小。因此，如果正在执行 CRC 计算，则在更改多项式前，应用程序必须先复位，或者先执行 CRC_DR 读操作。

默认多项式值为 CRC-32（以太网）多项式：0x4C11DB7。

15.4 CRC 寄存器

15.4.1 数据寄存器（CRC_DR）

偏移地址：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								

位 31:0	**DR[31:0]**：数据寄存器位 该寄存器用于向 CRC 计算器写入新数据。读取寄存器时可读出之前的 CRC 计算结果。 如果数据大小小于 32 位，则最低有效位可用于写入/读取正确值。

15.4.2 独立数据寄存器（CRC_IDR）

偏移地址：0x04

复位值：0x0000 0000

15.4.3 控制寄存器（CRC_CR）

偏移地址：0x08

复位值：0x0000 0000

位 31:8	保留，必须保持清零。
位 7	REV_OUT：反转输出数据 该位用于控制输出数据位顺序的反转。0：位顺序不受影响 1：位反转输出格式
位 6:5	**REV_IN[1:0]**：反转输入数据。 这些位用于控制输入数据位顺序的反转。 00：位顺序不受影响 01：按字节执行位倒序 10：按半字执行位倒序 11：按字执行位倒序
位 4:3	**POLYSIZE[1:0]**：多项式大小。 这些位用于控制多项式的大小。 00：32 位多项式 01：16 位多项式 10：8 位多项式 11：7 位多项式
位 2:1	保留，必须保持清零。

位 0	**RESET**：RESET 位 此位由软件置 1，用于复位 CRC 计算单元并将数据寄存器设置为存储在 CRC_INIT 寄存器中的值。此位只能置 1，将由硬件自动进行清零。

15.4.4 CRC 初始值（CRC_INIT）

偏移地址：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_INIT[15:0]															
								rw							

位 31:0	**CRC_INIT：***可编程初始 CRC 值。*此寄存器用于写入 CRC 初始值。

15.4.5 CRC 多项式（CRC_POL）

偏移地址：0x14

复位值：0x04C11DB7

15.5 CRC 寄存器映射

表 47. CRC 寄存器映射和复位值

偏移	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
0x00	**CRC_DR**	DR[31:0]																															
	复位值	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**	IDR[31:0]																															
	复位值	0	0	0	0	0	0	0	0	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	**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[1:0]		POLYSIZE[1:0]		Res.	Res.	RESET
	复位值																									0	0	0					0
0x10	**CRC_INIT**	CRC_INIT[31:0]																															
	复位值	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
0x14	**CRC_POL**	POL[31:0]																															
	复位值	0x04C11DB7																															

请参见第 2.2.2 节“存储器映射和寄存器边界地址”了解寄存器边界地址。

16 高级控制定时器（TIM1）

在本节中，由于 BlueNRG-LP 器件中只有该定时器的一个实例，因此“TIMx”应理解为“TIM1”。

16.1 TIM1 introduction

高级控制定时器（TIM1）包含 16 位自动重载计数器，该计数器由可编程预分频器驱动。

此类定时器可用于各种用途，包括测量输入信号的脉冲长度（输入捕获），或者生成输出波形（输出比较、PWM 和带死区插入的互补 PWM）。

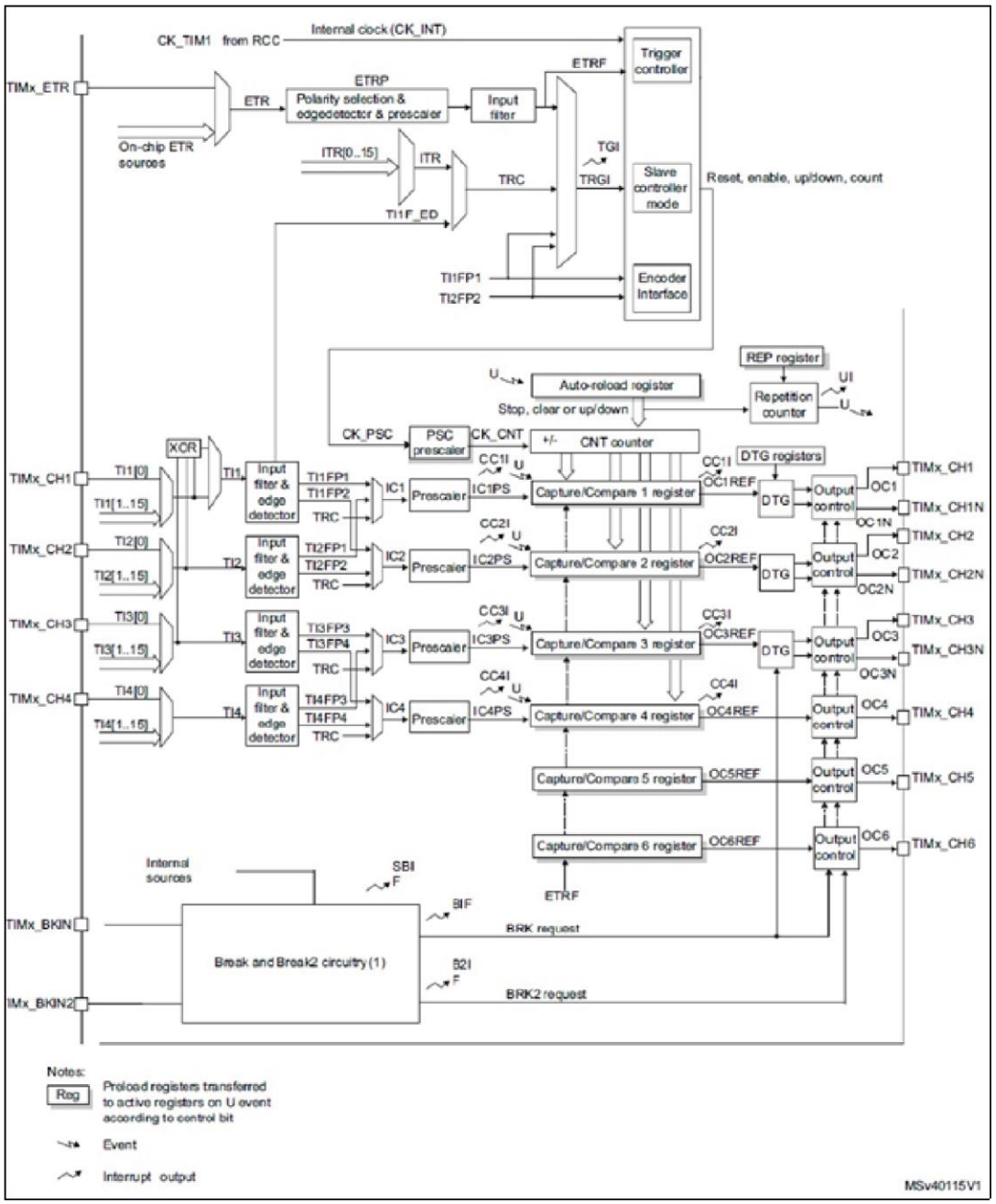
通过对定时器输入时钟（64 MHz）使用定时器预分频器，可将脉冲长度和波形周期从几微秒调制成若干毫秒。

16.2 TIM1 主要特性

TIM1 定时器特性包括：

- 16 位递增、递减、递增/递减自动重载计数器
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（可实时修改），分频系数介于 1 到 65536 之间。
- 多达 6 个独立通道，可用于：
  - 输入捕获（通道 5 和 6 除外）
  - 输出比较
  - PWM 生成（边沿和中心对齐模式）
  - 单脉冲模式输出
- 具有可编程死区时间的互补输出
- 同步电路，用于通过外部信号控制定时器
- 重复计数器，用于仅在给定数目的计数器周期后更新定时器寄存器
- 2 个断路输入，用于将定时器的输出信号置于用户可选的安全配置中
- 发生如下事件时生成中断：
  - 更新：计数器上溢/下溢、计数器初始化（通过软件或内部/外部触发）
  - 触发事件（计数器启动、停止、初始化或通过内部/外部触发计数）
  - 输入捕获
  - 输出比较
- 支持增量式（正交）编码器用于定位
- 触发输入用作外部时钟或逐周期电流管理。

图 28. 高级控制定时器框图



16.3 TIM1 功能说明

16.3.1 时基单元

可编程高级控制定时器的主要模块是一个 16 位计数器及其相关的自动重载寄存器。计数器可递增计数、递减计数或交替进行递增和递减计数。计数器的时钟可通过预分频器进行分频。

计数器、自动重载寄存器和预分频器寄存器可通过软件进行读写。即使在计数器运行时也可执行读写操作。

时基单元包括：

- 计数器寄存器（TIMx_CNT）
- 预分频器寄存器（TIMx_PSC）
- 自动重载寄存器（TIMx_ARR）
- 重复计数器寄存器（TIMx_RCR）

自动重载寄存器是预装载的。对自动重载寄存器执行写入或读取操作时会访问预装载寄存器。预装载寄存器的内容既可以直接传送到影子寄存器，也可以在每次发生更新事件（UEV）时传送到影子寄存器，这取决于 TIMx_CR1 寄存器中的自动重载预装载使能位（ARPE）。当计数器达到上溢值（或者在递减计数时达到下溢值）并且 TIMx_CR1 寄存器中的 UDIS 位为 0 时，将发送更新事件。该更新事件也可由软件产生。下文将针对各配置的更新事件的产生进行详细介绍。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIMx_CR1 寄存器中的计数器启动位（CEN）置 1 时，才会启动计数器（有关计数器使能的更多详细信息，另请参见从模式控制器的相关说明）。

注意，计数器将在 TIMx_CR1 寄存器的 CEN 位置 1 时刻的一个时钟周期后开始计数。

预分频器说明

预分频器可对计数器时钟频率进行分频，分频系数介于 1 和 65536 之间。该预分频器基于 TIMx_PSC 寄存器中的 16 位寄存器所控制的 16 位计数器。由于该控制寄存器具有缓冲功能，因此预分频器可实现实时更改。而新的预分频比将在下一更新事件发生时被采用。

图 29. 预分频器分频由 1 变为 2 时的计数器时序图以及图 30. 预分频器分频由 1 变为 4 时的计数器时序图给出了预分频因子发生实时变化时计数器行为的一些示例

图 29. 预分频器分频由 1 变为 2 时的计数器时序图

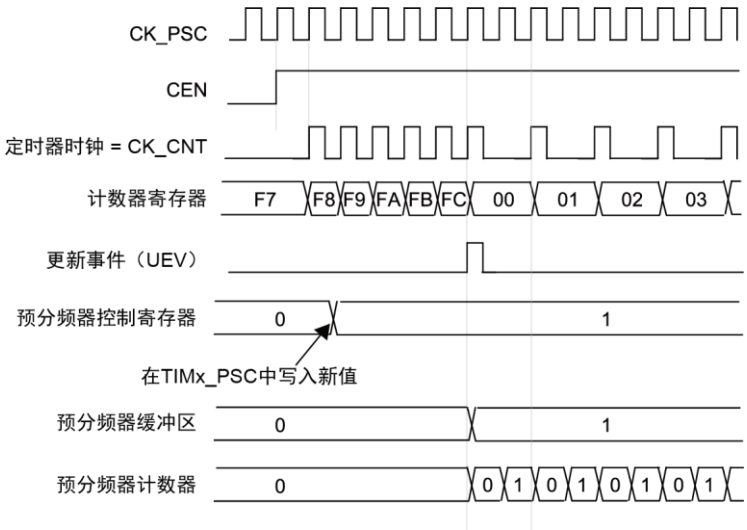
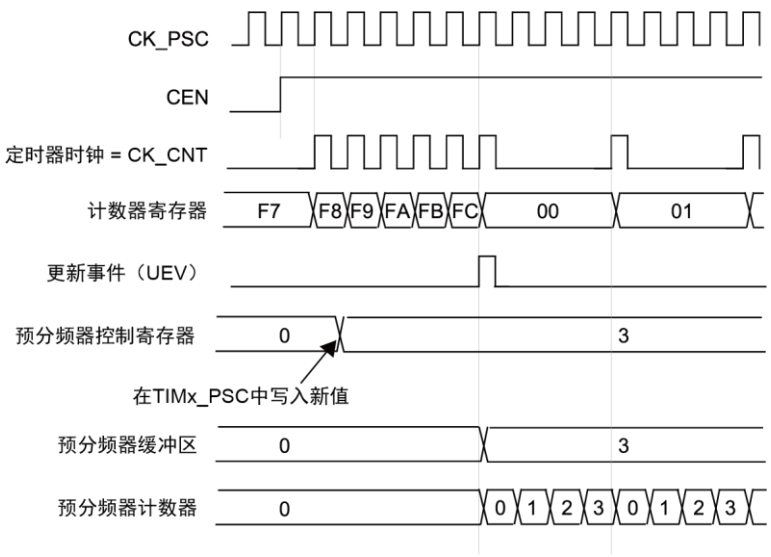


图 30. 预分频器分频由 1 变为 4 时的计数器时序图



16.3.2 计数器模式

向上计数模式

在递增计数模式下，计数器从 0 计数到自动重载值（TIMx_ARR 寄存器的内容），然后重新从 0 开始计数并生成计数器上溢事件。

如果使用重复计数器，则当递增计数的重复次数达到重复计数器寄存器中编程的次数加一次（(TIMx_RCR）+ 1）后，将生成更新事件（UEV）。否则，将在每次计数器上溢时生成更新事件。

将 TIMx_EGR 寄存器的 UG 位置 1（通过软件或使用从模式控制器）时，也将产生更新事件。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIMx_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 重复计数器中将重新装载 TIMx_RCR 寄存器的内容
- 使用预装载值（TIMx_ARR）更新自动重载影子寄存器
- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）。

以下各图以一些示例说明当 TIMx_ARR=0x36 时不同时钟频率下计数器的行为。

图 31. 计数器时序图，1 分频内部时钟

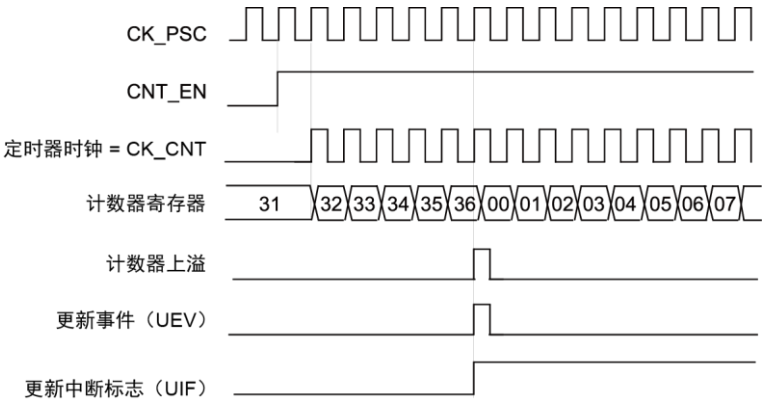


图 32. 计数器时序图，2 分频内部时钟

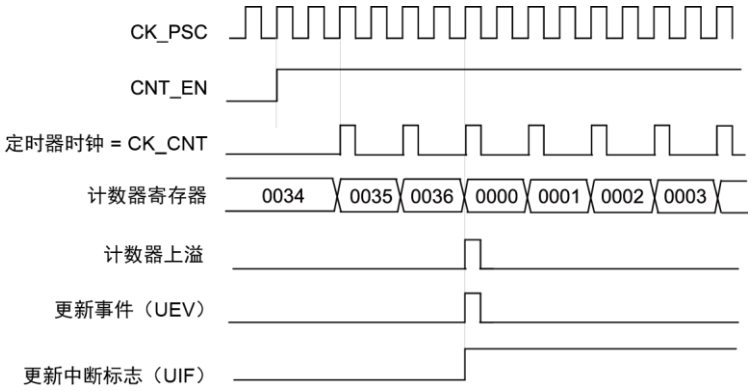


图 33. 计数器时序图，4 分频内部时钟

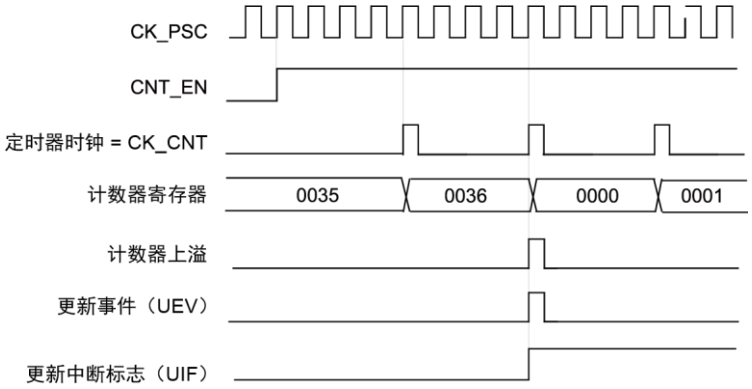


图 34. 计数器时序图，N 分频内部时钟

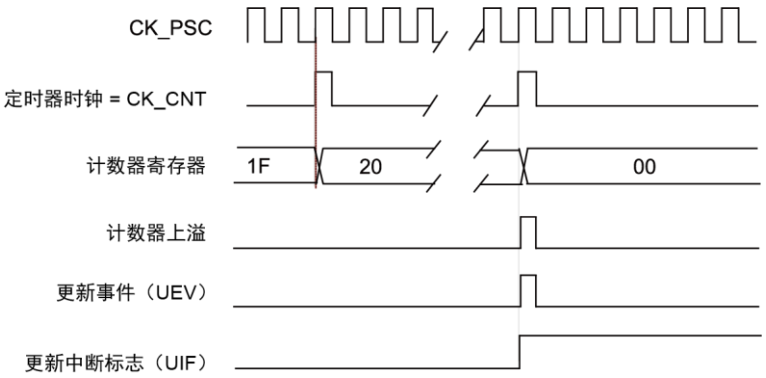


图 35. 计数器时序图，ARPE=0 时更新事件（TIMx_ARR 未预装载）

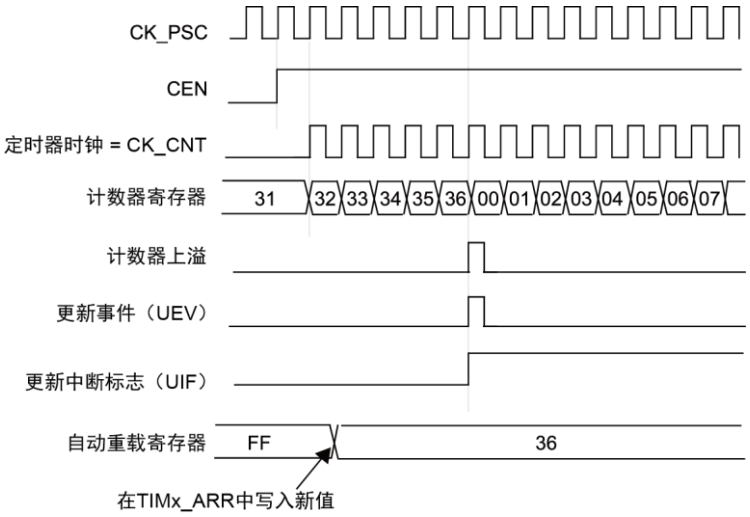
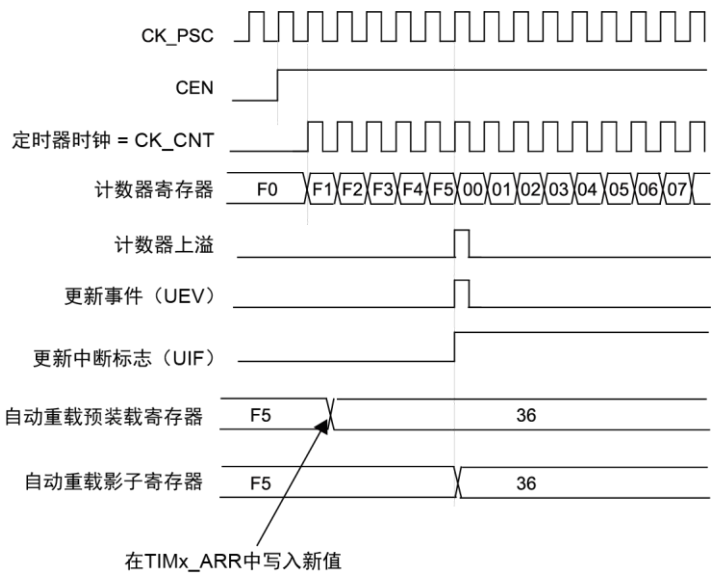


图 36. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 已预装载）



递减计数模式

在递减计数模式下，计数器从自动重载值（TIMx_ARR 寄存器的内容）开始递减计数到 0，然后重新从自动重载值开始计数并生成计数器下溢事件。

如果使用重复计数器，则当递减计数的重复次数达到重复计数器寄存器中编程的次数加一次（(TIMx_RCR）+ 1 后，将生成更新事件（UEV）。否则，将在每次计数器下溢时生成更新事件。

将 TIMx_EGR 寄存器的 UG 位置 1（通过软件或使用从模式控制器）时，也将产生更新事件。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器会重新从当前自动重载值开始计数，而预分频器计数器则重新从 0 开始计数（但预分频比保持不变）。

此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIMx_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 重复计数器中将重新装载 TIMx_RCR 寄存器的内容
- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）
- 自动重载活动寄存器将以预装载值（TIMx_ARR 寄存器的内容）进行更新。注意，自动重载活动寄存器更新在计数器重载之前被更新，因此下一个周期就是预期的值。

以下各图以一些示例说明当 TIMx_ARR=0x36 时不同时钟频率下计数器的行为。

图 37. 计数器时序图，1 分频内部时钟

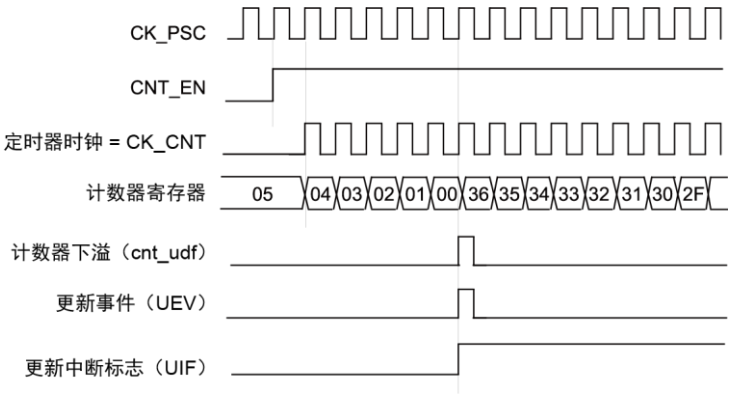


图 38. 计数器时序图，2 分频内部时钟

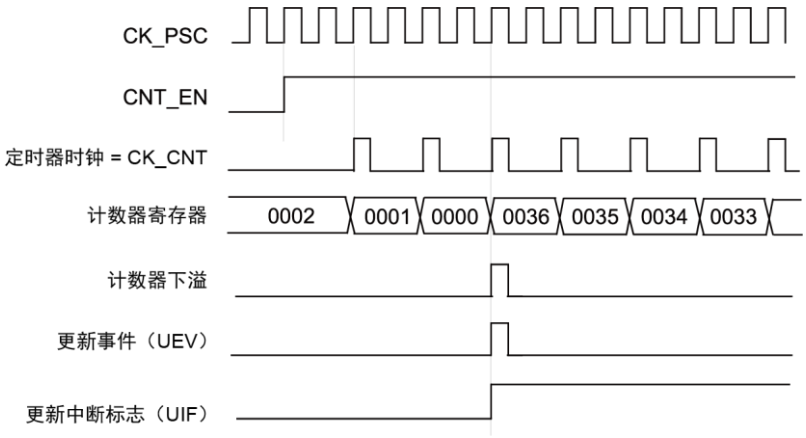


图 39. 计数器时序图，4 分频内部时钟

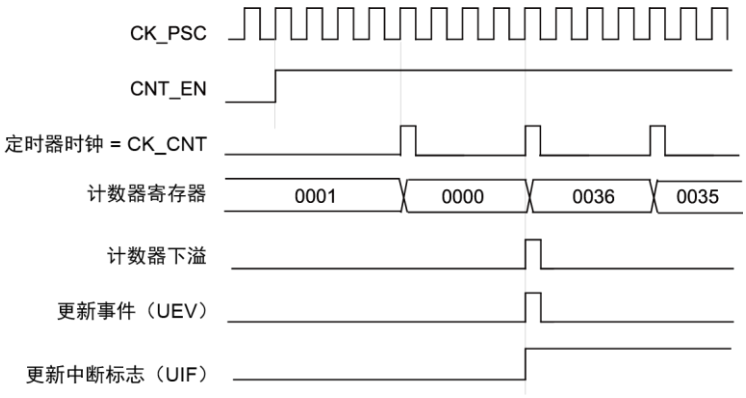


图 40. 计数器时序图，N 分频内部时钟

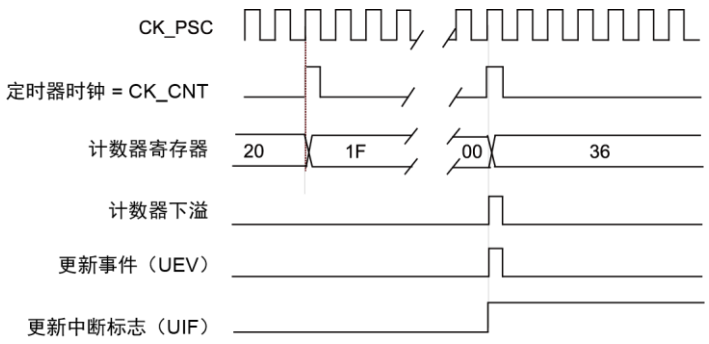
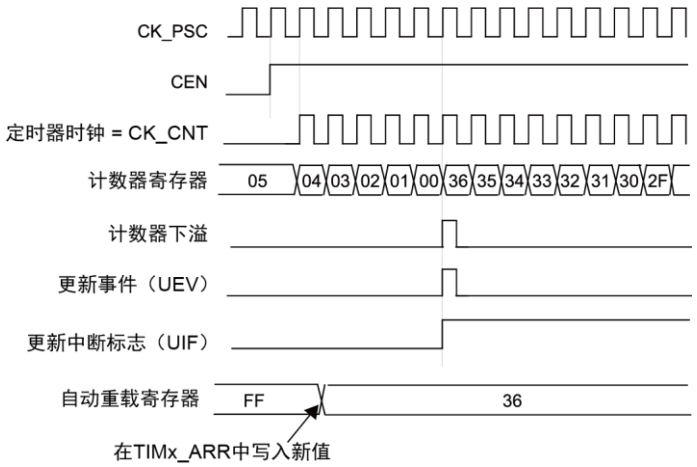


图 41. 计数器时序图，未使用重复计数器时更新事件



中心对齐模式（递增/递减计数）

在中心对齐模式下，计数器从 0 开始计数到自动重载值（TIMx_ARR 寄存器的内容）– 1，生成计数器上溢事件；然后从自动重载值开始向下计数到 1 并生成计数器下溢事件。之后从 0 开始重新计数。

当 TIMx_CR1 寄存器中的 CMS 位不为“00”时，中心对齐模式有效。将通道配置为输出模式时，其输出比较中断标志将在以下模式下置 1，即：计数器递减计数（中心对齐模式 1，CMS = “01”）、计数器递增计数（中心对齐模式 2，CMS = “10”）以及计数器递增/递减计数（中心对齐模式 3，CMS = “11”）。

在此模式下，TIMx_CR1 寄存器的 DIR 方向位不可写入值，而是由硬件更新并指示当前计数器方向。

每次发生计数器上溢和下溢时都会生成更新事件，或将 TIMx_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件。这种情况下，计数器以及预分频器计数器将重新从 0 开始计数。

UEV 更新事件可通过软件禁止，只需将 TIMx_CR1 寄存器中的 UDIS 位置 1。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器仍会根据当前自动重载值进行递增和递减计数。

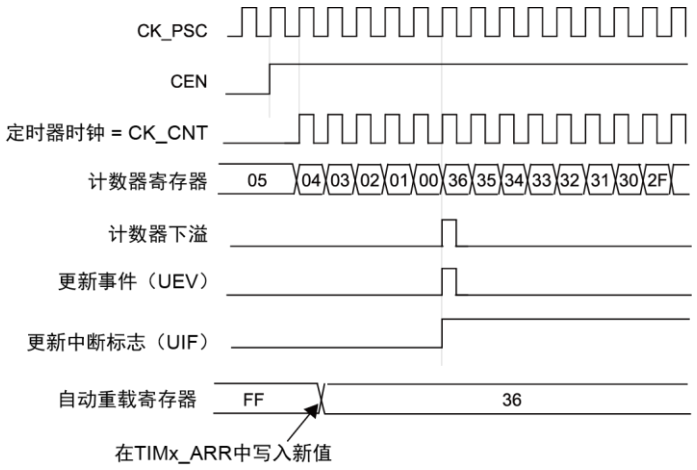
此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成 UEV 更新事件，但不会将 UIF 标志置 1（因此，不会发送任何中断）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIMx_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 重复计数器中将重新装载 TIMx_RCR 寄存器的内容
- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）
- 自动重载活动寄存器将以预装载值（TIMx_ARR 寄存器的内容）进行更新。注意，如果更新操作是由计数器上溢触发的，则自动重载活动寄存器在计数器重载之前更新，因此，下一个计数周期就是我们所希望的新的周期长度（计数器被重载新的值）。

以下各图以一些示例说明不同时钟频率下计数器的行为。

图 42. 计数器时序图，1 分频内部时钟，TIMx_ARR = 0x6



注意：　这里使用中心对齐模式 1（参见第 16.4 节“TIM1 寄存器”了解更多信息）。

图 43. 计数器时序图，2 分频内部时钟

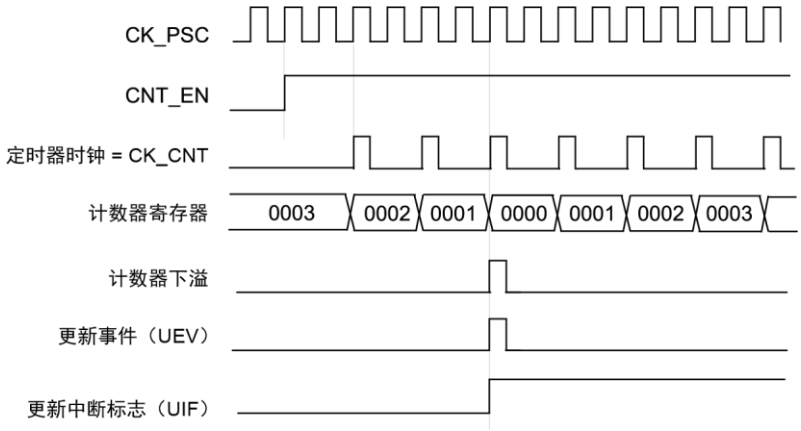
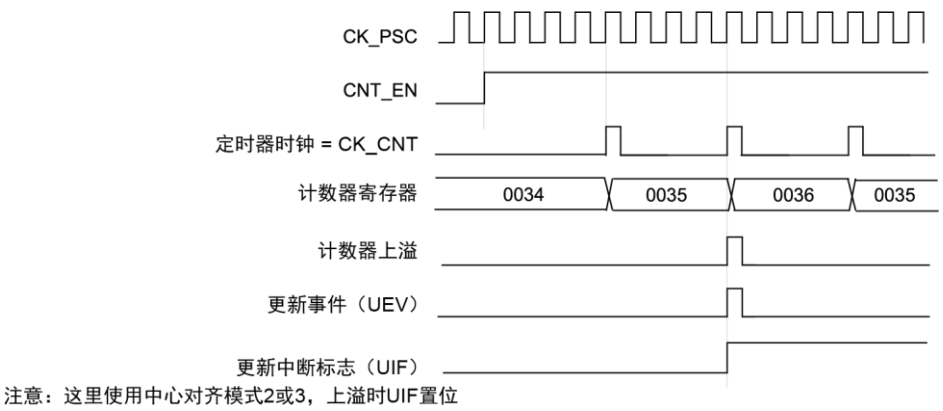


图 44. 计数器时序图，4 分频内部时钟，TIMx_ARR=0x36



注意：这里使用中心对齐模式2或3，上溢时UIF置位

图 45. 计数器时序图，N 分频内部时钟

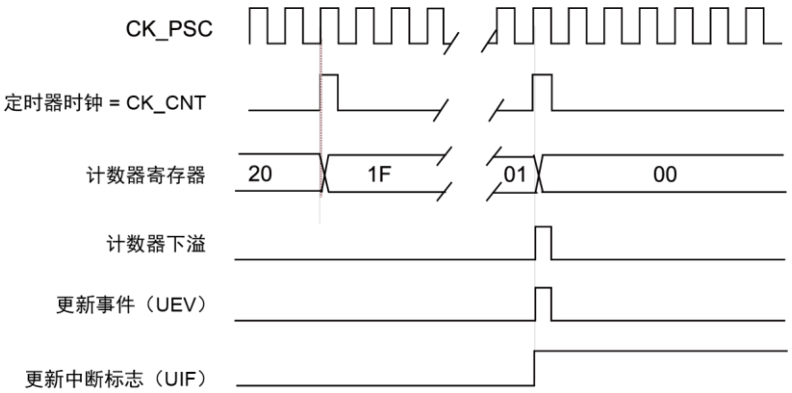


图 46. 计数器时序图，ARPE=1 时的更新事件（计数器下溢）

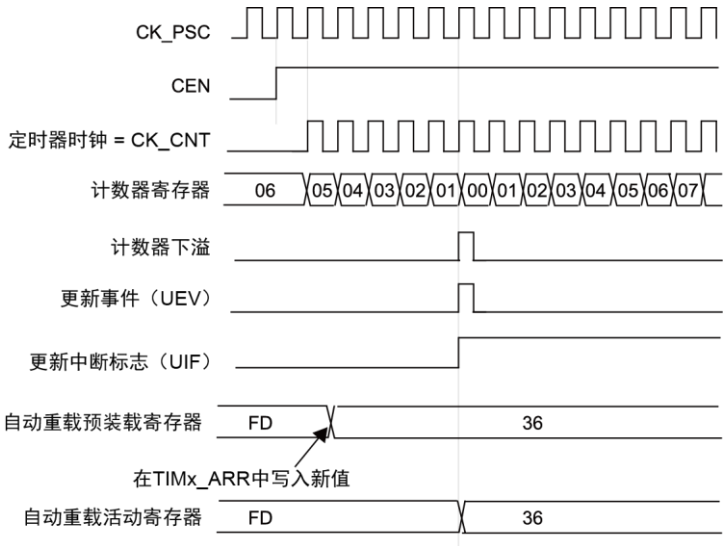
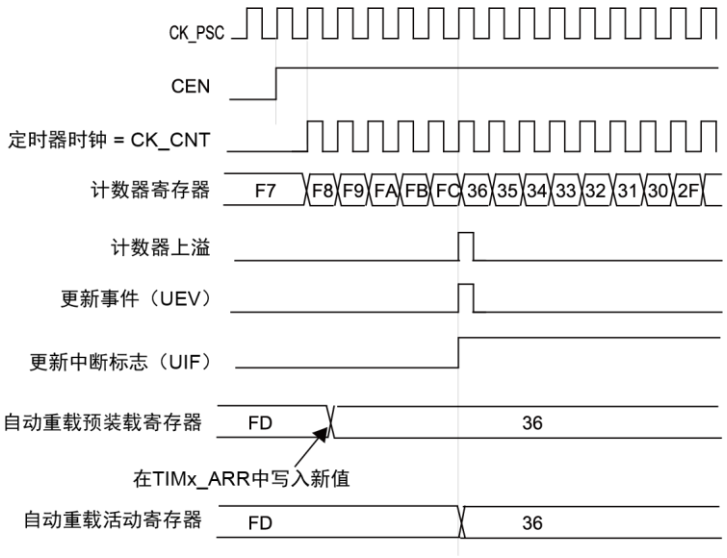


图 47. 计数器时序图，ARPE=1 时的更新事件（计数器上溢）



16.3.3 重复计数器

第 16.3.1 节“时基单元”描述了如何就计数器上溢/下溢生成更新事件（UEV）。实际上，只有当重复计数器达到零时，才会生成更新事件。这在生成 PWM 信号时很有用。

这意味着，每当发生 N+1 个计数器上溢或下溢（其中，N 是 TIMx_RCR 重复计数器寄存器中的值），数据就将从预装载寄存器转移到影子寄存器（TIMx_ARR 自动重载寄存器、TIMx_PSC 预分频器寄存器以及比较模式下的 TIMx_CCRx 捕获/比较寄存器）。

重复计数器在下列情况下递减：

- 递增计数模式下每当计数器上溢时
- 递减计数模式下每当计数器下溢时
- 中心对齐模式下每个计数器上溢和计数器下溢。尽管这使得最大重复次数不超过 32768 个 PWM 周期，但在每个 PWM 周期内可更新占空比两次。当在中心对齐模式下，每个 PWM 周期仅刷新一次比较寄存器时，由于模式的对称性，最大分辨率为 2xTck。

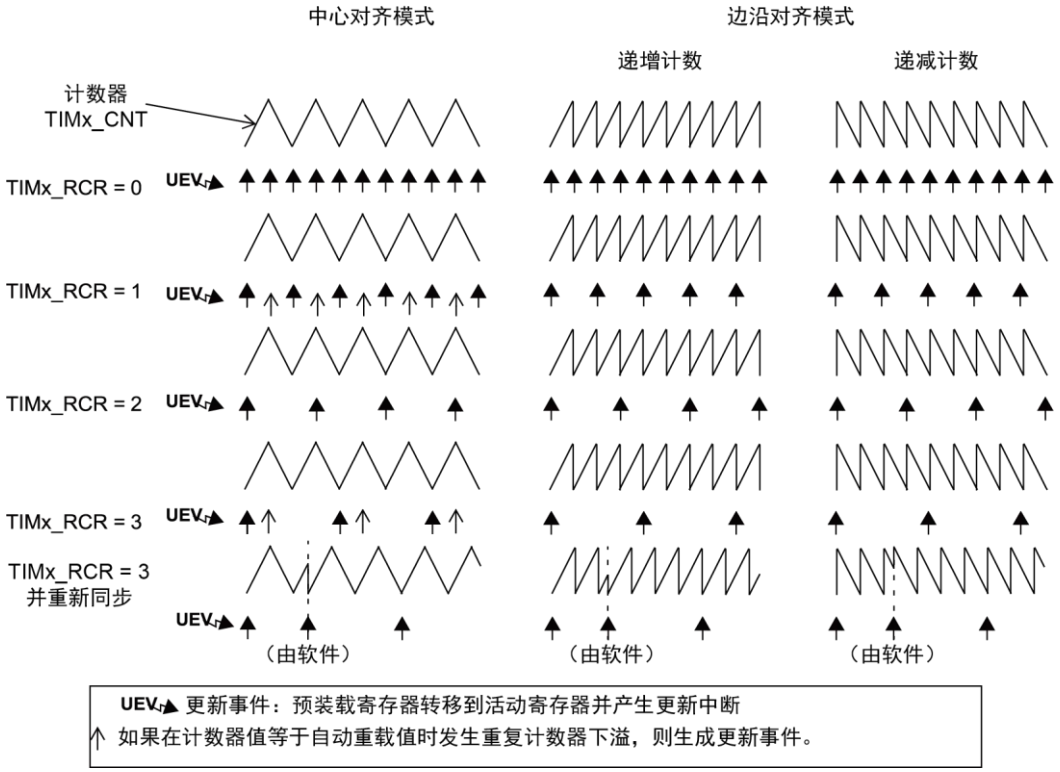
重复计数器是自动重载类型；其重复率为 TIMx_RCR 寄存器所定义的值（请参见图 48. 不同模式和 TIMx_RCR 寄存器设置下的更新频率示例）。

当更新事件由软件（通过将 TIMx_EGR 寄存器的 UG 位置 1）或硬件（通过从模式控制器）生成时，无论重复计数器的值为多少，更新事件都将立即发生，并且在重复计数器中重新装载 TIMx_RCR 寄存器的内容。

在中心对齐模式下，如果 RCR 值为奇数，更新事件将在上溢或下溢时发生，这取决于何时写入 RCR 寄存器以及何时启动计数器：如果在启动计数器前写入 RCR，则 UEV 在上溢时发生。如果在启动计数器后写入 RCR，则 UEV 在下溢时发生。

例如，如果 RCR = 3，UEV 将在每个周期的第四个上溢或下溢事件时产生（取决于何时写入 RCR）。

图 48. 不同模式和 TIMx_RCR 寄存器设置下的更新频率示例



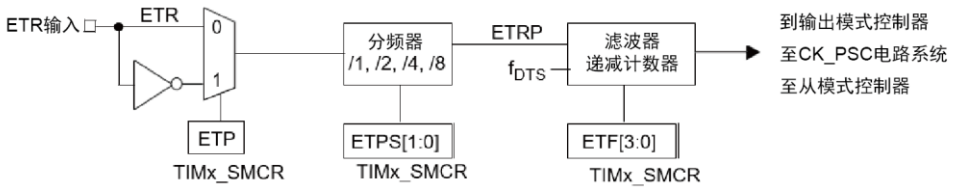
16.3.4 外部触发输入

定时器具有外部触发输入 ETR。它可以用作：

- 外部时钟（外部时钟模式 2，参见第 16.3.5 节）
- 从模式的触发（参见第 16.4 节“TIM1 寄存器”）
- 用于逐周期电流调节的 PWM 复位输入（第 16.3.17 节“在发生外部事件时将 OCxREF 信号清零”）。

图 49. 外部触发输入模块下面描述 ETR 输入条件。用 TIMx_SMCR 寄存器中的 ETP 位定义输入极性。可使用通过 ETPS[1:0] 位字段设定的分频因子对触发信号进行预分频，并通过 ETF[3:0] 位字段使能数字滤波。

图 49. 外部触发输入模块



ETR 输入来自输入引脚（参见“表 7. GPIO 复用选项 AF3 - AF4”和“表 8. I/O 模拟特性映射”）。

16.3.5 时钟选择

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

注意：只有通道 1 和通道 2 支持外部时钟模式 1。

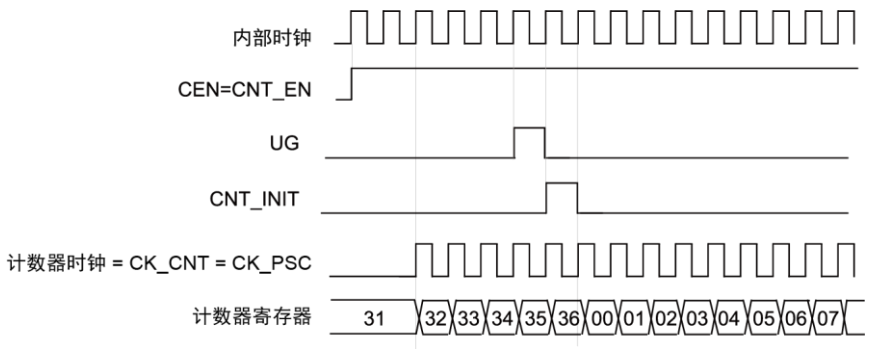
- 外部时钟模式 2：外部触发输入 ETR
- 编码器模式。

内部时钟源（CK_INT）

如果禁止从模式控制器（SMS=000），则 CEN 位、DIR 位（TIMx_CR1 寄存器中）和 UG 位（TIMx_EGR 寄存器中）为实际控制位，并且只能通过软件进行更改（UG 除外，仍保持自动清零）。当对 CEN 位写入 1 时，预分频器的时钟就由内部时钟 CK_INT 提供。

图 50. 正常模式下的控制电路，1 分频内部时钟显示了正常模式下的控制电路和递增计数器的行为（无预分频器）。

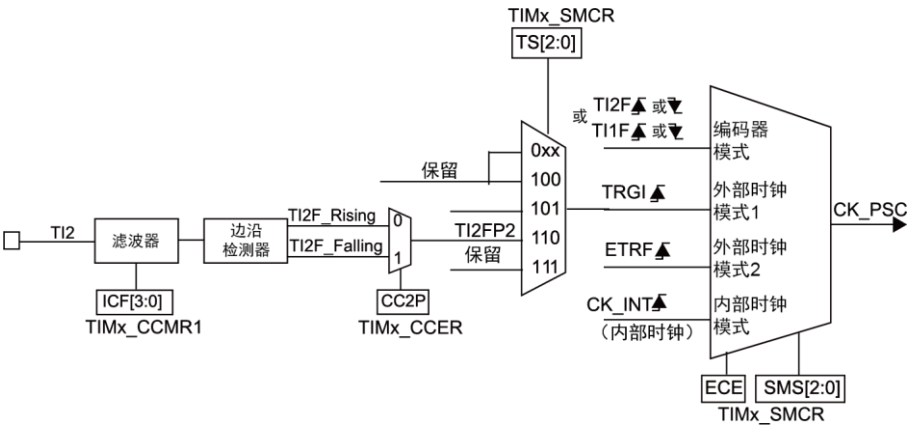
图 50. 正常模式下的控制电路，1 分频内部时钟



外部时钟源模式 1

当 TIMx_SMCR 寄存器中的 SMS=111 时，可选择此模式。计数器可在选定的输入信号上出现上升沿或下降沿时计数。

图 51. TI2 外部时钟连接示例



例如，要配置递增计数器在 TI2 输入出现上升沿时计数，请执行以下步骤：

1. 通过 TIMx_TISEL 寄存器中的 TI2SEL[3:0] 位选择合适的 TI2x 源（内部或外部）。

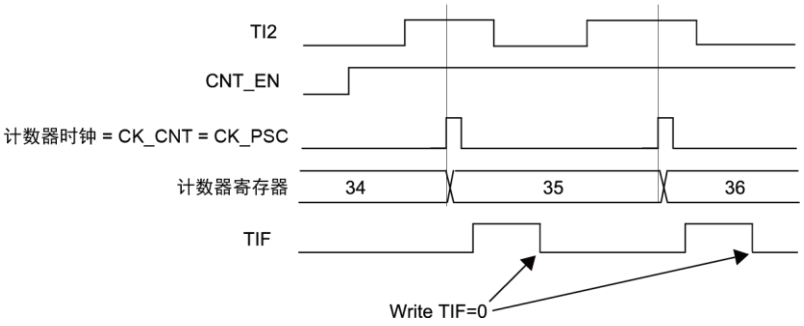
2. 通过在 TIMx_CCMR1 寄存器中写入 CC2S = “01”来配置通道 2，使其能够检测 TI2 输入的上升沿。
3. 通过在 TIMx_CCMR1 寄存器中写入 IC2F[3:0] 位来配置输入滤波带宽（如果不需要任何滤波器，请保持 IC2F=0000）。
4. 通过在 TIMx_CCER 寄存器中写入 CC2P=0 和 CC2NP=0 来选择上升沿极性。
5. 通过在 TIMx_SMCR 寄存器中写入 SMS=111，配置定时器在外部时钟模式 1。
6. 通过在 TIMx_SMCR 寄存器中写入 TS=110 来选择 TI2 作为触发输入源。
7. 通过在 TIMx_CR1 寄存器中写入 CEN=1 来使能计数器。

注意： *捕获预分频器不用于触发，因此无需配置它。*

当 TI2 出现上升沿时，计数器便会计数一次并且 TIF 标志置 1。

TI2 的上升沿与实际计数器时钟之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 52. 外部时钟模式 1 下的控制电路



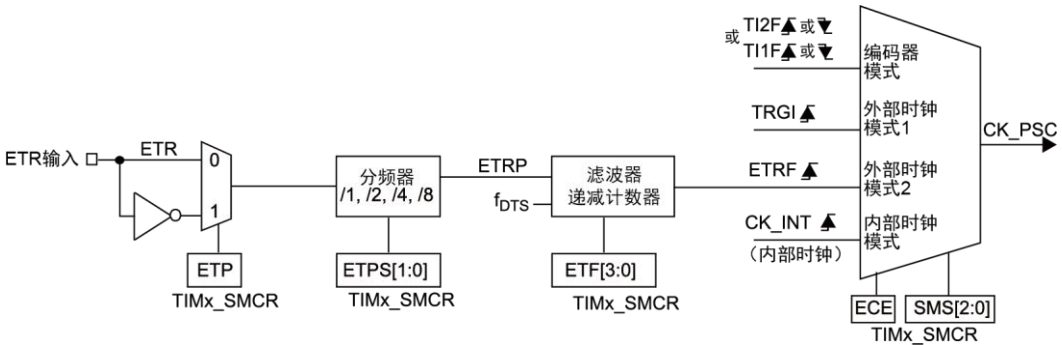
外部时钟源模式 2

通过在 TIMx_SMCR 寄存器中写入 ECE=1 可选择此模式。

计数器可在外部触发输入 ETR 出现上升沿或下降沿时计数。

见下面的图 53. 外部触发输入模块。

图 53. 外部触发输入模块



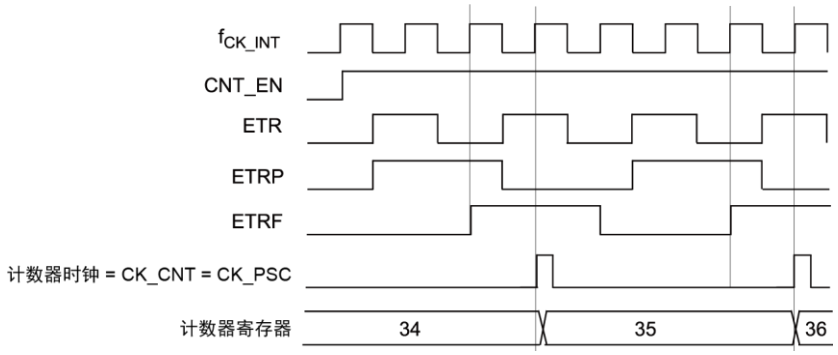
例如，要配置递增计数器在 ETR 每出现 2 个上升沿时计数，请执行以下步骤：

1. 由于此例中不需滤波器，因此在 TIMx_SMCR 寄存器中写入 ETF[3:0]=0000。
2. 通过在 TIMx_SMCR 寄存器中写入 ETPS[1:0]=01 来设置预分频器。
3. 通过在 TIMx_SMCR 寄存器中写入 ETP=0 来选择 ETR 引脚的上升沿检测。
4. 通过在 TIMx_SMCR 寄存器中写入 ECE=1 来使能外部时钟模式 2。
5. 通过在 TIMx_CR1 寄存器中写入 CEN=1 来使能计数器。

ETR 每出现 2 个上升沿，计数器计数一次。

ETR 的上升沿与实际计数器时钟之间的延迟是由于 ETRP 信号的重新同步电路引起的。

图 54. 外部时钟模式 2 下的控制电路



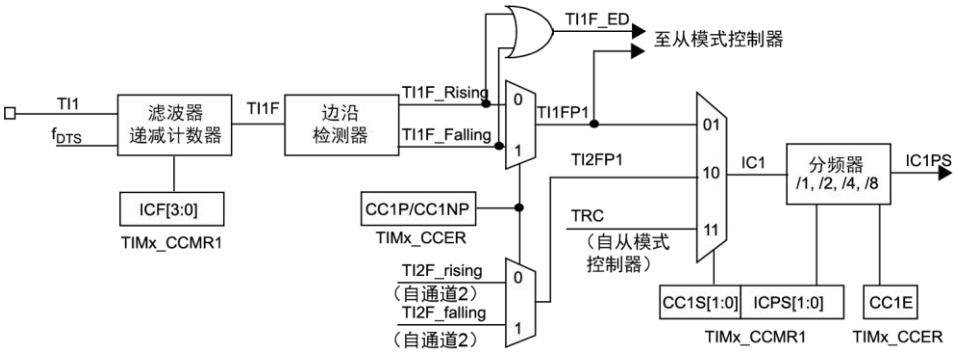
16.3.6 捕获/比较通道

每个捕获/比较通道均围绕一个捕获/比较寄存器（包括一个影子寄存器）、一个捕获输入阶段（数字滤波、多路复用和预分频器，通道 5 和通道 6 除外）和一个输出阶段（比较器和输出控制）构建而成。

图 55. 捕获/比较通道（例如：通道 1 输入阶段），图 58. 捕获/比较通道的输出阶段（通道 4）显示了一个捕获/比较通道的概况。

输入阶段对相应的 TIx 输入进行采样，生成一个滤波后的信号 TIxF。然后，带有极性选择功能的边沿检测器生成一个信号（TIxFPx），该信号可用作从模式控制器的触发输入，也可用作捕获指令。该信号先进行预分频（ICxPS），而后再进入捕获寄存器。

图 55. 捕获/比较通道（例如：通道 1 输入阶段）



输出阶段生成一个中间波形作为基准：OCxRef（高电平有效）。链的末端决定最终输出信号的极性。

图 56. 捕获/比较通道 1 主电路

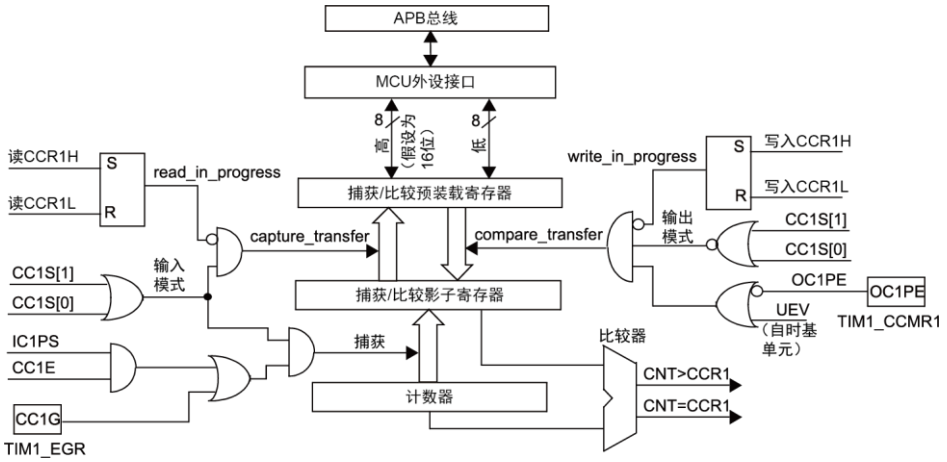
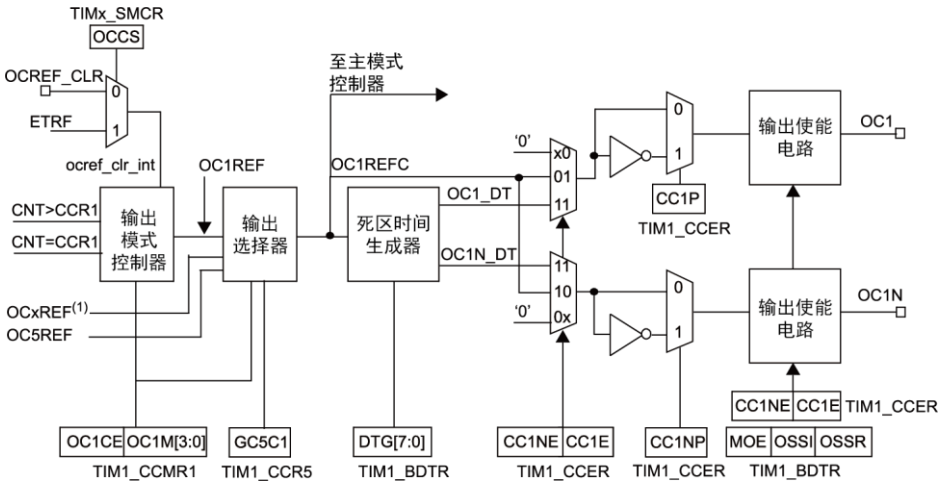


图 57. 捕获/比较通道的输出阶段（通道 1，通道 2 和 3 同前）



注意：　OCxREF，其中 x 为互补通道的序号。

图 58. 捕获/比较通道的输出阶段（通道 4）

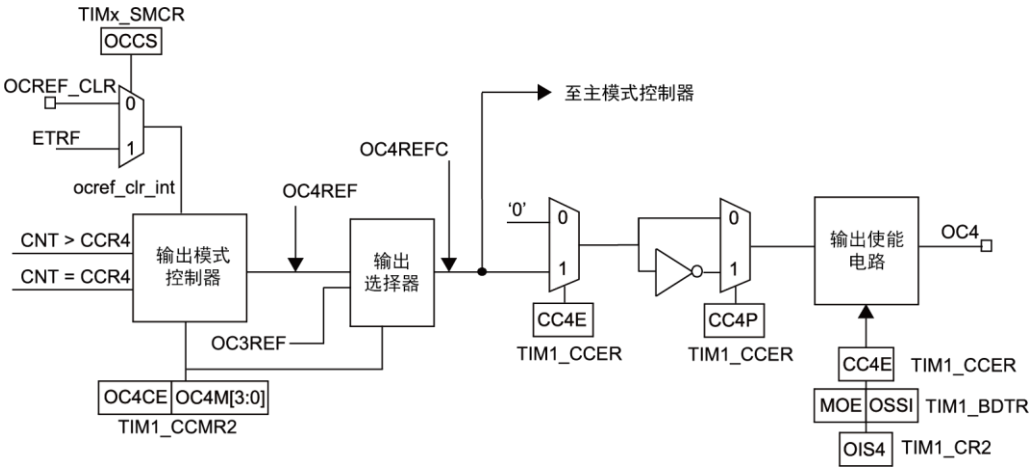
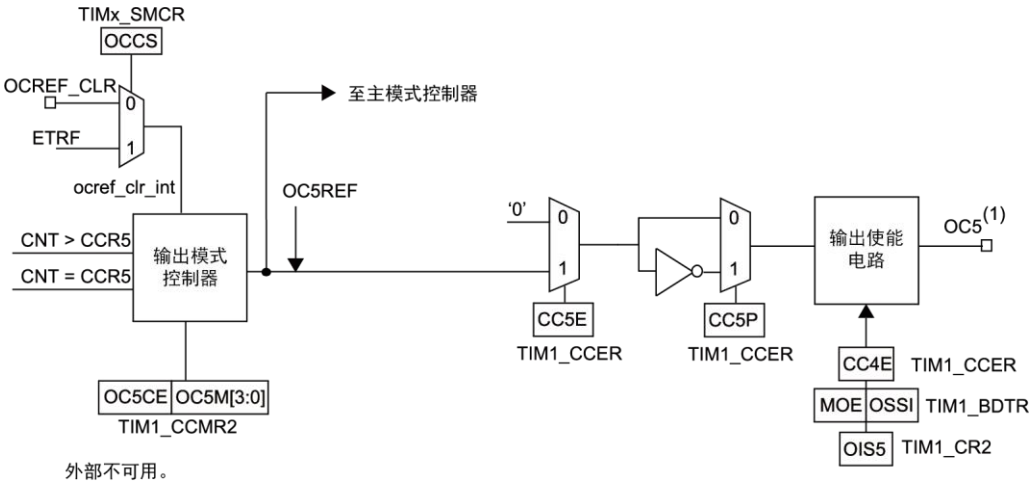


图 59. 捕获/比较通道的输出阶段（通道 5，通道 6 同前）



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。始终可通过读写操作访问预装载寄存器。

在捕获模式下，捕获实际发生在影子寄存器中，然后将影子寄存器的内容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将复制到影子寄存器中，然后将影子寄存器的内容与计数器进行比较。

16.3.7 输入捕获模式

在输入捕获模式下，当相应的 ICx 信号检测到跳变沿后，将使用捕获/比较寄存器（TIMx_CCRx）来锁存计数器的值。当发生捕获时，相应的 CCXIF 标志（TIMx_SR 寄存器）置位并可能发送中断（如使能）。如果在 CCxIF 标志已是高电平时发生捕获，则重复捕获标志 CCxOF（TIMx_SR 寄存器）置位。可通过软件将 CCxIF 清零，方法是：向 CCxIF 写入“0”，或读取存储在 TIMx_CCRx 寄存器中的已捕获数据。向 CCxOF 写入“0”后会将其清零。

以下示例说明了如何在 TI1 输入出现上升沿时将计数器的值捕获到 TIMx_CCR1 中。具体操作步骤如下：

- 选择有效输入：TIMx_CCR1 必须连接到 TI1 输入，因此向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01。只要 CC1S 不等于 00，就会将通道配置为输入模式，并且 TIMx_CCR1 寄存器将处于只读状态。

- 根据连接到定时器的信号，对所需的输入滤波带宽进行编程（如果输入为 TIx 之一，则对 TIMx_CCMRx 寄存器中的 ICxF 位进行编程）。假设在切换时，输入信号在至多 5 个内部时钟周期内不稳定。必须设定长于这 5 个时钟周期的滤波带宽。当检测到具有新电平的 8 个连续采样（以 fDTS 频率采样）时，可以确认 TI1 上的转换有效。然后向 TIMx_CCMR1 寄存器中的 IC1F 位写入 0011。
- 通过在 TIMx_CCER 寄存器中将 CC1P 位和 CC1NP 位写入 0，选择 TI1 上的有效转换边沿（本例中为上升沿）。
- 对输入预分频器进行编程。在本例中，我们希望每次有效转换时都执行捕获操作，因此需要禁止预分频器（向 TIMx_CCMR1 寄存器中的 IC1PS 位写入“00”）。
- 通过将 TIMx_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。
- 如果需要，可通过将 TIMx_DIER 寄存器中的 CC1IE 位置 1 来使能相关中断请求。发生输入捕获时：
  - 发生有效跳变沿时，TIMx_CCR1 寄存器会获取计数器的值。
  - 将 CC1IF 标志置 1（中断标志）。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样 CC1OF 捕获溢出标志会被置 1。
  - 根据 CC1IE 位生成中断。

要处理重复捕获，建议在读出捕获溢出标志之前读取数据。这样可避免丢失在读取捕获溢出标志之后与读取数据之前可能出现的重复捕获信息。

注意： *通过软件将 TIMx_EGR 寄存器中的相应 CCxG 位置 1 可生成 IC 中断。*

16.3.8 PWM 输入模式

注意： *只有通道 1 和通道 2 支持此 PWM 输入模式。*

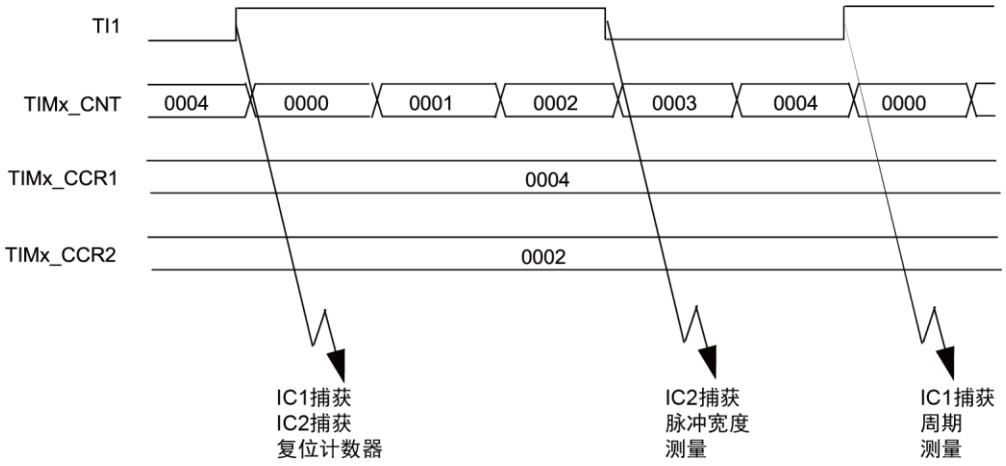
此模式是输入捕获模式的一个特例。其实现步骤与输入捕获模式基本相同，仅存在以下不同之处：

- 两个 ICx 信号被映射至同一个 TIx 输入
- 这两个 ICx 信号在边沿处有效，但极性相反
- 选择两个 TIxFP 信号之一作为触发输入，并将从模式控制器配置为复位模式。

例如，可通过以下步骤对应用于 TI1 的 PWM 的周期（位于 TIMx_CCR1 寄存器中）和占空比（位于 TIMx_CCR2 寄存器中）进行测量（取决于 CK_INT 频率和预分频器的值）：

- 选择 TIMx_CCR1 的有效输入：向 theTIMx_CCMR1 寄存器中的 CC1S 写入 01（选择 TI1）。
- 选择 TI1FP1 的有效极性（用于在 TIMx_CCR1 中捕获和计数器清零）：向 CC1P 位和 CC1NP 位写入“0”（上升沿有效）。
- 选择 TIMx_CCR2 的有效输入：向 theTIMx_CCMR1 寄存器中的 CC2S 写入 10（选择 TI1）。
- 选择 TI1FP2 的有效极性（用于在 TIMx_CCR2 中捕获）：向 CC2P 位和 CC2NP 位写入 CC2P/CC2NP=“10”（下降沿有效）。
- 选择有效触发输入：向 TIMx_SMCR 寄存器中的 TS 位写入 101（选择 TI1FP1）。
- 将从模式控制器配置为复位模式：向 TIMx_SMCR 寄存器中的 SMS 位写入 0100。
- 使能捕获：向 TIMx_CCER 寄存器中的 CC1E 位和 CC2E 位写入“1”。

图 60. PWM 输入模式时序



16.3.9 强制输出模式

在输出模式（TIMx_CCMRx 寄存器中的 CCxS 位 = 00）下，可直接由软件将每个输出比较信号（OCxREF 和 OCx/OCxN）强制设置为有效电平或无效电平，而无需考虑输出比较寄存器和计数器之间的任何比较结果。

要将输出比较信号（OCXREF/OCx）强制设置为有效电平，只需向相应 TIMx_CCMRx 寄存器中的 OCxM 位写入 0101。因此，OCXREF 被强制为高电平（OCxREF 始终为高电平有效）且 OCx 获得 CCxP 极性位的相反值。

例如：CCxP=0（OCx 高电平有效）=> OCx 强制设置为高电平。

通过向 TIMx_CCMRx 寄存器中的 OCxM 位写入 0100，可将 OCxREF 信号强制设置为低电平。

无论如何，TIMx_CCRx 影子寄存器与计数器之间的比较仍会执行，而且允许将标志置 1。可相应地发送中断。下面的输出比较模式一节对此进行了介绍。

16.3.10 输出比较模式

此功能用于控制输出波形，或指示已经过某一时间段。可输出通道 1 至 6。

当捕获/比较寄存器与计数器之间相匹配时，输出比较功能：

- 将为相应的输出引脚分配一个可编程值，该值由输出比较模式（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。匹配时，输出引脚既可保持其电平（OCXM=0000），也可设置为有效电平（OCXM=0001）、无效电平（OCXM=0010）或进行翻转（OCxM=0011）。
- 将中断状态寄存器中的标志置 1（TIMx_SR 寄存器中的 CCxIF 位）。
- 如果相应中断使能位（TIMx_DIER 寄存器中的 CCXIE 位）置 1，将生成中断。

使用 TIMx_CCMRx 寄存器中的 OCxPE 位，可将 TIMx_CCRx 寄存器配置为带或不带预装载寄存器。

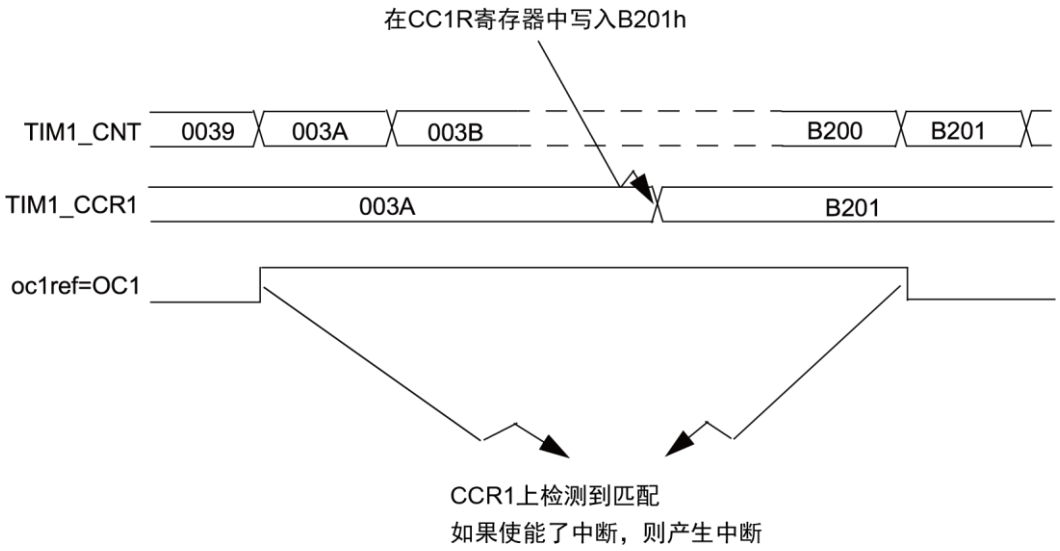
在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出毫无影响。同步的精度可以达到计数器的一个计数周期。输出比较模式也可用于输出单脉冲（在单脉冲模式下）。

步骤：

1. 选择计数器时钟（内部、外部、预分频器）
2. 在 TIMx_ARR 和 TIMx_CCRx 寄存器中写入所需数据。
3. 如果要生成中断请求，则需将 CCxIE 位置 1。
4. 选择输出模式。例如：
   a. 当 CNT 与 CCRx 匹配时，写入 OCxM = 0011 以翻转 OCx 输出引脚
   b. 写入 OCxPE = 0 以禁止预装载寄存器
   c. 写入 CCxP = 0 以选择高电平有效极性
   d. 写入 CCxE = 1 以使能输出
5. 通过将 TIMx_CR1 寄存器中的 CEN 位置 1 来使能计数器。

可随时通过软件更新 TIMx_CCRx 寄存器以控制输出波形，前提是未使能预装载寄存器（OCxPE=0，仅当发生下一个更新事件 UEV 时，才会更新 TIMx_CCRx 影子寄存器）。示例见图 61. 输出比较模式，翻转 OC1

图 61. 输出比较模式，翻转 OC1



16.3.11 PWM 模式

脉冲宽度调制模式可以生成一个信号，该信号频率由 TIMx_ARR 寄存器值决定，其占空比则由 TIMx_CCRx 寄存器值决定。

各通道可以独立选择 PWM 模式（每个 OCx 输出对应一个 PWM），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“0110”（PWM 模式 1）或“0111”（PWM 模式 2）。您必须通过将 TIMx_CCMRx 寄存器中的 OCxPE 位置位来使能相应的预装载寄存器，并通过将 TIMx_CR1 寄存器中的 ARPE 位置位来最终使能自动重载预装载寄存器（在递增计数或中心对齐模式下）。

由于只有在发生更新事件时预装载寄存器才会传送到影子寄存器，因此启动计数器之前，必须通过将 TIMx_EGR 寄存器中的 UG 位置 1 来初始化所有寄存器。

OCx 极性可通过软件来编程（使用 TIMx_CCER 寄存器的 CCxP 位）。可将其编程为高电平有效或低电平有效。通过 CCxE、CCxNE、MOE、OSSI 和 OSSR 位（TIMx_CCER 和 TIMx_BDTR 寄存器）的组合使能 OCx 输出。有关详细信息，请参见 TIMx_CCER 寄存器说明。

在 PWM 模式（1 或 2）下，TIMx_CNT 总是与 TIMx_CCRx 进行比较，以确定是 TIMx_CCRx≤TIMx_CNT 还是 TIMx_CNT≤TIMx_CCRx（取决于计数器计数方向）。

根据 TIMx_CR1 寄存器中的 CMS 位状态，定时器能够产生边沿对齐模式或中心对齐模式的 PWM 信号。

PWM 边沿对齐模式

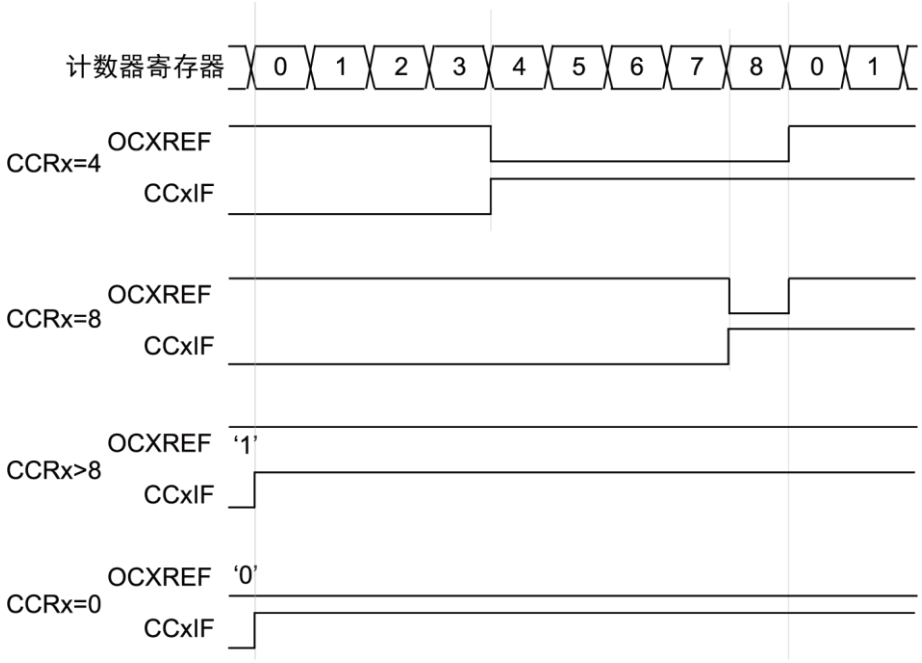
- 递增计数配置

当 TIMx_CR1 寄存器中的 DIR 位为低时递增计数有效。请参见第 16.3.2 节“计数器模式”。

以下以 PWM 模式 1 为例。只要 TIMx_CNT < TIMx_CCRx，PWM 参考信号 OCxREF 便为高电平，否则为低电平。如果 TIMx_CCRx 中的比较值大于自动重载值（TIMx_ARR 中），则 OCxREF 保持为“1”。如果比较值为 0，则 OCxRef 保持为“0”。

图 62. 边沿对齐 PWM 波形（ARR=8）用一个示例展示了一些边沿对齐 PWM 波形，其中的 TIMx_ARR=8。

图 62. 边沿对齐模式的 PWM 波形（ARR=8）



- 递减计数配置

  当 TIMx_CR1 寄存器中的 DIR 位为高时递减计数有效。请参见第 16.3.2 节“计数器模式”。

  在 PWM 模式 1 下，只要 TIMx_CNT > TIMx_CCRx，参考信号 OCxRef 即为低电平，否则其为高电平。如果 TIMx_CCRx 中的比较值大于 TIMx_ARR 中的自动重载值，则 OCxREF 保持为“1”。此模式下不可能产生 0% 的 PWM 波形。

PWM 中心对齐模式

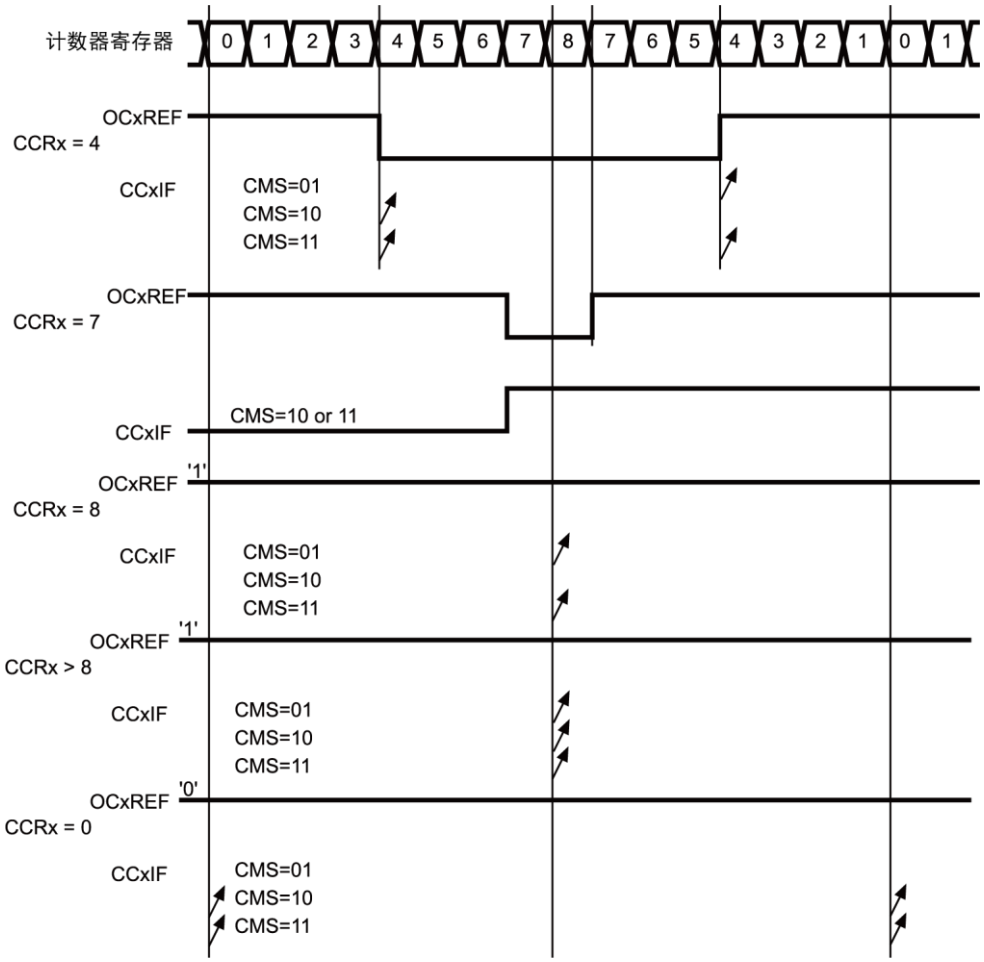
当 TIMx_CR1 寄存器中的 CMS 位不为“00”（其余所有配置对 OCxRef/OCx 信号具有相同的作用），中心对齐模式有效。根据 CMS 位的配置，可以在计数器递增计数、递减计数或同时递增和递减计数时将比较标志置 1。TIMx_CR1 寄存器中的方向位（DIR）由硬件更新，不得通过软件更改。

请参见第 16.3.2 节“计数器模式”。

图 63. 中心对齐 PWM 波形（ARR=8）用一个示例展示了一些中心对齐 PWM 波形，其中：

- TIMx_ARR=8
- PWM 模式为 PWM 模式 1
- 在根据 TIMx_CR1 寄存器中 CMS=01 而选择的中心对齐模式 1 下，当计数器递减计数时，比较标志置 1。

图 63. 中心对齐模式 PWM 波形（ARR=8）



中心对齐模式使用建议

- 启动中心对齐模式时将使用当前的递增/递减计数配置。这意味着计数器将根据写入 TIMx_CR1 寄存器中 DIR 位的值进行递增或递减计数。此外，不得同时通过软件修改 DIR 和 CMS 位。
- 不建议在运行中心对齐模式时对计数器执行写操作，否则将发生意想不到的结果。尤其是：
  1. 如果写入计数器中的值大于自动重载值（TIMx_CNT>TIMx_ARR），计数方向不会更新。例如，如果计数器之前递增计数，则继续递增计数。
  2. 如果向计数器写入 0 或 TIMx_ARR 的值，计数方向会更新，但不生成更新事件 UEV。
- 使用中心对齐模式最为保险的方法是：在启动计数器前通过软件生成更新（将 TIMx_EGR 寄存器中的 UG 位置 1），并且不要在计数器运行过程中对其执行写操作。

16.3.12 不对称 PWM 模式

在不对称模式下，生成的两个中心对齐 PWM 信号间允许存在可编程相移。频率由 TIMx_ARR 寄存器的值确定，而占空比和相移则由一对 TIMx_CCRx 寄存器确定。两个寄存器分别控制递增计数和递减计数期间的 PWM，这样每半个 PWM 周期便会调节一次 PWM：

- OC1REFC（或 OC2REFC）由 TIMx_CCR1 和 TIMx_CCR2 控制
- OC3REFC（或 OC4REFC）由 TIMx_CCR3 和 TIMx_CCR4 控制

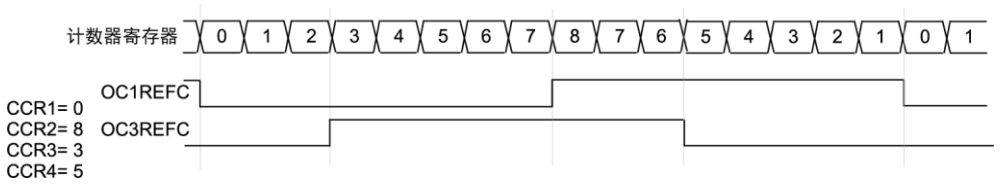
两个通道可以独立选择不对称 PWM 模式（每对 CCR 寄存器一个 OCx 输出），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“1110”（不对称 PWM 模式 1）或“1111”（不对称 PWM 模式 2）。

注意： *出于兼容性原因，OCxM[3:0] 位字段分为两部分，最高有效位与最低有效的 3 位不相邻。*

给定通道用作不对称 PWM 通道时，也可使用其互补通道。例如，如果通道 1 上产生 OC1REFC 信号（不对称 PWM 模式 1），则由于不对称 PWM 模式 1 的原因，通道 2 上可输出 OC2REF 信号或 OC2REFC 信号。

图 64. 50%占空比时产生的 2 个相移 PWM 信号展示了可使用不对称 PWM 模式生成的信号的一个示例（在不对称 PWM 模式 1 下配置通道 1 至 4）。与死区发生器配合使用时，这可控制相移全桥直流到直流转换器。

图 64. 50%占空比时产生的 2 个相移 PWM 信号



16.3.13 组合 PWM 模式

在组合 PWM 模式下，生成的两个边沿或中心对齐 PWM 信号的各个脉冲间允许存在可编程延时和相移。频率由 TIMx_ARR 寄存器的值确定，而占空比和延时则由两个 TIMx_CCRx 寄存器确定。产生的信号 OCxREFC 由两个参考 PWM 的逻辑或运算或者逻辑与运算组合组成：

- OC1REFC（或 OC2REFC）由 TIMx_CCR1 和 TIMx_CCR2 控制
- OC3REFC（或 OC4REFC）由 TIMx_CCR3 和 TIMx_CCR4 控制

两个通道可以独立选择组合 PWM 模式（每对 CCR 寄存器一个 OCx 输出），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“1100”（组合 PWM 模式 1）或“1101”（组合 PWM 模式 2）。

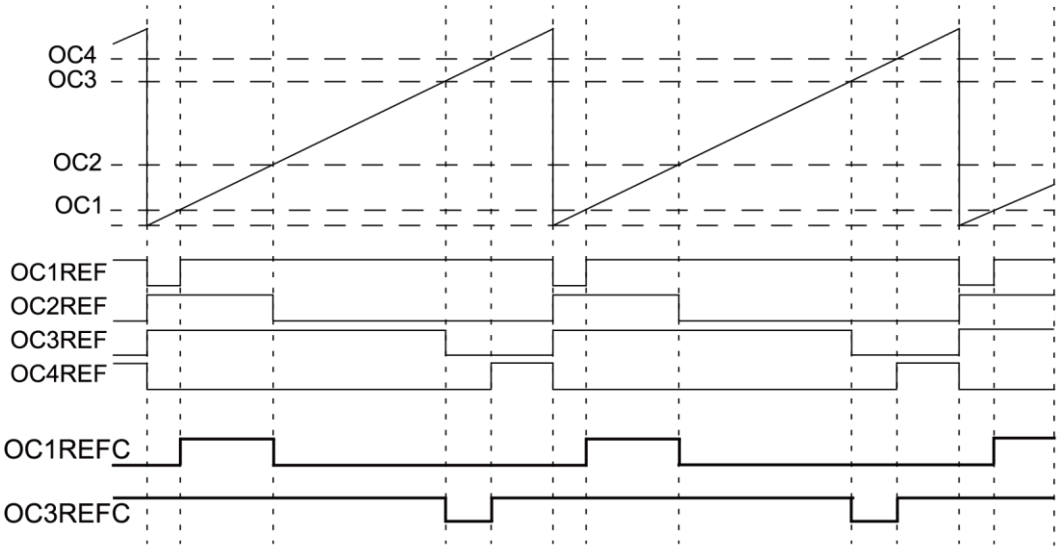
当给定通道用作组合 PWM 通道时，其互补通道必须在相反的 PWM 模式下配置（例如，一个通道在组合 PWM 模式 1 下配置，另一个通道在组合 PWM 模式 2 下配置）。

注意： *出于兼容性原因，OCxM[3:0]位字段分为两部分，最高有效位与最低有效的 3 位不相邻。*

图 65. 通道 1 和通道 3 上的组合 PWM 模式展示了可使用不对称 PWM 模式生成的信号的一个示例，使用以下配置获得：

- 通道 1 在组合 PWM 模式 2 下配置
- 通道 2 在 PWM 模式 1 下配置
- 通道 3 在组合 PWM 模式 2 下配置
- 通道 4 在 PWM 模式 1 下配置。

图 65. 通道 1 和通道 3 上的组合 PWM 模式



OC1REFC = OC1REF和OC2REF
OC3REFC = OC3REF或OC4REF

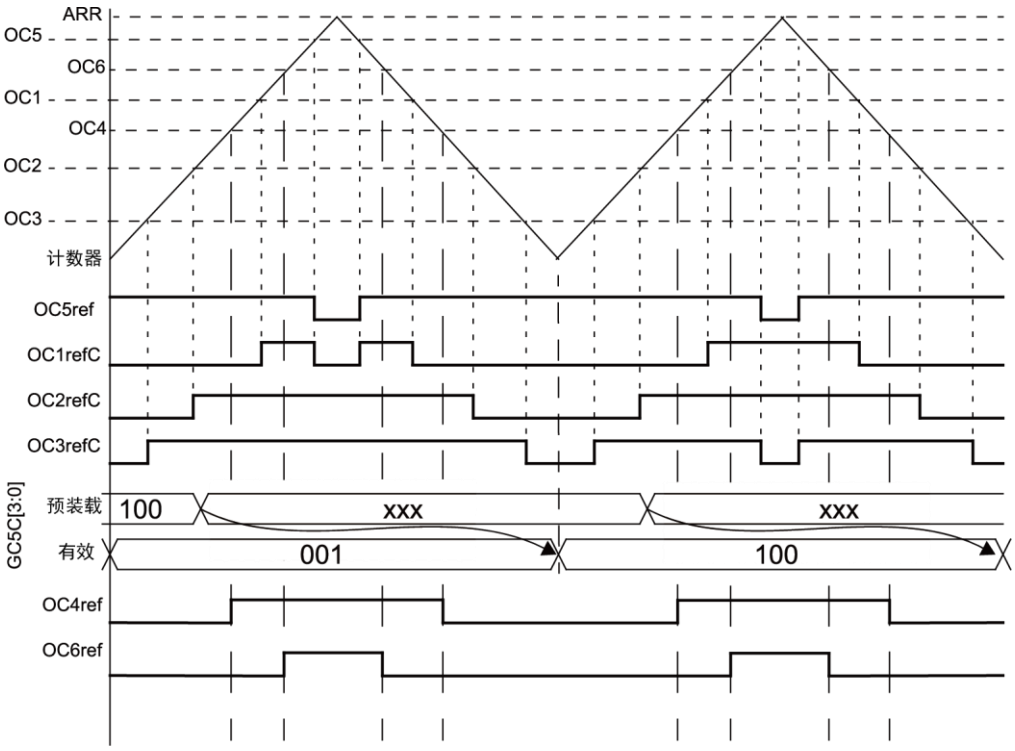
16.3.14 组合三相 PWM 模式

在组合三相 PWM 模式下，产生的一至三个中心对齐 PWM 信号与一个可编程信号间允许在脉冲中间进行逻辑与运算。OC5REF 信号用于定义产生的组合信号。凭借 TIMx_CCR5 中的 3 位 GC5C[3:1]，可以选择 OC5REF 与哪个参考信号组合。产生的信号 OCxREFC 由两个参考 PWM 的逻辑与运算组合组成：

- 如果 GC5C1 置 1，则 OC1REFC 由 TIMx_CCR1 和 TIMx_CCR5 控制
- 如果 GC5C2 置 1，则 OC2REFC 由 TIMx_CCR2 和 TIMx_CCR5 控制
- 如果 GC5C3 置 1，则 OC3REFC 由 TIMx_CCR3 和 TIMx_CCR5 控制

通道 1 到通道 3 可独立选择组合三相 PWM 模式，只需将 3 位 GC5C[3:1] 中的至少一位置 1。

图 66. 三相组合 PWM 信号（每个周期多个触发脉冲）



16.3.15 互补输出和死区时间插入

高级控制定时器（TIM1）可以输出两路互补信号，并管理输出的关断与接通瞬间。

注意： *此特性只与通道 1 至 4 有关。*

这段时间通常称为死区，用户必须根据与输出相连接的器件及其特性（电平转换器的固有延迟、功率开关器件产生的延迟…）来调整死区时间

每路输出可以独立选择输出极性（主输出 OCx 或互补输出 OCxN）。可通过对 TIMx_CCER 寄存器中的 CCxP 和 CCxNP 位执行写操作来完成极性选择。

互补信号 OCx 和 OCxN 通过以下多个控制位的组合进行激活：TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位以及 TIMx_BDTR 和 TIMx_CR2 寄存器中的 MOE、OISx、OISxN、OSSI 和 OSSR 位。

请参见“表 50. 具有断路功能的互补通道 OCx 和 OCxN 的输出控制位”，了解更多信息。特别是在切换至空闲状态时激活死区时间（MOE 降至 0）。

通过将 CCxE 和 CCxNE 位以及 MOE 位（如果存在断路）置位使能死区时间插入。每个通道有一个 10 位死区时间发生器。将基于参考波形 OCxREF 生成 2 个输出 OCx 和 OCxN。如果 OCx 和 OCxN 为高电平有效：

- 输出信号 OCx 与参考信号相同，只是其上升沿相对参考上升沿存在延迟
- 输出信号 OCxN 与参考信号相反，并且其上升沿相对参考下降沿存在延迟

如果延迟时间大于有效输出（OCx 或 OCxN）的宽度，则不会产生相应的脉冲。

下图所示为死区发生器的输出信号与参考信号 OCxREF 之间的关系。（在这些示例中，假定 CCxP=0、CCxNP=0、MOE=1、CCxE=1 并且 CCxNE=1）。

图 67. 插入了死区时间的互补输出

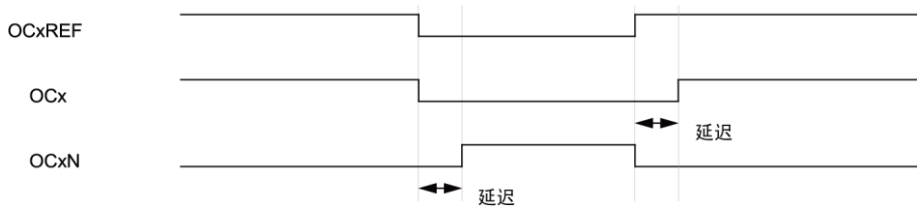


图 68. 延时大于负脉冲的死区时间波形

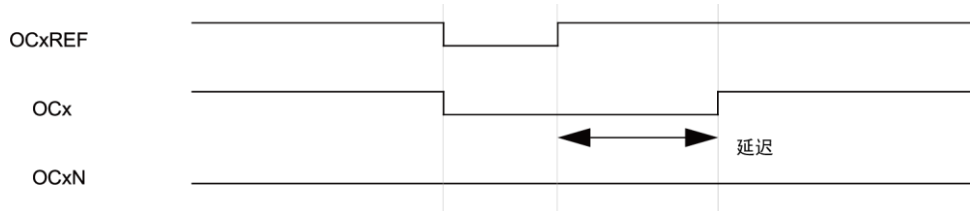
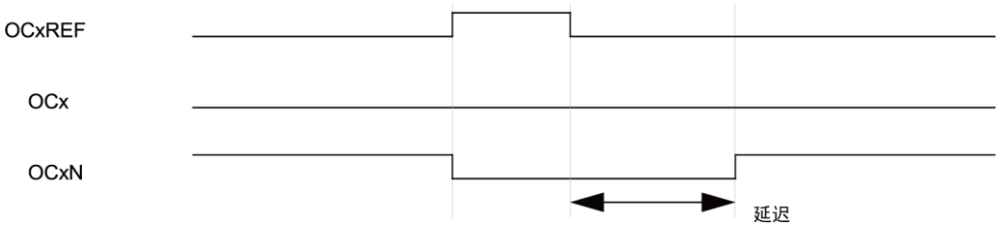


图 69. 延时大于正脉冲的死区时间波形



每个通道的死区时间延时相同，并且可通过 TIMx_BDTR 寄存器中的 DTG 位进行编程。请参见第 16.4.18 节“TIM1 断路和死区时间寄存器（TIMx_BDTR）”了解延时计算。

将 OCxREF 重定向到 OCx 或 OCxN

在输出模式（强制输出模式、输出比较模式或 PWM 模式）下，通过配置 TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位，可将 OCxREF 重定向到 OCx 输出或 OCxN 输出。

通过此功能，可以在一个输出上发送特定波形（如 PWM 或静态有效电平），而同时使互补输出保持其无效电平。其他可供选择的可能性是两个输出均为无效电平或两个输出均有效并补偿以死区时间。

注意： *如果仅使能 OCxN（CCxE=0, CCxNE=1），两者不互补，一旦 OCxREF 为高电平，OCxN 即变为有效。例如，如果 CCxNP=0，则 OCxN=OCxRef。另一方面，如果同时使能 OCx 和 OCxN（CCxE=CCxNE=1），OCx 在 OCxREF 为高电平时变为有效，而 OCxN 则与之互补，在 OCxREF 为低电平时变为有效。*

16.3.16 使用断路功能

断路功能的目的是保护由 TIM1 定时器生成的 PWM 信号所驱动的功率开关器件。断路输入通常连接到功率级和三相逆变器的故障输出。激活时，断路电路会关闭 PWM 输出，并将其强制为预定义的安全状态。

断路有两个通道，可从输入引脚收集应用程序故障。break2 通道能够将输出强制变为无效状态。断路期间的输出使能信号和输出电平取决于多个控制位：

- TIMx_BDTR 寄存器中的 MOE 位用于通过软件使能/禁用输出，它会在发生 break 或 break2 事件时复位。

- TIMx_BDTR 寄存器中的 OSSI 位，定义定时器将输出控制在无效状态下，还是释放对 GPIO 控制器的控制（通常使其处于高阻态模式）
- TIMx_CR2 寄存器中的 OISx 和 OISxN 位，设置输出关闭电平（有效或无效）。无论 OISx 和 OISxN 的值为何，均无法在给定时间将 OCx 和 OCxN 输出同时设置为有效电平。请参见“表 50. 具有断路功能的互补通道 OCx 和 OCxN 的输出控制位”，了解更多信息。

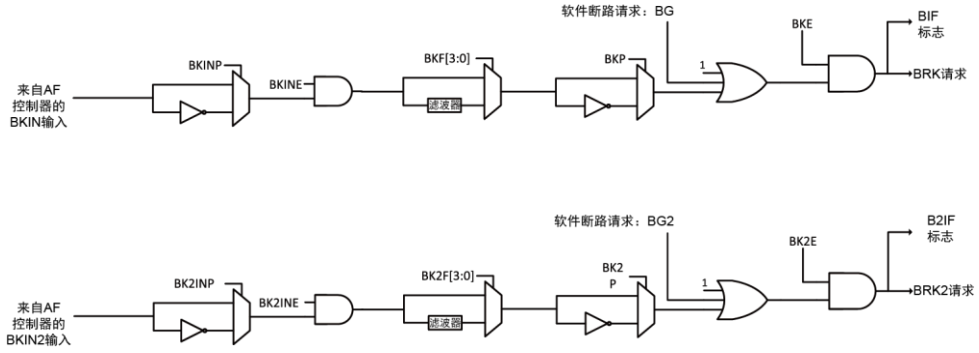
退出复位状态后，断路电路处于禁止状态，MOE 位处于低电平。将 TIMx_BDTR 寄存器中的 BKE 和 BKE2 位置 1 可全局使能断路功能。通过配置 TIMx_BDTR 寄存器中的 BKP 和 BKP2 位可选择断路输入全局极性。BKEx 和 BKPx 位可同时修改。对 BKEx 和 BKPx 位执行写操作时，写操作会在 1 个 APB 时钟周期的延迟后生效。因此，执行写操作后，需要等待 1 个 APB 时钟周期，才能准确回读该位。

注意： *由于 MOE 下降沿可能是异步信号，因此在实际信号（作用于输出）与同步控制位（位于 TIMx_BDTR 寄存器中）之间插入了再同步电路，从而在异步信号与同步信号之间产生延迟。具体而言，如果在 MOE 处于低电平时向其写入 1，则必须首先插入延迟（空指令），才能准确进行读取。这是因为写入的是异步信号，而读取的却是同步信号。*

可从多个源生成断路，这些源可单独使能并具有可编程的边沿敏感性（使用 TIMx_AF1 和 TIMx_AF2 寄存器）。

也可由软件通过 TIMx_EGR 寄存器中的 BG 和 B2G 位产生断路事件。无论 BKE 和 BKE2 使能位的值是多少，使用 BG 和 BG2 的软件断路生成均为有效状态。

图 70. Break 和 break2 电路系统总览



注意

只有禁止可编程滤波器时才能保证异步（无时钟）操作。如使能，必须使用失效保护时钟模式保证断路事件得到处理。

发生断路之一（其中一个断路输入上出现所选电平）时：

- MOE 位异步清零，使输出处于无效状态、空闲状态甚至释放对 GPIO 控制器的控制（通过 OSSI 位进行选择）。即使 MCU 振荡器关闭，该功能仍然有效
- MOE=0 时，将以 TIMx_CR2 寄存器 OISx 位中编程的电平驱动每个输出通道。如果 OSSI=0，定时器将释放输出控制（由 GPIO 控制器接管），否则使能输出保持高电平

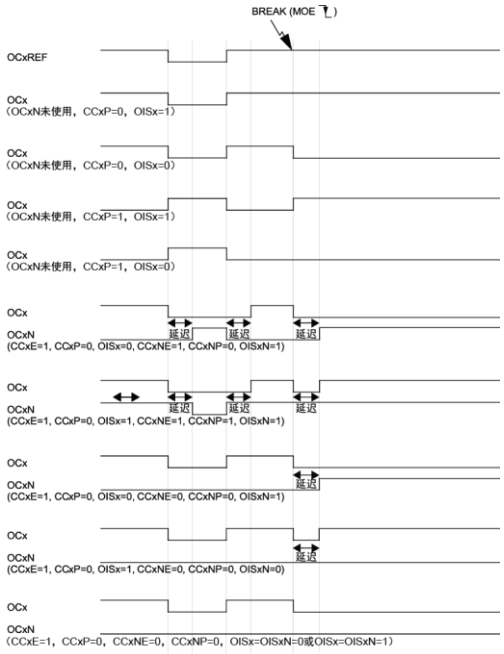
- 使用互补输出时：
  1. 输出首先被置于无效状态（取决于极性）。这是异步操作，因此即使没有为定时器提供时钟，该操作仍有效
  2. 如果定时器时钟仍存在，则将重新激活死区发生器，进而在死区后以 OISx 和 OISxN 位中编程的电平驱动输出。即使在这种情况下，也不能同时将 OCx 和 OCxN 驱动至其有效电平。请注意，MOE 进行再同步，因此死区的持续时间会比通常情况长一些（约 2 个 ck_tim 时钟周期）
  3. 如果 OSSI=0，定时器将释放输出控制（由强制高阻态的 GPIO 控制器接管），否则使能输出将保持高电平或在 CCxE 或 CCxNE 位之一为高电平时立即变为高电平
- 断路状态标志（TIMx_SR 寄存器中的 BIF 和 B2IF 位）置位。如果 TIMx_DIER 寄存器中的 BIE 位置 1，可产生中断
- 如果 TIMx_BDTR 寄存器中的 AOE 位置 1，则 MOE 位会在发生下一更新事件（UEV）时自动再次置 1。这一特性有许多用处，比如，可用于实现调节器的功能。否则，MOE 保持低电平，直至用户再次将其写为“1”。这种情况下，这一特性可用于确保安全。可以将断路输入连接到功率驱动器的警报、温度传感器或任何安全元件。

注意： *断路输入为电平有效。因此，当断路输入有效电平时，不能将 MOE 位置 1（自动或通过软件）。同时，不能将状态标志 BIF 和 B2IF 清零。*

除断路输入和输出管理外，断路电路内部还实施了写保护，用以保护应用的安全。通过该功能，用户可冻结多个参数配置（死区持续时间、OCx/OCxN 极性和禁止时的状态、OCxM 配置、断路使能和极性）。可以通过 TIMx_BDTR 寄存器中的 LOCK 位，从 3 种保护级别中进行选择。请参见第 16.4.18 节“TIM1 断路和死区时间寄存器（TIMx_BDTR）”。MCU 复位后只能对 LOCK 位执行一次写操作。

图 71. 响应 BRK 上的断路事件的不同输出行为（OSSI = 1）显示了响应断路的输出行为示例。

图 71. 响应 BRK 上的断路事件的不同输出行为（OSSI = 1）



两个断路输入针对定时器输出具有不同的行为：

- BRKIN 输入可禁止（无效状态）PWM 输出，也可将 PWM 输出强制为预定义的安全状态
- BRKIN2 只能禁止（无效状态）PWM 输出。

BRKIN 的优先级高于 BRKIN2 输入，如“表 48. 不同 BRK/BK2 输入的定时器输出行为”所示。

注意： *BRKIN2 必须只在 OSSR = OSSI = 1 时使用。*

表 48. 不同 BRK/BK2 输入的定时器输出行为

BRKIN	BRKIN2	定时器输出状态	典型用例	
			OCxN 输出（低侧开关）	OCx 输出（高侧开关）
有效	X	• 无效，之后强制为输出状态（死区后） • 如果 OSSI = 0，则禁止输出（由 GPIO 逻辑接管控制）	死区插入后开启	OFF
无效	有效	无效	OFF	OFF

图 72. 跟随 BRK 和 BRK2 引脚断言的 PWM 输出状态（OSSI=1）给出了 BRK 和 BRK2 输入上有有源信号时 OCx 和 OCxN 输出行为的示例。在这种情况下，两个输出的极性均为高电平有效（TIMx_CCER 寄存器中的 CCxP = CCxNP = 0）。

图 72. 跟随 BRK 和 BRK2 引脚断言的 PWM 输出状态（OSSI=1）

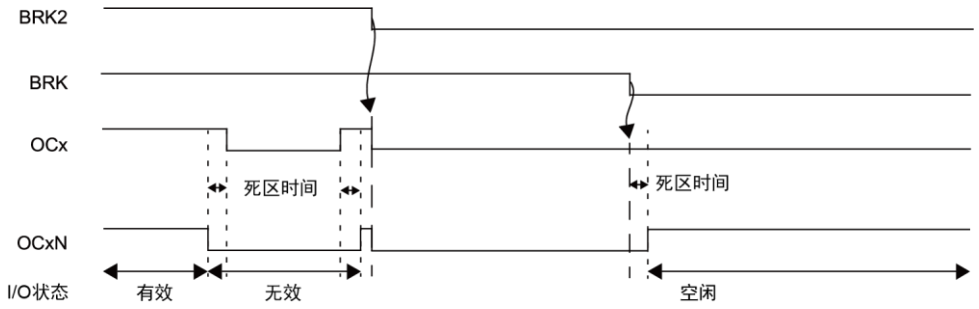
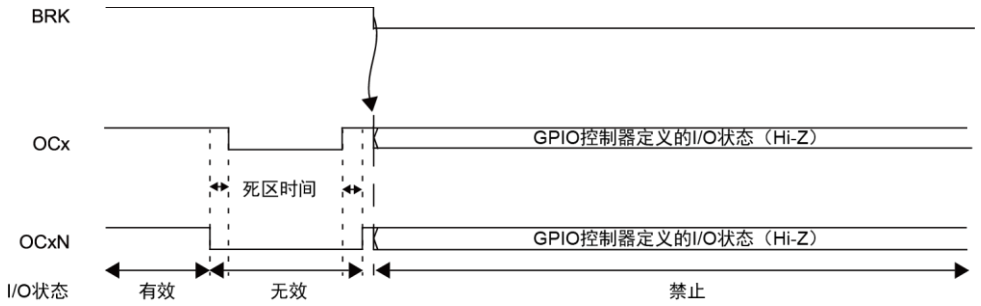


图 73. 跟随 BRK 断言的 PWM 输出状态（OSSI=0）



16.3.17 发生外部事件时清除 OCxREF 信号

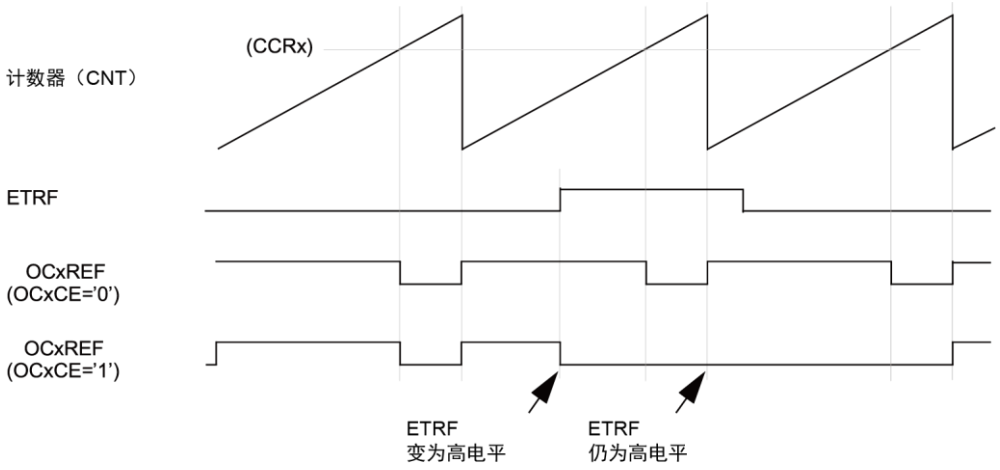
如果 TIMx_SMCR.OCCS 位置为 1，在对 ETRF 输入施加高电平时，可清除给定通道的 OCxREF 信号（相应 TIMx_CCMRx 寄存器中的 OCxCE 使能位置为 1）。

该功能只能在输出比较模式和 PWM 模式下使用。在强制模式下不起作用。

1. 必须关闭外部触发预分频器：TIMx_SMCR 寄存器中的 ETPS[1:0] 位置“00”。
2. 必须禁止外部时钟模式 2：TIMx_SMCR 寄存器中的 ECE 位置“0”。
3. 外部触发极性（ETP）和外部触发滤波器（ETF）可根据用户需要进行配置。

图 74. 清除 TIMx 的 OCxREF 显示了对于使能位 OCxCE 的两个值而言，当 ETRF 输入变为高电平时 OCxREF 信号的行为。在本例中，定时器 TIMx 编程为 PWM 模式。

图 74. 清除 TIMx 的 OCxREF



注意： *如果 PWM 的占空比为 100%（CCRx>ARR），则下次计数器上溢时会再次使能 OCxREF。*

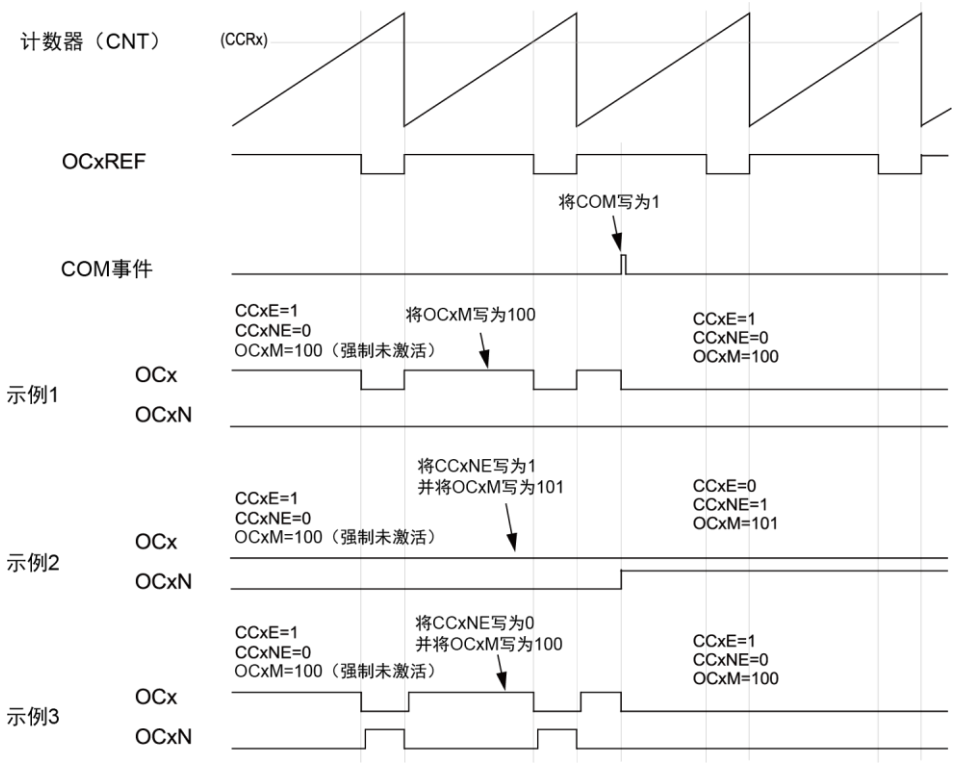
16.3.17.1 生成 6 步 PWM

当通道使用互补输出时，预装载位在 OCxM、CCxE 和 CCxNE 位上可用。发生 COM 换向事件时，这些预装载位将传输到影子位。因此，用户可以提前设定下一步的配置，以及同时更改所有通道的配置。COM 可由软件通过将 TIMx_EGR 寄存器中的 COM 位置 1 而生成，也可以由硬件在 TRGI 上升沿生成。

当发生 COM 事件时，标志置位（TIMx_SR 寄存器中的 COMIF 位），可生成中断（如果 TIMx_DIER 寄存器中的 COMIE 位置 1）。

图 75. COM 事件生成 6 步 PWM 的示例（OSSR=1）描述了在设定配置的 3 个不同示例中，发生 COM 事件时 OCx 和 OCxN 输出的行为。

图 75. COM 事件生成 6 步 PWM 的示例（OSSR=1）



16.3.18 单脉冲模式

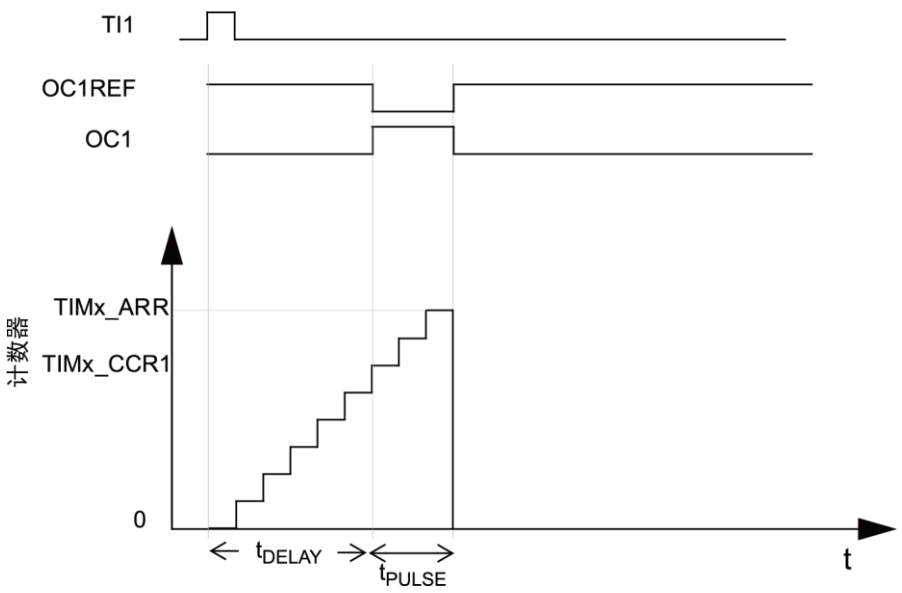
单脉冲模式（OPM）是上述模式的一个特例。在这种模式下，计数器可以在一个激励信号的触发下启动，并可在一段可编程的延时后产生一个脉宽可编程的脉冲。

可以通过从模式控制器启动计数器。可在输出比较模式或 PWM 模式下生成波形。将 TIMx_CR1 寄存器中的 OPM 位置 1，即可选择单脉冲模式。这样，发生下一更新事件 UEV 时，计数器将自动停止。

只有当比较值与计数器初始值不同时，才能正确产生一个脉冲。启动前（定时器等待触发时），必须进行如下配置：

- 在递增计数时：CNT < CCRx <= ARR（特别是，0 <CCRx）
- 在递减计数时：CNT >CCRx

图 76. 单脉冲模式示例



例如，用户希望达到这样的效果：在 TI2 输入引脚检测到上升沿时，经过 tDELAY 的延迟，在 OC1 上产生一个长度为 tPULSE 的正脉冲。

我们使用 TI2FP2 作为触发信号 1：

- 在 TIMx_CCMR1 寄存器中写入 CC2S=“01”，以将 TI2FP2 映射到 TI2
- TI2FP2 必须检测上升沿，在 TIMx_CCER 寄存器中写入 CC2P=“0”和 CC2NP=“0”
- 在 TIMx_SMCR 寄存器中写入 TS=110，将 TI2FP2 配置为从模式控制器的触发（TRGI）
- 在 TIMx_SMCR 寄存器中写入 SMS=“110”（触发模式），以使用 TI2FP2 启动计数器。

OPM 波形通过对比较寄存器执行写操作来定义（考虑时钟频率和计数器预分频器）。

- t_{DELAY} 由在 TIMx_CCR1 寄存器中写入的值来定义。
- t_{PULSE} 由自动重载值与比较值之差（TIMx_ARR - TIMx_CCR1）来定义
- 假设希望产生这样的波形：信号在发生比较匹配时从“0”变为“1”，在计数器达到自动重载值时由“1”变为“0”。为此，应在 TIMx_CCMR1 寄存器中写入 OC1M=111，以使能 PWM 模式 2。如果需要，可选择在 TIMx_CCMR1 寄存器的 OC1PE 和 TIMx_CR1 寄存器的 ARPE 中写入“1”，以使能预装载寄存器。这种情况下，必须在 TIMx_CCR1 寄存器中写入比较值并在 TIMx_ARR 寄存器中写入自动重载值，通过将 UG 位置 1 来产生更新，然后等待 TI2 上的外部触发事件。本例中，CC1P 的值被写为“0”。

在本例中，TIMx_CR1 寄存器中的 DIR 和 CMS 位应为低。

由于仅需要 1 个脉冲（单脉冲模式），因此应向 TIMx_CR1 寄存器的 OPM 位写入“1”，以便在发生下一更新事件（计数器从自动重载值返回到 0）时使计数器停止计数。TIMx_CR1 寄存器中的 OPM 位置“0”时，即选择重复模式。

特例：OCx 快速使能：

在单脉冲模式下，TIx 输入的边沿检测会将 CEN 位置 1，表示使能计数器。然后，在计数器值与比较值之间发生比较时，将切换输出。这些操作需要数个时钟周期。最短延时 tDELAY 是我们能够获得的最小值。

如果要输出延迟时间最短的波形，可以将 TIMx_CCMRx 寄存器中的 OCxFE 位置 1。这样会强制 OCxRef（和 OCx）对激励信号做出响应，而不再考虑比较的结果。其新电平与发生比较匹配时相同。仅当通道配置为 PWM1 或 PWM2 模式时，OCxFE 才会起作用。

16.3.19 可再触发单脉冲模式（OPM）

在这种模式下，计数器可以在一个激励信号的触发下启动，并且可以生成一个长度可编程的脉冲，但与第16.3.18 节“单脉冲模式”中描述的不可再触发单脉冲模式相比存在以下差异：

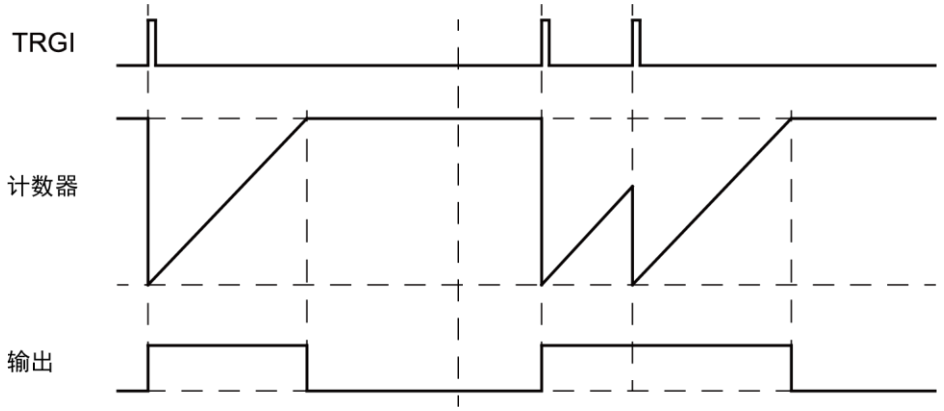
- 发生触发时，脉冲立即产生（无可编程延时）
- 如果在上一个触发完成前发生新的触发，脉冲将延长。

定时器必须处于从模式，TIMx_SMCR 寄存器中的位 SMS[3:0] = “1000”（组合复位 + 触发模式），针对可再触发 OPM 模式 1 或模式 2 将 OCxM[3:0] 位设置为“1000”或“1001”。

定时器配置为递增计数模式时，相应的 CCRx 必须置 0（ARR 寄存器设置脉冲长度）。定时器配置为递减计数模式时，ARR 必须置 0（CCRx 寄存器设置脉冲长度）。

注意： *出于兼容性原因，OCxM[3:0]和 SMS[3:0]位字段分为两部分，最高有效位与最低有效的 3 位不相邻。在可再触发单脉冲模式下，CCxIF 标志没有意义。*

图 77. 可再触发单脉冲模式



16.3.20 编码器接口模式

选择编码器接口模式时，如果计数器仅在 TI2 边沿处计数，在 TIMx_SMCR 寄存器中写入 SMS=“001”；如果计数器仅在 TI1 边沿处计数，写入 SMS=“010”；如果计数器在 TI1 和 TI2 边沿处均计数，则写入 SMS=“011”。

通过编程 TIMx_CCER 寄存器的 CC1P 和 CC2P 位，选择 TI1 和 TI2 极性。如果需要，还可对输入滤波器进行编程。CC1NP 和 CC2NP 必须保持低电平。

TI1 和 TI2 两个输入用于连接正交编码器。请参见“表 49. 不同编码器信号的计数方向”。

如果使能计数器（在 TIMx_CR1 寄存器的 CEN 位中写入“1”），则计数器的时钟由 TI1FP1 或 TI2FP2 上的每次有效信号转换提供。TI1FP1 和 TI2FP2 是进行输入滤波器和极性选择后 TI1 和 TI2 的信号，如果不进行滤波和反相，则 TI1FP1=TI1，TI2FP2=TI2。将根据两个输入的信号转换序列，产生计数脉冲和方向信号。根据该信号转换序列，计数器相应递增或递减计数，同时硬件对 TIMx_CR1 寄存器的 DIR 位进行相应修改。任何输入（TI1 或 TI2）发生信号转换时，都会计算 DIR 位，无论计数器是仅在 TI1 或 TI2 边沿处计数，还是同时在 TI1 和 TI2 处计数。

编码器接口模式就相当于带有方向选择的外部时钟。这意味着，计数器仅在 0 到 TIMx_ARR 寄存器中的自动重载值之间进行连续计数（根据具体方向，从 0 递增计数到 ARR，或从 ARR 递减计数到 0）。因此，在启动前必须先配置 TIMx_ARR。同样，捕获、比较、预分频器、重复计数器和触发输出功能继续正常工作。编码器模式和外部时钟模式 2 不兼容，因此不能同时选择。

注意： *当编码器模式使能时，必须将预分频器置 0。*

在此模式下，计数器会根据正交编码器的速度和方向自动进行修改，因此，其内容始终表示编码器的位置。计数方向对应于所连传感器的旋转方向。下表总结了可能的组合，假设 TI1 和 TI2 不同时切换。

表 49. 不同编码器信号的计数方向

有效边沿	相反信号的电平（TI1FP1 对应 TI2，TI2FP2 对应 TI1）	TI1FP1 信号		TI2FP2 信号	
		上升	下降	上升	下降
仅在 TI1 处计数	高	递减	递增	不计数	不计数
	低	递增	递减	不计数	不计数
仅在 TI2 处计数	高	不计数	不计数	递增	递减
	低	不计数	不计数	递减	递增
在 TI1 和 TI2 处均计数	高	递减	递增	递增	递减
	低	递增	递减	递减	递增

正交编码器可直接与 MCU 相连，无需外部接口逻辑。不过，通常使用比较器将编码器的差分输出转换为数字信号。这样大幅提高了抗噪声性能。用于指示机械零位的第三个编码器输出可与外部中断输入相连，用以触发计数器复位。

图 78. 编码器接口模式下的计数器工作示例提供了计数器的工作示例，展示了计数信号生成和方向控制。同时也说明了选择双边沿时如何对输入抖动进行补偿。将传感器靠近其中一个切换点放置时可能出现这种情况。本例中假设配置如下：

- CC1S=“01”（TIMx_CCMR1 寄存器，TI1FP1 映射到 TI1 上）
- CC2S=01（TIMx_CCMR2 寄存器，TI1FP2 映射到 TI2 上）
- CC1P=“0”，CC1NP=“0”（TIMx_CCER 寄存器，TI1FP1 未反相，TI1FP1=TI1）
- CC2P=“0”，CC2NP=“0”（TIMx_CCER 寄存器，TI1FP2 未反相，TI1FP2= TI2）
- SMS=011（TIMx_SMCR 寄存器，两个输入在上升沿和下降沿均有效）
- CEN=“1”（TIMx_CR1 寄存器，使能计数器）。

图 78. 编码器接口模式下的计数器工作示例

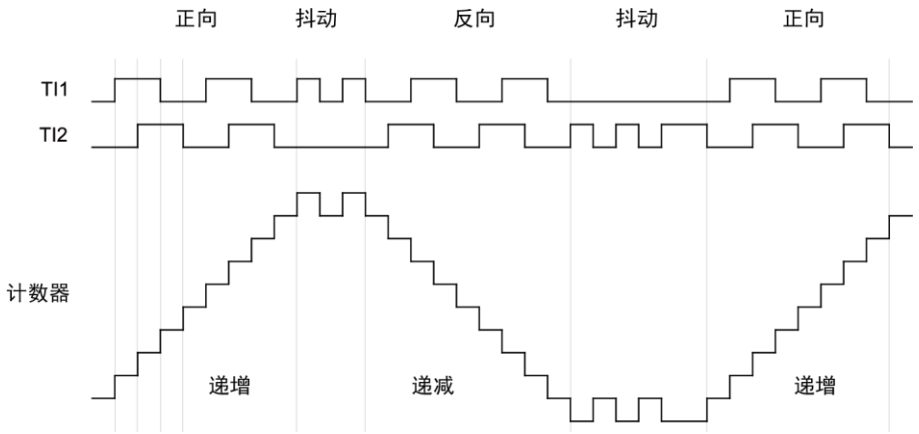
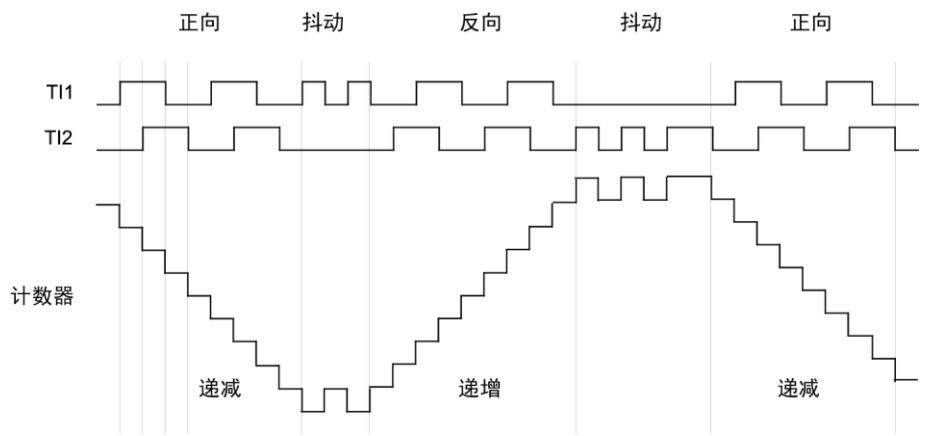


图 79. TI1FP1 极性反相时的编码器接口模式示例提供了当 TI1FP1 极性反相时计数器行为的示例（配置同上，只是 CC1P=“1”）。

图 79. TI1FP1 极性反相时的编码器接口模式示例



当配置为编码器接口模式时，定时器提供一些关于传感器当前位置的信息。通过使用配置为捕获模式的第二个定时器测量两个编码器事件之间的周期（在 BlueNRG-LP 器件中不可用），您可以获得动态信息（速度、加速度和减速度）。指示机械零位的编码器输出即可用于此目的。根据两个事件之间的时间间隔，还可定期读取计数器。如果可能，可以将计数器值锁存到第三个输入捕获寄存器来实现此目的（捕获信号必须为周期性信号，可以由另一个定时器产生）。

TIMx_CR1 寄存器中的 IUFREMAP 位强制将更新中断标志（UIF）连续复制到定时计数器寄存器的位 31（TIMxCNT[31]）中。这样便可按原子方式读取计数器值以及由 UIFCPY 标志发出的潜在的电位翻转条件。这可避免在后台任务（计数器读）和中断（更新中断）之间共享处理时产生竞争条件，从而简化角速度的计算。

UIF 和 UIFCPY 标志使能之间没有延迟。

在 32 位定时器实现中，当 IUFREMAP 位置 1 时，计数器的位 31 在读访问时由 UIFCPY 标志覆盖（计数器的最高有效位只能在写模式下访问）。

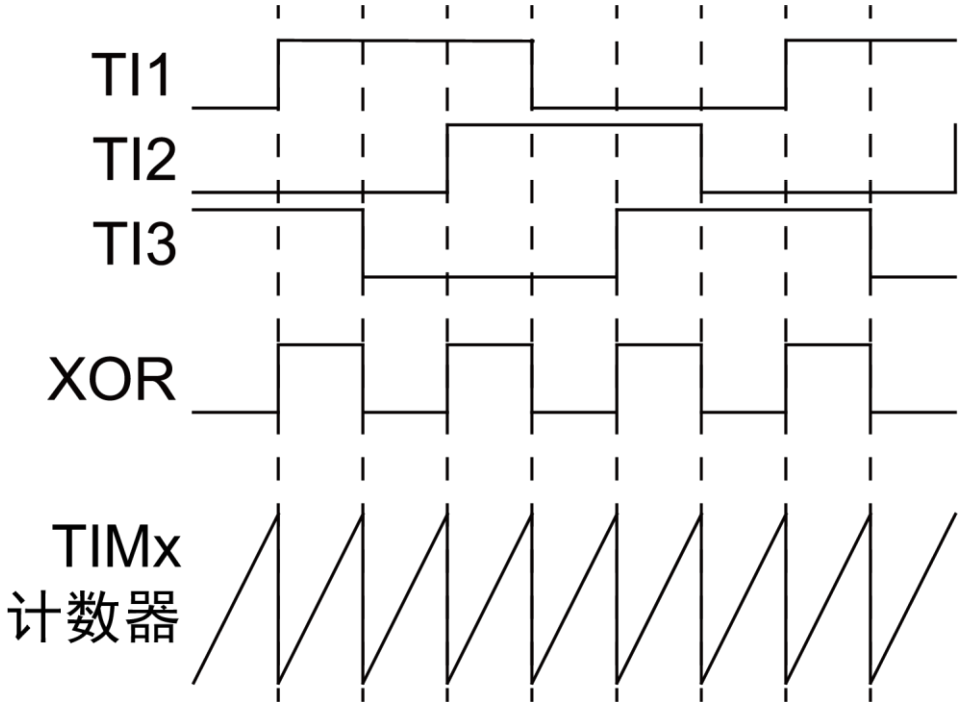
16.3.21 UIF 位重映射

TIMx_CR1 寄存器中的 IUFREMAP 位强制将更新中断标志（UIF）连续复制到定时计数器寄存器的位 31（TIMxCNT[31]）中。这样便可按原子方式读取计数器值以及由 UIFCPY 标志发出的潜在的电位翻转条件。在特定情况下，这可避免在后台任务（计数器读）和中断（更新中断）之间共享处理时产生竞争条件，从而简化计算。UIF 和 UIFCPY 标志使能之间没有延迟。

16.3.22 定时器输入异或功能

通过 TIMx_CR2 寄存器中的 TI1S 位，可将通道 1 的输入滤波器连接到异或门的输出，从而将 TIMx_CH1 和 TIMx_CH2 这两个输入引脚组合在一起。异或输出可与触发或输入捕获等所有定时器输入功能配合使用。便于测量两个输入信号上边沿之间的时间间隔，如“图 80. 测量 3 个信号上边沿之间的时间间隔”所示。

图 80. 测量 3 个信号上边沿之间的时间间隔



16.4 TIM1 寄存器

16.4.1 TIM1 控制寄存器 1（TIMx_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.	UIFREMAP	Res.	CKD[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
				rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:12	保留，始终读为 0。
位 11	**UIFREMAP**：UIF 状态位重映射。 0：无重映射。UIF 状态位不复制到 TIMx_CNT 寄存器的位 31。 1：重映射使能。UIF 状态位复制到 TIMx_CNT 寄存器的位 31。
位 10	保留，始终读为 0。
位 9:8	**CKD[1:0]**：时钟分频。 此位字段指示定时器时钟（CK_INT）频率与死区时间发生器和数字滤波器（ETR,TIx）使用的死区时间和采样时钟（tDTS）之间的分频比。 00：t_{DTS}=tCK_INT 01：t_{DTS}=2*tCK_INT 10：t_{DTS}=4*tCK_INT 11：保留，不要设置成此值
位 7	**ARPE**：自动重载预装载使能。 0：TIMx_ARR 寄存器不缓冲 1：TIMx_ARR 寄存器进行缓冲
位 6:5	**CMS[1:0]**：中心对齐模式选择。 00：边沿对齐模式。计数器根据方向位（DIR）递增计数或递减计数。 01：中心对齐模式 1。计数器交替进行递增计数和递减计数。仅当计数器递减计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CxS=00）的输出比较中断标志才置 1。 10：中心对齐模式 2。计数器交替进行递增计数和递减计数。仅当计数器递增计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CxS=00）的输出比较中断标志才置 1。 11：中心对齐模式 3。计数器交替进行递增计数和递减计数。当计数器递增计数或递减计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CxS=00）的输出比较中断标志都会置 1。 注意：只要计数器处于使能状态（CEN=1），就不得从边沿对齐模式切换为中心对齐模式。
位 4	**DIR**：方向 0：计数器用作递增计数器 1：计数器用作递减计数器 注意：当定时器配置为中心对齐模式或编码器模式时，该位为只读状态。
位 3	**OPM**：单脉冲模式。 0：计数器在发生更新事件时不会停止计数 1：计数器在发生下一更新事件时停止计数（将 CEN 位清零）
位 2	**URS**：更新请求源。 此位由软件置 1 和清零，用以选择 UEV 事件源。 0：使能后，所有以下事件都会生成更新中断。此类事件包括： • 计数器上溢/下溢

	• 将 UG 位置 1 • 通过从模式控制器更新生成 1：使能后，只有计数器上溢/下溢生成更新中断
位 1	**UDIS**：更新禁止。 此位由软件置 1 和清零，用以使能/禁止 UEV 事件生成。 0：使能 UEV。更新（UEV）事件可通过以下事件之一产生： • 计数器上溢/下溢 • 将 UG 位置 1 • 然后，通过从模式控制器缓冲寄存器完成的更新生成加载其预装载值。 1：禁止 UEV。不会生成更新事件，各影子寄存器的值（ARR、PSC 和 CCRx）保持不变。但如果将 UG 位置 1，或者从从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器。
位 0	**CEN**：使能计数器。 0：计数器禁用 1：计数器已使能 注意：只有事先通过软件将 CEN 位置 1，才可以使用外部时钟、门控模式和编码器模式。而触发模式可通过硬件自动将 CEN 位置 1。

16.4.2 TIM1 控制寄存器 2（TIMx_CR2）

偏移地址：0x04

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.	OIS6	Res.	OIS5
													rw		rw
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	Res.	Res.	Res.	Res.	CCUS	Res.	CCPC
	rw	rw	rw	rw	rw	rw	rw	rw					rw		rw

位 31:19	保留，始终读为 0。
位 18	**OIS6**：输出空闲状态 6（OC6 输出）。请参见 OIS1 位。
位 17	保留，始终读为 0。
位 16	**OIS5**：输出空闲状态 5（OC5 输出）。请参见 OIS1 位。
位 15	保留，始终读为 0。
位 14	**OIS4**：输出空闲状态 4（OC4 输出）。 请参见 OIS1 位。
位 13	**OIS3N**：输出空闲状态 3（OC3N 输出）。请参见 OIS1N 位。
位 12	**OIS3**：输出空闲状态 3（OC3 输出）。请参见 OIS1 位。
位 11	**OIS2N**：输出空闲状态 2（OC2N 输出）。请参见 OIS1N 位。
位 10	**OIS2**：输出空闲状态 2（OC2 输出）。请参见 OIS1 位。

位 9	**OIS1N**：输出空闲状态 1（OC1N 输出）。 0：当 MOE=0 时，死区时间后 OC1N=0 1：当 MOE=1 时，死区时间后 OC1N=0 注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1、2 或 3，此位即无法修改。
位 8	**OIS1**：输出空闲状态 1（OC1 输出）。 0：当 MOE=0 时，OC1=0（如果应用了 OC1N，则在死区时间后） 1：当 MOE=1 时，OC1=0（如果应用了 OC1N，则在死区时间后） 注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1、2 或 3，此位即无法修改。
位 7	**TI1S**：TI1 选择。 0：TIMx_CH1 引脚连接到 TI1 输入 1：TIMx_CH1、CH2 和 CH3 引脚连接到 TI1 输入（异或组合）
位 6:3	保留，始终读为 0。
位 2	**CCUS**：捕获/比较控制更新选择。 0：如果捕获/比较控制位（CCPC=1）进行预装载，仅通过将 COMG 位置 1 来对这些位进行更新 1：如果捕获/比较控制位（CCPC=1）进行预装载，可通过将 COMG 位置 1 或 TRGI 的上升沿对这些位进行更新 注意：此位仅对具有互补输出的通道有效。
位 1	保留，始终读为 0。
位 0	**CCPC**：捕获/比较预装载控制。 0：CCxE、CCxNE 和 OCxM 位未进行预装载 1：CCxE、CCxNE 和 OCxM 位进行了预装载，写入这些位后，仅当发生换向事件（COM）（COMG 位置 1 或在 TRGI 上检测到上升沿，取决于 CCUS 位）时才会对这些位进行更新 注意：此位仅对具有互补输出的通道有效。

16.4.3 TIM1 从模式控制寄存器（TIMx_SMCR）

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.	SMS[3]
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				Res.	TS[2:0]			OCCS	SMS[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31:17	保留，始终读为 0。
位 16	**SMS[3]**：从模式选择 - 位 3。请参见 SMS 说明 - 位 2:0。
位 15	**ETP**：外部触发信号极性。 此位可选择将 ETR 还是 ETR 用于触发操作。 0：ETR 未反相，高电平或上升沿有效 1：ETR 反相，低电平或下降沿有效

位 14	**ECE**：外部时钟使能。 此位可使能外部时钟模式 2。 0：禁止外部时钟模式 2 1：使能外部时钟模式 2。计数器时钟由 ETRF 信号的任意有效边沿提供。 注意：1：将 ECE 位置 1 与选择外部时钟模式 1 并将 TRGI 连接到 ETRF（SMS=111 且 TS=111）具有相同效果。 2：外部时钟模式 2 可以和以下从模式同时使用：复位模式、门控模式和触发模式。不过此类情况下 TRGI 不得连接 ETRF（TS 位不得为 111）。 3：如果同时使能外部时钟模式 1 和外部时钟模式 2，则外部时钟输入为 ETRF。
位 13:12	**ETPS[1:0]**：外部触发信号预分频器。 外部触发信号 ETRP 频率不得超过 TIMxCLK 频率的 1/4。可通过使能预分频器来降低 ETRP 频率。这种方法在输入快速外部时钟时非常有用。 00：预分频器关闭 01：2 分频 ETRP 频率 10：4 分频 ETRP 频率 11：8 分频 ETRP 频率
位 11:8	**CNT[3:0]**：外部触发信号滤波器。 此位域可定义 ETRP 信号的采样频率和适用于 ETRP 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个事件才视为一个有效边沿： 0000：无滤波器，在 fDTS 执行采样 0001：$f_{SAMPLING}$=fCK_INT, N=2 0010：$f_{SAMPLING}=f_{CK_INT}$, N=4 0011：$f_{SAMPLING}=f_{CK_INT}$, N=8 0100：$f_{SAMPLING}=f_{DTS}/2$, N=6 0101：$f_{SAMPLING}=f_{DTS}/2$, N=8 0110：$f_{SAMPLING}=f_{DTS}/4$, N=6 0111：$f_{SAMPLING}=f_{DTS}/4$, N=8 1000：$f_{SAMPLING}=f_{DTS}/8$, N=6 1001：$f_{SAMPLING}=f_{DTS}/8$, N=8 1010：$f_{SAMPLING}=f_{DTS}/16$, N=5 1011：$f_{SAMPLING}=f_{DTS}/16$, N=6 1100：$f_{SAMPLING}=f_{DTS}/16$, N=8 1101：$f_{SAMPLING}=f_{DTS}/32$, N=5 1110：$f_{SAMPLING}=f_{DTS}/32$, N=6 1111：$f_{SAMPLING}=f_{DTS}/32$, N=8
位 7	保留，始终读为 0。
位 6:4	**TS[2:0]**：触发信号选择。 此位字段可选择将要用于同步计数器的触发输入。 101：滤波后的定时器输入 1（TI1FP1） 110：滤波后的定时器输入 2（TI2FP2） 其他：保留 注意：这些位只能在未使用的情况下（例如，SMS=000 时）进行更改，以避免转换时出现错误的边沿检测。
位 3	**OCCS**：OCREF 清零选择。 该位用于选择 OCREF 清零源。 0：OCREF_CLR_INT 连接到 OCREF_CLR 输入（保持为 0，因此无影响）

	1：OCREF_CLR_INT 连接到 ETRF
位 2:0	**SMS**：从模式选择。 选择外部信号时，触发信号（TRGI）的有效边沿与外部输入上所选的极性相关（请参见输入控制寄存器和控制寄存器说明）。 0000：禁止从模式——如果 CEN = “1”，预分频器时钟直接由内部时钟提供 0001：编码器模式 1——计数器根据 TI1FP1 电平在 TI2FP2 边沿递增/递减计数 0010：编码器模式 2——计数器根据 TI2FP2 电平在 TI1FP1 边沿递增/递减计数 0011：编码器模式 3——计数器在 TI1FP1 和 TI2FP2 的边沿计数，计数的方向取决于另外一个信号的电平 0100：复位模式——在出现所选触发输入（TRGI）上升沿时，重新初始化计数器并生成一个寄存器更新事件 0101：门控模式——触发输入（TRGI）为高电平时使能计数器时钟。只要触发输入变为低电平，计数器立即停止计数（但不复位）。计数器的启动和停止都被控制。 0110：触发模式——触发信号 TRGI 出现上升沿时启动计数器（但不复位）。只控制计数器的启动。 0111：外部时钟模式 1——在所选触发（TRGI）的上升沿驱动计数器。 1000：组合复位 + 触发模式——在出现所选触发输入（TRGI）上升沿时，重新初始化计数器，生成一个寄存器更新事件并启动计数器。 高于 1000 的代码：保留 注意：如果将 TI1F_ED 选作触发输入（TS=“100”），则不得使用门控模式。实际上，TI1F 每次转换时，TI1F_ED 都输出 1 个脉冲，而门控模式检查的则是触发信号的电平。

16.4.4 TIM1 中断使能寄存器（TIMx_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.	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
								rw	rw	rw	rw	rw	rw	rw	rw

位 15:8	保留，始终读为 0。
位 7	**BIE**：断路中断使能。 0：禁止断路中断 1：使能断路中断
位 6	**TIE**：使能触发信号（TGRI）中断。 0：禁止触发信号（TGRI）中断 1：使能触发信号（TGRI）中断
位 5	**COMIE**：COM 中断使能。 0：禁止 COM 中断 1：使能 COM 中断
位 4	**CC4IE**：捕获/比较 4 中断使能。 0：禁止 CC4 中断 1：使能 CC4 中断
位 3	**CC3IE**：捕获/比较 3 中断使能。 0：禁止 CC3 中断 1：使能 CC3 中断

位 2	**CC2IE**：捕获/比较 2 中断使能。 0：禁止 CC2 中断 1：使能 CC2 中断
位 1	**CC1IE**：捕获/比较 1 中断使能。 0：禁止 CC1 中断 1：使能 CC1 中断
位 0	**UIE**：更新中断使能。 0：禁止更新中断 1：使能更新中断

16.4.5 TIM1 状态寄存器（TIMx_SR）

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.	CC6IF	CC5IF
														rc_w0	rc_w0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CC4OF	CC3OF	CC2OF	CC1OF	B2IF	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	rc_w0

位 31:18	保留，始终读为 0。
位 17	**CC6IF**：比较 6 中断标志。 请参见 CC1IF 说明（注意：通道 6 只能配置为输出）。
位 16	**CC5IF**：比较 5 中断标志。 请参见 CC1IF 说明（注意：通道 5 只能配置为输出）。
位 15:13	保留，始终读为 0。
位 12	**CC4OF**：捕获/比较 4 重复捕获标志。请参见 CC1OF 说明。
位 11	**CC3OF**：捕获/比较 3 重复捕获标志。请参见 CC1OF 说明。
位 10	**CC2OF**：捕获/比较 2 重复捕获标志。请参见 CC1OF 说明。
位 9	**CC1OF**：捕获/比较 1 重复捕获标志。 仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。通过软件写入“0”可将该位清零。 0：未检测到重复捕获 1：TIMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1
位 8	**B2IF**：断路 2 中断标志。 只要断路 2 输入变为有效状态，此标志便由硬件置 1。断路 2 输入无效后可通过软件对其清零。 0：未发生断路事件 1：在断路 2 输入上检测到有效电平。如果 TIMx_DIER 寄存器中 BIE=1，则会生成中断。
位 7	**BIF**：断路中断标志。 只要断路输入变为有效状态，此标志便由硬件置 1。断路输入无效后可通过软件对其清零。 0：未发生断路事件 1：在断路输入上检测到有效电平。如果 TIMx_DIER 寄存器中 BIE=1，则会生成中断。
位 6	**TIF**：触发中断标志。

	在除门控模式以外的所有模式下，当使能从模式控制器后在 TRGI 输入上检测到有效边沿时，该标志将由硬件置 1。选择门控模式时，该标志将在计数器启动或停止时置 1。但需要通过软件清零。 0：未发生触发事件 1：触发中断挂起
位 5	**COMIF**：COM 中断标志。 此标志在发生 COM 事件时（捕获/比较控制位 CCxE、CCxNE 和 OCxM 已更新时）由硬件置 1。但需要通过软件清零。 0：未发生 COM 事件 1：COM 中断挂起
位 4	**CC4IF**：捕获/比较 4 中断标志。请参见 CC1IF 说明。
位 3	**CC3IF**：捕获/比较 3 中断标志。请参见 CC1IF 说明。
位 2	**CC2IF**：捕获/比较 2 中断标志。请参见 CC1IF 说明。
位 1	**CC1IF**：捕获/比较 1 中断标志。 **如果通道 CC1 配置为输出：** 当计数器与比较值匹配时，此标志由硬件置 1，中心对齐模式下除外（请参见 TIMx_CR1 寄存器中的 CMS 位说明）。但需要通过软件清零。 0：不匹配 1：TIMx_CNT 计数器的值与 TIMx_CCR1 寄存器的值匹配。当 TIMx_CCR1 的内容大于 TIMx_ARR 的内容时，CC1IF 位在计数器上溢（在递增计数和递增/递减计数模式下）或下溢（在递减计数模式下）时变为高电平。 **如果通道 CC1 配置为输入：** 此位将在发生捕获事件时由硬件置 1。通过软件或读取 TIMx_CCR1 寄存器将该位清零。 0：未发生输入捕获事件 1：TIMx_CCR1 寄存器中已捕获到计数器值（IC1 上已检测到与所选极性匹配的边沿）
位 0	**UIF**：更新中断标志。 该位在发生更新事件时通过硬件置 1。但需要通过软件清零。 0：未发生更新 1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1： • TIMx_CR1 寄存器中的 UDIS=0，并且重复计数器值上溢或下溢时（重复计数器 = 0 时更新）。 • TIMx_CR1 寄存器中的 URS = 0 且 UDIS = 0，并且由软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。 • TIMx_CR1 寄存器中的 URS = 0 且 UDIS = 0，并且由触发事件（参见第 16.4.3 节“TIM1 从模式控制寄存器（TIMx_SMCR）”）重新初始化 CNT 时。

16.4.6 TIM1 事件生成寄存器（TIMx_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.	B2G	BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
							w	w	w	w	w	w	w	w	w

位 15:9	保留，始终读为 0。
位 8	**B2G**：断路 2 生成。 此位由软件置 1 以生成事件，并由硬件自动清零。 0：不执行任何操作 1：生成断路 2 事件。MOE 位清零且 B2IF 标志置 1。使能后可发生相关中断。
位 7	**BG**：断路生成。 此位由软件置 1 以生成事件，并由硬件自动清零。 0：不执行任何操作 1：生成断路事件。MOE 位清零且 BIF 标志置 1。使能后可发生相关中断。
位 6	**TG**：触发信号生成。 此位由软件置 1 以生成事件，并由硬件自动清零。 0：不执行任何操作 1：TIMx_SR 寄存器中的 TIF 标志置 1。使能后可发生相关中断。
位 5	**COMG**：捕获/比较控制更新生成。 该位可通过软件置 1，并由硬件自动清零。 0：不执行任何操作 1：CCPC 位置 1 时，可更新 CCxE、CCxNE 和 OCxM 位 注意：此位仅对具有互补输出的通道有效。
位 4	**CC4G**：捕获/比较 4 生成。请参见 CC1G 说明。
位 3	**CC3G**：捕获/比较 3 生成。请参见 CC1G 说明。
位 2	**CC2G**：捕获/比较 2 生成。请参见 CC1G 说明。
位 1	**CC1G**：捕获/比较 1 生成。 此位由软件置 1 以生成事件，并由硬件自动清零。 0：不执行任何操作 1：通道 1 上生成捕获/比较事件： **如果通道 CC1 配置为输出：** 使能后，CC1IF 标志置 1 并发送相应的中断。 **如果通道 CC1 配置为输入：** TIMx_CCR1 寄存器中将捕获到计数器当前值。使能后，CC1IF 标志置 1 并发送相应中断。如果 CC1IF 标志已为高电平，CC1OF 标志将置 1。
位 0	**UG**：更新生成。 该位可通过软件置 1，并由硬件自动清零。 0：不执行任何操作 1：重新初始化计数器并生成一个寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。如果选择中心对齐模式或 DIR=0（递增计数），则计数器清零，否则自动重载。 如果 DIR=1（递减计数），为值（TIMx_ARR）。

16.4.7 TIM1 捕获/比较模式寄存器 1（TIMx_CCMR1）

偏移地址：0x18

复位值：0x0000

这些通道可用于输入（捕获模式）或输出（比较模式）模式。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其它位在输入模式和输出模式下的功能均不同。对于任一给定位，OCxx 用于说明通道配置为输出时该位对应的功能，ICxx 则用于说明通道配置为输入时该位对应的功能。因此，必须注意同一个位在输入阶段和输出阶段具有不同的含义。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]
							Res.								Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2 CE	OC2M[2:0]			OC2 PE	OC2 FE	CC2S[1:0]		OC1 CE	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：

位 31:25	保留，始终读为 0。
位 24	**OC2M[3]**：输出比较 2 模式 - 位 3
位 23:17	保留，始终读为 0。
Bits16	**OC1M[3]**：输出比较 1 模式 - 位 3。请参见 OC1M 说明 - 位 6:4。
位 15	**OC2CE**：输出比较 2 清零使能。
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]**：捕获/比较 2 选择。 此位字段定义通道方向（输入/输出）以及所使用的输入。 00：CC2 通道配置为输出 01：CC2 通道配置为输入，IC2 映射到 TI2 上 10：CC2 通道配置为输入，IC2 映射到 TI1 上 11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效。 注意：仅当通道关闭时（TIMx_CCER 中的 CC2E = “0”），才可向 CC2S 位写入数据。
位 7	**OC1CE**：输出比较 1 清零使能。 0：OC1 Ref 不受 ETRF 输入的影响 1：OC1 Ref 输入上检测到高电平时，OC1 立即清零
位 6:4	**OC1M**：输出比较 1 模式。 这些位定义提供 OC1 和 OC1N 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效，而 OC1 和 OC1N 的有效电平则取决于 CC1P 位和 CC1NP 位。 0000：冻结 - 输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 进行比较不会对输出造成任何影响。（该模式用于生成时基）。 0001：将通道 1 设置为匹配时输出有效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1（TIMx_CCR1）匹配时，OC1REF 信号强制变为高电平。 0010：将通道 1 设置为匹配时输出无效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1（TIMx_CCR1）匹配时，OC1REF 信号强制变为低电平。 0011：翻转 - TIMx_CNT=TIMx_CCR1 时，OC1REF 发生翻转。 0100：强制变为无效电平 - OC1REF 强制变为低电平。

	0101：强制变为有效电平 - OC1REF 强制变为高电平。 0110：PWM 模式 1 - 在递增计数模式下，只要 TIMx_CNT<TIMx_CCR1，则通道 1 为有效状态，否则为无效状态。在递减计数模式下，只要 TIMx_CNT>TIMx_CCR1，则通道 1 为无效状态（OC1REF='0'），否则为有效状态（OC1REF='1'）。 0111：PWM 模式 2 - 在递增计数模式下，只要 TIMx_CNT<TIMx_CCR1，则通道 1 为无效状态，否则为有效状态。在递减计数模式下，只要 TIMx_CNT>TIMx_CCR1，则通道 1 为有效状态，否则为无效状态。 1000：可再触发 OPM 模式 1 - 在递增计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为有效状态。在递减计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为无效状态。 1001：可再触发 OPM 模式 2 - 在递增计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 2 下进行比较，通道会在下一次更新时再次变为无效状态。在递减计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为有效状态。 1010：保留 1011：保留 1100：组合 PWM 模式 1 - OC1REF 与在 PWM 模式 1 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑或运算结果。 1101：组合 PWM 模式 2 - OC1REF 与在 PWM 模式 2 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑与运算结果。 1110：不对称 PWM 模式 1 - OC1REF 与在 PWM 模式 1 下的行为相同。计数器递增计数时，OC1REFC 输出 OC1REF；计数器递减计数时，OC1REFC 输出 OC2REF。 1111：不对称 PWM 模式 2 - OC1REF 与在 PWM 模式 2 下的行为相同。计数器递增计数时，OC1REFC 输出 OC1REF；计数器递减计数时，OC1REFC 输出 OC2REF。 注意：1：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 3 且 CC1S=“00”（通道配置为输出），这些位即无法修改。 注 2：在 PWM 模式下，仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时，OCREF 电平才会发生更改。 注 3：此位字段将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成 COM 事件时，OC1M 有效位才会从预装载位获取新值。
位 3	**OC1PE**：输出比较 1 预装载使能 0：禁止与 TIMx_CCR1 相关的预装载寄存器。可随时向 TIMx_CCR1 写入数据，写入后将立即使用新值。 1：使能与 TIMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIMx_CCR1 预装载值在每次生成更新事件时都会装载到活动寄存器中。 注 1：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 3 且 CC1S=“00”（通道配置为输出），这些位即无法修改。 注 2：只有单脉冲模式下才可在未验证预装载寄存器的情况下使用 PWM 模式（TIMx_CR1 寄存器中的 OPM 位置 1）。否则，行为无法保证。
位 2	**OC1FE**：输出比较 1 快速使能 此位用于加快触发输入事件对 CC 输出的影响。 0：即使触发开启，CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时，激活 CC1 输出的最短延迟时间为 5 个时钟周期。 1：触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OCFE 才会起作用。
位 1:0	**CC1S**：捕获/比较 1 选择。 此位字段定义通道方向（输入/输出）以及所使用的输入。 00：CC1 通道配置为输出 01：CC1 通道配置为输入，IC1 映射到 TI1 上 10：CC1 通道配置为输入，IC1 映射到 TI2 上

	11：CC1channel 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效。 注意：仅当通道关闭时（TIMx_CCER 中的 CC1E = “0”），才可向 CC1S 位写入数据。

输入捕获模式

位 31:16	保留，始终读为 0。
位 15:12	**IC2F**：输入捕获 2 滤波器。
位 11:10	**IC2PSC[1:0]**：输入捕获 2 预分频器。
位 9:8	**CC2S**：捕获/比较 2 选择。 此位字段定义通道方向（输入/输出）以及所使用的输入。 00：CC2 通道配置为输出 01：CC2 通道配置为输入，IC2 映射到 TI2 上 10：CC2 通道配置为输入，IC2 映射到 TI1 上 11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效。 注意：仅当通道关闭时（TIMx_CCER 中的 CC2E = “0”），才可向 CC2S 位写入数据。
位 7:4	**IC1F[3:0]**：输入捕获 1 滤波器。 此位字段可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个事件才视为一个有效边沿： 0000：无滤波器，在 f_{DTS} 执行采样 0001：$f_{SAMPLING}=f_{CK_INT}$, N=2 0010：$f_{SAMPLING}=f_{CK_INT}$, N=4 0011：$f_{SAMPLING}=f_{CK_INT}$, N=8 0100：$f_{SAMPLING}=f_{DTS}/2$, N=6 0101：$f_{SAMPLING}=f_{DTS}/2$, N=8 0110：$f_{SAMPLING}=f_{DTS}/4$, N=6 0111：$f_{SAMPLING}=f_{DTS}/4$, N=8 1000：$f_{SAMPLING}=f_{DTS}/8$, N=6 1001：$f_{SAMPLING}=f_{DTS}/8$, N=8 1010：$f_{SAMPLING}=f_{DTS}/16$, N=5 1011：$f_{SAMPLING}=f_{DTS}/16$, N=6 1100：$f_{SAMPLING}=f_{DTS}/16$, N=8 1101：$f_{SAMPLING}=f_{DTS}/32$, N=5 1110：$f_{SAMPLING}=f_{DTS}/32$, N=6 1111：$f_{SAMPLING}=f_{DTS}/32$, N=8
位 3:2	**IC1PSC**：输入捕获 1 预分频器。 此位字段定义 CC1 输入（IC1）的预分频比。只要 CC1E=“0”（TIMx_CCER 寄存器），预分频器便立即复位。 00：无预分频器，捕获输入上每检测到一个边沿便执行捕获 01：每发生 2 个事件便执行一次捕获 10：每发生 4 个事件便执行一次捕获 11：每发生 8 个事件便执行一次捕获

位 1:0	**CC1S**：捕获/比较 1 选择 此位字段定义通道方向（输入/输出）以及所使用的输入。 00：CC1 通道配置为输出 01：CC1 通道配置为输入，IC1 映射到 TI1 上 10：CC1 通道配置为输入，IC1 映射到 TI2 上 11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效。 注意：仅当通道关闭时（TIMx_CCER 中的 CC1E = “0”），才可向 CC1S 位写入数据。

16.4.8 TIM1 捕获/比较模式寄存器 2（TIMx_CCMR2）

偏移地址：0x1C

复位值：0x0000

请参见第 16.4.7 节“TIM1 捕获/比较模式寄存器 1（TIMx_CCMR1）”。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC3M[3]
							Res.								Res.
							rw								rw
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]		OC3CE	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 31:25	保留，始终读为 0。
位 24	**OC4M[3]**：输出比较 4 模式 - 位 3
位 23:17	保留，始终读为 0。
位 16	**OC3M[3]**：输出比较 3 模式 - 位 3。
位 15	**OC4CE**：输出比较 4 清零使能。
位 14:12	**OC4M**：输出比较 4 模式。
位 11	**OC4PE**：输出比较 4 预装载使能。
位 10	**OC4FE**：输出比较 4 快速使能。
位 9:8	**CC4S**：捕获/比较 4 选择。 此位字段定义通道方向（输入/输出）以及所使用的输入。 00：CC4 通道配置为输出 01：CC4 通道配置为输入，IC4 映射到 TI4 上 10：CC4 通道配置为输入，IC4 映射到 TI3 上 11：CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效。 注意：仅当通道关闭时（TIMx_CCER 中的 CC4E = “0”），才可向 CC4S 位写入数据。
位 7	**OC3CE**：输出比较 3 清零使能。
位 6:4	**OC3M**：输出比较 3 模式。

位 3	**OC3PE**：输出比较 3 预装载使能。
位 2	**OC3FE**：输出比较 3 快速使能。
位 1:0	**CC3S**：捕获/比较 3 选择。 此位字段定义通道方向（输入/输出）以及所使用的输入。 00：CC3 通道配置为输出 01：CC3 通道配置为输入，IC3 映射到 TI3 上 10：CC3 通道配置为输入，IC3 映射到 TI4 上 11：CC3 通道配置为输入，IC3 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效。 注意：仅当通道关闭时（TIMx_CCER 中的 CC3E = “0”），才可向 CC3S 位写入数据。

输入捕获模式

位 31:16	保留，始终读为 0。
位 15:12	**IC4F**：输入捕获 4 滤波器。
位 11:10	**IC4PSC**：输入捕获 4 预分频器。
位 9:8	**CC4S**：捕获/比较 4 选择。 此位字段定义通道方向（输入/输出）以及所使用的输入。 00：CC4 通道配置为输出 01：CC4 通道配置为输入，IC4 映射到 TI4 上 10：CC4 通道配置为输入，IC4 映射到 TI3 上 11：CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效。 注意：仅当通道关闭时（TIMx_CCER 中的 CC4E = “0”），才可向 CC4S 位写入数据。
位 7:4	**IC3F**：输入捕获 3 滤波器。
位 3:2	**IC3PSC**：输入捕获 3 预分频器。
位 1:0	**CC3S**：捕获/比较 3 选择。 此位字段定义通道方向（输入/输出）以及所使用的输入。 00：CC3 通道配置为输出 01：CC3 通道配置为输入，IC3 映射到 TI3 上 10：CC3 通道配置为输入，IC3 映射到 TI4 上 11：CC3 通道配置为输入，IC3 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效。 注意：仅当通道关闭时（TIMx_CCER 中的 CC3E = “0”），才可向 CC3S 位写入数据。

16.4.9 TIM1 捕获/比较使能寄存器（TIMx_CCER）

偏移地址：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.	CC6P	CC6E	Res.	Res.	CC5P	CC5E
										rw	rw			rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

CC4NP	CC4NE	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	rw	rw

位 31:22	保留，始终读为 0。
位 21	**CC6P**：捕获/比较 6 输出极性。请参见 CC1P 说明。
位 20	**CC6E**：捕获/比较 6 输出使能。请参见 CC1E 说明。
位 19:18	保留，始终读为 0。
位 17	**CC5P**：捕获/比较 5 输出极性。请参见 CC1P 说明。
位 16	**CC5E**：捕获/比较 5 输出使能。请参见 CC1E 说明。
位 15	**CC4NP**：捕获/比较 4 互补输出极性。 请参见 CC1NP 说明。
位 14	**CC4NE**：捕捉/比较 4 互补输出使能。请参见 CC1NE 说明。
位 13	**CC4P**：捕获/比较 4 输出极性。请参见 CC1P 说明。
位 12	**CC4E**：捕获/比较 4 输出使能。请参见 CC1E 说明。
位 11	**CC3NP**：捕获/比较 3 互补输出极性。请参见 CC1NP 说明。
位 10	**CC3NE**：捕捉/比较 3 互补输出使能。请参见 CC1NE 说明。
位 9	**CC3P**：捕获/比较 3 输出极性。请参见 CC1P 说明。
位 8	**CC3E**：捕获/比较 3 输出使能。请参见 CC1E 说明。
位 7	**CC2NP**：捕获/比较 2 互补输出极性。请参见 CC1NP 说明。
位 6	**CC2NE**：捕捉/比较 2 互补输出使能。请参见 CC1NE 说明。
位 5	**CC2P**：捕获/比较 2 输出极性。请参见 CC1P 说明。
位 4	**CC2E**：捕获/比较 2 输出使能。请参见 CC1E 说明。
位 3	**CC1NP**：捕获/比较 1 互补输出极性。 **CC1 通道配置为输出：** 0:OC1N 高电平 1 有效：OC1N 低电平有效 **CC1 通道配置为输入：** 此位与 CC1P 配合使用，用以定义 TI1FP1 和 TI2FP1 的极性。请参见 CC1P 说明。 注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 或 3 且 CC1S=“00”（通道配置为输出），此位立即变为不可写状态。 注意：此位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1NP 有效位才会从预装载位获取新值。
位 2	**CC1NE**：捕捉/比较 1 互补输出使能。 0：关闭 - OC1N 未激活。OC1N 电平是 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的函数。 1：开启 - OC1N 信号是相应输出引脚上的输出，具体取决于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位 注意：此位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1NE 有效位才会从预装载位获取新值。
位 1	**CC1P**：捕获/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 位可针对触发或捕获操作选择 TI1FP1 和 TI2FP1 的有效极性。 00：非反相/上升沿。电路对 TIxFP1 上升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触

	发操作），TIxFP1 未反相（在门控模式或编码器模式下执行触发操作）。 01：反相/下降沿。电路对 TIxFP1 下降沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 反相（在门控模式或编码器模式下执行触发操作）。 10：保留，不使用此配置。 11：非反相/上升沿和下降沿均触发/电路对 TIxFP1 上升沿和下降沿都敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式下执行触发操作）。编码器模式下不得使用此配置。 注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 或 3，此位立即变为不可写状态。 注意：此位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1P 有效位才会从预装载位获取新值。
位 0	**CC1E**：捕获/比较 1 输出使能。 **CC1 通道配置为输出：** 0：关闭 - OC1 未激活。OC1 电平是 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的函数。 1：开启 - OC1 信号是相应输出引脚上的输出，具体取决于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位 **CC1 通道配置为输入：** 此位决定了是否可以实际将计数器值捕获到输入捕获/比较寄存器 1（TIMx_CCR1）中。 0：禁止捕捉。 1：使能捕捉。 注意：此位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1E 有效位才会从预装载位获取新值。

表 50. 具有断路功能的互补通道 OCx 和 OCxN 的输出控制位

控制位					输出状态(1)	
MOE 位	OSSI 位	OSSR 位	CCxE 位	CCxNE 位	OCx 输出状态	OCxN 输出状态
1	X	X	0	0	禁止输出（不由定时器驱动：Hi-Z）OCx=0, OCxN=0	
		0	0	1	禁止输出（不由定时器驱动：Hi-Z） OCx=0	OCxREF + 极性 OCxN = OCxREF 异或 CCxNP
		0	1	0	OCxREF + 极性 OCx=OCxREF 异或 CCxP	禁止输出（不由定时器驱动：Hi-Z） OCxN=0
		X	1	1	OCREF + 极性 + 死区时间	对 OCREF（非 OCREF）的补充 + 极性 + 死区时间
		1	0	1	关闭状态（输出使能为无效状态）OCx=CCxP	OCxREF + 极性 OCxN = OCxREF 异或 CCxNP
		1	1	0	OCxREF + 极性 OCx=OCxREF 异或 CCxP	关闭状态（输出使能为无效状态） OCxN=CCxNP
0	0	X	X	X	禁止输出（不由定时器驱动：Hi-Z）OCx=CCxP, OCxN=CCxNP	
	1		0	0		
			0	1	异步关闭状态（输出使能为无效状态）：OCx=CCxP，OCxN=CCxNP（如果触发了 BRKIN 或 BRKIN2）。 然后（仅在触发了 BRKIN 时有效），如果有时钟：死区时间后 OCx=OISx 且 OCxN=OISxN，假定 OISx 和 OISxN 不对应于 OCX 和 OCxN，二者都处于有效状态（在半桥配置中驱动开关时可能导致短路）。 **注意：**仅当 OSSI = OSSR = 1 时，才能使用 BRKIN2	
			1	0		
			1	1		

1. 如果一个通道的两个输出均未使用（由 GPIO 接管控制），则 OISx、OISxN、CCxP 和 CCxNP 位必须保持清零状态。

注意：与互补通道 OCx 和 OCxN 相连的外部 I/O 引脚的状态取决于通道 OCx 和 OCxN 的状态以及 GPIO 寄存器。

16.4.10 TIM1 计数器（TIMx_CNT）

偏移地址：0x24

复位值：0x0000

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

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	**UIFCPY**：UIF 副本 该位是 TIMx_ISR 寄存器中 UIF 位的只读副本。如果 TIMxCR1 中的 UIFREMAP 位复位，则位 31 保留，读为 0。
位 30:16	保留，始终读为 0。
位 15:0	**CNT[15:0]**：计数器值

16.4.11 TIM1 预分频器（TIMx_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 计数器时钟频率（CK_CNT）等于 fCK_PSC /（PSC[15:0] + 1）。 PSC 包含每次发生更新事件（包括计数器通过 TIMx_EGR 寄存器中的 UG 位清零时，或在配置为“复位模式”时通过触发控制器清零时）时要装载到活动预分频器寄存器的值。

16.4.12 TIM1 自动重载寄存器（TIMx_ARR）

偏移地址：0x2C

复位值：0xFFFF

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]**：预分频器值。 ARR 为要装载到实际自动重载寄存器的值。 请参见第 16.3.1 节“时基单元”了解更多关于 ARR 更新和行为的信息。 当自动重载值为空时，计数器不工作。

16.4.13 TIM1 重复计数器寄存器（TIMx_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
REP[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0	**REP[15:0]**：重复计数器值。 使能预装载寄存器时，用户可通过这些位设置比较寄存器的更新频率（即，从预装载寄存器向活动寄存器周期性传输数据）；使能更新中断时，也可设置更新中断的生成速率。 每当与 REP_CNT 相关的递减计数器达到零时，将生成更新事件并从 REP 值重新开始计数。由于只有生成重复更新事件 U_RC 时，REP_CNT 才会重载 REP 值，因此在生成下一重复更新事件之前，无论向 TIMx_RCR 寄存器写入何值都无影响。 这意味着 PWM 模式下（REP+1）相当于： 边沿对齐模式下的 PWM 周期数 中心对齐模式下的 PWM 半周期数

16.4.14 TIM1 捕获/比较寄存器 1（TIMx_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]**：捕获/比较 1 值。 **如果通道 CC1 配置为输出：** CCR1 是实际捕获/比较 1 寄存器中要装载的值（预装载值）。 如果没有通过 TIMx_CCMR1 寄存器中的 OC1PE 位来使能预装载功能，则该值永久生效。否则只在发生更新事件时，预装载值（拷贝到实际起作用的捕获/比较 1 寄存器）。 实际捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC1 输出上发出信号的值。 **如果通道 CC1 配置为输入：** CCR1 为上一个输入捕获 1 事件（IC1）发生时的计数器值。

16.4.15 TIM1 捕获/比较寄存器 2（TIMx_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]**：捕获/比较 2 值 **如果通道 CC2 配置为输出：** CCR2 是实际捕获/比较 2 寄存器中要装载的值（预装载值）。 如果没有通过 TIMx_CCMR2 寄存器中的 OC2PE 位来使能预装载功能，则该值永久生效。否则只在发生更新事件时，预装载值（拷贝到实际起作用的捕获/比较 2 寄存器）。 实际捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC2 输出上发出信号的值。 **如果通道 CC2 配置为输入：** CCR2 为上一个输入捕获 2 事件（IC2）发生时的计数器值。

16.4.16 TIM1 捕获/比较寄存器 3（TIMx_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]**：捕获/比较值 **如果通道 CC3 配置为输出：** CCR3 是实际捕获/比较 3 寄存器中要装载的值（预装载值）。 如果没有通过 TIMx_CCMR3 寄存器中的 OC3PE 位来使能预装载功能，则该值永久生效。否则只在发生更新事件时，预装载值（拷贝到实际起作用的捕获/比较 3 寄存器）。 实际捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC3 输出上发出信号的值。 **如果通道 CC3 配置为输入：** CCR3 为上一个输入捕获 3 事件（IC3）发生时的计数器值。

16.4.17 TIM1 捕获/比较寄存器 4（TIMx_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]**：捕获/比较值 **如果通道 CC4 配置为输出：** CCR4 是实际捕获/比较 4 寄存器中要装载的值（预装载值）。 如果没有通过 TIMx_CCMR4 寄存器中的 OC4PE 位来使能预装载功能，则该值永久生效。否则只在发生更新事件时，预装载值（拷贝到实际起作用的捕获/比较 4 寄存器）。 实际捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC4 输出上发出信号的值。 **如果通道 CC4 配置为输入：** CCR4 为上一个输入捕获 4 事件（IC4）发生时的计数器值。

16.4.18 TIM1 断路和死区时间寄存器（TIMx_BDTR）

偏移地址：0x44

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.	BK2P	BK2E	BK2F[3:0]				BKF[3:0]			
						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
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

注意：由于可以根据 LOCK 配置锁定位 BK2P、BK2E、BK2F[3:0]、BKF[3:0]、AOE、BKP、BKE、OSSI、OSSR 和 DTG[7:0]的写操作，因此必须在第一次对 TIMx_BDTR 寄存器执行写访问时对这些位进行配置。

位 31:26	保留。
位 25	**BK2P**：断路 2 极性。 0：断路输入 BRK2 为低电平有效 1：断路输入 BRK2 为高电平有效 注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。 注意：对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。
位 24	**BK2E**：断路 2 使能。 此位使能完整的断路 2 保护（包括连接到 bk_acth 的所有源和 BKIN 源）。 0：断路 2 功能禁用 1：断路 2 功能使能 注意：BRKIN2 必须只在 OSSR = OSSI = 1 时使用。 注意：编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。 注意：对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。
位 23:20	**BK2F[3:0]**：断路 2 滤波器。 此位字段可定义 BRK2 输入的采样频率和适用于 BRK2 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个事件才视为一个有效边沿： 注意： 0000：无滤波器，BRK2 异步操作 0001：$f_{SAMPLING}=f_{CK_INT}$, N=2 0010：$f_{SAMPLING}=f_{CK_INT}$, N=4

	0011：$f_{SAMPLING}=f_{CK_INT}$, N=8 0100：$f_{SAMPLING}=f_{CK_INT}/2$, N=6 0101：$f_{SAMPLING}=f_{CK_INT}/2$, N=8 0110：$f_{SAMPLING}=f_{CK_INT}/4$, N=6 0111：$f_{SAMPLING}=f_{CK_INT}/4$, N=8 1000：$f_{SAMPLING}=f_{CK_INT}/8$, N=6 1001：$f_{SAMPLING}=f_{CK_INT}/8$, N=8 1010：$f_{SAMPLING}=f_{DTS}/16$, N=5 1011：$f_{SAMPLING}=f_{CK_INT}/16$, N=6 1100：$f_{SAMPLING}=f_{CK_INT}/16$, N=8 1101：$f_{SAMPLING}=f_{CK_INT}/32$, N=5 1110：$f_{SAMPLING}=f_{CK_INT}/32$, N=6 1111：$f_{SAMPLING}=f_{CK_INT}/32$, N=8 注意：编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。
位 19:16	**BKF[3:0]**：断路滤波器。 此位字段可定义 BRK 输入的采样频率和适用于 BRK 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个事件才视为一个有效边沿： 0000：无滤波器，BRK 异步操作 0001：$f_{SAMPLING}=f_{CK_INT}$, N=2 0010：$f_{SAMPLING}=f_{CK_INT}$, N=4 0011：$f_{SAMPLING}=f_{CK_INT}$, N=8 0100：$f_{SAMPLING}=f_{DTS}/2$, N=6 0101：$f_{SAMPLING}=f_{DTS}/2$, N=8 0110：$f_{SAMPLING}=f_{DTS}/4$, N=6 0111：$f_{SAMPLING}=f_{DTS}/4$, N=8 1000：$f_{SAMPLING}=f_{DTS}/8$, N=6 1001：$f_{SAMPLING}=f_{DTS}/8$, N=8 1010：$f_{SAMPLING}=f_{DTS}/16$, N=5 1011：$f_{SAMPLING}=f_{DTS}/16$, N=6 1100：$f_{SAMPLING}=f_{DTS}/16$, N=8 1101：$f_{SAMPLING}=f_{DTS}/32$, N=5 1110：$f_{SAMPLING}=f_{DTS}/32$, N=6 1111：$f_{SAMPLING}=f_{DTS}/32$, N=8 编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。
位 15	**MOE**：主输出使能。 只要断路输入（BRK 或 BRK2）为有效状态，此位便由硬件异步清零。此位由软件置 1，也可根据 AOE 位状态自动置 1。此位仅对配置为输出的通道有效。 0：响应断路 2 事件。禁止 OC 和 OCN 输出 对断路事件或对 MOE 写操作的响应。 0：OC 和 OCN 输出禁用或强制为空闲状态，具体取决于 OSSI 位。 1：如果 OC 和 OCN 输出的相应使能位（TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位）均置 1，则使能 OC 和 OCN 输出。 参见 OC/OCN 使能说明了解更多信息（第 16.4.9 节“TIM1 捕获/比较使能寄存器（TIMx_CCER）”）。

位 14	**AOE**：自动输出使能。 0：MOE 只能由软件置 1 1：MOE 可由软件置 1，也可在发生下一更新事件时自动置 1（如果断路输入 BRK 和 BRK2 均无效）。 注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。
位 13	**BKP**：断路极性。 0：断路输入 BRK 为低电平有效 1：断路输入 BRK 为高电平有效 注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。 注意：对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。
位 12	**BKE**：断路使能。 此位使能完整的断路保护（包括连接到 bk_acth 的所有源和 BKIN 源）。 0：断路功能禁用。 1：断路功能使能。 注意：编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。 注意：对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。
位 11	**OSSR**：运行模式下的关闭状态选择。 此位在 MOE=1 时作用于配置为输出模式且具有互补输出的通道。如果定时器中没有互补输出，则不存在 OSSR。 参见 OC/OCN 使能说明了解更多信息（第 16.4.9 节“TIM1 捕获/比较使能寄存器（TIMx_CCER）”）。 0：处于无效状态时，禁止 OC/OCN 输出（定时器释放输出控制，由强制高阻态的 GPIO 逻辑接管）。 1：处于无效状态时，一旦 CCxE=1 或 CCxNE=1，便使能 OC/OCN 输出并将其设为无效电平（输出仍由定时器控制）。 注意：编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 后，此位即无法修改。
位 10	**OSSI**：空闲模式下的关闭状态选择。 当由于断路事件或软件写操作而使 MOE=0 时，此位作用于配置为输出的通道。 参见 OC/OCN 使能说明了解更多信息（第 16.4.9 节“TIM1 捕获/比较使能寄存器（TIMx_CCER）”）。 0：处于无效状态时，禁止 OC/OCN 输出（定时器释放输出控制，由强制高阻态的 GPIO 逻辑接管）。 1：处于无效状态时，首先将 OC/OCN 输出强制为其无效电平，然后在死区后将其强制为空闲电平。定时器始终控制输出。 注意：编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 后，此位即无法修改。
位 9:8	**LOCK[1:0]**：锁定配置。 这些位用于针对软件错误提供写保护。 00：关闭锁定 - 不对任何位提供写保护。 01： LOCK 级别 1 = TIMx_BDTR 寄存器中的 DTG 位、TIMx_CR2 寄存器中的 OISx 和 OISxN 位以及 TIMx_BDTR 寄存器中的 BKE/BKP/AOE 位不能再写入。 10：LOCK 级别 2 = LOCK 级别 1 + CC 极性位（TIMx_CCER 寄存器中的 CCxP/CCxNP 位，只要相关通道通过 CCxS 位配置为输出）以及 OSSR 和 OSSI 位不能再写入。 11：LOCK 级别 3 = LOCK 级别 2 + CC 控制位（TIMx_CCMRx 寄存器中的 OCxM 和 OCxPE 位，只要相关通道通过 CCxS 位配置为输出）不能再写入。 注意：复位后只能对 LOCK 位执行一次写操作。对 TIMx_BDTR 寄存器执行写操作后其中的内容将冻结，直到下一次复位。

位 7:0	**DTG[7:0]**：死区时间发生器设置。 此位字段定义在互补输出之间插入的死区时间的持续时间。DT 匹配此持续时间。 DTG[7:5]=0xx => DT=DTG[7:0]x t_{dtg}，$t_{dtg}=t_{DTS}$。DTG[7:5]=10x => DT=(64+DTG[5:0]）xt_{dtg}，$T_{dtg}=2xt_{DTS}$。DTG[7:5]=110 => DT=(32+DTG[4:0] ）xt_{dtg}，$T_{dtg}=8xt_{DTS}$。DTG[7:5]=111 => DT=(32+DTG[4:0] ）xt_{dtg}，$T_{dtg}=16xt_{DTS}$。 例如，如果 T_{DTS} = 125 ns（8 MHz），则死区时间的可能值为： 0 到 15875 ns（步长为 125 ns）， 16 us 至 31750 ns，步长为 250 ns；32 us 至 63 us，步长为 1 us， 64 us 到 126 us（步长为 2 us） 注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1、2 或 3，此位字段即无法修改。

16.4.19 TIM1 捕获/比较模式寄存器 3（TIMx_CCMR3）

偏移地址：0x54

复位值：0x00000000

请参见第 16.4.7 节“TIM1 捕获/比较模式寄存器 1（TIMx_CCMR1）”了解通道 5 和 6 的配置。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC6M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC5M[3]
							rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC6 CE	OC6M[2:0]			OC6 PE	OC6FE	Res.	Res.	OC5 CE.	OC5M[2:0]			OC5PE	OC5FE	Res.	Res.
rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw		

输出比较模式

位 31:25	保留，始终读为 0。
位 24	**OC6M[3]**：输出比较 6 模式——位 3。
位 23:17	保留，始终读为 0。
位 16	**OC5M[3]**：输出比较 5 模式——位 3。
位 15	**OC6CE**：输出比较 6 清零使能。
位 14:12	**OC6M**：输出比较 6 模式。
位 11	**OC6PE**：输出比较 6 预装载使能。
位 10	**OC6FE**：输出比较 6 快速使能。
位 9:8	保留，始终读为 0。
位 7	**OC5CE**：输出比较 5 清零使能。
位 6:4	**OC5M**：输出比较 5 模式。

位 3	**OC5PE**：输出比较 5 预装载使能。
位 2	**OC5FE**：输出比较 5 快速使能。
位 1:0	保留，始终读为 0。

16.4.20 TIM1 捕获/比较寄存器 5（TIMx_CCR5）

偏移地址：0x58

复位值：0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GC5C3	GC5C2	GC5C1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	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
CCR5[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31	**GC5C3**：通道 3 输出上的组合通道 5 和通道 3 失真： 0：OC5REF 不影响 OC3REFC 1：OC3REFC 是 OC3REFC 和 OC5REF 的逻辑与运算结果 该位可以立即生效，也可预装载并在更新事件后执行（如果在 TIMxCCMR2 中选择了预装载功能）。 注意：也可在组合 PWM 信号上应用此失真。
位 30	**GC5C2**：通道 2 输出上的组合通道 5 和通道 2 失真： 0：OC5REF 不影响 OC2REFC 1：OC2REFC 是 OC2REFC 和 OC5REF 的逻辑与运算结果 该位可以立即生效，也可预装载并在更新事件后执行（如果在 TIMxCCMR1 中选择了预装载功能）。 注意：也可在组合 PWM 信号上应用此失真。
位 29	**GC5C1**：通道 1 输出上的组合通道 5 和通道 1 失真： 0：OC5REF 不影响 OC1REFC5 1：OC1REFC 是 OC1REFC 和 OC5REF 的逻辑与运算结果 该位可以立即生效，也可预装载并在更新事件后执行（如果在 TIMxCCMR1 中选择了预装载功能）。 注意：也可在组合 PWM 信号上应用此失真。
位 28:16	保留，必须保持复位值。
位 15:0	**CCR5[15:0]**：捕获/比较 5 值 CCR5 是实际捕获/比较 5 寄存器中要装载的值（预装载值）。 如果没有通过 TIMx_CCMR3 寄存器中的 OC5PE 位来使能预装载功能，则该值永久生效。否则只在发生更新事件时，预装载值（拷贝到实际起作用的捕获/比较 5 寄存器）。 实际捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC5 输出上发出信号的值。

16.4.21 TIM1 捕获/比较寄存器 6（TIMx_CCR6）

偏移地址：0x5C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR6[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0	**CCR6[15:0]**：捕获/比较 6 值。 CCR6 是实际捕获/比较 6 寄存器中要装载的值（预装载值）。 如果没有通过 TIMx_CCMR3 寄存器中的 OC6PE 位来使能预装载功能，则该值永久生效。否则只在发生更新事件时，预装载值（拷贝到实际起作用的捕获/比较 6 寄存器）。 实际捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC6 输出上发出信号的值。

16.4.22 TIM1 复用功能选项寄存器 1（TIMx_AF1）

偏移地址：0x60

复位值：0x0001

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.	BKCM P2P	BKCM P1P	BKINP	Res.	Res.	Res.	Res.	Res.	Res.	BKCMP2 E	BKCMP 1E	BKINE
				rw	rw	rw							rw	rw	rw

位 31:12	保留，必须保持复位值
位 11	**BKCMP2P**：BRK COMP2 输入极性 此位选择 COMP2 输入灵敏度。它必须和 BKP 极性位一起设定。 0：COMP2 输入为低电平有效 1：COMP2 输入为高电平有效 注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。
位 10	**BKCMP1P**：BRK COMP1 输入极性 此位选择 COMP1 输入灵敏度。它必须和 BKP 极性位一起设定。 0：COMP1 输入为低电平有效 1：COMP1 输入为高电平有效 注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。
位 9	**BKINP**：BRK BKIN 输入极性 此位选择 BKIN 复用功能输入灵敏度。它必须和 BKP 极性位一起设定。 0：BKIN 输入为低电平有效 1：BKIN 输入为高电平有效 注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。
位 8:3	保留，必须保持复位值
位 2	**BKCMP2E**：BRK COMP2 使能 此位为定时器的 BRK 输入使能 COMP2。COMP2 输出与其他 BRK 源构成逻辑或关系。 0：COMP2 输入禁用 1：COMP2 输入使能 注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 1	**BKCMP1E**：BRK COMP1 使能 此位为定时器的 BRK 输入使能 COMP1。COMP1 输出与其他 BRK 源构成逻辑或关系。 0：COMP1 输入禁用 1：COMP1 输入使能 注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。
位 0	**BKINE**：BRK BKIN 输入使能 此位为定时器的 BRK 输入使能 BKIN 复用功能输入。BKIN 输入与其他 BRK 源构成逻辑或关系。 0：BKIN 输入禁用 1：BKIN 输入使能 注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

16.4.23 TIM1 复用功能选项寄存器 2（TIMx_AF2）

偏移地址：0x64

复位值：0x0001

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.	BK2C MP2P	BK2C MP1P	BK2IN P	Res.	Res.	Res.	Res.	Res.	Res.	BK2CMP 2E	BK2CM P1E	BK2INE
				rw	rw	rw							rw	rw	rw

位 31:12	保留，必须保持复位值。
位 11	**BK2CMP2P**：BRK2 COMP2 输入极性。 此位选择 COMP2 输入灵敏度。它必须和 BKP2 极性位一起设定。 0：COMP2 输入为低电平有效 1：COMP2 输入为高电平有效 注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。
位 10	**BK2CMP1P**：BRK2 COMP1 输入极性。 此位选择 COMP1 输入灵敏度。它必须和 BKP2 极性位一起设定。 0：COMP1 输入为低电平有效 1：COMP1 输入为高电平有效 注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。
位 9	**BK2INP**：BRK2 BKIN2 输入极性。 此位选择 BKIN2 复用功能输入灵敏度。它必须和 BKP2 极性位一起设定。 0：BKIN2 输入为低电平有效 1：BKIN2 输入为高电平有效 注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。
位 8:3	保留，必须保持复位值。

位 2	**BK2CMP2E**：BRK2 COMP2 使能。 此位为定时器的 BRK2 输入使能 COMP2。COMP2 输出与其他 BRK2 源构成逻辑或关系。 0：COMP2 输入禁用 1：COMP2 输入使能 注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。
位 1	**BK2CMP1E**：BRK2 COMP1 使能。 此位为定时器的 BRK2 输入使能 COMP1。COMP1 输出与其他 BRK2 源构成逻辑或关系。 0：COMP1 输入禁用 1：COMP1 输入使能 注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。
位 0	**BK2INE**：BRK2 BKIN 输入使能。 此位为定时器的 BRK2 输入使能 BKIN2 复用功能输入。BKIN2 输入与其他 BRK2 源构成逻辑或关系。 0：BKIN2 输入禁用 1：BKIN2 输入使能 注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

16.5 TIM1 寄存器映射

TIM1 寄存器可映射为 16 位可寻址寄存器，如下表所述：

表 51. TIM1 寄存器映射和复位值

偏移	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
0x00	TIMx_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD [1:0]		ARPE	CMS [1:0]		DIR	OPM	URS	UDIS	CEN
	复位值																					0		0	0	0	0	0	0	0	0	0	0
0x04	TIMx_CR2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	OIS6	Res	OIS5	Res	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	Res	Res	Res	CCDS	CCUS	Res	CCPC
	复位值														0		0		0	0	0	0	0	0	0	0				0	0		0
0x08	TIMx_SMCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SMS[3]	ETP	ECE	ETPS [1:0]		ETF[3:0]				Res	TS[2:0]			OCCS	SMS[2:0]		
	复位值																0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0
0x0C	TIMx_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	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	复位值																									0	0	0	0	0	0	0	0
0x10	TIMx_SR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CC6IF	CC5IF	Res	Res	Res	CC4OF	CC3OF	CC2OF	CC1OF	B2IF	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
	复位值															0	0				0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	TIMx_EGR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	B2G	BG	TG	COM	CC4G	CC3G	CC2G	CC1G	UG
	复位值																								0	0	0	0	0	0	0	0	0

偏移	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
0x18	TIMx_CCMR1 输出比较模式	Res	Res	Res	Res	Res	Res	Res	OC2M[3]	Res	Res	Res	Res	Res	Res	Res	OC1M[3]	OC2CE	OC2M: [2:0]			OC2PE	OC2FE	CC2S: [1:0]		OC1CE	OC1M: [2:0]			OC1PE	OC1FE	CC1S: [1:0]	
	复位值								0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIMx_CCMR1 输入捕获模式	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]	
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	TIMx_CCMR2 输出比较模式	Res	Res	Res	Res	Res	Res	Res	OC4M[3]	Res	Res	Res	Res	Res	Res	Res	OC3M[3]	OC4CE	OC4M: [2:0]			OC4PE	OC4FE	CC4S: [1:0]		OC3CE	OC3M: [2:0]			OC3PE	OC3FE	CC3S: [1:0]	
	复位值								0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

偏移	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
0x1C	TIMx_CCMR2 输入捕获模式	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]	
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	TIMx_CCER	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CC6P	CC6E	Res	Res	CC5P	CC5E	CC4NP	CC4NE	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
	复位值											0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	TIMx_CNT	UIFCPY	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CNT[15:0]															
	复位值	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	TIMx_PSC	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PSC[15:0]															
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	TIMx_ARR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ARR[15:0]															
	复位值																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x30	TIMx_RCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	REP[15:0]															
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x34	TIMx_CCR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CCR1[15:0]															
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

偏移	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
0x38	TIMx_CCR2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CCR2[15:0]															
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	TIMx_CCR3	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CCR3[15:0]															
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	TIMx_CCR4	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CCR4[15:0]															
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	TIMx_BDTR	Res	Res	Res	Res	Res	Res	BK2P	BK2E	BK2F[3:0]				BKF[3:0]				MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK [1:0]		DT[7:0]							
	复位值							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48-0x50	保留																																
0x54	TIMx_CCMR3 输出比较模式	Res	Res	Res	Res	Res	Res	Res	OC6M[3]	Res	Res	Res	Res	Res	Res	Res	OC5M[3]	OC6CE	OC6M: [2:0]			OC6PE	OC6FE	Res	Res	OC5CE	OC5M: [2:0]			OC5PE	OC5FE	Res	Res
	复位值								0								0	0	0	0	0	0	0			0	0	0	0	0	0		

请参见第 2.2.2 节“存储器映射和寄存器边界地址”了解寄存器边界地址。

17 实时时钟（RTC）

17.1 引言

实时时钟（RTC）是一个独立的 BCD 定时器/计数器。RTC 提供具有可编程闹钟中断功能的日历时钟/日历。

RTC 还包含具有中断功能的周期性可编程唤醒标志。实时时钟（RTC）提供用于管理所有低功耗模式的自动唤醒单元。

两个 32 位寄存器包含二进码十进数格式（BCD）的秒、分钟、小时（12 或 24 小时制）、星期几、日期、月份和年份。此外，还可提供二进制格式的亚秒值。

系统可以自动将月份的天数补偿为 28、29（闰年）、30 和 31 天。并且还可以进行夏令时补偿。

其他 32 位寄存器还包含可编程的闹钟亚秒、秒、分钟、小时、星期几和日期。此外，还可以使用数字校准功能对晶振精度的偏差进行补偿。

上电复位后，所有 RTC 寄存器都会受到保护，以防止可能的非正常写访问。

无论器件状态如何（运行模式、低功耗模式或处于系统复位状态），只要电源电压保持在工作范围内，RTC 便不会停止工作。

注意： *当调试器暂停 CPU 时，RTC 计数器不冻结。*

17.2 RTC 主要特性

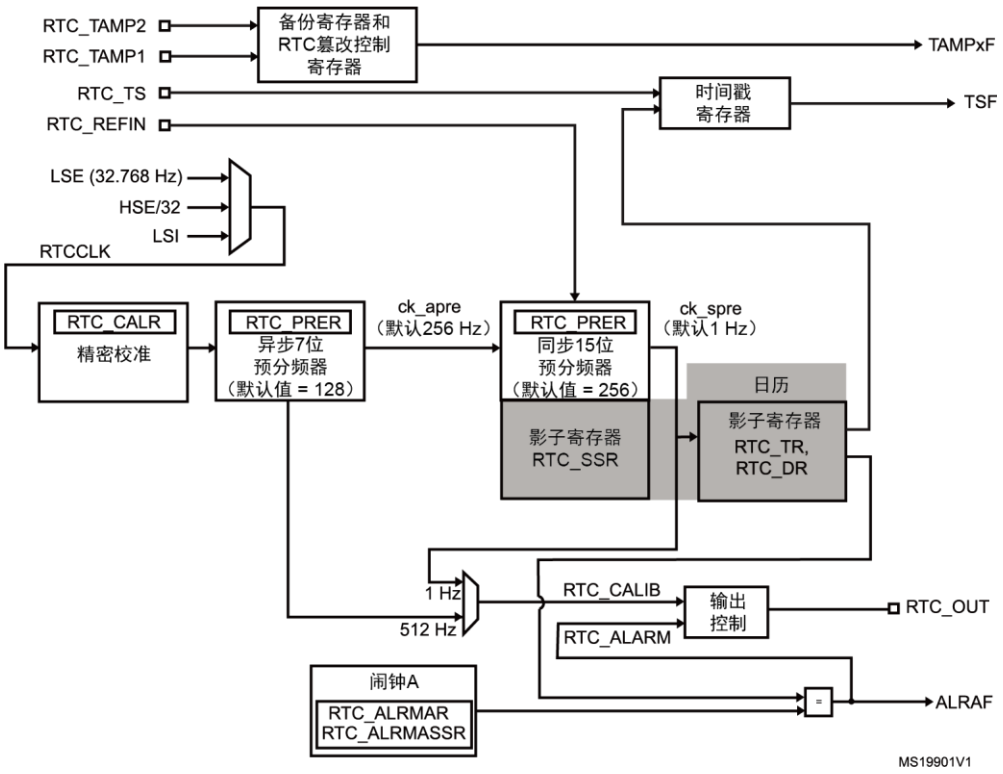
RTC 单元的主要特性如下（参见图 81. RTC 框图）：

- 包含亚秒、秒、分钟、小时（12 或 24 小时制）、星期几、日期、月份和年份的日历。
- 软件可编程的夏令时补偿
- 具有中断功能的可编程闹钟。可通过任意日历字段的组合触发闹钟
- 自动唤醒单元，可周期性地生成标志以触发自动唤醒中断
- 数字校准回路（周期性计数器校正）：0.95 ppm 精度，在数秒的校准窗口内获得
- 可屏蔽中断/事件：
  - 闹钟 A
  - 唤醒中断。

17.3 RTC 功能说明

17.3.1 RTC 框图

图 81. RTC 框图



17.3.2 时钟和预分频器

RTC 时钟源（RTCCLK）通过时钟控制器从 LSE 时钟、LSI 振荡器时钟以及 HSI_64M 时钟三者中选择。如需关于 RTC 时钟源配置的更多信息，请参见第 6 节“复位和时钟控制器（RCC）”。

可编程的预分频器阶段可生成 1 Hz 的时钟，用于更新日历。为使功耗最小化，预分频器被分成 2 个可编程预分频器（参见图 81. RTC 框图）：

- 一个通过 RTC_PRER 寄存器的 PREDIV_A 位配置的 7 位异步预分频器
- 一个通过 RTC_PRER 寄存器的 PREDIV_S 位配置的 15 位同步预分频器。

注意： *使用两个预分频器时，推荐将异步预分频器配置为较高的值，以最大程度降低功耗。*

要使用频率为 32.768 kHz 的 LSE 获得频率为 1 Hz 的内部时钟（ck_spre），需要将异步预分频系数设置为 128，并将同步预分频系数设置为 256。

分频系数的最小值为 1，最大值为 2^{22}。这对应于约 4 MHz 的最高输入频率。f_{ck_apre} 由以下公式给出：

$$f_{CK_APRE} = \frac{f_{RTCCLK}}{PREDIV_A + 1} \quad (2)$$

ck_apre 时钟用于二进制 RTC_SSR 亚秒递减计数器的计时。当它达到 0 时，用 PREDIV_S 的内容重载 RTC_SSR。

f_{ck_spre} 由以下公式给出：

$$f_{CK_SPRE} = \frac{f_{RTCCLK}}{(PREDIV_S + 1) \times (PREDIV_A + 1)} \quad (3)$$

ck_spre 时钟可用于更新日历或作为 16 位唤醒自动重载定时器的时基。为获得较短的超时周期，还可以用经可编程 4 位异步预分频器分频的 RTCCLK 运行 16 位唤醒自动重载定时器（参见第 17.3.5 节“周期性自动唤醒”）。

17.3.3 实时时钟和日历

RTC 日历时间和日期寄存器可通过与 PCLK（APB 时钟）同步的影子寄存器来访问。这些时间和日期寄存器也可以直接访问，这样可避免等待同步的持续时间。

- RTC_SSR 对应于亚秒
- RTC_TR 对应于时间
- RTC_DR 对应于日期

每隔两个 RTCCLK 周期，将当前日历值复制到影子寄存器，并且 RTC_ISR 寄存器的 RSF 位置位（参见第 17.6.4 节“RTC 初始化和状态寄存器（RTC_ISR）”）。

在停机和待机模式下不会执行复制操作。退出这两种模式时，影子寄存器会在最长 2 个 RTCCLK 周期后进行更新。

当应用读取日历寄存器时，它会访问影子寄存器的内容。也可以通过将 RTC_CR 寄存器的 BYPSHAD 控制位置 1 来直接访问日历寄存器。默认情况下，该位被清零，用户访问影子寄存器。

在 BYPSHAD=0 模式下读取 RTC_SSR、RTC_TR 或 RTC_DR 寄存器时，APB 时钟频率（fAPB）必须至少为 RTC 时钟频率（fRTCCLK）的 7 倍。

影子寄存器通过系统复位来复位。

17.3.4 可编程闹铃

RTC 单元提供可编程闹铃：闹铃 A。通过 RTC_CR 寄存器中的 ALRAE 位使能可编程闹铃功能。如果日历亚秒、秒、分钟、小时、日期和星期几与在闹铃寄存器 RTC_ALRMASSR 和 RTC_ALRMAR 中设定的值一致，则 ALRAF 置为 1。每个日历字段均可通过 RTC_ALRMAR 寄存器中的 MSKx 位和 RTC_ALRMASSR 寄存器中的 MASKSSx 位单独选择。可通过 RTC_CR 寄存器中的 ALRAIE 位来使能闹钟中断。

注意：如果选择秒字段（RTC_ALRMAR 中的 MSK0 位复位），则 RTC_PRER 寄存器中设置的同步预分频器分频系数必须至少为 3，才能确保闹钟正确地运行。

闹铃 A（若已通过 RTC_CR 寄存器中的 OSEL[0:1]位使能）可路由至 RTC_ALARM 输出。可通过 RTC_CR 寄存器的 POL 位配置 RTC_ALARM 输出极性。

17.3.5 周期性自动唤醒

周期性唤醒标志由 16 位可编程自动重载递减计数器生成。唤醒定时器范围可扩展至 17 位。

唤醒功能通过 RTC_CR 寄存器中的 WUTE 位使能。唤醒定时器时钟输入可以是：

- 2、4、8 或 16 分频的 RTC 时钟（RTCCLK）。

  当 RTCCLK 为 LSE（32.768 kHz）时，可将唤醒中断周期配置为 122 µs 至 32 s，精度可达 61 µs。

- ck_spre（通常为 1 Hz 内部时钟）

  当 ck_spre 频率为 1 Hz 时，可实现 1 s 至约 36 小时的唤醒时间，精度为 1 s。这一较大的可编程时间范围分为两部分：

  – 当 WUCKSEL [2:1] = 10 时，为 1 s 至 18 小时

  – 当 WUCKSEL[2:1] = 11 时，为约 18 小时至 36 小时。在最后这种情况下，将 216 添加到 16 位计数器当前值。当初始化序列完成（参见第 17.3.6 节“RTC 初始化和配置”）时，定时器开始递减计数。当唤醒功能使能时，递减计数在低功耗模式下保持活跃。此外，当计数器计数到 0 时，RTC_ISR 寄存器的 WUTF 标志会置 1，并且唤醒寄存器会使用其重载值（RTC_WUTR 寄存器值）动重载。

之后必须用软件清零 WUTF 标志。

通过将 RTC_CR2 寄存器中的 WUTIE 位置 1 来使能周期性唤醒中断时，它会使器件退出低功耗模式。

周期性唤醒标志可路由至 RTC_ALARM 输出，前提是它已通过 RTC_CR 寄存器的 OSEL[0:1]位使能。可通过 RTC_CR 寄存器的 POL 位配置 RTC_ALARM 输出极性。

系统复位和低功耗模式（DEEPSTOP）不影响唤醒定时器。

17.3.6 RTC 初始化和配置

RTC 寄存器访问

RTC 寄存器为 32 位寄存器。除了当 BYPSHAD=0 时对日历影子寄存器执行的读访问之外，APB 接口会在访问 RTC 寄存器时引入 2 个等待周期。

RTC 寄存器写保护

上电复位后，所有 RTC 寄存器均受到写保护。通过向写保护寄存器（RTC_WPR）写入一个密钥来使能对 RTC 寄存器的写操作。

要解锁所有 RTC 寄存器（RTC_ISR[13:8]和 RTC_BKPxR 除外）的写保护，需要执行以下步骤。

1. 将“0xCA”写入 RTC_WPR 寄存器。
2. 将“0x53”写入 RTC_WPR 寄存器。

写入一个错误的关键字会再次激活写保护。

保护机制不受系统复位影响。

日历初始化和配置

要编程包括时间格式和预分频器配置在内的初始时间和日期日历值，需按照以下顺序操作：

1. 将 RTC_ISR 寄存器中的 INIT 位置 1 进入初始化模式。在此模式下，日历计数器将停止工作并且其值可更新。
2. 轮询 RTC_ISR 寄存器中的 INITF 位。当 INITF 置 1 时进入初始化阶段模式。大约需要 2 个 RTCCLK 时钟周期（由于时钟同步）。
3. 要为日历计数器生成 1 Hz 时钟，应编程 RTC_PRER 寄存器中的两个预分频系数。
4. 在影子寄存器（RTC_TR 和 RTC_DR）中加载初始时间和日期值，然后通过 RTC_CR 寄存器中的 FMT 位配置时间格式（12 或 24 小时制）。
5. 通过清零 INIT 位退出初始化模式。随后，自动加载实际日历计数器值，在 4 个 RTCCLK 时钟周期后重新开始计数。

当初始化序列完成之后，日历开始计数。

注意： *1. 系统复位后，应用可读取 RTC_ISR 寄存器中的 INITS 标志，以检查日历是否已初始化。如果该标志为 0，表明日历尚未初始化，因为年份字段设置为其上电复位时的默认值（0x00）。*

注意： *2. 要在初始化之后读取日历，必须首先用软件检查 RTC_ISR 寄存器的 RSF 标志是否置 1。*

夏令时

可通过 RTC_CR 寄存器的 SUB1H、ADD1H 和 BKP 位管理夏令时。

利用 SUB1H 或 ADD1H，软件只需单次操作便可在日历中减去或增加一个小时，无需执行整个初始化步骤。

此外，软件还可以使用 BKP 位来记录是否曾经执行过此操作。

编程闹钟

可编程闹钟的编程或更新必须遵循相似的程序。

1. 将 RTC_CR 中的 ALRAE 位清零以禁止闹钟 A
2. 编程闹钟 A 寄存器（RTC_ALRMASSR/RTC_ALRMAR）
3. 置位 RTC_CR 寄存器中的 ALRAE 位以再次使能闹钟 A。

唤醒定时器的编程

为了配置或更改唤醒定时器自动重载值（RTC_WUTR 中的 WUT[15:0]），需按照以下顺序操作：

1. 将 RTC_CR 中的 WUTE 清零以禁用唤醒定时器。
2. 轮询 RTC_ISR 寄存器中的 WUTWF 位，直至其置位，以确保对唤醒自动重载计数器和 WUCKSEL[2:0] 位的访问被允许。大约需要 2 个 RTCCLK 时钟周期（由于时钟同步）。
3. 设定唤醒自动重载值 WUT[15:0]和唤醒时钟选择（RTC_CR 中的 WUCKSEL[2:0]位）。置位 RTC_CR 寄存器中的 WUTE 位以再次使能定时器。唤醒定时器重新开始递减计数。

17.3.7 读取日历

- 当 RTC_CR 寄存器中的 BYPSHAD 控制位清零时：

为了正确地读取 RTC 日历寄存器（RTC_SSR、RTC_TR 和 RTC_DR），APB1 时钟频率（fPCLK）必须等于或大于 fRTCCLKRTC 时钟频率的七倍。这可以确保同步机制行为的安全性。

如果 APB0 时钟频率低于 RTC 时钟频率的七倍，则软件必须分两次读取日历时间寄存器和日期寄存器。这样，当两次读取的 RTC_TR 结果相同时，才能确保数据正确。否则必须执行第三次读访问。任何情况下，APB0 的时钟频率都不能低于 RTC 的时钟频率。

每当日历寄存器被复制到 RTC_TR 和 RTC_DR 影子寄存器中时，RTC_ISR 寄存器中的 RSF 位置位。每两个 RTCCLK 周期执行一次复制。为确保这 3 个值来自同一时刻点，读取 RTC_SSR 或 RTC_TR 时会锁定高阶日历影子寄存器中的值，直到读取 RTC_DR。为避免软件对日历执行读访问的时间间隔小于 2 个 RTCCLK 周期：在第一次日历读取后，软件必须将 RSF 清零，然后软件必须等待至 RSF 置 1 之后才可以再次读取 RTC_SSR、RTC_TR 和 RTC_DR 寄存器。

在从低功耗模式（DEEPSTOP）唤醒后，软件必须将 RSF 清零。之后，软件必须等待至 RSF 再次置 1 之后才可以读取 RTC_SSR、RTC_TR 和 RTC_DR 寄存器。

RSF 位必须在唤醒之后而不是进入低功耗模式之前进行清零。

系统复位之后，软件必须等待至 RSF 置 1 之后才可以读取 RTC_SSR、RTC_TR 和 RTC_DR 寄存器。实际上，系统复位会将影子寄存器复位为其默认值。

初始化之后，软件必须等待至 RSF 置 1 之后才可以读取 RTC_SSR、RTC_TR 和 RTC_DR 寄存器。

同步（参见第 17.3.9 节“RTC 同步”）后：软件必须等待至 RSF 置 1 之后才可以读取 RTC_SSR、RTC_TR 和 RTC_DR 寄存器。

- 当 RTC_CR 寄存器中的 BYPSHAD 控制位置 1 时（旁路影子寄存器）

读取日历寄存器时会直接从日历计数器获取值，这样便无需等待至 RSF 位置 1。这对于从低功耗模式（停机模式或待机模式）退出后的情况特别有用，因为影子寄存器在这些模式下不更新。

当 BYPSHAD 位置 1 时，如果在对寄存器的两次读访问之间出现 RTCCLK 沿，则不同寄存器的结果彼此可能不一致。此外，如果在读操作期间出现 RTCCLK 沿，则可能导致其中一个寄存器的值不正确。软件必须分两次读取所有寄存器，然后将两次结果加以比较来确认数据是否一致和正确。此外，软件也可以只比较两次读取日历寄存器得到的结果的最低位。

注意： *当 BYPSHAD=1 时，读取日历寄存器的指令需要一个额外的 APB 周期才能完成。*

17.3.8 复位 RTC

日历影子寄存器（RTC_SSR、RTC_TR 和 RTC_DR）和 RTC 状态寄存器（RTC_ISR）被所有可用的系统复位源复位至其默认值。

相反地，以下寄存器通过上电复位复位至其默认值且不受系统复位的影响：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 备份寄存器（RTC_BKPxR）、唤醒定时器寄存器（RTC_WUTR）和闹铃 A 寄存器（RTC_ALRMASSR/ RTC_ALRMAR）。

此外，如果复位源不是上电复位源，则发生系统复位时，RTC 会继续工作。发生上电复位时，RTC 会停止工作，并且所有 RTC 寄存器都会设置为各自的复位值。

17.3.9 RTC 同步

RTC 可与高精度的远程时钟同步。在读取亚秒字段后（RTC_SSR 或 RTC_TSSSR），即可计算远程时钟的时间与 RTC 之间的精准偏移。之后，可使用 RTC_SHIFTR 对 RTC 的时钟进行零点几秒的“平移”，经过调整后可消除此偏移。

RTC_SSR 包含同步预分频器计数器的值。因此，可计算 RTC 维持的精确时间，精度可达 1 /（PREDIV_S + 1）秒。因此，可通过增大同步预分频器的值（PREDIV_S[14:0]）来提高分辨率。将 PREDIV_S 设置为 0x7FFF 时，可得到允许的最大分辨率（30.52 µs，时钟频率为 32768 Hz）。

但是，提高 PREDIV_S 意味着必须降低 PREDIV_A 才能将同步预分频器的输出维持在 1 Hz。这样，异步预分频器的输出频率会增大，RTC 的动态功耗也会相应增加。

可以使用 RTC 平移控制寄存器（RTC_SHIFTR）对 RTC 进行微调。可以用大小为 1/(PREDIV_S + 1）秒的分辨率对 RTC_SHIFTR 进行写操作，将时钟平移（延迟或提前）最长 1 秒。在这种平移操作中，会将 SUBFS[14:0]值加到同步预分频器计数器 SS[15:0]中：这将使时钟产生延迟。如果同时将 ADD1S 位置 1，则会增加一秒，与此同时减去的时间为零点几秒，因此将使时钟提前。

注意：初始化平移操作前，用户必须检查确认 SS[15] = 0，以确保不会发生上溢。

对 RTC_SHIFTR 寄存器执行写操作以启动平移操作时，硬件会将 SHPF 标志置 1 以指示平移操作挂起。完成平移操作时，硬件会将该位清零。

17.3.10 RTC 精密数字校准

RTC 频率可采用约 0.954 ppm 的分辨率进行数字校准，校准范围为 -487.1 ppm 到 +488.5 ppm。使用一系列微调（增加和/或减少单独的 RTCCLK 脉冲）进行频率校正。这些微调的分布非常均匀，因此 RTC 的校准效果相当好，即使在短时间内持续观察也是如此。

当输入频率为 32768 Hz 时，精密数字校准的周期约为 220 个 RTCCLK 脉冲或 32 秒。此周期由一个通过 RTCCLK 提供时钟信号的 20 位计数器 calib_cnt[19:0]维持。

精密数字校准寄存器（RTC_CALR）可指定 32 秒周期内要减少的 RTCCLK 时钟周期数：

- 将位 CALM[0]置 1 时，32 秒周期内将只屏蔽 1 个脉冲
- 将 CALM[1]置 1 时，将再屏蔽两个周期
- 将 CALM[2]置 1 时，将再屏蔽四个周期
- 依此类推，将 CALM[8]置 1 时，将减少 256 个时钟

注意： *CALM[8:0]（RTC_CALR）可指定 32 秒周期内要减少的 RTCCLK 脉冲数。将位 CALM[0]置 1 时，32 秒周期内将只屏蔽一个脉冲（当 cal_cnt[19:0] = 0x80000 时）；将 CALM[1]置 1 时，将再屏蔽两个周期（当 cal_cnt = 0x40000 和 0xC0000 时）；将 CALM[2] 置 1 时，将再屏蔽四个周期（当 cal_cnt = 0x20000/0x60000/0xA0000/0xE0000 时）；依此类推，将 CALM[8]置 1 时，将屏蔽 256 个时钟（当 cal_cnt = 0xXX800 时）。*

CALM 能以高精度将 RTC 频率降低至多 487.1 ppm（511/(220+511）），CALP 位则可用于将频率提高 488.5 ppm（512/(220-512））。将 CALP 置“1”，可每隔 211 个 RTCCLK 周期有效插入一个额外的 RTCCLK 脉冲，这意味着每 32 秒周期可增加 512 个时钟。

与 CALM 和 CALP 配合使用时，可在 32 秒周期内增加一个范围为 -511 到 +512 RTCCLK 周期的偏移，对应的校准范围为 -487.1 ppm 到 +488.5 ppm，分辨率约为 0.954 ppm。

若输入频率（FRTCCLK）已知，可通过以下公式计算有效校准频率（FCAL）：

$F_{CAL} = F_{RTCCLK} \times [\,1 + (CALP \times 512 - CALM) / (2^{20} + CALM - CALP \times 512)\,]$

当 PREDIV_A < 3 时校准

当异步预分频器值（RTC_PRER 寄存器中的 PREDIV_A 位）小于 3 时，CALP 位不能置为 1。如果 CALP 已置 1 并且 PREDIV_A 位的值小于 3，则会忽略 CALP，即假定 CALP 等于 0 而执行校准。

要在 PREDIV_A 小于 3 的条件下执行校准，应降低同步预分频器值（PREDIV_S）以便每秒内可加速 8 个 RTCCLK 时钟周期，这意味着每 32 秒可增加 256 个时钟周期。因此，仅使用 CALM 位，可在每 32 秒内有效增加 255 到 256 个时钟脉冲（对应的校准范围为 243.3 ppm 到 244.1 ppm）。

在标称 RTCCLK 频率 32768 Hz 下，当 PREDIV_A 等于 1 时（分频系数为 2），应将 PREDIV_S 设置为 16379 而不是 16383（少 4）。唯一相关的其它情况是，当 PREDIV_A 等于 0 时，应将 PREDIV_S 设置为 32759 而不是 32767（少 8）。

如果 PREDIV_S 以这种方式减少，则给出校准输入时钟有效频率的公式如下：

$F_{CAL} = F_{RTCCLK} \times [\ 1 + (256 - CALM) / (2^{20} + CALM - 256)]$

在这种情况下，如果 RTCCLK 恰好为 32768.00 Hz，则当 CALM[7:0]等于 0x100 时（CALM 范围的中值），说明设置正确。

注意： *当 PREDIV_A=2（异步预分频器 3 分频）时似乎不大可能有用，除非标称输入频率是 3 的倍数。例如，如果 RTCCLK 标称值为 98304 Hz（32768 Hz x 3），将 PREDIV_S 设置为 32759 而不是 32767（少 8）可使以上公式有效。*

验证 RTC 校准

通过测量 R_{TCCLK} 的精确频率，计算正确的 CALM 和 CALP 值以确保 RTC 精度。此外，还为应用提供了一个可选的 1 Hz 输出，用来测量和验证 RTC 精度。

如果在有限的间隔内测量 RTC 的精确频率，则会导致测量期间产生最多 2 个 R_{TCCLK} 时钟周期的测量误差，具体取决于数字校准周期与测量周期的对齐方式。

但是，如果测量周期与校准周期的长度相同，则可以消除此测量误差。在这种情况下，观测到的唯一误差是由数字校准的分辨率导致的误差。

- 默认情况下，校准周期为 32 秒。

在此模式下，测量整个 32 秒内 1 Hz 输出的精度，可确保测量误差在 0.477 ppm 内（32 秒内为 0.5 个 RTCCLK 周期，受校准分辨率限制）。

- 可将 RTC_CALR 寄存器的 CALW16 位置 1，以强制 16 秒的校准周期。

此时，可在 16 秒内测量 RTC 精度，产生的最大误差为 0.954 ppm（16 秒内为 0.5 个 R_{TCCLK} 周期）。但是，由于校准分辨率降低，长期的 RTC 精度也会降到 0.954 ppm：将 CALW16 置 1 时，CALM[0]位将始终保持为 0。

- 可将 RTC_CALR 寄存器的 CALW8 位置 1，以强制 8 秒的校准周期。

此时，可在 8 秒内测量 RTC 精度，产生的最大误差为 1.907 ppm（8 秒内为 0.5 个 R_{TCCLK} 周期）。长期的 RTC 精度也会降到 1.907 ppm：将 CALW8 置 1 时，CALM[1:0]位将始终保持为 00。

实时重校准

当 RTC_ISR/INITF=0 时，可实时更新校准寄存器（RTC_CALR），具体步骤如下：

1. 轮询 RTC_ISR/RECALPF（重新校准挂起标志）。
2. 如果该标志为 0，则可以根据需要向 RTC_CALR 写入新值。随后 RECALPF 位会被自动置为 1。
3. 新校准设置将在对 RTC_CALR 执行写操作之后的三个 ck_apre 周期内生效。

注意： *然后，RECALPF 自动变为“0”。请注意，在对 RTC_CALR 执行写操作后，RECALPF 可在 4 个 ck_apre 周期 + 2 个系统时钟周期内保持为“1”。在初始化模式（RTC_ISR/INIT=1）下，RECALPF 可无限期地保持为“1”。*

17.3.11 校准时钟输出

将 RTC_CR 寄存器中的 COE 位置 1 时，会在 RTC_CALIB 器件输出上提供一个参考时钟。

注意： *如果 I/O 被设定为相关的 AFx 模式，则此 RTC_CALIB 信息是 RTC_OUT I/O 信号上的输出，参见第 5 节“电源控制器（PWRC）”。*

如果 RTC_CR 寄存器中的 COSEL 位复位且 PREDIV_A = 0x7F，则 RTC_CALIB 频率为 $f_{RTCCLK}/64$。这相当于 R_{TCCLK} 频率为 32.768 kHz 时，512 Hz 的校准输出。RTC_CALIB 占空比是不规则的：下降沿上存在轻微抖动。因此推荐使用上升沿。

当 COSEL 置位且“PREDIV_S+1”为 256 的非零倍数（即：PREDIV_S[7:0] = 0xFF）时，RTC_CALIB 频率为 f_{RTCCLK}/（256*（PREDIV_A+1））。这相当于 RTCCLK 频率为 32.768 kHz 时，1 Hz 的校准输出，其中预分频器为默认值（PREDIV_A = Ox7F、PREDIV_S = 0xFF）。

17.3.12 闹钟输出

RTC_CR 寄存器中的 OSEL[1:0]控制位用于激活闹钟复用功能输出 RTC_ALARM，以及选择输出的功能。这些功能可反映 RTC_ISR 寄存器中相应标志的内容。输出的极性由 RTC_CR 中的 POL 控制位确定，这样当 POL 置 1 时会输出选定标志位的相反值。

注意： *如果 I/O 被设定为相关的 AFx 模式，则 RTC_ALARM 是 RTC_OUT I/O 信号上的输出，参见第 5 节“电源控制器（PWRC）”。*

注意： *在使能 RTC_ALARM 输出后，其优先级高于 RTC_CALIB（COE 位必须保持清零，这意味着 RTC_OUT I/O 输出 RTC_ALARM）。*

17.4 RTC 低功耗模式

RTC 能够在 DEEPSTOP 模式下运行并生成唤醒事件，以便通过 RTC 闹铃和 RTC 唤醒根本原因唤醒器件。

注意： *唤醒后，软件必须将 RTC 中的 RTC_ISR.WUTF 标志清零，否则它会阻止器件再次进入低功耗模式。PWRC 模块仅在其自己的唤醒标志寄存器中反映 RTC 唤醒信号。*

17.5 RTC 中断

将所有 RTC 中断组合并连接到 NVIC 控制器。请参见第 2.3.2 节“中断”。要使能 RTC 闹钟中断，需按照以下顺序操作：

1. 配置并使能 NVIC 中的 RTC_ALARM IRQ 通道
2. 配置 RTC 以生成 RTC 闹铃（闹铃 A）

要使能唤醒定时器中断，需按照以下顺序操作：

1. 配置并使能 NVIC 中的 RTC IRQ 通道
2. 配置 RTC 以检测 WUT 事件

17.6 RTC 寄存器

请参见参考手册的第 1.3 节“缩略语”获取寄存器说明中所用缩略语的列表。外设寄存器可按字（32 位）进行访问。

17.6.1 RTC 时间寄存器（RTC_TR）

RTC_TR 是日历时间影子寄存器。只能在初始化模式下对该寄存器执行写操作。请参见第 17.3.7 节“读取日历”。

此寄存器受写保护。第 17.3.6 节“RTC 初始化和配置”中描述了写访问流程。

偏移地址：0x00

上电复位值：0x0000 0000

系统复位：当 BYPSHAD = 0 时为 0x0000 0000。当 BYPSHAD = 1 时不受影响。

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]			
									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，必须保持复位值。
位 22	**PM**：AM/PM 符号 0：AM 或 24 小时制 1：PM
位 21:20	**HT[1:0]**：小时的十位（BCD 格式）
位 16:16	**HU[3:0]**：小时的个位（BCD 格式）
位 15	保留，必须保持复位值。
位 14:12	**MNT[2:0]**：分钟的十位（BCD 格式）
位 11:8	**MNU[3:0]**：分钟的个位（BCD 格式）
位 7	保留，必须保持复位值。
位 6:4	**ST[2:0]**：秒的十位（BCD 格式）
位 3:0	**SU[3:0]**：秒的个位（BCD 格式）

17.6.2 RTC 日期寄存器（RTC_DR）

RTC_TR 是日历时间影子寄存器。只能在初始化模式下对该寄存器执行写操作。请参见第 17.3.7 节“读取日历”。

此寄存器受写保护。第 17.3.6 节“RTC 初始化和配置”中描述了写访问流程。

偏移地址：0x04

上电复位值：0x0000 2101

系统复位：当 BYPSHAD = 0 时为 0x0000 2101。当 BYPSHAD = 1 时不受影响。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	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，必须保持复位值。
位 23:20	**YT[3:0]**：年份的十位（BCD 格式）。
位 19:16	**YU[3:0]**：年份的个位（BCD 格式）。
位 15:13	**WDU[2:0]**：星期几的个位。 000：禁止 001：星期一 **…** 111：星期日
位 12	**MT**：月份的十位（BCD 格式）。
位 11:8	**MU**：月份的个位（BCD 格式）。
位 7:6	保留，必须保持复位值。
位 5:4	**DT[1:0]**：日期的十位（BCD 格式）。
位 3:0	**DU[3:0]**：日期的个位（BCD 格式）。

17.6.3 RTC 控制寄存器（RTC_CR）

偏移地址：0x08

上电复位值：0x0000 0000

系统复位：不受影响

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
Res.	WUTIE		ALRAIE	Res.	WUTE		ALRAE	Res.	FMT	BYPS HAD	Res.	Res.	WUCKSEL[2:0]		
	rw		rw		rw		rw		rw	rw			rw	rw	rw

位 31:25	保留，必须保持复位值。
位 23	**COE**：校准输出使能。 该位使能 RTC_CALIB 输出。 0：禁止校准输出 1：使能校准输出
位 22:21	**OSEL[1:0]**：输出选择。 这些位用于选择要连接到 RTC_OUT 输出的标志。 00：禁止输出 01：使能闹钟 A 输出

	10： 11：使能唤醒输出
位 20	**POL**：输出极性。 该位用于配置 RTC_ALARM 输出的极性。 0：当 ALRAF/WUTF 置 1 时（取决于 OSEL[1:0]），该引脚为高电平 1：当 ALRAF/ALRBF/WUTF 置 1 时（取决于 OSEL[1:0]），该引脚为高电平
位 19	**COSEL**：校准输出选择。 当 COE=1 时，该位可选择 RTC_CALIB 上输出的信号。 0：校准输出为 512 Hz 1：校准输出为 1 Hz 在 RTCCLK 为 32.768 kHz 且预分频器为其默认值（PREDIV_A=127 且 PREDIV_S=255）的条件下，这些频率有效。请参见第 17.3.11 节“校准时钟输出”。
位 18	**BKP**：备用 用户可对此位执行写操作以记录是否已对夏令时进行更改。
位 17	**SUB1H**：减少 1 小时（冬季时间更改）。 当该位在初始化模式以外的模式下置 1 时，如果当前小时不是 0，则日历时间将减少 1 小时。此位始终读为 0。 当前小时为 0 时，置位此位没有任何作用。 0：无操作 1：将当前时间减少 1 小时。这可用于冬季时间更改。
位 16	**ADD1H**：增加 1 小时（夏季时间更改）。 当该位在初始化模式以外的模式下置 1 时，日历时间将增加 1 小时。此位始终读为 0。 0：无操作 1：将当前时间增加 1 小时。这可用于夏季时间更改。
位 15	保留，必须保持复位值。
位 14	**WUTIE**：唤醒定时器中断使能。 0：禁止唤醒定时器中断 1：使能唤醒定时器中断
位 12	**ALRAIE**：闹钟 A 中断使能 0：禁止闹钟 A 中断 1：使能闹钟 A 中断
位 11	保留，必须保持复位值。
位 10	**WUTE**：唤醒定时器使能。 0：禁止唤醒定时器 1：使能唤醒定时器
位 8	**ALRAE**：使能闹钟 A。 0：禁止闹钟 A 1：使能闹钟 A
位 7	保留，必须保持复位值。
位 6	**FMT**：小时格式。 0：24 小时/天格式 1：AM/PM 小时格式
位 5	**BYPSHAD**：旁路影子寄存器。

	0：日历值（从 RTC_SSR、RTC_TR 和 RTC_DR 读取时）取自影子寄存器，该影子寄存器每两个 RTCCLK 周期更新一次 1：日历值（从 RTC_SSR、RTC_TR 和 RTC_DR 读取时）直接取自日历计数器 注意：如果 APB1 时钟的频率低于 7 倍的 RTCCLK 频率，则必须将 BYPSHAD 置“1”。
位 4	保留，必须保持复位值。
位 3	保留，必须保持复位值。
位 2:0	**WUCKSEL[2:0]**：唤醒时钟选择 000：选择 RTC/16 时钟。 001：选择 RTC/8 时钟 010：选择 RTC/4 时钟 011：选择 RTC/2 时钟 10x：选择 ck_spre 时钟（通常为 1 Hz） 11x：选择 ck_spre 时钟（通常为 1 Hz）并将 WUT 计数器值增加 216（见下面的注释）

注意：

1. *只能在初始化模式下（RTC_ISR/INITF = 1）对该寄存器的位 7、6 和 4 执行写操作。*
2. *WUT = 唤醒单位计数器值。当 WUCKSEL[2:1 = 11]时，WUT =（0x0000 至 0xFFFF）+ 0x10000（增加的值）。*
3. *只能在 RTC_CR WUTE 位 = 0 且 RTC_ISR WUTWF 位 = 1 时对该寄存器的位 2 到 0 执行写操作。*
4. *建议不要在日历小时递增时更改小时，因为这样做会屏蔽日历小时的增量。*
5. *ADD1H 和 SUB1H 的更改在下一秒生效。*
6. *此寄存器受写保护。第 17.6 节“RTC 寄存器”中描述了写访问流程。*

17.6.4 RTC 初始化和状态寄存器（RTC_ISR）

此寄存器受写保护（RTC_ISR[-17:8]位除外）。

第 17.6.3 节“RTC 控制寄存器（RTC_CR）”中描述了写访问流程。

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.	RECAL PF
															r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	WUTF		ALRAF	INIT	INITF	RSF	INITS	SHPF	WUTWF		ALRAW F
					rc_w0		rc_w0	rw	r	rc_w0	r	rc_w0	r		r

位 31:18	保留，必须保持复位值。
位 16	**RECALPF**：重新校准挂起标志。 当软件对 RTC_CALR 寄存器执行写操作时，RECALPF 状态标志将自动置“1”，指示 RTC_CALR 寄存器已屏蔽。当采用新的校准设置时，该位恢复为“0”。
位 15	保留，必须保持复位值。
位 14	保留，必须保持复位值。
位 13	保留，必须保持复位值。
位 12	保留，必须保持复位值。
位 11	保留，必须保持复位值。
位 10	**WUTF**：唤醒定时器标志 当唤醒自动重载计数器达到 0 时，此标志由硬件置位。该标志由软件写零清除。

	软件必须在 WUTF 再次置 1 的 1.5 个 R_{TCCLK} 周期之前将该标志清零。
位 8	**ALRAF**：闹钟 A 标志。 当时间/日期寄存器（RTC_TR 和 RTC_DR）与闹钟 A 寄存器（RTC_ALRMAR）匹配时，由硬件将该标志置 1。 该标志由软件写零清除。
位 7	**INIT**：初始化模式。 0：自激模式 1：初始化模式，用于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）。计数器停止计数，当 INIT 被复位后，计数器从新值开始计数。
位 6	**INITF**：初始化标志。 当此位置 1 时，RTC 处于初始化状态，此时可更新事件、日期和预分频器寄存器。 0：不允许更新日历寄存器 1：允许更新日历寄存器
位 5	**RSF**：寄存器同步标志。 每次将日历寄存器的值复制到影子寄存器（RTC_SSRx、RTC_TRx 和 RTC_DRx）时，都会由硬件将此位置 1。在初始化模式下、平移操作挂起时（SHPF=1）或在旁路影子寄存器模式（BYPSHAD=1）下，该位由硬件清零。该位还可由软件清零。 在初始化模式下，该位可由软件或硬件清零。 0：日历影子寄存器尚未同步 1：日历影子寄存器已同步
位 4	**INITS**：初始化状态标志。 当日历年份字段不为 0 时（上电复位状态），由硬件将该位置 1。 0：日历尚未初始化 1：日历已经初始化
位 3	**SHPF**：平移操作挂起。 0：没有平移操作挂起 1：某个平移操作挂起 只要通过对 RTC_SHIFTR 寄存器执行写操作来启动平移操作，此标志便由硬件置 1。执行完相应的平移操作后，此标志由硬件清零。 对 SHPF 位执行写入操作不起作用。
位 2	**WUTWF**：唤醒定时器写入标志。 在 RTC_CR 寄存器中的 WUTE 位置 0 后，当唤醒定时器值可更改时，由硬件将该位置 1。 0：不允许唤醒定时器配置更新 1：允许唤醒定时器配置更新
位 0	**ALRAWF**：闹钟 A 写标志。 在 RTC_CR 寄存器中的 ALRAE 位置 0 后，当闹钟 A 的值可更改时，由硬件将该位置 1。 该位在初始化模式下由硬件清零。 0：不允许更新闹钟 A 1：允许更新闹钟 A

注意：　在将 ALRAF 和 WUTF 位设定为 0 两个 APB 时钟周期后，这两个位被清零。

17.6.5 RTC 预分频器寄存器（RTC_PRER）

只能在初始化模式下对该寄存器执行写操作。必须通过两次独立的写访问执行初始化。

此寄存器受写保护。第 17.6 节“RTC 寄存器”中描述了写访问流程。

偏移地址：0x10

上电复位值：0x007F 00FF

系统复位：不受影响

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，必须保持复位值。
位 22:16	**PREDIV_A[6:0]**：异步预分频系数。下面是异步分频系数的公式： ck_apre 频率 = RTCCLK 频率/（PREDIV_A+1）
位 15	保留，必须保持复位值。
位 14:0	**PREDIV_S[14:0]**：同步预分频系数。下面是同步分频系数的公式： ck_spre 频率 = ck_apre 频率/（PREDIV_S+1）

17.6.6 RTC 唤醒定时器寄存器（RTC_WUTR）

仅当 RTC_ISR 中的 WUTWF 置 1 时才可对该寄存器执行写操作。

此寄存器受写保护。第 17.6 节“RTC 寄存器”中描述了写访问流程。

偏移地址：0x14

上电复位值：0x0000 FFFF

系统复位：不受影响

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
WUT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16	保留，必须保持复位值。
位 15:0	**WUT[15:0]**：唤醒自动重载值位。 当唤醒定时器使能（WUTE 置 1）时，每个（WUT[15:0]+1）ck_wut 周期 WUTF 标志都置 1。ck_wut 周期通过 RTC_CR 寄存器的 WUCKSEL[2:0]位进行选择。 当 WUCKSEL[2] = 1 时，唤醒定时器变为 17 位，WUCKSEL[1]等效为 WUT[16]，即要重载到定时器的最高有效位。 WUTF 第一次置 1 发生在 WUTE 置 1 之后（WUT+1）个 ck_wut 周期。禁止在 WUCKSEL[2:0]=011(RTCCLK/2）时将 WUT[15:0]设置为 0x0000。

17.6.7 RTC 闹钟 A 寄存器（RTC_ALRMAR）

仅当 RTC_ISR 中的 ALRAWF 置 1 时或在初始化模式下，才可以对该寄存器执行写操作。

此寄存器受写保护。第 17.6 节“RTC 寄存器”中描述了写访问流程。

偏移地址：0x1C

上电复位值：0x0000 0000

系统复位：不受影响

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**：闹钟 A 日期掩码 0：如果日期/日匹配，则闹钟 A 置 1 1：日期/日在闹钟 A 的比较中无意义
位 30	**WDSEL**：星期几选择 0：DU[3:0] 代表日期的个位 1：DU[3:0] 代表星期几。 DT[1:0] 无意义
位 29:28	**DT[1:0]**：日期的十位（BCD 格式）
位 27:24	**DU[3:0]**：日期个位或日（BCD 格式）
位 23	**MSK3**：闹钟 A 小时掩码 0：如果小时匹配，则闹钟 A 置 1 1：小时在闹钟 A 的比较中无意义
位 22	**PM**：AM/PM 符号 0：AM 或 24 小时制 1：PM
位 21:20	**HT[1:0]**：小时的十位（BCD 格式）
位 19:16	**HU[3:0]**：小时的个位（BCD 格式）
位 15	**MSK2**：闹钟 A 分钟掩码 0：如果分钟匹配，则闹钟 A 置 1 1：分钟在闹钟 A 的比较中无意义
位 14:12	**MNT[2:0]**：分钟的十位（BCD 格式）
位 11:8	**MNU[3:0]**：分钟的个位（BCD 格式）
位 7	**MSK1**：闹钟 A 秒掩码 0：如果秒匹配，则闹钟 A 置 1 1：秒在闹钟 A 的比较中无意义
位 6:4	**ST[2:0]**：秒的十位（BCD 格式）
位 3:0	**SU[3:0]**：秒的个位（BCD 格式）

17.6.8 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[7:0]							
								w	w	w	w	w	w	w	w

位 31:8	保留，必须保持复位值。
位 7:0	**KEY[7:0]**：写保护密钥。 可通过软件对该字节执行写操作。 读取该字节时，始终返回 0x00。 请参见第 17.6 节“RTC 寄存器”获取关于如何解锁 RTC 寄存器写保护的说明。

17.6.9 RTC 亚秒寄存器（RTC_SSR）

偏移地址：0x28

上电复位值：0x0000 0000

系统复位：当 BYPSHAD = 0 时为 0x0000 0000。当 BYPSHAD = 1 时不受影响。

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，必须保持复位值。
位 15:0	**SS**：亚秒值。 SS[15:0] 是同步预分频器计数器的值。此亚秒值可根据以下公式得出： 亚秒值 =（PREDIV_S - SS）/（PREDIV_S + 1） 注意：仅当执行平移操作之后，SS 才能大于 PREDIV_S。在这种情况下，正确的时间/日期比 RTC_TR/RTC_DR 所指示的时间/日期慢一秒钟。

17.6.10 RTC 平移控制寄存器（RTC_SHIFTR）

此寄存器受写保护。第 17.6.8 节“RTC 写保护寄存器（RTC_WPR）”中描述了写访问流程。

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 1 秒。 0：无操作 1：对时钟/日历增加一秒钟 此位为只写位且始终读为 0。当平移操作挂起（RTC_ISR 中的 SHPF=1）时，对该位执行写操作无作用。 此函数应与 SUBFS 配合使用（请参见下文介绍），以便有效地向原子操作机制的时钟添加亚秒值。
位 31:15	保留，必须保持复位值。
位 14:0	**SUBFS**：减少亚秒值。 这些位为只“写”位且始终“读”为 0。当平移操作挂起（RTC_ISR 中的 SHPF=1）时，对该位执行写操作无作用。 写入 SUBFS 的值将加到同步预分频器计数器中。由于该计数器递减计数，此操作可有效地从时钟减去（延迟）以下时间： 延迟（秒）= SUBFS /（PREDIV_S + 1） 当 ADD1S 函数与 SUBFS 结合使用时，可有效地将亚秒值增加到时钟（提前时钟），使时钟提前以下时间： 提前（秒）=（1 -（SUBFS /（PREDIV_S + 1））） 。 注意：对 SUBFS 执行写操作将使 RSF 清零。软件随后会等待至 RSF=1 以确定影子寄存器已更新为平移后的时间。 请参见第 17.3.9 节“RTC 同步”。

17.6.11 RTC 校准寄存器（RTC_CALR）

此寄存器受写保护。第 17.6.8 节“RTC 写保护寄存器（RTC_WPR）”中描述了写访问流程。

偏移地址：0x3C

上电复位值：0x0000 0000

系统复位：不受影响

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，必须保持复位值。
位 15	CALP：将 RTC 的频率增加 488.5 ppm 0：不增加 RTCCLK 脉冲 1：每隔 211 个脉冲有效插入一个 RTCCLK 脉冲（频率增加 488.5 ppm）。 此功能应与 CALM 结合使用，后者在高分辨率下会降低日历的频率。如果输入频率为 32768 Hz，则在 32 秒窗口中增加的 RTCCLK 脉冲数按如下公式计算：

	（512 * CALP）- CALM。 请参见第 17.3.10 节“RTC 精密数字校准”。
位 14	**CALW8**：使用 8 秒校准周期。 当 CALW8 置“1”时，选择 8 秒校准周期。 注意：CALW8=“1”时，CALM[1:0]将始终保持为“00”。请参见 17.3.10 节“RTC 精密数字校准”。
位 13	**CALW16**：使用 16 秒校准周期。 当 CALW16 置“1”时，选择 16 秒校准周期。如果 CALW8=1，则必须将该位置“1”。 注意：当 CALW16=“1”时，CALM[0]将始终保持为“0”。请参见 17.3.10 节“RTC 精密数字校准”。
位 12:9	保留，必须保持复位值。
位 8:0	**CALM[8:0]**：校准负值 在 220 个 RTCCLK 脉冲内屏蔽 CALM 个脉冲（如果输入频率为 32768 Hz，则为 32 秒）来降低日历的频率。其分辨率为 0.9537 ppm。 要提高日历的频率，则应将此功能与 CALP 结合使用。请参见第 17.3.10 节“RTC 精密数字校准”。

17.6.12 RTC 闹钟 A 亚秒寄存器（RTC_ALRMASSR）

仅当 RTC_CR 中的 ALRAE 复位时或在初始化模式下，才可以对该寄存器执行写操作。

此寄存器受写保护。第 17.6.8 节“RTC 写保护寄存器（RTC_WPR）”中描述了写访问流程。

偏移地址：0x44

上电复位值：0x0000 0000

系统复位：不受影响

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，必须保持复位值。
位 27:24	MASKSS[3:0]：屏蔽从此位开始的最高有效位 0：不对闹钟 A 的亚秒进行比较。当秒单元递增时设置闹钟（假定其余字段均匹配）。 1：SS[14:1] 在闹钟 A 的比较中无意义。仅比较 SS[0]。 2：SS[14:2] 在闹钟 A 的比较中无意义。仅比较 SS[1:0]。 3：SS[14:3] 在闹钟 A 的比较中无意义。仅比较 SS[2:0]。 ... 12：SS[14:12] 在闹钟 A 的比较中无意义 比较 SS[11:0] 13：SS[14:13] 在闹钟 A 的比较中无意义。比较 SS[12:0]。 14：SS[14] 在闹钟 A 的比较中无意义。比较 SS[13:0]。 15：所有 15 个 SS 位均进行比较，并且必须全部匹配才能激活闹钟 同步计数器的上溢位（位 15）从不进行比较。仅当执行平移操作之后，此位才不为 0。

位 23:15	保留，必须保持复位值。
位 14:0	SS[14:0]：亚秒值 该值与同步预分频器计数器的内容进行比较以确定是否要激活闹钟 A。仅比较位 0 到 MASKSS-1。

17.6.13 RTC 备份寄存器（RTC_BKPxR）

偏移地址：0x50 至 0x54

上电复位值：0x0000 0000

系统复位：不受影响

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] 应用可向/从这些寄存器写入/读取数据。 它们通过 VDD12o 供电，因此在 DEEPSTOP 模式下得到保留。 应用可向/从这些寄存器写入/读取数据。该寄存器只在 PORESETn 时复位。

17.7 RTC 寄存器映射

表 52. RTC 寄存器映射和复位值

偏移	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
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]			
	复位值										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]			
	复位值									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	COSEL	BKP	SUB1H	ADD1H	Res.	WUTIE		ALRAIE	Res	WUTE		ALRAE	Res.	FMT	BYPSHAD	Res.	Res.	WUCKSE L[2:0]		
	复位值									0	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.	RECALPF	Res.	Res	Res.	Res.	Res.	WUTF		ALRAF	INIT	INITF	RSF	INITS	SHPF	WUT WF		ALRAWF
	复位值									0							0						0		0	0	0	0	0	0	1		1

偏移	0x28		0x24		0x1C		0x14		0x10																							
寄存器	RTC_SSR		RTC_WPR		RTC_ALRMAR		RTC_WUTR		RTC_PRER																							
	复位值		复位值		复位值		复位值		复位值																							
31	Res.		Res.		MSK4		Res.		Res.																							
30	Res.		Res.		WDSEL		Res.		Res.																							
29	Res.		Res.		DT [1:0]		Res.		Res.																							
28	Res.		Res.				Res.		Res.																							
27	Res.		Res.		DU[3:0]		Res.		Res.																							
26	Res.		Res.				Res.		Res.																							
25	Res.		Res.				Res.		Res.																							
24	Res.		Res.				Res.		Res.																							
23	Res.		Res.		MSK3		Res.		Res.																							
22	Res.		Res.		PM		Res.		PREDIV_A[6:0]																							
21	Res.		Res.		HT [1:0]		Res.																									
20	Res.		Res.				Res.																									
19	Res.		Res.		HU[3:0]		Res.																									
18	Res.		Res.				Res.																									
17	Res.		Res.				Res.																									
16	Res.		Res.				Res.																									
15	Res.		Res.		MSK2		WUT[15:0]		Res.																							
14	Res.		Res.		MNT[2:0]				PREDIV_S[14:0]																							
13	Res.		Res.																													
12	Res.		Res.																													
11	Res.		Res.		MNU[3:0]																											
10	Res.		Res.																													
9	Res.		Res.																													
8	Res.		Res.																													
7	SS[15:0]		KEY[7:0]		MSK1																											
6					ST[2:0]																											
5																																
4																																
3					SU[3:0]																											
2																																
1																																
0																																
复位值	0		0		0		1		1																							

偏移	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
0x2C	RTC_SHIFTR	ADD1S	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBFS[14:0]															
	复位值	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.	Res.	CALM[8:0]							
	复位值																	0	0	0					0	0	0	0	0	0	0	0	0
0x40	保留																																
0x44	RTC_ALRMASSR	Res.	Res.	Res.	Res.	MASKSS[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SS[14:0]														
	复位值					0	0	0	0									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x50-0x54	RTC_BKP0R	BKP[31:0]																															
	复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

偏移	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
0x50-0x54	RTC_BKP1R	BKP[31:0]																															
	复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

请参见第 2.2.2 节“存储器映射和寄存器边界地址”了解寄存器边界地址。

18 独立看门狗（IWDG）

18.1 引言

此器件具有一个嵌入式看门狗外设，具有安全性高、定时准确及使用灵活的优点。此独立看门狗外设可用于检测并解决由软件错误导致的故障，并在计数器达到给定的超时值时触发系统复位。

独立看门狗（IWDG）由其专用低速时钟（LSI）驱动，因此即便在主时钟发生故障时仍然保持工作状态。

IWDG 最适合应用于那些需要看门狗作为一个在主程序之外，能够完全独立工作，并且对时间精度要求较低的场合。

18.2 IWDG 主要特性

- 自由运行递减计数器
- 时钟由独立 RC 振荡器提供（可在待机和停止模式下运行）
- 复位条件
  - 在递减计数器值小于 000h 时复位（若激活了看门狗）
  - 在递减计数器在窗口外重载时复位（若激活了看门狗）

18.3 IWDG 功能说明

图 82. 独立看门狗框图显示了独立看门狗模块的功能块。

当通过对关键字寄存器（IWDG_KR）写入值 0x0000 CCCC 启动独立看门狗时，计数器开始从复位值 0xFFF 递减计数。当计数器计数到终值（0x000）时会产生一个复位信号（IWDG 复位）。

任何时候将密钥值 0x0000 AAAA 写到 IWDG_KR 寄存器中，IWDG_RLR 的值就会被重载到计数器，从而避免产生看门狗复位。

18.3.1 窗口选项

通过在 IWDG_WINR 寄存器中设置合适的窗口，IWDG 也可以用作窗口看门狗。

当计数器值大于窗口寄存器（IWDG_WINR）中存储的值时，如果执行重载操作，则会产生复位。

IWDG_WINR 的默认值为 0x0000 0FFF，因此，如果不更新此默认值，将禁止窗口选项。

窗口值一经更改，便执行重载操作，以便将递减计数器复位为 IWDG_RLR 值，并方便计算周期数以生成下一次重载。

使能窗口选项时配置 IWDG

1. 通过在 IWDG_KR 寄存器中写入 0x0000 CCCC 来使能 IWDG。
2. 通过在 IWDG_KR 寄存器中写入 0x0000 5555 来使能寄存器访问。
3. 通过将 IWDG_PR 编程为 0~7 中的数值来配置 IWDG 预分频器。
4. 写重载寄存器（IWDG_RLR）。
5. 等待计数器更新（IWDG_SR = 0x00000000）。
6. 写入窗口寄存器 IWDG_WINR。这将自动刷新计数器值 IWDG_RLR。

注意： *当 IWDG_SR 置为 0x0000 0000 时，写入窗口值可通过 RLR 刷新计数器值。*

禁止窗口选项时配置 IWDG。

不使用窗口选项时，可按以下步骤配置 IWDG：

1. 通过在 IWDG_KR 寄存器中写入 0x0000 5555 来使能寄存器访问。
2. 通过将 IWDG_PR 编程为 0~7 中的数值来配置 IWDG 预分频器。
3. 写重载寄存器（IWDG_RLR）。
4. 等待计数器更新（IWDG_SR = 0x00000000）。
5. 用 IWDG_RLR（IWDG_KR = 0x0000AAAA）刷新计数器值。
6. 通过在 IWDG_KR 中写入 0x0000 CCCC 来使能 IWDG。

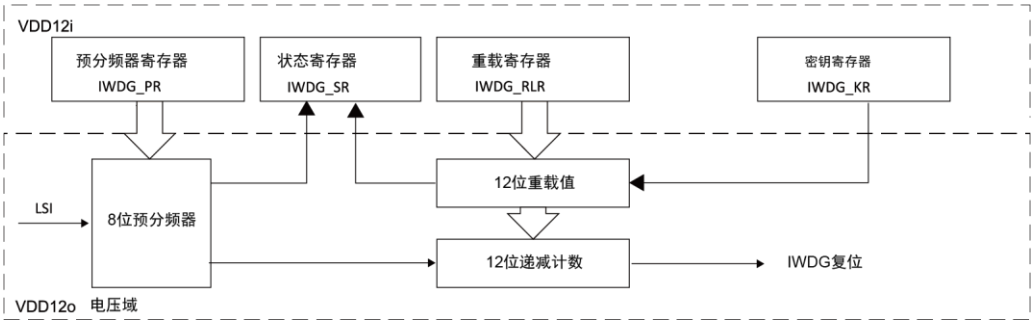
18.3.2 寄存器访问保护

IWDG_PR、IWDG_RLR 和 IWDG_WINR 寄存器具有写访问保护。若要修改寄存器，必须首先对 IWDG_KR 寄存器写入代码 0x0000 5555。将其他值写入该寄存器会打断序列且寄存器访问会重新受到保护。这意味着重装载操作（即写入 0x0000 AAAA）也会启动写保护功能。状态寄存器指示预分频值、递减计数器重载值或窗口值是否正在被更新。

18.3.3 调试模式

BlueNRG-LP 中没有实现特定的调试模式。即使调试器暂停 CPU，定时器仍继续计数。

图 82. 独立看门狗框图



注意： *在 VDD12o 电源域中实现了看门狗，它在 DEEPSTOP 模式下仍然工作。*

18.4 IWDG 寄存器

请参见“表 2. 缩略语”获取寄存器说明中所用缩略语的列表。外设寄存器可支持半字（16 位）或字（32 位）访问。

18.4.1 密钥寄存器（IWDG_KR）

偏移地址：0x00

复位值：0x0000 0000（通过待机模式复位）

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.	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，必须保持复位值。
位 15:0	KEY[15:0]：密钥值（只写位，读为 0x0000）。 必须每隔一段时间便通过软件对这些位写入密钥值 0xAAAA，否则当计数器计数到 0 时，看门狗会产生复位。 写入密钥值 0x5555 使能对 IWDG_PR、IWDG_RLR 和 IWDG_WINR 寄存器的访问（参见第 18.3.2 节“寄存器访问保护”）。 写入密钥值 CCCCh 可启动看门狗（选中硬件看门狗选项的情况除外）。

18.4.2 预分频器寄存器（IWDG_PR）

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.	Res.	Res.	Res.	Res.	Res.	PR[2:0]		
													rw	rw	rw

位 31:3	保留，必须保持复位值。
位 2:0	**PR[2:0]**：预分频器。 这些位受写保护，参见第 18.3.2 节“寄存器访问保护”。通过软件设置这些位来选择计数器时钟的预分频因子。若要更改预分频器的分频系数，IWDG_SR 的 PVU 位必须为 0。 000：4 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 注意：读取该寄存器会返回 VDD12o 电压域的预分频器值。如果正在对该寄存器执行写操作，则读取的值可能不是最新的/有效的。因此，只有在 IWDG_SR 寄存器中的 PVU 位为 0 时，从寄存器读取的值才有效。

18.4.3 重载寄存器（IWDG_RLR）

偏移地址：0x08

复位值：0x0000 0FFF（待机模式时复位）

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.	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，必须保持复位值。
Bits11:0	**RL[11:0]**：看门狗计数器重载值。 这些位受写保护，参见第 18.3.2 节“寄存器访问保护”。这个值由软件设置，每次对 IWDG_KR 寄存器写入值 0xAAAA 时，这个值就会重装载到看门狗计数器中。之后，看门狗计数器便从该装载的值开始递减计数。超时周期由该值和时钟预分频器共同决定。有关超时信息，请参见数据手册。 若要更改重载值，IWDG_SR 中的 RVU 位必须为 0。 注意：读取该寄存器会返回 VDD12o 电压域的重载值。如果正在对该寄存器执行写操作，则读取的值可能不是最新的/有效的。因此，只有在 IWDG_SR 寄存器中的 RVU 位为 0 时，从寄存器读取的值才有效。

18.4.4 状态寄存器（IWDG_SR）

偏移地址：0x0C

复位值：0x0000 0000（待机模式时不复位）

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.	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，必须保持复位值。
位 2	**WVU**：看门狗计数器窗口值更新。 可通过硬件将该位置 1 以指示窗口值正在更新。当在 VDD12o 电压域下完成重载值更新操作后（需要多达 5 个 RC 40 kHz 周期），会通过硬件将该位复位。 窗口值只有在 WVU 位为 0 时才可更新。此位只有在通用“窗口”= 1 时才生成。

位 1	**RVU**：看门狗计数器重载值更新。 可通过硬件将该位置 1 以指示重载值正在更新。当在 VDD12o 电压域下完成重载值更新操作后（需要多达 5 个 RC 40 kHz 周期），会通过硬件将该位复位。 重载值只有在 RVU 位为 0 时才可更新。
位 0	**PVU**：看门狗预分频器值更新。 可通过硬件将该位置 1 以指示预分频器值正在更新。当在 VDD12o 电压域下完成预分频器值更新操作后（需要多达 5 个 RC 40 kHz 周期），会通过硬件将该位复位。 预分频器值只有在 PVU 位为 0 时才可更新。

18.4.5 窗口寄存器（IWDG_WINR）

偏移地址：0x10

复位值：0x0000 0FFF（待机模式时复位）

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.	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，必须保持复位值。
Bits11:0	**WIN[11:0]**：看门狗计数器窗口值。 这些位受写保护，参见第 18.3.2 节“寄存器访问保护”。这些位包含要与递减计数器进行比较的窗口值上限。 为防止发生复位，当递减计数器的值低于窗口寄存器值且大于 0x0 时必须重载。 若要更改重载值，IWDG_SR 中的 WVU 位必须为 0。 注意：读取该寄存器会返回 VDD12o 电压域的重载值。如果正在对该寄存器执行写操作，则读取的值可能不是有效的。因此，只有在 IWDG_SR 寄存器中的 WVU 位为 0 时，从寄存器读取的值才有效。

注意： *如果应用使用了数个重载、预分频器或窗口值，则必须等到 RVU 位复位才能更改重载值，等到 PVU 位复位才能更改预分频器值，并等到 WVU 位复位才能更改窗口值。但是，在更新预分频器和/或重载/窗口值之后，则无需等到 RVU、PVU 或 WVU 复位后再继续执行代码（进入低功耗模式时除外）。*

18.5 IWDG 寄存器映射

下表提供了 IWDG 寄存器映射和复位值。

表 53. IWDG 寄存器映射

偏移	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
0x00	**IWDG_KR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEY[15:0]															
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	**IWDG_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]		
	复位值																														0	0	0
0x08	**IWDG_RLR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RL[11:0]											
	复位值																					1	1	1	1	1	1	1	1	1	1	1	1
0x0C	**IWDG_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
	复位值	0	0	0																													
0x08	**IWDG_WINR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WIN[11:0]											
	复位值																					1	1	1	1	1	1	1	1	1	1	1	1

请参见第 2.2.2 节“存储器映射和寄存器边界地址”了解寄存器边界地址。

19 内部集成电路（I²C）接口

19.1 引言

I²C（内部集成电路）总线接口处理微控制器与串行 I²C 总线之间的通信。它提供多主机功能，并控制所有 I²C 总线特定的排序、协议、仲裁和时序。它支持标准模式（Sm）、快速模式（Fm）和超快速模式（Fm+）。

它还与 SMBus（系统管理总线）和 PMBus（电源管理总线）兼容。

可使用 DMA 来减轻 CPU 的工作量。

19.2 I²C 主要特性

- 兼容 I²C 总线规范第 03 版：
  - 从和主模式
  - 多主设备功能
  - 标准速度模式（高达 100 kHz）
  - 快速模式（高达 400 kHz）
  - 超快速模式（高达 1 MHz）
  - 7 位和 10 位寻址模式
  - 多个 7 位从设备地址（2 个地址，1 个具有可配置的掩码）
  - 所有 7 位地址应答模式
  - 广播呼叫
  - 总线上的数据建立和保持时间可软件配置
  - 方便易用的事件管理
  - 可选的时钟延长
  - 软件复位
- 带 DMA 功能的 1 字节缓冲
- 可编程模拟和数字噪声滤波器。

根据产品实现，还有以下附加特性可供使用（参见第 19.3 节“I²C 实现”）：

- 兼容 SMBus 规范第 2.0 版：
  - 具有 ACK 控制的硬件 PEC（数据包错误校验）生成和验证
  - 指令和数据应答控制
  - 支持地址解析协议（ARP）
  - 主机和器件支持
  - SMBus 报警
  - 超时和空闲条件检测
- 兼容 PMBus 第 1.1 版标准
- 独立时钟：选择独立时钟源可使 I²C 通信速度不受 PCLK 时钟频率更改的影响。

注意： *对于超快速模式，强烈建议使用支持开漏的 I²C 引脚（内置 10ns/50ns 滤波器）。*

19.3 I²C 实现

本手册描述了 I2C1 和 I2C2 中实现的特性的完整集合。

表 54. BlueNRG-LP I2C 实现

I2C 特性(1)	I2C1	I2C2
7 位寻址模式	X	X
10 位寻址模式	X	X
标准速度模式（高达 100 kbps）	X	X
快速模式（高达 400 kbps）	X	X
超快速模式，20 mA 输出驱动 I/O（高达 1 Mbit/s）	X	X
独立时钟	X	X
SMBus	X	X

1. *X = 支持*

19.4 I²C 功能说明

除了接收和发送数据之外，此接口还可以从串行格式转换为并行格式，反之亦然。中断由软件使能或禁止。此接口通过数据引脚（SDA）和时钟引脚（SCL）连接到 I²C 总线。它可以连接标准、快速或超快速模式 I²C 总线。

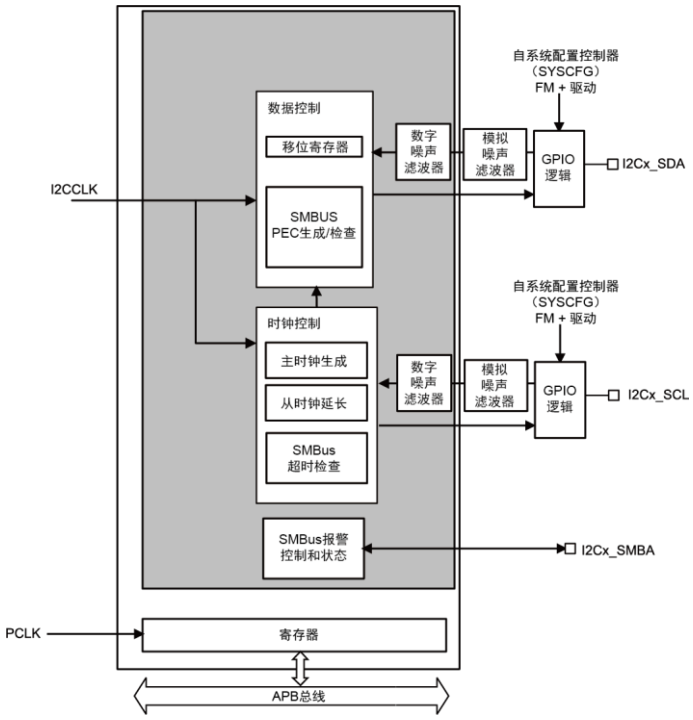
该接口也可通过数据引脚（SDA）和时钟引脚（SCL）连接到 SMBus。

如果支持 SMBus 功能：还可使用额外的可选 SMBus 报警引脚（SMBA）。

19.4.1 I²C 框图

I²C 接口的框图如下。

图 83. I²C 框图



I²C 的时钟由独立时钟源提供，这使得 I²C 能够独立于 PCLK 频率工作。

此独立时钟源为固定的 16-MHz 时钟。参见第 6 节“复位和时钟控制器（RCC）”了解更多信息。

I²C I/O 支持 20 mA 输出电流驱动，用于超快速模式操作。这可以通过将 SCL 和 SDA 的驱动功能控制位置 1 来使能（参见第 8.2.3 节“I²C 超快速模式引脚功能控制寄存器（I²C_FMP_CTRL）”）。

19.4.2 I²C 时钟要求

I²C 内核的时钟由 I2CCLK 提供。

I2CCLK 周期 t_{I2CCLK} 必须遵循以下条件：

$t_{I2CCLK} < (t_{LOW} - t_{filters})/4$ 且 $t_{I2CCLK} < t_{HIGH}$

其中：

t_{LOW}：SCL 低电平时间和 t_{HIGH}：SCL 高电平时间

$t_{filters}$：滤波器使能时，该值为模拟滤波器和数字滤波器引入的延时总和。模拟滤波器延时最大值为 260 ns。数字滤波器延时为 DNF x t_{I2CCLK}。

PCLK 时钟周期 tPCLK 必须满足以下条件：$t_{PCLK} < 4/3\ t_{SCL}$

其中的 t_{SCL} 为 SCL 周期。

19.4.3 模式选择

该接口在工作时可选用以下四种模式之一：

- 从发射器
- 从接收器
- 主发射器
- 主接收器

默认情况下，它以从模式工作。接口在生成起始位后会自动由从模式切换为主模式，并在出现仲裁丢失或生成停止位时从主模式切换为从模式，从而实现多主模式功能。

通信流程

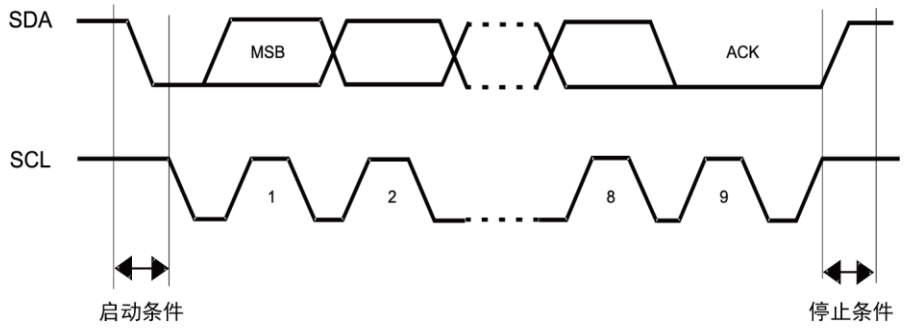
在主模式下，I²C 接口会启动数据传输并生成时钟信号。串行数据传输始终是在出现起始位时开始，在出现停止位时结束。起始位和停止位均在主模式下由软件生成。

在从模式下，该接口能够识别其自身地址（7 或 10 位）以及广播呼叫地址。广播呼叫地址检测可由软件使能或禁止。保留的 SMBus 地址也可由软件使能。

数据和地址均以 8 位字节传输，MSB 在前。起始位后紧随地址字节（7 位地址占据一个字节；10 位地址占据两个字节）。地址总是以主模式发送。

在字节传输 8 个时钟周期后是第 9 个时钟脉冲，在此期间接收器必须向发射器发送一个应答位。请参见下图。

图 84. I²C 总线协议



应答位可由软件使能或禁止。I²C 接口地址可通过软件进行选择。

19.4.4 I²C 初始化

使能和禁止外设

必须在时钟控制器中配置和使能 I²C 外设时钟（参见第 6 节“复位和时钟控制器（RCC）”）。

然后可通过将 I2C_CR1 寄存器中的 PE 位置 1 使能 I²C。当 I²C 禁用（PE=0）时，I²C 执行软件复位。参见第 19.4.5 节“软件复位”了解更多信息。

噪声滤波器

通过将 I2C_CR1 寄存器中的 PE 位置 1 来使能 I²C 外设之前，如有必要，必须配置噪声滤波器。默认情况下，SDA 和 SCL 输入上集成了模拟噪声滤波器。此模拟滤波器兼容 I²C 规范，该规范要求尖峰抑制，在快速模式和超快速模式下脉冲宽度不超过 50 ns。可通过将 ANFOFF 位置 1 来禁止该模拟滤波器，和/或通过配置 I2C_CR1 寄存器中的 DNF[3:0]位来选择数字滤波器。

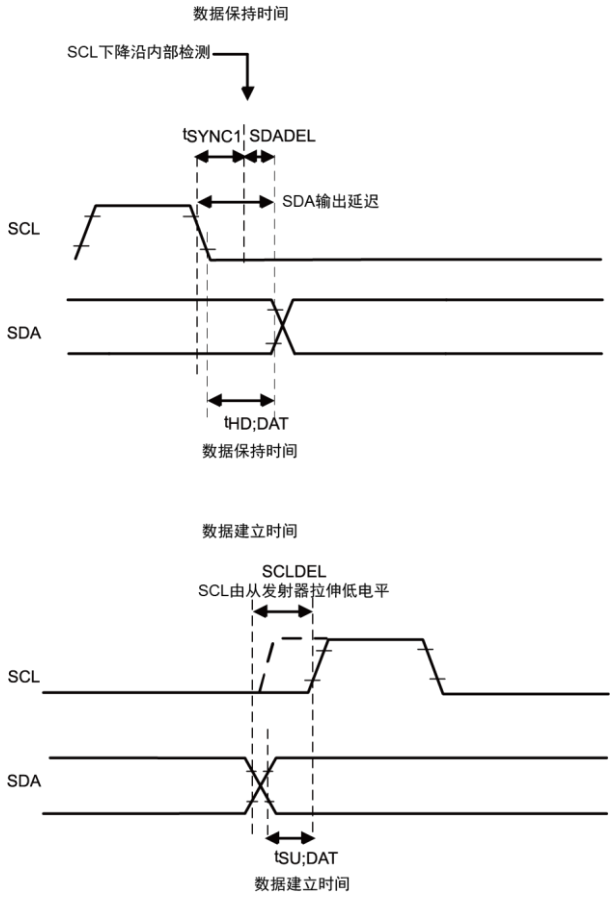
使能数字滤波器时，SCL 或 SDA 线的电平只有在电平稳定时间超过 DNF x I2CCLK 个周期后才会发生内部变化，如此可用 1 至 15 个 I2CCLK 周期的可编程长度进行尖峰抑制。

注意：使能 I²C 时，不允许更改滤波器配置。

I²C 时序

为保证正确的数据保持和建立时间，必须配置在主和从模式下使用的时序。配置方法是编程 I2C_TIMINGR 寄存器中的 PRESC[3:0]、SCLDEL[3:0]和 SDADEL[3:0]位。

图 85. 建立和保持时序



- 当内部检测到 SCL 下降沿时，会在发送 SDA 输出之前插入一段延时。该延时为 t_{SDADEL}= SDADEL x t_{PRESC}+ t_{I2CCLK}，其中的 t_{PRESC}=（PRESC+1）x t_{I2CCLK}。
  T_{SDADEL}，会影响保持时间 $t_{HD;DAT}$。
  SDA 总输出延时为：
  t_{SYNC1} + {[SDADEL x（PRESC+1）+ 1] x tI2CCLK}
  t_{SYNC1} 持续时间取决于这些参数：
  - SCL 下降斜率
  - 当使能时，模拟滤波器引起的输入延时 tSYNC1：$t_{AF(min)}$< t_{AF}< $t_{AF(max)}$ns。
  - 当使能时，数字滤波器引起的输入延时：t_{DNF}= DNF x t_{I2CCLK}
  - SCL 与 I2CCLK 时钟建立同步而产生的延时（2 到 3 个 I2CCLK 周期）

  为了桥接 SCL 下降沿的未定义区域，编程 SDADEL 时必须遵循以下条件：
  {$t_{f(max)}$+$t_{HD;DAT(min)}$ - $t_{AF(min)}$- [(DNF +3)x t_{I2CCLK}]}/{(PRESC +1)x t_{I2CCLK}} ≤ SDADEL
  SDADEL ≤ {$t_{HD;DAT(max)}$-$t_{AF(max)}$ - [(DNF+4) x t_{I2CCLK}]}/{(PRESC +1)x t_{I2CCLK}}

注意： *仅当模拟滤波器使能时，$t_{AF(min)}$/ $t_{AF(max)}$才是等式的一部分。参见器件数据手册获取 tAF 值。*
最大 $t_{HD;DAT}$ 在标准模式、快速模式和超快速模式下分别可以是 3.45 µs、0.9 µs 和 0.45 µs，但必须比 $t_{VD;DAT}$ 最大值少一个过渡时间。只有器件未延长 SCL 信号的低电平周期（t_{LOW}）时，才必须满足该最大值条件。如果时钟延长 SCL，数据必须在建立时间内保持有效，之后才能释放时钟。

SDA 上升沿通常为最坏情况，因此在这种情况下，上述公式变成如下形式：

SDADEL ≤ {$t_{VD;DAT(max)}$-$t_{r(max)}$-260 ns - [(DNF+4) x t_{I2CCLK}]}/{(PRESC +1) x t_{I2CCLK}}。

注意： *NOSTRETCH=0 时会违反该条件，这是因为器件会根据 SCLDEL 值来延长 SCL 低电平时间，以保证建立时间。*

请参见“表 55. I²C-SMBUS 规范数据建立时间和保持时间”的 t_f、t_r、$t_{HD;DAT}$ 和 $t_{VD;DAT}$ 标准值。

- 在发送 SDA 输出后，SCL 线在建立时间内保持低电平。此建立时间为 t_{SCLDEL}=（SCLDEL+1）x t_{PRESC}，其中的 t_{PRESC} =（PRESC+1）x t_{I2CCLK} t_{SCLDEL} 影响建立时间 $t_{SU;DAT}$。

为了桥接 SDA 跳变沿（上升沿通常为最坏情况）的未定义区域，编程 SCLDEL 时必须遵循以下条件：

{[$t_{r(max)}$+ $t_{SU;DAT(min)}$]/[(PRESC+1)] x t_{I2CCLK}]} - 1 <=SCLDEL

请参见“表 55. I²C-SMBUS 规范数据建立时间和保持时间”的 t_r 和 $t_{SU;DAT}$ 标准值。

将使用的 SDA 和 SCL 跳变时间值就是应用中的值。使用最大值而非标准值会增加 SDADEL 和 SCLDEL 计算的约束条件，但能够确保任意应用的特性。

表 55. I²C-SMBUS 规范数据建立时间和保持时间

符号	参数	标准模式（Sm）		快速模式（Fm）		超快速模式（Fm）		SMBUS		单位
		最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值	
$t_{HD;DAT}$	数据保持时间	0	-	0	-	0	-	0.3	-	µs
$t_{VD;DAT}$	数据有效时间	-	3.45	-	0.9	-	0.45	-	-	
$t_{SU;DAT}$	数据建立时间	250	-	100		50		250		ns
t_r	SDA 和 SCL 信号的上升时间	-	1000		300	-	120	-	1000	
tf	SDA 和 SCL 信号的下降时间	-	300		300	-	120	-	300	

此外，在主模式下，必须通过编程 I2C_TIMINGR 寄存器中的 PRESC[3:0]、SCLH[7:0]和 SCLL[7:0]位来配置 SCL 时钟的高电平和低电平。

- 当内部检测到 SCL 下降沿时，会在释放 SCL 输出之前插入一段延时。此延时为 t_{SCLL} =（SCLL+1）x t_{PRESC}，其中的 t_{PRESC} =（PRESC+1）x t_{I2CCLK}。

t_{SCLL} 影响 SCL 低电平时间 t_{LOW}。

- 当内部检测到 SCL 上升沿时，会在将 SCL 输出强制为低电平之前插入一段延时。此延时为 t_{SCLH} =（SCLH+1）x t_{PRESC}，其中的 t_{PRESC} =（PRESC+1）x t_{I2CCLK}。t_{SCLH}，会影响 SCL 高电平时间 t_{HIGH}。

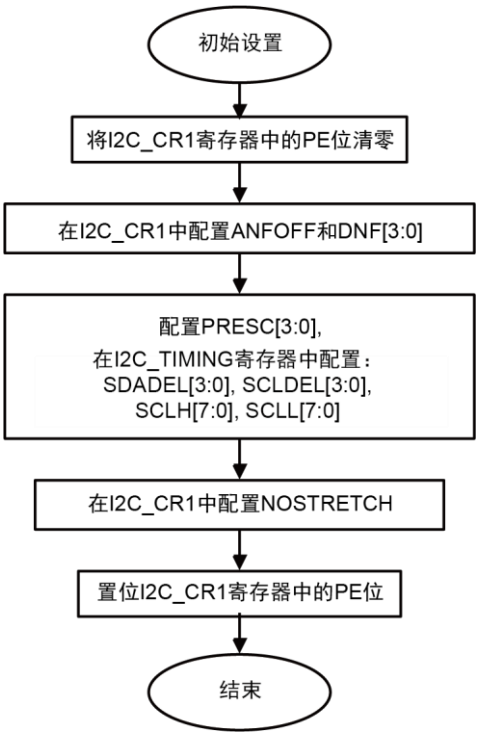
参见第 19.4.8 节“I²C 主模式”了解更多信息。

注意：使能 I²C 后，不允许更改时序配置。

在使能外围设备之前，还必须配置 I²C 从 NOSTRETCH 模式。

注意：使能 I²C 后，不允许更改 NOSTRETCH 配置。

图 86. I²C 初始化流程图



19.4.5 软件复位

可通过将 I2C_CR1 寄存器中的 PE 位清零来执行软件复位。在这种情况下，I²C 线 SCL 和 SDA 被释放。内部状态机复位，通信控制位和状态位恢复为其复位值。配置寄存器不受影响。

下面列出了受影响的寄存器位：

1. I2C_CR2 寄存器：START、STOP 和 NACK
2. I2C_ISR 寄存器：BUSY、TXE、TXIS、RXNE、ADDR、NACKF、TCR、TC、STOPF、BERR、ARLO 和 OVR

支持 SMBus 功能时还会影响到以下寄存器位：

1. I2C_CR2 寄存器：PECBYTE
2. I2C_ISR 寄存器：PECERR、TIMEOUT 和 ALERT

必须使 PE 保持低电平持续至少 3 个 APB 时钟周期，才能成功执行软件复位。通过写入以下软件序列确保这一点：- 写入 PE=0 - 检查 PE=0 - 写入 PE=1。

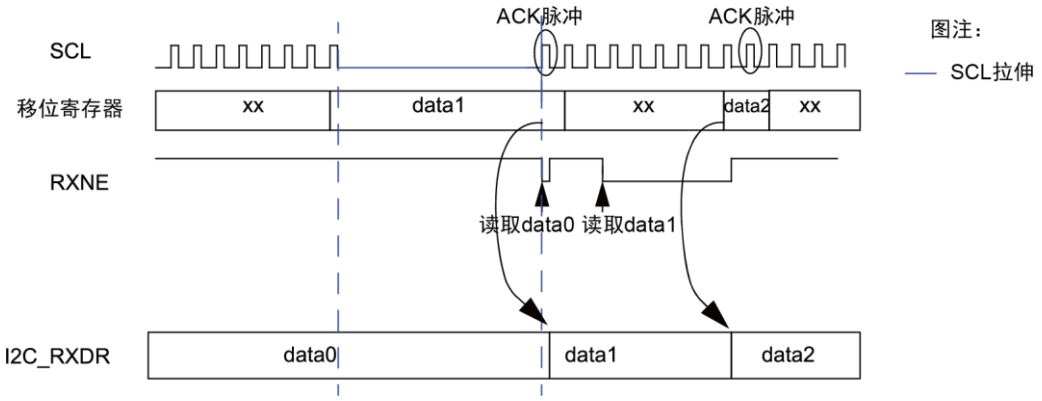
19.4.6 数据传输

数据传输由发送和接收数据寄存器以及移位寄存器来管理。

接收

SDA 输入填充移位寄存器。在第 8 个 SCL 脉冲后（接收到完整的数据字节时），如果 I2C_RXDR 寄存器为空（RXNE=0），则移位寄存器的内容会复制到其中。如果 RXNE=1（意味着尚未读取上一次接收到的数据字节），则将延长 SCL 线的低电平时间，直到读取了 I2C_RXDR 为止。在第 8 个和第 9 个 SCL 脉冲之间（应答脉冲之前）插入一段延长的时间。

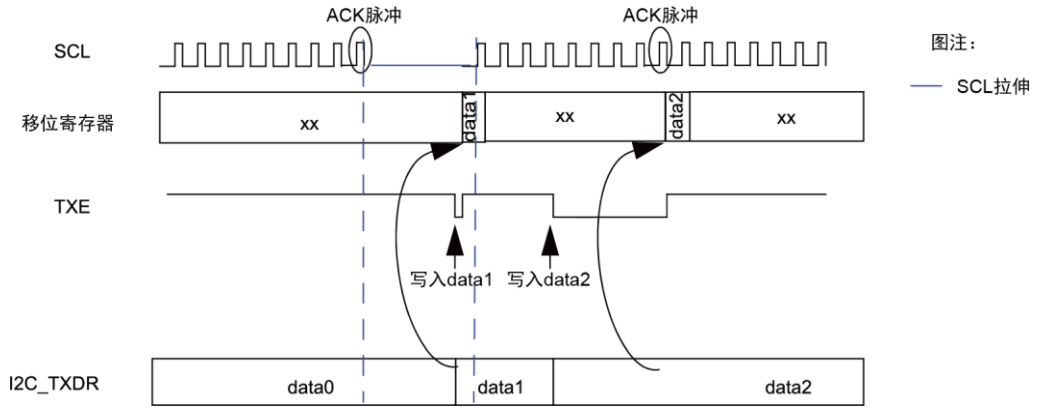
图 87. 数据接收



传输

如果 I2C_TXDR 寄存器不为空（TXE=0），则其内容会在第 9 个 SCL 脉冲（应答脉冲）后复制到移位寄存器中。然后，移位寄存器内容移位并通过 SDA 线输出。如果 TXE=1（意味着 I2C_TXDR 内尚未写入任何数据），则将延长 SCL 线的低电平时间，直到写入了 I2C_TXDR 为止。在第 9 个 SCL 脉冲后进行延长。

图 88. 数据发送



硬件传输管理

I²C 在硬件中内置了字节计数器，以便在下列各种模式下管理字节传输和结束通信：

- 主模式下生成 NACK、STOP 和 ReSTART
- 从接收器模式下控制 ACK 是否发出
- SMBus 模式下生成/校验 PEC

字节计数器总是在主模式下使用。在从模式下，字节计数器默认为禁止状态，但可以通过软件来使能，方法是将 I2C_CR2 寄存器中的 SBC（从字节控制）位置 1。

待传输的字节数在 I2C_CR2 寄存器的 NBYTES[7:0]位字段中进行编程。如果待传输的字节数（NBYTES）大于 255，或者接收方希望控制是否对接收到的数据字节进行应答，则必须选择重载模式，方法是将 I2C_CR2 寄存器的 RELOAD 位置 1。在该模式下，完成 NBYTES 中所编程字节数的数据传输之后，TCR 标志将置 1，并且 TCIE 置 1 时将生成中断。只要 TCR 标志置 1，SCL 便会延长。当往 NBYTES 写入一个非零值时，TCR 由软件清零。

在往 NBYTE 中设置最后一次传输的字节数前，必须把 RELOAD 位清零。

当主模式下 RELOAD=0 时，可在以下 2 种模式下使用计数器：

- **自动结束模式**（I2C_CR2 寄存器中的 AUTOEND = “1”）。在该模式下，一旦完成 NBYTES[7:0]位字段中所编程字节数的数据传输，主设备便会自动发送停止位。
- **软件结束模式**（I2C_CR2 寄存器中的 AUTOEND = “0”）。在该模式下，一旦完成 NBYTES[7:0]位字段中所编程字节数的数据传输，TC 标志将置 1，并且 TCIE 置 1 时将生成中断。只要 TC 标志置 1，SCL 信号便会延长，需要软件介入操作。当软件把 I2C_CR2 寄存器中的起始位或停止位置 1 时，TC 标志将被清零。当主设备要发送重复起始位时，必须使用该模式。

注意：当 RELOAD 位置 1 时，AUTOEND 位将不起作用。

表 56. I2C 可配置表

功能	SBC 位	RELOAD 位	AUTOEND 位
主 Tx/Rx NBYTES + STOP	x	0	1
主 Tx/Rx + NBYTES + RESTART	x	0	0
从 Tx/Rx 接收的所有字节都要回复应答	0	x	x
具有 ACK 控制的从 Rx	1	1	x

19.4.7 I²C 从模式

I²C 从设备初始化

要在从模式下工作，必须至少使能一个从设备地址。可使用 I2C_OAR1 和 I2C_OAR2 这两个寄存器来编程自身从设备地址 OA1 和 OA2。

- OA1 既可配置为 7 位寻址模式（默认），也可通过将 I2C_OAR1 寄存器的 OA1MODE 位置 1 配置为 10 位寻址模式。

  通过将 I2C_OAR1 寄存器中的 OA1EN 位置 1 来使能 OA1。

- 如果需要额外的从设备地址，可配置第 2 个从设备地址 OA2。将 I2C_OAR2 寄存器的 OA2MSK[2:0]位置 1 最多可屏蔽 7 个 OA2 LSB。因此，当 OA2MSK 配置为 1 到 6 时，将分别只有 OA2[7:2]、OA2[7:3]、OA2[7:4]、OA2[7:5]、OA2[7:6]或 OA2[7]与接收到的地址作比较。只要 OA2MSK 不等于 0，OA2 的地址比较器便会排除 I²C 保留地址（0000 XXX 和 1111 XXX），这些地址将不会得到应答。如果 OA2MSK=7，接收到的所有 7 位地址（保留地址除外）均得到应答。OA2 始终为 7 位地址。

  如果这些保留地址在 I2C_OAR1 或 I2C_OAR2 寄存器中进行了编程并且 OA2MSK=0，则它们可以在通过特定使能位使能后得到应答。

  通过将 I2C_OAR2 寄存器中的 OA2EN 位置 1 来使能 OA2。

- 通过将 I2C_CR1 寄存器中的 GCEN 位置 1 来使能广播呼叫地址。

当通过 I²C 的其中一个使能地址来寻址到该 I2C 设备时，ADDR 中断状态标志将置 1，并且 ADDRIE 位置 1 时将生成中断。

默认情况下，从设备使用其时钟延长功能（即必要时延长 SCL 信号的低电平时间）来为软件操作的执行提供时机。如果主设备不支持时钟延长，则必须对 I²C 进行如下配置：将 I2C_CR1 寄存器中 NOSTRETCH 位置 1。

接收到 ADDR 中断后，如果从设备使能多个地址，则必须读取 I2C_ISR 寄存器中的 ADDCODE[6:0]位，以检查哪个地址匹配。还必须检查 DIR 标志，以获悉传输方向。

带时钟延长的从模式（NOSTRETCH = 0）

在默认模式下，I²C 从设备会在以下情况下延长 SCL 时钟：

- ADDR 标志置 1 时：接收到的地址与其中一个使能的从设备地址匹配。通过软件将 ADDRCF 位置 1 以清零 ADRR 标志时，将释放该时钟延展。
- 发送时，前一次数据传输已完成但 I2C_TXDR 寄存器中未写入任何新数据，或者 ADDR 标志清零（TXE=1）时未写入第一个数据字节。往 I2C_TXDR 寄存器中写入数据时，将释放该时钟延展。
- 接收时，尚未读取 I2C_RXDR 寄存器但新的数据接收已完成。读取 I2C_RXDR 时，将释放该时钟延展。
- 当从设备字节控制模式和重载模式（SBC=1 且 RELOAD=1）下 TCR = 1 时，这意味着最后一个数据字节已完成传输。通过向 NBYTES[7:0]字段写入一个非零值以将 TCR 清零时，将释放该时钟延展。
- 在 SCL 下降沿检测之后，I²C 会延长 SCL 的低电平时间（不超过 [(SDADEL+SCLDEL+1）x（PRESC+1）+ 1] x tI2CCLK）。

不带时钟延长的从模式（NOSTRETCH = 1）

当 I2C_CR1 寄存器中的 NOSTRETCH = 1 时，I²C 从设备不会延长 SCL 信号。

- ADDR 标志置 1 时，不会延长 SCL 时钟。
- 发送时，必须在与发送数据对应的第一个 SCL 脉冲出现之前，向 I2C_TXDR 寄存器写入数据。否则，会发生下溢，I2C_ISR 寄存器中的 OVR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。当第一次数据发送开始而 STOPF 位仍置 1（尚未清零）时，OVR 标志也将置 1。因此，如果写入下一次传输要发送的第一个数据后才清零上一次传输的 STOPF 标志，则应检查 OVR 状态，甚至对于待发送的第一个数据。
- 接收时，必须在下一个数据字节的第 9 个 SCL 脉冲（ACK 脉冲）出现之前，从 I2C_RXDR 寄存器读取数据。否则，会发生上溢，I2C_ISR 寄存器中的 OVR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

从设备字节控制模式

要在从接收模式下实现字节 ACK 控制，必须通过将 I2C_CR1 寄存器中的 SBC 位置 1 来使能从设备字节控制模式。这样符合 SMBus 标准。

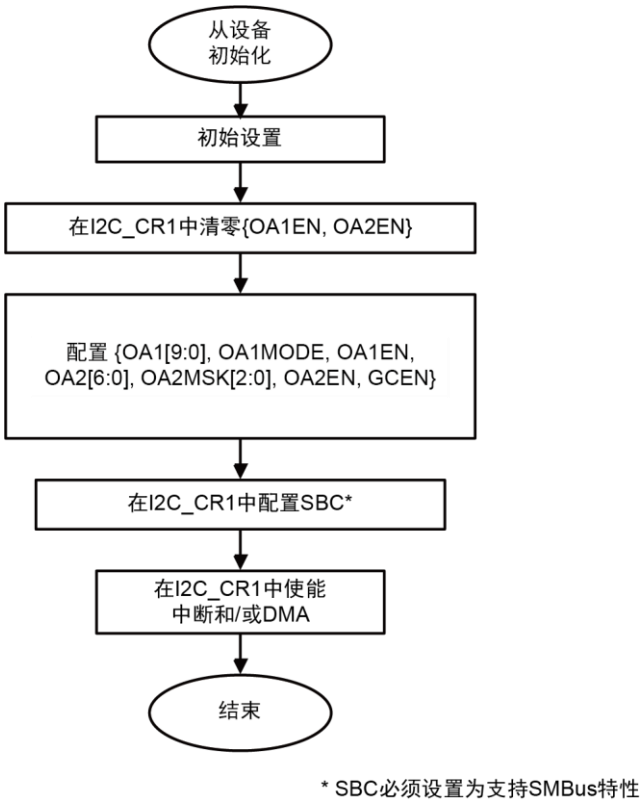
要在从接收模式下实现字节 ACK 控制，必须选择重载模式（RELOAD=1）。要控制每个字节，必须在 ADDR 中断子程序中将 NBYTES 初始化为 0x1，并在每接收一个字节后将 NBYTE 重载为 0x1。接收到字节后，TCR 位将置 1，从而延长 SCL 信号的第 8 个和第 9 个脉冲之间的低电平时间。可以从 I2C_RXDR 寄存器中读取数据，然后决定是否配置 I2C_CR2 寄存器中的 ACK 位以进行应答。通过将 NBYTES 编程为非零值来释放 SCL 延长：发送应答或不应答信号，然后可继续接收下一个字节。

NBYTES 可加载大于 0x1 的值，在这种情况下，接收流在 NBYTES 个数据接收期间是连续的。

注意： *SBC 位只能在 I²C 被禁止时、从设备不被寻址时或 ADDR=1 时配置。ADDR=1 或 TCR=1 时，可以更改 RELOAD 位的值。*

注意：从设备字节控制模式与 NOSTRETCH 模式不兼容。不允许在 NOSTRETCH=1 时置位 SBC。

图 89. 从设备初始化流程图



从发射器

当 I2C_TXDR 寄存器为空时，将生成发送中断状态（TXIS）。如果 I2C_CR1 寄存器中的 TXIE 位置 1，则生成中断。

I2C_TXDR 寄存器中写入待发送的下一个数据字节时，TXIS 位将被清零。

接收到 NACK 时，I2C_ISR 寄存器中的 NACKF 位将置 1，如果 I2C_CR1 寄存器中的 NACKIE 位置 1，还将生成中断。从设备自动释放 SCL 和 SDA 线，以使主设备执行停止或重复起始位的发送。收到 NACK 时，TXIS 位不会置 1。

当接收到停止位且 I2C_CR1 寄存器中的 STOPIE 位置 1 时，I2C_ISR 寄存器中的 STOPF 标志将置 1 并且会生成中断。在大多数应用中，SBC 位通常编程为“0”。在这种情况下，如果接收到从设备地址（ADDR=1）时 TXE = 0，可以选择发送 I2C_TXDR 寄存器的内容作为第一个数据字节，也可以选择通过将 TXE 位置 1 来刷新 I2C_TXDR 寄存器以编程新的数据字节。

在从设备字节控制模式（SBC=1）下，必须在地址匹配中断子程序（ADDR=1）中向 NBYTE 写入待发送数据的个数。在这种情况下，传输期间 TXIS 事件的数量对应于 NBYTES 中编程的值。

注意：如果 NOSTRETCH=1，当 ADDR 标志置 1 时不会延长 SCL 时钟，因此无法通过在 ADDR 子程序中刷新 I2C_TXDR 寄存器的内容来编程第一个数据字节。必须在 I2C_TXDR 寄存器中预编程待发送的第一个数据字节：

- 该数据可以是前一个传输消息的最后一个 TXIS 事件中写入的数据

- 如果该数据字节不是待发送的数据字节，可通过将 TXE 位置 1 来刷新 I2C_TXDR 寄存器，从而编程新的数据字节。必须仅在执行完这些操作后再清零 STOPF 位，以确保在第一次数据传输之后地址应答之前执行这些操作。

  当第一次数据发送开始而 STOPF 位仍置 1 时，生成下溢错误（OVR 标志置 1）。

  如果需要 TXIS 事件（发送中断或发送 DMA 请求），则必须将 TXE 位和 TXIS 位均置 1，以便生成 TXIS 事件。

图 90. I2C 从发射器的传输序列流程图（NOSTRETCH=0）

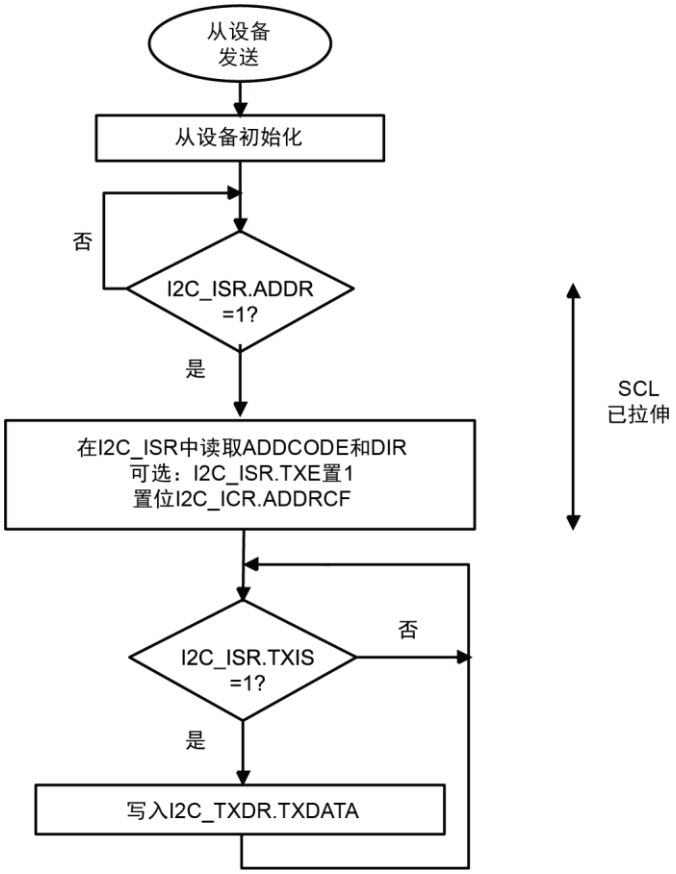


图 91. I2C 从发射器的传输序列流程图（NOSTRETCH=1）

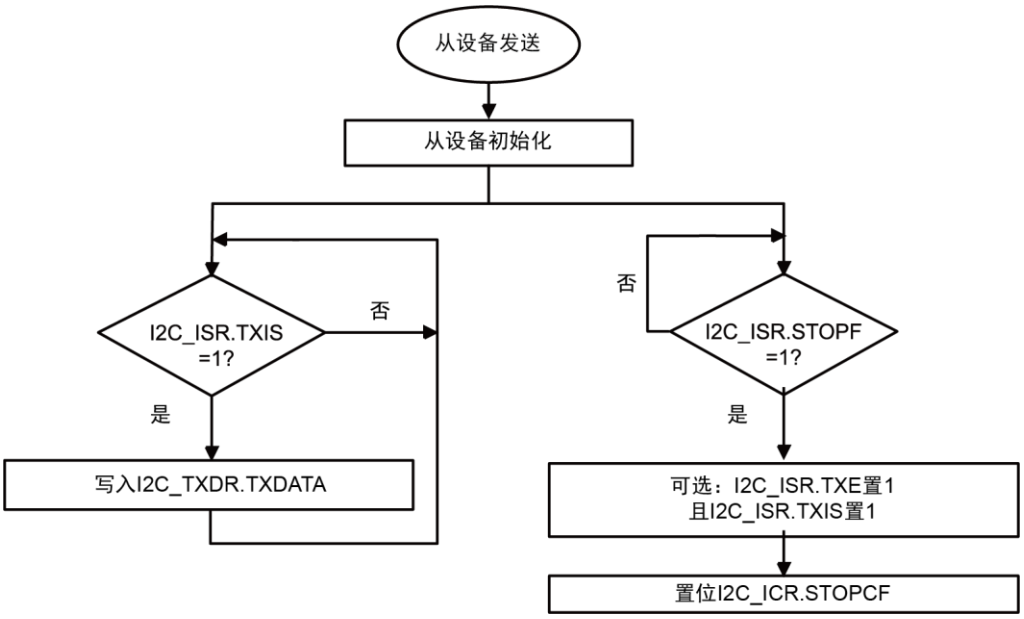
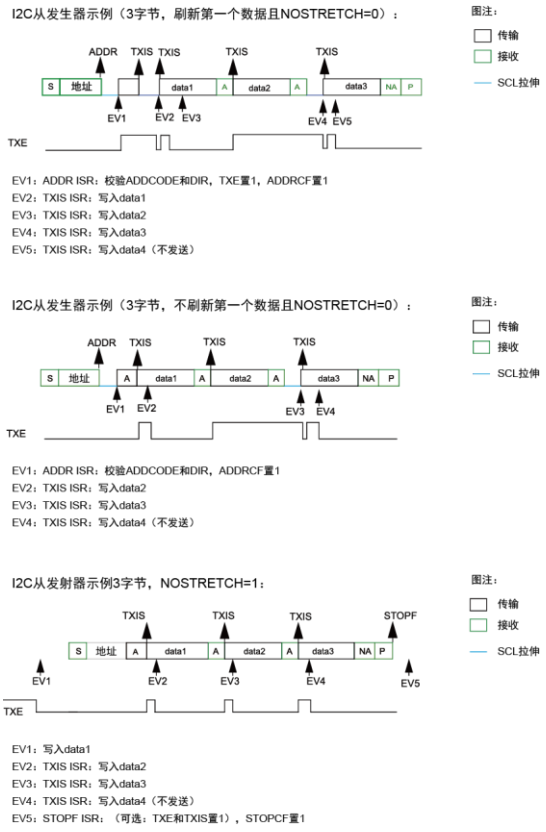


图 92. I2C 从发射器的传输总线图



从接收器

当 I2C_RXDR 满时，I2C_ISR 中的 RXNE 将置 1，如果 I2C_CR1 中的 RXIE 置 1，还将生成中断。读取 I2C_RXDR 时，将清零 RXNE。

接收到停止条件且 I2C_CR1 寄存器中的 STOPIE 置 1 时，I2C_ISR 中的 STOPF 将置 1 并且会生成中断。

图 93. 从接收器的传输序列流程图（NOSTRETCH=0）

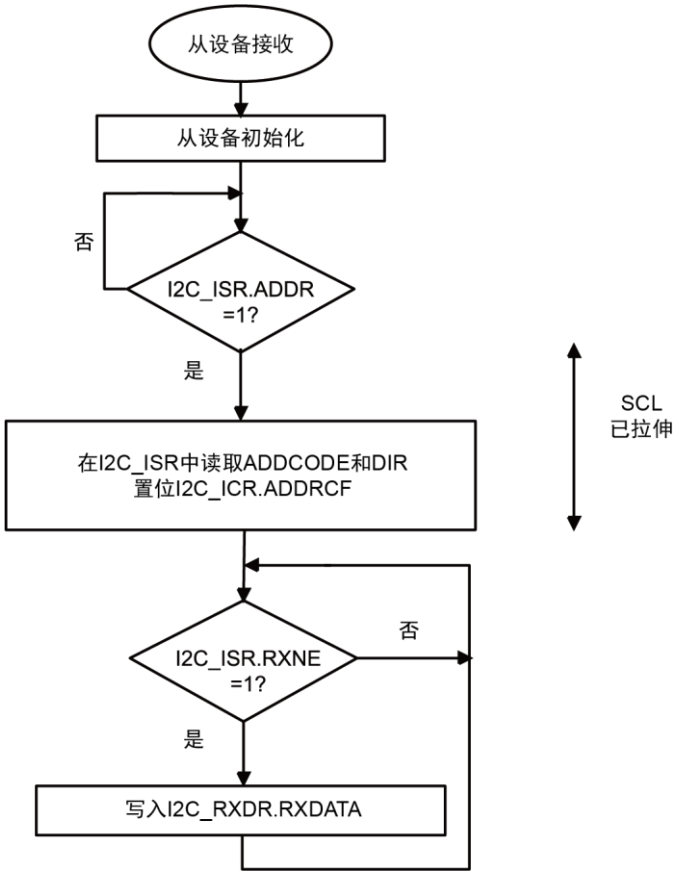


图 94. 从接收器的传输序列流程图（NOSTRETCH=1）

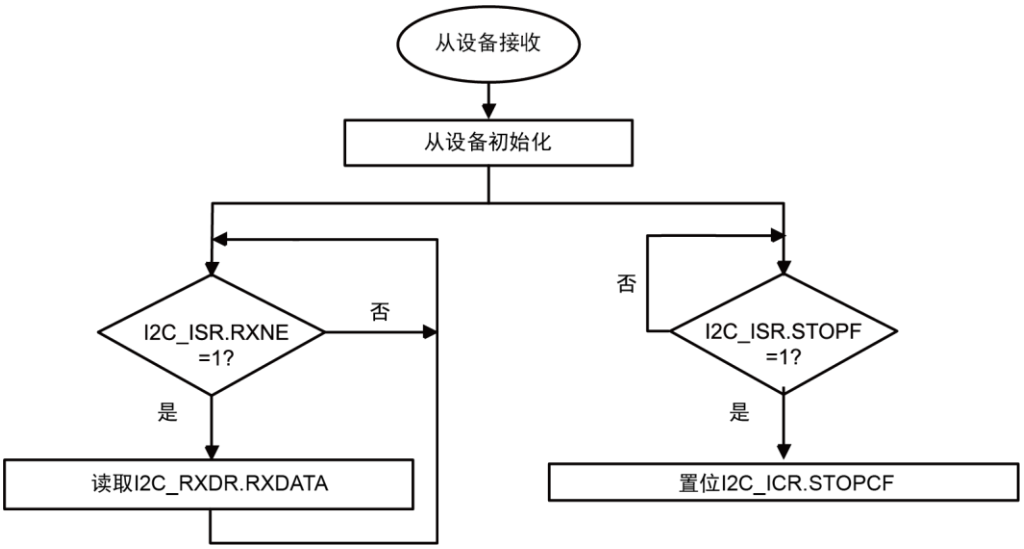
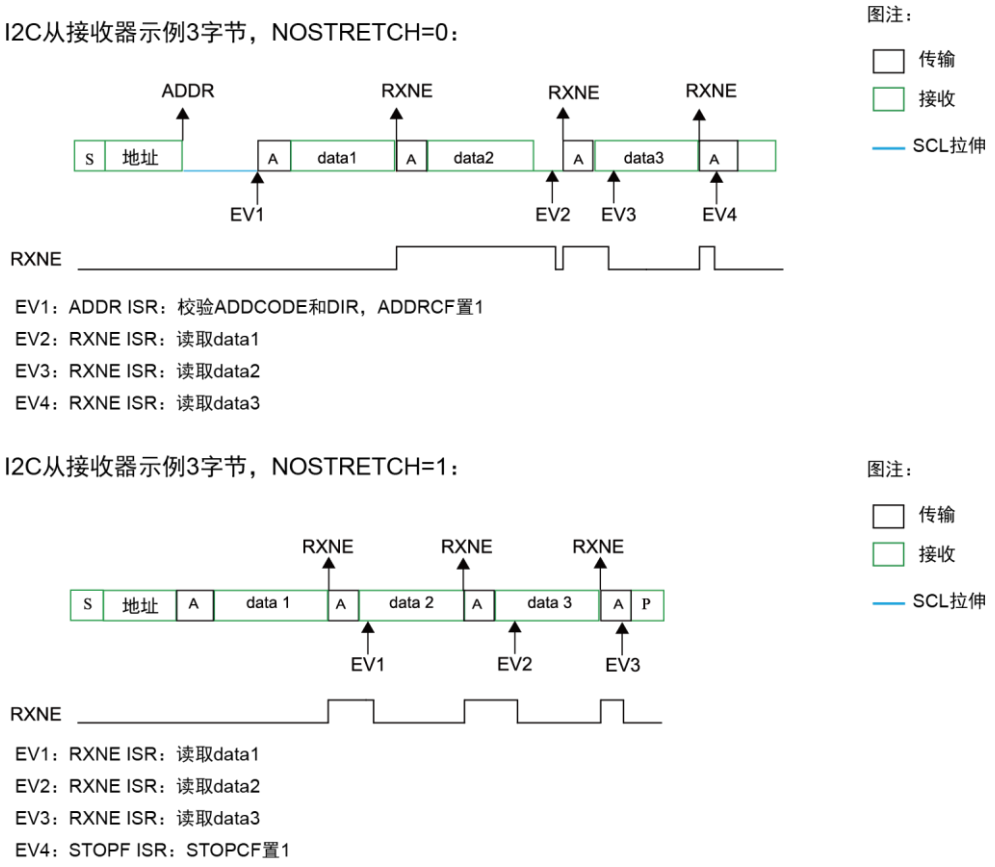


图 95. I2C 从接收器的传输总线图



19.4.8 I²C 主模式

I²C 主模式初始化

使能外设前，必须通过设置 I2C_TIMINGR 寄存器中的 SCLH 和 SCLL 位来配置 I^2C 主时钟。

为了支持多主环境和从时钟延长，I2C 实现了时钟同步机制。

为了实现时钟同步，需执行以下操作：

- 从内部检测到 SCL 低电平开始，使用 SCLL 计数器对时钟的低电平时长进行计数。
- 从内部检测到 SCL 高电平开始，使用 SCLH 计数器对时钟的高电平时长进行计数。

I^2C 经过 t_{SYNC1} 延时后检测其自身的 SCL 低电平，该延时取决于 SCL 下降沿、SCL 输入噪声滤波器（模拟 + 数字）以及 SCL 与 I2CxCLK 时钟的同步。一旦 SCLL 计数器达到 I2C_TIMINGR 寄存器的 SCLL[7:0]位中编程的值，I^2C 便会将 SCL 释放为高电平。

I^2C 经过 t_{SYNC2} 延时后检测其自身的 SCL 高电平，该延时取决于 SCL 上升沿、SCL 输入噪声滤波器（模拟 + 数字）以及 SCL 与 I2CxCLK 时钟的同步。一旦 SCLH 计数器达到 I2C_TIMINGR 寄存器的 SCLH[7:0]位中编程的值，I^2C 便会使 SCL 变为低电平。

因此，主时钟周期为：

$t_{SCL} = t_{SYNC1} + t_{SYNC2} + \{[（SCLH+1）+（SCLL+1）] x（PRESC+1）x t_{I2CCLK}\}$

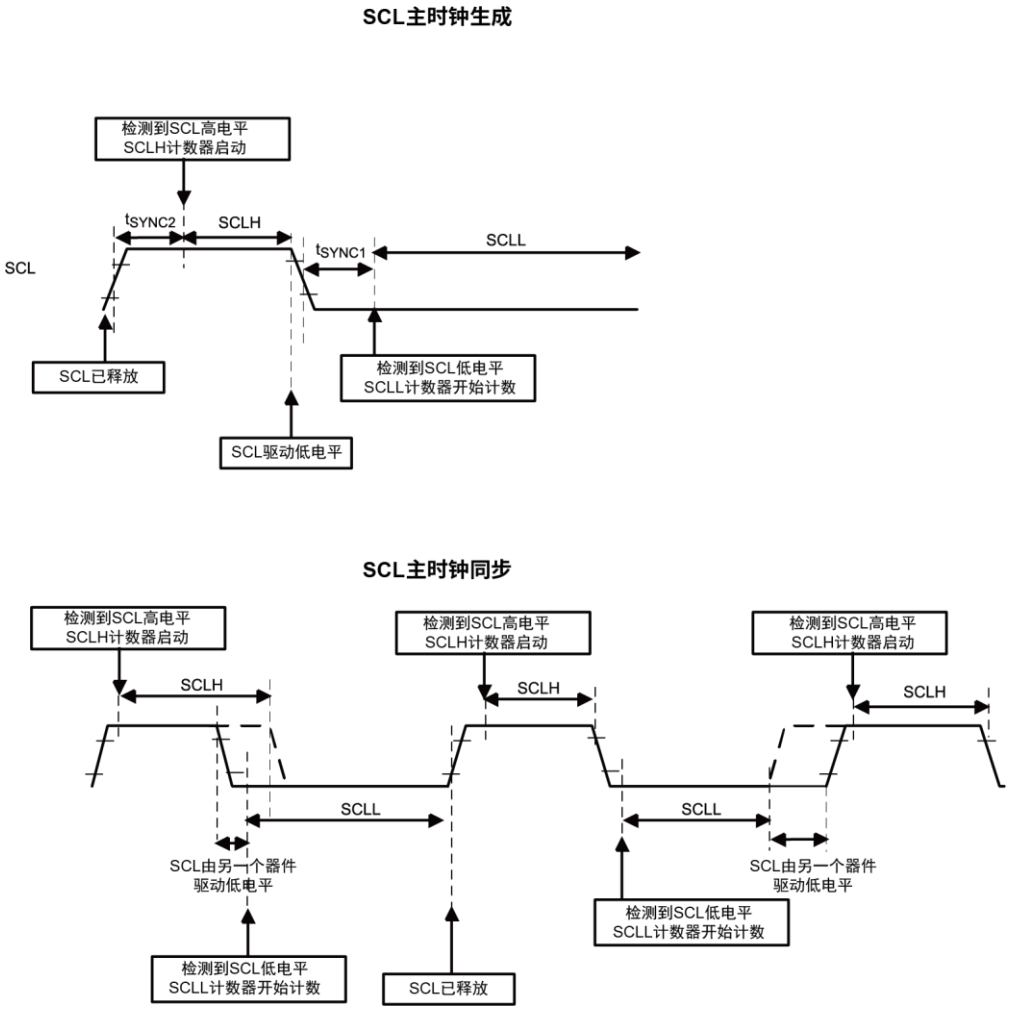
t_{SYNC1} 的持续时间取决于以下参数：

- SCL 下降斜率
- 模拟滤波器（使能时）引入的输入延时
- 数字滤波器（使能时）引入的输入延时：DNF x t_{I2CCLK}
- SCL 与 I2CCLK 时钟建立同步而产生的延时（2 到 3 个 I2CCLK 周期）。

t_{SYNC2} 的持续时间取决于以下参数：

- SCL 上升斜率
- 模拟滤波器（使能时）引入的输入延时
- 数字滤波器（使能时）引入的输入延时：DNF xt_{I2CCLK}
- SCL 与 I2CCLK 时钟建立同步而产生的延时（2 到 3 个 I2CCLK 周期）。

图 96. 主时钟生成



注意：为了兼容 I²C 或 SMBus，主时钟必须遵循以下给定时序：

表 57. I²C-SMBUS 规范时钟时序

符号	参数	标准模式（Sm）		快速模式（Fm）		超快速模式（Fm）		SMBUS		单位
		最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值	
f_{SCL}	SCL 时钟频率		100		400		1000		100	kHz
$t_{HD:STA}$	（重复）起始条件的保持时间	4.0	-	0.6		0.26	-	4.0	-	µs
$t_{SU:STA}$	重复起始条件的建立时间	4.7	-	0.6		0.26	-	4.7	-	µs
$t_{SU:STO}$	停止条件的建立时间	4.0	-	0.6		0.26	-	4.0	-	µs
t_{BUF}	停止条件和起始条件之间的总线空闲时间	4.7	-	1.3		0.5	-	4.7	-	µs
t_{LOW}	SCL 时钟的低电平周期	4.7	-	1.3		0.5	-	4.7	-	µs

符号	参数	标准模式（Sm）		快速模式（Fm）		超快速模式（Fm）		SMBUS		单位
		最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值	
f_{SCL}	SCL 时钟频率		100		400		1000		100	kHz
t_{HIGH}	SCL 时钟的高电平周期	4.0	-	0.6		0.26	-	4.0	50	µs
t_r	SDA 和 SCL 信号的上升时间	-	1000	-	300		120	-	1000	ns
t_f	SDA 和 SCL 信号的下降时间	-	300	-	300		120	-	300	ns

注意： *SCLL 还用于生成 t_{BUF} 和 $t_{SU:STA}$ 时序。*

SCLL 还用于生成 tHD:STA 和 tSU:STO 时序。

参见第 19.4.9 节“I2C_TIMINGR 寄存器配置示例”获取不同 I2C_TIMINGR 设置的 I2CCLK 频率的示例。

主设备通信初始化（地址阶段）

要发起通信，必须在 I2C_CR2 寄存器中为寻址的从设备编程以下参数：

- 寻址模式（7 位或 10 位）：ADD10
- 要发送的从设备地址：SADD[9:0]
- 传输方向：RD_WRN
- 若为 10 位地址读取：HEAD10R 位。必须对 HEAD10R 进行相应配置，以指示传输方向变化时必须发送完整的地址序列，还是只发送地址头。
- 要传输的字节数：NBYTES[7:0].如果字节数等于或大于 255，则初始化时必须将 NBYTES[7:0]填充为 0xFF。

然后，必须将 I2C_CR2 寄存器中的 START 位置 1。START 位置 1 时，不允许更改上述所有位。

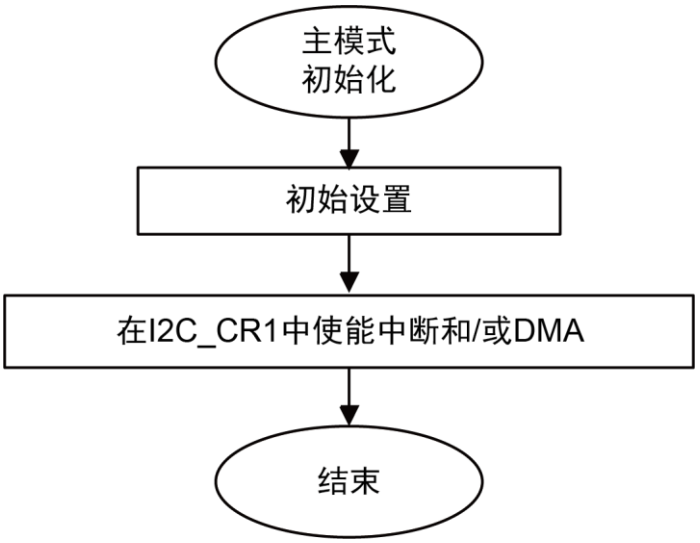
之后，当主设备检测到总线空闲（BUSY = 0）时，它会在经过 t_{BUF} 的延时后自动发送起始位，随后发出从设备地址。

仲裁丢失时，主设备将自动切换回从模式，如果作为从设备被寻址，还可对其自身地址进行应答。

注意： *无论接收到的应答值为何，只要已在总线上发送从设备地址，START 位便会由硬件复位。如果仲裁丢失，START 位也会由硬件复位。如果当 START 位置 1 时，I2C 作为从设备（ADDR=1）被寻址，则 I2C 将切换为从模式，START 位将在 ADDRCF 位置 1 时清零。*

注意： *该步骤同样适用于重复起始位。在这种情况下，BUSY=1。*

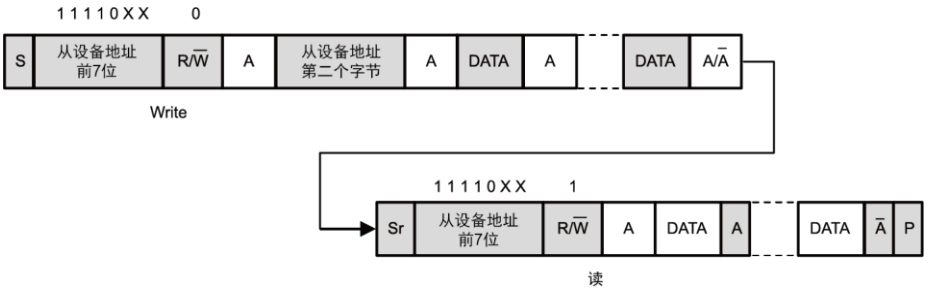
图 97. 主模式初始化流程图



主接收器寻址 10 位地址从设备的初始化过程

- 如果从设备地址为 10 位格式，可选择将 I2C_CR2 寄存器中的 HEAD10R 位清零来发送完整的读序列。这种情况下，在 START 位置 1 后，主设备自动发送以下完整序列：（重复）起始位 + 从设备地址 10 位地址头写入 + 从设备地址第 2 个字节 + 起始位 + 从设备地址 10 位地址头读取
- 如果主设备对 10 位地址从设备进行寻址、向该从设备发送数据、然后再从该从设备读取数据，则必须首先完成主设备发送过程。然后，重复起始位置 1，10 位从设备地址配置为 HEAD10R=1。在这种情况下，主设备发送此序列：重复起始位 + 从设备地址 10 位地址头读取。

图 98. 10 位地址读访问（HEAD10R=1）



主发射器

写传输时，在发送完每个字节（即第 9 个 SCL 脉冲（接收到 ACK 时））后，TXIS 标志将置 1。

如果 I2C_CR1 寄存器中的 TXIE 位置 1，TXIS 事件将生成中断。当 I2C_TXDR 寄存器中写入待发送的下一个数据字节时，该标志将被清零。

传输期间的 TXIS 事件的数量对应于 NBYTES[7:0]中编程的值。如果待发送的数据字节总数大于 255，则必须通过将 I2C_CR2 寄存器中的 RELOAD 位置 1 来选择重载模式。在这种情况下，当 NBYTES 数据传输完成时，TCR 标志将置 1，并且 SCL 线的低电平将被延展，直到 NBYTES[7:0]被写入非零值。

收到 NACK 时，TXIS 标志不会置 1。

- 当 RELOAD=0 且 NBYTES 数据传输完成时：
  - 在自动结束模式（AUTOEND=1）下，将自动发送停止位。
  - 在软件结束模式（AUTOEND=0）下，TC 标志将置 1 且 SCL 线的低电平将被延展，以便执行以下软件操作：

    可通过将 I2C_CR2 寄存器中的 START 位置 1 并配置适当的从设备地址和待传输字节数来请求发送重复起始位。置位 START 位会将 TC 标志清零，并在总线上发送起始位。

    可通过将 I2C_CR2 寄存器中的 STOP 位置 1 来请求停止位。置位 STOP 位会将 TC 标志清零，并在总线上发送 STOP 位。
- 如果接收到 NACK：TXIS 标志不置位，并在接收到 NACK 后自动发送 STOP 位。I2C_ISR 寄存器中的 NACKF 标志置 1，并生成中断（如果 NACKIE 位置 1）。

图 99. I2C 主发射器的传输序列流程图（N 255 字节）

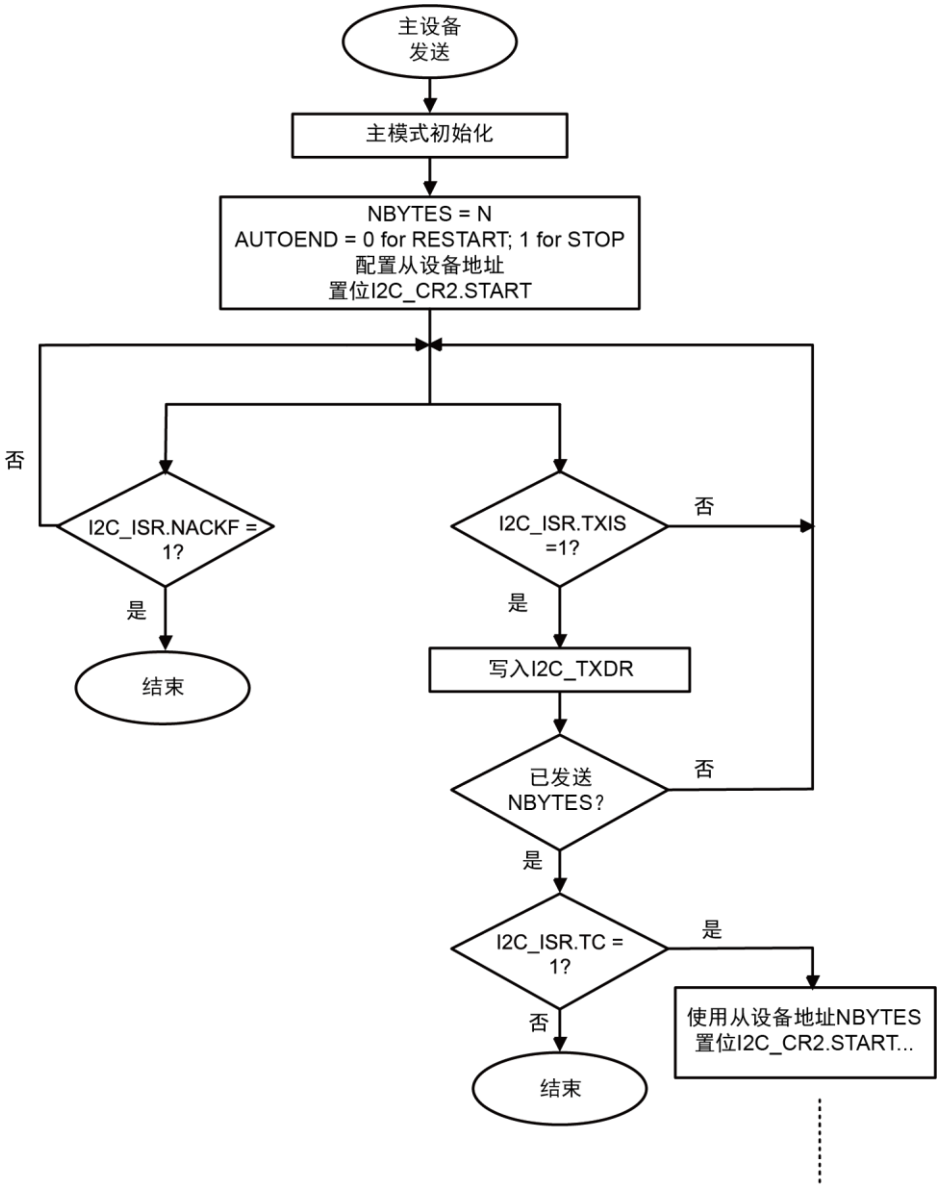


图 100. I2C 主发射器的传输序列流程图（N>255 字节）

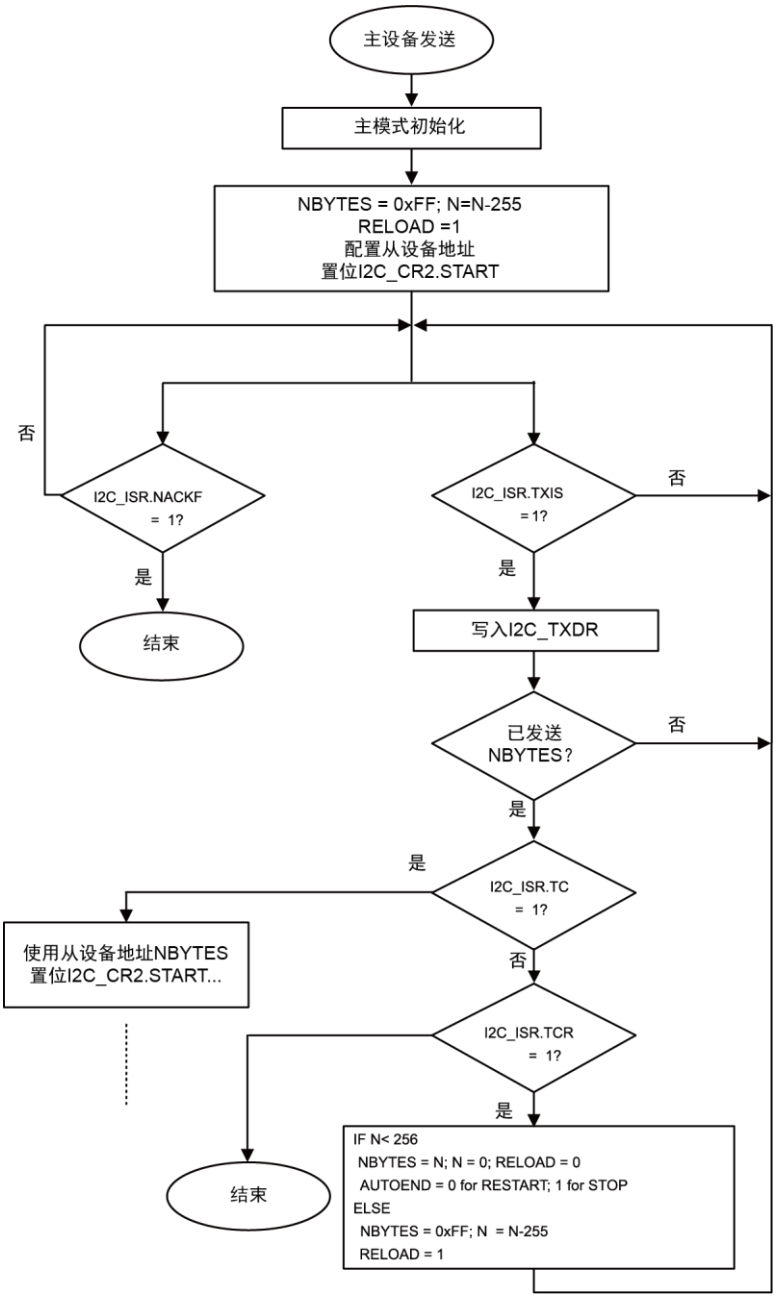
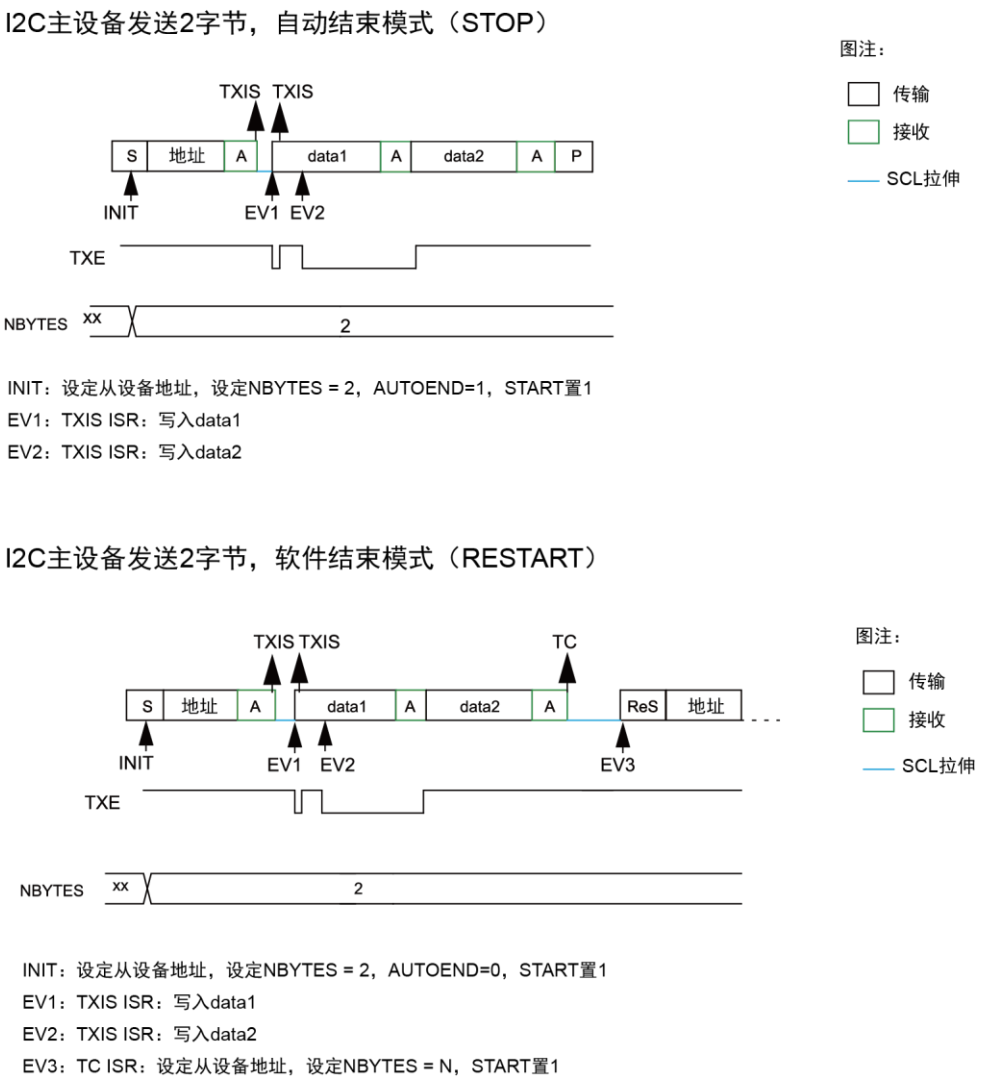


图 101. I2C 主发射器的传输总线图



主接收器

读传输时，在接收到每个字节（即第 8 个 SCL 脉冲）后，RXNE 标志将置 1。如果 I2C_CR1 寄存器中的 RXIE 位置 1，RXNE 事件将生成中断。读取 I2C_RXDR 时，将清零该标志。

如果待接收的数据字节总数大于 255，则必须通过将 I2C_CR2 寄存器中的 RELOAD 位置 1 来选择重载模式。在这种情况下，当 NBYTES[7:0]数据传输完成时，TCR 标志将置 1，并且 SCL 线的低电平将被延展，直到 NBYTES[7:0]被写入非零值。

- 当 RELOAD=0 且 NBYTES[7:0]数据传输完成时：
  - 在自动结束模式（AUTOEND=1）下，接收到最后一个字节后，将自动发送 NACK 和停止位
  - 在软件结束模式（AUTOEND=0）下，接收到最后一个字节后，将自动发送 NACK，TC 标志将置 1 且 SCL 线的低电平将被延展，以便执行以下软件操作：

    可通过将 I2C_CR2 寄存器中的 START 位置 1 并配置适当的从设备地址和待传输字节数来请求发送重复起始位。置位 START 位会将 TC 标志清零，并在总线上发送起始位，后跟从设备地址。

    可通过将 I2C_CR2 寄存器中的 STOP 位置 1 来请求停止位。置位 STOP 位会将 TC 标志清零，并在总线上发送 STOP 位。

图 102. I2C 主接收器的传输序列流程图（N>255 字节）

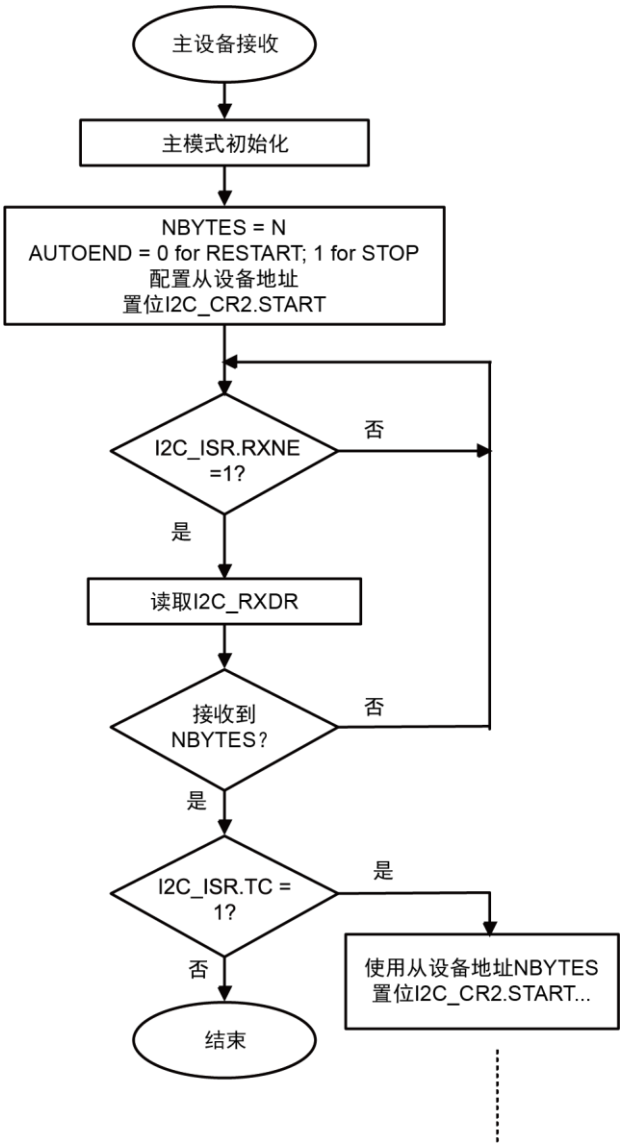


图 103. I2C 主接收器的传输序列流程图（N>255 字节）

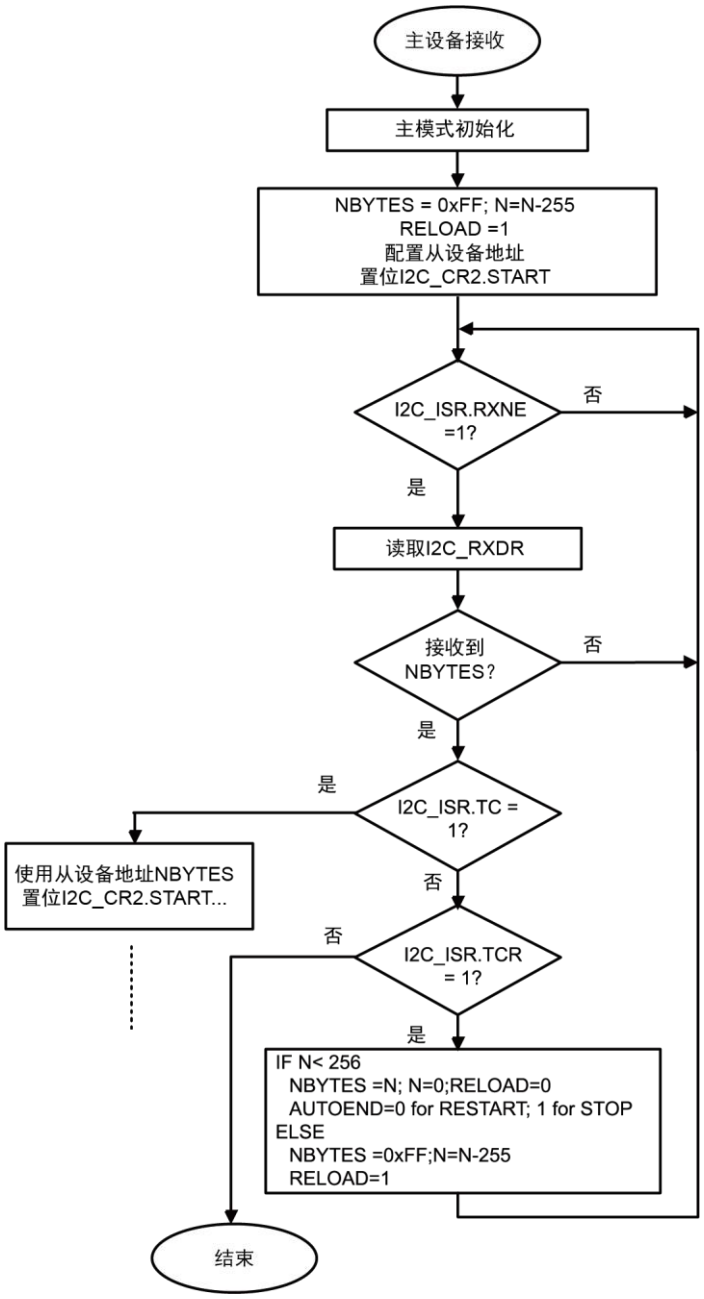
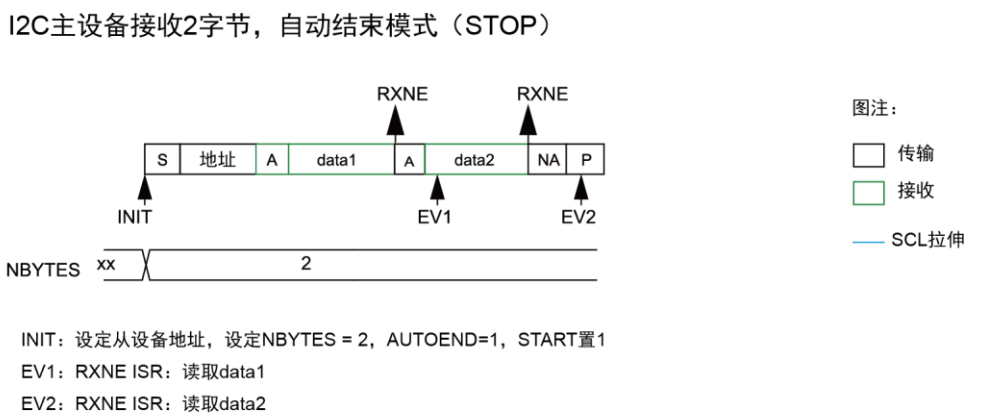


图 104. I²C 主接收器的传输总线图



19.4.9 I2C_TIMINGR 寄存器配置示例

下表提供了如何进行 I2C_TIMINGR 编程以获得与 I²C 规范兼容的时序的示例。

表 58. fI2CCLK = 16 MHz 时的时序设置示例

参数	标准模式（Sm）		快速模式（Fm）	超快速模式（Fm）
	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 125 ns = 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)

参数	标准模式（Sm）		快速模式（Fm）	超快速模式（Fm）
	10 kHz	100 kHz	400 kHz	1000 kHz
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 内部检测延时，SCL 周期 tSCL 大于 t_{SCLL} + t_{SCLH}。为 t_{SCL} 提供的值只是示例。*
2. *t_{SYNC1}+ t_{SYNC2} 的最小值为 4 x t_{I2CCLK}=250 ns。t_{SYNC1}+ t_{SYNC2}=1000 ns 的示例*
3. *t_{SYNC1}+ t_{SYNC2} 的最小值为 4 x t_{I2CCLK}=250 ns。t_{SYNC1}+ t_{SYNC2}=750 ns 的示例*
4. *t_{SYNC1}+ t_{SYNC2} 的最小值为 4 x t_{I2CCLK}=250 ns。t_{SYNC1}+ t_{SYNC2}=500 ns 的示例*

19.4.10 SMBus 特性

仅当支持 SMBus 功能时，才涉及本节内容。请参见第 19.3 节“I2C 实现”。

引言

系统管理总线（SMBus）是一个双线制接口，各器件可通过它在彼此之间或者与系统的其余部分进行通信。它基于 I^2C 操作原理。SMBus 可针对系统和电源管理相关的任务提供控制总线。

该外设与 SMBUS 规范第 2.0 版兼容（http://smbus.org）。

系统管理总线规范涉及三类器件。

- 从设备，用于接收或响应指令
- 主设备，用于发出命令、生成时钟和中止传输
- 主机，一种专用主设备，提供面向系统 CPU 的主接口。主机必须具有主 - 从设备功能，并且必须支持 SMBus 主机通知协议。系统中只允许存在一个主机。

该外设可配置为主设备或从设备，也可配置为主机。

SMBUS 基于 I^2C 规范第 2.1 版。

总线协议

任何给定器件都有十一种可用指令协议。器件既可以在这十一种协议中任选其一，也可以使用全部十一种协议进行通信。这十一种协议分别为快速指令、发送字节、接收字节、写入字节、写入字、读取字节、读取字、过程调用、块读取、块写入以及块写入-块读取过程调用。这些协议应通过用户软件实施。

有关这些协议的详细信息，请参见 SMBus 规范第 2.0 版（http://smbus.org）。

地址解析协议（ARP）

通过为各个从设备动态分配一个新的唯一地址可解决 SMBus 从设备地址冲突的问题。为了提供一种机制来针对地址分配隔离各个器件，各器件必须具有唯一的器件标识符（UDID）。该 128 位数字由软件实现。

该外设支持地址解析协议（ARP）。通过将 I2C_CR1 寄存器中的 SMBDEN 位置 1 来使能 SMBus 器件默认地址（0b1100 001）。ARP 命令应通过用户软件实现。

此外，还将在从模式下执行仲裁以支持 ARP。

有关 SMBus 地址解析协议的详细信息，请参见 SMBus 规范第 2.0 版（http://smbus.org）。

接收到的指令和数据应答控制

SMBus 接收器必须能够对接收到的每个指令或数据进行否定应答。要在从模式下实现 ACK 控制，必须通过将 I2C_CR1 寄存器中的 SBC 位置 1 来使能从字节控制模式。

主机通知协议

该外设通过将 I2C_CR1 寄存器中的 SMBHEN 位置 1 来支持主机通知协议。在这种情况下，主机应答 SMBus 主机地址（0b0001 000）。

使用该协议时，器件用作主设备，而主机用作从设备。

SMBus 报警

器件支持 SMBus ALERT 可选信号。只具备从功能的器件可通过 SMBALERT# 引脚向主机发出信号，指示它想要通信。主机会处理该中断并通过报警响应地址（0b0001 100）同时访问所有 SMBALERT# 器件。只有那些将 SMBALERT# 拉到低电平的器件会应答报警响应地址。

如果配置为从设备（SMBHEN=0），则通过将 I2C_CR1 寄存器中的 ALERTEN 位置 1 来将 SMBA 引脚拉为低电平。这同时还会使能报警响应地址。

如果配置为主机（SMBHEN=1），则当 SMBA 引脚上检测到下降沿且 ALERTEN=1 时，I2C_ISR 寄存器中的 ALERT 标志置 1。如果 I2C_CR1 寄存器中的 ERRIE 位置 1，则生成中断。当 ALERTEN=0 时，即使外部 SMBA 引脚为低电平，ALERT 线也将被视为高电平。

如果无需 SMBus ALERT 引脚，则当 ALERTEN=0 时，SMBA 引脚可用作标准 GPIO。

数据包错误校验

SMBus 规范中引入了数据包错误校验机制来提高可靠性和通信稳定性。数据包错误校验的实施方式是在每次消息传输结束时附加数据包错误代码（PEC）。使用 C(x）= x8 + x2 + x + 1 CRC-8 多项式基于所有消息字节计算 PEC（包括地址和读/写位）。

外设内置硬件 PEC 计算器，能够在接收到的字节与硬件计算的 PEC 不匹配时自动发送不确认。

超时

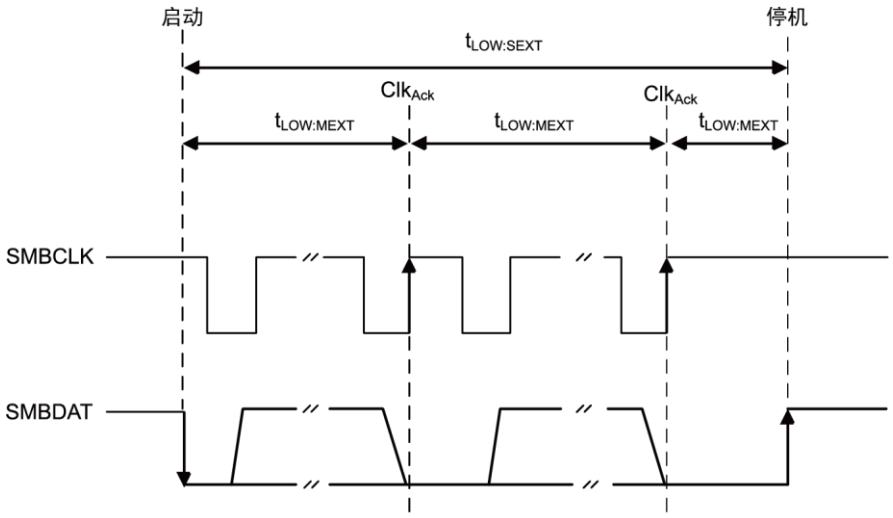
该外设内置了硬件定时器，以便符合 SMBus 规范第 2.0 版中定义的 3 个超时。

表 59. SMBus 超时规范

符号	参数	限值		单位
		最小值	最大值	
$t_{TIMEOUT}$	检测时钟低电平超时	25	35	ms
$t_{LOW:SEXT}$(1)	累积时钟低电平延长时间（从设备）		25	ms
$t_{LOW:MEXT}$(2)	累积时钟低电平延长时间（主设备）		10	ms

1. • *$t_{LOW:SEXT}$ 是指在一个消息中从初始 START 到 STOP，给定从设备被允许延长的时钟周期的累积时间。另一个从设备或主设备也可能延长时钟，导致组合时钟低电平延长时间大于 $t_{LOW:SEXT}$。因此，测量该参数时该全速主设备只寻址一个从设备。*
2. • *$t_{LOW:MEXT}$ 是指在一个消息的每个字节（被定义为 START-to-ACK、ACK-to-ACK 或 ACK-to-STOP）中，主设备被允许延长的时钟周期的累积时间。从设备或另一个主设备也可能延长时钟，导致给定字节的组合时钟低电平时间大于 $t_{LOW:MEXT}$。因此，测量该参数时该全速主设备只寻址一个从设备。*

图 105. $t_{LOW:SEXT}$ 和 $t_{LOW:MEXT}$ 的超时间隔



总线空闲检测

如果主设备检测到时钟和数据信号维持高电平的时间 tIDLE 大于 tHIGH, MAX.，它可假定总线空闲。（参见“表 57. I²C-SMBUS 规范时钟时序”）。

该时序参数已考虑如下情况：主设备已动态添加至总线，但可能尚未检测到 SMBCLK 或 SMBDAT 线上的状态转换。在这种情况下，主设备必须等待足够长的时间，以确定当前未进行传输。外设支持硬件总线空闲检测。

19.4.11 SMBus 初始化

仅当支持 SMBus 功能时，才涉及本节内容。请参见第 19.3 节“I²C 实现”。

除了 I²C 初始化之外，还必须进行一些其它的特定初始化，以便执行 SMBus 通信：

接收到的指令和数据应答控制（从模式）

SMBus 接收器必须能够对接收到的每个指令或数据进行否定应答。要在从模式下实现 ACK 控制，必须通过将 I2C_CR1 寄存器中的 SBC 位置 1 来使能从字节控制模式。参见第 19.4.7 节“I2C 从模式”中的“从字节控制模式”了解更多信息。

特定地址（从模式）

必要时应使能特定的 SMBus 地址。

- 通过将 I2C_CR1 寄存器中的 SMBDEN 位置 1 来使能 SMBus 器件默认地址（0b11001100 001）
- 通过将 I2C_CR1 寄存器中的 SMBHEN 位置 1 来使能 SMBus 主机地址（0b0001 000）。
- 通过将 I2C_CR1 寄存器中的 ALERTEN 位置 1 来使能报警响应地址（0b0001100）。

数据包错误校验

通过将 I2C_CR1 寄存器中的 PECEN 位置 1 来使能 PEC 的计算。然后，借助于硬件字节计数器管理 PEC 传输：I2C_CR2 寄存器中的 NBYTES[7:0]。使能 I²C 之前，必须配置 PECEN 位。

PEC 传输由硬件字节计数器来管理，因此在从模式下连接 SMBus 时必须将 SBC 位置 1。当 PECBYTE 位置 1 且 RELOAD 位清零时，传输完 NBYTES-1 字节的数据后会传输 PEC。如果 RELOAD 置 1，PECBYTE 将不起作用。

注意：使能 I²C 后，不允许更改 PECEN 配置。

表 60. 具有 PEC 配置的 SMBUS

模式	SBC 位	RELOAD 位	AUTOEND 位	PECBYTE 位
主 Tx/Rx NBYTES + PEC+ STOP	x	0	1	1
主 Tx/Rx NBYTES + PEC + ReSTART	x	0	0	1
从 Tx/Rx + PEC	1	0	x	1

超时检测

将 I2C_TIMEOUTR 寄存器中的 TIMOUTEN 和 TEXTEN 位置 1 来使能超时检测。定时器必须按如下方式编程：即在 SMBus 规范第 2.0 版规定的时间最大值之前检测出超时情况。

- $t_{TIMEOUT}$ 检查

为了使能 $t_{TIMEOUT}$ 检查，必须用定时器重载值进行 12 位 TIMEOUTA[11:0]位的编程，以便检查 tTIMEOUT 参数。TIDLE 位必须配置为“0”以便检测 SCL 低电平超时。

然后，通过将 I2C_TIMEOUTR 寄存器中的 TIMOUTEN 位置 1 来使能定时器。

如果 SCL 的低电平持续时间超过（TIMEOUTA+1）x 2048 x t_{I2CCLK}，I2C_ISR 寄存器中的 TIMEOUT 标志将置 1。

请参见“表 61. TIMEOUTA 设置的示例（$t_{TIMEOUT}$ 最大值 = 25 ms）”。

注意：TIMEOUTEN 位置 1 时，不允许更改 TIMEOUTA[11:0]位和 TIDLE 位的配置。

- $t_{LOW:SEXT}$ 和 $t_{LOW:MEXT}$ 检查

必须配置 12 位 TIMEOUTB 定时器以便检查 $t_{LOW:SEXT}$（对于从模式）和 $t_{LOW:MEXT}$（对于主模式），具体取决于外设是被配置为主模式还是从模式。由于标准只规定了最大值，因此可以为这两个参数选择相同的值。

然后，通过将 I2C_TIMEOUTR 寄存器中的 TEXTEN 位置 1 来使能定时器。

如果 SMBus 外设执行 SCL 延展的累积时间超过（TIMEOUTB+1）x 2048 x t_{I2CCLK}，则超时间隔，即 I2C_ISR 寄存器中的 TIMEOUT 标志置 1。

请参见“表 62. TIMEOUTB 置位的示例”。

注意：TEXTEN 位置 1 时，不允许更改 TIMEOUTB 配置。

总线空闲检测

为了使能 t_{IDLE} 检查，必须用定时器重载值进行 12 位 TIMEOUTA[11:0]字段的编程，以便获得 t_{IDLE} 参数。TIDLE 位必须配置为“1”以便检测 SCL 和 SDA 高电平超时。

然后，通过将 I2C_TIMEOUTR 寄存器中的 TIMOUTEN 位置 1 来使能定时器。

如果 SCL 和 SDA 线的高电平持续时间超过（TIMEOUTA+1）x 4 x t_{I2CCLK}，I2C_ISR 寄存器中的 TIMEOUT 标志将置 1。

请参见“表 63. TIMEOUTA 设置的示例（t_{IDLE} 最大值 = 50 µs）”。

注意：TIMEOUTEN 置 1 时，不允许更改 TIMEOUTA 和 TIDLE 配置。

19.4.12 SMBus:I2C_TIMEOUTR 寄存器配置示例

仅当支持 SMBus 功能时，才涉及本节内容。请参见第 19.3 节“I2C 实现”。

- 将 $t_{TIMEOUT}$ 的最长持续时间配置为 25 ms：

表 61. TIMEOUTA 设置的示例（$t_{TIMEOUT}$ 最大值 = 25 ms）

f_{I2CCLK}	TIMEOUTA[11:0]位	TIDLE 位	TIMEOUTEN 位	$t_{TIMEOUT}$
16 MHz	0xC3	0	1	196 x 2048 x 62.5 ns = 25 ms

- 将 $t_{LOW:SEXT}$ 和 $t_{LOW:MEXT}$ 的最长持续时间配置为 8 ms：

表 62. TIMEOUTB 设置的示例

f_{I2CCLK}	TIMEOUTB[11:0]位	TEXTEN 位	$t_{LOW:EXT}$
16 MHz	0xC3	0	196 x 2048 x 62.5 ns = 8 ms

- 将 t_{IDLE} 的最长持续时间配置为 50 µs：

表 63. TIMEOUTA 设置的示例（t_{IDLE} 最大值 = 50 µs）

f_{I2CCLK}	TIMEOUTA[11:0]位	TIDLE 位	TIMEOUTEN 位	t_{IDLE}
16 MHz	0xC7	0	1	200 x 4 x 62.5 ns = 50 µs

19.4.13 SMBus 从模式

仅当支持 SMBus 功能时，才涉及本节内容。请参见第 19.3 节“I2C 实现”。

除了 I^2C 从模式传输管理（参见第 19.4.7 节“I2C 从模式”），还提供了一些额外的软件流程图用于支持 SMBus。

SMBus 从发射器

在 SMBus 模式下使用 IP 时，必须将 SBC 编程为“1”，以便在完成已编程数据字节数的传输后进行 PEC 传输。当 PECBYTE 位置 1 时，NBYTES[7:0]中编程的字节数包含 PEC 传输。在这种情况下，总 TXIS 中断数为 NBYTES-1，如果主设备在完成 NBYTES-1 字节的数据传输后请求传输额外的字节，则将自动发送 I2C_PECR 寄存器的内容。

注意：当 RELOAD 位置 1 时，PECBYTE 位将不起作用。

图 106. SMBus 从发射器的传输序列流程图（N 字节 + PEC）

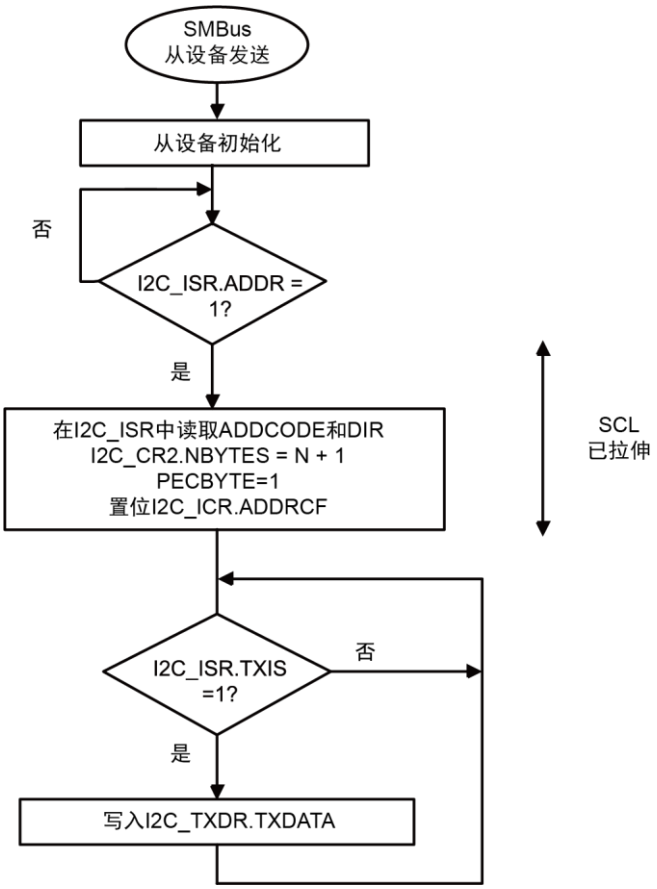
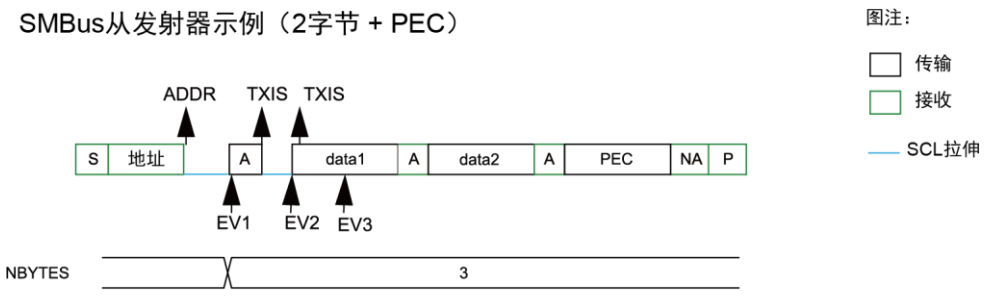


图 107. SMBus 从发射器的传输总线图（SBC=1）



EV1：ADDR ISR：校验ADDCODE，设定NBYTES=3，PECBYTE置1，ADDRCF置1
EV2：TXIS ISR：写入data1
EV3：TXIS ISR：写入data2

SMBus 从接收器

在 SMBus 模式下使用 I²C 时，必须将 SBC 编程为“1”，以便在完成已编程数据字节数的传输后进行 PEC 校验。要对每个字节进行 ACK 控制，必须选择重载模式（RELOAD=1）。

参见第 19.4.7 节“I2C 从模式”中的“从字节控制模式”了解更多信息。

要校验 PEC 字节，必须将 RELOAD 位清零并将 PECBYTE 位置 1。这种情况下，在接收到 NBYTES-1 数据后，将接收到的下一个字节与内部

I2C_PECR 寄存器的内容进行比较。如果比较不匹配，则将自动生成 NACK 信号；如果比较匹配，则将自动生成 ACK 信号，而与 ACK 位的值无关。PEC 字节一经接收，便会像任何其它数据一样复制到 I2C_RXDR 寄存器中，并且 RXNE 标志将置 1。

当 PEC 不匹配时，PECERR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

如果无需 ACK 软件控制，可设置 PECBYTE=1，在同样的写操作下，将 NBYTES 设置为将连续接收的字节数。接收到 NBYTES-1 字节的数据后，会将接收的下一个字节视为 PEC 进行校验。

注意：当 RELOAD 位置 1 时，PECBYTE 位将不起作用。

图 108. SMBus 从接收器的传输序列流程图（N 字节 + PEC）

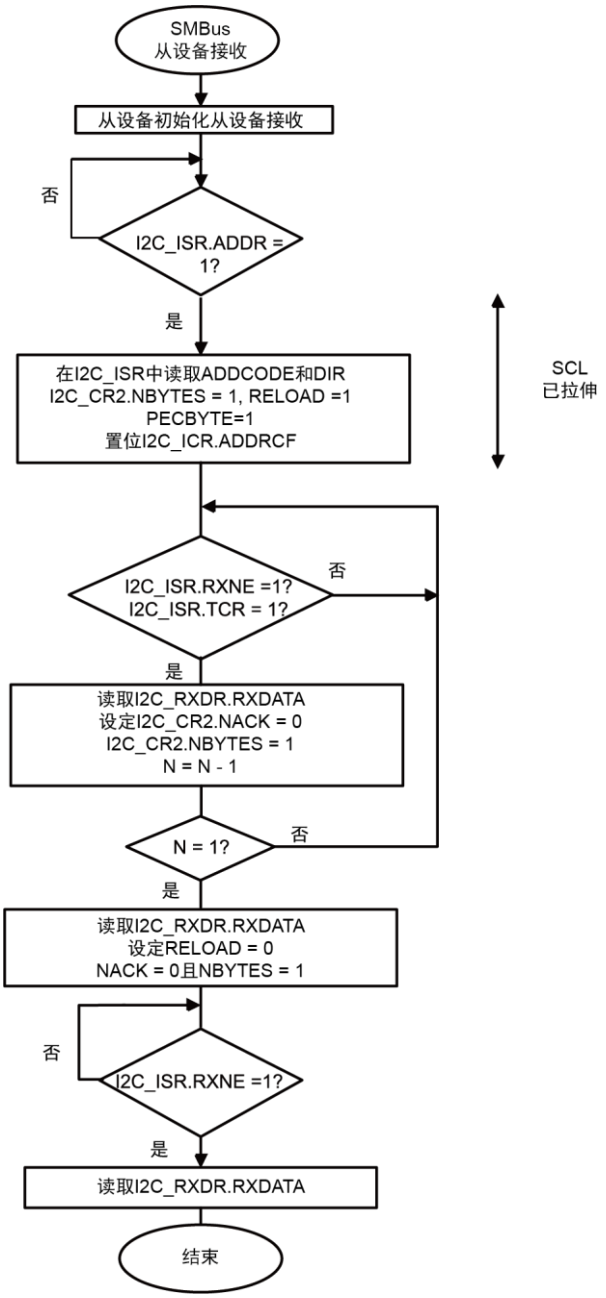
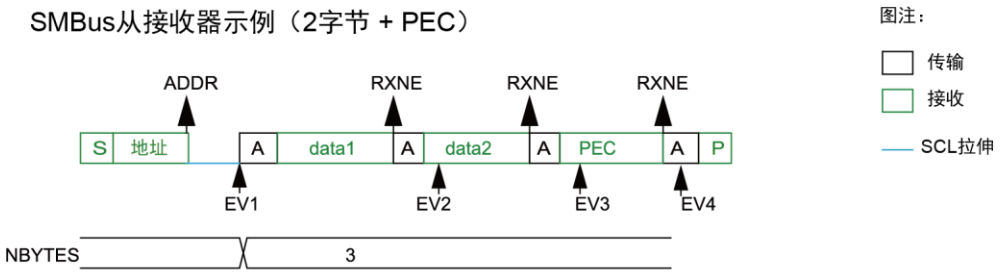
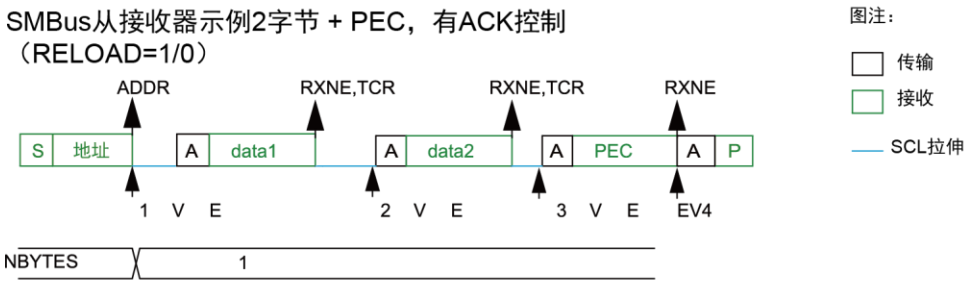


图 109. SMBus 从接收器的总线传输图（SBC=1）



EV1：ADDR ISR：校验ADDCODE和DIR，设定NBYTES = 3，PECBYTE=1，RELOAD=0，ADDRCF置1
EV2：RXNE ISR：读取data1
EV3：RXNE ISR：读取data2
EV4：RXNE ISR：读取PEC



EV1：ADDR ISR：校验ADDCODE和DIR，设定NBYTES = 1，PECBYTE=1，RELOAD=1，ADDRCF置1
EV2：RXNE-TCR ISR：读取data1，设定NACK=0且NBYTES = 1
EV3：RXNE-TCR ISR：读取data2，设定NACK=0，NBYTES = 1且RELOAD=0
EV4：RXNE-TCR ISR：读取PEC

仅当支持 SMSBus 功能时，才涉及本节内容。请参见第 19.3 节“I2C 实现”。

除了 I2C 主模式传输管理（参见第 19.4.8 节“I2C 主模式”），还提供了一些额外的软件流程图用于支持 SMBus。

SMBus 主发射器

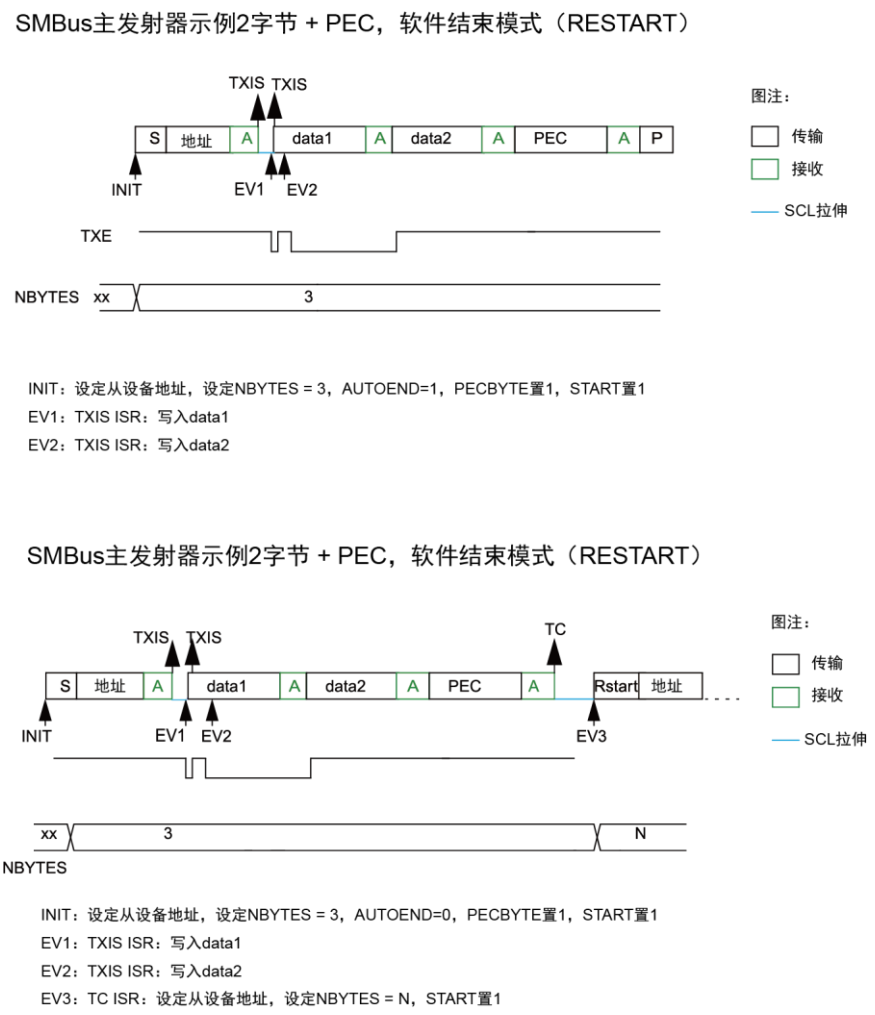
当 SMBus 主设备想要发送 PEC 时，必须在 START 位置 1 前，将 PECBYTE 位置 1 并在 NBYTES[7:0]字段中设置字节数。在这种情况下，TXIS 中断的总数为 NBYTES-1。因此，如果 PECBYTE 位在 NBYTES=0x1 时置 1，则将自动发送 I2C_PECR 寄存器的内容。

如果 SMBus 主设备想要在 PEC 后发送停止位，则应选择自动结束模式（AUTOEND=1）。在这种情况下，传输 PEC 后将自动发送停止位。

如果 SMBus 主设备想要在 PEC 后发送重复起始位，则必须选择软件模式（AUTOEND=0）。在这种情况下，发送 NBYTES-1 字节的数据后，将发送 I2C_PECR 寄存器的内容，TC 标志将在传输完 PEC 之后置 1，SCL 线的低电平时间将延长。必须在 TC 中断子程序中设置重复起始位。

注意：当 RELOAD 位置 1 时，PECBYTE 位将不起作用。

图 110. SMBus 主发射器的总线传输图



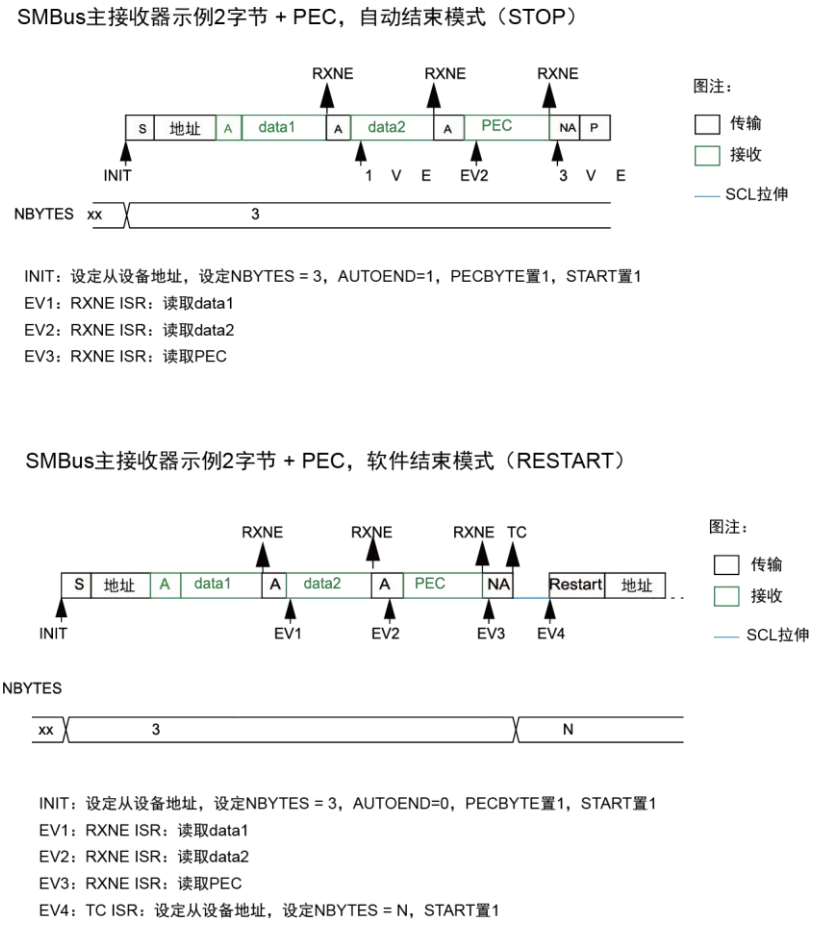
SMBus 主接收器

当 SMBus 主设备想要接收 PEC，并在传输结束后接收 STOP 时，可选择自动结束模式（AUTOEND=1）。将 START 位置 1 之前，必须将 PECBYTE 位置 1 并设置从设备地址。在这种情况下，当接收到 NBYTES-1 字节的数据后，将自动使用 I2C_PECR 寄存器的内容对接收的下一个字节进行校验。PEC 字节（其后跟有停止位）将得到 NACK 响应。

当 SMBus 主接收器想要接收 PEC 字节，并且在传输结束后接收重复起始位时，必须选择软件模式（AUTOEND=0）。将 START 位置 1 之前，必须将 PECBYTE 位置 1 并设置从设备地址。在这种情况下，当接收到 NBYTES-1 字节的数据后，将自动使用 I2C_PECR 寄存器的内容对接收的下一个字节进行校验。接收到 PEC 字节后，TC 标志将置 1，SCL 线的低电平时间将延长。可以在 TC 中断子程序中设置重复起始位。

注意：当 RELOAD 位置 1 时，PECBYTE 位将不起作用。

图 111. SMBus 主接收器的总线传输图



19.4.14 错误条件

下面是可能导致通信失败的错误条件。

总线错误（BERR）

当检测到起始位或停止位但不位于第 9N 个 SCL 时钟脉冲之后时，会检测到总线错误。当 SDA 边沿出现且 SCL 为高电平时，会检测到起始或停止位。

只有当 I²C 在传输过程中用作主设备或被寻址为从设备时（即未处于从模式下的地址阶段），才会将总线错误标志置 1。

在从模式下检测到错位的起始位或重复起始位时，I²C 会像接收到正确的起始位一样进入地址识别状态。

检测到总线错误时，I2C_ISR 寄存器中的 BERR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

仲裁丢失（ARLO）

当 SDA 线上发送高电平但在 SCL 上升沿却采样到低电平时，会检测到仲裁丢失。

- 在主模式下，将在地址阶段、数据阶段和数据应答阶段检测到仲裁丢失。在这种情况下，SDA 线和 SCL 线被释放，起始控制位由硬件清零，主设备自动切换为从模式。
- 在从模式下，将在数据阶段和数据应答阶段检测到仲裁丢失。在这种情况下，传输停止，SCL 和 SDA 线被释放。

检测到仲裁丢失时，I2C_ISR 寄存器中的 ARLO 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

上溢/下溢错误（OVR）

当满足 NOSTRETCH=1 和以下条件时，将在从模式下检测到上溢或下溢错误：

- 接收过程中接收到一个新字节但尚未读取 RXDR 寄存器。接收的新字节丢失，自动发送 NACK 来响应新字节。
- 在发送中：
  - 当 STOPF=1 且应发送第一个数据字节时。TXE=0 时发送 I2C_TXDR 寄存器的内容，否则发送 0xFF。
  - 应发送一个新字节但尚未向 I2C_TXDR 寄存器写入数据时，将发送 0xFF。

检测到上溢或下溢错误时，I2C_ISR 寄存器中的 OVR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

数据包错误校验错误（PECERR）

仅当支持 SMBus 功能时，才涉及本节内容。请参见第 19.3 节“I2C 实现”。

当接收到的 PEC 字节与 I2C_PECR 寄存器的内容不匹配时，将检测到 PEC 错误。接收到错误的 PEC 后，将自动发送 NACK。

检测到 PEC 错误时，I2C_ISR 寄存器中的 PECERR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

超时错误（TIMEOUT）

仅当支持 SMBus 功能时，才涉及本节内容。请参见第 19.3 节“I2C 实现”。

满足以下任何条件均会出现超时错误：

- TIDLE=0 且 SCL 的低电平持续时间达到 TIMEOUTA[11:0]位中定义的时间：这用于检测 SMBus 超时
- TIDLE=1 且 SDA 和 SCL 的高电平持续时间达到 TIMEOUTA[11:0]位中定义的时间：这用于检测总线空闲情况
- 主设备累积时钟低电平延时达到了在 TIMEOUTB[11:0]位（SMBus tLOW:MEXTparameter）定义的时间
- 从设备累积时钟低电平延时达到了在 TIMEOUTB[11:0]位（SMBus tLOW:SEXTparameter）定义的时间。

当在主模式下检测到超时时，将自动发送停止位。

当在从模式下检测到超时时，将自动释放 SDA 和 SCL 线。

检测到超时错误时，I2C_ISR 寄存器中的 TIMEOUT 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

报警（ALERT）

仅当支持 SMBus 功能时，才涉及本节内容。请参见第 19.3 节“I2C 实现”。

当 I^2C 接口配置为主机（SMBHEN=1）、使能了报警引脚检测（ALERTEN=1）并且在 SMBA 引脚上检测到下降沿时，ALERT 标志将置 1。如果 I2C_CR1 寄存器中的 ERRIE 位置 1，则生成中断。

19.4.15 DMA 请求

使用 DMA 发送

将 I2C_CR1 寄存器中的 TXDMAEN 位置 1 可以使能 DMA（直接存储器访问）进行发送。每当 TXIS 位置 1 时，将数据从使用 DMA 外设配置的 SRAM 区（参见第 10 节“DMA 控制器（DMA）”）加载到 I2C_TXDR 寄存器。

只有数据字节采用 DMA 进行传输。

- 在主模式下：通过软件进行初始化、从设备地址、方向、字节数和 START 位的编程（发送的从设备地址不能通过 DMA 传输）。当所有数据均通过 DMA 传输时，必须在起始位置 1 之前初始化 DMA。通过 NBYTES 计数器管理传输结束。

- 从模式：
  - 当 NOSTRETCH=0 时，如果所有数据均通过 DMA 传输，则必须在地址匹配事件之前（或清零 ADDR 之前在 ADDR 中断子程序中）初始化 DMA
  - 当 NOSTRETCH=1 时，必须在地址匹配事件之前初始化 DMA。
- 支持 SMBus 时：PEC 传输由 NBYTES 计数器管理。

参见第 19.4.13 节“SMBus 从模式”中的“SMBus 从发射器”和“SMBus 主发射器”部分了解更多信息。

注意： *如果使用 DMA 进行发送，则无需使能 TXIE 位。*

使用 DMA 接收

将 I2C_CR1 寄存器中的 RXDMAEN 位置 1 可以使能 DMA（直接存储器访问）进行接收。每当 RXNE 位置 1 时，将数据从 I2C_RXDR 寄存器加载到使用 DMA 外设配置的 SRAM 区（参见第 10 节“DMA 控制器（DMA）”）。只有数据字节（包括 PEC）采用 DMA 进行传输。

- 在主模式下，通过软件进行初始化、从设备地址、方向、字节数和 START 位的编程。当所有数据均通过 DMA 传输时，必须在起始位置 1 之前初始化 DMA。通过 NBYTES 计数器管理传输结束。
- 在从模式下，当 NOSTRETCH=0 时，如果所有数据均通过 DMA 传输，则必须在地址匹配事件之前（或清零 ADDR 标志之前在 ADDR 中断子程序中）初始化 DMA。
- 如果支持 SMBus（参见第 19.3 节“I2C 实现”）：通过 NBYTES 计数器管理 PEC 传输。

注意： *如果使用 DMA 进行接收，则无需使能 RXIE 位。*

19.5 I²C 中断

下表给出了 I²C 中断请求列表。

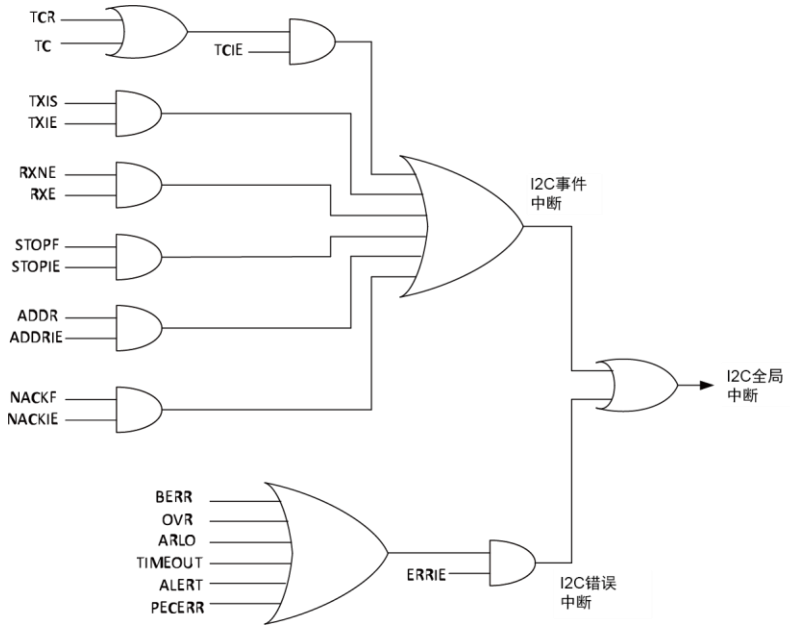
表 64. I²C 中断请求

中断事件	事件标志	事件标志/中断清除方法	中断使能控制位
接收缓冲区非空	RXNE	读取 I2C_RXDR 寄存器	RXIE
发送缓冲区中断状态	TXIS	写入 I2C_TXDR 寄存器	TXIE
停止位检测中断标志	STOPF	写入 STOPCF=1	STOPIE
传输完成等待重载	TCR	写入 I2C_CR2（NBYTES[7:0] [1] 0）	TCIE
传输完成	TC	写入 START=1 或 STOP=1	
地址匹配	ADDR	写入 ADDRCF=1	ADDRIE
NACK 接收	NACKF	写入 NACKCF=1	NACKIE
总线错误	BERR	写入 BERRCF=1	ERRIE
仲裁丢失	ARLO	写入 ARLOCF=1	
上溢/下溢	OVR	写入 OVRCF=1	
PEC 错误	PECERR	写入 PECERRCF=1	
超时/Tlow 错误	TIMEOUT	写入 TIMEOUTCF=1	
SMBus 报警	ALERT	写入 ALERTCF=1	

根据产品实现的不同，上述所有中断既可以共享同一个中断向量（I²C 全局中断），也可以分配到 2 个不同的中断向量上（I²C 事件中断和 I2C 错误中断）。要使能 I²C 中断，需按照以下顺序操作：

- 配置 NVIC 中的 I²C IRQ 通道并将其使能
- 配置 I²C 以生成中断

图 112. I²C 中断映射图



19.6 I²C 寄存器

请参见第 1.3 节“缩略语”获取寄存器说明中所用缩略语的列表。外设寄存器按字（32 位）进行访问。

19.6.1 控制寄存器 1（I2C_CR1）

偏移地址：0x00 复位值：0x0000 0000

访问：无等待周期，正在对该寄存器执行写访问时出现另一个写访问的情况除外。在这种情况下，会在第二个写访问中插入等待周期，直到前一个写访问完成为止。第二个写访问的延时最长可达 2 x PCLK1 + 6 x I2CCLK。

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	Res.	NOSTR ETCH	SBC
								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	Res.	ANF OFF	DNF				ERRIE	TCIE	STOP IE	NACK IE	ADDR IE	RXIE	TXIE	PE
rw	rw		rw	rw				rw	rw	rw	rw	rw	rw	rw	rw

位 31:24	保留，必须保持复位值。
位 23	**PECEN**：PEC 使能。 0：禁止 PEC 计算 1：使能 PEC 计算 注意：如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。 请参见第 19.3 节“I2C 实现”。

位 22	**ALERTEN**：SMBus 报警使能。 从机模式（SMBHEN=0）： 0：释放 SMBA 引脚高电平且报警响应地址头禁用：0001100x 后接 NACK 1：驱动 SMBA 引脚低电平且报警响应地址头使能：0001100x 后接 ACK **主机模式（SMBHEN=1）：** 0：不支持 SMBus 报警引脚（SMBA） 1：支持 SMBus 报警引脚（SMBA） 注意：当 ALERTEN=0 时，SMBA 引脚可用作标准 GPIO。 如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 19.3 节“I2C 实现”。
位 21	**SMBDEN**：SMBus 器件默认地址使能。 0：器件默认地址禁用。不对地址 0b1100001x 应答。 1：器件默认地址使能。对地址 0b1100001x 应答。 注意：如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。 请参见第 19.3 节“I2C 实现”。
位 20	**SMBHEN**：SMBus 主机地址使能。 0：主机地址禁用。不对地址 0b0001000x 应答。 1：主机地址使能。对地址 0b0001000x 应答。 注意：如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。 请参见第 19.3 节“I2C 实现”。
位 19	**GCEN**：广播呼叫使能。 0：禁止广播呼叫。不对地址 0b00000000 应答。 1：使能广播呼叫。对地址 0b00000000 应答。
位 18	保留，必须保持复位值。
位 17	**NOSTRETCH**：禁止时钟延长。 该位用于在从模式下禁止时钟延长。它在主模式下必须保持清零。 0：使能时钟延长 1：禁止时钟延长 注意：该位只能在 I2C 禁止时（PE = 0）编程。
位 16	**SBC**：从设备字节控制。 该位用于在从设备模式下使能硬件字节控制。 0：从设备字节控制禁用 1：从设备字节控制使能
位 15	**RXDMAEN**：DMA 接收请求使能 0：针对接收禁止 DMA 模式 1：针对接收使能 DMA 模式
位 14	**TXDMAEN**：DMA 发送请求使能。 0：针对发送禁止 DMA 模式 1：针对发送使能 DMA 模式
位 13	保留，必须保持复位值。
位 12	**ANFOFF**：模拟噪声滤波器关闭。 0：使能模拟噪声滤波器 1：禁止模拟噪声滤波器 注意：该位只能在 I2C 禁止时（PE = 0）编程。
位 11:8	**DNF[3:0]**：数字噪声滤波器。 这些位用于配置 SDA 和 SCL 输入端的数字噪声滤波器。数字滤波器过滤长度至多 DNF[3:0] * tI2CCLK 的尖

	峰。 0000：数字滤波器禁用 0001：数字滤波器使能且滤波能力最高 1 tI2CCLK … 1111：数字滤波器使能且滤波能力最高 15 tI2CCLK 注意：如果模拟滤波器也已使能，数字滤波将叠加在模拟滤波之上。 该滤波器只能在 I2C 禁止时（PE = 0）编程。
位 7	**ERRIE**：错误中断使能。 0：错误检测中断禁用 1：错误检测中断使能 注意：以下任一错误均会生成中断：仲裁丢失（ARLO） 总线错误检测（BERR） 上溢/下溢（OVR） 超时检测（TIMEOUT） PEC 错误检测（PECERR） 报警引脚事件检测（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 注意：以下任意事件都会生成中断：传输完成（TC）。 传输完成等待重载（TCR）。
位 5	**STOPIE**：STOP 检测中断使能。 0：停止检测（STOPF）中断禁用 1：停止检测（STOPF）中断使能
位 4	**NACKIE**：不应答接收中断使能。 0：不应答（NACKF）接收中断禁用 1：不应答（NACKF）接收中断使能
位 3	**ADDRIE**：地址匹配中断使能（仅从模式）。 0：地址匹配（ADDR）中断禁用 1：地址匹配（ADDR）中断使能
位 2	**RXIE**：RX 中断使能。 0：接收（RXNE）中断禁用 1：接收（RXNE）中断使能
位 1	**TXIE**：TX 中断使能。 0：发送（TXIS）中断禁用 1：发送（TXIS）中断使能
位 0	**PE**：使能外设。 0：禁止外设 1：使能外设 注意：当 PE=0 时，将释放 I2C SCL 线和 SDA 线。内部状态机和状态位均恢复为复位值。清零时，PE 必须保持低电平状态至少 3 个 APB 时钟周期。

19.6.2 控制寄存器 2（I2C_CR2）

偏移地址：0x04

复位值：0x0000 0000

访问：无等待周期，正在对该寄存器执行写访问时出现另一个写访问的情况除外。在这种情况下，会在第二个写访问中插入等待周期，直到前一个写访问完成为止。第二个写访问的延时最长可达 2 x PCLK1 + 6 x I2CCLK。

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，必须保持复位值。
位 26	**PECBYTE**：数据包错误校验字节。 此位由软件置 1，并可在 PEC 传输完成时、接收到停止位或匹配地址时、或者 PE=0 时由硬件清零。 0：不传输 PEC 1：请求 PEC 发送/接收 注意：向该位写入“0”不起作用。 当 RELOAD 置 1 时，该位不起作用。 当 SBC=0 时，该位在从模式下不起作用。 如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 19.3 节“I2C 实现”。
位 25	**AUTOEND**：自动结束模式（主模式）。此位由软件置 1 和清零。 0：软件结束模式：当 NBYTES 数据传输完成时，TC 标志将置 1,SCL 的低电平时间将延长直到相应软件操作结束。 1：自动结束模式：当 NBYTES 数据传输完成时，将自动发送停止位。 注意：在从模式下，或当 RELOAD 位置 1 时，该位不起作用。
位 24	**RELOAD**：NBYTES 重载模式。 此位由软件置 1 和清零。 0：NBYTES 数据传输后（随后是 STOP 或 RESTART）传输完成 1：NBYTES 数据传输后传输未完成（NBYTES 重载）。当 NBYTES 数据传输完成时，TCR 标志将置 1，SCL 的低电平时间将延长直到相应软件操作结束。
位 23:16	**NBYTES[7:0]**：字节数。 在此设置待发送/接收的字节数。在从模式下，当 SBC=0 时，该字段为无关字段。 注意：START 位置 1 时，不允许更改这些位。
位 15	**NACK**：NACK 生成（从模式）。 此位由软件置 1，并可在发送 NACK 时、接收到停止位或匹配地址时、或者 PE=0 时由硬件清零。 0：在当前接收的字节后发送 ACK。 1：在当前接收的字节后发送 NACK。 注意：向该位写入“0”不起作用。 该位仅在从模式下使用：在主接收器模式下，无论 NACK 位的值为何，最后一个字节（后跟停止位或重复起始位）后都将自动生成 NACK。 当从接收器 NOSTRETCH 模式下发生上溢时，无论 NACK 位的值为何，都将自动生成 NACK。 使能硬件 PEC 校验时（PECBYTE=1），PEC 应答值与 NACK 值无关。

位 14	**STOP**：停止生成（主模式）。 该位由软件置 1，并可在检测到停止位时或 PE = 0 时由硬件清零。 **在主模式下：** 0：不停止生成 1：当前字节传输后停止生成 注意：向该位写入“0”不起作用。
位 13	**START**：开始生成。 该位由软件置 1，并可在发送起始位（后跟地址序列）之后、发生仲裁丢失时、出现超时错误时、或者 PE = 0 时由硬件清零。它也可由软件清零，方法是向 I2C_ICR 寄存器中的 ADDRCF 位写入“1”。 0：不开始生成 1：重新开始/开始生成： • 如果 I^2C 已处于主模式下且 AUTOEND = 0，则将该位置 1 会在 NBYTES 传输结束后且 RELOAD=0 的情况下生成重复起始位。 • 否则，在总线空闲后，此位置 1 会生成 START 状态。 注意：向该位写入“0”不起作用。 即使总线繁忙或 I^2C 处于从模式，也可将 START 位置 1。当 RELOAD 置 1 时，该位不起作用。
位 12	**HEAD10R**：10 位地址头只读方向（主接收器模式）。 0：主设备发送完整的 10 位从设备地址读序列：起始位 + 带写方向的 2 字节 10 位地址 + 重复起始位 + 带读方向的 10 位地址的前 7 位 1：主设备只发送 10 位地址的前 7 位,后跟读方向 注意：START 位置 1 时，不允许更改此位。
位 11	**ADD10**：10 位寻址模式（主模式）。 0：主设备工作在 7 位寻址模式下 1：主设备工作在 10 位寻址模式下 注意：START 位置 1 时，不允许更改此位。
位 10	**RD_WRN**：传输方向（主模式）。 0：主设备请求写传输 1：主设备请求读传输 注意：START 位置 1 时，不允许更改此位。
位 9:8	**SADD[9:8]**：从设备地址位 9:8（主模式）。 **在 7 位寻址模式（ADD10 = 0）下：**这些位无意义 **在 10 位寻址模式（ADD10 = 1）下：** 这些位应写入待发送从设备地址的第 8 和第 9 位 注意：START 位置 1 时，不允许更改这些位。
位 7:1	**SADD[7:1]**：从设备地址位 7:1（主模式）。 **在 7 位寻址模式（ADD10 = 0）下：** 这些位应写入待发送的 7 位从设备地址 **在 10 位寻址模式（ADD10 = 1）下：** 这些位应写入待发送从设备地址的第 1 和第 7 位 注意：START 位置 1 时，不允许更改这些位。
位 0	**SADD0**：从设备地址位 0（主模式）。 **在 7 位寻址模式（ADD10 = 0）下：**该位无意义 **在 10 位寻址模式（ADD10 = 1）下：** 该位应写入待发送从设备地址的第 0 位

	注意：START 位置 1 时，不允许更改这些位。

19.6.3 设备自身地址 1 寄存器（I2C_OAR1）

偏移地址：0x08

复位值：0x0000 0000

访问：无等待周期，正在对该寄存器执行写访问时出现另一个写访问的情况除外。在这种情况下，会在第二个写访问中插入等待周期，直到前一个写访问完成为止。第二个写访问的延时最长可达 2 x PCLK1 + 6 x I2CCLK。

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，必须保持复位值。
位 15	**OA1EN**：设备自身地址 1 使能。 0：设备自身地址 1 禁用。不对接收的从设备地址 OA1 应答。 1：设备自身地址 1 使能。对接收的从设备地址 OA1 应答。
位 14:11	保留，必须保持复位值。
位 10	**OA1MODE**：设备自身地址 1 10 位模式。 0：设备自身地址 1 为 7 位地址 1：设备自身地址 1 为 10 位地址 注意：仅可在 OA1EN=0 时写入该位。
位 9:8	**OA1[9:8]**：接口地址。 7 位寻址模式：无意义 10 位寻址模式：地址位 9:8 注意：仅可在 OA1EN=0 时写入这些位。
位 7:1	**OA1[7:1]**：地址的接口地址位 7:1。 注意：仅可在 OA1EN=0 时写入这些位。
位 0	**OA1[0]**：接口地址。 7 位寻址模式：无意义 10 位寻址模式：地址位 0 注意：仅可在 OA1EN=0 时写入该位。

19.6.4 设备自身地址 2 寄存器（I2C_OAR2）

偏移地址：0x0C

复位值：0x0000 0000

访问：无等待周期，正在对该寄存器执行写访问时出现另一个写访问的情况除外。在这种情况下，会在第二个写访问中插入等待周期，直到前一个写访问完成为止。第二个写访问的延时最长可达 2 x PCLK1 + 6 x I2CCLK。

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，必须保持复位值。
位 15	**OA2EN**：设备自身地址 2 使能。 0：设备自身地址 2 禁用。不对接收的从设备地址 OA2 应答。 1：设备自身地址 2 使能。对接收的从设备地址 OA2 应答。
位 14:11	保留，必须保持复位值。
位 10:8	**OA2MSK[2:0]**：设备自身地址 2 屏蔽。 000：不屏蔽 001：OA2[1] 被屏蔽且无意义。仅比较 OA2[7:2]。 010：OA2[2:1] 被屏蔽且无意义。仅比较 OA2[7:3]。011:OA2[3:1] 被屏蔽且无意义。仅比较 OA2[7:4]。100:OA2[4:1]被屏蔽且无意义。仅比较 OA2[7:5]。101:OA2[5:1]被屏蔽且无意义。仅比较 OA2[7:6]。110:OA2[6:1]被屏蔽且无意义。仅比较 OA2[7]。 111：OA2[7:1]被屏蔽且无意义。不进行比较，对接收到的全部 7 位地址（保留位除外）应答。 注意：仅可在 OA2EN=0 时写入这些位。 只要 OA2MSK 不等于 0，即使比较匹配，也不会对保留的 I2C 地址（0b0000xxx 和 0b1111xxx）应答。
位 7:1	**OA2[7:1]**：地址的接口地址位 7:1。 注意：仅可在 OA2EN=0 时写入这些位。
位 0	保留，必须保持复位值。

19.6.5 时序寄存器（I2C_TIMINGR）

偏移地址：0x10

复位值：0x0000 0000

访问：无等待周期

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。 该字段用于预分频 I2CCLK，以便生成时钟周期 t_{PRESC} 供数据建立和保持计数器以及 SCL 高和低电平计数器使用。 t_{PRESC} =（PRESC+1）x tI2CCLK

位 27:24	保留，必须保持复位值。
位 23:20	**SCLDEL[3:0]**：数据建立时间。 该字段用于在 SDA 边沿与 SCL 上升沿之间生成延时 t_{SCLDEL}。 t_{SCLDEL} =（SCLDEL+1）x t_{PRESC} 注：t_{SCLDEL} 用于生成 $t_{SU:DAT}$ 时序。
位 19:16	**SDADEL[3:0]**：数据保持时间。 在发送模式下，该字段用于在 SCL 下降沿和 SDA 边沿之间生成延时 t_{SDADEL}。 t_{SDADEL}= SDADEL x t_{PRESC} 注意：SDADEL 用于生成 $t_{HD:DAT}$ 时序。
位 15:8	**SCLH[7:0]**：SCL 高电平周期（主模式）。 该字段用于在主模式下生成 SCL 高电平周期。t_{SCLH} =（SCLH+1）x t_{PRESC}。 注意：SCLH 还用于生成 $t_{SU:STO}$ 和 $t_{HD:STA}$ 时序。
位 7:0	**SCLL[7:0]**：SCL 低电平周期（主模式）。 该字段用于在主模式下生成 SCL 低电平周期。t_{SCLL} =（SCLL+1）x t_{PRESC}。 注意：SCLL 还用于生成 t_{BUF} 和 $t_{SU:STA}$ 时序。

注意： *该寄存器必须在 I^2C 禁止时（PE = 0）进行配置。*

19.6.6 超时寄存器（I2C_TIMEOUTR）

偏移地址：0x14

复位值：0x0000 0000

访问：无等待周期，正在对该寄存器执行写访问时出现另一个写访问的情况除外。在这种情况下，会在第二个写访问中插入等待周期，直到前一个写访问完成为止。第二个写访问的延时最长可达 2 x PCLK1 + 6 x I2CCLK。

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

位 31	**TEXTEN**：延长时钟超时使能。 0：延长时钟超时检测禁用 1：延长时钟超时检测使能。当 I^2C 接口执行 SCL 延展的累积时间超过 $t_{LOW:EXT}$ 时，将检测到超时错误（TIMEOUT=1）。
位 30:28	保留，必须保持复位值。
位 27:16	**TIMEOUTB[11:0]**：总线超时 B。 该字段用于配置累积时钟延展超时： 在主模式下检测到主设备累积时钟低电平延长时间（$t_{LOW:MEXT}$），在从模式下检测到从设备累积时钟低电平延长时间（$t_{LOW:SEXT}$），$t_{LOW:EXT}$=（TIMEOUTB+1）x 2048 x t_{I2CCLK}。 注意：仅可在 TEXTEN=0 时写入这些位。
位 15	**TIMOUTEN**：时钟超时使能。 0：SCL 超时检测禁用 1：SCL 超时检测使能：当 SCL 为低电平的时间超过 $t_{TIMEOUT}$（TIDLE=0）或为高电平的时间超过 t_{IDLE}（TIDLE=1）时，会检测到超时错误（TIMEOUT=1）

位 14:13	保留，必须保持复位值。
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 用于检测 SCL 和 SDA 高电平超时（总线空闲条件） 注意：仅可在 TIMOUTEN=0 时写入该位。
位 11:0	**TIMEOUTA[11:0]**：总线超时 A。该字段用于配置： • SCL 低电平超时状态 $t_{TIMEOUT}$，当 TIDLE=0 $t_{TIMEOUT}$ =（TIMEOUTA+1）x 2048 xt_{I2CCLK} 时 • 总线空闲状态（SCL 和 SDA 均为高电平），当 TIDLE=1 t_{IDLE}=（TIMEOUTA+1）x 4 xt_{I2CCLK} 时 注意：仅可在 TIMOUTEN=0 时写入这些位。

注意： *如果不支持 SMBus 功能，该寄存器保留，并由硬件强制为“0x00000000”。请参见第 19.3 节 “I2C 实现”。*

19.6.7 中断和状态寄存器（I2C_ISR）

偏移地址：0x18

复位值：0x0000 0001

访问：无等待周期

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

位 31:24	保留，必须保持复位值。
位 23:17	**ADDCODE[6:0]**：地址匹配代码（从模式）。 发生地址匹配事件时（ADDR = 1），这些位更新为接收到的地址。 在 10 位地址的情况下，ADDCODE 提供 10 位头文件，后跟地址的 2 个 MSB。
位 16	**DIR**：传输方向（从模式）。 该标志在发生地址匹配事件时（ADDR=1）更新。 0：写传输，从设备进入接收器模式 1：读传输，从设备进入发射器模式
位 15	**BUSY**：总线忙碌。 该标志指示总线上正在进行通信。当检测到 START 状态时，该位由硬件置 1。当检测到停止位或 PE = 0 时，该位由硬件清零。
位 14	保留，必须保持复位值。
位 13	**ALERT**：SMBus 报警。 当 SMBHEN=1（SMBus 主机配置）、ALERTEN=1 且在 SMBA 引脚上检测到 SMBALERT 事件（下降沿）时，该标志由硬件置 1。该位由软件清零，方法是将 ALERTCF 位置 1。 注意：当 PE=0 时，该位由硬件清零。 如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 19.3 节“I2C 实现”。
位 12	**TIMEOUT**：超时或 t_{LOW} 检测标志。 发生超时或延长时钟超时时，该标志由硬件置 1。该位由软件清零，方法是将 TIMEOUTCF 位置 1。 注意：当 PE=0 时，该位由硬件清零。

	如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 19.3 节“I2C 实现”。
位 11	**PECERR**：接收中的 PEC 错误。 当接收到的 PEC 与 PEC 寄存器的内容不匹配时，该标志由硬件置 1。接收到错误的 PEC 后，将自动发送 NACK。该位由软件清零，方法是将 PECCF 位置 1。 注意：当 PE=0 时，该位由硬件清零。 如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 19.3 节“I2C 实现”。
位 10	**OVR**：上溢/下溢（从模式）。 在从模式下且 NOSTRETCH=1 时，如果发生上溢/下溢错误，该标志由硬件置 1。该位由软件清零，方法是将 OVRCF 位置 1。 注意：当 PE=0 时，该位由硬件清零。
位 9	**ARLO**：仲裁丢失。 发生仲裁丢失时，该标志由硬件置 1。该位由软件清零，方法是将 ARLOCF 位置 1。 注意：当 PE=0 时，该位由硬件清零。
位 8	**BERR**：总线错误。 当检测到错位的起始位或停止位，而外设也参与传输时，该标志由硬件置 1。在从模式下的地址阶段，该标志不会置 1。该位由软件清零，方法是将 BERRCF 位置 1。 注意：当 PE=0 时，该位由硬件清零。
位 7	**TCR**：传输完成等待重载。 当 RELOAD=1 且 NBYTES 数据传输完成时，该标志由硬件置 1。当 NBYTES 写入一个非零值时，该标志由软件清零。 注意：当 PE=0 时，该位由硬件清零。 该标志单独用于主模式，SBC 位置 1 时单独用于从模式。
位 6	**TC**：传输完成（主模式）。 当 RELOAD=0、AUTOEND=0 且 NBYTES 数据传输完成时，该标志由硬件置 1。当 START 位或 STOP 位置 1 时，该标志由软件清零。 注意：当 PE=0 时，该位由硬件清零。
位 5	**STOPF**：停止位检测标志。 当在总线上检测到停止位，且外设也参与本次传输时，该标志由硬件置 1： • 外设作为主设备，该位置位的前提是外设已经发出停止位。 • 外设作为从设备，该位置位的前提条件是此次传输的寻址对象就是该外设。 该位由软件清零，方法是将 STOPCF 位置 1。 注意：当 PE=0 时，该位由硬件清零。
位 4	**NACKF**：不应答接收标志。 传输完字节后接收到 NACK 时，该标志由硬件置 1。该位由软件清零，方法是将 NACKCF 位置 1。 注意：当 PE=0 时，该位由硬件清零。
位 3	**ADDR**：地址匹配（从模式）。 接收到的地址与使能的从设备地址之一匹配时，该位由硬件置 1。该位由软件清零，方法是将 ADDRCF 位置 1。 注意：当 PE=0 时，该位由硬件清零。

位 2	**RXNE**：接收数据寄存器不为空（接收器）。 当接收到的数据已复制到 I2C_RXDR 寄存器且准备就绪可供读取时，该位由硬件置 1。读取 I2C_RXDR 时，将清零该位。 注意：当 PE=0 时，该位由硬件清零。
位 1	**TXIS**：发送中断状态（发射器）。 当 I2C_TXDR 寄存器为空时，该位由硬件置 1，待发送的数据必须写入 I2C_TXDR 寄存器。下一个待发送的数据写入 I2C_TXDR 寄存器时，该位被清零。 该位只能在 NOSTRETCH=1 时由软件写入“1”，以生成 TXIS 事件（TXIE=1 时为中断，TXDMAEN=1 时为 DMA 请求）。 注意：当 PE=0 时，该位由硬件清零。
位 0	**TXE**：发送数据寄存器为空（发射器）。 当 I2C_TXDR 寄存器为空时，该位由硬件置 1。下一个待发送的数据写入 I2C_TXDR 寄存器时，该位被清零。 该位可由软件写入“1”，以刷新发送数据寄存器 I2C_TXDR。 注意：当 PE=0 时，该位由硬件置 1。

19.6.8 中断清零寄存器（I2C_ICR）

偏移地址：0x1C

复位值：0x0000 0000

访问：无等待周期

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	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，必须保持复位值。
位 13	**ALERTCF**：报警标志清零。 将 1 写入此位时，I2C_ISR 寄存器中的 ALERT 标志将清零。 注意：如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。 请参见第 19.3 节“I2C 实现”。
位 12	**TIMOUTCF**：超时检测标志清零。 将 1 写入此位时，I2C_ISR 寄存器中的 TIMEOUT 标志将清零。 注意：如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。 请参见第 19.3 节“I2C 实现”。
位 11	**PECCF**：PEC 错误标志清零。 将 1 写入此位时，I2C_ISR 寄存器中的 PECERR 标志将清零。 注意：如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。 请参见第 19.3 节“I2C 实现”。
位 10	**OVRCF**：上溢/下溢标志清零。 将 1 写入此位时，I2C_ISR 寄存器中的 OVR 标志将清零。
位 9	**ARLOCF**：仲裁丢失标志清零。

	将 1 写入此位时，I2C_ISR 寄存器中的 ARLO 标志将清零。
位 8	**BERRCF**：总线错误标志清零。 将 1 写入此位时，I2C_ISR 寄存器中的 BERRF 标志将清零。
位 7:6	保留，必须保持复位值。
位 5	**STOPCF**：停止检测标志清零。 将 1 写入此位时，I2C_ISR 寄存器中的 STOPF 标志将清零。
位 4	**NACKCF**：不应答标志清零。 将 1 写入此位时，I2C_ISR 寄存器中的 ACKF 标志将清零。
位 3	**ADDRCF**：地址匹配标志清零。 将 1 写入此位时，I2C_ISR 寄存器中的 ADDR 标志将清零。将 1 写入此位时，I2C_CR2 寄存器中的 START 位也将清零。
位 2:0	保留，必须保持复位值。

19.6.9 PEC 寄存器（I2C_PECR）

偏移地址：0x20

复位值：0x0000 0000

访问：无等待周期

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	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，必须保持复位值。
位 7:0	**PEC[7:0]**：数据包错误校验寄存器。 当 PECEN=1 时，此字段包含内部 PEC。当 PE=0 时，PEC 由硬件清零。

注意： *如果不支持 SMBus 功能，该寄存器保留，并由硬件强制为“0x00000000”。请参见第 19.3 节 “I2C 实现”。*

19.6.10 接收数据寄存器（I2C_RXDR）

偏移地址：0x24

复位值：0x0000 0000

访问：无等待周期

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	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，必须保持复位值。
位 7:0	**RXDATA[7:0]**：8 位接收数据。 从 I²C 总线接收到的数据字节。

19.6.11 发送数据寄存器（I2C_TXDR）

偏移地址：0x28

复位值：0x0000 0000

访问：无等待周期

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	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，必须保持复位值。
位 7:0	**TXDATA[7:0]**：8 位发送数据。 要发送到 I²C 总线的数据字节。 注意：仅可在 TXE=1 时写入这些位。

19.7 I2C 寄存器映射

下表提供了 I²C 寄存器映射和复位值。

表 65. I²C 寄存器映射

偏移	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
0x0	I2C_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PECEN	ALERTEN	SMBDEN	SMBHEN	GCEN	Res.	NOSTRETCH	SBC	RXDMAEN	TXDMAEN	Res.	ANFOFF	DNF[3:0]				ERRIE	TCIE	STOPIE	NACKIE	ADDRIE	RXIE	TXIE	PE
	复位值									0	0	0	0	0		0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0
0x4	I2C_CR2	Res.	Res.	Res.	Res.	Res.	PECBYTE	AUTOEND	RELOAD	NBYTES[7:0]								NACK	STOP	START	HEAD10R	ADD10	RD_WRN	SADD[9:0]									
	复位值						0	0	0	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	I2C_OAR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OA1EN	Res.	Res.	Res.	Res.	OA1MODE	OA1[9:0]									
	复位值																	0					0	0	0	0	0	0	0	0	0	0	0
0xC	I2C_OAR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OA2EN	Res.	Res.	Res.	Res.	OA2MSK[2:0]			OA2[7:1]							Res.
	复位值																	0					0	0	0	0	0	0	0	0	0	0	
0x10	I2C_TIMINGR	PRESC[3:0]				Res.	Res.	Res.	Res.	SCLDEL[3:0]				SDADEL[3:0]				SCLH[7:0]								SCLL[7:0]							
	复位值	0	0	0	0					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	I2C_ TIMEOUTR	TEXTEN	Res.	Res.	Res.	TIMEOUTB[11:0]												TIMOUTEN	Res.	TIDLE	TIMEOUTA[11:0]												
	复位值	0				0	0	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	I2C_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDCODE[6:0]							DIR	BUSY	Res.	ALERT	TIMEOUT	PECERR	OVR	ARLO	BERR	TCR	TC	STOPF	NACKF	ADDR	RXNE	TXIS	TXE
	复位值									0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	1

偏移	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
0x1C	I2C_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.
	复位值																			0	0	0	0	0	0			0	0	0			
0x20	I2C_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[7:0]							
	复位值																									0	0	0	0	0	0	0	0
0x24	I2C_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[7:0]							
	复位值																									0	0	0	0	0	0	0	0
0x28	I2C_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[7:0]							
	复位值																									0	0	0	0	0	0	0	0

请参见“表 3. BlueNRG-LP 存储器映射和外设寄存器边界地址”，了解寄存器边界地址。

20 通用同步/异步收发器（USART）

20.1 USART 简介

通用同步异步收发器（USART）能够灵活地与外部设备进行全双工数据交换，满足外部设备对工业标准 NRZ 异步串行数据格式的要求。USART 通过小数波特率发生器提供了多种波特率。

它支持同步单向通信和半双工单线通信。还支持 LIN（局域互连网络）、智能卡协议与 IrDA（红外线数据协会）SIR ENDEC 规范，以及调制解调器操作（CTS/RTS）。它还支持多处理器通信。

对于多缓冲区配置，可使用 DMA（直接存储器访问）进行高速数据通信。

20.2 USART 主要特性

- 全双工异步通信
- NRZ 标准格式（标记/空格）
- 可配置为 16 倍过采样或 8 倍过采样，因而为速度容差与时钟容差的灵活配置提供了可能
- 波特率发生器系统
- 两个内部 FIFO 用于发送和接收数据，可通过软件使能/禁用。FIFO 具有 FIFO 状态标志
- 在 16MHz 时钟频率和 8 倍过采样的情况下，最高可实现 2Mbit/s 的共用可编程发送和接收波特率
- 具有专用内核时钟的双时钟域可使波特率编程不受 PCLK 重新编程的影响
- 自动波特率检测
- 数据字长度可编程（7 位、8 位或 9 位）
- 可编程的数据顺序，最先移位 MSB 或 LSB
- 停止位可配置（支持 1 个或 2 个停止位）
- 用于同步通信的同步主/从模式和时钟输出/输入
- SPI 从设备发送下溢错误标志
- 单线半双工通信
- 使用 DMA 进行连续通信
- 使用中央 DMA 在预留的 SRAM 缓冲区中收/发字节
- 发射器和接收器有单独的使能位
- 发送和接收的单独信号极性控制
- Tx/Rx 引脚配置可交换
- 调制解调器和 RS-485 收发器的硬件流控制
- 通信控制/错误检测标志
- 奇偶校验控制：
  - 发送奇偶校验位
  - 检查接收的数据字节的奇偶性
- 中断源标志位
- 多处理器通信
- 从静默模式唤醒（通过空闲线检测或地址标记检测）

20.3 USART 扩展特性

- LIN 主模式同步断路发送功能和 LIN 从模式断路检测功能
  - 对 USART 进行 LIN 硬件配置时可生成 13 位停止符号和检测 10/11 位停止符号
- 正常模式下支持 3/16 位持续时间的 IrDA SIR 编解码器
- 智能卡模式
  - 对于 ISO/IEC 7816-3 标准中定义的智能卡，支持 T=0 和 T=1 异步协议
  - 0.5 和 1.5 个停止位用于智能卡操作
- 支持 Modbus 通信
  - 超时功能
  - CR/LF 字符识别

20.4 USART 实现

表 66. USART/LPUART 特性描述了 BlueNRG-LP 器件上的 USART 和 LPUART 实现。

表 66. USART/LPUART 特性

USART 模式/特性(1)	USART	LPUART
调制解调器的硬件流控	X	X
使用 DMA 进行连续通信	X	X
多处理器通信	X	X
同步模式（主/从）	X	-
智能卡模式	X	-
单线半双工通信	X	X
IrDA SIR ENDEC 模块	X	-
LIN 模式	X	-
双时钟域	X	X
接收器超时中断	X	-
Modbus 通信	X	-
自动波特率检测	X	-
驱动器使能	X	X
USART 数据长度	7 位、8 位和 9 位	
Tx/Rx FIFO	X	X
Tx/Rx FIFO 大小	8	

1. *X = 支持。*

20.5 USART 功能说明

任何 USART 双向通信均需要至少两个引脚：接收数据输入引脚（RX）和发送数据输出引脚（TX）：

- **RX**：接收数据输入

  串行数据输入引脚。过采样技术可区分有效输入数据和噪声，从而用于恢复数据。

- **TX**：发送数据输出引脚

  如果关闭发射器，该输出引脚模式由其 I/O 端口配置决定。如果使能了发射器但没有待发送的数据，则 TX 引脚处于高电平。在单线和智能卡模式下，该 I/O 用于发送和接收数据。

正常 USART 模式下，通过这些引脚发送和接收串行数据。这些帧包括：

- 发送或接收前保持空闲线路
- 起始位
- 数据字（字长 7 位、8 位或 9 位），最低有效位在前
- 用于指示帧传输已完成的 0.5 个、1 个、1.5 个、2 个停止位
- USART 接口使用波特率发生器
- 状态寄存器（USART_ISR）
- 接收和发送数据寄存器（USART_RDR, USART_TDR）。当 FIFO 模式使能时，写入 USART_TDR 会将一个数据添加到发送 FIFO；而从 USART_RDR 读取则会将一个数据从接收 FIFO 中移除
- 波特率寄存器（USART_BRR）
- 智能卡模式下的保护时间寄存器（USART_GTPR）。

请参见第 20.7 节“USART 寄存器”了解每一位的定义。

在同步模式和智能卡模式下连接时需要以下引脚：

- **SCLK**：此引脚在同步主模式和智能卡模式下充当时钟输出。它在同步从模式下充当时钟输入。在同步主模式下，此引脚输出对应于 SPI 主模式的发射器数据时钟用于同步发送（起始位和停止位无时钟脉冲，软件选项用于发送最后一个数据位的时钟脉冲）。RX 上可同步接收并行数据。这一点可用于控制带移位寄存器的外设（如 LCD 驱动器）。时钟相位和极性可通过软件编程。

在智能卡模式下，SCLK 输出可向智能卡提供时钟。

- **NSS**：该引脚在同步从模式下充当从设备选择输入。

在 RS232 硬件流控制模式下需要以下引脚：

- **nCTS**：“清除以发送”用于在当前传输结束时阻止数据发送（高电平时）
- **nRTS**：“请求以发送”用于指示 USART 已准备好接收数据（低电平时）。

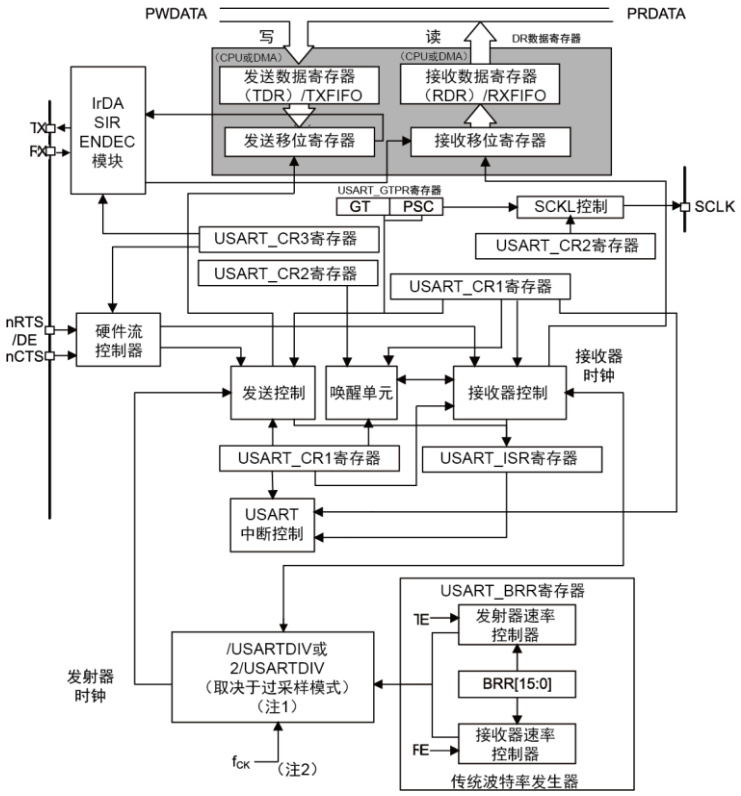
在 RS485 硬件控制模式下需要以下引脚：

- **DE**：“驱动器使能”用于激活外部收发器的发送模式。

注意： *DE 和 nRTS 共用同一个引脚。*

注意： *NSS 和 nCTS 共用同一个引脚。*

图 113. USART 框图



注意： *有关 USARTx_BRR 寄存器中 USARTDIV 编码的详细信息，请参见第 20.5.5 节“波特率生成”。*

注意： *f_{CK} 为 16 MHz。*

20.5.1 USART 字符说明

通过在 USART_CR1 寄存器中设定 M 位（M0：位 12 和 M1：位 28），字长可选择为 7 或 8 或 9 位。（参见图 114. 字长编程）。

- 7 位字符长度：M[1:0] = 10
- 8 位字符长度：M[1:0] = 00
- 9 位字符长度：M[1:0] = 01

注意： *在 7 位数据长度模式下，不支持智能卡模式、LIN 主模式和自动波特率（0x7F 和 0x55 帧检测）。*

在默认配置下，信号（TX 或 RX）在起始位工作期间处于低电平状态。在停止位工作期间处于高电平状态。

通过极性配置控制，可以单独针对每个信号对这些值取反。

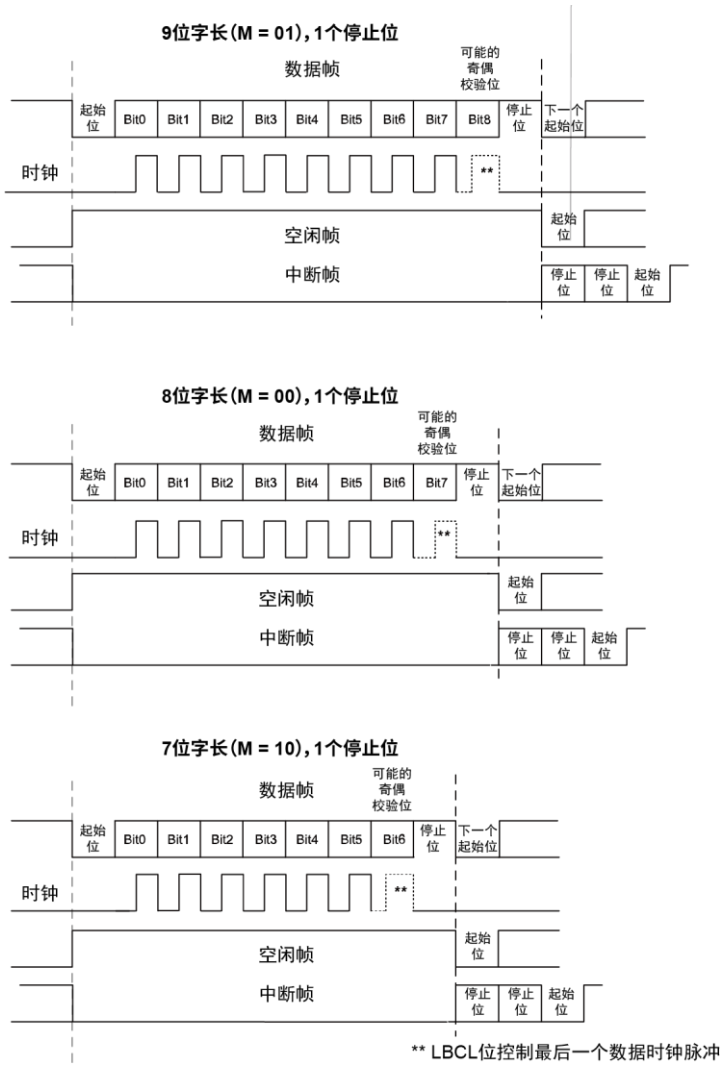
***空闲字符**可理解为一整个帧的“1”（“1”的数量包括停止位的数量）。*

***停止字符**可理解为在一个帧周期内接收到的电平均为“0”。发射器在中断帧的末尾插入 2 个停止位。*

发送和接收由通用波特率发生器驱动，发射器和接收器的使能位分别置 1 时将生成相应的发送时钟和接收时钟。

下面给出了各个块的详细信息。

图 114. 字长编程



20.5.2 FIFO 和阈值

USART 可在 FIFO 模式下工作，FIFO 缓冲区的深度为 16 字节（8 字节发送，8 字节接收）。

USART 具有发送 FIFO（TXFIFO）和接收 FIFO（RXFIFO）。通过将 USARTx_CR1 寄存器中的位 29 FIFOEN 置 1 来使能 FIFO 模式。

仅 UART、SPI 和智能卡模式支持 FIFO 模式。

最大数据字长为 9 位，TXFIFO 的宽度为 9 位。但是，RXFIFO 的默认宽度为 12 位。这是因为接收器不仅将数据，还将与每个字符相关的错误标志（奇偶校验错误、噪声错误和数据拆包错误标志）放入 FIFO。

注意： *接收的数据及其标志被存储在 RXFIFO 中。但是，在读取 RDR 时，只读取数据。状态标志位于 USART_ISR 寄存器中。*

可以定义触发 Tx 和 RX 中断的 TXFIFO 和 RXFIFO 级别。这些阈值通过 USARTx_CR3 控制寄存器中的位字段 RXFTCFG 和 TXFTCFG 进行编程。

在这种情况下：

- 当 RXFIFO 中接收到的数据量达到在 RXFTCFG 位字段中编程的阈值时，生成接收中断

- 当 TXFIFO 中空位的数量达到在 TXFTCFG 位字段中编程的阈值时，生成发送中断。

RXFIFO 阈值

使用 USARTx_CR3 寄存器中的 RXFTCFG 位字段配置 RXFIFO 阈值。

当接收的数据量等于编程的 RXFTCFG 时，USART_ISR 寄存器中的标志 RXFT 置位。

RXFT 标志置位意味着接收到了 RXFTCFG 个数据：USARTx_RDR 中 1 个数据和 RXFIFO 中（RXFTCFG - 1）个数据。因此，当 RXFTCFG 被编程为 «101» 时，RXFT 标志会在接收到 8 个数据时置位：RXFIFO 中 7 个数据和 USARTx_RDR 中 1 个数据。因此，接收到第 9 个数据不会使上溢标志置位。

20.5.3 发射器

发射器可发送 7 位、8 位或 9 位的数据字，具体取决于 M 位的状态。要激活发射器功能，必须将发送使能位（TE）置 1。发送移位寄存器中的数据在 TX 引脚输出，相应的时钟脉冲在 SCLK 引脚输出。

字符发送

USART 发送期间，首先通过 TX 引脚移出数据的最低有效位（默认配置）。该模式下，USARTx_TDR 寄存器的缓冲区（TDR）位于内部总线和发送移位寄存器之间。

当 FIFO 模式使能时，写入到发送数据寄存器 USART_TDR 中的数据在 TXFIFO 中排成队列。

每个字符前面都有一个起始位，其逻辑电平在一个位周期内为低电平。字符由可配置数量的停止位终止。

USART 支持以下停止位：0.5、1、1.5 和 2 个停止位。

注意： *向 USART_TDR 中写入要发送的数据前，TE 位必须先置 1。数据发送期间不应复位 TE 位。如果在发送期间复位 TE 位，由于波特率计数器发生冻结，会导致 TX 引脚上的数据损坏。当前正在发送的数据将丢失。在 TE 位使能后发送空闲帧。*

可配置的停止位

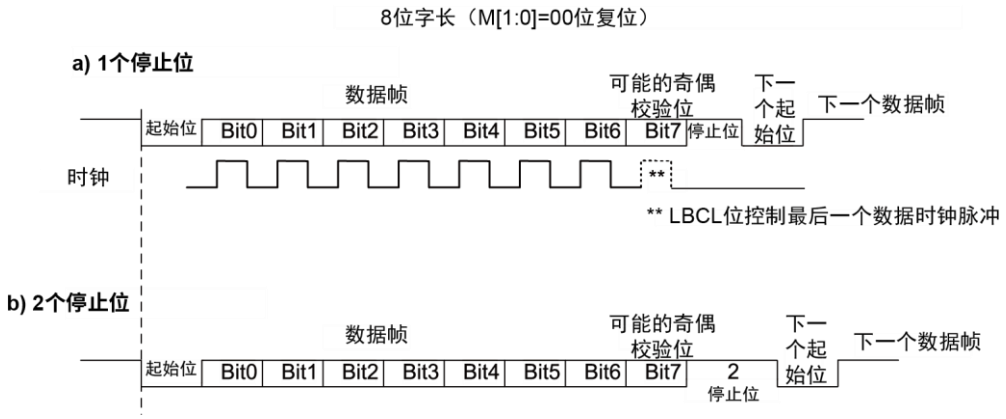
可以在控制寄存器 2 的位 13 和位 12 中编程将随各个字符发送的停止位的数量。

- 1 个停止位：这是停止位数量的默认值
- 2 个停止位：正常 USART、单线和调制解调器模式支持 2 个停止位
- 1.5 个停止位：用于智能卡模式。

空闲帧发送包含停止位。

中断帧发送是 10 个低位（当 M[1:0] = 00）或 11 个低位（当 M[1:0] = 01）或 9 个低位（当 M[1:0] = 10）后接 2 个停止位（参见图 115. 可配置的停止位）。无法传送长中断（中断长度大于 9/10/11 个低电平位）。

图 115. 可配置的停止位



字符发送步骤

1. 对 USART_CR1 中的 M 位进行编程以定义字长。
2. 使用 USART_BRR 寄存器选择所需波特率。
3. 对 USART_CR2 中的停止位数量进行编程。
4. 通过向 USART_CR1 寄存器中的 UE 位写入 1 使能 USART。
5. 如果将进行多缓冲区通信，请选择 USART_CR3 中的 DMA 使能（DMAT）。按照多缓冲区通信中的说明配置 DMA 寄存器。
6. 置位 USART_CR1 中的 TE 位以便在首次发送时发送一个空闲帧。
7. 在 USART_TDR 寄存器中写入要发送的数据。为每个要在单缓冲区模式下发送的数据重复这一步骤。
   a. 当 FIFO 模式禁用时，在 USART_TDR 中写入数据会将 TXE 标志清零。
   b. 当 FIFO 模式使能时，在 USART_TDR 中写入数据会将一个数据添加到 TXFIFO，并且 USART_TDR 中的写操作会在 TXFNF 标志置位时执行。在 TXFIFO 装满之前，此标志保持置位。
8. 向 USART_TDR 寄存器写入最后一个数据后，等待至 TC=1。
   – 当 FIFO 模式禁用时，这表明最后一个帧的发送已完成。
   – 当 FIFO 模式使能时，这表明 TXFIFO 和移位寄存器均为空。

单字节通信

- 当 FIFO 模式禁用时：

始终通过向发送数据寄存器写入数据将 TXE 标志清零。TXE 标志由硬件置 1，表示：

- 数据已从 USARTx_TDR 寄存器移到移位寄存器中且数据发送已开始
- USARTx_TDR 寄存器为空
- USARTx_TDR 寄存器中可写入下一个数据，而不会覆盖前一个数据。

TXEIE 位置 1 时该标志位会生成中断。

发送时，要传入 USARTx_TDR 寄存器的写指令中存有 TDR 寄存器中的数据，该数据将在当前发送结束时复制到移位寄存器中。

未发送时，要传入 USARTx_TDR 寄存器的写指令将数据置于移位寄存器中，数据发送开始时，TXE 位置 1。

- 当 FIFO 模式使能时，TXFNF（TXFIFO 未满）标志由硬件置位，这表明：
  - TXFIFO 未满
  - USART_TDR 寄存器为空
  - USART_TDR 寄存器中可写入下一个数据，而不会覆盖前一个数据。发送时，对 USART_TDR 寄存器的写操作将数据存储在 TXFIFO 中。在当前发送结束时，会将数据从 TXFIFO 复制到移位寄存器。

当 TXFIFO 未满时，即使在写入 USART_TDR 后，TXFNF 标志仍保持为 1。当 TXFIFO 已满时，该标志清零。

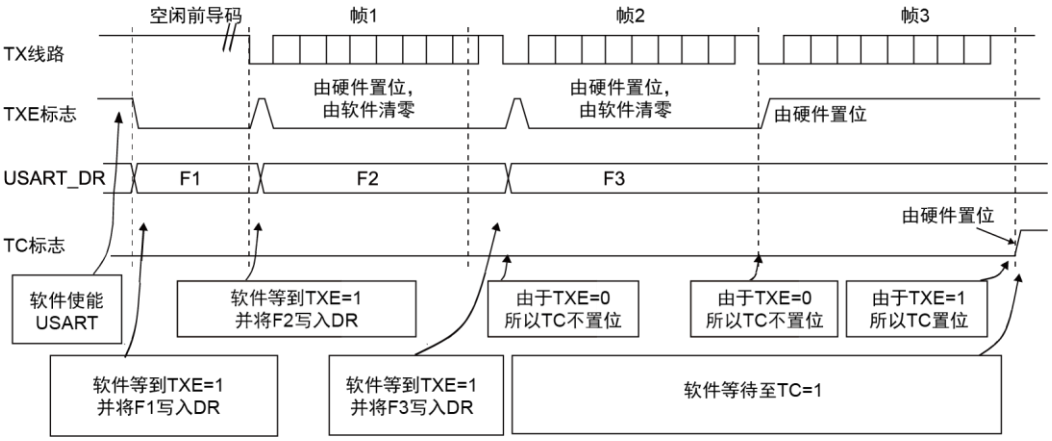
TXFNFIE 位置 1 时该标志位会生成中断。

或者，当达到 TXFIFO 阈值时，可生成中断，并能将数据写入 FIFO。在这种情况下，CPU 可写入由设定的触发值定义的数据块。

如果发送了帧（在停止位之后）且 TXE 标志（FIFO 模式下为 TXFE）置位，则 TC 标志置位。如果 USART_CR1 寄存器中的 TCIE 位置 1，则生成中断。

在 USART_TDR 寄存器中写入最后一个数据后，必须等到 TC=1 才能禁用 USART，否则会导致微控制器进入低功耗模式（参见图 116. 发送时的 TC/TXE 行为）。

图 116. 发送时的 TC/TXE 行为



注意： *当 FIFO 管理使能时，将 TXFNF 标志用于数据发送。*

中断字符

置位 SBKRQ 位将发送一个中断字符。中断帧长度取决于 M 位（参见图 114. 字长编程）。

如果将“1”写入 SBKRQ 位，则当前字符发送完成后，将在 TX 线路上发送一个中断字符。该写操作将 SBKF 位置 1 并在中断字符发送完成时（发送中断字符后的停止位期间）由硬件复位。USART 在中断帧末尾的两位持续时间内插入一个逻辑“1”信号（STOP），以确保识别下个帧的起始位。

当 SBKRQ 位置 1 时，在当前发送结束时发送断路字符。

当 FIFO 模式使能时，发送断路字符优先于发送数据，即使 TXFIFO 已满。

空闲字符

置位 TE 位会驱动 USART 在第一个数据帧之前发送一个空闲帧。

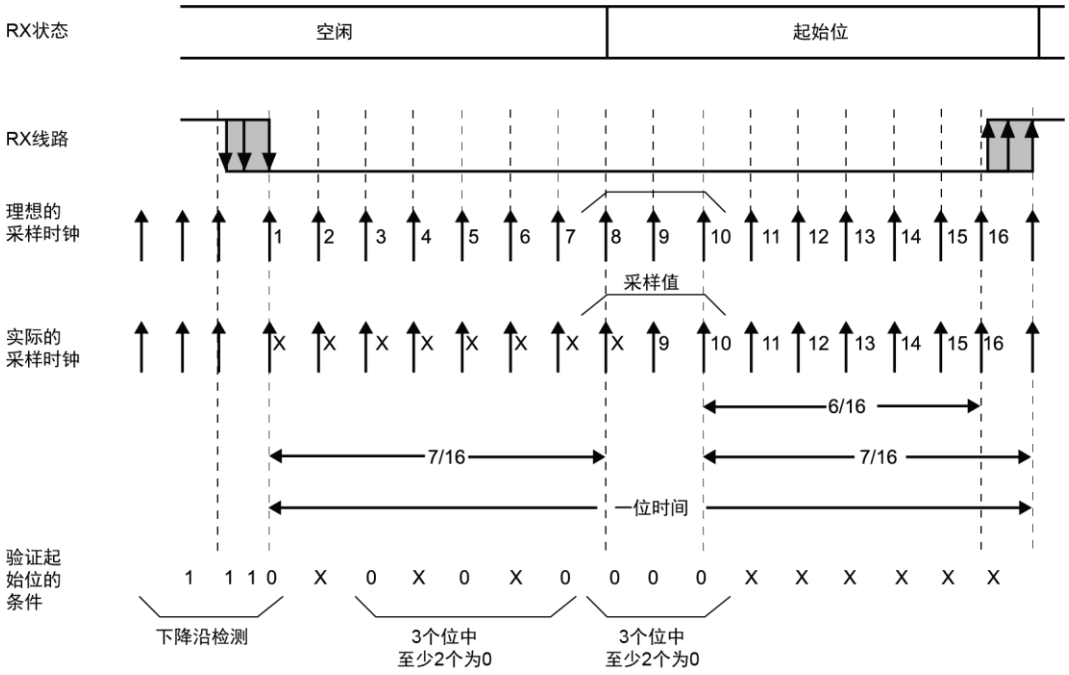
20.5.4 接收器

USART 可接收 7、 8 位或 9 位的数据字，具体取决于 USART_CR1 寄存器中的 M 位。

起始位检测

16 倍或 8 倍过采样时，起始位检测序列相同。在 USART 中，识别出特定序列的采样时会检测起始位。该序列为：1 1 1 0 X 0 X 0X 0X 0 X 0X 0。

图 117. 16 倍或 8 倍过采样时的起始位检测



注意： *如果序列不完整，起始位检测将中止，接收器将返回空闲状态（无标志位置 1）等待下降沿。*

如果 3 个采样位为 0（对第 3 位、第 5 位和第 7 位进行第一次采样发现 3 个位为 0，对第 8 位、第 9 位和第 10 位进行第二次采样也发现 3 个位为 0），则起始位确认（当 FIFO 模式使能时，如果 RXNEIE=1，则 RXNE 标志置 1 并生成中断；如果 RXFNEIE=1，则 RXFNE 标志置 1 并生成中断）。

在以下情况下，起始位有效但 NF 噪声标志置 1：

1. 对于两次采样，3 个采样位中有 2 位为 0（针对第 3 位、第 5 位和第 7 位进行采样；针对第 8 位、第 9 位和第 10 位采样）

   或

2. 如果其中一次采样时（对第 3 位、第 5 位和第 7 位进行采样或对第 8 位、第 9 位和第 10 位进行采样），3 个采样位中有 2 个为 0。

如果以上两种情况都不符合，起始位检测将中止，接收器将返回空闲状态（无标志位置 1）。

字符接收

USART 接收期间，首先通过 RX 引脚移入数据的最低有效位（默认配置）。

字符接收步骤

1. 对 USART_CR1 中的 M 位进行编程以定义字长
2. 使用波特率寄存器 USART_BRR 选择所需波特率
3. 对 USART_CR2 中的停止位数量进行编程
4. 通过向 USART_CR1 寄存器中的 UE 位写入 1 使能 USART
5. 如果将进行多缓冲区通信，请选择 USART_CR3 中的 DMA 使能（DMAR）。按照多缓冲区通信中的说明配置 DMA 寄存器
6. 置位 RE 位 USART_CR1。这一操作将使能接收器开始搜索起始位。

接收到字符时

- 当 FIFO 模式禁用时，RXNE 位置 1 表示移位寄存器的内容被传输到 RDR。也就是说，已接收到并可读取数据（以及其相应的错误标志）

- 当 FIFO 使能时，RXFNE 位置 1 表示 RXFIFO 不为空。读取 USART_RDR 将获取 RXFIFO 中最早的条目。在接收数据时，数据和与之相关的错误位均保存在接收 RXFIFO 中
- 如果 RXNEIE（FIFO 模式下为 RXFNEIE）位置 1，将生成中断
- 如果接收期间已检测到帧错误、噪声错误、奇偶校验错误或上溢错误，错误标志位可置 1
- 在多缓冲区通信中：
  - 当 FIFO 模式禁用时，RXNE 在每个字节被接收后置 1，并通过接收数据寄存器的 DMA 读操作清零。
  - 当 FIFO 模式使能时，RXFNE 在 RXFIFO 不为空时置 1。在每次 DMA 请求后，将从 RXFIFO 提取数据。如果不为空，即 RXFIFO 中有等待读取的数据，RXFIFO 将触发 DMA 请求。
- 在单缓冲区模式下，
  - 当 FIFO 模式禁用时：通过软件对 USARTx_RDR 寄存器的读取执行 RXNE 标志清零。也可以通过将 USARTx_RQR 寄存器中的 RXFRQ 位置 1 将 RXNE 标志位清零。RXNE 标志必须在结束接收下一个字符前清零，以避免发生上溢错误。
  - 当 FIFO 模式使能时，RXFNE 在 RXFIFO 不为空时置 1。在每次读取 USART_RDR 后，从 RXFIFO 提取数据。当 RXFIFO 为空时，RXFNE 标志清零。也可以通过将 USART_RQR 寄存器中的 RXFRQ 位置 1 将 RXFNE 标志位清零。当 RXFIFO 已满时，必须在下一个字符的接收结束前读取 RXFIFO 中的第一个条目，以避免发生上溢错误。RXFNEIE 位置 1 时 RXFNE 标志位会生成中断。或者，当达到 RXFIFO 阈值时，可生成中断，并能从 RXFIFO 读取数据。在这种情况下，CPU 可读取由设定的阈值定义的数据块。

中断字符

接收到中断字符时，USART 将会按照数据拆包错误对其进行处理。

空闲字符

检测到空闲帧时，处理步骤与接收到数据的情况相同；如果 IDLEIE 位置 1，则会产生中断。

上溢错误

- FIFO 模式禁用：

如果在 RXNE 未复位时接收到字符，则会发生上溢错误。RXNE 位清零前，数据无法从移位寄存器传送到 RDR 寄存器。每接收到一个字节后，RXNE 标志位都将置 1。当 RXNE 标志位是 1 时，如果在接收到下一个数据或尚未处理上一个 DMA 请求时，则会发生上溢错误。发生上溢错误时：

1. ORE 位置 1。
2. RDR 内容不丢失。对 USARTx_RDR 执行读操作时可使用先前的数据。
3. 移位寄存器被重写。之后，上溢期间接收到的任何数据都将丢失。
4. 如果 RXNEIE 位置 1 或 EIE 位置 1，则会生成中断。

- FIFO 模式启用

  当接收 FIFO 已满时，如果移位寄存器准备进行传输，将发生上溢错误。在 RXFIFO 中有一个空位之前，无法将数据从移位寄存器传输到 USARTx_RDR 寄存器。当 RXFIFO 不为空时，RXFNE 标志置 1。如果 RXFIFO 已满且移位寄存器准备进行传输，将发生上溢错误。发生上溢错误时：

1. ORE 位置 1。
2. RXFIFO 中的第一个条目不会丢失。在对 USART_RDR 执行读取时，可读取该条目。
3. 移位寄存器被重写。之后，上溢期间接收到的任何数据都将丢失。
4. 如果 RXFNEIE 位置 1 或 EIE 位置 1，则会生成中断。

通过设置 USARTx_ICR 寄存器中的 ORECF 位复位 ORE 位。

注意： *ORE 位置 1 时表示至少 1 个数据丢失。当 FIFO 模式禁用时，有两种可能性*

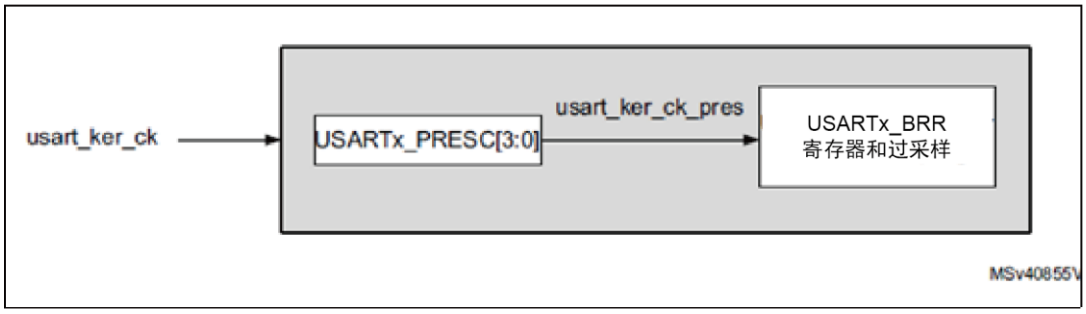
- *如果 RXNE=1，则最后一个有效数据存储于接收寄存器 RDR 中并且可进行读取*
- *如果 RXNE=0，则表示最后一个有效数据已被读取，因此 RDR 中没有要读取的数据。接收到新（丢失）数据的同时已读取 RDR 中的最后一个有效数据时，会发生该情况。*

选择时钟源和合适的过采样方法

时钟源频率为 f_{CK}（16 MHz）。

可通过 USARTx_PRESC 寄存器中的可编程因子对 f_{CK} 进行分频。

图 118. usart_ker_ck 时钟分频器框图



时钟源还决定通信速度范围（尤其是最大通信速度）。

接收器采用不同的用户可配置过采样技术（除了同步模式下），可以从噪声中提取有效数据。这可在最大通信速度与抗噪声/时钟误差性能之间实现平衡。

可通过编程 USART_CR1 寄存器中的 OVER8 位来选择过采样方法，可以是波特率时钟的 16 倍或 8 倍（图 119. 16 倍过采样时的数据采样和图 120. 8 倍过采样时的数据采样）。

根据应用：

- 选择 8 倍过采样（OVER8=1）以获得更高速度（最高 $f_{CKPRES}/8$）。在这种情况下，接收器对时钟偏差的最大容差将降低（参见第 20.5.6 节“USART 接收器对时钟偏差的容差”）。
- 选择 16 倍过采样（OVER8=0）以增加接收器对时钟偏差的容差。在这种情况下，最大速度被限制在最大值 $f_{CKPRES}/16$。

其中的 f_{CKPRES} 是经预分频器分频的 USART 输入时钟。

可通过编程 USART_CR3 寄存器中的 ONEBIT 位选择用于评估逻辑电平的方法。有两种选择：

- 在已接收位的中心进行三次采样，从而进行多数表决。这种情况下，如果用于多数表决的 3 次采样结果不相等，NF 位置 1
- 在已接收位的中心进行单次采样

  根据应用：

  – 在噪声环境中工作时，选择三个采样的多数表决法（ONEBIT=0），并在检测到噪声时拒绝数据（参见“图 68. 延时大于负脉冲的死区时间波形”，因为这表示采样期间发生了干扰）。

  – 当线路无噪声时，为了增大接收器对时钟偏差的容差，选择单次采样法（ONEBIT=1）（参见第 20.5.6 节“USART 接收器对时钟偏差的容差”）。在这种情况下，NF 位从不置位。

帧中检测到噪声时：

- NF 位在 RXNE 位（FIFO 模式使能时为 RXFNE）的上升沿置位
- 无效数据从移位寄存器传送到 USART_RDR 寄存器

- 单字节通信时无中断产生。然而，在 RXNE 位（FIFO 模式使能时为 RXFNE）生成中断时，此位出现上升沿。在多缓冲区通信中，如果 USART_CR3 寄存器中的 EIE 位置 1，则生成中断。

通过设置 ICR 寄存器中的 NFCF 位复位 NF 位。

注意： *SPI 模式不支持噪声错误。*

注意： *智能卡、IrDA 和 LIN 模式下不可采用 8 倍过采样。在这些模式下，OVER8 位由硬件强制清零。*

图 119. 16 倍过采样时的数据采样

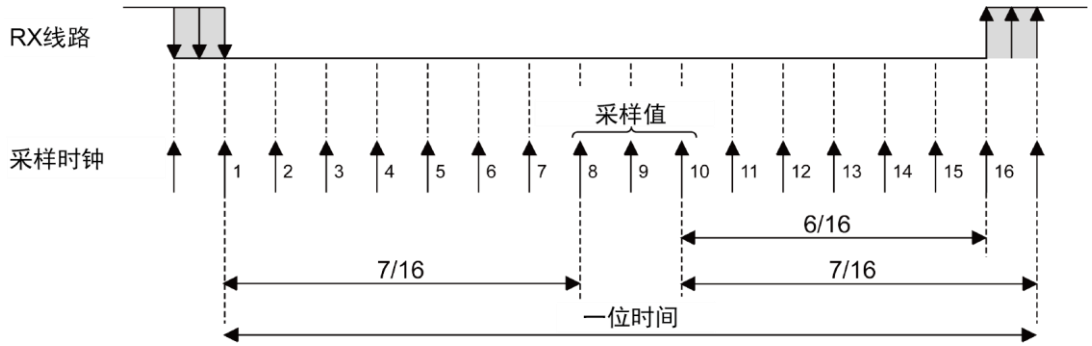


图 120. 8 倍过采样时的数据采样

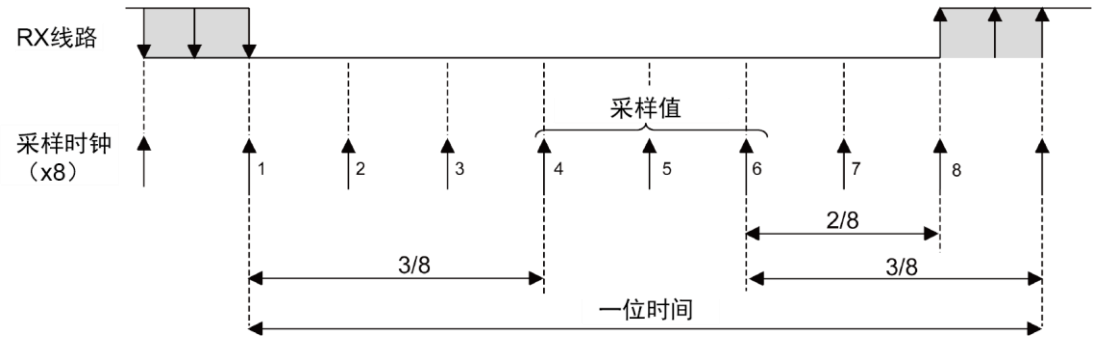


表 67. 通过采样数据进行噪声检测

采样值	NE 状态	接收的位值
000	0	0
001	1	0
010	1	0
011	1	1
100	1	0
101	1	1
110	1	1
111	0	1

数据拆包错误

在去同步化或过度的噪声后，如果接收时未在预期时间识别出停止位，将检测到数据拆包错误。

检测到数据拆包错误时：

- FE 位由硬件置 1

- 无效数据从移位寄存器传输到 USART_RDR 寄存器（FIFO 模式使能时为 RXFIFO）
- 单字节通信时无中断产生。然而，在 RXNE 位（FIFO 模式使能时为 RXFNE）生成中断时，此位出现上升沿。在多缓冲区通信中，如果 USART_CR3 寄存器中的 EIE 位置 1，则生成中断。

通过将 1 写入 USART_ICR 寄存器中的 FECF 位来复位 FE 位。

注意： *SPI 模式不支持数据拆包错误。*

接收期间的可配置停止位

可通过控制寄存器 2 中的控制位配置要接收的停止位的数量 - 可以是 1 或 2 个（正常模式下），也可以是 0.5 或 1.5 个（智能卡模式下）。

- **0.5 个停止位（在智能卡模式下接收）**：不会对 0.5 个停止位进行采样。结果，选择 0.5 个停止位时，无法检测到数据拆包错误和中断帧。
- ***1 个停止位***：将在第 8、第 9 和第 10 次采样时对 1 个停止位进行采样。
- ***1.5 个停止位（智能卡模式）***：在智能卡模式下发送时，设备必须检查数据是否正确发送。因此必须使能接收器块（USART_CR1 寄存器中的 RE =1）并检查停止位，以测试智能卡是否已检测到奇偶校验错误。发生奇偶校验错误时，智能卡会在采样时将数据信号强制为低电平，即 NACK 信号-，该信号被标记为数据拆包错误。然后，在 1.5 个停止位结束时，FE 和 RXNE（FIFO 模式使能时为 RXFNE）标志置 1。在第 16、第 17 和第 18 次采样时对 1.5 个停止位进行采样（停止位采样开始后维持 1 个波特时钟周期）。1.5 个停止位可分为 2 个部分：0.5 个波特时钟周期（未发生任何动作），然后是 1 个正常的停止位周期（一半时间处进行采样）。参见第 20.5.14 节“接收器超时”了解更多信息。
- ***2 个停止位***：采样 2 个停止位时在第 8、第 9 和第 10 次采样时对第一个停止位进行采样。如果在第一个停止位期间检测到数据拆包错误，则数据拆包错误标志置 1。发生数据拆包错误时不检测第 2 个停止位。在第一个停止位结束时，RXNE（FIFO 模式使能时为 RXFNE）标志置 1。

20.5.5 波特率生成

接收器和发射器（Rx 和 Tx）的波特率设置为 USART_BRR 寄存器中编程的相同值。

以下公式用于计算标准 USART（包含 SPI 模式）下的波特率（OVER8= 0 或 1）。

16 倍过采样时的公式为：

$$Tx/Rxbaud = \frac{f_{CKPRES}}{USARTDIC} \quad (4)$$

8 倍过采样时的公式为：

$$Tx/Rxbaud = \frac{2 \times f_{CKPRES}}{USARTDIC} \quad (5)$$

以下公式用于计算智能卡、LIN 和 IrDA 模式下的波特率（OVER8= 0）：

$$Tx/Rxbaud = \frac{f_{CKPRES}}{USARTDIC} \quad (6)$$

f_{ckpres} 是 USART 时钟，是经 USART_PRES 寄存器中配置的预分频因子分频的 USART 输入时钟。

USARTDIV 是一个存放在 USART_BRR 寄存器中的无符号定点数。

- 当 OVER8 = 0 时，BRR = USARTDIV
- 当 OVER8 = 1 时
  - BRR[2:0] = USARTDIV[3:0]，右移 1 位
  - BRR[3] 必须保持清零
  - BRR[15:4] = USARTDIV[15:4]

注意： *对 USART_BRR 执行写操作后，波特率计数器更新为波特率寄存器中的新值。因此，波特率寄存器的值不应在通信时发生更改。16 倍和 8 倍过采样时，USARTDIV 必须大于或等于 16d。*

如何从 USART_BRR 寄存器中获取 USARTDIV

示例 1

为了在 f_{CKPRES} = 8 MHz 时达到 9600 波特。

- 16 倍过采样时：
  USARTDIV = 8 000 000/9600
  BRR = USARTDIV = 833d = 0341h
- 8 倍过采样时：
  USARTDIV = 2 * 8 000 000/9600
  USARTDIV = 1666,66（1667d = 683h）
  BRR[3:0] = 3h >>1 = 1h
  BRR = 0x681

20.5.6 USART 接收器对时钟偏差的容差

仅当总时钟系统偏差小于 USART 接收器的容差时，USART 异步接收器才能正常工作。影响总偏差的因素包括：

- DTRA：发射器误差引起的偏差（其中还包括发射器本地振荡器的偏差）
- DQUANT：接收器的波特率量化引起的误差
- DREC：接收器本地振荡器的偏差
- DTCL：传输线路引起的偏差（通常是由于收发器所引起，它可能会在低电平到高电平转换时序与高电平到低电平转换时序之间引入不对称）

  DTRA + DQUANT + DREC + DTCL < USART 接收器的容差。

USART 接收器可在不超过“表 68. BRR [3:0] = 0000 时的 USART 接收器容差（高密度器件）”和“表 69. BRR [3:0] ≠ 0000 时的 USART 接收器容差（高密度器件）”中指定的最大容许偏差的情况下正确地接收数据，具体取决于以下选择：

- 由 USART_CR1 寄存器中的 M 位定义的 9、 10 位或 11 位字符长度
- 由 USART_CR1 寄存器中的 OVER8 位定义的 8 倍或 16 倍过采样
- USART_BRR 寄存器的 BRR[3:0]位等于或不等于 0000
- 使用 1 位或 3 位对数据进行采样，取决于 USART_CR3 寄存器中 ONEBIT 位的值。

表 68. BRR [3:0] = 0000 时的 USART 接收器容差（高密度器件）

M 位	OVER8 位 = 0		OVER8 位 = 1	
	ONEBIT=0	ONEBIT=1	ONEBIT=0	ONEBIT=1
00	3.75%	4.375%	2.50%	3.75%
01	3.41%	3.97%	2.27%	3.41%
10	4.16	4.86	2.77	4.16

表 69. BRR [3:0] ≠ 0000 时的 USART 接收器容差（高密度器件）

M 位	OVER8 位 = 0		OVER8 位 = 1	
	ONEBIT=0	ONEBIT=1	ONEBIT=0	ONEBIT=1
00	3.33%	3.88%	2%	3%
01	3.03%	3.53%	1.82%	2.73%
10	3.7	4.31	2.22	3.33

注意：在特殊情况下，当接收的帧包含一些刚好 10 位持续时间（M 位 = 00，M = 01 时为 11 位持续时间，M = 10 时为 9 位持续时间）的空闲帧时，“表 68. BRR [3:0] = 0000 时的 USART 接收器容差（高密度器件）”和“表 69. BRR[3:0] ≠ 0000 时的 USART 接收器容差（高密度器件）”中所列数据可能稍有不同。

20.5.7 自动波特率检测

USART 能够根据接收一个字符检测并自动设置 USART_BRR 寄存器的值。自动波特率检测在以下两种情况下非常有用：

- 事先不知道系统的通信速度
- 系统正在使用精确度相对较低的时钟源且该机制允许在不测量时钟偏差的情况下获得正确的波特率。

时钟源频率必须与预期通信速率兼容（在 16 倍过采样时，波特率介于 f_{CK}/65535 和 f_{CK}/16 之间）。8 倍过采样时，波特率介于 f_{CK}/65535 与 f_{CK}/8 之间。

激活自动波特率检测前，必须先选择自动波特率检测模式。有四种基于不同字符模式的模式。

可通过 USART_CR2 寄存器中的 ABRMOD[1:0]字段选择模式。在这些自动波特率模式下，波特率在同步接收数据期间被多次测量，每次测量的结果都与前一次进行比较。

这些模式包括：

- **模式 0**：以“1”位为开头的任意字符。

  这种情况下，USART 会测量起始位的持续时间（下降沿到上升沿）。
- **模式 1**：以 10xx 位模式开头的任何字符。

  这种情况下，USART 会测量起始位和第一个数据位的持续时间。

  测量在下降沿到下降沿期间完成，可在信号斜率较小时确保较高的精度。
- **模式 2**：0x7F 字符帧（可以是采用 LSB 优先模式的 0x7F 字符或采用 MSB 优先模式的 0xFE）。

  在这种情况下，将首先在起始位结束时，然后在位 6 结束时更新波特率（BRs）（基于从下降沿至下降沿执行的测量：BR6）。Bit0 至 Bit6 在 BRs 采样，而更多字符位则在 BR6 采样。
- **模式 3**：0x55 字符帧。

  在这种情况下，将首先在起始位结束时，然后在 bit0 结束时（基于从下降沿至下降沿执行的测量：BR0），最后在 bit6（BR6）结束时更新波特率（BRs）。Bit0 在 BRs 采样，Bit1 至 Bit6 在 BR0 采样，而更多字符位则在 BR6 采样。同时，对 RX 线路的每个中间转换执行其它检查。如果 RX 上的转换与接收器（基于根据位 0 计算的波特率的接收器）未充分同步，则生成错误。

激活自动波特率检测前，必须先通过向 USART_BRR 寄存器写入非零的波特率值来初始化该寄存器。

通过将 USART_CR2 寄存器中的 ABREN 位置 1 来激活自动波特率检测。然后，USART 等待 RX 线路上的第一个字符。通过将 USART_ISR 寄存器中的 ABRF 标志置 1 来指示自动波特率操作完成。如果线路繁忙，则无法保证正确的波特率检测。在这种情况下，BRR 值可能损坏且 ABRE 错误标志置 1。如果通信速度不在自动波特率检测范围（位持续时间不在 16 个和 65536 个时钟周期（16 倍过采样时）之间，也不在 8 个和 65536 个时钟周期（8 倍过采样时）之间）内，也会出现这种情况。

稍后可通过复位 ABRF 标志（通过写入 0）重新启动自动波特率检测。

当 FIFO 管理禁用时，如果发生自动波特率错误，则 ABRE 标志与 RXNE 和 FE 一同置 1。

当 FIFO 管理使能时，如果发生自动波特率错误，则 ABRE 标志与 RXFNE 和 FE 一同置 1。

当 FIFO 模式使能时，应使用第一个 RXFIFO 位置上的数据执行自动波特率检测。因此，在启动自动波特率检测之前，用户应使用 USARTx_ISR 寄存器中的 RXFNE 标志确保 RXFIFO 为空。

注意： *在自动波特率操作期间，如果 USART 禁用（UE=0），则 BRR 值可能发生损坏。*

20.5.8 多处理器通信

可以通过 USART 进行多处理器通信（多个 USART 连接在一个网络中）。例如，其中一个 USART 可以是主 USART，其 TX 输出与其它 USART 的 RX 输入相连接。其它 USART 为从 USART，其各自的 TX 输出在逻辑上通过与运算连在一起，并与主 USART 的 RX 输入相连接。

在多处理器配置中，理想情况下通常只有预期的消息接收方主动接收完整的消息内容，从而减少由所有未被寻址的接收器造成的冗余 USART 服务开销。

可通过静默功能将未被寻址的器件置于静默模式下。为了使用静默模式功能，必须将 USARTx_CR1 寄存器中的 MME 位置 1。

注意： *当 FIFO 管理使能且 MME 已置位时，不得将 MME 位清零然后再次快速置位（在两个 UCLK 周期内），否则静默模式可能保持有效状态。*

在静默模式下：

- 不得将接收状态位置 1
- 禁止任何接收中断
- USART_ISR 寄存器中的 RWU 位置 1。在某些情况下，RWU 可以由硬件或软件通过 USART_RQR 寄存器中的 MMRQ 位自动控制。

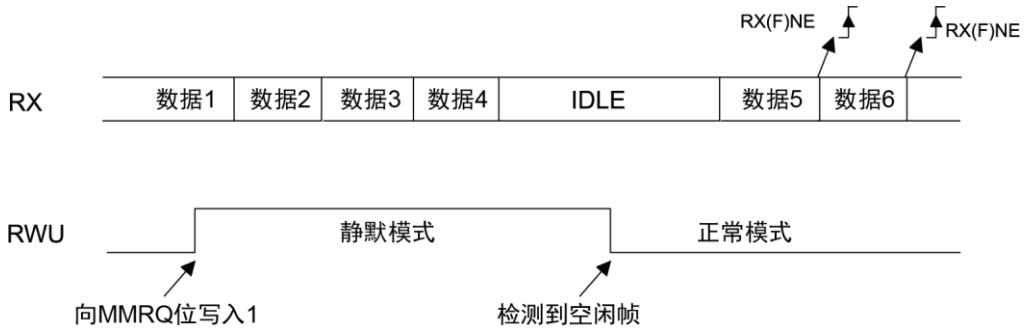
根据 USART_CR1 寄存器中 WAKE 位的设置，USART 可使用以下两种方法进入或退出静默模式：

- 如果 WAKE 位被复位，则进行空闲线路检测
- 如果 WAKE 位置 1，则进行地址标记检测。

空闲线路检测（WAKE=0）

当向 MMRQ 位写入 1 且 RWU 位自动置 1 时，USART 进入静默模式。当检测到空闲帧时，它会被唤醒。此时 RWU 位会由硬件清零，但 USART_ISR 寄存器中的 IDLE 位不会置 1。使用空闲线路检测的静默模式行为的示例见“图 121. 使用空闲线路检测的静默模式”。

图 121. 使用空闲线路检测的静默模式



注意： *如果在 IDLE 字符已经过去时将 MMRQ 位置 1，将不会进入静默模式（RWU 未置 1）。*

注意： *如果在线路处于空闲状态时激活 USART，在一个 IDLE 帧持续时间后（不只在接收一个字符帧后）会检测到空闲状态。*

4 位/7 位地址标记检测（WAKE=1）

在此模式下，如果字节的 MSB 为 1，则将这些字节识别为地址，否则将其识别为数据。在地址字节中，目标接收器的地址位于 4 个或 7 个 LSB 中。7 位或 4 位地址检测通过 ADDM7 位来选择。接收器会将此 4 位/7 位字与其地址进行比较，该接收器的地址在 USART_CR2 寄存器的 ADD 位中进行设置。

注意： *在 7 位和 9 位数据模式下，地址检测分别在 6 位和 8 位地址上完成（ADD[5:0]和 ADD[7:0]）。*

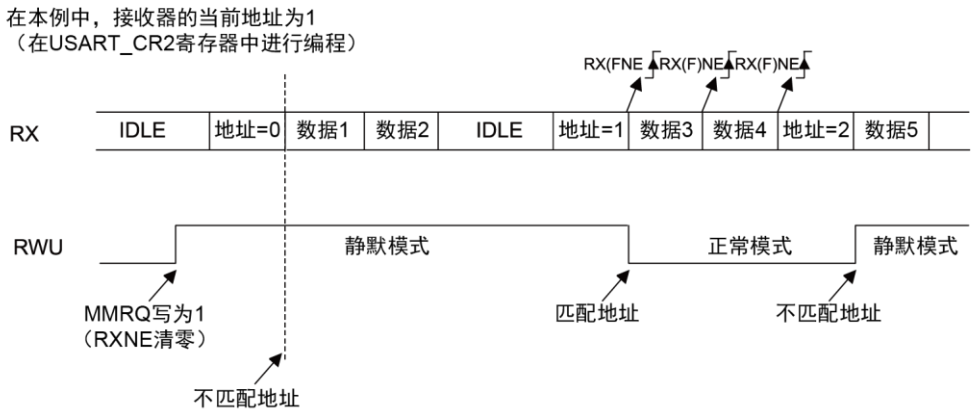
当接收到与其编程地址不匹配的地址字符时，USART 会进入静默模式。此时，RWU 位将由硬件置 1。USART 进入静默模式后，RXNE 标志不会针对此地址字节置 1，也不会发出中断或 DMA 请求。当 FIFO 管理使能时，在进入静默模式之前，软件应确保 RXFIFO 中有至少一个空位。

当向 MMRQ 位写入 1 时，USART 也会进入静默模式。这种情况下，RWU 位也自动置 1。当接收到与编程地址匹配的地址字符时，USART 会退出静默模式。然后 RWU 位被清零，可以开始正常接收后续字节。由于 RWU 位已清零，RXNE/ RXFNE 位会针对地址字符置 1。

注意： *当 FIFO 管理使能时，MMRQ 在接收器对数据的最后一位进行采样时置 1，这些数据可能会在进入静默模式之前被接收。*

使用地址标记检测的静默模式行为的示例见“图 122. 使用地址标记检测的静默模式”。

图 122. 使用地址标记检测的静默模式



20.5.9 Modbus 通信

USART 为 Modbus/RTU 和 Modbus/ASCII 协议的实现提供基本支持。Modbus/RTU 是一个半双工块传输协议。该协议的控制部分（地址识别、块完整性控制和指令解析）必须用软件实现。

USART 为块结束检测提供基本支持，无需软件开销或其它资源。

Modbus/RTU

在此模式下，一个块的结束通过超过 2 个字符时间的“静默”（空闲线路）来识别。此功能通过可编程的超时功能实现。

超时功能和中断必须分别通过 USART_CR2 寄存器中的 RTOEN 位和 USART_CR1 寄存器中的 RTOIE 位激活。与 2 个字符时间（例如 22 x 位持续时间）的超时相对应的值必须在 RTO 寄存器中编程。如果在此期间接收线路空闲，则在接收到最后一个停止位后，将生成中断，同时通知软件当前块接收已完成。

Modbus/ASCII

在此模式下，块结束通过特定（CR/LF）字符序列识别。USART 通过字符匹配功能管理此机制。通过在 ADD[7:0]字段中编程 LF ASCII 码以及激活字符匹配中断（CMIE=1），软件可在接收到 LF 时获得通知并检查 DMA 缓存区中的 CR/LF。

20.5.10 奇偶校验控制

将 USART_CR1 寄存器中的 PCE 位置 1，可以使能奇偶校验控制（发送时生成奇偶校验位，接收时进行奇偶校验检查）。根据 M 位定义的帧长度，“表 70. 帧格式”列出了可能的 USART 帧格式。帧格式。

表 70. 帧格式

M 位	PCE 位	USART 帧(1)
00	0	| SB | 8 位数据 | STB |
00	1	| SB | 7 位数据 | PB | STB |
01	0	| SB | 9 位数据 | STB |
01	1	| SB | 8 位数据 PB | STB |
10	0	| SB | 7 位数据 | STB |
10	1	| SB | 6 位数据 | PB | STB |

1. *图注：SB：起始位，STB：停止位，P：奇偶校验位。在数据寄存器中，PB 始终位于 MSB 位置（第 8 位或第 7 位，具体取决于 M 位的值）。*

偶校验

对奇偶校验位进行计算，使帧和奇偶校验位中“1”的数量为偶数（帧由 6 个、7 个或 8 个 LSB 位组成，具体取决于 M 位的值）。

例如，如果数据=00110101 且 4 个位置 1，则在选择偶校验（USART_CR1 寄存器中的 PS 位 = 0）时，校验位为 0。

奇校验

对奇偶校验位进行计算，使帧和奇偶校验位中“1”的数量为奇数（帧由 6 个、7 个或 8 个 LSB 位组成，具体取决于 M 位的值）。

例如，如果数据=00110101 且 4 个位置 1，则在选择奇校验（USART_CR1 寄存器中的 PS 位 = 1）时，校验位为 1。

接收时进行奇偶校验检查

如果奇偶校验检查失败，则 USART_SR 寄存器中的 PE 标志置 1；如果 USART_CR1 寄存器中 PEIE 位置 1，则会生成中断。通过软件将 1 写入 USART_ICR 寄存器中的 PECF 位来清零 PE 标志。

发送中的奇偶校验生成

如果 USART_CR1 寄存器中的 PCE 位置 1，则在数据寄存器中所写入数据的 MSB 位会进行传送，但是会由奇偶校验位进行更改（如果选择偶校验（PS=0），则“1”的数量为偶数；如果选择奇校验（PS=1），则“1”的数量为奇数）。

20.5.11 LIN（局域互连网络）模式

仅在支持 LIN 模式时才与本节相关。请参见第 20.4 节“USART 实现”。

通过将 USART_CR2 寄存器中的 LINEN 位置 1 来选择 LIN 模式。在 LIN 模式下，必须将以下位清零：

- USART_CR2 寄存器中的 CLKEN 位
- USART_CR3 寄存器中的 STOP[1:0]、SCEN、HDSEL 和 IREN 位。

LIN 发送

必须对 LIN 主设备发送应用第 20.5.2 节“FIFO 和阈值”中描述的程序。介绍的步骤，同时还具有以下区别：

- M 位清零以配置 8 位字长度
- 置位 LINEN 位以进入 LIN 模式。此时，将 SBKRQ 位置 1 会发送 13 个“0”位作为断路字符。然后会发送值为“1”的 2 个位以进行下一启动检测。

LIN 接收

使能 LIN 模式后，将激活断路检测电路。该检测完全独立于正常的 USART 接收器。在空闲状态或某个帧期间，只要发生断路即可检测出来。

接收器（USART_CR1 寄存器中 RE=1）使能后，电路便开始监测启动信号的 RX 输入。检测起始位的方法与搜索断路字符或数据的方法相同。检测到起始位后，电路会对接下来的位进行采样，方法与数据采样相同（第 8、第 9 和第 10 次采样）。如果 10 个（USART_CR2 寄存器中 LBDL = 0 时）或 11 个（USART_CR2 寄存器中 LBDL=1 时）连续位均检测为“0”，且其后跟随分隔符，则 USART_ISR 寄存器中的 LBDF 标志将会置 1。如果 LBDIE 位=1，则会生成中断。在验证断路前，会对分隔符进行检查，因为它表示 RX 线路已恢复到高电平。

如果在第 10 或第 11 次采样前已对“1”采样，则断路检测电路会取消当前检测，并重新搜索起始位。

如果禁止 LIN 模式（LINEN=0），接收器会作为正常的 USART 继续工作，不会再进行断路检测。

如果使能 LIN 模式（LINEN=1），只要发生数据拆包错误（例如，在“0”处检测到停止位，这种情况可能出现在任何断路帧中），接收器即会停止，直到断路检测电路接收到“1”（断路字不完整时）或接收到分隔符（检测到断路时）为止。

断路检测器状态机和断路标志的行为见图 123. LIN 模式下的断路检测（11 位断路长度 - LBDL 位置 1）。

下面是断帧的示例：

图 123. LIN 模式下的断路检测（11 位断路长度 - LBDL 位置 1）

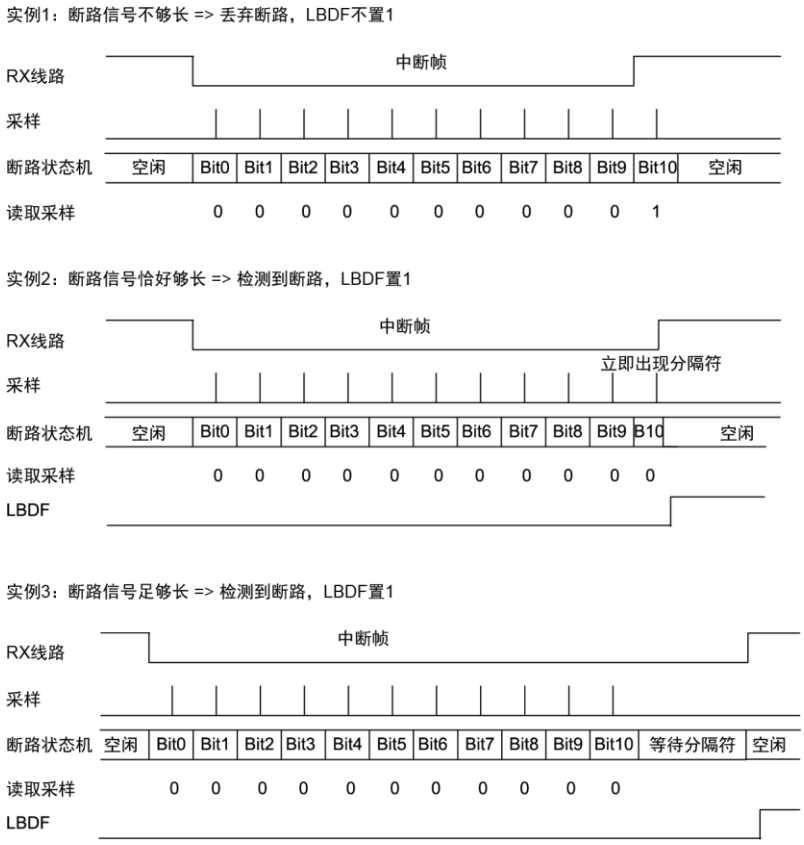
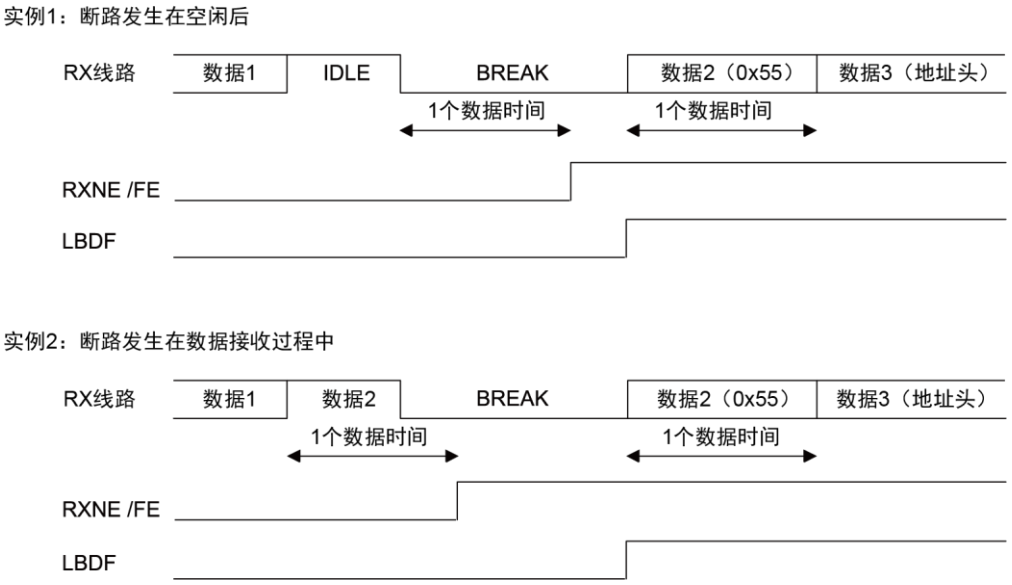


图 124. LIN 模式下的断路检测与数据拆包错误检测



20.5.12 USART 同步模式

主模式

通过将 USART_CR2 寄存器中的 CLKEN 位写入 1 来选择同步主模式。在同步模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN 位
- USART_CR3 寄存器中的 SCEN、HDSEL 和 IREN 位。

在此模式下，USART 可用于在主模式下控制双向同步串行通信。SCLK 引脚是 USART 发射器时钟的输出。在起始位或停止位期间，不会向 SCLK 引脚发送时钟脉冲。在最后一个有效数据位（地址标记）期间，会（也可能不会）生成时钟脉冲，这取决于 USART_CR2 寄存器中 LBCL 位的状态。USART_CR2 寄存器中的 CPOL 位用于选择时钟极性；USART_CR2 寄存器中的 CPHA 位用于选择外部时钟的相位（请参见图 125. 同步主设备发送的 USART 示例和图 126. USART 数据时钟时序图（M=0）。

在空闲状态、前导码模式和发送断路期间，外部 SCLK 时钟处于未激活状态。

USART 发射器在同步主模式下的工作方式与异步模式下完全相同。但是由于 SCLK 与 TX 同步（根据 CPOL 和 CPHA），因此 TX 上的数据是同步的。

在此模式下，USART 接收器的工作方式与异步模式下不同。如果 RE=1，则数据在 SCLK 上采样（上升或下降沿，取决于 CPOL 和 CPHA），而不会进行任何过采样。此时必须确保建立时间和保持时间（取决于波特率：1/16 位持续时间）。

注意： *在主模式下，SCLK 引脚与 TX 引脚协同工作。因此，仅当使能发射器（TE=1）且正在发送数据时（数据寄存器 USART_DR 已写入），才会提供时钟。这意味着，没有发送数据的情况下无法接收同步数据。*

图 125. 同步主设备发送的 USART 示例

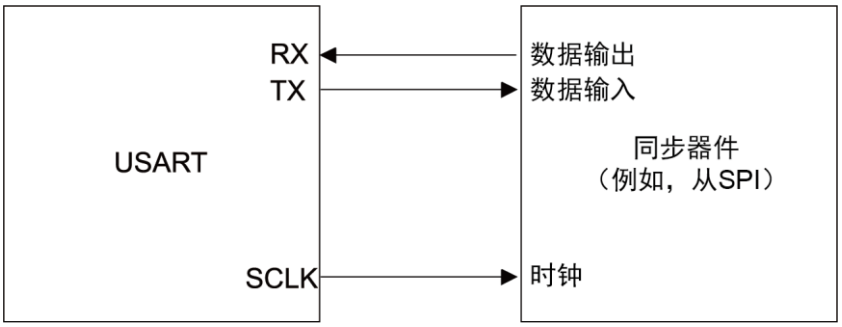


图 126. USART 数据时钟时序图 M=0

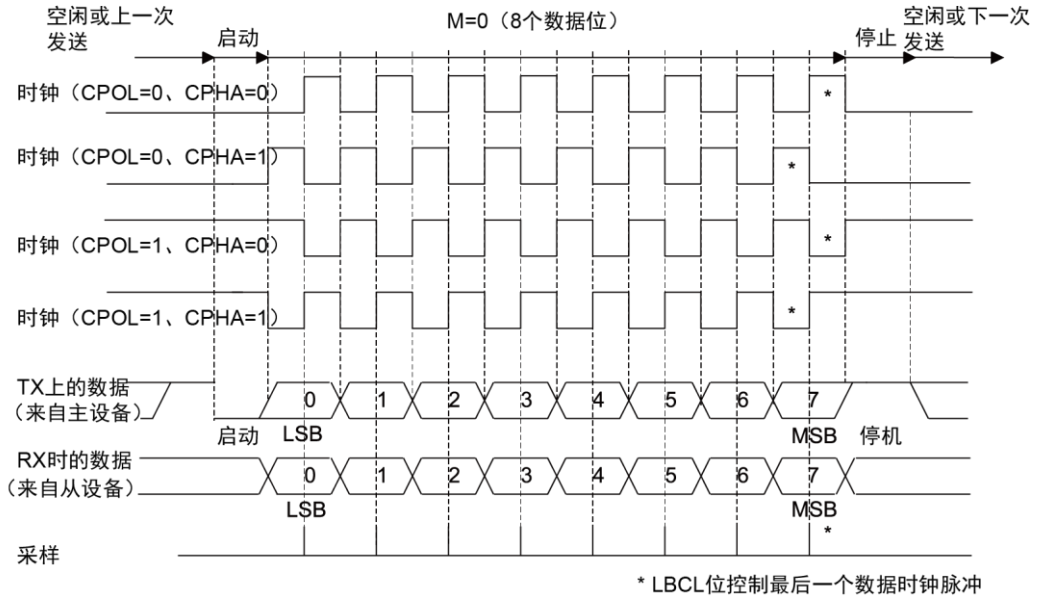


图 127. USART 数据时钟时序图（M 位 = 01）

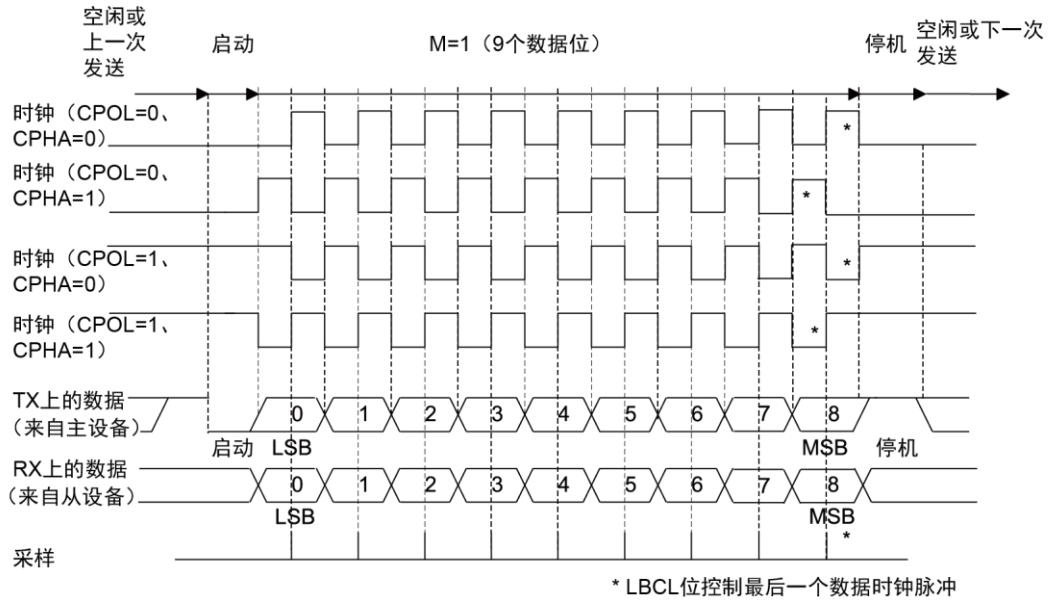
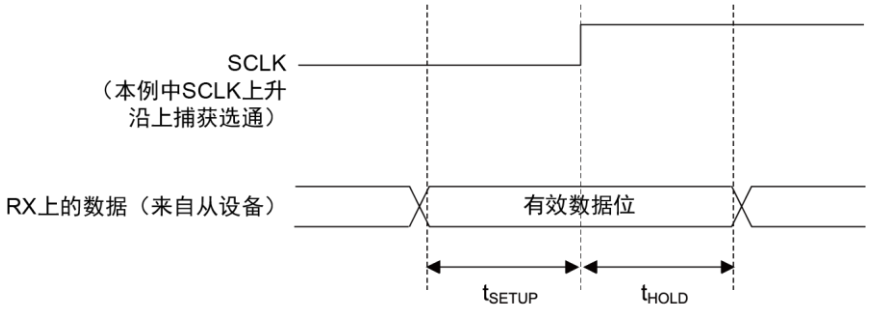


图 128. RX 数据建立/保持时间



t_{SETUP}=t_{HOLD} 1/16位持续时间

从模式

通过将 USART_CR2 寄存器中的 SLVEN 位写入 1 来选择同步从模式。在同步从模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN 和 CLKEN 位，
- USART_CR3 寄存器中的 SCEN 、HDSEL 和 IREN 位。

在此模式下，USART 可用于在从模式下控制双向同步串行通信。在从模式下，SCLK 引脚是 USART 的输入。

注意： *在 SPI 从模式下使用外设时，外设时钟源（fck_pres）必须大于 3xSCLK 输入时钟。*

USART_CR2 寄存器中的 CPOL 位用于选择时钟极性；USART_CR2 寄存器中的 CPHA 位用于选择外部时钟的相位（请参见图 125. 同步主设备发送的 USART 示例、图 126. USART 数据时钟时序图（M=0）和图 127. USART 数据时钟时序图（M 位 = 01）。

在从设备发送模式下，可使用下溢错误标志。如果在软件尚未将任何值加载到 USARTx_TDR 之前出现第一个数据发送时钟，此标志将置 1。

从设备支持硬件和软件 NSS 管理。

从设备选择（NSS）引脚管理

可以使用 USART_CR2 寄存器中的 DIS_NSS 位设置硬件或软件从设备选择管理。

- 软件 NSS 管理（DIS_NSS =1）
  总是选择 SPI 从设备并忽略 NSS 输入引脚。外部 NSS 引脚空闲，可供其它应用使用。
- 硬件 NSS 管理（DIS_NSS =0）
  SPI 从设备的选择取决于 NSS 输入引脚。在 NSS 为低电平时选择从设备，在 NSS 为高电平时取消选择。

注意： *当 USART 禁用（UE=0）时，为确保时钟脉冲正常工作，必须选择 LBCL（仅在 SPI 主模式下使用）、CPOL 和 CPHA 位。*

注意： *在 SPI 从模式下，必须在主设备开始通信前（或者当时钟稳定时在帧与帧之间）使能 USART。否则，如果在主设备处于一个帧的中间时使能 USART 从设备，它与主设备将不同步。从设备的数据寄存器需在通信时钟的第一个边沿之前或正在进行的通信结束之前准备就绪，否则 SPI 从设备将发送零。*

SPI 从设备下溢错误

当发生下溢错误时，SPI 从设备发送最后一个数据，直至软件中的下溢错误标志清零。

下溢标志在帧开始时置 1。

下溢错误标志通过 USART_ICR 寄存器中 UDRCF 位的置位进行清零。

在下溢状态下，可以对 TDR 寄存器执行写访问。将下溢错误清零后，可发送新数据。

如果发生下溢错误且 TDR 中没有写入新数据，则 TC 标志在帧结束时置 1。

注意： *如果在过于接近第一个 SCLK 发送边沿时将数据写入 USARTx_TDR，则可能发生下溢错误。为避免此下溢错误，应在第一个 SCLK 边沿的 3 个 UCLK 周期之前写入 USART_TDR。*

20.5.13 单线半双工通信

通过将 USART_CR3 寄存器中的 HDSEL 位置 1 来选择单线半双工模式。在此模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN 和 CLKEN 位
- USART_CR3 寄存器中的 SCEN 和 IREN 位。

USART 可以配置为遵循单线半双工协议，其中 TX 和 RX 线路从内部相连接。使用 USART_CR3 寄存器中的控制位 HDSEL 可在半双工通信和全双工通信间进行选择。

一旦向 HDSEL 位写入 1：

- TX 和 RX 线路从内部相连接
- 不能再使用 RX 引脚
- 无数据传输时，TX 引脚始终处于释放状态。因此，它在空闲状态或接收过程中用作标准 I/O。这意味着，必须对 I/O 进行配置，以便将 TX 配置为复用功能开漏并外接上拉电阻。

除此之外，通信协议与正常 USART 模式下的通信协议相似。此线路上的任何冲突必须由软件管理（例如，使用中央仲裁器）。尤其要注意，发送过程永远不会被硬件封锁，只要数据是在 TE 位置 1 的情况下写入，发送就会持续进行。

注意： *由于 TX 线路和 RX 线路连在一起，发送的所有数据与从外部器件接收的数据一样，都存储在 RX FIFO 中。在发送阶段后，软件必须谨慎地丢弃其“自己的”信息。在半双工模式下，由于没有针对节点之间可能发生的冲突的硬件保护，回读发送的数据以检查数据是否正确始终是一种明智的做法。如果不想让 RX FIFO 存储发送的值，则必须在发送时禁用接收器部分（通过将 USART_CR1 寄存器中的 RE 位清零）。*

20.5.14 接收器超时

接收器超时特性可通过将 USART_CR2 控制寄存器中的 RTOEN 位置 1 来使能。

使用 USARTx_RTOR 寄存器中的 RTO 位字段设定超时持续时间。

接收器超时计数器开始计数

- 当 STOP = 00 和 STOP = 11 时，从停止位的末尾开始
- 当 STOP = 10 时，从第二个停止位的末尾开始
- 当 STOP = 01 时，从停止位的开头开始。

当超时持续时间结束时，如果 USARTx_CR1 寄存器中的 RTOIE 位置 1，则 USARTx_ISR 寄存器中的 RTOF 标志置 1 并生成超时。

20.5.15 智能卡模式

仅在支持智能卡模式时才涉及本节内容。请参见第 20.4 节“USART 实现”。

通过将 USART_CR3 寄存器中的 SCEN 位置 1 选择智能卡模式。在智能卡模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN 位
- USART_CR3 寄存器中的 HDSEL 和 IREN 位。

为了给智能卡提供时钟，可将 CLKEN 位置 1。

智能卡接口支持符合 ISO 7816-3 标准的异步协议智能卡。同时支持 T=0（字符模式）和 T=1（块模式）。

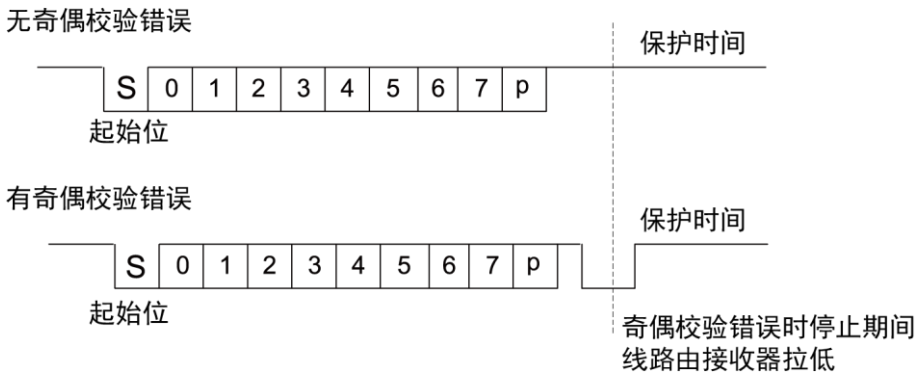
USART 应如下所示进行配置：

- 8 个位加奇偶校验：当 USART_CR1 寄存器中 M=1 且 PCE=1 时
- 发送和接收数据时使用 1.5 个停止位：当 USART_CR2 寄存器中 STOP=11 时。还可以选择 0.5 个停止位用于接收。

在 T=0（字符）模式下，奇偶校验错误在保护时间周期内的每个字符结束时指示。

图 129. ISO 7816-3 异步协议提供了有/无奇偶校验错误时数据线路上可见内容的示例。

图 129. ISO 7816-3 异步协议



连接到智能卡时，USART 的 TX 输出会驱动一条双向线（它也由智能卡驱动）。必须将 TX 引脚配置为开漏引脚。

智能卡模式采用单线半双工通信协议。

- 从发送移位寄存器发送数据会经过至少 1/2 个时钟周期的延迟。正常工作时，已满的发送移位寄存器会在下一个波特时钟边沿开始移位。在智能卡模式下，此发送过程还会进一步经过 1/2 波特时钟周期的延迟。

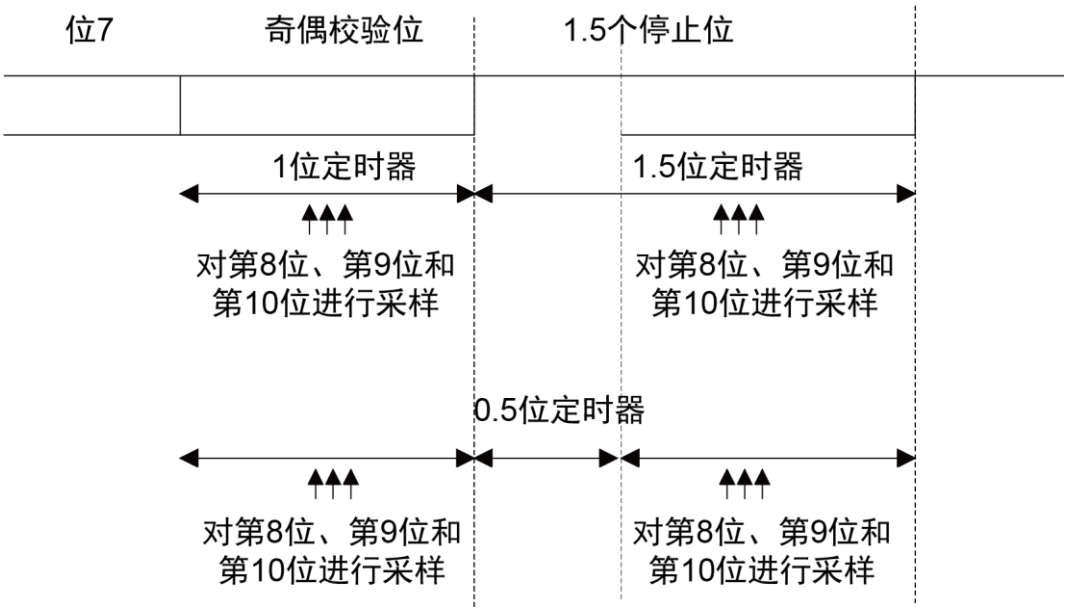
- 发送时，如果智能卡检测到奇偶校验错误，它会通过将线路驱动为低电平（NACK）告知 USART 此状态。此 NACK 信号（将发送线拉低 1 个时钟周期）会导致发射器端（配置为 1.5 个停止位）出现帧错误。USART 可根据协议自动重新发送数据。重试次数在 SCARCNT 位字段中编程。如果经过编程次数的重试后，USART 继续收到 NACK，USART 会停止发送并将该错误以数据拆包错误形式发出。可使用 USART_RQR 寄存器中的 TXFRQ 位将 TXE 位（FIFO 模式使能时为 TXFNF 位）置 1。
- 发送时智能卡自动重试：在 USART 检测 NACK 与重复字符的起始位之间插入 2.5 个波特率周期的延迟。接收最后一个重复字符后，立即将 TC 位置 1（无保护时间）。如果软件要重复此操作，必须确保标准要求的最短 2 个波特率周期。
- 如果在接收一个使用 1.5 个停止位编程的帧期间检测到奇偶校验错误，则在完成接收帧后，发送线会被拉低一个时钟周期。这是为了向智能卡指出发送到 USART 的数据尚未正确接收。如果 NACK 控制位置 1，接收器会向奇偶校验错误发送“NACK”信号；否则不会发送 NACK 信号（将在 T=1 模式下会使用）。如果接收的字符是错误的，将不激活 RXNE（FIFO 模式使能时为 RXFNE）/接收 DMA 请求。根据协议规范，智能卡必须重新发送相同的字符。如果经过 SCARCNT 位字段中指定的最大重试次数后接收到的字符仍然错误，USART 会停止发送 NACK 信号，并将错误以奇偶校验错误的形式发出。
- 接收时智能卡自动重试：如果 USART 向智能卡发送 NACK 信号，但智能卡不重复字符，则 BUSY 标志将保持置 1。
- 发送时，USART 会在两个连续字符之间插入保护时间（按照保护时间寄存器中编程的值）。由于保护时间在前一个字符的停止位后测量，因此必须将 GT[7:0]寄存器编程为所需 CGT（字符保护时间，如 7816-3 规范所定义）减去 12（一个字符的持续时间）。
- 通过对保护时间寄存器进行编程，可以延迟 TC 标志的置位。正常工作时，当发送移位寄存器为空且没有新的发送请求出现时，会对 TC 标志进行置位。在智能卡模式下，空的发送移位寄存器会触发保护时间计数器，使其递增计数至保护时间寄存器中的值。在此期间，TC 标志被强制为低电平。当保护时间计数器达到设置值时，TC 置位为高电平。TCBGT 标志可用于检测数据传输结束，无需等待保护时间结束。如果没有从智能卡接收到 NACK，此标志将在帧发送结束后置 1。
- TC 标志的取消置位不受智能卡模式的影响。
- 如果在发送端检测到数据拆包错误（由来自接收器的 NACK 信号引起），则发送端的接收块不会将 NACK 作为起始位进行检测。根据 ISO 协议，接收到的 NACK 信号的持续时间可以是 1 或 2 个时钟周期。
- 在接收端，如果检测到奇偶校验错误并发送了 NACK 信号，则接收端不会将 NACK 作为起始位进行检测。

注意： *中断字符在智能卡模式下无效。带有数据拆包错误的 0x00 数据被视为数据，而非中断。*

注意： *当翻转 TE 位时，不会发送空闲帧。空闲帧（在其它配置中进行了定义）在 ISO 协议中未进行定义。*

图 130. 使用 1.5 个停止位的奇偶校验错误检测详细描述了 USART 如何对 NACK 信号进行采样。在本例中，USART 正在发送数据并配置了 1.5 个停止位。USART 的接收部分已被使能，以检查数据的完整性和 NACK 信号。

图 130. 使用 1.5 个停止位的奇偶校验错误检测



USART 可以通过 SCLK 输出为智能卡提供时钟。在智能卡模式下，SCLK 仅通过一个 5 位预分频器由内部外设输入时钟提供。分频比在预分频器寄存器 USART_中进行配置。SCLK 频率可设定为范围 $f_{CKPRES}/2$ 至 $f_{CKPRES}/62$ 以内的值，其中的 f_{CKPRES} 是经设定预分频因子分频的外设输入时钟。

块模式（T=1）

在 T=1（块）模式下，可通过将 UART_CR3 寄存器中的 NACK 位清零来停用奇偶校验错误发送。

在请求读取智能卡时，在块模式下，软件必须将 RTOR 寄存器设定为 BWT（块等待时间）- 11 值。如果在该时间段到期前没有收到来自智能卡的应答，将生成超时中断。如果该时间段后接收到第一个字符，则通过 RXNE/RXFNE 中断发出信号指示。

注意： *即使在使用 DMA 模式下的 USART 从块模式下的智能卡读取时，也必须使能 RXNE/RXFNE 中断。同时，只有在接收到第一个字节后才可使能 DMA。*

在接收到第一个字符（RXNE/RXFNE 中断）后，必须将 RTO 寄存器设定为 CWT（字符等待时间）- 11 值，以便自动检查两个连续字符之间的最长等待时间。此时间以波特率时间单位表示。前一个字符结束后，如果智能卡未在小于 CWT 的时间段内发送新字符，USART 将通过 RTOF 标志和中断（当 RTOIE 位置 1 时）向软件指示此情况。

注意： *根据智能卡协议的定义，应从最后一个字符的开头（起始位）定义 BWT/CWT 值。必须将 RTO 寄存器分别编程为 BWT 11 或 CWT 11，并考虑最后一个字符本身的长度。*

块长度计数器用于对 USART 接收到的所有字符进行计数。当 USART 进行发送时，该计数器复位。块长度由智能卡在块的第三个字节（起始字段）中传达。必须将此值编程到 USART_RTOR 寄存器中的 BLEN 字段。使用 DMA 模式时，在块开始之前，必须将此寄存器字段编程为最小值（0x0）。对于该值，在接收到第四个字符后生成中断。软件必须读取 LEN 字段（第三个字节），其值必须从接收缓存区中读取。

在中断驱动接收模式下，块长度可以由软件或者通过编程 BLEN 值来检查。但是在块开始前，可以编程 BLEN 的最大值（0xFF）。在接收到第三个字符数据后可设定实际使用的值。

如果块使用 LRC 纵向冗余校验（1 个结尾字节），则 BLEN=LEN。如果块使用 CRC 机制（2 个结尾字节），则必须设定 BLEN=LEN+1。块总长度（包括起始字段、结尾字段和信息字段）等于 BLEN+4。块结束的信号通过 EOBF 标志和中断（EOBIE 位置 1 时）发送给软件。如果块长度出现错误，则通过 RTO 中断（字符等待时间上溢）发送块结束信号。

注意： *错误检查代码（LRC/CRC）必须通过软件计算/验证。*

正向约定和反向约定

智能卡协议定义了两种约定：正向约定和反向约定。

正向约定的定义如下：LSB 优先，逻辑位的值 1 对应于线路的 H 状态且奇偶校验结果为偶。为了使用此约定，必须对以下控制位进行编程：MSBFIRST=0，DATAINV=0（默认值）。

反向约定的定义如下：MSB 优先，逻辑位的值 1 对应于信号线的 L 状态且奇偶校验结果为偶。为了使用此约定，必须对以下控制位进行编程：MSBFIRST=1，DATAINV=1。

注意： *将逻辑数据值取反（0=H，1=L）时，奇偶校验位将同样取反。*

为识别智能卡约定，智能卡会将初始字符 TS 作为 ATR（复位应答）的第一个字符发送。两种可能的 TS 模式为：LHHL LLL LLH 和 LHHL HHH LLH。

- （H）LHHL LLL LLH 建立反向约定：状态 L 编码为值 1，时间分量 2 传送最高有效位（MSB 在前）。按反向约定解码时，传送的字节等于“3F”。
- （H）LHHL HHH LLH 建立正向约定：状态 H 编码为值 1，时间分量 2 传送最低有效位（LSB 在前）。按正向约定解码时，传送的字节等于“3B”。

在 2 到 10 的九个时间分量中，如果有偶数个位设置为 1，则字符的奇偶校验正确。

由于 USART 不了解智能卡使用哪种约定，因此 USART 需要能够识别任意一种模式并相应操作。模式识别不在硬件中完成，而是通过软件序列完成。此外，假定 USART 配置为正向约定（默认）而卡以反向约定应答，TS = LHHL LLL LLH => USART 接收到的字符为“03”且奇偶校验结果为奇。

因此，有两种方法可用于识别 TS 模式：

方法 1

将 USART 编程为标准智能卡模式/正向约定。这种情况下，TS 模式接收会向智能卡生成奇偶校验错误中断和错误信号。

- 奇偶校验错误中断通知软件，卡未以正向约定正确地应答。之后，软件重新将 USART 编程为反向约定
- 为响应错误信号，智能卡会重试同一 TS 字符，此时重新编程后的 USART 将正确接收该字符。

或者，为应答奇偶校验错误中断，软件可决定重新编程 USART，同时向智能卡生成新的复位指令，然后再次等待 TS。

方法 2

将 USART 编程为 9 位/无奇偶校验模式，无位反向。在此模式下，按如下方式接收两种 TS 模式中的任一种：

- （H）LHHL LLL LLH = 0x103 -> 选择反向约定
- （H）LHHL LLL LLH = 0x13B -> 选择正向约定。

软件根据这两种模式检查接收到的字符，如果其中任意一种匹配，则相应编程 USART 以接收下一个字符。

如果二者均未能识别，则需要复位智能卡并重新开始协商过程。

20.5.16 IrDA SIR ENDEC 模块

仅在支持 IrDA 模式时才涉及本节内容。请参见第 20.4 节“USART 实现”。

通过将 USART_CR3 寄存器中的 IREN 位置 1 来选择 IrDA 模式。在 IrDA 模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN、STOP 和 CLKEN 位
- USART_CR3 寄存器中的 SCEN 和 HDSEL 位。

IrDA SIR 物理层规定使用归零反转（RZI）调制方案，该方案用红外光脉冲代表逻辑 0（参见图 131. IrDA SIR ENDEC - 框图）。

SIR 发送编码器用于调制 USART 发出的非归零（NRZ）位流。输出脉冲流会发送到外部输出驱动器和红外线 LED。对于 SIR ENDEC，USART 只支持不超过 115.2 Kbps 的比特率。在正常模式下，所发送的脉冲宽度规定为一个位周期的 3/16。

SIR 接收解码器用于解调由红外探测器发出的归零位流，并将接收到的 NRZ 串行位流输出到 USART。在空闲状态下，解码器输入通常为高电平（标记状态）。发送编码器输出的极性与解码器输入相反。当解码器输入为低电平时，会检测到起始位。

- IrDA 是一个半双工通信协议。如果发射器忙（USART 正在向 IrDA 编码器发送数据时），则 IrDA 解码器会忽略 IrDA 接收线上的所有数据；如果接收器忙（USART 正在接收来自 IrDA 解码器的解码数据时），则 IrDA 不会对 USART 发送到 IrDA 的 TX 上的数据进行编码。在接收数据时，应避免同时进行发送，因为这样做可能会破坏要发送的数据。
- 0 作为高电平脉冲发送，而 1 作为 0 发送。脉冲宽度指定为正常模式下选定位周期的 3/16（参见图 132. IrDA 数据调制（3/16） - 正常模式）。
- SIR 解码器用于将兼容 IrDA 的接收信号转换为 USART 的位流。
- SIR 接收逻辑将高电平状态视为逻辑“1”，将低电平脉冲视为逻辑“0”。
- 发送编码器输出的极性与解码器输入相反。SIR 输出在空闲时处于低电平状态。
- IIrDA 规范要求脉冲容忍值要大于 1.41 µs。可接受的脉冲宽度可通过寄存器设置。接收器端的干扰检测逻辑会滤除宽度小于 2 个 PSC 周期的脉冲（PSC 是在 USART_GTPR 中编程的预分频器值）。宽度小于 1 个 PSC 周期的脉冲都将被拒绝，但宽度大于 1 个而小于 2 个周期的脉冲可能被接受也可能被拒绝，而宽度大于 2 个周期的脉冲将被接受作为有效脉冲。当 PSC=0 时，IrDA 编码器/解码器不工作。
- 接收器能够与低功耗发射器进行通信。
- 在 IrDA 模式下，USART_CR2 寄存器中的 STOP 位必须配置为“1 个停止位”。

IrDA 低功耗模式

发射器

在低功耗模式下，脉冲宽度不再保持为位周期的 3/16。此时的脉冲宽度为低功耗波特率的 3 倍，最小可为 1.42 MHz。通常此值是 1.8432 MHz（1.42 MHz < PSC < 2.12 MHz）。低功耗模式下的可编程分频器会对系统时钟进行分频，以达到此值。

接收器：

在低功耗模式下接收与在正常模式下接收类似。为进行干扰检测，USART 应丢弃持续时间短于 1/PSC 的脉冲。只有当持续时间大于 2 个 IrDA 低功耗波特时钟周期（USART_GTPR 寄存器中的 PSC 值）时，才是有效低电平。

注意： *宽度小于两个但大于一个 PSC 周期的脉冲可能被接受，也可能被拒绝。*

注意： *接收器的建立时间应由软件进行管理。IrDA 物理层规范规定发送和接收之间至少要经过 10 ms 的延迟（IrDA 是一个半双工协议）。*

图 131. IrDA SIR ENDEC - 框图

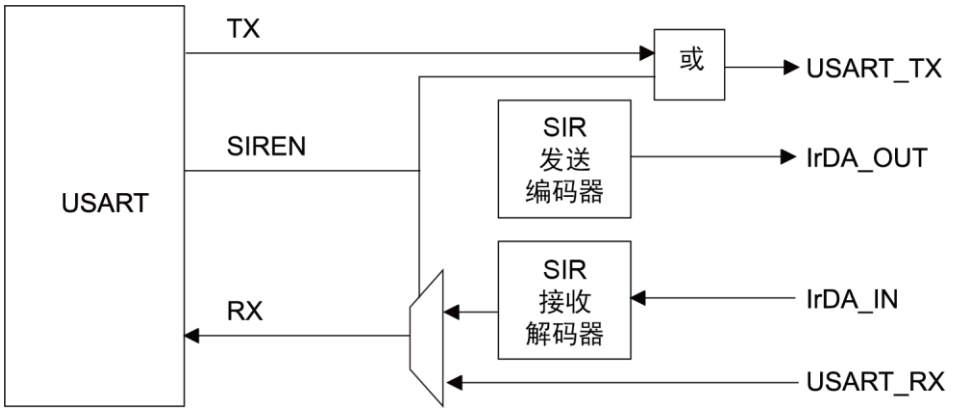
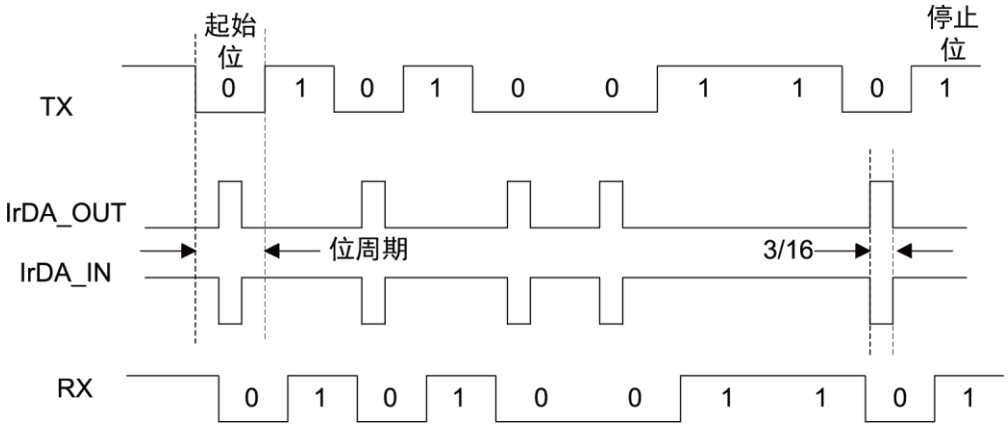


图 132. IrDA 数据调制（3/16） - 正常模式



20.5.17 使用 DMA 进行连续通信

USART 能够使用 DMA 执行连续通信。接收缓冲区和发送缓冲区的 DMA 请求是独立的。

注意： *请参见第 20.4 节“USART 实现”，以确定是否支持 DMA 模式。如果不支持 DMA 模式，则按照第 20.5.4 节“接收器”所述使用 USART。如需执行连续通信，可在 FIFO 禁用时，将 USART_ISR 寄存器中的 TXE/RXNE 标志清零。*

使用 DMA 发送

将 USART_CR3 寄存器中的 DMAT 位置 1 可以使能 DMA 模式进行发送。每当 TXE 标志（FIFO 模式使能时为 TXFNF 标志）置 1 时，将数据从使用 DMA 外设配置的 SRAM 区（参见第 10 节“DMA 控制器（DMA）”）加载到 USART_TDR 寄存器。要映射一个 DMA 通道以进行 USART 发送，请按以下步骤操作（x 表示通道编号）：

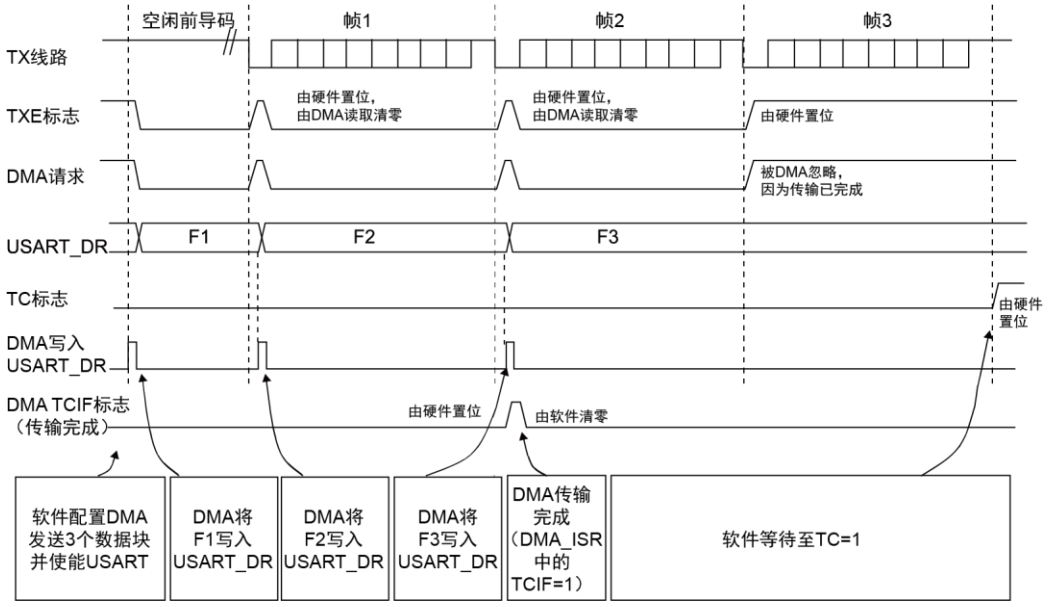
1. 在 DMA 控制寄存器中写入 USARTx_TDR 寄存器地址，将其配置为传输的目标地址。在每个 TXE（FIFO 模式使能时为 TXFNF）事件后，数据会从存储器移动到此地址。
2. 在 DMA 控制寄存器中写入存储器地址，将其配置为传输的源地址。在每个 TXE（FIFO 模式使能时为 TXFNF）事件后，数据会从该存储区加载到 USART_TDR 寄存器中。
3. 在 DMA 控制寄存器中配置要传输的总字节数。

4. 在 DMA 寄存器中配置通道优先级。
5. 根据应用的需求，在完成一半或全部传输后产生 DMA 中断。
6. 通过将 USART_ICR 寄存器中的 TCCF 位置 1，将 USART_ISR 寄存器中的 TC 标志清零。
7. 在 DMA 寄存器中激活该通道。

当达到在 DMA 控制器中设置的数据传输量时，DMA 控制器会在 DMA 通道的中断向量上产生一个中断。

在发送模式下，DMA 对所有要发送的数据执行了写操作（DMA_ISR 寄存器中的 TCIF 标志置 1）后，可以对 TC 标志进行监视，以确保 USART 通信已完成。在禁止 USART 或进入停止模式前必须执行此步骤，以避免损坏最后一次发送。软件必须等待直到 TC=1。TC 标志在所有数据发送期间都保持清零状态，然后在最后一帧发送结束时由硬件置 1。

图 133. 使用 DMA 发送



注意： *当 FIFO 管理使能时，发送 FIFO 未满（即 TXFNF = 1）会触发 DMA 请求。*

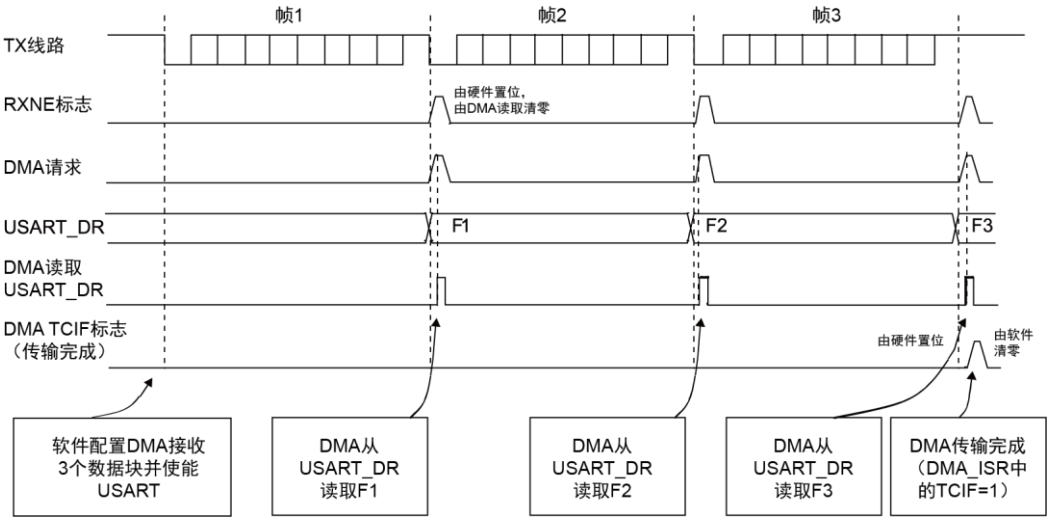
使用 DMA 接收

将 USART_CR3 寄存器中的 DMAR 位置 1 可以使能 DMA 模式进行接收。每当接收到数据字节时，将数据从 USART_RDR 寄存器加载到使用 DMA 外设配置的 SRAM 区（参见第 10 节“DMA 控制器（DMA）”）。要映射一个 DMA 通道以进行 USART 接收，请按以下步骤操作：

1. 在 DMA 控制寄存器中写入 USART_RDR 寄存器地址，将其配置为传输的源地址。在每个 RXNE（FIFO 模式使能时为 RXFNE）事件后，数据会从此地址移动到存储器。
2. 在 DMA 控制寄存器中写入存储器地址，将其配置为传输的目标地址。在每个 RXNE（FIFO 模式使能时为 RXFNE）事件后，数据会从 USART_RDR 寄存器加载到该存储区。
3. 在 DMA 控制寄存器中配置要传输的总字节数。
4. 在 DMA 控制寄存器中配置通道优先级。
5. 根据应用的需求，在完成一半或全部传输后产生中断。
6. 在 DMA 控制寄存器中激活该通道。

当达到在 DMA 控制器中设置的数据传输量时，DMA 控制器会在 DMA 通道的中断向量上产生一个中断。

图 134. 使用 DMA 接收



注意： *当 FIFO 管理使能时，接收 FIFO 不为空（即 RXFNE = 1）会触发 DMA 请求。*

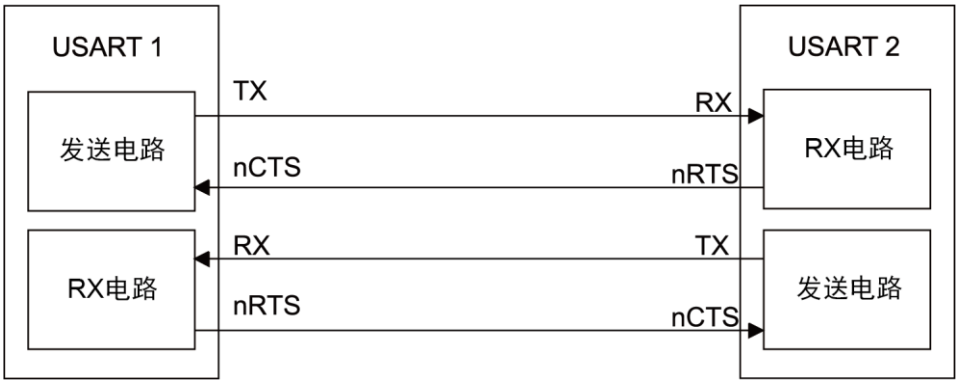
多缓冲区通信中的错误标志和中断生成

在多缓冲区通信中，如果传输过程中发生任何错误，都会在当前字节后放置错误标志。如果中断使能标志置 1，则会产生中断。对于在单字节接收中与 RXNE（FIFO 模式使能时为 RXFNE）一同置位的数据拆包错误、上溢错误和噪声标志，有一个单独的错误标志中断使能位（USART_CR3 寄存器中的 EIE 位），如果此位置 1，将在发生这些错误中的任意一种时在当前字节后使能中断。

20.5.18 RS232 硬件流控制和 RS485 驱动器使能

使用 nCTS 输入和 nRTS 输出可以控制 2 个器件间的串行数据流。图 135. 2 个 USART 间的硬件流控制显示了如何在该模式下连接 2 个器件。

图 135. 2 个 USART 间的硬件流控制

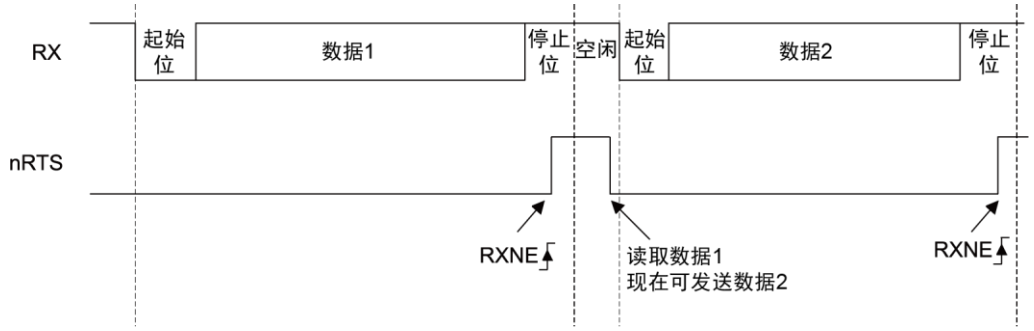


分别向 USART_CR3 寄存器中的 RTSE 位和 CTSE 位写入 1，可以分别使能 RS232 RTS 和 CTS 流控制。

RS232 RTS 流控制

如果使能 RTS 流控制（RTSE=1），只要 USART 接收器准备好接收新数据，便会将 nRTS 变为有效（连接到低电平）。当接收寄存器已满时，会将 nRTS 变为无效，表明发送过程会在当前帧结束后停止。图 136. RS232 RTS 流控制显示了在 RTS 流控制使能时进行通信的示例。

图 136. RS232 RTS 流控制



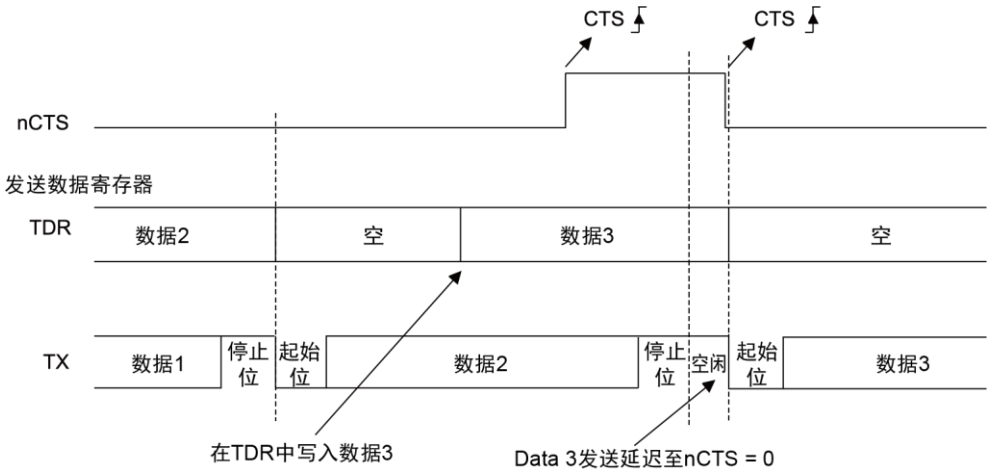
注意： *当 FIFO 模式使能时，nRTS 仅在 RXFIFO 已满时变为无效。*

RS232 CTS 流控制

如果使能 CTS 流控制（CTSE=1），则发射器会在发送下一帧前检查 nCTS。如果 nCTS 有效（连接到低电平），则会发送下一数据（假设数据已准备好发送，即 TXE/TXFE=0）；否则不会进行发送。如果在发送过程中 nCTS 变为无效，则当前发送完成之后，发射器停止。

当 CTSE=1 时，只要 CTS 发生变化，CTSIF 状态位便会由硬件自动置 1。这指示接收器是否已准备好进行通信。如果 USART_CR3 寄存器中的 CTSIE 位置 1，则会产生中断。图 137. RS232 CTS 流控制显示了在 CTS 流控制使能时进行通信的示例。

图 137. RS232 CTS 流控制



注意： *为正常运行，必须在当前字符结束前至少 3 个 USART 时钟源周期内使能 nCTS。此外还应注意，当脉冲短于 2 个 PCLK 周期时无法将 CTSCF 标志置 1。*

RS485 驱动器使能

驱动器使能功能可通过将 USART_CR3 控制寄存器中的 DEM 位置 1 使能。这样用户便可通过 DE（驱动器使能）信号激活外部收发器控制。使能时间为激活 DE 信号与 START 位开始间的时间。可以通过 USART_CR1 控制寄存器中的 DEAT [4:0]位字段编程禁止时间。禁止时间为发送的消息中最后一个停止位结束与取消激活 DE 信号间的时间。可以通过 USART_CR1 控制寄存器中的 DEDT [4:0]位字段编程禁止时间。DE 信号的极性可使用 USART_CR3 控制寄存器中的 DEP 位配置。

在 USART 中，DEAT 和 DEDT 以采样时间单位表示（1/8 或 1/16 位持续时间，具体取决于过采样速率）。

20.6 USART 中断

表 71. USART 中断请求

中断事件	事件标志	使能控制位
发送数据寄存器为空	TXE	TXEIE
发送 FIFO 未满	TXFNF	TXFNFIE
发送 FIFO 为空	TXFE	TXFEIE
CTS 中断	CTSIF	CTSIE
发送完成	TC	TCIE
在保护时间之前发送完成	TCBGT	TCBGTIE
接收数据寄存器不为空（已准备好读取数据）	RXNE	RXNEIE
接收 FIFO 非空	RXFNE	RXFNEIE
接收 FIFO 已满	RXFF	RXFFIE
检测到溢出错误	ORE	RXNEIE/RXF- NEIE
检测到空闲线路	IDLE	IDLEIE
奇偶校验错误	PE	PEIE
LIN 断路	LBDF	LBDIE
多缓冲区通信中的噪声标志、上溢错误和数据拆包错误	NF 或 ORE 或 FE	EIE
字符匹配	CMF	CMIE
接收器超时错误	RTOF	RTOIE
块结束	EOBF	EOBIE
SPI 从设备下溢错误	UDR	EIE

如果相应的使能控制位置 1，则这些事件会生成中断。

20.7 USART 寄存器

请参见第 1.3 节“缩略语”获取寄存器说明中所用缩略语的列表。

20.7.1 控制寄存器 1（USARTx_CR1）

偏移地址：0x00

复位值：0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXFFI E	TXFEIE	FIFOE N	M1	EOBIE	RTOIE	DEAT[4:0]					DEDT[4:0]				
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
OVER8	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXEIE/ TXFNFI E	TCIE	RXNEIE/ RXFNEIE	IDLEIE	TE	RE	Res.	UE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

位 31	**RXFFIE**：RXFIFO 已满中断使能。此位由软件置 1 和清零。 0：禁止中断 1：当 USART_ISR 寄存器中的 RXFF=1 时，生成 USART 中断 注意：当 FIFO 模式禁用时，此位保留且必须保持复位值。
位 30	**TXFEIE**：TXFIFO 空中断使能。此位由软件置 1 和清零。 0：禁止中断 1：当 USART_ISR 寄存器中的 TXFE=1 时，生成 USART 中断 注意：当 FIFO 模式禁用时，此位保留且必须保持复位值。
位 29	**FIFOEN**：FIFO 模式使能。 此位由软件置 1 和清零。 0：FIFO 模式禁用 1：FIFO 模式使能 只有在禁止 USART（UE=0）后才能写入此位字段。 注意：可以在 SPI 主/从模式和智能卡模式下对标准 UART 通信使用 FIFO 模式。不得在 IrDA 和 LIN 模式下使能 FIFO 模式。
位 28	**M1**：字长。 此位和位 12（M0）用于确定字长度。该位由软件置 1 或清零。 M[1:0] = 00:1 起始位，8 数据位，n 停止位 M[1:0] = 01:1 起始位，9 数据位，n 停止位 M[1:0] = 10:1 起始位，7 数据位，n 停止位 只有在禁止 USART（UE=0）时才能写入此位。 注意：在 7 位数据长度模式下，不支持智能卡模式、LIN 主模式和自动波特率（0x7F 和 0x55 帧检测）。
位 27	**EOBIE**：块结束中断使能。此位由软件置 1 和清零。 0：禁止中断 1：USART_ISR 寄存器中的 EOBF 标志置 1 时生成 USART 中断 注意：如果 USART 不支持智能卡模式，该位保留并由硬件强制清零。请参见第 20.4 节“USART 实现”。
位 26	**RTOIE**：接收器超时中断使能。此位由软件置 1 和清零。 0：禁止中断

	1：USART_ISR 寄存器中的 RTOF 标志置 1 时生成 USART 中断 注意：如果 USART 不支持接收器超时功能，该位保留并由硬件强制清零。第 20.4 节“USART 实现”。
位 25:21	**DEAT[4:0]**：驱动器使能置位时间。 该 5 位值用于定义激活 DE（启动器使能）信号与起始位开始间的时间。此时间以采样时间单位表示（1/8 或 1/16 位持续时间，具体取决于过采样速率）。 只有在禁止 USART（UE=0）后才能写入此位字段。 注意：如果不支持驱动器使能功能，该位保留并且必须保持清零。请参见第 20.4 节“USART 实现”。
位 20:16	**DEDT[4:0]**：驱动器使能取消置位时间。 该 5 位值用于定义发送的消息中最后一个停止位结束与取消激活 DE（驱动器使能）信号间的时间。此时间以采样时间单位表示（1/8 或 1/16 位持续时间，具体取决于过采样速率）。 如果在 DEDT 时间内对 USART_TDR 寄存器执行写操作，则新数据仅在经过 DEDT 和 DEAT 时间后才会发送。 只有在禁止 USART（UE=0）后才能写入此位字段。 注意：如果不支持驱动器使能功能，该位保留并且必须保持清零。请参见第 20.4 节“USART 实现”。
位 15	**OVER8**：过采样模式。 0:16 倍过采样 1:8 倍过采样 只有在禁止 USART（UE=0）时才能写入此位。 注意：在 LIN、IrDA 和智能卡模式下，此位必须保持清零。
位 14	**CMIE**：字符匹配中断使能。此位由软件置 1 和清零。 0：禁止中断 1：USART_ISR 寄存器中的 CMF 标志置 1 时生成 USART 中断
位 13	**MME**：静默模式使能。 此位激活 USART 的静默模式功能。当此位置 1 时，USART 可按照 WAKE 位的定义，在活动模式与静默模式之间切换。该位由软件置 1 和清零。 0：接收器永久处于活动模式 1：接收器可在静默模式和活动模式之间切换
位 12	**M0**：字长。 此位和位 28（M1）用于确定字长度。该位由软件置 1 或清零。 请参见位 28（M1）的说明。 只有在禁止 USART（UE=0）时才能写入此位。
位 11	**WAKE**：接收器唤醒方法。 此位决定了从静默模式唤醒 USART 的方法。该位由软件置 1 或清零。 0：空闲线 1：地址标记 只有在禁止 USART（UE=0）后才能写入此位字段。
位 10	**PCE**：奇偶校验控制使能。 该位选择硬件奇偶校验控制（生成和检测）。使能奇偶校验控制时，计算出的奇偶校验位被插入到 MSB 位置（如果 M=1，则为第 9 位；如果 M=0，则为第 8 位），并对接收到的数据检查奇偶校验位。此位由软件置 1 和清零。一旦该位置 1，PCE 在当前字节的后面处于活动状态（在接收和发送时）。 0：禁止奇偶校验控制 1：使能奇偶校验控制 只有在禁止 USART（UE=0）后才能写入此位字段。

位 9	**PS**：奇偶校验的选择。 该位用于在使能奇偶校验生成/检测（PCE 位置 1）时选择奇校验或偶校验。该位由软件置 1 和清零。在当前字节之后选择奇偶校验。 0：偶校验 1：奇校验 只有在禁止 USART（UE=0）后才能写入此位字段。
位 8	**PEIE**：PE 中断使能。 此位由软件置 1 和清零。 0：禁止中断 1：当 USART_ISR 寄存器中的 PE=1 时，生成 USART 中断
位 7	**TXEIE/TXFNFIE**：发送数据寄存器为空/TXFIFO 未满中断使能。 此位由软件置 1 和清零。 0：禁止中断 1：当 USART_ISR 寄存器中的 TXE/TXFNF=1 时，生成 USART 中断
位 6	**TCIE**：传送完成中断使能。此位由软件置 1 和清零。 0：禁止中断 1：当 USART_ISR 寄存器中的 TC=1 时，生成 USART 中断
位 5	**RXNEIE/RXFNEIE**：接收数据寄存器不为空/RXFIFO 不为空中断使能。 此位由软件置 1 和清零。 0：禁止中断 1：当 USART_ISR 寄存器中的 ORE=1 或 RXNE/RXFNE=1 时，生成 USART 中断
位 4	**IDLEIE**：IDLE 中断使能。 此位由软件置 1 和清零。 0：禁止中断 1：当 USART_ISR 寄存器中的 IDLE=1 时，生成 USART 中断
位 3	**TE**：发射器使能。 该位使能发射器。该位由软件置 1 和清零。 0：禁止发射器 1：使能发射器 注意：除了在智能卡模式下以外，传送期间 TE 位上的“0”脉冲（“0”后紧跟的是“1”）会在当前字的后面发送一个前导码（空闲线路）。为生成空闲字符，TE 不能立即写入 1。为确保所需的持续时间，软件可轮询 USART_ISR 寄存器中的 TEACK 位。 智能卡模式下，当 TE 置 1 时，在发送开始前存在 1 位的时间延迟。
位 2	**RE**：接收器使能。 该位使能接收器。该位由软件置 1 和清零。 0：禁止接收器 1：使能接收器并开始搜索起始位
位 1	保留，必须保持复位值。
位 0	**UE**：使能 USART。 此位清零后，USART 预分频器和输出将立即停止，并丢弃当前操作。USART 的配置会被保留，但 USART_ISR 中的所有状态标志均复位。此位由软件置 1 和清零。 0：USART 预分频器和输出禁用，低功耗模式 1：使能 USART 注意：为进入低功耗模式而不在线路上生成错误，之前必须复位 TE 位，并且软件必须等待 USART_ISR 寄存器中的 TC 位置 1 后才能复位 UE 位。

	UE = 0 时也会复位 DMA 请求，因必须在复位 UE 位前禁止 DMA 通道。 注意：在智能卡模式（SCEN = 1）下，当 CLKEN = 1 时，SCLK 始终可用，与 UE 位的值无关。

20.7.2 控制寄存器 2（USARTx_CR2）

偏移地址：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	MSBFI RST	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	Res.	LBDIE	LBDL	ADDM7	DIS_N SS.	Res.	Res.	SLVEN.
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw			rw

位 31:28	**ADD[7:4]**：USART 节点的地址。 此位字段用于指定 USART 节点的地址或要识别的字符代码。 在静默模式或停止模式下的多处理器通信中，它被用于通过 7 位地址标记检测进行唤醒。发射器发送字符的 MSB 应为 1。此位字段还可用于正常接收和静默模式无效时的字符检测（例如，ModBus 协议中的块结束检测）。这种情况下，接收到的整个字符（8 位）将与 ADD[7:0]值进行比较，如果匹配，CMF 标志将置 1。 仅在禁止接收（RE = 0）或禁止 USART（UE=0）时才能写入该位字段。
位 27:24	**ADD[3:0]**：USART 节点的地址。 此位字段用于指定 USART 节点的地址或要识别的字符代码。 在静默模式或停止模式下的多处理器通信中，它被用于通过地址标记检测进行唤醒。 仅在禁止接收（RE = 0）或禁止 USART（UE=0）时才能写入该位字段。
位 23	**RTOEN**：接收器超时使能。 此位由软件置 1 和清零。 0：接收器超时特性禁用 1：接收器超时特性使能 使能此功能后，如果 RX 线路在 RTOR（接收器超时寄存器）中编程的持续时间内处于空闲状态（无接收），则 USART_ISR 寄存器中的 RTOF 标志置 1。 注意：如果 USART 不支持接收器超时功能，该位保留并由硬件强制清零。请参见第 20.4 节“USART 实现”。
位 22:21	**ABRMOD[1:0]**：自动波特率模式。 这些位将由软件置 1 和清零。 00：通过测量起始位检测波特率 01：下降沿至下降沿的测量（接收到的帧必须以一个 = 1 的位开头 -> 帧 = Start10xxxxxx） 10:0x7F 帧检测 11:0x55 帧检测 仅在 ABREN = 0 时或禁止 USART（UE=0）时才能写入该位字段。 注意：如果 DATAINV=1 且/或 MSBFIRST=1，这些模式必须与在线路上时相同，例如 MSBFIRST 的 0xAA。 如果 USART 不支持自动波特率功能，则此位保留并由硬件强制清零。请参见第 20.4 节“USART 实现”。
位 20	**ABREN**：自动波特率使能。 此位由软件置 1 和清零。0:自动波特率检测禁用 1：自动波特率检测使能 注意：如果 USART 不支持自动波特率功能，则此位保留并由硬件强制清零。请参见第 20.4 节“USART 实现”。

位 19	**MSBFIRST**：最高有效位优先。 此位由软件置 1 和清零。 0：发送/接收数据时位 0 在前，后跟起始位 1：发送/接收数据时 MSB（位 7/8）在前，后跟起始位。只有在禁止 USART（UE=0）后才能写入此位字段。
位 18	**DATAINV**：二进制数据取反。 此位由软件置 1 和清零。 0：按正/正向逻辑发送/接收数据寄存器中的逻辑数据。（1=H，0=L） 1：按负/反向逻辑发送/接收数据寄存器中的逻辑数据。（1=L，0=H）。奇偶校验位也取反。 只有在禁止 USART（UE=0）后才能写入此位字段。
位 17	**TXINV**：TX 引脚有效电平取反。此位由软件置 1 和清零。 0：TX 引脚信号使用标准逻辑电平工作（VDD=1/空闲，Gnd=0/标记） 1：TX 引脚信号值取反。（VDD=0/标记，Gnd=1/空闲）。允许在 TX 线路上使用外部反相器。 只有在禁止 USART（UE=0）后才能写入此位字段。
位 16	**RXINV**：RX 引脚有效电平取反。此位由软件置 1 和清零。 0：RX 引脚信号使用标准逻辑电平工作（VDD=1/空闲，Gnd=0/标记） 1：RX 引脚信号值取反。（VDD=0/标记，Gnd=1/空闲）。允许在 RX 线路上使用外部反相器。 只有在禁止 USART（UE=0）后才能写入此位字段。
位 15	**SWAP**：交换 TX/RX 引脚。 此位由软件置 1 和清零。 0：按标准引脚排列定义使用 TX/RX 引脚 1：交换 TX 和 RX 引脚功能。有助于在与另一个 UART 的交叉连接时工作。 只有在禁止 USART（UE=0）后才能写入此位字段。
位 14	**LINEN**：LIN 模式使能。 此位由软件置 1 和清零。 0：LIN 模式禁用 1：LIN 模式启用 LIN 模式可以使用 USART_CR1 寄存器中的 SBKRQ 位发送 LIN 同步断路（13 个低位），并可检测 LIN 同步断路。 只有在禁止 USART（UE=0）后才能写入此位字段。 注意：如果 USART 不支持 LIN 模式，此位保留并由硬件强制清零。请参见第 20.4 节“USART 实现”。
位 13:12	**STOP[1:0]**：STOP 位。 这些位用于编程停止位。 00：1 个停止位 01：0.5 个停止位 10：2 个停止位 11：1.5 个停止位 只有在禁止 USART（UE=0）后才能写入此位字段。

位 11	**CLKEN**：时钟使能。 该位允许用户使能 SCLK 引脚。 0：禁止 SCLK 引脚 1：使能 SCLK 引脚 只有在禁止 USART（UE=0）时才能写入此位。 注意：如果既不支持同步模式，也不支持智能卡模式，则该位保留并由硬件强制清零。请参见第 20.4 节“USART 实现”。 注意：在智能卡模式下，为了正确地向智能卡提供 SCLK 时钟，必须遵循以下步骤： • UE=0 • SCEN=1 • GTPR 配置 • CLKEN=1 • UE=1
位 10	**CPOL**：时钟极性。 该位允许用户在同步模式下选择 SCLK 引脚上时钟输出的极性。它与 CPHA 位结合使用可获得所需的时钟/数据关系。 0：传输窗口外 SCLK 引脚上的稳定低值 1：传输窗口外 SCLK 引脚上的稳定高值 只有在禁止 USART（UE=0）时才能写入此位。 注意：如果不支持同步模式，该位保留并由硬件强制清零。 请参见第 20.4 节“USART 实现”。
位 9	**CPHA**：时钟相位。 此位用于在同步模式下选择 SCLK 引脚上时钟输出的相位。它与 CPOL 位结合使用可获得所需的时钟/数据关系（请参见图 119. 16 倍过采样时的数据采样和图 120. 8 倍过采样时的数据采样）。 0：从第一个时钟边沿开始采样数据 1：从第二个时钟边沿开始采样数据 只有在禁止 USART（UE=0）时才能写入此位。 注意：如果不支持同步模式，该位保留并由硬件强制清零。 请参见第 20.4 节“USART 实现”。
位 8	**LBCL**：最后一个位时钟脉冲。 此位用于在同步模式下选择与发送的最后一个数据位（MSB）关联的时钟脉冲是否必须在 SCLK 引脚上输出。 0：最后一个数据位的时钟脉冲不在 SCLK 引脚上输出 1：最后一个数据位的时钟脉冲在 SCLK 引脚上输出 **注意：**最后一位为发送的第 7 个、第 8 个或第 9 个数据位，具体取决于 USART_CR1 寄存器中 M 位所选择的 7 位、8 位或 9 位格式。 只有在禁止 USART（UE=0）时才能写入此位。 注意：如果不支持同步模式，该位保留并由硬件强制清零。 请参见第 20.4 节“USART 实现”。
位 7	保留，必须保持复位值。
位 6	LBDIE：LIN 断路检测中断使能。 断路中断屏蔽（使用断路分隔符进行断路检测）。 0：禁止中断 1：当 USART_ISR 寄存器中的 LBDF=1 时，生成中断 注意：如果 LIN 不支持智能卡模式，该位保留并由硬件强制清零。请参见第 20.4 节“USART 实现”。

位 5	**LBDL**：LIN 断路检测长度。 该位用于选择 11 位断路检测或 10 位断路检测。 0：10 位断路检测 1：11 位断路检测 只有在禁止 USART（UE=0）时才能写入此位。 注意：如果 LIN 不支持智能卡模式，该位保留并由硬件强制清零。请参见第 20.4 节“USART 实现”。
位 4	**ADDM7**：7 位地址检测/4 位地址检测。 此位用于选择 4 位地址检测或 7 位地址检测。 0：4 位地址检测。 1：7 位地址检测（在 8 位数据模式下） 只有在禁止 USART（UE=0）时才能写入此位。 注意：在 7 位和 9 位数据模式下，地址检测分别在 6 位和 8 位地址上完成（ADD[5:0]和 ADD[7:0]）。
位 3	**DIS_NSS** 当 DSI_NSS 位置 1 时，NSS 引脚输入被忽略。 0：SPI 从设备的选择取决于 NSS 输入引脚 1：总是选择 SPI 从设备并忽略 NSS 输入引脚 注意：当不支持 SPI 从模式时，此位保留且必须保持复位值。
位 2	保留，必须保持复位值。
位 1	保留，必须保持复位值。
位 0	**SLVEN**：同步从模式使能。 当 SLVEN 位置 1 时，同步从模式使能。 0：从模式禁用 1：从模式使能 注意：当不支持 SPI 从模式时，此位保留且必须保持复位值。

注意： *当发射器使能时，不应对 CPOL、CPHA 和 LBCL 位执行写操作。*

20.7.3 控制寄存器 3（USARTx_CR3）

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
TXFTCFG			RXFTI E.	RXFTCFG			TCBGT IE	TXFTIE	Res.	Res.		SCARCNT2:0]			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
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 31:29	**TXFTCFG**：TXFIFO 阈值配置。 000：TXFIFO 达到其深度的 1/8 001：TXFIFO 达到其深度的 1/4 010：TXFIFO 达到其深度的 1/2 011：TXFIFO 达到其深度的 3/4。

	100：TXFIFO 达到其深度的 7/8。 101：TXFIFO 变为空 剩余组合：保留。
Bit28	**RXFTIE**：RXFIFO 阈值中断使能。此位由软件置 1 和清零。 0：禁止中断 1：当接收 FIFO 达到在 RXFTCFG 中设定的阈值时，生成 USART 中断
位 27:25	**RXFTCFG**：接收 FIFO 阈值配置。 000：接收 FIFO 达到其深度的 1/8 001：接收 FIFO 达到其深度的 1/4 010：接收 FIFO 达到其深度的 1/2 011：接收 FIFO 达到其深度的 3/4 100：接收 FIFO 达到其深度的 7/8 101：接收 FIFO 已满 剩余组合：保留。
位 24	**TCBGTIE**：在保护时间之前发送完成，中断使能。 此位由软件置 1 和清零。 0：禁止中断 1：当 USARTx_ISR 寄存器中的 TCBGT=1 时，生成 USART 中断 注意：如果 USART 不支持智能卡模式，该位保留并由硬件强制清零。
位 23	**TXFTIE**：TXFIFO 阈值中断使能。 此位由软件置 1 和清零。 0：禁止中断 1：当 TXFIFO 达到在 TXFTCFG 中设定的阈值时，生成 USART 中断
位 22:20	保留，必须保持复位值。
位 19:17	**SCARCNT[2:0]**：智能卡自动重试计数。 此位字段用于指定智能卡模式下发送和接收的重试次数。在发送模式下，此位字段用于指定生成发送错误（FE 位置 1）前自动重新发送的重试次数。 在接收模式下，此位字段用于指定生成接收错误（RXNE/ RXFNE 位和 PE 位置 1）前错误接收尝试的次数。 只有在禁止 USART（UE=0）时才能编程此位字段。 使能 USART（UE=1）时，此位字段只能写入 0x0，以停止重新发送。 0x0：重新发送禁用 - 在发送模式下不自动重新发送 0x1 到 0x7：自动重新发送尝试次数（在发生信号发送错误之前） 注意：如果不支持智能卡模式，该位保留并由硬件强制清零。 请参见第 20.4 节“USART 实现”。
位 16	保留，必须保持复位值。
位 15	**DEP**：驱动器使能极性选择。 0：DE 信号为高电平有效 1：DE 信号为低电平有效 只有在禁止 USART（UE=0）时才能写入此位。 注意：如果不支持驱动器使能功能，该位保留并且必须保持清零。请参见第 20.4 节“USART 实现”。
位 14	**DEM**：驱动器使能模式。 此位用于通过 DE 信号激活外部收发器控制。

	0：DE 功能禁用 1：DE 功能使能。DE 信号在 RTS 引脚上输出。 只有在禁止 USART（UE=0）时才能写入此位。 注意：如果不支持驱动器使能功能，该位保留并且必须保持清零。第 20.4 节“USART 实现”。
位 13	**DDRE**：接收出错时 DMA 禁用。 0：接收出错时不禁止 DMA。相应的错误标志置 1，但 RXNE 保持为 0 以防止上溢。因此，将不使能 DMA 请求，从而不会传送错误数据（无 DMA 请求），但会传送接收到的下一个正确数据（用于智能卡模式）。 1：接收出错后禁止 DMA。相应的错误标志置 1，RXNE 也置 1。在错误标志清零之前，DMA 请求被屏蔽。这意味着软件必须首先禁用 DMA 请求（DMAR=0）或者将 RXNE（FIFO 模式使能时为 RXFNE）清零，然后再将错误标志清零。 只有在禁止 USART（UE=0）时才能写入此位。 注意：接收错误包括：奇偶校验错误、数据拆包错误或噪声错误。
位 12	**OVRDIS**：上溢禁用。 此位用于禁止接收上溢检测。 0：接收新数据前未读取已接收的数据时，上溢错误标志 ORE 置 1 1：上溢功能被禁用。如果在 RXNE 标志仍然置位时接收到新数据，ORE 将不置位且接收到的新数据会覆盖 USART_RDR 寄存器之前的内容。当 FIFO 模式使能时，RXFIFO 被旁路且数据被直接写入 USARTx_RDR 寄存器。即使使能了 FIFO 管理，仍将使用 RXNE 标志。 只有在禁止 USART（UE=0）时才能写入此位。 注意：此控制位用于在不读取数据的情况下检查通信流。
位 11	**ONEBIT**：一个采样位方法使能。 该位允许用户选择采样方法。选择一个采样位方法后，将禁止噪声检测标志（NF）。 0：三个采样位方法 1：一个采样位方法 只有在禁止 USART（UE=0）时才能写入此位。
位 10	**CTSIE**：CTS 中断使能。 0：禁止中断 1：当 USART_ISR 寄存器中的 CTSIF=1 时，生成中断 注意：如果不支持硬件流控制功能，该位保留并由硬件强制清零。请参见第 20.4 节“USART 实现”。
位 9	**CTSE**：使能 CTS。 0：禁止 CTS 硬件流控制 1：使能 CTS 模式，仅当 nCTS 输入有效（连接到 0）时才发送数据。如果在发送数据时使 nCTS 输入无效，会在停止之前完成发送。如果使 nCTS 无效时数据已写入数据寄存器，则将延迟发送，直到 nCTS 有效。 只有在禁止 USART（UE=0）时才能写入此位。 注意：如果不支持硬件流控制功能，该位保留并由硬件强制清零。请参见第 20.4 节“USART 实现”。
位 8	**RTSE**：使能 RTS。 0：禁止 RTS 硬件流控制 1：使能 RTS 输出，仅当接收缓冲区中有空间时才会请求数据。发送完当前字符后应停止发送数据。可以接收数据时使 nRTS 输出有效（拉至 0）。 只有在禁止 USART（UE=0）时才能写入此位。 注意：如果不支持硬件流控制功能，该位保留并由硬件强制清零。请参见第 20.4 节“USART 实现”。
位 7	**DMAT**：DMA 使能发射器。该位由软件置 1/复位。 1：针对发送使能 DMA 模式 0：针对发送禁止 DMA 模式
位 6	**DMAR**：DMA 使能接收器。该位由软件置 1/复位。

	1：针对接收使能 DMA 模式 0：针对接收禁止 DMA 模式
位 5	**SCEN**：智能卡模式使能。 该位用于使能智能卡模式。 0：禁止智能卡模式 1：智能卡模式使能 只有在禁止 USART（UE=0）后才能写入此位字段。 注意：如果 USART 不支持智能卡模式，该位保留并由硬件强制清零。 请参见 20.4 节“USART 实现”。
位 4	**NACK**：智能卡 NACK 使能。 0：出现奇偶校验错误时禁止 NACK 发送 1：出现奇偶校验错误时使能 NACK 发送 只有在禁止 USART（UE=0）后才能写入此位字段。 注意：如果 USART 不支持智能卡模式，该位保留并由硬件强制清零。请参见第 20.4 节“USART 实现”。
位 3	**HDSEL**：半双工选择。 单线半双工模式的选择。 0：未选择半双工模式 1：选择半双工模式 只有在禁止 USART（UE=0）时才能写入此位。
位 2	**IRLP**：IrDA 低功耗。 该位用于选择正常模式和低功耗 IrDA 模式。 0：正常模式 1：低功耗模式 只有在禁止 USART（UE=0）时才能写入此位。 注意：如果不支持 IrDA 模式，该位保留并由硬件强制清零。请参见第 20.4 节“USART 实现”。
位 1	**IREN**：IrDA 模式使能。 此位由软件置 1 和清零。 0：IrDA 禁用 1：IrDA 使能 只有在禁止 USART（UE=0）时才能写入此位。 注意：如果不支持 IrDA 模式，该位保留并由硬件强制清零。请参见第 20.4 节“USART 实现”。
位 0	**EIE**：错误中断使能。 如果发生数据拆包错误、上溢错误、噪声标志或 SPI 从设备下溢错误（USART_ISR 寄存器中 FE = 1 或 ORE = 1 或 NF = 1 或 UDR = 1），则需要使用错误中断使能位来使能中断生成。 0：禁止中断 1：当 USART_ISR 寄存器中的 FE = 1 或 ORE = 1 或 NF = 1 或 UDR = 1（在 SPI 从模式下）时，生成中断

20.7.4 波特率寄存器（USARTx_BRR）

只有在禁止 USART（UE=0）时才能写入此寄存器。在自动波特率检测模式下，该位由硬件自动更新。

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
BRR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16	保留，必须保持复位值。
位 15:4	**BRR[15:4]** BRR[15:4] = USARTDIV[15:4]
位 3:0	**BRR[3:0]** 当 OVER8 = 0 时，BRR[3:0] = USARTDIV[3:0]。 当 OVER8 = 1 时： BRR[2:0] = USARTDIV[3:0]，右移 1 位。 BRR[3] 必须保持清零。

20.7.5 保护时间和预分频器寄存器（USARTx_GTPR）

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
GT[7:0]								PSC[7:0]							
rw								rw							

位 31:16	保留，必须保持复位值。
位 15:8	**GT[7:0]**：保护时间值。 此位字段用于编程保护时间值（以波特时钟周期数为单位）。 该位用于智能卡模式。发送完成标志在该保护时间值之后置 1。 只有在禁止 USART（UE=0）后才能写入此位字段。 注意：如果不支持智能卡模式，该位保留并由硬件强制清零。 请参见第 20.4 节“USART 实现”。
位 7:0	**PSC[7:0]**：预分频器值。 **在 IrDA 低功耗和正常的 IrDA 模式下：** PSC[7:0] = IrDA 正常和低功耗波特率 用于编程预分频器，进行 USART 源时钟分频以获得低功耗频率： 使用寄存器中给出的值（8 个有效位）对源时钟进行分频： 00000000：保留 - 不编程此值 00000001：对源时钟进行 1 分频 00000010：对源时钟进行 2 分频 ...

	在智能卡模式下： PSC[4:0]：预分频器值 用于编程预分频器，进行 USART 源时钟分频以提供智能卡时钟。 将寄存器中给出的值（5 个有效位）乘以 2 得出源时钟频率的分频系数： 00000：保留 - 不编程此值 00001：对源时钟进行 2 分频 00010：对源时钟进行 4 分频 00011：对源时钟进行 6 分频 … 只有在禁止 USART（UE=0）后才能写入此位字段。 注意：如果使用智能卡模式，则位[7:5]必须保持清零。 不支持智能卡和 IrDA 模式时，该位字段保留并由硬件强制清零。 请参见第 20.4 节“USART 实现”。

20.7.6 接收器超时寄存器（USARTx_RTOR）

偏移地址：0x14

复位值：0x0000

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。 此位字段用于提供智能卡 T=1 接收下的块长度。其值等于信息字符数 + 结尾字段的长度（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） 在智能卡模式下，块长度计数器在 TXE=0（FIFO 模式使能时为 TXFE=0）时复位。 此位字段也可用于其它模式。这种情况下，块长度计数器在 RE=0（禁止接收器）和/或 EOBCF 位写入 1 时复位。 注意：块接收开始后可编程此值（使用起始字段中 LEN 字符中的数据）。每个接收到的块只能对此值编程一次。
位 23:0	**RTO[23:0]**：接收器超时值。 此位字段给出了接收器超时值（以波特时钟周期数计）。 在标准模式下，如果在接收到最后一个字符后，在 RTO 值对应的时间内未检测到新的起始位，则 RTOF 标志置 1。 在智能卡模式下，该值用于实现 CWT 和 BWT。参见智能卡相关章节了解更多信息。按照标准，从接收到的最后一个字符的起始位开始执行 CWT/BWT 测量。 注意：每个接收到的字符只能对此值编程一次。

注意： *可以实时写入 RTOR。如果新值小于或等于计数器的值，RTOF 标志置 1。如果不支持接收器超时功能，此寄存器保留并由硬件强制为“0x00000000”。请参见第 20.4 节“USART 实现”。*

20.7.7 请求寄存器（USARTx_RQR）

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	
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，必须保持复位值。
位 4	**TXFRQ**：发送数据刷新请求。 当 FIFO 模式禁用时，在此位写入 1 会将 TXE 标志置 1。 如此可丢弃发送数据。由于错误（NACK）而未发送数据且 USART_ISR 寄存器中的 FE 标志有效时，只能在智能卡模式下使用此位。如果 USART 不支持智能卡模式，该位保留并由硬件强制清零。 当 FIFO 使能时，将 TXFRQ 位置 1 可刷新整个 FIFO。这会使标志 TXFE 置 1（发送 FIFO 为空，USART_ISR 寄存器中的位 23）。UART 和智能卡模式均支持刷新发送 FIFO。 注意：在 FIFO 模式下，为确保无任何数据被写入数据寄存器，会在刷新请求期间复位 TXFNF 标志，直至 TxFIFO 为空。
位 3	**RXFRQ**：接收数据刷新请求。 将 1 写入此位将清空整个接收 FIFO，即 RXFNE 位清零。 如此可在不读取已接收数据的情况下丢弃这些数据，并避免上溢状态。
位 2	**MMRQ**：静默模式请求。 向此位写入 1 可将 USART 置于静默模式，并将 RWU 标志置 1。
位 1	**SBKRQ**：发送断路请求。 向此位写入 1 可将 SBKF 标志置 1 并在发送设备可用后立即请求在线路上发送 BREAK。 注意：这种情况下，应用需要在之前插入的所有数据（包括尚未发送的数据）后发送中断字符，软件应等到 TXE 标志使能后将 SBKRQ 位置 1。
位 0	**ABRRQ**：自动波特率请求。 向此位写入 1 可复位 USART_ISR 寄存器中的 ABRF 标志，并请求对下一个接收到的数据帧进行自动波特率测量。 注意：如果 USART 不支持自动波特率功能，则此位保留并由硬件强制清零。请参见第 20.4 节“USART 实现”。

20.7.8 中断和状态寄存器（USARTx _ISR）

偏移地址：0x1C

复位值：0x00C0（FIFO 禁用时）。

复位值：0x28000C0（FIFO/智能卡模式使能时）。

复位值：0x08000C0（FIFO 使能/智能卡模式禁用时）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TXFT	RXFT	TCBGT	RXFF	TXFE	RE ACK	TE ACK	Res.	RWU	SBKF	CMF	BUSY
				r	r	r	r	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	UDR	EOBF	RTOF	CTS	CTSIF	LBDF	TXE/TX FNF	TC	RXNE/RXFNE	IDLE	ORE	NF	FE	PE
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:28	保留，必须保持复位值。
位 27	**TXFT**：TXFIFO 阈值标志。 当 TXFIFO 达到 USARTx_CR3 寄存器的 TXFTCFG 位字段中的设定阈值（即 TXFIFO 包含 TXFTCFG 空位）时，此位由硬件置 1。如果 USART_CR3 寄存器中 TXFTIE 位 = 1（位 31），则会生成中断。 0：TXFIFO 未达到设定阈值 1：TXFIFO 达到设定阈值
位 26	**RXFT**：RXFIFO 阈值标志。 当达到 USARTx_CR3 寄存器的 RXFTCFG 位字段中的设定阈值时，此位由硬件置 1。这意味着接收 FIFO 中有（RXFTCFG - 1）个数据且 USART_RDR 寄存器中有一个数据。如果 USART_CR3 寄存器中 RXFTIE 位 = 1（位 27），则会生成中断。 0：接收 FIFO 未达到设定阈值 1：接收 FIFO 达到设定阈值 注意：当 RXFTCFG 阈值配置为 «101» 时，如果有 16 个数据可用（即 RXFIFO 中有 15 个数据且 USARTx_RDR 中有 1 个数据），则 RXFT 标志置 1。因此，接收到第 17 个数据不会导致上溢错误。上溢错误发生在接收到第 18 个数据之后。
位 25	**TCBGT**：在保护时间之前发送完成标志。 此位指示写入 USART_TDR 的最后一个数据何时正确地从移位寄存器发送出去。 在智能卡模式下，如果已完成对包含数据的帧的发送且没有收到来自智能卡的 NACK，则由硬件置 1。如果 USART_CR3 寄存器中 TCBGTIE=1，则会生成中断。通过向 USART_ICR 寄存器中的 TCBGTCF 写入 1 或向 USART_TDR 寄存器执行写操作，此位由软件清零。 0：发送未完成或发送未成功完成（即收到来自智能卡的 NACK） 1：发送成功完成（在保护时间结束之前完成且没有收到来自智能卡的 NACK）。 注意：如果 USART 不支持智能卡模式，该位保留并由硬件强制清零。如果 USART 支持智能卡模式且智能卡模式使能，则 TCBGT 复位值为“1”。
位 24	**RXFF**：RXFIFO 已满。 当 RXFIFO 已满时，该位由硬件置 1。 如果 USART_CR1 寄存器中 RXFFIE 位 = 1，则会生成中断。 0：RXFIFO 未满 1：RXFIFO 已满
位 23	**TXFE**：TXFIFO 为空。 当 TXFIFO 为空时，该位由硬件置 1。当 TXFIFO 包含至少一个数据时，此标志清零。也可以通过将 USART_RQR 寄存器中的 TXFRQ（位 4）位置 1 将 TXFE 标志位清零。 如果 USART_CR1 寄存器中 TXFEIE 位 = 1（位 30），则会生成中断。 0：TXFIFO 不为空 1：TXFIFO 为空
位 22	**REACK**：接收使能应答标志。 USART 采用接收使能值时，通过硬件将此位置 1/复位。 它可用于确认在进入停止模式前 USART 已做好接收准备。

	注意：如果 USART 不支持从停止模式唤醒特性，则此位保留并由硬件强制清零。
位 21	**TEACK**：发送使能确认标志。 USART 采用发送使能值时，通过硬件将此位置 1/复位。 通过写入 TE=0 生成空闲帧请求，然后在 USART_CR1 寄存器中写入 TE=1 以遵循 TE=0 最短周期时，可使用此位。
位 20	保留，必须保持复位值。
位 19	**RWU**：从静默模式唤醒接收器。 此位指示 USART 是否处于静默模式。当识别出唤醒/静默序列时，此位由硬件清零/置 1。通过 USART_CR1 寄存器中的 WAKE 位选择静默模式控制序列（地址或 IDLE）。 如果选择从 IDLE 模式唤醒，则此位只能由软件置位，即在 USART_RQR 寄存器中的 MMRQ 位写入 1。 0：接收器处于活动模式 1：接收器处于静音模式 注意：如果 USART 不支持从停止模式唤醒特性，则此位保留并由硬件强制清零。
位 18	**SBKF**：发送断路标志。 此位指示已请求发送断路字符。通过将 1 写入 USART_CR3 寄存器中的 SBKRQ 位，此位由软件置 1。此位在断路发送的停止位期间由硬件自动复位。 0：不发送断路字符 1：发送断路字符
位 17	**CMF**：字符匹配标志。 接收到由 ADD[7:0]定义的字符后由硬件将此位置 1。通过向 USART_ICR 寄存器中的 CMCF 写入 1，此位由软件清零。 如果 USART_CR1 寄存器中 CMIE=1，则会生成中断。 0：未检测到字符匹配 1：检测到字符匹配
位 16	**BUSY**：忙标志。 此位由硬件置 1 和复位。当 RX 线路上正在进行通信时，此位有效（成功检测到起始位）。在接收结束（成功或失败）时复位。 0：USART 空闲（无接收） 1：正在接收
位 15	**ABRF**：自动波特率标志。 已设置自动波特率（RXNE 也将置 1，并在 RXNEIE = 1 时生成中断），或者自动波特率操作未成功完成时，此位由硬件置 1（ABRE=1）（此时，ABRE、RXNE 和 FE 也置 1）。 为请求新的自动波特率检测，通过向 USART_RQR 寄存器中的 ABRRQ 写入 1，此位由软件清零。 注意：如果 USART 不支持自动波特率功能，则此位保留并由硬件强制清零。
位 14	**ABRE**：自动波特率错误。 如果波特率测量失败（波特率超出范围或字符比较失败），此位由硬件置 1。 通过将 1 写入 USART_CR3 寄存器中的 ABRRQ 位，此位由软件清零。 注意：如果 USART 不支持自动波特率功能，则此位保留并由硬件强制清零。
位 13	**UDR**：SPI 从设备下溢错误标志。 在从发送模式下，如果在软件尚未将任何值加载到 USARTx_DR 之前出现第一个数据发送时钟，此标志将置 1。 0：无下溢错误 1：下溢错误 注意：如果 USART 不支持 SPI 从模式，该位保留并由硬件强制清零。

位 12	**EOBF**：块结束标志。 接收到完整块后，此位由硬件置 1（例如 T=1 智能卡模式）。接收到的字节数（从块的起始处，包括起始字段）等于或大于 BLEN + 4 时执行检测。 如果 USART_CR2 寄存器中 EOBIE = 1，则会生成中断。 通过向 USART_ICR 寄存器中的 EOBCF 写入 1，此位由软件清零。 0：未达到块结束 1：达到块结束（字符数） 注意：如果不支持智能卡模式，该位保留并由硬件强制清零。 请参见第 20.4 节“USART 实现”。
位 11	**RTOF**：接收器超时。 已经过在 RTOR 寄存器中编程的超时值后，如果无任何通信，此位由硬件置 1。通过向 USART_ICR 寄存器中的 RTOCF 写入 1，此位由软件清零。 如果 USART_CR2 寄存器中 RTOIE=1，则会生成中断。 在智能卡模式下，该超时对应于 CWT 或 BWT 时间。 0：未达到超时值 1：已达到超时值，未接收到任何数据 注意：如果 RTOR 寄存器中编程的时间值将 2 个字符隔开，则 RTOF 不置 1。如果此时间大于该值 + 2 个采样时间（2/16 或 2/8，具体取决于过采样方法），则 RTOF 标志置 1。 即使 RE = 0，计数器仍会计数，但 RTOF 仅在 RE = 1 时置 1。当 RE 置 1 时，如果超时已结束，则 RTOF 置 1。 如果 USART 不支持接收器超时功能，该位保留并由硬件强制清零。
位 10	**CTS**：CTS 标志。 此位由硬件置 1/复位。此位是对 nCTS 输入引脚的状态取反。 0：nCTS 线置 1 1：nCTS 线复位 注意：如果不支持硬件流控制功能，该位保留并由硬件强制清零。
位 9	**CTSIF**：CTS 中断标志。 如果 CTSE 位置 1，当 nCTS 输入切换时，此位由硬件置 1。通过向 USART_ICR 寄存器中的 CTSCF 位写入 1，此位由软件清零。 如果 USART_CR3 寄存器中 CTSIE=1，则会生成中断。 0：nCTS 状态线上未发生变化 1：nCTS 状态线上发生变化 注意：如果不支持硬件流控制功能，该位保留并由硬件强制清零。
位 8	**LBDF**：LIN 断路检测标志。 检测到 LIN 断路时，该位由硬件置 1。通过向 USART_ICR 寄存器中的 LBDCF 写入 1，此位由软件清零。 如果 USART_CR2 寄存器中 LBDIE=1，则会生成中断。 0：未检测到 LIN 断路 1：检测到 LIN 断路 注意：如果 USART 不支持 LIN 模式，该位保留并由硬件强制清零。请参见第 20.4 节“USART 实现”。
位 7	**TXE/TXFNF**：发送数据寄存器为空/TXFIFO 未满。 当 FIFO 模式禁用时，如果 USARTx_TDR 寄存器的内容已传输到移位寄存器，则 TXE 由硬件置 1。通过对 USARTx_TDR 寄存器执行写入操作将该位清零。为丢弃数据（仅在智能卡 T=0 模式下出现发送故障时），也可以通过向 USART_RQR 寄存器中的 TXFRQ 写入 1 来将 TXE 标志位置 1。 当 FIFO 模式使能时，如果 TXFIFO 未满因而可将数据写入 USART_TDR，则 TXFNF 由硬件置 1。对 USART_TDR 的每次写操作都会将数据放入 TXFIFO。在 TXFIFO 装满之前，此标志保持置位。当 TXFIFO 已

	满时，此标志清零，表示无法将数据写入 USART_TDR。 注意：在刷新请求期间，TXFNF 保持复位，直至 TXFIFO 为空。在发送刷新请求（通过将 TXFRQ 位置 1）后，应先检查标志 TXFNF，然后再写入 TXFIFO。（TXFNF 和 TXFE 同时置 1）。 如果 USART_CR1 寄存器中 TXEIE/TXFNFIE 位 = 1，则会生成中断。 0：数据寄存器已满/发送 FIFO 已满 1：数据寄存器/发送 FIFO 未满 注意：单缓冲区发送期间使用该位。
位 6	**TC**：发送完成 此位指示写入 USART_TDR 的最后一个数据何时从移位寄存器发送出去。 如果已完成对包含数据的帧的发送并且 TXE/TXFE 置 1，则由硬件置 1。如果 USART_CR1 寄存器中 TCIE=1，则会生成中断。通过向 USART_ICR 寄存器中的 TCCF 写入 1 或向 USART_TDR 寄存器执行写操作，此位由软件清零。 如果 USART_CR1 寄存器中 TCIE=1，则会生成中断。 0：传送未完成 1：传送已完成 注意：如果 TE 位复位且没有正在进行的发送，则 TC 位立即置 1。
位 5	**RXNE/RXFNE**：读取数据寄存器不为空/RXFIFO 不为空。 如果 USARTx_RDR 移位寄存器的内容已传输到 USARTx_RDR 寄存器，则 RXNE 位由硬件置 1。通过对 USARTx_RDR 寄存器执行读入操作将该位清零。也可以通过将 USARTx_RQR 寄存器中的 RXFRQ 位置 1 将 RXNE 标志位清零。如果 RXFIFO 不为空因而可从 USART_RDR 寄存器读取数据，则 RXFNE 位由硬件置 1。每次读取 USART_RDR 都将释放 RXFIFO 中的一个位置。当 RXFIFO 为空时，此位清零。 也可以通过将 USARTx_RQR 寄存器中的 RXFRQ 位置 1 将 RXNE/RXFNE 标志位清零。 如果 USART_CR1 寄存器中 RXNEIE/RXFNEIE=1，则会生成中断。 0：未接收到数据 1：已准备好读取接收到的数据
位 4	**IDLE**：检测到空闲线路。 检测到空闲线路时，该位由硬件置 1。如果 USART_CR1 寄存器中 IDLEIE=1，则会生成中断。通过向 USART_ICR 寄存器中的 IDLECF 写入 1，此位由软件清零。 0：未检测到空闲线路 1：检测到空闲线路 注意：直到 RXNE 位已置 1 时（即，当出现新的空闲线路时）IDLE 位才会被再次置 1。 使能静默模式（MME=1）后，如果 USART 未静默（RWU=0），则 IDLE 置 1，无论是否通过 WAKE 位选择了静默模式。如果 RWU=1，IDLE 不置 1。
位 3	**ORE**：上溢错误。 当移位寄存器中目前正在接收的数据准备就绪可以传输到 USARTx_RDR 寄存器且 RXNE=1（FIFO 模式使能时为 RXFF=1）时，此位由硬件置 1。通过向 USARTx_ICR 寄存器中的 ORECF 写入 1，此位由软件清零。 如果 USARTx_CR1 寄存器中 RXNEIE/ RXFNEIE=1 或 EIE=1，则生成中断。 0：无上溢错误 1：检测到上溢错误 注意：当此位置 1 时，USART_RDR 寄存器的内容不会丢失，但移位寄存器会被覆盖。当 EIE 位置 1 时，如果在多缓冲区通信期间 ORE 标志置 1，则生成中断。 当 USART_CR3 寄存器中的 OVRDIS 位置 1 时，此位永久强制为 0（无上溢检测）。
位 2	**NF**：START 位噪声检测标志。 当在接收的帧上检测到噪声时，该位由硬件置 1。通过向 USART_ICR 寄存器中的 NFCF 写入 1，此位由软件清零。

	0：未检测到噪声 1：检测到噪声 注意：该位不会生成中断，因为该位出现的时间与本身生成中断的 RXNE/RXFNE 位出现的时间相同。当 EIE 位置 1 时，如果在多缓冲区通信期间 NF 标志置 1，则生成中断。 注意：当线路无噪声时，可通过将 ONEBIT 位编程为 1 禁用 NF 标志，以便增大 USART 对偏差的容差（参见第 20.5.6 节“USART 接收器对时钟偏差的容差”）。 注意：在 FIFO 模式下，此错误与 USARTx_RDR 中的字符有关。
位 1	**FE**：数据拆包错误。 当检测到去同步化、过度的噪声或中断字符时，该位由硬件置 1。通过向 USART_ICR 寄存器中的 FECF 写入 1，此位由软件清零。在智能卡模式下发送期间，如果在达到最大发送尝试次数后仍未成功（智能卡向数据帧发送 NACK 信号），则此位置 1。 如果 USART_CR1 寄存器中 EIE=1，则会生成中断。 0：未检测到数据拆包错误 1：检测到数据拆包错误或中断字符 注意：在 FIFO 模式下，此错误与 USARTx_RDR 中的字符有关。
位 0	**PE**：奇偶校验错误。 当在接收器模式下发生奇偶校验错误时，该位由硬件置 1。 通过向 USART_ICR 寄存器中的 PECF 写入 1，此位由软件清零。 如果 USART_CR1 寄存器中 PEIE=1，则会生成中断。 0：无奇偶校验错误 1：奇偶校验错误

注意： *在 FIFO 模式下，此错误与 USARTx_RDR 中的字符有关。*

20.7.9 中断标志清零寄存器（USART_ICR）

偏移地址：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.	Res.	Res.	Res.	CMCF	Res.
														w	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	UDRCF	EOBCF	RTOCF	Res.	CTSCF	LBDCF	TCBGT CF	TCCF	TXFEC F	IDLECF	ORECF	NECF	FECF	PECF
		w	w	w		w	w	w	w	w	w	w	w	w	w

位 31:20	保留，必须保持复位值。
位 19:18	保留，必须保持复位值。
位 17	**CMCF**：字符匹配清零标志。 将 1 写入此位时，USART_ISR 寄存器中 CMF 标志将清零。
位 16:14	保留，必须保持复位值。
位 13	**UDRCF**：SPI 从设备下溢清零标志。 将 1 写入此位时，USART_ISR 寄存器中 UDRF 标志将清零。 注意：如果 USART 不支持 SPI 从模式，该位保留并由硬件强制清零。请参见第 20.4 节“USART 实现”。
位 12	**EOBCF**：块结束清零标志。 将 1 写入此位时，USART_ISR 寄存器中 EOBF 标志将清零。

	注意：如果 USART 不支持智能卡模式，该位保留并由硬件强制清零。请参见第 20.4 节“USART 实现”。
位 11	**RTOCF**：接收器超时清零标志。 将 1 写入此位时，USART_ISR 寄存器中 RTOF 标志将清零。 注意：如果 USART 不支持接收器超时功能，该位保留并由硬件强制清零。请参见第 20.4 节“USART 实现”。
位 10	保留，必须保持复位值。
位 9	**CTSCF**：CTS 清零标志。 将 1 写入此位时，USART_ISR 寄存器中 CTSIF 标志将清零。 注意：如果不支持硬件流控制功能，该位保留并由硬件强制清零。请参见第 20.4 节“USART 实现”。
位 8	**LBDCF**：LIN 断路检测清零标志。 将 1 写入此位时，USART_ISR 寄存器中 LBDF 标志将清零。 注意：如果 LIN 不支持智能卡模式，该位保留并由硬件强制清零。请参见第 20.4 节“USART 实现”。
位 7	**TCBGTCF**：保护时间之前发送完成清零标志将 1 写入此位会将 USART_ISR 寄存器中的 TCBGT 标志清零。
位 6	**TCCF**：发送完成清零标志将 1 写入此位会将 USART_ISR 寄存器中的 TC 标志清零。
位 5	**TXFECF**：TXFIFO 为空清零标志。 将 1 写入此位时，USART_ISR 寄存器中 TXFE 标志将清零。
位 4	**IDLECF**：检测到空闲线路清零标志。 将 1 写入此位时，USART_ISR 寄存器中 IDLE 标志将清零。
位 3	**ORECF**：上溢错误清零标志。 将 1 写入此位时，USART_ISR 寄存器中 ORE 标志将清零。
位 2	**NECF**：检测到噪声清零标志。 将 1 写入此位时，USART_ISR 寄存器中 NF 标志将清零。
位 1	**FECF**：数据拆包错误清零标志。 将 1 写入此位时，USART_ISR 寄存器中 FE 标志将清零。
位 0	**PECF**：奇偶校验错误清零标志。 将 1 写入此位时，USART_ISR 寄存器中 PE 标志将清零。

20.7.10 接收数据寄存器（USART_RDR）

偏移地址：0x24

复位值：未定义

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，必须保持复位值。
位 8:0	**RDR[8:0]**：接收数据值。包含接收到的数据字符。 RDR 寄存器在输入移位寄存器和内部总线之间提供了并行接口（参见图 113. USART 框图）。 在使能奇偶校验位的情况下进行接收时，从 MSB 位中读取的值为接收到的奇偶校验位。

20.7.11 发送数据寄存器（USART_TDR）

偏移地址：0x28

复位值：未定义

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，必须保持复位值。
位 8:0	**TDR[8:0]**：发送数据值。 包含要发送的数据字符。 USARTx_TDR 寄存器在内部总线和输出移位寄存器之间提供了并行接口（参见图 113. USART 框图）。 在使能奇偶校验位的情况下（USART_CR1 寄存器中的 PCE 位被置 1）进行发送时，由于 MSB 的写入值（位 7 或位 8，具体取决于数据长度）会被奇偶校验位所取代，因此该值不起任何作用。 注意：该寄存器只能在 TXE/TXFNF=1 时写入。

20.7.12 预分频器寄存器（USARTx_PRESC）

只有在禁止 USART（UE=0）时才能写入此寄存器。

偏移地址：0x2C

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.	Res.	Res.	Res.	Res.	Res.	PRESCALER[3:0]			
												rw	rw	rw	rw

位 31:4	保留，必须保持复位值。
位 3:0	**PRESCALER[3:0]**：时钟预分频器。 USART 输入时钟可通过预分频器进行分频： 0000：输入时钟不分频 0001：输入时钟 2 分频 0010：输入时钟 4 分频 0011：输入时钟 6 分频 0100：输入时钟 8 分频 0101：输入时钟 10 分频 0110：输入时钟 12 分频 0111：输入时钟 16 分频

	1000：输入时钟 32 分频 1001：输入时钟 64 分频 1010：输入时钟 128 分频 1011：输入时钟 256 分频 剩余组合：保留。 注意：在用允许值以外的值对 PRESCALER 进行编程时，预分频因子的编程值为 «1011»，即输入时钟的 256 分频。

20.8 USART 寄存器映射

表 72. USART 寄存器映射

偏移	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
0x00	USART_CR1	RXFFIE	RXFEIE	FIFOEN	M1	EOBIE	RTOIE	DEAT[4:0]					DEDT[4:0]					OVER8	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	Res.	UE
	复位值	0	0	0	0	0	0	0	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	ABRMOD[1:0]		ABREN	MSBFIRST	DATAINV	TXINV	RXINV	SWAP	LINEN	STOP [1:0]		CLKEN	CPOL	CPHA	LBCL	Res.	LBDIE	LBDL	ADDM7	DIS_NSS	Res.	Res.	SLVEN
	复位值	0	0	0	0	0	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	TXFTCFG[2:0]			RXFTIE	RXFTCFG[2:0]			TCBGTIE	TXFTIE	Res.	Res.	Res.	SCAR CNT2:0]			Res.	DEP	DEM	DDRE	OVRDIS	ONEBIT	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSEL	IRLP	IREN	EIE
	复位值	0	0	0	0	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	USART_BRR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BRR[15:0]															
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

偏移	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
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]							
	复位值																	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]																							
	复位值	0	0	0	0	0	0	0	0	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
	复位值																												0	0	0	0	0
0x1C	USART_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REACK	TEACK	Res.	RWU	SBKF	CMF	BUSY	ABRF	ABRE	Res.	EOBF	RTOF	CTS	CTSIF	LBDF	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE
	复位值										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.	Res.	Res.	Res.	CMCF	Res.	Res.	Res.	Res.	EOBCF	RTOCF	Res.	CTSCF	LBDCF	Res.	TCCF	Res.	IDLECF	ORECF	NECF	FECF	PECF
	复位值															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]								
	复位值																								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]								
	复位值																								0	0	0	0	0	0	0	0	0

偏移	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
0x2C	USART_PRES C	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.	PRESCALER[3:0]			
	复位值																													0	0	0	0

请参见“表 3. BlueNRG-LP 存储器映射和外设寄存器边界地址”，了解寄存器边界地址。

21 通用异步收发器（LPUART）

21.1 LPUART 简介

通用异步收发器（LPUART）是可以进行双向 UART 通信的 UART。

注意： *在 BlueNRG-LP 中，LPUART 内核时钟是固定的 16 MHz 时钟。*

LPUART 包含所有必要的硬件支持，使在最小功耗下可以进行异步串行通信。

它支持半双工单线通信和调制解调器操作（CTS/RTS）。它还支持多处理器通信。DMA（直接存储器访问）可用于数据发送/接收。

21.2 LPUART 主要特性

- 全双工异步通信
- NRZ 标准格式（标记/空格）
- 可编程波特率
- 两个内部 FIFO 用于发送和接收数据
- 每个 FIFO 均可通过软件使能/禁用并附带 FIFO 状态的状态标志
- 具有专用内核时钟的双时钟域可使波特率编程不受 PCLK 重新编程的影响
- 数据字长度可编程（7 位、8 位或 9 位）
- 可编程的数据顺序，最先移位 MSB 或 LSB
- 停止位可配置（支持 1 个或 2 个停止位）
- 单线半双工通信
- 使用 DMA 进行连续通信
- 使用中央 DMA 在预留的 SRAM 缓冲区中收/发字节
- 发射器和接收器有单独的使能位
- 发送和接收的单独信号极性控制
- Tx/Rx 引脚配置可交换
- 调制解调器和 RS-485 收发器的硬件流控制
- 传输检测标志：
  - 接收缓冲区已满
  - 发送缓冲空
  - 忙碌和发送结束标志
- 奇偶校验控制：
  - 发送奇偶校验位
  - 检查接收的数据字节的奇偶性
- 四个错误检测标志：
  - 上溢错误
  - 噪声检测
  - 帧错误
  - 奇偶校验错误
- 中断源标志位
- 多处理器通信
- 如果地址不匹配，LPUART 将进入静默模式
- 从静默模式唤醒（通过空闲线检测或地址标记检测）。

21.3 LPUART 功能说明

任何 LPUART 双向通信均需要至少两个引脚：接收数据输入引脚（RX）和发送数据引脚输出（TX）：

- **RX**：接收数据输入。串行数据输入引脚
- **TX**：发送数据输出引脚

如果关闭发射器，该输出引脚模式由其 I/O 端口配置决定。如果使能了发射器但没有待发送的数据，则 TX 引脚处于高电平。在单线模式下，该 I/O 用于发送和接收数据。

在正常 LPUART 模式下，串行数据通过这些引脚以帧的形式发送和接收，帧包含：

- 发送或接收前保持空闲线路
- 起始位
- 数据字（字长 7 位、8 位或 9 位），最低有效位在前
- 用于指示帧传输已完成的 1 个、2 个停止位
- LPUART 接口使用波特率发生器
- 状态寄存器（LPUART_ISR）
- 接收和发送数据寄存器（LPUART_RDR,LPUART_TDR）
- 波特率寄存器（LPUART_BRR）。

请参见第 21.5 节“LPUART 寄存器”了解每一位的定义。在 RS232 硬件流控制模式下需要以下引脚：

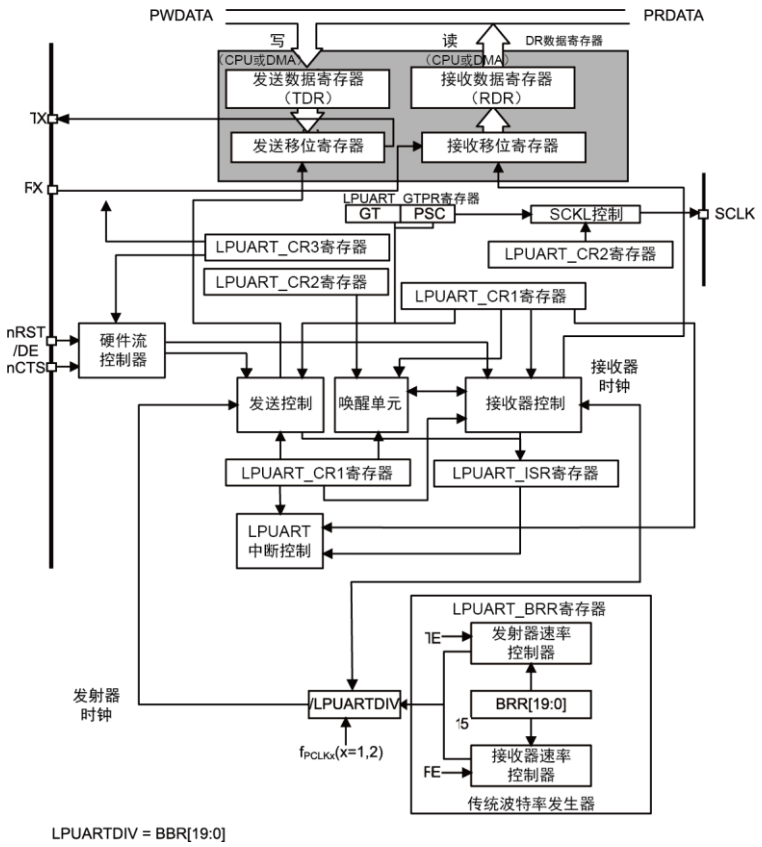
- **nCTS**：“清除以发送”用于在当前传输结束时阻止数据发送（高电平时）
- **nRTS**：“请求以发送”用于指示 LPUART 已准备好接收数据（低电平时）。

在 RS485 硬件控制模式下需要以下引脚：

- **DE**：“驱动器使能”用于激活外部收发器的发送模式。

注意： ***DE 和 nRTS 共用同一个引脚。***

图 138. LPUART 框图



21.3.1 LPUART 字符说明

通过在 LPUART_CR1 寄存器中设定 M 位（M0：位 12 和 M1：位 28），字长可选择为 7 或 8 或 9 位。（参见图 114. 字长编程）。

- 7 位字符长度：M[1:0] =10
- 8 位字符长度：M[1:0] =00
- 9 位字符长度：M[1:0] =01

在默认配置下，信号（TX 或 RX）在起始位工作期间处于低电平状态。在停止位工作期间处于高电平状态。

通过极性配置控制，可以单独针对每个信号对这些值取反。

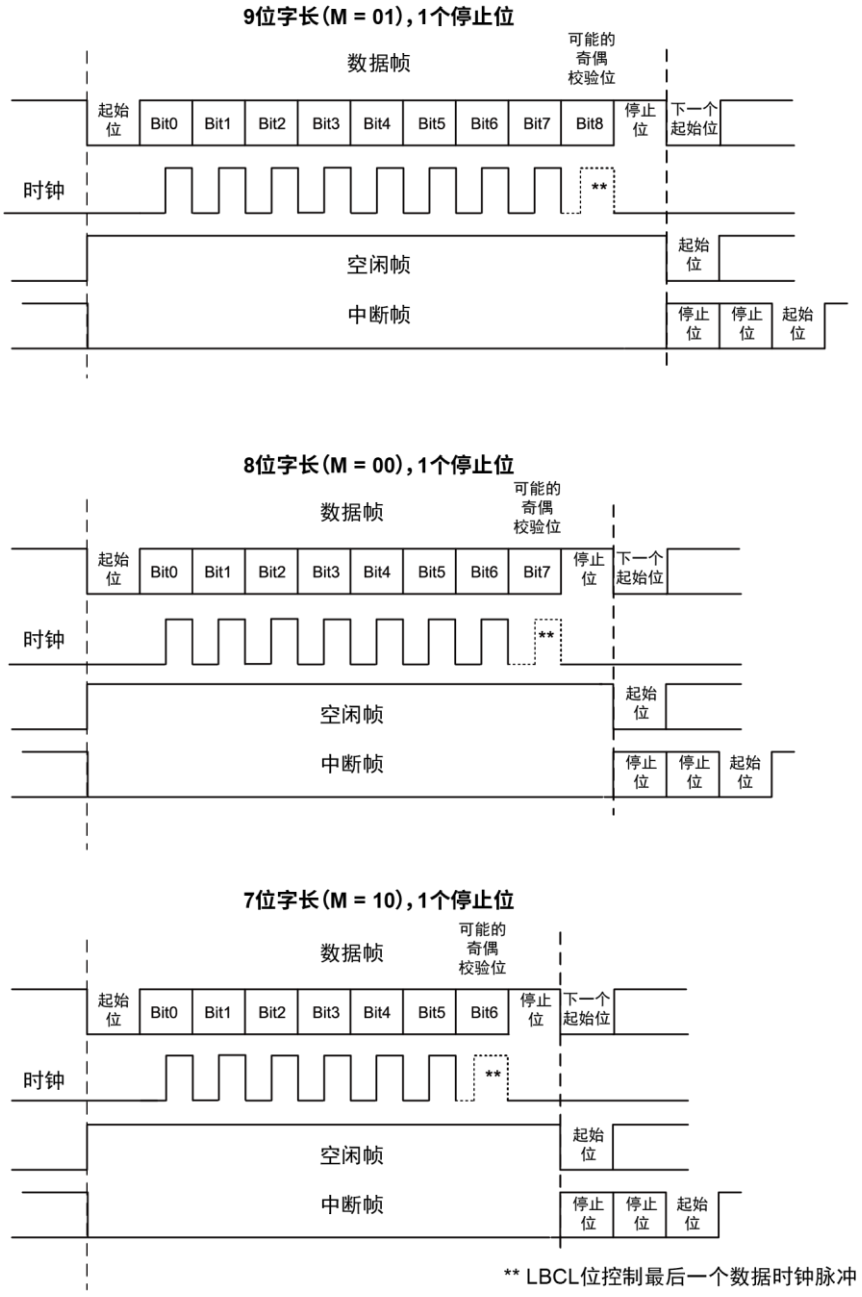
空闲字符可理解为一整个帧的“1”（“1”的数量包括停止位的数量）。

停止字符可理解为在一个帧周期内接收到的电平均为“0”。发射器在中断帧的末尾插入 2 个停止位。

发送和接收由通用波特率发生器驱动，发射器和接收器的使能位分别置 1 时将生成相应的发送时钟和接收时钟。

下面给出了各个块的详细信息。

图 139. LPUART 字长编程



21.3.2 FIFO 和阈值

LPUART 可在 FIFO 模式下工作，FIFO 缓冲区的深度为 8 字节。LPUART 具有发送 FIFO（TXFIFO）和接收 FIFO（RXFIFO）。通过将 USARTx_CR1 寄存器中的位 29 FIFOEN 置 1 来使能 FIFO 模式。

最大数据字长为 9 位，TXFIFO 的宽度为 9 位。但是，RXFIFO 的默认宽度为 12 位。这是因为接收器不仅将数据，还将与每个字符相关的错误标志（奇偶校验错误、噪声错误和数据拆包错误标志）放入 FIFO。

注意： *接收的数据及其标志被存储在 RXFIFO 中。但是，在读取 RDR 时，只读取数据。状态标志位于 LPUART_ISR 寄存器中。*

可以定义触发 Tx 和 RX 中断的 TXFIFO 和 RXFIFO 级别。这些阈值通过 LPUART_CR3 控制寄存器中的位字段 RXFTCFG 和 TXFTCFG 进行编程。

在这种情况下：

- 当 RXFIFO 中接收到的数据量达到在 RXFTCFG 位字段中编程的阈值时，生成接收中断。
- 当 TXFIFO 中空位的数量达到在 TXFTCFG 位字段中编程的阈值时，生成发送中断。

RXFIFO 阈值

使用 LPUART_CR3 寄存器中的 RXFTCFG 位字段配置 RXFIFO 阈值。

当接收的数据量等于编程的 RXFTCFG 时，LPUART_ISR 寄存器中的标志 RXFT 置位。

RXFT 标志置位意味着接收到了 RXFTCFG 个数据：LPUART_RDR 中 1 个数据和 RXFIFO 中（RXFTCFG - 1）个数据。因此，当 RXFTCFG 被编程为 «101» 时，RXFT 标志会在接收到 8 个数据时置位：RXFIFO 中 7 个数据和 LPUARTx_RDR 中 1 个数据。因此，接收到第 9 个数据不会使上溢标志置位。

21.3.3 发射器

发射器可发送 7 位、8 位或 9 位的数据字，具体取决于 M 位的状态。要激活发射器功能，必须将发送使能位（TE）置 1。发送移位寄存器中的数据在 TX 引脚输出。

字符发送

LPUART 发送期间，首先通过 TX 引脚移出数据的最低有效位（默认配置）。该模式下，LPUART_TDR 寄存器的缓冲区（TDR）位于内部总线和发送移位寄存器之间（见图 106. SMBus 从发射器的传输序列流程图（N 字节 + PEC））。

当 FIFO 模式使能时，写入 LPUART_TDR 寄存器的数据在 TXFIFO 中排成队列。

每个字符前面都有一个起始位，其逻辑电平在一个位周期内为低电平。字符由可配置数量的停止位终止。

LPUART 支持以下停止位：1 或 2 个停止位。

注意： *向 LPUART_TDR 中写入要发送的数据前，TE 位必须先置 1。*

注意： *数据发送期间不应复位 TE 位。如果在发送期间复位 TE 位，由于波特率计数器发生冻结，会导致 TX 引脚上的数据损坏。当前正在发送的数据将丢失。在 TE 位使能后发送空闲帧。*

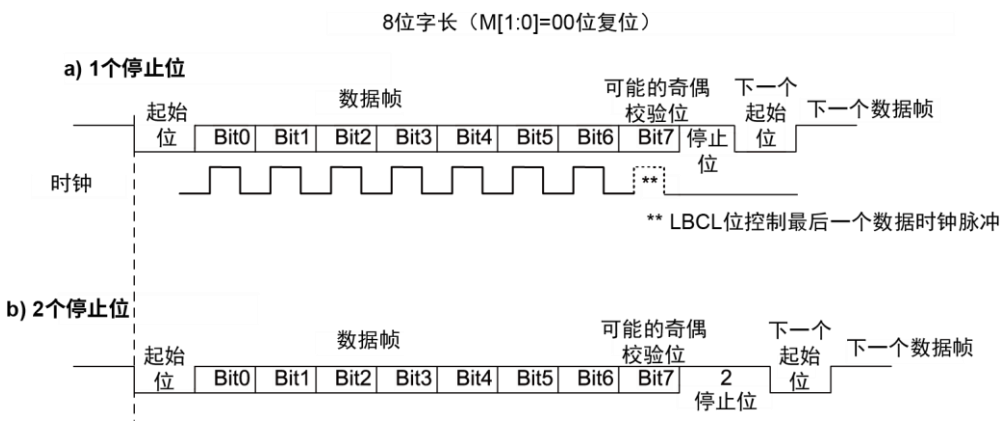
可配置的停止位

可以在控制寄存器 2 的位 13 和位 12 中编程将随各个字符发送的停止位的数量。

- **1 个停止位**：这是停止位数量的默认值。
- **2 个停止位**：正常 LPUART、单线和调制解调器模式支持 2 个停止位。空闲帧发送包含停止位。

中断帧发送是 10 个低位（当 M[1:0] = 00）或 11 个低位（当 M[1:0] = 01）或 9 个低位（当 M[1:0] = 10）后接 2 个停止位。无法传送长中断（中断长度大于 9/10/11 个低电平位）。

图 140. 可配置的停止位



字符发送步骤

1. 对 LPUART_CR1 中的 M 位进行编程以定义字长。
2. 使用 LPUART_BRR 寄存器选择所需波特率。
3. 对 LPUART_CR2 中的停止位数量进行编程。
4. 通过向 LPUART_CR1 寄存器中的 UE 位写入 1 使能 LPUART。
5. 如果将进行多缓冲区通信，请选择 LPUART_CR3 中的 DMA 使能（DMAT）。按照第 21.3.6 节“多处理器通信”中的说明配置 DMA 寄存器。
6. 置位 LPUART_CR1 中的 TE 位以便在首次发送时发送一个空闲帧。
7. 在 LPUART_TDR 寄存器中写入要发送的数据。为每个要在单缓冲区模式下发送的数据重复这一步骤。
   - 当 FIFO 模式禁用时，在 LPUART_TDR 中写入数据会将 TXE 标志清零。
   - 当 FIFO 模式使能时，在 LPUART_TDR 中写入数据会将一个数据添加到 TXFIFO，并且 LPUART_TDR 中的写操作会在 TXFNF 标志置位时执行。在 TXFIFO 装满之前，此标志保持置位。
8. 向 LPUART_TDR 寄存器写入最后一个数据后，等待至 TC=1。这表明最后一个帧的传送已完成。禁止 LPUART 或进入暂停模式时需要此步骤，以避免损坏最后一次发送。
   - 当 FIFO 模式禁用时，这表明最后一个帧的发送已完成。
   - 当 FIFO 模式使能时，这表明 TXFIFO 和移位寄存器均为空。

单字节通信

- 当 FIFO 模式禁用时：

对发送数据寄存器的写操作总是将 TXE 位清零。TXE 标志由硬件置 1，表示：

- 数据已从 LPUART_TDR 寄存器移到移位寄存器中且数据发送已开始。
- LPUART_TDR 寄存器为空。
- LPUART_TDR 寄存器中可写入下一个数据，而不会覆盖前一个数据。

TXEIE 位置 1 时该标志位会生成中断。

发送时，要传入 LPUART_TDR 寄存器的写指令中存有 TDR 寄存器中的数据，该数据将在当前发送结束时复制到移位寄存器中。

未发送时，要传入 LPUART_TDR 寄存器的写指令将数据置于移位寄存器中，数据发送开始时，TXE 位置 1。

- 当 FIFO 模式使能时，TXFNF（TXFIFO 未满）标志由硬件置位，这表明：
  - TXFIFO 未满。
  - LPUART_TDR 寄存器为空。
  - LPUART_TDR 寄存器中可写入下一个数据，而不会覆盖前一个数据。发送时，对 LPUART_TDR 寄存器的写操作将数据存储在 TXFIFO 中。在当前发送结束时，会将数据从 TXFIFO 复制到移位寄存器。

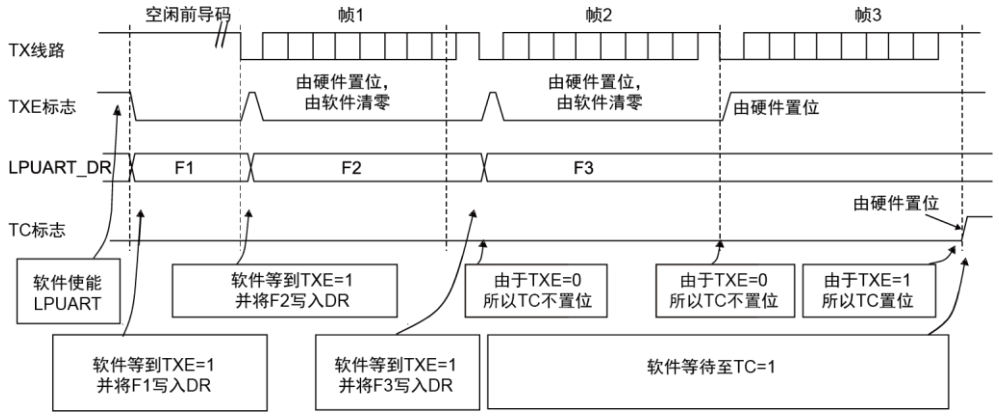
当 TXFIFO 未满时，即使在写入 LPUART_TDR 后，TXFNF 标志仍保持为 1。当 TXFIFO 已满时，该标志清零。TXFNEIE 位置 1 时该标志位会生成中断。

或者，当达到 TXFIFO 阈值时，可生成中断，并能将数据写入 TXFIFO。在这种情况下，CPU 可写入由设定的阈值定义的数据块。

如果发送了帧（在停止位之后）且 TXE 标志（FIFO 模式下为 TXFE）置位，则 TC 位置位。如果 LPUART_CR1 寄存器中的 TCIE 位置 1，则生成中断。

在 LPUART_TDR 寄存器中写入最后一个数据后，必须等到 TC=1 才能禁用 LPUART，否则会导致微控制器进入低功耗模式（参见图 141. 发送时的 TC/TXE 行为）。

图 141. 发送时的 TC/TXE 行为）



注意： *当 FIFO 管理使能时，将 TXFNF 标志用于数据发送。*

中断字符

置位 SBKRQ 位将发送一个中断字符。中断帧长度取决于 M 位（参见图 109. SMBus 从接收器的总线传输图（SBC=1））。

如果将“1”写入 SBKRQ 位，则当前字符发送完成后，将在 TX 线路上发送一个中断字符。该写操作将 SBKF 位置 1 并在中断字符发送完成时（发送中断字符后的停止位期间）由硬件复位。LPUART 在中断帧末尾的两位持续时间内插入一个逻辑“1”信号（STOP），以确保识别下个帧的起始位。

当 SBKRQ 位置 1 时，在当前发送结束时发送断路字符。

当 FIFO 模式使能时，发送断路字符优先于发送数据，即使 TXFIFO 已满。

空闲字符

置位 TE 位会驱动 LPUART 在第一个数据帧之前发送一个空闲帧。

21.3.4 接收器

LPUART 可接收 7、 8 位或 9 位的数据字，具体取决于 LPUART_CR1 寄存器中的 M 位。

起始位检测

在 LPUART 中，为了检测 START 位，应首先在 Rx 线路上检测下降沿，然后在起始位的中间采样以确认它仍然为“0”。如果起始采样为“1”，则噪声错误标志（NF）置 1，然后 START 位会被丢弃，接收器等待新的 START 位。否则，接收器继续对所有传入位进行正常采样。

字符接收

LPUART 接收期间，首先通过 RX 引脚移入数据的最低有效位（默认配置）。该模式下，LPUART_RDR 寄存器的缓冲区（RDR）位于内部总线和接收移位寄存器之间。

字符接收步骤

1. 对 LPUART_CR1 中的 M 位进行编程以定义字长。
2. 使用波特率寄存器 LPUART_BRR 选择所需波特率。
3. 对 LPUART_CR2 中的停止位数量进行编程。
4. 通过向 LPUART_CR1 寄存器中的 UE 位写入 1 使能 LPUART。
5. 如果将进行多缓冲区通信，请选择 LPUART_CR3 中的 DMA 使能（DMAR）。按照第 21.3.6 节“多处理器通信”中的说明配置 DMA 寄存器。
6. 置位 RE 位 LPUART_CR1。这一操作将使能接收器开始搜索起始位。

接收到字符时：

- 当 FIFO 模式禁用时，RXNE 位置 1。它表示移位寄存器的内容被传输到 RDR。也就是说，已接收到并可读取数据（以及其相应的错误标志）。
- 当 FIFO 模式使能时，RXFNE 位置 1 表示 RXFIFO 不为空。读取 LPUART_RDR 将获取 RXFIFO 中最早的条目。在接收到数据时，数据和与之相关的错误位均保存在接收 RXFIFO 中。
- 如果 RXNEIE（FIFO 模式下为 RXFNEIE）位置 1，将生成中断。
- 如果接收期间已检测到帧错误、噪声错误或上溢错误，错误标志位可置 1。
- 在多缓冲区通信中：
  - 当 FIFO 模式禁用时，RXNE 在每个字节被接收后置 1，并通过接收数据寄存器的 DMA 读操作清零。
  - 当 FIFO 模式使能时，RXFNE 标志在 RXFIFO 不为空时置 1。在每次 DMA 请求后，将从 RXFIFO 提取数据。如果不为空，即 RXFIFO 中有等待读取的数据，RXFIFO 将触发 DMA 请求。
- 在单缓冲区模式下：
  - 当 FIFO 模式禁用时：通过软件对 LPUART_RDR 寄存器的读取执行 RXNE 位清零。也可以通过将 LPUART_RQR 寄存器中的 RXFRQ 位置 1 将 RXNE 标志位清零。RXNE 位必须在结束接收下一个字符前清零，以避免发生上溢错误。
  - 当 FIFO 模式使能时，RXFNE 标志在 RXFIFO 不为空时置 1。在每次读取 LPUART_RDR 寄存器后，从 RXFIFO 提取数据。当 RXFIFO 为空时，RXFNE 标志清零。也可以通过将 LPUART_RQR 寄存器中的 RXFRQ 位置 1 将 RXFNE 标志位清零。当 RXFIFO 已满时，必须在下一个字符的接收结束前读取 RXFIFO 中的第一个条目，以避免发生上溢错误。RXFNEIE 位置 1 时 RXFNE 标志位会生成中断。或者，当达到 RXFIFO 阈值时，可生成中断，并能从 RXFIFO 读取数据。在这种情况下，CPU 可读取由设定的阈值定义的数据块。

中断字符

接收到中断字符时，LPUART 将会按照数据拆包错误对其进行处理。

空闲字符

检测到空闲帧时，处理步骤与接收到数据的情况相同；如果 IDLEIE 位置 1，则会产生中断。

上溢错误

- FIFO 模式禁用

如果在 RXNE 未复位时接收到字符，则会发生上溢错误。RXNE 位清零前，数据无法从移位寄存器传送到 RDR 寄存器。每接收到一个字节后，RXNE 标志位都将置 1。当 RXNE 标志位是 1 时，如果在接收到下一个数据或尚未处理上一个 DMA 请求时，则会发生上溢错误。发生上溢错误时：

- ORE 位置 1。
- RDR 内容不丢失。对 LPUART_RDR 执行读操作时可使用先前的数据。
- 移位寄存器被重写。之后，上溢期间接收到的任何数据都将丢失。

- 如果 RXNEIE 位置 1 或 EIE 位置 1，则会生成中断。
- FIFO 模式启用。

当接收 FIFO 已满时，如果移位寄存器准备进行传输，将发生上溢错误。在 RXFIFO 中有一个空位之前，无法将数据从移位寄存器传输到 LPUART_RDR 寄存器。当 RXFIFO 不为空时，RXFNE 标志置 1。如果 RXFIFO 已满且移位寄存器准备进行传输，将发生上溢错误。发生上溢错误时：

- ORE 位置 1。
- RXFIFO 中的第一个条目不会丢失。在对 LPUART_RDR 执行读取时，可读取该条目。
- 移位寄存器被重写。之后，上溢期间接收到的任何数据都将丢失。
- 如果 RXFNEIE 位置 1 或 EIE 位置 1，则会生成中断。

通过设置 ICR 寄存器中的 ORECF 位复位 ORE 位。

注意： *ORE 位置 1 时表示至少 1 个数据丢失。*

注意： *当 FIFO 模式禁用时，有两种可能性。*

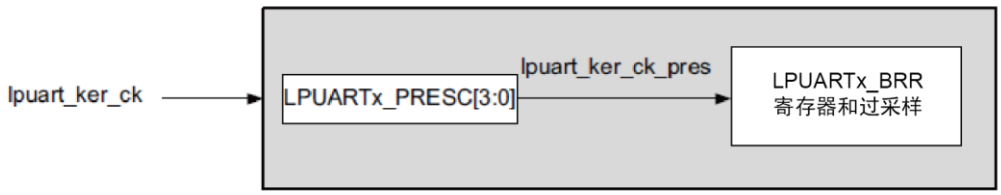
- *如果 RXNE=1，则最后一个有效数据存储于接收寄存器 RDR 中并且可进行读取*
- *如果 RXNE=0，则表示最后一个有效数据已被读取，因此 RDR 中没有要读取的数据。接收到新（丢失）数据的同时已读取 RDR 中的最后一个有效数据时，会发生该情况。*

选择时钟源

时钟源频率为 f_{CK}（16 MHz）。

可通过 LPUARTx_PRESC 寄存器中的可编程因子对 f_{CK} 进行分频。

图 142. lpuart_ker_ck 时钟分频器框图



时钟源还决定通信速度范围（尤其是最大通信速度）。

接收器在尽可能靠近波特周期中间的位置对每个传入波特进行采样。每个传入波特只采集一个样本。

注意： *无数据噪声检测。*

数据拆包错误

在去同步化或过度的噪声后，如果接收时未在预期时间识别出停止位，将检测到数据拆包错误。

检测到数据拆包错误时：

- FE 位由硬件置 1
- 无效数据从移位寄存器传送到 LPUART_RDR 寄存器
- 单字节通信时无中断产生。然而，在 RXNE 位产生中断时，该位出现上升沿。在多缓冲区通信中，如果 LPUART_CR3 寄存器中的 EIE 位置 1，则生成中断。

通过将 1 写入 LPUART_ICR 寄存器中的 FECF 位来复位 FE 位。

接收期间的可配置停止位

可通过 LPUART_CR2 寄存器中的控制位配置要接收的停止位的数量：在正常模式下，可以是 1 或 2 个。

- **1 个停止位**：将在第 8、第 9 和第 10 次采样时对 1 个停止位进行采样。
- **2 个停止位**：2 个停止位的采样在第二个停止位的中间进行。此次采样后（即第二个停止位期间）RXNE 和 FE 标志置 1。发生数据拆包错误时不检测第 1 个停止位。

21.3.5 波特率生成

接收器和发射器（Rx 和 Tx）的波特率设置为 LPUART_BRR 寄存器中编程的相同值。

$$Tx/Rxbaud = \frac{256xf_{CKPRES}}{LPUARTDIV} \quad (7)$$

在 LPUART_BRR 寄存器中进行 LPUARTDIV 编码。

注意： *对 LPUART_BRR 执行写操作后，波特率计数器更新为波特率寄存器中的新值。因此，波特率寄存器的值不应在通信时发生更改。禁止在 LPUART_BRR 寄存器中写入小于 0x300 的值。f_{CK} 必须处于范围（3 x 波特率）至（4096 x 波特率）以内。*

21.3.6 多处理器通信

可以通过 LPUART 进行多处理器通信（多个 LPUART 连接在一个网络中）。例如，其中一个 LPUART 可以是主 USART，其 TX 输出与其它 LPUART 的 RX 输入相连接。其它 USART 为从 USART，其各自的 TX 输出在逻辑上通过与运算连在一起，并与主 USART 的 RX 输入相连接。

在多处理器配置中，理想情况下通常只有预期的消息接收方主动接收完整的消息内容，从而减少由所有未被寻址的接收器造成的冗余 LPUART 服务开销。

可通过静默功能将未被寻址的器件置于静默模式下。为了使用静默模式功能，必须将 LPUART_CR1 寄存器中的 MME 位置 1。

注意： *当 FIFO 管理使能且 MME 已置位时，不得将 MME 位清零然后再次快速置位（在两个 UCLK 周期内），否则静默模式可能保持有效状态。*

在静默模式下：

- 不得将接收状态位置 1
- 禁止任何接收中断
- LPUART_ISR 寄存器中的 RWU 位置 1。在某些情况下，RWU 可以由硬件或软件通过 LPUART_RQR 寄存器中的 MMRQ 位自动控制。

根据 LPUART_CR1 寄存器中 WAKE 位的设置，LPUART 可使用以下两种方法进入或退出静默模式：

- 如果 WAKE 位被复位，则进行空闲线路检测
- 如果 WAKE 位置 1，则进行地址标记检测

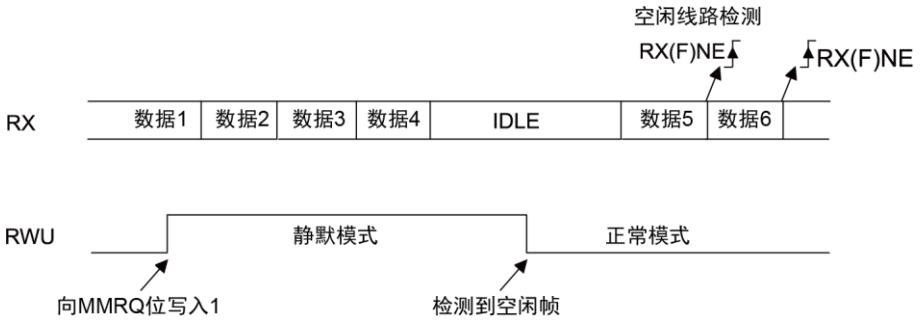
空闲线路检测（WAKE=0）

当向 MMRQ 位写入 1 且 RWU 位自动置 1 时，LPUART 进入静默模式。

当检测到空闲帧时，它会被唤醒。然后，RWU 位会由硬件清零，

但 LPUART_ISR 寄存器中的 IDLE 位不会置 1。使用空闲线路检测的静默模式行为的示例见“图 121. 使用空闲线路检测的静默模式”。

图 143. 使用空闲线路检测的静默模式



注意： *如果在 IDLE 字符已经过去时将 MMRQ 位置 1，将不会进入静默模式（RWU 未置 1）。如果在线路处于空闲状态时激活 LPUART，在一个 IDLE 帧持续时间后（不只在接收一个字符帧后）会检测到空闲状态。*

4 位/7 位地址标记检测（WAKE=1）

在此模式下，如果字节的 MSB 为 1，则将这些字节识别为地址，否则将其识别为数据。在地址字节中，目标接收器的地址位于 4 个或 7 个 LSB 中。7 位或 4 位地址检测通过 ADDM7 位来选择。接收器会将此 4 位/7 位字与其地址进行比较，该接收器的地址在 LPUART_CR2 寄存器的 ADD 位中进行设置。

注意： *在 7 位和 9 位数据模式下，地址检测分别在 6 位和 8 位地址上完成（ADD[5:0]和 ADD[7:0]）。*

当接收到与其编程地址不匹配的地址字符时，LPUART 会进入静默模式。此时，RWU 位将由硬件置 1。LPUART 进入静默模式后，RXNE 标志不会针对此地址字节置 1，也不会发出中断或 DMA 请求。

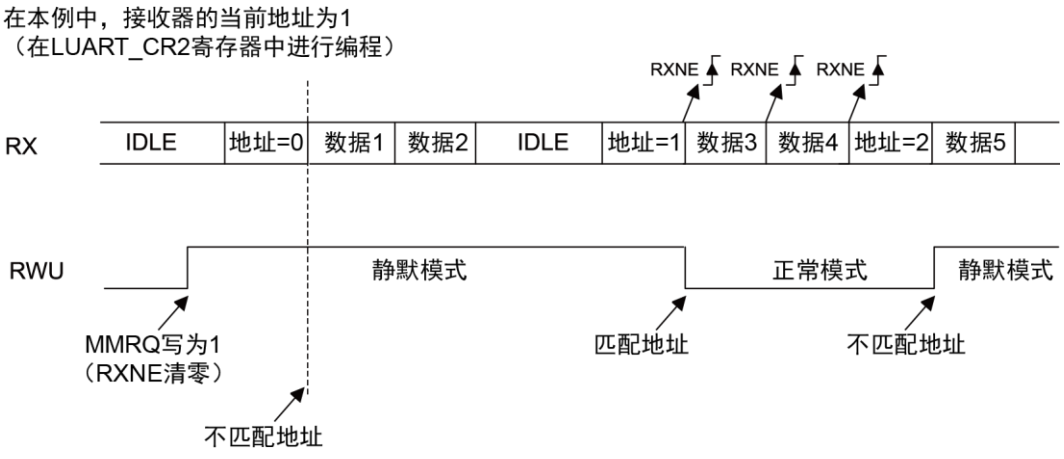
当向 MMRQ 位写入 1 时，LPUART 也会进入静默模式。这种情况下，RWU 位也自动置 1。

当接收到与编程地址匹配的地址字符时，LPUART 会退出静默模式。然后 RWU 位被清零，可以开始正常接收后续字节。由于 RWU 位已清零，RXNE/RXFNE 位会针对地址字符置 1。

注意： *当 FIFO 管理使能时，MMRQ 位在接收器对数据的最后一位进行采样时置 1；此数据可能会在实际进入静默模式之前被接收。*

使用地址标记检测的静默模式行为的示例见“图 122. 使用地址标记检测的静默模式”。

图 144. 使用地址标记检测的静默模式



21.3.7 奇偶校验控制

将 LPUART_CR1 寄存器中的 PCE 位置 1，可以使能奇偶校验控制（发送时生成奇偶校验位，接收时进行奇偶校验检查）。根据 M 位定义的帧长度，“表 73. 帧格式”列出了可能的 LPUART 帧格式。帧格式。

表 73. 帧格式

M 位	PCE 位	LPUART 帧(1)
00	0	| SB | 8 位数据 | STB |
00	1	| SB | 7 位数据 | PB | STB |
01	0	| SB | 9 位数据 | STB |
01	1	| SB | 8 位数据 PB | STB |
10	0	| SB | 7 位数据 | STB |
10	1	| SB | 6 位数据 | PB | STB |

1. *图注：SB：起始位，STB：停止位，P：奇偶校验位。在数据寄存器中，PB 始终位于 MSB 位置（第 8 位或第 7 位，具体取决于 M 位的值）。*

偶校验

对奇偶校验位进行计算，使帧和奇偶校验位中“1”的数量为偶数（帧由 6 个、7 个或 8 个 LSB 位组成，具体取决于 M 位的值）。

例如，如果数据=00110101 且 4 个位置 1，则在选择偶校验（LPUART_CR1 寄存器中的 PS 位 = 0）时，校验位为 0。

奇校验

对奇偶校验位进行计算，使帧和奇偶校验位中“1”的数量为奇数（帧由 6 个、7 个或 8 个 LSB 位组成，具体取决于 M 位的值）。

例如，如果数据=00110101 且 4 个位置 1，则在选择奇校验（LPUART_CR1 寄存器中的 PS 位 = 1）时，校验位为 1。

接收时进行奇偶校验检查

如果奇偶校验检查失败，则 LPUART_ISR 寄存器中的 PE 标志置 1；如果 LPUART_CR1 寄存器中 PEIE 位置 1，则会生成中断。通过软件将 1 写入 LPUART_ICR 寄存器中的 PECF 位来清零 PE 标志。

发送中的奇偶校验生成

如果 LPUART_CR1 寄存器中的 PCE 位置 1，则在数据寄存器中所写入数据的 MSB 位会进行传送，但是会由奇偶校验位进行更改（如果选择偶校验（PS=0），则“1”的数量为偶数；如果选择奇校验（PS=1），则“1”的数量为奇数）。

21.3.8 单线半双工通信

通过将 LPUART_CR3 寄存器中的 HDSEL 位置 1 来选择单线半双工模式。

LPUART 可以配置为遵循单线半双工协议，其中 TX 和 RX 线路从内部相连接。使用 LPUART_CR3 寄存器中的控制位 HDSEL 可在半双工通信和全双工通信间进行选择。

一旦向 HDSEL 位写入 1：

- TX 和 RX 线路从内部相连接
- 不能再使用 RX 引脚
- 无数据传输时，TX 引脚始终处于释放状态。因此，它在空闲状态或接收过程中用作标准 I/O。这意味着，必须对 I/O 进行配置，以便将 TX 配置为复用功能开漏并外接上拉电阻。

除此之外，通信协议与正常 LPUART 模式下的通信协议相似。此线路上的任何冲突必须由软件管理（例如，使用中央仲裁器）。尤其要注意，发送过程永远不会被硬件封锁，只要数据是在 TE 位置 1 的情况下写入，发送就会持续进行。

注意： *由于 TX 线路和 RX 线路连在一起，发送的所有数据与从外部器件接收的数据一样，都存储在 RX FIFO 中。在发送阶段后，软件必须谨慎地丢弃其“自己的”信息。在半双工模式下，由于没有针对节点之间可能发生的冲突的硬件保护，回读发送的数据以检查数据是否正确始终是一种明智的做法。如果不想让 RX FIFO 存储发送的值，则必须在发送时禁用接收器部分（通过将 USART_CR1 寄存器中的 RE 位清零）。*

注意： *在 LPUART 中，如果采用 1-STOP 位配置，RXNE 标志将在 STOP 位的中间置位。*

21.3.9 使用 DMA 进行连续通信

LPUART 能够使用 DMA 执行连续通信。接收缓冲区和发送缓冲区的 DMA 请求是独立的。

注意： *请参见第 20.4 节“USART 实现”，以确定是否支持 DMA 模式。如果不支持 DMA 模式，则使用 LPUSRT 执行连续通信。当 FIFO 禁用时，可将 LPUART_ISR 寄存器中的 TXE/RXNE 标志清零。*

使用 DMA 发送

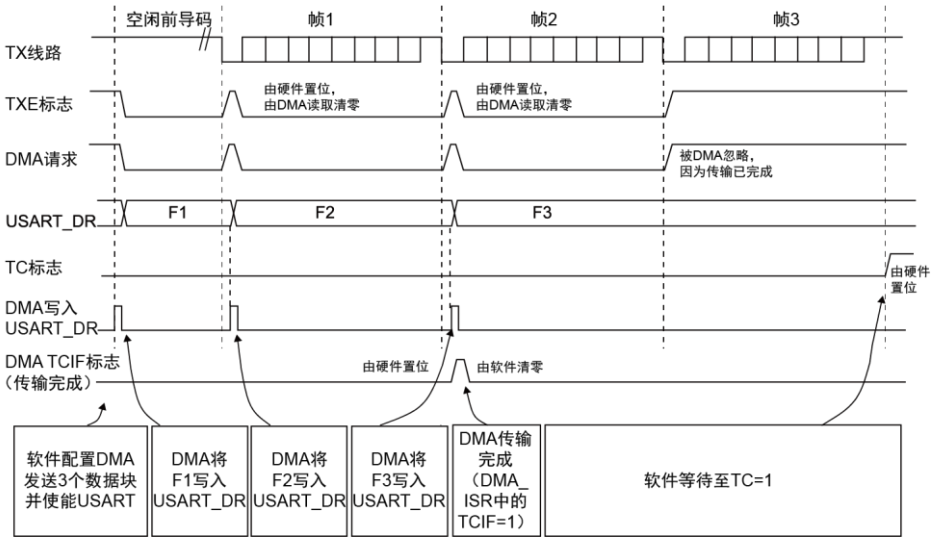
将 LPUART_CR3 寄存器中的 DMAT 位置 1 可以使能 DMA 模式进行发送。每当 TXE 标志（FIFO 模式使能时为 TXFNF 标志）置 1 时，将数据从使用 DMA 外设配置的 SRAM 区加载到 LPUART_TDR 寄存器。要映射一个 DMA 通道以进行 LPUART 发送，请按以下步骤操作（x 表示通道编号）：

1. 在 DMA 控制寄存器中写入 LPUART_TDR 寄存器地址，将其配置为传输的目标地址。在每个 TXE（FIFO 模式使能时为 TXFNF）事件后，数据会从存储器移动到此地址。
2. 在 DMA 控制寄存器中写入存储器地址，将其配置为传输的源地址。在每个 TXE（FIFO 模式使能时为 TXFNF）事件后，数据会从该存储区加载到 LPUART_TDR 寄存器中。
3. 在 DMA 控制寄存器中配置要传输的总字节数。
4. 在 DMA 寄存器中配置通道优先级。
5. 根据应用的需求，在完成一半或全部传输后产生 DMA 中断。
6. 通过将 LPUART_ICR 寄存器中的 TCCF 位置 1，将 LPUART_ISR 寄存器中的 TC 标志清零。
7. 在 DMA 寄存器中激活该通道。

当达到在 DMA 控制器中设置的数据传输量时，DMA 控制器会在 DMA 通道的中断向量上产生一个中断。

在发送模式下，DMA 对所有要发送的数据执行了写操作（DMA_ISR 寄存器中的 TCIF 标志置 1）后，可以对 TC 标志进行监视，以确保 LPUART 通信已完成。在禁止 LPUART 或进入停止模式前必须执行此步骤，以避免损坏最后一次发送。软件必须等待直到 TC=1。TC 标志在所有数据发送期间都保持清零状态，然后在最后一帧发送结束时由硬件置 1。

图 145. 使用 DMA 发送



注意： *当 FIFO 管理使能时，发送 FIFO 未满（即 TXFNF = 1）会触发 DMA 请求。*

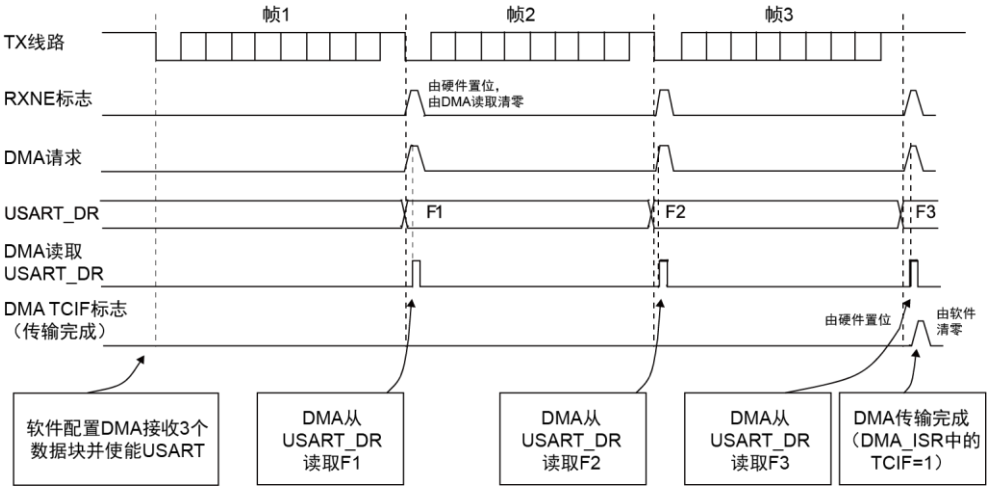
使用 DMA 接收

将 LPUART_CR3 寄存器中的 DMAR 位置 1 可以使能 DMA 模式进行接收。每当接收到数据字节时，将数据从 LPUART_RDR 寄存器加载到使用 DMA 外设配置的 SRAM 区。要映射一个 DMA 通道以进行 LPUART 接收，请按以下步骤操作：

1. 在 DMA 控制寄存器中写入 LPUART_RDR 寄存器地址，将其配置为传输的源地址。在每个 RXNE（FIFO 模式使能时为 RXFNE）事件后，数据会从此地址移动到存储器。
2. 在 DMA 控制寄存器中写入存储器地址，将其配置为传输的目标地址。在每个 RXNE（FIFO 模式使能时为 RXFNE）事件后，数据会从 LPUART_RDR 寄存器加载到该存储区。
3. 在 DMA 控制寄存器中配置要传输的总字节数。
4. 在 DMA 控制寄存器中配置通道优先级。
5. 根据应用的需求，在完成一半或全部传输后产生中断。
6. 在 DMA 控制寄存器中激活该通道。

当达到在 DMA 控制器中设置的数据传输量时，DMA 控制器会在 DMA 通道的中断向量上产生一个中断。

图 146. 使用 DMA 接收



注意： *当 FIFO 管理使能时，接收 FIFO 不为空（即 RXFNE = 1）会触发 DMA 请求。*

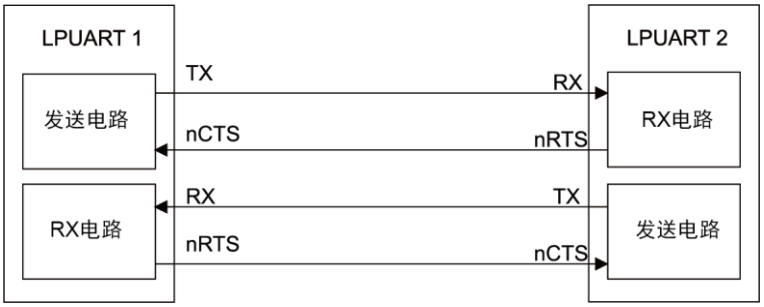
多缓冲区通信中的错误标志和中断生成

在多缓冲区通信中，如果事务中发生任何错误，都会在当前字节后放置错误标志。如果中断使能标志置 1，则会产生中断。对于在单字节接收中与 RXNE（FIFO 模式使能时为 RXFNE）一同置位的数据拆包错误、上溢错误和噪声标志，有一个单独的错误标志中断使能位（LPUART_CR3 寄存器中的 EIE 位），如果此位置 1，将在发生这些错误中的任意一种时在当前字节后使能中断。

21.3.10 RS232 硬件流控制和 RS485 驱动器使能

使用 nCTS 输入和 nRTS 输出可以控制 2 个器件间的串行数据流。图 135. 2 个 USART 间的硬件流控制显示了如何在该模式下连接 2 个器件：

图 147. 2 个 LPUART 间的硬件流控制

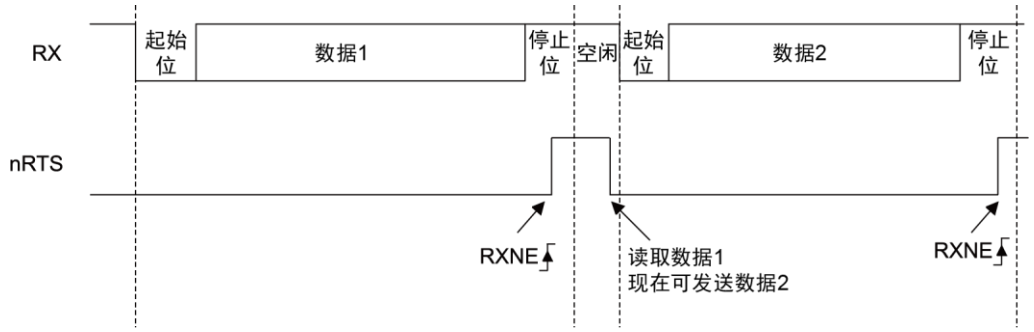


分别向 LPUART_CR3 寄存器中的 RTSE 位和 CTSE 位写入 1，可以分别使能 RS232 RTS 和 CTS 流控制。

RS232 RTS 流控制

如果使能 RTS 流控制（RTSE=1），只要 LPUART 接收器准备好接收新数据，便会将 nRTS 变为有效（连接到低电平）。当接收寄存器已满时，会将 nRTS 变为无效，表明发送过程会在当前帧结束后停止。图 136. RS232 RTS 流控制显示了在 RTS 流控制使能时进行通信的示例。

图 148. **RS232 RTS 流控制**



注意： *当 FIFO 模式使能时，nRTS 仅在 RXFIFO 已满时变为无效。*

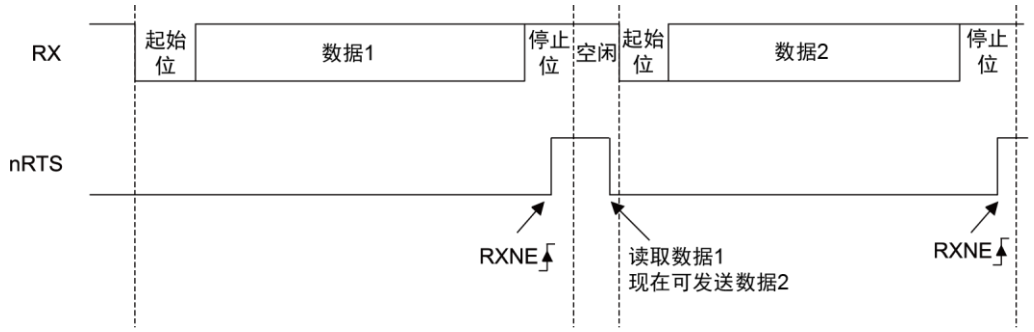
RS232 CTS 流控制

如果使能 CTS 流控制（CTSE=1），则发射器会在发送下一帧前检查 nCTS。如果 nCTS 有效（连接到低电平），则会发送下一数据（假设数据已准备好发送，即 TXE/TXFE=0）；否则不会进行发送。如果在发送过程中 nCTS 变为无效，则当前发送完成之后，发射器停止。

当 CTSE=1 时，只要 CTS 发生变化，CTSIF 状态位便会由硬件自动置 1。这指示接收器是否已准备好进行通信。

如果 LPUART_CR3 寄存器中的 CTSIE 位置 1，则会产生中断。下图显示了在 CTS 流控制使能时进行通信的示例。

图 149. **RS232 RTS 流控制**



注意： *为正常运行，必须在当前字符结束前至少 3 个 LPUART 时钟源周期内使能 nCTS。此外还应注意，当脉冲短于 2 个 PCLK 周期时无法将 CTSCF 标志置 1。*

RS485 驱动器使能

驱动器使能功能可通过将 LPUART_CR3 控制寄存器中的 DEM 位置 1 使能。这样用户便可通过 DE（驱动器使能）信号激活外部收发器控制。使能时间为激活 DE 信号与 START 位开始间的时间。可以通过 LPUART_CR1 控制寄存器中的 DEAT [4:0]位字段编程禁止时间。禁止时间为发送的消息中最后一个停止位结束与取消激活 DE 信号间的时间。可以通过 LPUART_CR1 控制寄存器中的 DEDT [4:0]位字段编程禁止时间。DE 信号的极性可使用 LPUART_CR3 控制寄存器中的 DEP 位配置。

在 LPUART 中，以 LPUART 时钟源（fCK）周期为单位表示 DEAT 和 DEDT：

- 驱动器使能置位时间 =
  - （1 +（DEAT x P））x f_{CK}，如果 P /=0
  - （1 + DEAT）x f_{CK}，如果 P =0
- 驱动器使能取消置位时间 =
  - （1 +（DEDT x P））x f_{CK}，如果 P /=0
  - （1 + DEDT）x f_{CK}，如果 P =0

且 P = BRR[20:11]

21.4 LPUART 中断

表 74. LPUART 中断请求

中断事件	事件标志	使能控制位
发送数据寄存器为空	TXE	TXEIE
发送 FIFO 未满	TXFNF	TXFNFIE
发送 FIFO 为空	TXFE	TXFEIE
CTS 中断	CTSIF	CTSIE
发送完成	TC	TCIE
接收数据寄存器不为空（已准备好读取数据）	RXNE	RXNEIE
接收 FIFO 非空	RXFNE	RXFNEIE
接收 FIFO 已满	RXFF	RXFFIE
检测到溢出错误	ORE	RXNEIE/RXF- NEIE
检测到空闲线路	IDLE	IDLEIE
奇偶校验错误	PE	PEIE
多缓冲区通信中的噪声标志、上溢错误和数据拆包错误	NF 或 ORE 或 FE	EIE
字符匹配	CMF	CMIE

如果相应的使能控制位置 1，则这些事件会生成中断。

21.5 LPUART 寄存器

请参见第 1.3 节“缩略语”获取寄存器说明中所用缩略语的列表。

21.5.1 控制寄存器 1（LPUART_CR1）

偏移地址：0x00

复位值：0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXFFI E	TXFEIE	FIFOE N	M1	Res.	Res.	DEAT[4:0]					DEDT[4:0]				
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
Res.	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXEIE TXFNFI E	TCIE	RXNEIE RXFNEI E	IDLEIE	TE	RE	Res.	UE
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

位 31	**RXFFIE**：RXFIFO 已满中断使能。此位由软件置 1 和清零。 0：禁止中断 1：当 LPUART_ISR 寄存器中的 RXFF=1 时，生成 LPUART 中断 注意：当 FIFO 模式禁用时，此位保留且必须保持复位值。
位 30	**TXFEIE**：TXFIFO 空中断使能。此位由软件置 1 和清零。 0：禁止中断 1：当 LPUART_ISR 寄存器中的 TXFE=1 时，生成 LPUART 中断。 注意：当 FIFO 模式禁用时，此位保留且必须保持复位值。
位 29	**FIFOEN**：FIFO 模式使能。 此位由软件置 1 和清零。 0：FIFO 模式禁用 1：FIFO 模式使能
位 28	**M1**：字长。 此位和位 12（M0）用于确定字长度。该位由软件置 1 或清零。 M[1:0] = 00:1 起始位，8 数据位，n 停止位 M[1:0] = 01:1 起始位，9 数据位，n 停止位 M[1:0] = 10:1 起始位，7 数据位，n 停止位 只有在禁止 LPUART（UE=0）时才能写入此位。 注：在 7 位数据长度模式下，不支持智能卡模式、LIN 主模式和自动波特率（0x7F 和 0x55 帧检测）。
位 27	保留，必须保持复位值。
位 26	保留，必须保持复位值。
位 25:21	**DEAT[4:0]**：驱动器使能置位时间。 该 5 位值用于定义激活 DE（启动器使能）信号与起始位开始间的时间。它以 UCLK（LPUART 时钟）时钟周期为单位。 请参见 RS485 驱动器使能一节了解更多信息。 只有在禁止 LPUART（UE=0）后才能写入此位字段。
位 20:16	**DEDT[4:0]**：驱动器使能取消置位时间。 该 5 位值用于定义发送的消息中最后一个停止位结束与取消激活 DE（驱动器使能）信号间的时间。它以 UCLK（LPUART 时钟）时钟周期为单位。请参见 RS485 驱动器使能一节了解更多信息。如果在 DEDT 时间内对 LPUART_TDR 寄存器执行写操作，则新数据仅在经过 DEDT 和 DEAT 时间后才会发送。 只有在禁止 LPUART（UE=0）后才能写入此位字段。
位 15	保留，必须保持复位值。
位 14	**CMIE**：字符匹配中断使能。此位由软件置 1 和清零。 0：禁止中断 1：LPUART_ISR 寄存器中的 CMF 位置 1 时生成 LPUART 中断
位 13	**MME**：静默模式使能。 此位激活 LPUART 的静默模式功能。当此位置 1 时，LPUART 可按照 WAKE 位的定义，在活动模式与静默模式之间切换。该位由软件置 1 和清零。 0：接收器永久处于活动模式 1：接收器可在静默模式和活动模式之间切换
位 12	**M0**：字长。 此位和位 28（M1）用于确定字长度。该位由软件置 1 或清零。请参见位 28（M1）的说明。 只有在禁止 LPUART（UE=0）时才能写入此位。

位 11	**WAKE**：接收器唤醒方法。 此位决定了从静默模式唤醒 LPUART 的方法。该位由软件置 1 或清零。 0：空闲线 1：地址标记 只有在禁止 LPUART（UE=0）后才能写入此位字段。
位 10	**PCE**：奇偶校验控制使能。 该位选择硬件奇偶校验控制（生成和检测）。使能奇偶校验控制时，计算出的奇偶校验位被插入到 MSB 位置（如果 M=1，则为第 9 位；如果 M=0，则为第 8 位），并对接收到的数据检查奇偶校验位。此位由软件置 1 和清零。一旦该位置 1，PCE 在当前字节的后面处于活动状态（在接收和发送时）。 0：禁止奇偶校验控制 1：使能奇偶校验控制 只有在禁止 LPUART（UE=0）后才能写入此位字段。
位 9	**PS**：奇偶校验的选择。 该位用于在使能奇偶校验生成/检测（PCE 位置 1）时选择奇校验或偶校验。该位由软件置 1 和清零。在当前字节之后选择奇偶校验。 0：偶校验 1：奇校验 只有在禁止 LPUART（UE=0）后才能写入此位字段。
位 8	**PEIE**：PE 中断使能。 此位由软件置 1 和清零。 0：禁止中断 1：每当 LPUART_ISR 寄存器中的 PE=1 时，生成 LPUART 中断
位 7	**TXEIE/TXFNFIE**：发送数据寄存器为空/TXFIFO 未满中断使能。此位由软件置 1 和清零。 0：禁止中断 1：每当 LPUART_ISR 寄存器中的 TXE/TXFNF=1 时，生成 LPUART 中断
位 6	**TCIE**：传送完成中断使能。此位由软件置 1 和清零。 0：禁止中断 1：每当 LPUART_ISR 寄存器中的 TC=1 时，生成 LPUART 中断
位 5	**RXNEIE/RXFNEIE**：接收数据寄存器不为空/RXFIFO 不为空中断使能。此位由软件置 1 和清零。 0：禁止中断 1：每当 LPUART_ISR 寄存器中的 ORE=1 或 RXNE/RXFNE=1 时，生成 LPUART 中断
位 4	**IDLEIE**：IDLE 中断使能。 此位由软件置 1 和清零。 0：禁止中断 1：每当 LPUART_ISR 寄存器中的 IDLE=1 时，生成 LPUART 中断
位 3	**TE**：发射器使能。 该位使能发射器。该位由软件置 1 和清零。 0：禁止发射器 1：使能发射器 注意：在传送期间，TE 位上的“0”脉冲（“0”后紧跟的是“1”）会在当前字的后面发送一个前导码（空闲线路）。为生成空闲字符，TE 不能立即写入 1。为确保所需的持续时间，软件可轮询 LPUART_ISR 寄存器中的 TEACK 位。 当 TE 置 1 时，在发送开始前存在 1 位的时间延迟。

位 2	**RE**：接收器使能。 该位使能接收器。该位由软件置 1 和清零。 0：禁止接收器 1：使能接收器并开始搜索起始位
位 1	保留，必须保持复位值。
位 0	**UE**：使能 LPUART。 此位清零后，LPUART 预分频器和输出将立即停止，并丢弃当前操作。LPUART 的配置会被保留，但 LPUART_ISR 中的所有状态标志均复位。此位由软件置 1 和清零。 0：LPUART 预分频器和输出禁用，低功耗模式 1：使能 LPUART 注意：为进入低功耗模式而不在线路上生成错误，之前必须复位 TE 位，并且软件必须等待 LPUART_ISR 寄存器中的 TC 位置 1 后才能复位 UE 位。 UE = 0 时也会复位 DMA 请求，因必须在复位 UE 位前禁止 DMA 通道。

21.5.2 控制寄存器 2（LPUART_CR2）

偏移地址：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]				Res.	Res.	Res.	Res.	MSBFI RST	DATAINV	TXINV	RXINV
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	Res.	STOP[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDM7	Res.	Res.	Res.	Res.
rw		rw	rw								rw				

位 31:28	**ADD[7:4]**：LPUART 节点的地址。 此位字段用于指定 LPUART 节点的地址或要识别的字符代码。 在静默模式或停止模式下的多处理器通信中，它被用于通过 7 位地址标记检测进行唤醒。发射器发送字符的 MSB 应为 1。此位字段还可用于正常接收和静默模式无效时的字符检测（例如，ModBus 协议中的块结束检测）。这种情况下，接收到的整个字符（8 位）将与 ADD[7:0]值进行比较，如果匹配，CMF 标志将置 1。 仅在禁止接收（RE = 0）或禁止 LPUART（UE=0）时才能写入该位字段。
位 27:24	**ADD[3:0]**：LPUART 节点的地址。 此位字段用于指定 LPUART 节点的地址或要识别的字符代码。 在静默模式或停止模式下的多处理器通信中，它被用于通过地址标记检测进行唤醒。 仅在禁止接收（RE = 0）或禁止 LPUART（UE=0）时才能写入该位字段
位 23:20	保留，必须保持复位值。
位 19	**MSBFIRST**：最高有效位优先。 此位由软件置 1 和清零。 0：发送/接收数据时位 0 在前，后跟起始位 1：发送/接收数据时 MSB（位 7/8）在前，后跟起始位。只有在禁止 LPUART（UE=0）后才能写入此位字段。
位 18	**DATAINV**：二进制数据取反。 此位由软件置 1 和清零。 0：按正/正向逻辑发送/接收数据寄存器中的逻辑数据。（1=H，0=L）

	1：按负/反向逻辑发送/接收数据寄存器中的逻辑数据。（1=L，0=H）。奇偶校验位也取反。 只有在禁止 LPUART（UE=0）后才能写入此位字段。
位 17	**TXINV**：TX 引脚有效电平取反。此位由软件置 1 和清零。 0：TX 引脚信号使用标准逻辑电平工作（VDD=1/空闲，Gnd=0/标记） 1：TX 引脚信号值取反。（VDD=0/标记，Gnd=1/空闲）。允许在 TX 线路上使用外部反相器。 只有在禁止 LPUART（UE=0）后才能写入此位字段。
位 16	**RXINV**：RX 引脚有效电平取反。此位由软件置 1 和清零。 0：RX 引脚信号使用标准逻辑电平工作（VDD=1/空闲，Gnd=0/标记） 1：RX 引脚信号值取反。（VDD=0/标记，Gnd=1/空闲）。允许在 RX 线路上使用外部反相器。 只有在禁止 LPUART（UE=0）后才能写入此位字段。
位 15	**SWAP**：交换 TX/RX 引脚。 此位由软件置 1 和清零。 0：按标准引脚排列定义使用 TX/RX 引脚 1：交换 TX 和 RX 引脚功能。允许在与另一个 UART 的交叉连接时工作。 只有在禁止 LPUART（UE=0）后才能写入此位字段。
位 14	保留，必须保持复位值。
位 13:12	**STOP[1:0]**：STOP 位。 这些位用于编程停止位。 00：1 个停止位 01：保留 10：2 个停止位 11：保留 只有在禁止 LPUART（UE=0）后才能写入此位字段。
位 11:5	保留，必须保持复位值。
位 4	**ADDM7**：7 位地址检测/4 位地址检测。 此位用于选择 4 位地址检测或 7 位地址检测。 0：4 位地址检测 1：7 位地址检测（在 8 位数据模式下） 只有在禁止 LPUART（UE=0）时才能写入此位。 注意：在 7 位和 9 位数据模式下，地址检测分别在 6 位和 8 位地址上完成（ADD[5:0]和 ADD[7:0]）。
位 3:0	保留，必须保持复位值。

21.5.3 控制寄存器 3（LPUART_CR3）

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
TXFTCFG			RXFTI E.	RXFTCFG			Res.	TXFTIE	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
DEP	DEM	DDRE	OVRDIS	Res.	CTSIE	CTSE	RTSE	DMAT	DMAR	Res.	Res.	HD SEL	Res.	Res.	EIE
rw	rw	rw	rw		rw	rw	rw	rw	rw			rw			rw

位 31:29	**TXFTCFG**：TXFIFO 阈值配置。 000：TXFIFO 达到其深度的 1/8 001：TXFIFO 达到其深度的 1/4 110：TXFIFO 达到其深度的 1/2 011：TXFIFO 达到其深度的 3/4 100：TXFIFO 达到其深度的 7/8 101：TXFIFO 变为空 剩余组合：保留。
Bit28	**RXFTIE**：RXFIFO 阈值中断使能。此位由软件置 1 和清零。 0：禁止中断 1：当接收 FIFO 达到在 RXFTCFG 中设定的阈值时，生成 LPUART 中断
位 27:25	**RXFTCFG**：接收 FIFO 阈值配置。 000：接收 FIFO 达到其深度的 1/8 001：接收 FIFO 达到其深度的 1/4 110：接收 FIFO 达到其深度的 1/2 011：接收 FIFO 达到其深度的 3/4 100：接收 FIFO 达到其深度的 7/8 101：接收 FIFO 已满 剩余组合：保留。
位 24	保留，必须保持复位值。
位 23	**TXFTIE**：TXFIFO 阈值中断使能。此位由软件置 1 和清零。 0：禁止中断 1：当 TXFIFO 达到在 TXFTCFG 中设定的阈值时，生成 LPUART 中断
位 22:16	保留，必须保持复位值。
位 15	**DEP**：驱动器使能极性选择。 0：DE 信号为高电平有效 1：DE 信号为低电平有效 只有在禁止 LPUART（UE=0）时才能写入此位。
位 14	**DEM**：驱动器使能模式。 此位用于通过 DE 信号激活外部收发器控制。 0：DE 功能禁用 1：DE 功能使能。DE 信号在 RTS 引脚上输出。 只有在禁止 LPUART（UE=0）时才能写入此位。
位 13	**DDRE**：接收出错时 DMA 禁用。 0：接收出错时不禁止 DMA。相应的错误标志置 1，但 RXNE 保持为 0 以防止上溢。因此，将不使能 DMA 请求，从而不会传送错误数据（无 DMA 请求），但会传送接收到的下一个正确数据。 1：接收出错后禁止 DMA。相应的错误标志置 1，RXNE 也置 1。在错误标志清零之前，DMA 请求被屏蔽。这意味着软件必须首先禁止 DMA 请求（DMAR = 0）或者将 RXNE 清零，然后再将错误标志清零。 只有在禁止 LPUART（UE=0）时才能写入此位。 注意：接收错误包括：奇偶校验错误、数据拆包错误或噪声错误。

位 12	**OVRDIS**：上溢禁用。 此位用于禁止接收上溢检测。 0：接收新数据前未读取已接收的数据时，上溢错误标志 ORE 置 1 1：上溢功能被禁用。如果在 RXNE 标志仍然置位时接收到新数据，ORE 将不置位且接收到的新数据会覆盖 LPUART_RDR 寄存器之前的内容。 只有在禁止 LPUART（UE=0）时才能写入此位。 注意：此控制位用于在不读取数据的情况下检查通信流。
位 11	保留，必须保持复位值。
位 10	**CTSIE**：CTS 中断使能。 0：禁止中断 1：如果 LPUART_ISR 寄存器中 CTSIF=1，则会生成中断
位 9	**CTSE**：使能 CTS。 0：禁止 CTS 硬件流控制 1：使能 CTS 模式，仅当 nCTS 输入有效（连接到 0）时才发送数据。如果在发送数据时使 nCTS 输入无效，会在停止之前完成发送。如果使 nCTS 无效时数据已写入数据寄存器，则将延迟发送，直到 nCTS 有效。 只有在禁止 LPUART（UE=0）时才能写入此位。
位 8	**RTSE**：使能 RTS。 0：禁止 RTS 硬件流控制 1：使能 RTS 输出，仅当接收缓冲区中有空间时才会请求数据。发送完当前字符后应停止发送数据。可以接收数据时使 nRTS 输出有效（拉至 0）。 只有在禁止 LPUART（UE=0）时才能写入此位。
位 7	**DMAT**：DMA 使能发射器。该位由软件置 1/复位。 1：针对发送使能 DMA 模式 0：针对发送禁止 DMA 模式
位 6	**DMAR**：DMA 使能接收器。该位由软件置 1/复位。 1：针对接收使能 DMA 模式 0：针对接收禁止 DMA 模式
位 5:4	保留，必须保持复位值。
位 3	**HDSEL**：半双工选择。 单线半双工模式的选择。 0：未选择半双工模式 1：选择半双工模式 只有在禁止 LPUART（UE=0）时才能写入此位。
位 2:1	保留，必须保持复位值。
位 0	**EIE**：错误中断使能。 如果发生数据拆包错误、上溢错误或出现噪声标志（LPUART_ISR 寄存器中 FE = 1 或 ORE = 1 或 NF = 1），则需要使用错误中断使能位来使能中断生成。 0：禁止中断 1：当 LPUART_ISR 寄存器中的 FE=1 或 ORE=1 或 NF=1 时，生成中断

21.5.4 波特率寄存器（LPUART_BRR）

只有在禁止 LPUART（UE=0）时才能写入此寄存器。在自动波特率检测模式下，该位由硬件自动更新。

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.	BRR[19:16]			
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:20	保留，必须保持复位值。
位 19:0	**BRR[19:0]**

注意： *禁止在 LPUART_BRR 寄存器中写入小于 0x300 的值。在 LPUART_BRR 必须 > = 0x300 且 LPUART_BRR 为 20 位的前提下，在使用高 fck 值生成高波特率时应注意，fck 必须处于范围[3 x 波特率 ..4096 x 波特率]以内。*

21.5.5 请求寄存器（LPUART_RQR）

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	
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXFRQ	RXFRQ	MMRQ	SBKRQ	Res.
											w	w	w	w	

位 31:5	保留，必须保持复位值。
位 4	**TXFRQ**：发送数据刷新请求。 此位在 FIFO 模式使能时使用。TXFRQ 位置 1 将刷新整个 FIFO。这会使标志 TXFE 置 1（TXFIFO 为空，LPUART_ISR 寄存器中的位 23）。 注意：在 FIFO 模式下，为确保无任何数据被写入数据寄存器，会在刷新请求期间复位 TXFNF 标志，直至 TxFIFO 为空。
位 3	**RXFRQ**：接收数据刷新请求。 向该位写入 1 时会将 RXNE 标志清零。 如此可在不读取已接收数据的情况下丢弃这些数据，并避免上溢状态。
位 2	**MMRQ**：静默模式请求。 向此位写入 1 可将 LPUART 置于静默模式，并将 RWU 标志置 1。
位 1	**SBKRQ**：发送断路请求。 向此位写入 1 可将 SBKF 标志置 1 并在发送设备可用后立即请求在线路上发送 BREAK。 注意：这种情况下，应用需要在之前插入的所有数据（包括尚未发送的数据）后发送中断字符，软件应等到 TXE 标志使能后将 SBKRQ 位置 1。
位 0	保留，必须保持复位值。

21.5.6 中断和状态寄存器（LPUART_ISR）

偏移地址：0x1C

复位值：0x00C0（FIFO 禁用时）

复位值：0x08000C0（FIFO 使能时）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TXFT	RXFT	Res.	RXFF	TXFE	RE ACK	TE ACK	Res.	RWU	SBKF	CMF	BUSY
				r	r		r	r	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.	CTS	CTSIF	Res.	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE
					r	r		r	r	r	r	r	r	r	r

位 31:28	保留，必须保持复位值。
位 27	**TXFT**：TXFIFO 阈值标志。 当 TXFIFO 达到 LPUARTx_CR3 寄存器的 TXFTCFG 位字段中的设定阈值（即 TXFIFO 包含 TXFTCFG 空位）时，此位由硬件置 1。如果 LPUART_CR3 寄存器中 TXFTIE 位 = 1（位 31），则会生成中断。 0：TXFIFO 未达到设定阈值 1：TXFIFO 达到设定阈值
位 26	**RXFT**：RXFIFO 阈值标志。 当 RXFIFO 达到 LPUARTx_CR3 寄存器的 RXFTCFG 位字段中的设定阈值（即接收 FIFO 包含 RXFTCFG 数据）时，此位由硬件置 1。如果 LPUART_CR3 寄存器中 RXFTIE 位 = 1（位 27），则会生成中断。 0：接收 FIFO 未达到设定阈值 1：接收 FIFO 达到设定阈值
位 25	保留，必须保持复位值。
位 24	**RXFF**：RXFIFO 已满。 当 RXFIFO 已满时，该位由硬件置 1。 如果 LPUART_CR1 寄存器中 RXFFIE 位 = 1，则会生成中断。 0：RXFIFO 未满 1：RXFIFO 已满
位 23	**TXFE**：TXFIFO 为空。 当 TXFIFO 为空时，该位由硬件置 1。当 TXFIFO 包含至少一个数据时，此标志清零。也可以通过将 LPUART_RQR 寄存器中的 TXFRQ （位 4）位置 1 将 TXFE 标志位清零。 如果 LPUART_CR1 寄存器中 TXFEIE 位 = 1（位 30），则会生成中断。 0：TXFIFO 不为空 1：TXFIFO 为空
位 22	**REACK**：接收使能应答标志。 LPUART 采用发送使能值时，通过硬件将此位置 1/复位。 它可用于确认在进入停止模式前 LPUART 已做好接收准备。 注意：如果 LPUART 不支持从停止模式唤醒特性，则此位保留并由硬件强制清零。
位 21	**TEACK**：发送使能应答标志。 LPUART 采用发送使能值时，通过硬件将此位置 1/复位。 通过写入 TE=0 生成空闲帧请求，然后在 LPUART_CR1 寄存器中写入 TE=1 以遵循 TE=0 最短周期时，可使用此位。

位 20	保留，必须保持复位值。
位 19	**RWU**：从静默模式唤醒接收器。 此位指示 LPUART 是否处于静默模式。当识别出唤醒/静默序列时，此位由硬件清零/置 1。通过 LPUART_CR1 寄存器中的 WAKE 位选择静默模式控制序列（地址或 IDLE）。 如果选择从 IDLE 模式唤醒，则此位只能由软件置位，即在 LPUART_RQR 寄存器中的 MMRQ 位写入 1。 0：接收器处于活动模式 1：接收器处于静音模式 注意：如果 LPUART 不支持从停止模式唤醒特性，则此位保留并由硬件强制清零。
位 18	**SBKF**：发送断路标志。 此位指示已请求发送断路字符。通过将 1 写入 LPUART_CR3 寄存器中的 SBKRQ 位，此位由软件置 1。此位在断路发送的停止位期间由硬件自动复位。 0：不发送断路字符 1：发送断路字符
位 17	**CMF**：字符匹配标志。 接收到由 ADD[7:0]定义的字符后由硬件将此位置 1。通过向 LPUART_ICR 寄存器中的 CMCF 写入 1，此位由软件清零。 如果 LPUART_CR1 寄存器中 CMIE=1，则会生成中断。 0：未检测到字符匹配 1：检测到字符匹配
位 16	**BUSY**：忙标志。 此位由硬件置 1 和复位。当 RX 线路上正在进行通信时，此位有效（成功检测到起始位）。在接收结束（成功或失败）时复位。 0：LPUART 空闲（无接收） 1：正在接收
位 15:11	保留，必须保持复位值。
位 10	**CTS**：CTS 标志。 此位由硬件置 1/复位。此位是对 nCTS 输入引脚的状态取反。 0：nCTS 线置 1 1：nCTS 线复位 注意：如果不支持硬件流控制功能，该位保留并由硬件强制清零。
位 9	**CTSIF**：CTS 中断标志。 如果 CTSE 位置 1，当 nCTS 输入切换时，此位由硬件置 1。通过向 LPUART_ICR 寄存器中的 CTSCF 位写入 1，此位由软件清零。 如果 LPUART_CR3 寄存器中 CTSIE=1，则会生成中断。 0：nCTS 状态线上未发生变化 1：nCTS 状态线上发生变化 注意：如果不支持硬件流控制功能，该位保留并由硬件强制清零。
位 8	保留，必须保持复位值。
位 7	**TXE/TXFNF**：发送数据寄存器为空/TXFIFO 未满。 当 FIFO 模式禁用时，如果 LPUARTx_TDR 寄存器的内容已传输到移位寄存器，则 TXE 由硬件置 1。通过对 LPUARTx_TDR 寄存器执行写入操作将该位清零。 当 FIFO 模式使能时，如果 TXFIFO 未满因而可将数据写入 LPUART_TDR，则 TXFNF 由硬件置 1。对 LPUART_TDR 的每次写操作都会将数据放入 TXFIFO。在 TXFIFO 装满之前，此标志保持置位。当 TXFIFO 已满时，此标志清零，表示无法将数据写入 LPUART_TDR。 注意：在刷新请求期间，TXFNF 保持复位，直至 TXFIFO 为空。在发送刷新请求（通过将 TXFRQ 位置 1）

	后，应先检查标志 TXFNF，然后再写入 TXFIFO。（TXFNF 和 TXFE 同时置 1）。 如果 LPUART_CR1 寄存器中 TXEIE/TXFNFIE 位 = 1，则会生成中断。 0：数据寄存器已满/发送 FIFO 已满 1：数据寄存器/发送 FIFO 未满 注意：单缓冲区发送期间使用该位。
位 6	**TC**：发送完成。 如果已完成对包含数据的帧的发送并且 TXE/TXFF 置 1，则该位由硬件置 1。如果 LPUART_CR1 寄存器中 TCIE=1，则会生成中断。通过向 LPUART_ICR 寄存器中的 TCCF 写入 1 或向 LPUART_TDR 寄存器执行写操作，此位由软件清零。 如果 LPUART_CR1 寄存器中 TCIE=1，则会生成中断。 0：传送未完成 1：传送已完成 注意：如果 TE 位复位且没有正在进行的发送，则 TC 位立即置 1。
位 5	**RXNE/RXFNE**：读取数据寄存器不为空/RXFIFO 不为空。 如果 LPUARTx_RDR 移位寄存器的内容已传输到 LPUARTx_RDR 寄存器，则 RXNE 位由硬件置 1。通过对 LPUARTx_RDR 寄存器执行读入操作将该位清零。也可以通过将 LPUARTx_RQR 寄存器中的 RXFRQ 位置 1 将 RXNE 标志位清零。如果 RXFIFO 不为空因而可从 LPUART_RDR 寄存器读取数据，则 RXFNE 位由硬件置 1。每次读取 LPUART_RDR 都将释放 RXFIFO 中的一个位置。当 RXFIFO 为空时，此位清零。 也可以通过将 LPUART_RQR 寄存器中的 RXFRQ 位置 1 将 RXNE/RXFNE 标志位清零。 如果 LPUART_CR1 寄存器中 RXNEIE/RXFNEIE=1，则会生成中断。 0：未接收到数据 1：已准备好读取接收到的数据
位 4	**IDLE**：检测到空闲线路。 检测到空闲线路时，该位由硬件置 1。如果 LPUART_CR1 寄存器中 IDLEIE=1，则会生成中断。通过向 LPUART_ICR 寄存器中的 IDLECF 写入 1，此位由软件清零。 0：未检测到空闲线路 1：检测到空闲线路 注意：直到 RXNE 位已置 1 时（即，当出现新的空闲线路时）IDLE 位才会被再次置 1。 使能静默模式（MME=1）后，如果 LPUART 未静默（RWU=0），则 IDLE 置 1，无论是否通过 WAKE 位选择了静默模式。如果 RWU=1，IDLE 不置 1。
位 3	**ORE**：上溢错误。 当移位寄存器中目前正在接收的数据准备就绪可以传输到 LPUARTx_RDR 寄存器且 RXNE=1（FIFO 模式使能时为 RXFF=1）时，此位由硬件置 1。通过向 LPUARTx_ICR 寄存器中的 ORECF 写入 1，此位由软件清零。 如果 LPUARTx_CR1 寄存器中 RXNEIE/ RXFNEIE=1 或 EIE=1，则生成中断。 0：无上溢错误 1：检测到上溢错误 注意：当此位置 1 时，LPUART_RDR 寄存器的内容不会丢失，但移位寄存器会被覆盖。当 EIE 位置 1 时，如果在多缓冲区通信期间 ORE 标志置 1，则生成中断。 当 LPUART_CR3 寄存器中的 OVRDIS 位置 1 时，此位永久强制为 0（无上溢检测）。
位 2	**NF**：START 位噪声检测标志。 当在接收的帧上 START 位检测到噪声时，该位由硬件置 1。通过向 LPUART_ICR 寄存器中的 NFCF 位写入 1，此位由软件清零。 0：未检测到噪声 1：检测到噪声 注意：该位不会生成中断，因为该位出现的时间与本身生成中断的 RXNE/RXFNE 位出现的时间相同。当 EIE

	位置 1 时，如果在多缓冲区通信期间 NF 标志置 1，则生成中断。 注意：在 FIFO 模式下，此错误与 LPUART_RDR 中的字符有关。
位 1	**FE**：数据拆包错误。 当检测到去同步化、过度的噪声或中断字符时，该位由硬件置 1。通过向 LPUART_ICR 寄存器中的 FECF 位写入 1，此位由软件清零。在智能卡模式下发送期间，如果在达到最大发送尝试次数后仍未成功（智能卡向数据帧发送 NACK 信号），则此位置 1。 如果 LPUART_CR1 寄存器中 EIE=1，则会生成中断。 0：未检测到数据拆包错误 1：检测到数据拆包错误或中断字符 注意：在 FIFO 模式下，此错误与 LPUART_RDR 中的字符有关。
位 0	**PE**：奇偶校验错误。 当在接收器模式下发生奇偶校验错误时，该位由硬件置 1。通过向 LPUART_ICR 寄存器中的 PECF 写入 1，此位由软件清零。 如果 LPUART_CR1 寄存器中 PEIE=1，则会生成中断。 0：无奇偶校验错误 1：奇偶校验错误 注意：在 FIFO 模式下，此错误与 LPUART_RDR 中的字符有关。

21.5.7 中断标志清零寄存器（LPUART_ICR）

偏移地址：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.	Res.	Res.	Res.	CTSCF	Res.	Res.	TCCF	Res.	IDLECF	ORECF	NECF	FECF	PECF
						w_r0			w_r0		w_r0	w_r0	w_r0	w_r0	w_r0

位 31:21	保留，必须保持复位值。
位 20	**WUCF**：从停止模式唤醒清零标志。将 1 写入此位时，LPUART_ISR 寄存器中 WUF 标志将清零。 注意：如果 LPUART 不支持从停止模式唤醒特性，则此位保留并由硬件强制清零。
位 19:18	保留，必须保持复位值。
位 17	**CMCF**：字符匹配清零标志。 将 1 写入此位时，LPUART_ISR 寄存器中 CMF 标志将清零。
位 16:10	保留，必须保持复位值。
位 9	**CTSCF**：CTS 清零标志。 将 1 写入此位时，LPUART_ISR 寄存器中 CTSIF 标志将清零。
位 8:7	保留，必须保持复位值。
位 7	保留，必须保持复位值。

位 6	**TCCF**：发送完成清零标志。将 1 写入此位时，LPUART_ISR 寄存器中 CTSIF 标志将清零。
位 5	保留，必须保持复位值。
位 4	**IDLECF**：检测到空闲线路清零标志。将 1 写入此位时，LPUART_ISR 寄存器中 IDLE 标志将清零。
位 3	**ORECF**：上溢错误清零标志。将 1 写入此位时，LPUART_ISR 寄存器中 ORE 标志将清零。
位 2	**NECF**：检测到噪声清零标志。将 1 写入此位时，LPUART_ISR 寄存器中 NF 标志将清零。
位 1	**FECF**：数据拆包错误清零标志。将 1 写入此位时，LPUART_ISR 寄存器中 FE 标志将清零。
位 0	**PECF**：奇偶校验错误清零标志。将 1 写入此位时，LPUART_ISR 寄存器中 PE 标志将清零。

21.5.8 接收数据寄存器（LPUART_RDR）

偏移地址：0x24

复位值：未定义

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，必须保持复位值。
位 8:0	**RDR[8:0]**：接收数据值包含接收到的数据字符。 RDR 寄存器在输入移位寄存器和内部总线之间提供了并行接口（参见图 113. USART 框图）。 在使能奇偶校验位的情况下进行接收时，从 MSB 位中读取的值为接收到的奇偶校验位。

21.5.9 发送数据寄存器（LPUART_TDR）

偏移地址：0x28

复位值：未定义

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，必须保持复位值。
位 8:0	**TDR[8:0]**：发送数据值包含要发送的数据字符。 TDR 寄存器在内部总线和输出移位寄存器之间提供了并行接口（参见图 113. USART 框图）。 在使能奇偶校验位的情况下（LPUART_CR1 寄存器中的 PCE 位被置 1）进行发送时，由于 MSB 的写入值（位 7 或位 8，具体取决于数据长度）会被奇偶校验位所取代，因此该值不起任何作用。 *注意：该寄存器只能在 TXE/TXFNF=1 时写入。*

21.5.10 预分频器寄存器（LPUART_PRESC）

只有在禁止 USART（UE=0）时才能写入此寄存器。

偏移地址：0x2C

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.	Res.	Res.	Res.	Res.	Res.	PRESCALER[3:0]			
												rw	rw	rw	rw

位 31:4	保留，必须保持复位值。
位 3:0	**PRESCALER[3:0]**：时钟预分频器。 USART 输入时钟可通过预分频器进行分频： 0000：输入时钟不分频 0001：输入时钟 2 分频 0010：输入时钟 4 分频 0011：输入时钟 6 分频 0100：输入时钟 8 分频 0101：输入时钟 10 分频 0110：输入时钟 12 分频 0111：输入时钟 16 分频 1000：输入时钟 32 分频 1001：输入时钟 64 分频 1010：输入时钟 128 分频 1011：输入时钟 256 分频。 剩余组合：保留。 *注意：在用允许值以外的值对 PRESCALER 进行编程时，预分频因子的编程值为 «1011»，即输入时钟的 256 分频。*

21.6 LPUART 寄存器映射

下表提供了 LPUART 寄存器映射和复位值。

表 75. LPUART 寄存器映射和复位值

偏移	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
0x00	LPUART_CR1	RXFFIE	TXFEIE	FIFOEN	M1	Res.	Res.	DEAT[4:0]					DEDT[4:0]					Res.	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	Res.	UE
	复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				

偏移	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
0x04	LPUART_CR2	ADDI[7:4]				ADDI[3:0]				Res.	Res.	Res.	Res.	MSBFIRST	DATAINV	TXINV	RXINV	SWAP	Res.	STOP [1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDM7	Res.	Res.	Res.	Res.
	复位值	0	0	0	0	0	0	0	0					0	0	0	0	0		0	0								0				
0x08	LPUART_CR3	TXFTCFG[2:0]			RXFTIE	RXFTCFG[2:0]			Res.	TXFTIE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DEP	DEM	DDRE	OVRDIS	Res.	CTSIE	CTSE	RTSE	DMAT	DMAR	Res.	Res.	HDSEL	Res.	Res.	EIE
	复位值	0	0	0	0	0	0	0		0								0	0	0	0		0	0	0	0	0			0			0
0x0C	LPUART_BRR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BRR[19:0]																			
	复位值																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10-0x14	保留																																
0x18	LPUART_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.	TXFRO	RXFRO	MMRO	SBKRQ	Res.
	复位值																												0	0	0	0	
0x1C	LPUART_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REACK	TEACK	WUF	RWU	SBKF	CMF	BUSY	Res.	Res.	Res.	Res.	Res.	CTS	CTSIF	Res.	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE
	复位值										0	0	0	0	0	0	0						0	0		1	0	0	0	0	0	0	0
0x20	LPUART_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUCF	Res.	Res.	CMCF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTSCF	Res.	Res.	TCCF	Res.	IDLECF	ORECF	NECF	FECF	PECF
	复位值												0			0								0			0		0	0	0	0	0
0x24	LPUART_RDR	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.	RDR[8:0]						
	复位值																									0	0	0	0	0	0	0	0
0x28	LPUART_TDR	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.	TDR[8:0]						
	复位值																									0	0	0	0	0	0	0	0
0x2C	LPUART_PRESC	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.	PRESCALER[3:0]			

偏移	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
0x2C	LPUART_PRESC																													0	0	0	0

请参见“表 3. BlueNRG-LP 存储器映射和外设寄存器边界地址”。

22 串行外设接口/集成电路内置音频总线（SPI/I2S）

在 BlueNRG-LP 器件中，只有 SPI2 和 SPI3 支持除 SPI 特性外的 I^2S 协议。SPI1 不支持 I^2S。

22.1 引言

SPI/I2S 接口可用于使用 SPI 协议或 I^2S 音频协议与外部器件通信。可通过软件选择 SPI 或 I^2S 模式。器件复位后默认选择 SPI Motorola 模式。

串行外设接口（SPI）协议支持与外部器件进行半双工、全双工和单工同步串行通信。该接口可配置为主模式，在这种情况下，它可为外部从设备提供通信时钟（SCK）。该接口还能够在多主模式配置下工作。

集成电路内置音频总线（I^2S）协议还是同步串行通信接口。它可以在从或主模式下工作，进行半双工通信。它可以处理四种不同的音频标准，包括 Philips I2S 标准、MSB 和 LSB 对齐标准以及 PCM 标准。

22.2 SPI 主要特性

- 主模式或从模式操作
- 基于三条线的全双工同步传输
- 基于两条线（有双向数据线）的半双工同步传输
- 基于两条线（有单向数据线）的单工同步传输
- 数据大小可选择 4 位到 16 位
- 多主模式功能
- 最高 fPCLK/2 的 8 个主模式波特率预分频因子
- 从模式频率最高 fPCLK/2
- 对于主模式和从模式都可通过硬件或软件进行 NSS 管理：动态切换主/从操作
- 可编程时钟极性和相位
- 可编程的数据顺序，最先移位 MSB 或 LSB
- 可触发中断的专用发送和接收标志
- SPI 总线忙状态标志
- 支持 SPI Motorola 模式
- 用于确保可靠通信的硬件 CRC 功能：
  - 在发送模式下可将 CRC 值作为最后一个字节发送
  - 根据收到的最后一个字节自动进行 CRC 错误校验
- 可触发中断的主模式故障和上溢标志
- CRC 错误
- 具有 DMA 功能的两个 32 位内置 Rx 和 Tx FIFO
- 支持 SPI TI 模式。

22.3 I2S 主要特性

- 半双工通信（仅发射器或接收器）
- 主模式或从模式操作
- 8 位可编程线性预分频器，可实现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
- 数据包帧由音频通道固定为 16 位（可容纳 16 位数据帧）或 32 位（可容纳 16 位、24 位、32 位数据帧）

- 可编程的时钟极性（稳定状态）
- 从发送模式下的下溢标志、接收模式下的上溢标志（主模式和从模式），以及接收和发送模式下的帧错误标志（仅从模式）
- 发送和接收使用同一个 16 位数据寄存器
- 支持的 I^2S 协议：
  - I^2S Philips 标准
  - MSB 对齐标准（左对齐）
  - LSB 对齐标准（右对齐）
  - PCM 标准（在 16 位通道帧或扩展为 32 位通道帧的 16 位数据帧上进行短帧和长帧同步）
- 数据方向始终为 MSB 在前
- 用于发送和接收的 DMA 功能（16 位宽）
- 可输出主时钟以驱动外部音频元件。比率固定为 256 × FS （其中 FS 为音频采样频率）。

22.4 SPI/I2S 实现

本手册描述了 SPI1、SPI2 和 SPI3 中实现的特性的完整集合。

表 76. BlueNRG-LP SPI 实现描述了 BlueNRG-LP 器件中的 SPI/I2S 实现。

表 76. BlueNRG-LP SPI 实现

SPI 特性	SPI1	SPI2	SPI3
硬件 CRC 计算	X	X	X
Rx/Tx FIFO	X	X	X
NSS 脉冲模式	X	X	X
I2S 模式	-	X	X
TI 模式	X	X	X

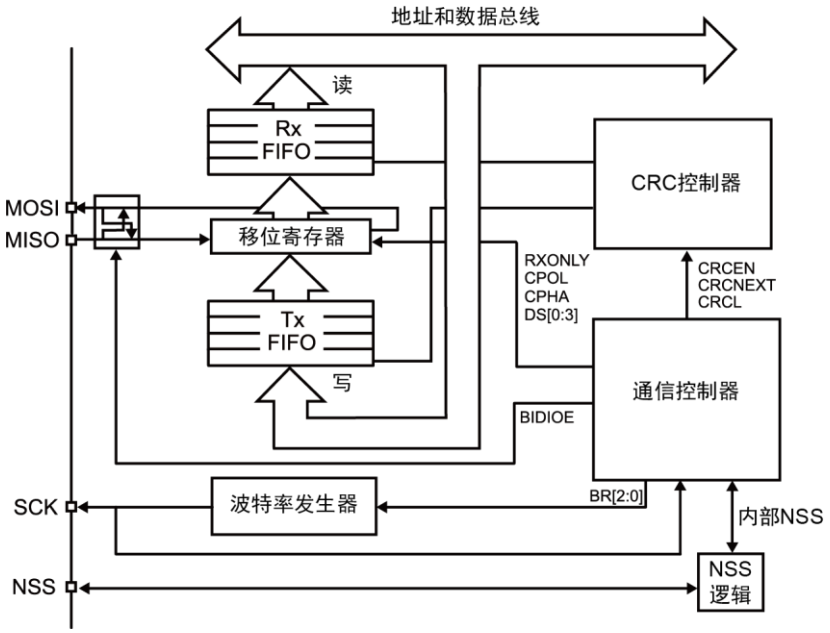
注意： *X = 支持。*

22.5 SPI 功能说明

22.5.1 概述

SPI 支持在 MCU 与外部器件之间进行同步串行通信。应用软件可通过轮询状态标志或使用专用 SPI 中断对通信进行管理。SPI 的主要组件及其交互方式如以下框图所示 。

图 150. SPI 框图



四个 I/O 引脚专用于与外部器件进行 SPI 通信。

- **MISO**：主输入/从输出数据。通常情况下，此引脚用于在从模式下发送数据和在主模式下接收数据。
- **MOSI**：主输出/从输入数据。通常情况下，此引脚用于在主模式下发送数据和在从模式下接收数据。
- **SCK**：SPI 主设备的串行时钟输出引脚和 SPI 从设备的输入引脚。

注意：如果 SPI 处于主模式且使用了焊盘的内部上/下拉，则软件必须谨慎地激活 I/O 的上/下拉极性，以便与 CPOL 编程保持一致（CPOL=0 时下拉，CPOL=1 时上拉）。

- **NSS**：从设备选择引脚。根据 SPI 和 NSS 设置，该引脚可用于：
  - 选择单个从设备以进行通信
  - 同步数据帧或
  - 检测多个主设备之间是否存在冲突

参见第 22.5.4 节“从设备选择（NSS）引脚管理”了解详情。

SPI 总线支持一个主设备与一个或多个从设备之间进行通信。该总线至少由两条线构成 - 一条用于时钟信号，另一条用于同步数据传输。其它信号可以根据 SPI 节点间的数据交换及其从设备选择信号管理进行添加。

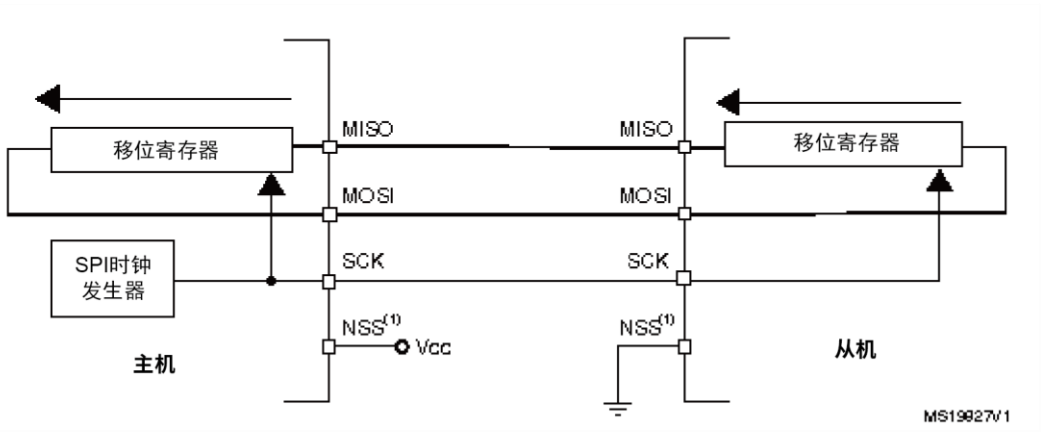
22.5.2 一个主设备和一个从设备之间的通信

SPI 支持 MCU 基于目标器件和应用要求使用不同的配置进行通信。这些配置使用 2 条或 3 条线（通过软件 NSS 管理），也可以使用 3 条或 4 条线（通过硬件 NSS 管理）。通信始终由主设备发起。

全双工通信

默认情况下，SPI 配置为进行全双工通信。在这种配置下，主设备和从设备的移位寄存器通过 MOSI 和 MISO 引脚之间的两条单向线连接。在 SPI 通信过程中，数据随主设备提供的 SCK 时钟边沿同步移位。主设备通过 MOSI 线将待发送的数据发送给从设备，通过 MISO 线经由从设备接收数据。当数据帧传输完成时（所有位均移出），主设备和从设备之间即完成信息交换。

图 151. 全双工单个主设备/单个从设备应用



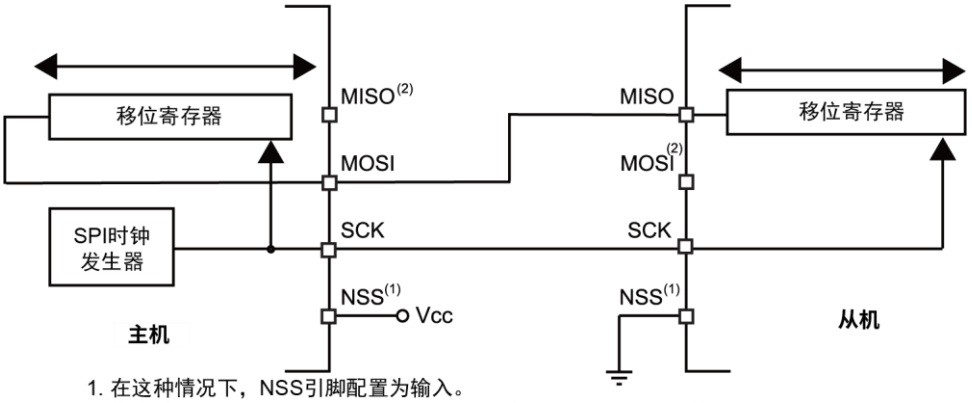
1. 在这种情况下，NSS引脚配置为输入。

1) 在这种情况下，NSS 引脚配置为输入。

半双工通信

通过将 SPIx_CR1 寄存器的 BIDIMODE 位置 1，SPI 可采用半双工模式进行通信。在这种配置下，使用一条交叉连接线将主设备和从设备的移位寄存器连接起来。在此通信过程中，数据随 SCK 时钟边沿在移位寄存器之间进行移位，传输方向由主设备和从设备通过各自 SPIx_CR1 寄存器中的 BDIOE 位进行选择。在这种配置下，主设备的 MISO 引脚和从设备的 MOSI 引脚空闲，可在其它应用中用作 GPIO。

图 152. 半双工单个主设备/单个从设备应用



1. 在这种情况下，NSS引脚配置为输入。
2. 在这种配置下，主设备的MISO引脚和从设备的MOSI引脚可用作GPIO。

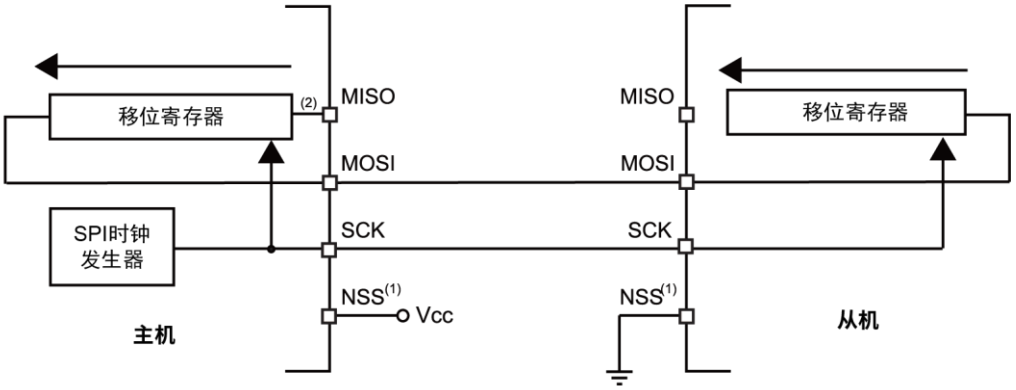
单工通信

通过 SPIx_CR2 寄存器中的 RXONLY 位将 SPI 设置为只发送模式或只接收模式，可使 SPI 以单工模式进行通信。在这种配置下，仅使用一条线在主设备和从设备的移位寄存器之间进行传输。其余 MISO 和 MOSI 引脚对不用于通信，可用作标准 GPIO。

- **只发送模式（RXONLY=0）**：配置设置与全双工相同。应用必须忽略在未使用的输入引脚上捕获的信息。该引脚可以用作标准 GPIO。

- **只接收模式（RXONLY=1）**：应用程序可通过将 RXONLY 位置 1 来禁用 SPI 输出功能。在从设备配置下，MISO 输出禁用且引脚可用作 GPIO。从设备在其从设备选择信号有效时继续接收来自 MOSI 引脚的数据（参见第 22.5.4 节“从设备选择（NSS）引脚管理”）。基于数据缓冲区的配置产生接收数据事件。在主设备配置下，MOSI 输出禁用且引脚可用作 GPIO。只要 SPI 使能，就会连续生成时钟信号。停止时钟的唯一方式是将 RXONLY 位或 SPE 位清零，直至来自 MISO 引脚的传入模式结束，然后基于相应配置填充数据缓冲区结构。

图 153. 单工单一主设备/单一从设备应用（在只发送模式下为主设备/在只接收模式下为从设备）



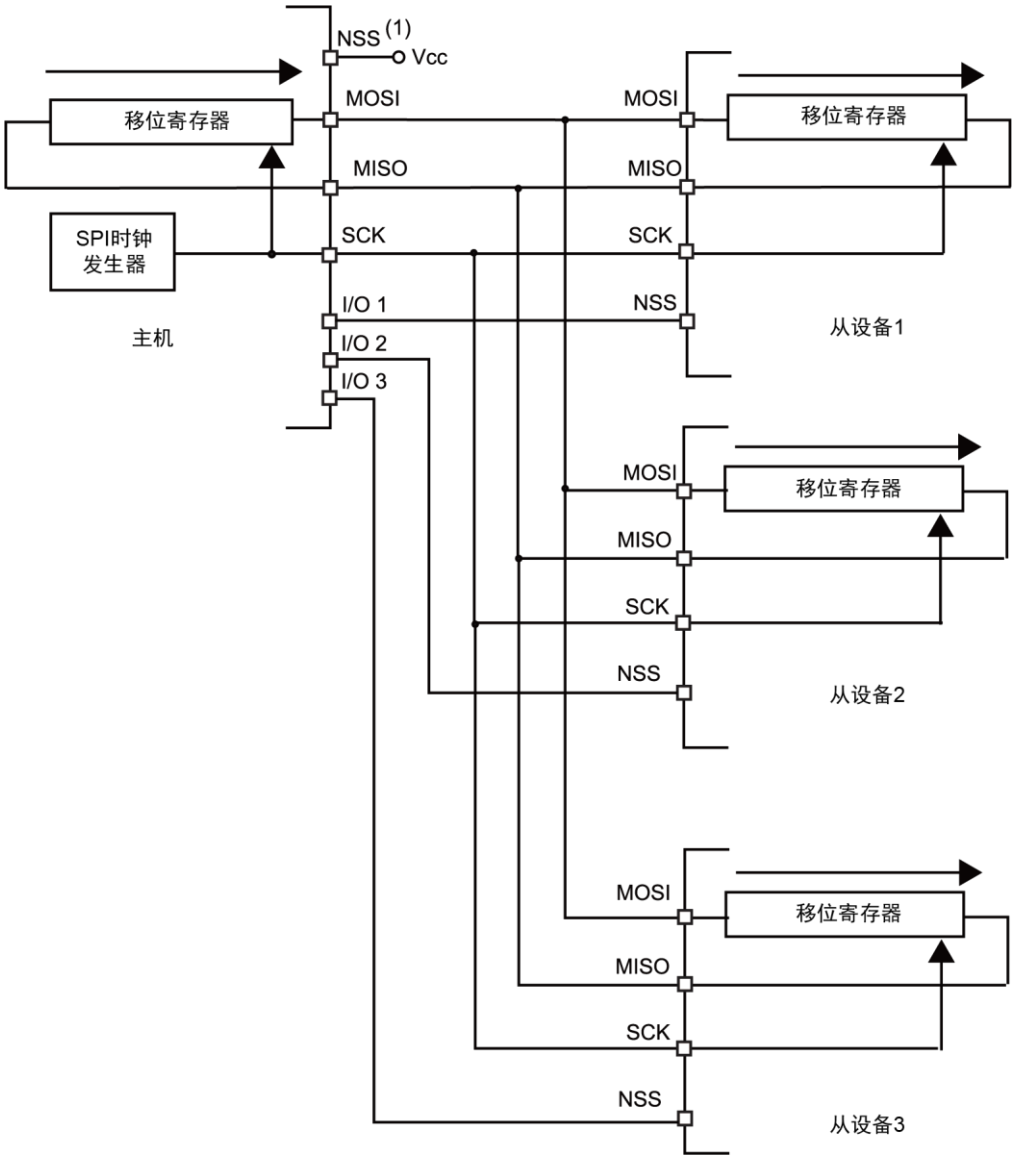
1. 在这种情况下，NSS引脚配置为输入。
2. 输入信息在移位寄存器中捕获，在标准的只发送模式下必须忽略此信息（例如，OVF标志）
3. 在这种配置下，两个MISO引脚均可用作GPIO。

注意： *任何单工通信均可以通过半双工通信的一种变型来替换，该变型中设置的数据传输方向不变（在 BDIO 位保持不变的同时，双向模式处于使能状态）。*

22.5.3 标准多从设备通信

在有两个或更多独立从设备的配置下，主设备使用 GPIO 引脚管理每个从设备的片选线（参见图 154. 主设备和三个独立从设备）。主设备必须通过拉低与从设备 NSS 输入相连的 GPIO 的电平来单独选择一个从设备。执行该操作后，便建立了标准主设备与专用从设备之间的通信。

图 154. 主设备和三个独立从设备



1. 由于从设备的 MISO 引脚连在一起，所有从设备 MISO 引脚的 GPIO 配置必须设置为复用功能开漏（请参见表 7. GPIO 复用选项 AF3 - AF4）。

22.5.4 从设备选择（NSS）引脚管理

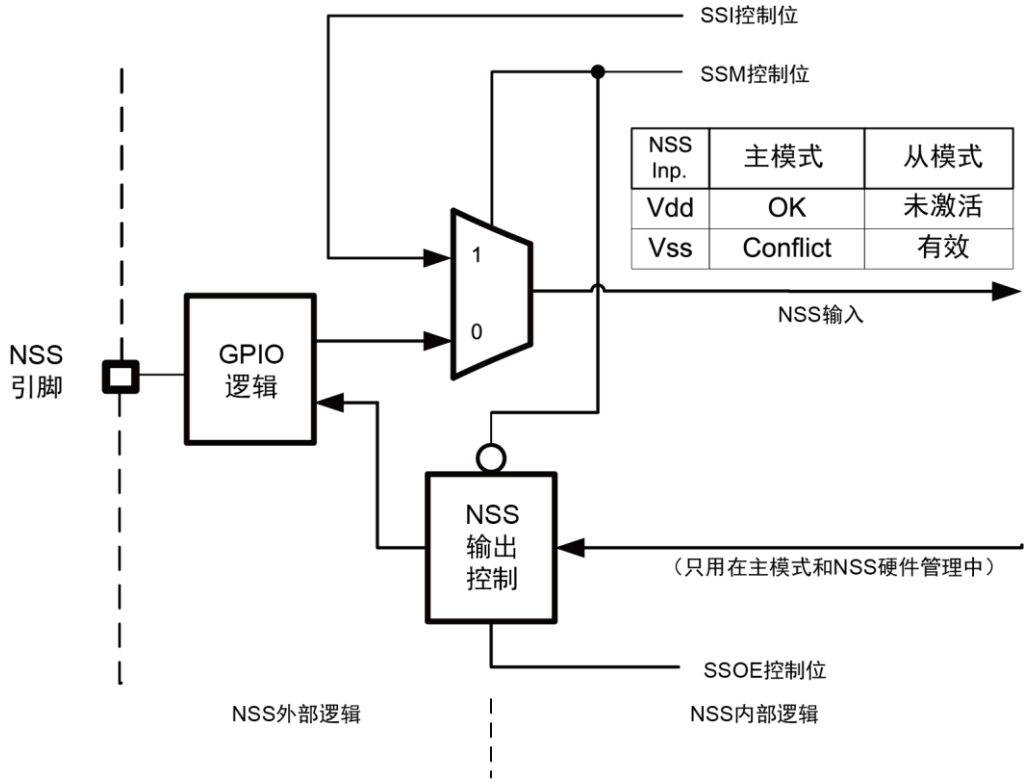
在从模式下，NSS 用作标准的“片选”输入，使从设备与主设备进行通信。在主模式下，NSS 可用作输出或输入。用作输入时，可防止多主模式总线冲突；用作输出时，可驱动单个从设备的从设备选择信号。

可以使用 SPIx_CR1 寄存器中的 SSM 位设置硬件或软件从设备选择管理：

- **软件 NSS 管理（SSM = 1）**：在该配置下，从设备选择信息由 SPIx_CR1 寄存器中 SSI 位的值内部驱动。外部 NSS 引脚空闲，可供其它应用使用。

- **硬件 NSS 管理（SSM = 0）**：在这种情况下，可行的配置有两种。所用配置取决于 NSS 输出配置（SPIx_CR1 寄存器中的 SSOE 位）。
  - NSS 输出使能（SSM=0 且 SSOE = 1）：仅在将 MCU 设置为主设备时才使用该配置。NSS 引脚由硬件管理。只要在主模式下使能 SPI（SPE=1），NSS 信号便会被驱动为低电平，并且会一直保持低电平状态，直至禁止 SPI（SPE =0）。如果激活 NSS 脉冲模式（NSSP=1），连续通信间会生成脉冲。SPI 无法在采用此 NSS 设置的多主模式配置下工作。
  - NSS 输出禁止（SSM=0 且 SSOE = 0）：如果微控制器在总线上用作主设备，此配置可实现多主模式功能。如果在该模式下将 NSS 引脚拉至低电平，SPI 将进入主模式故障状态，器件将在从模式下自动进行重新配置。在从模式下，NSS 引脚用作标准的“片选”输入，当 NSS 线为低电平时将选择从设备。

图 155. 硬件/软件从设备选择管理



22.5.5 通信格式

SPI 通信过程中，将同时执行接收和发送操作。串行时钟（SCK）对数据线上的信息的移位和采样进行同步。通信格式取决于时钟相位、时钟极性和数据帧格式。为了能够在彼此间进行通信，主设备和从设备必须遵循相同的通信格式。

时钟相位和极性控制

通过 SPIx_CR1 寄存器中的 CPOL 和 CPHA 位，可以用软件选择四种可能的时序关系。CPOL（时钟极性）位控制不传输任何数据时时钟的空闲状态值。此位对主设备和从设备都有作用。如果复位 CPOL，SCK 引脚在空闲状态处于低电平。如果将 CPOL 置 1，SCK 引脚在空闲状态处于高电平。

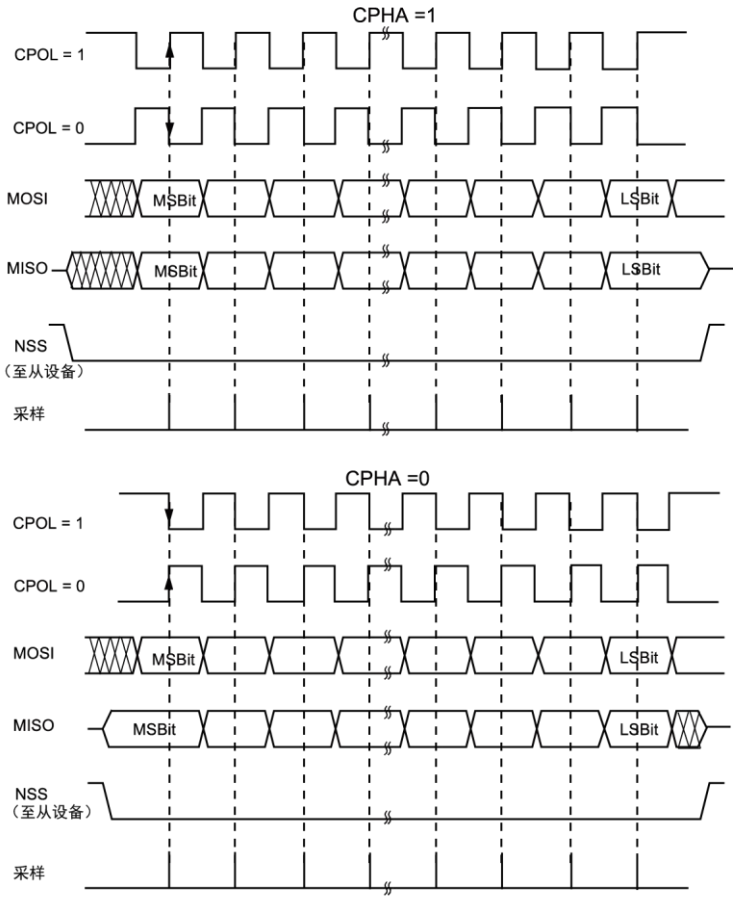
如果将 CPHA 位置 1，则会在 SCK 引脚的第二个边沿捕获传输的第一个数据位（如果复位 CPOL 位，则为下降沿；如果将 CPOL 位置 1，则为上升沿）。即，在每次出现该时钟边沿时锁存数据。如果将 CPHA 位复位，则会在 SCK 引脚的第一个边沿捕获传输的第一个数据位（如果将 CPOL 位置 1，则为下降沿；如果将 CPOL 位复位，则为上升沿）。即，在每次出现该时钟边沿时锁存数据。

CPOL（时钟极性）和 CPHA（时钟相位）位的组合用于选择数据捕获时钟边沿。

图 156. 数据时钟时序图用 CPHA 和 CPOL 位的四种组合展示了 SPI 全双工传输。

注意： *在切换 CPOL/CPHA 位之前，必须通过复位 SPE 位来禁止 SPI。SCK 的空闲状态必须与 SPIx_CR1 寄存器中选择的极性相对应（如果 CPOL=1，则上拉 SCK；如果 CPOL=0，则下拉 SCK）。*

图 156. 数据时钟时序图



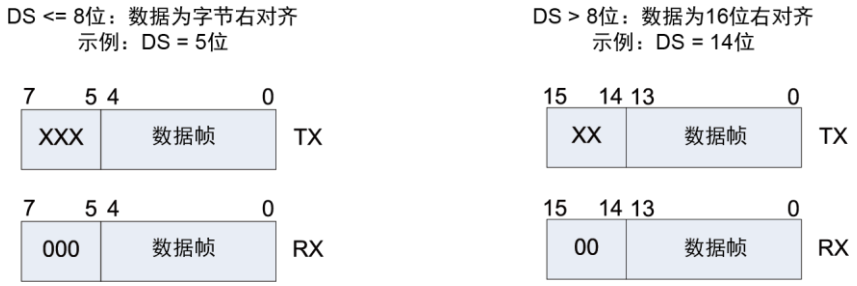
1. 数据位的顺序取决于LSBFIRST位的设置。

数据帧格式

SPI 移位寄存器可设置为以 MSB 在前或 LSB 在前的方式移出数据，具体取决于 LSBFIRST 位的值。数据帧的长度通过 DS 位进行选择。数据帧的长度可设置为 4 位到 16 位，且此设置对于发送和接收均适用。

无论所选的数据帧长度为何，必须依照 FRXTH 电平对 FIFO 进行读访问。访问 SPIx_DR 寄存器时，数据帧始终按字节（数据不超过一个字节时）或半字进行右对齐（请参见图 157. 数据长度不等于 8 位或 16 位时的数据对齐。在通信过程中，只会为数据帧内的位提供时钟并传输这些位。

图 157. 数据长度不等于 8 位或 16 位时的数据对齐



注意： *数据长度至少为 4 位。如果所选的数据长度不足 4 位，则会将数据帧长度强制为 8 位。*

22.5.6 SPI 配置

主设备和从设备的配置步骤几乎相同。关于特定的模式设置，请参见专门的章节。若要对标准通信进行初始化，请执行以下步骤：

1. 对适当的 GPIO 寄存器执行写操作：将 MOSI、MISO 和 SCK 引脚配置为 GPIO。
2. 对 SPI_CR1 寄存器执行写操作：
   a. 通过 BR[2:0]位配置串行时钟波特率(1)。
   b. 配置 CPOL 和 CPHA 位的组合，定义数据传输和串行时钟之间的关系（四种关系中的一种）（NSSP 模式下必须将 CPHA 清零）(2)。
   c. 通过配置 RXONLY 或 BIDIMODE 和 BIDIOE 来选择单工或半双工模式（RXONLY 和 BIDIMODE 不可同时置 1）。
   d. 配置 LSBFIRST 位以定义帧格式(2)。
   e. 如果需要 CRC，请配置 CRCL 和 CRCEN 位（SCK 时钟信号处于空闲状态时）。
   f. 配置 SSM 和 SSI(2)(3)。
   g. 配置 MSTR 位（在多主模式 NSS 配置下，如果主设备配置为防止发生 MODF 错误，则应避免 NSS 上出现状态冲突）。
3. 对 SPI_CR2 寄存器执行写操作：
   a. 配置 DS[3:0]位，选择传输的数据长度。
   b. 配置 SSOE(4)(2)(2)(3)。
   c. 如果需要 TI 协议，置位 FRF 位（在 TI 模式下使 NSSP 位保持清零状态）。
   d. 如果在两个数据单元之间需要 NSS 脉冲模式，置位 NSSP 位（NSSP 模式下将 CHPA 和 TI 位保持清零状态）。
   e. 配置 FRXTH 位。RXFIFO 阈值必须与 SPIx_DR 寄存器的读访问大小相符。
   f. 如果封装模式下使用 DMA，请初始化 LDMA_TX 和 LDMA_RX 位。
4. 对 SPI_CRCPR 寄存器执行写操作：配置 CRC 多项式（如果需要）。
5. 对合适的 DMA 寄存器执行写操作：如果使用 DMA 数据流，在 DMA 寄存器中配置专用于 SPI Tx 和 Rx 的 DMA 数据流。

注意：

1. *从模式下无需此步骤，但从设备在 TI 模式下工作时除外。*
2. *TI 模式下无需此步骤。*
3. *NSSP 模式下无需此步骤。*
4. *从模式下无需此步骤。*

22.5.7 使能 SPI 的步骤

建议在主设备发送时钟前使能 SPI 从设备。否则，数据传输可能会不正常。从设备的数据寄存器必须包含待发送的数据才能开始与主设备通信（在通信时钟的第一个边沿；如果时钟信号连续，则是在正在进行的通信结束前）。使能 SPI 从设备前，SCK 信号必须稳定为所选极性对应的空闲状态电平。

当 SPI 处于使能状态，且 TXFIFO 不为空或者对 TXFIFO 执行下一次写操作时，全双工模式（或任何只发送模式）下的主设备开始通信。

在任何主设备只接收模式（RXONLY=1 或 BIDIMODE=1 且 BIDIOE=0）下，使能 SPI 后，主设备立即开始通信且时钟立即开始运行。

关于 DMA，请参见专门的章节。

22.5.8 数据发送和接收过程

RXFIFO 和 TXFIFO

所有 SPI 数据交互均经由 32 位内置 FIFO。这使得 SPI 能够以连续流工作，并能防止在数据帧长度较短时发生上溢。每个方向都有自己的 FIFO，称为 TXFIFO 和 RXFIFO。这些 FIFO 用在所有 SPI 模式下，但使能了 CRC 计算的只接收器模式（从设备或主设备）除外（参见第 22.5.13 节“CRC 计算”）。

FIFO 的处理取决于数据交换模式（双工、单工）、数据帧格式（帧中的位数）、在 FIFO 数据寄存器上执行的访问大小（8 位或 16 位）以及在访问 FIFO 时是否使用了数据封装（参见第 22.5.12 节“TI 模式”）。

对 SPIx_DR 寄存器执行读访问时，会返回尚未读取的 RXFIFO 中存储的最早的值。对 SPIx_DR 执行写访问时，会在发送队列结束时将写入的数据存储到 TXFIFO 中。读访问必须始终符合由 SPIx_CR2 寄存器中的 FRXTH 位配置的 RXFIFO 阈值。FTLVL[1:0]和 FRLVL[1:0]位用于指示两个 FIFO 当前的占用水平。

SPIx_DR 寄存器的读访问必须通过 RXNE 事件进行管理。当数据存储到 RXFIFO 中且达到阈值（由 FRXTH 位定义）时会触发该事件。当 RXNE 清零时，RXFIFO 被视为空。同样地，待发送数据帧的写访问通过 TXE 事件进行管理。当 TXFIFO 占用水平小于或等于其容量的一半时会触发该事件。否则，TXE 清零，TXFIFO 被视为已满。通过这种方式，当数据帧格式不超过 8 位时，RXFIFO 最多可存储四个数据帧，而 TXFIFO 最多只能存储三个数据帧。当软件尝试以 16 位模式将更多数据写入 TXFIFO 时，此差异防止已存储在 TXFIFO 中的 3x 8 位数据帧发生损坏。TXE 和 RXNE 事件均可通过中断来轮询或处理。见图 159. 主设备全双工通信至图 162. 封装模式下的主设备全双工通信。

另一种管理数据交换的方式是使用 DMA（参见第 10 节“DMA 控制器（DMA）”）。

如果在 RXFIFO 已满时接收到下一个数据，将发生上溢事件（参见第 22.5.9 节“SPI 状态标志”中关于 OVR 标志的说明）。上溢事件可通过中断来轮询或处理。

BSY 位置 1 表示当前数据帧的事务正在进行。当时钟信号连续运行时，在主设备中，BSY 标志在数据帧之间保持置 1 状态，但在从设备中，BSY 标志在数据帧传输之间变为低电平并持续最短的一段时间（一个 SPI 时钟）。

序列处理

可通过一个序列传送一些数据帧从而完成一条消息。使能发送后，序列即开始，只要主设备的 TXFIFO 中存在数据便一直继续。时钟信号由主设备持续提供，直至 TXFIFO 变为空，之后时钟信号停止，等待其它数据。

在只接收模式、半双工模式（BIDIMODE=1 且 BIDIOE=0）或单工模式（BIDIMODE=0 且 RXONLY=1）下，当使能 SPI 并激活只接收模式后，主设备将立即启动序列。时钟信号由主设备提供，且直至主设备禁止 SPI 或只接收模式才会停止。在此之前，主设备会连续接收数据帧。

当主设备能够以连续模式（SCK 信号连续）提供所有交互时，任何时候都必须根据从设备功能来处理数据流及其内容。如有必要，主设备必须降低通信速率，并以足够的延时提供更慢的时钟或单独的帧或数据会话。请注意，SPI 模式下不存在主设备或从设备的下溢错误信号，来自从设备的数据始终由主设备处理，即使从设备无法及时正确地准备数据也是如此。从设备最好使用 DMA，特别是在数据帧较短且总线速率高时。

在多从设备系统中，每个序列必须通过 NSS 脉冲进行控制，从而只选择其中一个从设备进行通信。在单个从设备系统中，无需通过 NSS 来控制从设备，但此时提供此脉冲通常会更好，以在每个数据序列开始时同步从设备。可通过软件和硬件管理 NSS（参见第 22.5.4 节“从设备选择（NSS）引脚管理”）。

当 BSY 位置 1 时，表示数据帧事务正在进行。当所进行的帧交互完成时，RXNE 标志将置 1。最后一位采样后，整个数据帧会存储到 RXFIFO 中。

禁用 SPI 的步骤

当禁止 SPI 时，必须按照本段中介绍的禁止步骤进行操作。当外设时钟停止时，在系统进入低功耗模式前做到这一点是十分重要的。在这种情况下，正在进行的事务可能发生损坏。在某些模式下，禁止步骤是停止所进行的连续通信的唯一方式。

当处于全双工或只发送模式下的主设备停止提供待发送的数据时，可结束任何交互。在这种情况下，时钟在最后一个数据传输后停止。在封装模式下处理奇数个数据帧时必须特别注意，以防止一些空字节交换（参见“数据封装”一节）。在这些模式下禁止 SPI 之前，用户必须按照标准的禁止步骤进行操作。在主发射器上禁止 SPI，而同时正在处理帧交互或已将下一个数据帧存储到 TXFIFO 中时，SPI 行为无法保证。

只要主设备处于只接收模式，停止连续时钟的唯一方式就是通过 SPE=0 来禁止外设。这必须在最后一个数据帧传输内的特定时间段，即第一位采样与最后一位传输开始之间完成（以便接收全部数量的预期数据帧并防止在最后一个有效数据帧后读取任何其它的“空”数据）。在该模式下禁止 SPI 时必须遵循特定步骤。

禁止 SPI 后，已接收但未读取的数据始终存储在 RXFIFO 中，这些数据必须在下次使能 SPI 后进行处理，然后才能启动新序列。为防止存在未读取的数据，需确保禁止 SPI 时 RXFIFO 为空，可通过正确的禁止步骤来禁止 SPI，也可以通过控制外设复位专用的特定寄存器以软件复位的方式来初始化所有 SPI 寄存器从而禁止 SPI（请参见 RCC_APBiRSTR 寄存器中的 SPIiRST 位）。

标准禁止步骤通过轮询 BSY 状态以及 FTLVL[1:0]来检查发送会话是否完全结束。还可以在必须识别正在处理的传输是否结束的特定情况下完成这种检查，例如：

- 当 NSS 信号由软件管理且主设备必须为从设备提供 NSS 脉冲结束时，或者
- 最后一个数据帧或 CRC 帧传输仍在外设总线中处理时，来自 DMA 或 FIFO 的传输数据流完成。

正确的禁止步骤如下（使用只接收模式时除外）：

1. 等待至 FTLVL[1:0] = 00（无需发送更多数据）。
2. 等待至 BSY=0（最后一个数据帧已处理完）。
3. 禁止 SPI（SPE = 0）。
4. 读取数据，直至 FRLVL[1:0] = 00（读取接收的所有数据）。

某些只接收模式的正确禁止步骤如下：

1. 当最后一个数据帧正在处理时，通过在特定时间窗口内禁止 SPI（SPE=0）来中断接收流。
2. 等待至 BSY=0（最后一个数据帧已处理完）。
3. 读取数据，直至 FRLVL[1:0] = 00（读取接收的所有数据）。

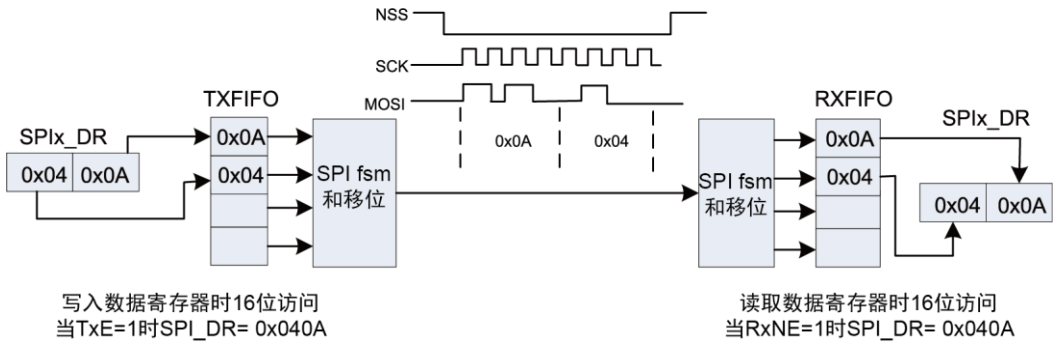
注意： *如果使用封装模式并且必须接收奇数数量的数据帧且数据帧格式为小于或等于 8 位（不超过一个字节），则 FRXTH 必须在 FRLVL[1:0] = 01 时置 1，以便生成 RXNE 事件从而读取最后的奇数编号数据帧并且使 FIFO 指针保持正确对齐。*

数据打包

若数据帧长度不足一个字节（小于或等于 8 位），当 SPIx_DR 寄存器上执行任何 16 位读写访问时将自动使用数据封装。在这种情况下将并行处理双数据帧模式。SPI 在工作时，首先使用存储在访问字的 LSB 中的模式，然后使用存储在 MSB 中的另一半。图 158. 将数据封装到 FIFO 中以进行发送和接收提供了数据封装模式序列处理的示例。在对发射器的 SPIx_DR 寄存器执行单次 16 位访问后发送两个数据帧。如果 RXFIFO 阈值设置为 16 位（FRXTH=0），该序列只会在接收器中生成一个 RXNE 事件。接收器随后必须通过对 SPIx_DR 执行单次 16 位读访问来访问这两个数据帧，从而响应该单个 RXNE 事件。RxFIFO 阈值设置和后续读访问必须始终与接收器侧保持一致，因为若不一致，则会丢失数据。

如果必须处理奇数数量的此类“不超过一个字节”的数据帧，则会出现特定问题。在发射器侧，只需将任意奇序列的最后一个数据帧以 8 位访问的方式写入 SPIx_DR 即可。接收器必须针对所接收的奇序列中的最后一个数据帧更改 Rx_FIFO 阈值大小，以生成 RXNE 事件。

图 158. 将数据封装到 FIFO 中以进行发送和接收



使用 DMA（直接存储器寻址）进行通信

为了以最大速度工作并且为了促进避免上溢所需的数据寄存器读/写过程，SPI 提供了 DMA 功能，该功能采用了简单的请求/应答协议。

将 SPIx_CR2 寄存器中的使能位 TXE 或 RXNE 置 1 时，将请求 DMA 访问。必须向发送缓冲区和接收缓冲区发出单独的请求。

- 在发送过程中，每次 TXE 位置 1 都会发出 DMA 请求。然后，DMA 将对 SPIx_DR 寄存器执行写操作。
- 在接收过程中，每次 RXNE 位置 1 都会发出 DMA 请求。然后，DMA 将对 SPIx_DR 寄存器执行读操作。

见图 159. 主设备全双工通信至图 162. 封装模式下的主设备全双工通信。

当 SPI 仅用于发送数据时，可以只使能 SPI Tx DMA 通道。在这种情况下，OVR 标志会置 1，因为未读取接收的数据。当 SPI 仅用于接收数据时，可以只使能 SPI Rx DMA 通道。

在发送模式下，DMA 写入所有要发送的数据（DMA_ISR 寄存器中的 TCIF 标志置 1）后，可以对 BSY 标志进行监视，以确保 SPI 通信已完成。在关闭 SPI 或进入停止模式前必须执行此步骤，以避免损坏最后一次数据的发送。软件必须首先等待 FTLVL[1:0]=00，再等待 BSY=0。

通过 DMA 开始通信时，为防止 DMA 通道管理引发错误事件，必须按顺序执行以下步骤：

1. 如果使用 DMA Rx，通过 SPI_CR2 寄存器中的 RXDMAEN 位来使能 DMA 接收缓冲区。
2. 如果使用数据流，通过 DMA 寄存器来使能 Tx 和 Rx 的 DMA 数据流。
3. 如果使用 DMA Tx，通过 SPI_CR2 寄存器中的 TXDMAEN 位来使能 DMA Tx 缓冲区。
4. 通过将 SPE 位置 1 使能 SPI。

要关闭通信，必须按顺序执行以下步骤：

1. 如果使用数据流，通过 DMA 寄存器来禁止 Tx 和 Rx 的 DMA 数据流。
2. 通过后续 SPI 禁止步骤来禁止 SPI。
3. 如果使用 DMA Tx 和/或 DMA Rx，通过将 SPI_CR2 寄存器中的 TXDMAEN 和 RXDMAEN 位清零来禁止 DMA 发送缓冲区和接收缓冲区。

使用 DMA 时的数据封装

如果由 DMA 管理传输（SPIx_CR2 寄存器中的 TXDMAEN 和 RXDMAEN 位置 1），则将根据为 SPI TX 和 SPI RX 的 DMA 通道配置的 PSIZE 值自动使能/禁止封装模式。如果 DMA 通道的 PSIZE 值等于 16 位，且 SPI 数据大小小于或等于 8 位，则将使能封装模式。然后，DMA 将自动管理对 SPIx_DR 寄存器的写操作。

如果使用数据封装模式，且要传输的数据数量不是 2 的倍数，则 LDMA_TX/LDMA_RX 位必须置 1。然后，对于最后一次 DMA 传输，SPI 只考虑进行一个数据的发送或接收（参见“数据封装”了解更多信息）。

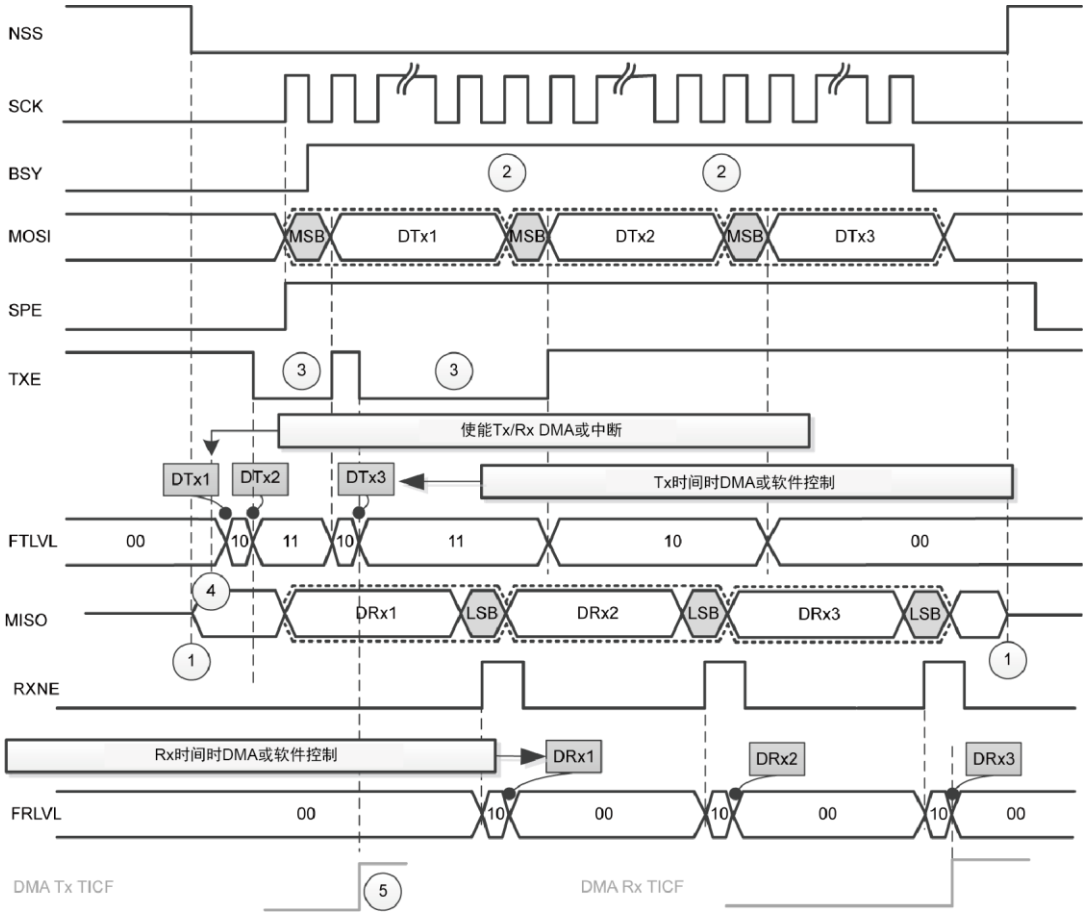
通信图

本部分将介绍一些典型的时序图。无论是通过上/下拉、中断还是 DMA 处理 SPI 事件，这些时序图都有效。为简单起见，此处使用设置 LSBFIRST=0、CPOL=0 和 CPHA=1 作为共同假设。不提供 DMA 数据流的完整配置。

以下带编号的注释同时适用于图 159. 主设备全双工通信至图 162. 封装模式下的主设备全双工通信。

1. 激活 NSS 并使能 SPI 后，从设备开始控制 MISO 线，而当其中一个条件不成立时，从设备将与 MISO 线断开。必须为从设备提供充足的时间，以便在传输开始前准备好主设备专用的数据。在主设备上，只有使能 SPI 后，SPI 外设才会控制 MOSI 和 SCK 信号（偶尔还会控制 NSS 信号）。如果禁止 SPI，SPI 外设会与 GPIO 逻辑断开，因此这些线上的电平只取决于 GPIO 设置。
2. 在主设备上，如果通信（时钟信号）连续，BSY 在数据帧之间保持有效。在从设备上，BSY 信号在数据帧之间始终会保持至少一个时钟周期的低电平状态。
3. 只有 TXFIFO 已满时，TXE 信号才会清零。
4. DMA 仲裁过程在 TXDMAEN 位置 1 后立即开始。TXE 中断在 TXEIE 置 1 后立即生成。TXE 信号处于有效电平时，开始向 TxFIFO 传输数据，直至 TxFIFO 已满或 DMA 传输完成。
5. 如果要发送的所有数据可装入 TxFIFO，则 DMA TxTCIF 标志甚至会在 SPI 总线上的通信开始前置 1。SPI 传输完成前，该标志始终为高电平状态。
6. 封装的 CRC 值在 SPIx_TxCRCR 和 SPIx_RxCRCR 寄存器中逐帧连续进行计算。完成整个数据封装后，CRC 信息可通过 DMA 自动处理（Tx 通道必须设置为要处理的数据帧数），也可由软件处理（用户必须在处理最后一个数据帧的过程中处理 CRCNEXT 位）。SPIx_TxCRCR 中计算的 CRC 值仅由发射器发出时，接收的 CRC 信息将加载到 RxFIFO 中，然后与 SPIx_RxCRCR 寄存器的内容进行比较（如果存在任何差异，CRC 错误标志会置 1）。因此用户必须注意刷新 FIFO 中的相关信息，可以通过软件读出 RxFIFO 中存储的所有内容的方式来实现，若已针对 Rx 通道预设置了适当数量的数据帧（数据帧数 + CRC 帧数），也可通过 DMA 的方式来实现（请参见示例假设中的设置）。
7. 在数据封装模式下，TxE 和 RxNE 事件成对出现，对 FIFO 的每次读/写访问都为 16 位宽，直至数据帧数为偶数。如果 TxFIFO 处于¾满状态，则 FTLVL 将保持 FIFO 满时对应的状态。因此在 TxFIFO 变为½满状态前，无法存储最后一个奇数编号的数据帧。该帧可通过软件或自动由 DMA（LDMA_TX 控制置 1 时）对其进行 8 位访问的方式存储在 TxFIFO 中。
8. 要在封装模式下接收最后一个奇数编号的数据帧，必须在处理最后一个数据帧后，将 Rx 阈值更改为 8 位，这可通过软件（设置为 FRXTH=1）实现或由 DMA 内部信号在 LDMA_RX 置 1 时自动实现。

图 159. 主设备全双工通信



主设备全双工通信示例的假设条件如下：

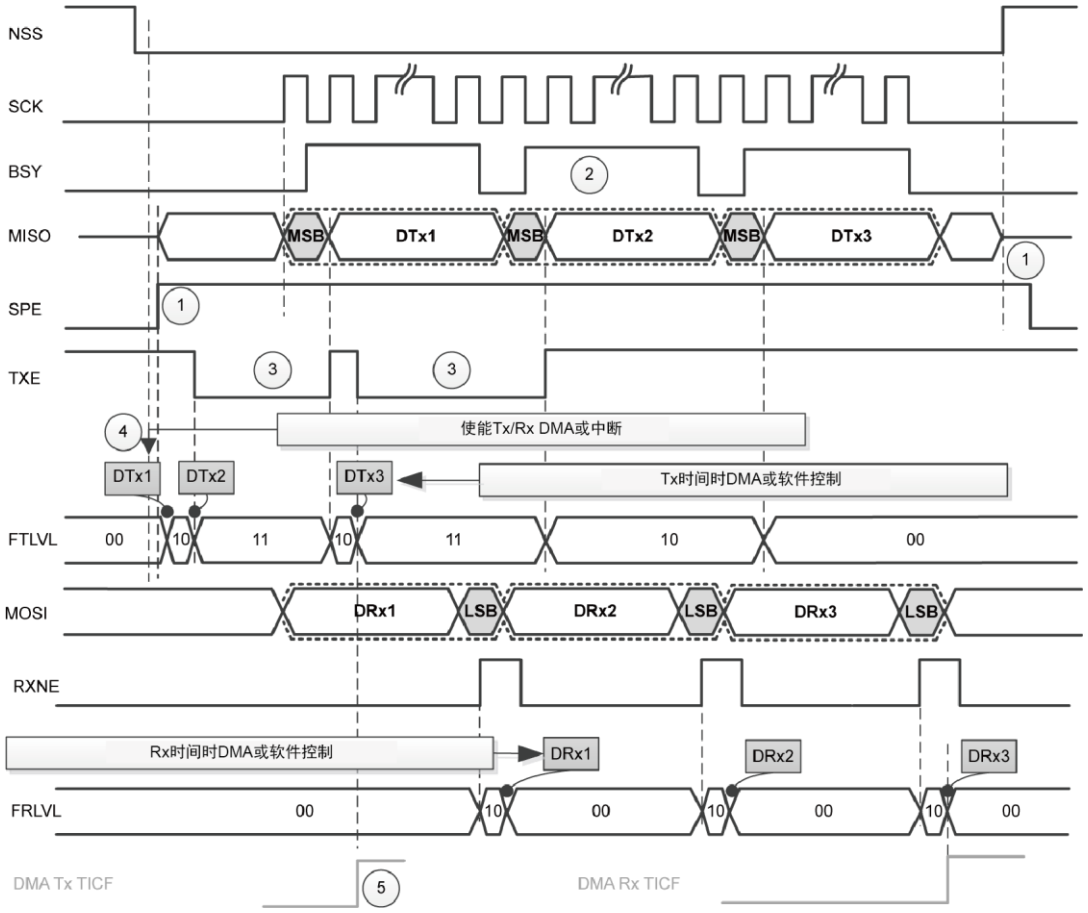
- 数据大小 > 8 位

如果使用 DMA：

- 由 DMA 处理的 Tx 帧的数量设置为 3
- 由 DMA 处理的 Rx 帧的数量设置为 3

另请参见：“通信图”一节获取关于共同假设和注释的详细信息。

图 160. 从设备全双工通信



从设备全双工通信示例的假设条件如下：

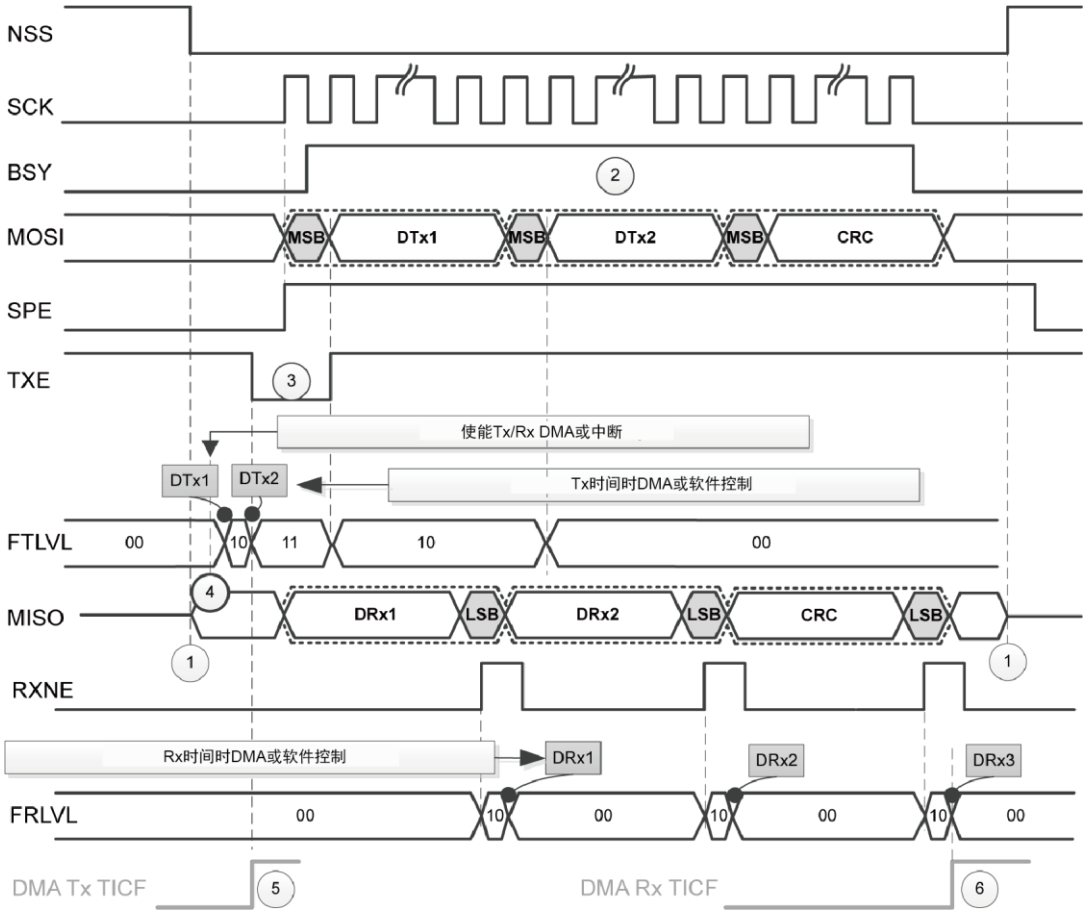
- 数据大小 > 8 位

如果使用 DMA：

- 由 DMA 处理的 Tx 帧的数量设置为 3
- 由 DMA 处理的 Rx 帧的数量设置为 3

另请参见：“通信图”一节获取关于共同假设和注释的详细信息。

图 161. 采用 CRC 的主设备全双工通信



采用 CRC 的主设备全双工通信的假设条件如下：

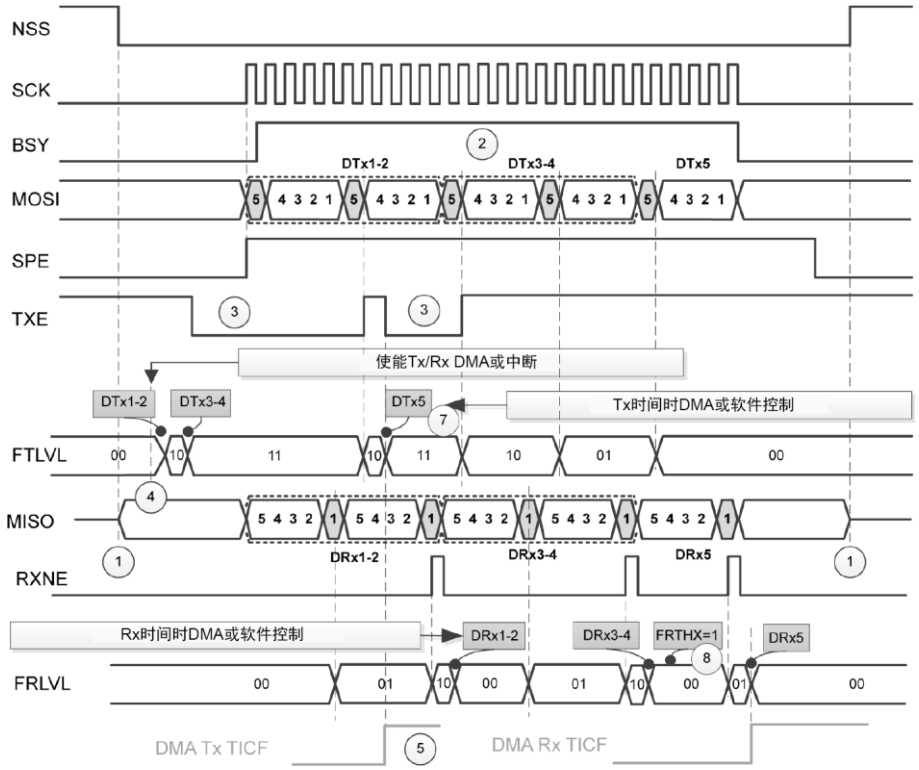
- 数据大小 = 16 位
- CRC 使能

如果使用 DMA：

- 由 DMA 处理的 Tx 帧的数量设置为 2
- 由 DMA 处理的 Rx 帧的数量设置为 3

另请参见：“通信图”一节获取关于共同假设和注释的详细信息。

图 162. 封装模式下的主设备全双工通信



封装模式下的主设备全双工通信示例的假设条件如下：

- 数据大小 = 5 位
- 主要通过 16 位访问的方式来读/写 FIFO
- FRXTH=0.

如果使用 DMA：

- 要由 DMA 处理的 Tx 帧的数量设置为 3
- 要由 DMA 处理的 Rx 帧的数量设置为 3
- Tx 和 Rx DMA 通道的 PSIZE 均设置为 16 位
- LDMA_TX=1 且 LDMA_RX=1

另请参见：“通信图”一节获取关于共同假设和注释的详细信息。

22.5.9 SPI 状态标志

应用可通过三种状态标志监视 SPI 总线的状态。

发送缓冲区为空（TXE）

当发送 TXFIFO 有足够的空间来存储要发送的数据时，TXE 标志将置 1。TXE 标志与 TXFIFO 占用水平相关。该标志变为高电平后将一直保持高电平状态，直至 TXFIFO 占用水平小于或等于 FIFO 深度的 1/2。如果 SPIx_CR2 寄存器中的 TXEIE 位置 1，可产生中断。当 TXFIFO 占用水平超过 1/2 时，该位将自动清零。

接收缓冲区非空（RXNE）

RXNE 标志根据 SPIx_CR2 寄存器中 FRXTH 位的值进行设置：

- 如果 FRXTH 置 1，RXNE 将变为高电平并一直保持高电平状态，直至 RXFIFO 占用水平大于或等于 1/4（8 位）。
- 如果 FRXTH 清零，RXNE 将变为高电平并一直保持高电平状态，直至 RXFIFO 占用水平大于或等于 1/2（16 位）。

如果 SPIx_CR2 寄存器中的 RXNEIE 位置 1，可产生中断。

当上述条件不再为真时，RXNE 将由硬件自动清零。

忙标志（BSY）

BSY 标志由硬件置 1 和清零（写入此标志没有任何作用）。

当 BSY 置 1 时，表示 SPI 上正在进行数据传输（SPI 总线繁忙）。

某些模式下可以使用 BSY 标志来检测传输是否结束，以便软件在进入低功耗模式（该模式下不提供外设时钟）前禁止 SPI 或其外设时钟。这可避免破坏最后一个数据的传输。

BSY 标志还可用于避免在多主模式系统中发生写冲突。在以下任意一种条件下，BSY 标志将清零：

- 正确禁止 SPI 时
- 在主模式下检测到故障时（MODF 位置 1）
- 在主模式下，完成了数据发送并且不准备发送任何新数据时
- 在从模式下，BSY 标志在各传输之间的至少一个 SPI 时钟周期内置为“0”时。

注意： *当主设备可以立即处理下一次发送时（例如，如果主设备处于只接收模式或其发送 FIFO 不为空），在主设备侧的传输之间，通信连续且 BSY 标志始终置“1”。尽管从设备并非如此，但建议始终使用 TXE 和 RXNE 标志（而非 BSY 标志）来处理数据发送或接收操作。*

22.5.10 SPI 错误标志

如果以下其中一个错误标志置 1 且已通过将 ERRIE 位置 1 使能了中断，则将生成 SPI 中断。

上溢标志（OVR）

当主设备或从设备接收了数据但 RXFIFO 没有足够的空间来存储接收的数据时，将出现上溢的情况。如果在只接收模式下使能了 CRC，当软件或 DMA 没有足够的时间来读取之前接收的数据（存储在 RXFIFO 中）或用于存储数据的空间有限（如 RXFIFO 不可用）时，可能发生这种情况，此时接收缓冲区仅限于单个数据帧缓冲区（参见第 22.5.13 节“CRC 计算”）。

当发生上溢状态时，RXFIFO 中新接收的值不会覆盖之前的值。新接收的值会被丢弃，之后发送的所有数据均丢失。要将 OVR 位清零，应首先对 SPI_DR 寄存器执行读访问，然后再对 SPI_SR 寄存器执行读访问。

模式故障（MODF）

当主设备的内部 NSS 信号（NSS 硬件模式下为 NSS 引脚，NSS 软件模式下为 SSI 位）被拉低时，将发生模式故障。这会自动将 MODF 位置 1。主模式故障会在以下几方面影响 SPI 接口：

- 如果 ERRIE 位置 1，MODF 位将置 1，并生成 SPI 中断
- SPE 位清零。这将关闭器件的所有输出，并禁止 SPI 接口。
- MSTR 位清零，从而强制器件进入从模式。

  使用以下软件序列将 MODF 位清零：

  1. 在 MODF 位置 1 时，对 SPIx_SR 寄存器执行读或写访问。
  2. 然后对 SPIx_CR1 寄存器执行写访问。

为避免包含多个 MCU 的系统中发生多从模式冲突，必须在 MODF 位清零序列期间将 NSS 引脚拉高。在该清零序列后，可以将 SPE 和 MSTR 位恢复到原始状态。安全起见，硬件不允许在 MODF 位置 1 时将 SPE 和 MSTR 位置 1。在从设备中，MODF 位不可置 1，但由前一次多主模式冲突引起时除外。

CRC 错误（CRCERR）

当 SPIx_CR1 寄存器中的 CRCEN 位置 1 时，此标志用于验证接收数据的有效性。如果移位寄存器中接收的值与 SPIx_RXCRCR 的值不匹配，SPIx_SR 寄存器中的 CRCERR 标志将置 1。该标志由软件清零。

TI 模式帧格式错误（FRE）

如果 SPI 在从模式下工作，并配置为符合 TI 模式协议，则在通信进行期间出现 NSS 脉冲时，将检测到 TI 模式帧格式错误。出现此错误时，SPIx_SR 寄存器中的 FRE 标志将置 1。发生错误时不会禁止 SPI，但会忽略 NSS 脉冲，并且 SPI 会等待下一个 NSS 脉冲，然后再开始新的传输。由于错误检测可能导致丢失两个数据字节，因此数据可能会损坏。

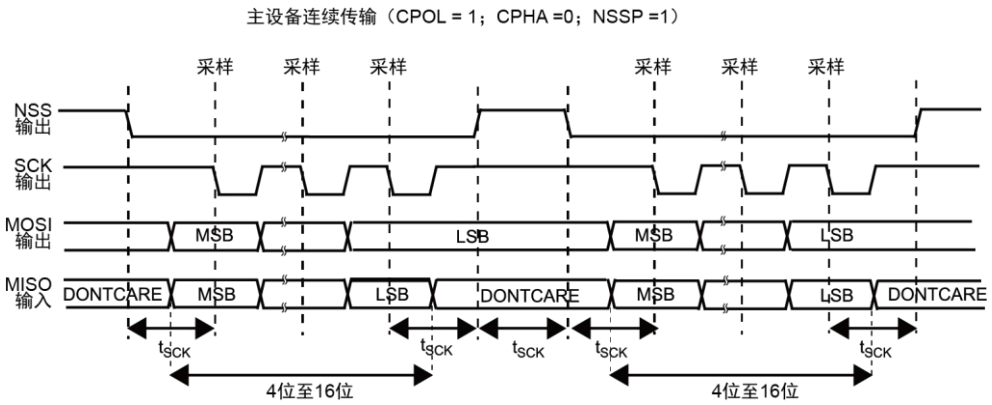
读取 SPIx_SR 寄存器时，将清零 FRE 标志。如果 ERRIE 位置 1，则检测到 NSS 错误时将生成中断。在这种情况下，由于无法保证数据的一致性，应禁止 SPI，并在重新使能从 SPI 后，由主设备重新发起通信。

22.5.11 NSS 脉冲模式

该模式通过 SPIx_CR2 寄存器中的 NSSP 位来激活，只有将 SPI 接口配置为 Motorola SPI 主模式（FRF=0）且在第一个边沿捕捉时，该模式才起作用（SPIx_CR1 CPHA = 0，CPOL 设置忽略）。激活后，当 NSS 至少保持一个时钟周期的高电平状态时，两个连续的数据帧传输间将生成 NSS 脉冲。该模式下，从设备可以锁存数据。NSSP 脉冲模式旨在用于具有一个主设备-从设备对的应用。

图 163. Motorola SPI 主模式下的 NSSP 脉冲生成显示了 NSSP 脉冲模式使能时的 NSS 引脚管理。

图 163. Motorola SPI 主模式下的 NSSP 脉冲生成



注意： *当 CPOL=0 时会遇到相似行为。在本例中，采样边沿是 SCK 的上升沿，NSS 使能和禁用指的是该采样边沿。*

22.5.12 TI 模式

主模式下的 TI 协议

SPI 接口与 TI 协议兼容。可以使用 SPIx_CR2 寄存器的 FRF 位来配置 SPI，以兼容此协议。

时钟极性和相位都被强制为遵循 TI 协议，和 SPIx_CR1 中的设置无关。NSS 管理也特定于 TI 协议，在这种情况下，无法通过 SPIx_CR1 和 SPIx_CR2 寄存器（SSM、SSI 和 SSOE）来对 NSS 管理进行配置。

在从模式下，SPI 波特率预分频器用于控制在当前事务完成时 MISO 引脚状态变为 HiZ 的时刻（参见图 164. TI 模式传输）。可以使用任意波特率，因此可以非常灵活地确定此时刻。

但是，波特率通常设置为外部主时钟波特率。MISO 信号变为高阻态的延时（trelease）取决于内部重新同步以及通过 SPIx_CR1 寄存器的 BR[2:0]位设置的波特率值。它可根据以下公式得出：

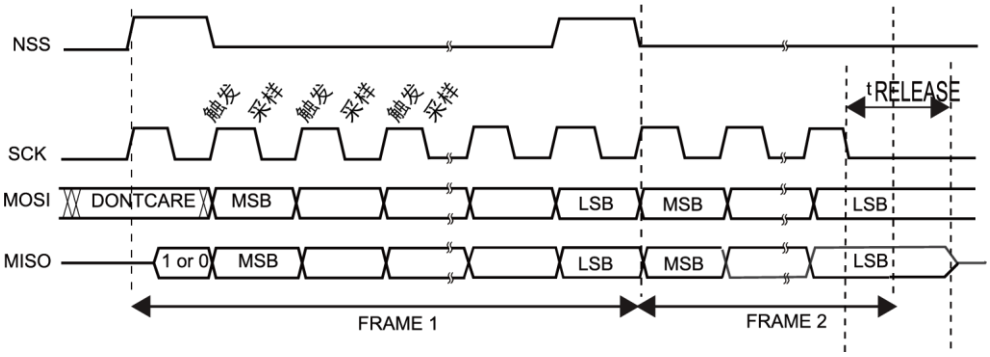
$$\frac{t_{baud_rate}}{2} + 4 \times t_{pclk} < t_{release} < \frac{t_{baud_rate}}{2} + 6 \times t_{pclk} \quad (8)$$

如果从设备在数据帧传输期间检测到错位的 NSS 脉冲，TIFRE 标志将置 1。

如果数据大小等于 4 位或 5 位，处于全双工模式或只发送模式的主设备使用的协议将在 LSB 之后再增加一个空数据位。将基于此空位时钟周期而不是每个周期中的 LSB 生成 TI NSS 脉冲。此特性不适用于 Motorola SPI 通信（FRF 位为 0）。

图 164. TI 模式传输显示了选择 TI 模式时的 SPI 通信波形。

图 164. TI 模式传输



22.5.13 CRC 计算

为检查发送数据和接收数据的可靠性，使用两个独立的 CRC 计算器。SPI 可提供 CRC8 或 CRC16 计算，而与固定为 8 位或 16 位的数据帧长度无关。对于所有其它的数据帧长度，CRC 均不适用。

CRC 原理

在使能 SPI（SPE = 1）前，通过将 SPIx_CR1 寄存器中的 CRCEN 位置 1 来使能 CRC 计算。对每一位使用可编程奇多项式计算 CRC 值。在由 SPIx_CR1 寄存器中的 CPHA 位和 CPOL 位定义的采样时钟边沿进行计算。在数据块结束时自动检查计算所得 CRC 值，并通过 CPU 或 DMA 进行传输。当检测到基于接收的数据内部计算得出的 CRC 与发射器发送的 CRC 不一致时，会将 CRCERR 标志置 1 以指示发生了数据损坏错误。CRC 计算的正确处理步骤取决于 SPI 配置和所选的传输管理。

注意： *多项式值只应为奇数。不支持任何偶数值。*

CPU 管理的 CRC 传输

通信开始后将一直持续到必须发送或接收 SPIx_DR 寄存器中的最后一个数据帧时。然后，必须将 SPIx_CR1 寄存器中的 CRCNEXT 位置 1，以指示 CRC 帧事务在当前处理的数据帧的事务之后。CRCNEXT 位必须在最后一个数据帧传输结束前置 1。在 CRC 传输期间，CRC 计算将冻结。

所接收的 CRC 以数据字节或数据字的形式存储在 RXFIFO 中。因此仅在 CRC 模式下，接收缓冲区才必须被视为一个 16 位缓冲区且一次仅接收一个数据帧。

CRC 帧的传输通常在数据序列结束时再传输一个数据帧。然而，在设置通过 16 位 CRC 校验的 8 位数据帧时，还需要另外两个帧才能发送完整的 CRC。

接收最后一个 CRC 数据后，将执行自动校验，将接收的值与 SPIx_RXCRC 寄存器中的值进行比较。软件必须校验 SPIx_SR 寄存器中的 CRCERR 标志，以确定数据传输是否损坏。软件通过向 CRCERR 标志写入“0”来将其清零。

接收 CRC 后，CRC 值存储到 RXFIFO 中，且必须在 SPIx_DR 寄存器中进行读取，以将 RXNE 标志清零。

DMA 管理的 CRC 传输

当使能的 SPI 通信支持 CRC 通信和 DMA 模式时，在通信结束时会自动发送和接收 CRC（在只接收模式下读取 CRC 数据时除外）。CRCNEXT 位不是一定要通过软件来处理。SPI 发送 DMA 通道计数器必须设置为要发送的数据帧数，其中不包括 CRC 帧。在接收器侧，接收的 CRC 值在传输结束时通过 DMA 自动处理，但用户必须注意刷新 RXFIFO 中接收的 CRC 信息（因为该信息始终加载到其中）。在全双工模式下，接收 DMA 通道计数器可设置为要接收的数据帧数，其中包括 CRC，这意味着在通过 16 位 CRC 校验的 8 位数据帧等特殊情况下：

DMA_RX = Numb_of_data + 2

在只接收模式下，DMA 接收通道计数器应仅包含已传输的数据量，而不包括 CRC 计算。之后，基于通过 DMA 实现的完整传输，必须通过软件从 FIFO 中读回所有 CRC 值，因为 FIFO 在该模式下用作单个缓冲区。

如果传输过程中出现损坏，则在数据和 CRC 传输结束时，SPIx_SR 寄存器中的 CRCERR 标志将置 1。

如果使用封装模式，则 LDMA_RX 位需要管理数据数是否为奇数。

复位 SPIx_TXCRC 和 SPIx_RXCRC 值

在 CRC 阶段后对新数据进行采样时，SPIx_TXCRC 和 SPIx_RXCRC 值将自动清零。借此，可使用 DMA 循环模式（只接收模式下不适用）从而不间断地传输数据（中间 CRC 校验阶段会涉及一些数据块）。

如果在通信过程中禁止 SPI，必须按以下顺序进行操作：

1. 禁止 SPI
2. 将 CRCEN 位清零
3. 使能 CRCEN 位
4. 使能 SPI

注意： *当 SPI 处于从模式时，只要 CRCEN 位置 1，无论 SPE 位的值如何，只要存在 SCK 从输入时钟，CRC 计算器就开始工作。为了避免 CRC 计算错误，软件必须仅在时钟稳定（处于稳态）时才能使能 CRC 计算。当 SPI 接口配置为从模式时，NSS 内部信号需要在数据阶段和 CRC 阶段之间保持低电平。*

22.6 SPI 中断

在 SPI 通信过程中，中断可由以下事件产生：

- 发送 TXFIFO 准备就绪，可以装载数据
- 接收 RXFIFO 中接收了数据
- 主模式故障
- 上溢错误
- TI 帧格式错误

中断可分别进行使能和禁止。

表 77. SPI 中断请求

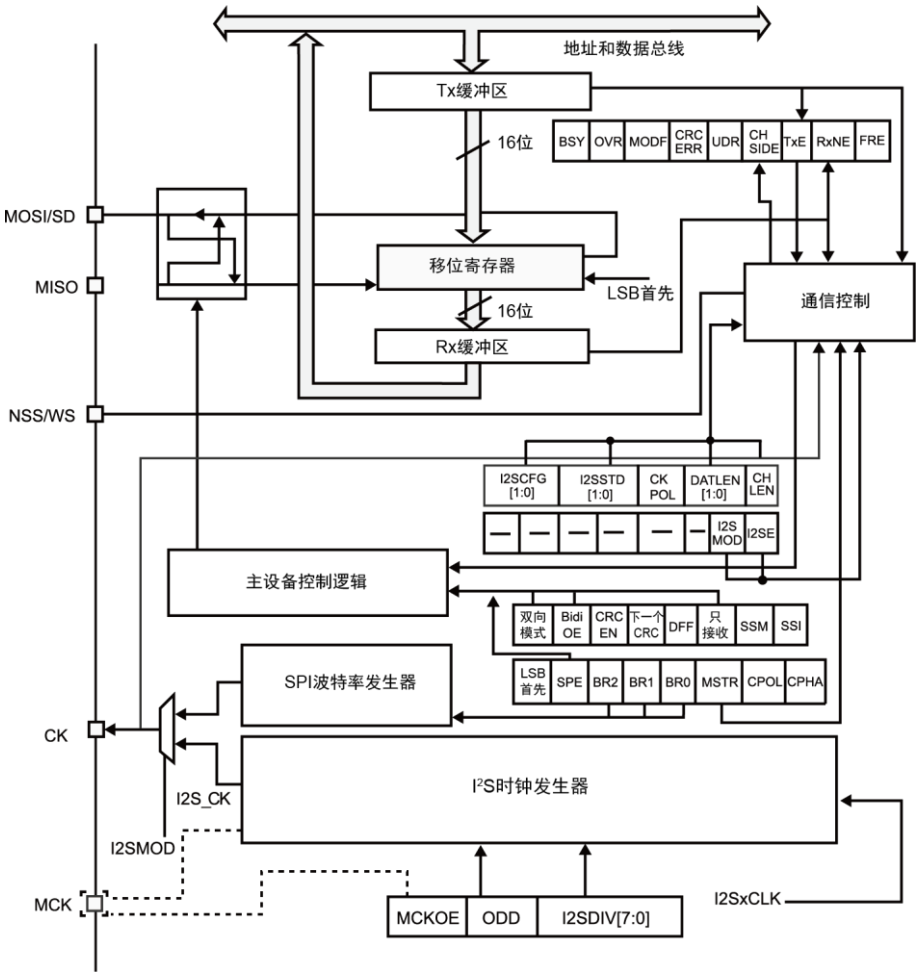
中断事件	事件标志	使能控制位
发送 TXFIFO 准备就绪，可以装载数据	TXE	TXEIE
RXFIFO 中接收了数据	RXNE	RXNEIE
主模式故障	MODF	ERRIE
上溢错误	OVR	
TI 帧格式错误	FRE	

22.7 I2S 功能说明

22.7.1 I2S 概述

I2S 的框图如下。

图 165. I²S 框图



当 I²S 功能使能（通过将 SPIx_I2SCFGR 寄存器中的 I2SMOD 位置 1）时，SPI 可充当音频 I²S 接口。此接口使用几乎与 SPI 相同的引脚、标志和中断。

I²S 与 SPI 共享三个共用引脚：

- SD：串行数据（映射到 MOSI 引脚），用于发送或接收两个时分复用的数据通道上的数据（仅半双工模式）。
- WS：字选择（映射到 NSS 引脚），是主模式下的数据控制信号输出以及从模式下的数据控制信号输入。
- CK：串行时钟（映射到 SCK 引脚），是主模式下的串行时钟输出以及从模式下的串行时钟输入。

当某些外部音频设备需要使用主时钟输出时，可以使用其它引脚：

- MCK：当 I²S 配置为主模式（并且 SPIx_I2SPR 寄存器中的 MCKOE 位置 1）时，使用主时钟（单独映射）输出此额外时钟，此时钟以等于 256 × f_S 的预配置频率生成，其中的 f_S 是音频采样频率。

当 I²S 配置为主模式时，它使用自己的时钟发生器生成通信时钟。此时钟发生器也是主时钟输出的源。I²S 模式下有两个额外的寄存器可供使用。一个是时钟发生器配置寄存器 SPIx_I2SPR，另一个是通用 I²S 配置寄存器 SPIx_I2SCFGR（音频标准、从/主模式、数据格式、数据包帧、时钟极性等）。

在 I²S 模式下，不使用 SPIx_CR1 寄存器和所有 CRC 寄存器。同样，也不使用 SPIx_CR2 寄存器中的 SSOE 位以及 SPIx_SR 中的 MODF 和 CRCERR 位。

I²S 使用与 16 位宽度模式的数据传输（SPIx_DR）相同的 SPI 寄存器。

22.7.2 支持的音频协议

三线总线必须处理两个通道上只有音频的数据（通常为时分复用）：右通道和左通道。但是，只有一个 16 位寄存器进行发送和接收。所以，需由软件将与每个通道对应的值写入数据寄存器，以及从数据寄存器中读取数据，并通过检查 SPIx_SR 寄存器中的 CHSIDE 位来识别对应的通道。始终先发送“左通道数据”，而后再发送右通道数据（对于 PCM 协议来说，CHSIDE 没有意义）。

数据和帧格式组合有四种，可采用下列格式发送数据：

- 将 16 位数据封装在 16 位帧中
- 将 16 位数据封装在 32 位帧中
- 将 24 位数据封装在 32 位帧中
- 将 32 位数据封装在 32 位帧中

当使用 32 位数据包中的 16 位数据时，前 16 位（MSB）为有效位，16 位 LSB 被强制清零，无需任何软件操作或 DMA 请求（只需一个读/写操作）。

如果应用程序首选 DMA，则 24 位和 32 位数据帧需要对 SPIx_DR 寄存器执行两次 CPU 读取或写入操作，或者需要两次 DMA 操作。对于 24 位数据帧，硬件会在低 8 位填充 8 个 0 扩展到 32 位。

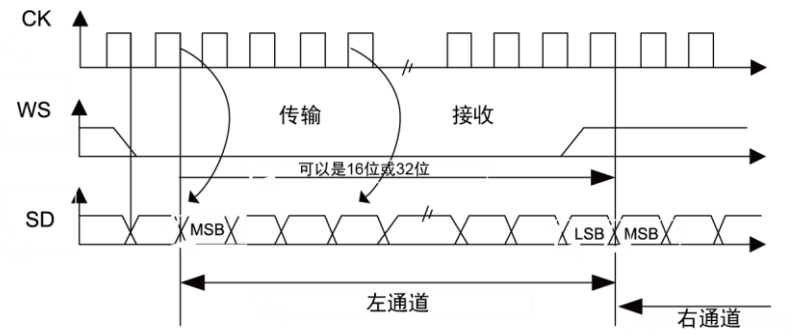
对于所有数据格式和通信标准而言，始终会先发送最高有效位（MSB 优先）。

I²S 接口支持四个音频标准，可使用 SPIx_I2SCFGR 寄存器中的 I2SSTD[1:0]和 PCMSYNC 位进行配置。

I²S Philips 标准

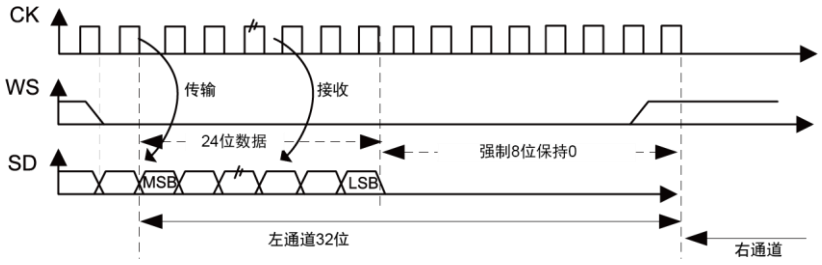
使用 WS 信号来指示当前正在发送的数据所属的通道。该信号从当前通道数据的第一个位（MSB）之前的一个时钟开始有效。

图 166. I²S Philips 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 CK 的下降沿变化。

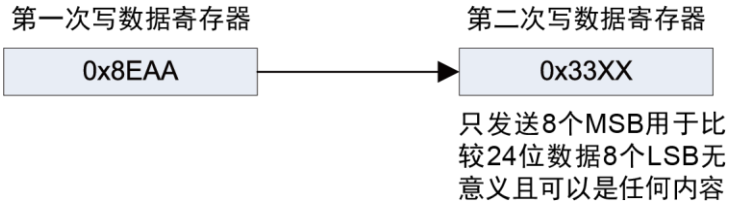
图 167. I²S Philips 标准波形（24 位帧，CPOL = 0）



该模式需要对 SPIx_DR 寄存器执行两次写入或读取操作。

- 在发送模式下：
  如果需要发送 0x8EAA33（24 位）：

图 168. 发送 0x8EAA33



- 在接收模式下：
  如果接收数据 0x8EAA33：

图 169. 接收 0x8EAA33

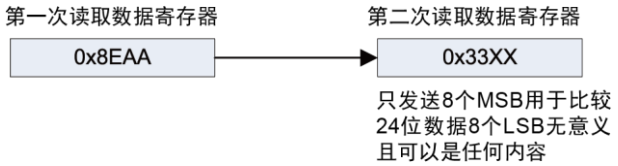
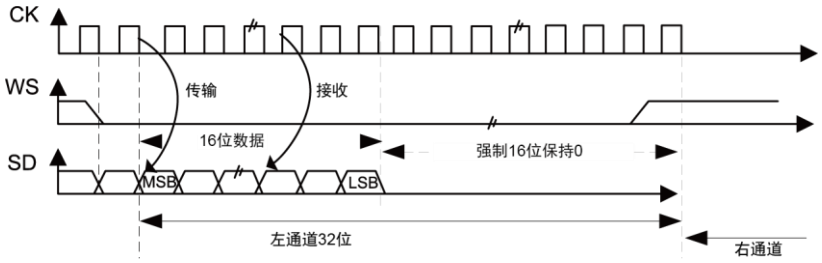


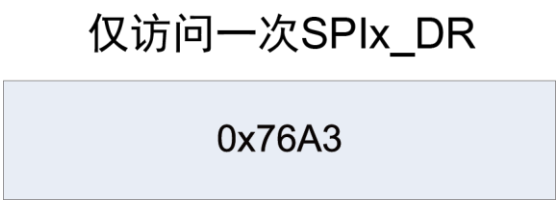
图 170. I²S Philips 标准（16 位扩展到 32 位数据包帧，CPOL = 0）



如果在 I2S 配置阶段选择将 16 位数据帧扩展到 32 位通道帧，则只需要访问一次 SPIx_DR 寄存器。扩展到 32 位中高半字 （16 位 MSB）被硬件置为 0x0000。

如果要发送的数据或已接收的数据是 0x76A3（0x76A30000 扩展到 32 位），则需要如“图 171. 16 位数据帧扩展到 32 位通道帧的示例”所示的操作。

图 171. 16 位数据帧扩展到 32 位通道帧的示例



发送时，每次将 MSB 写入 SPIx_DR，TXE 标志就会置 1，并在中断使能的情况下触发中断，以将要发送的新数据加载到 SPIx_DR 寄存器。即使硬件填充的低 16 位 0x0000 还未发送，也会如此，因为低 16 位是由硬件发送。

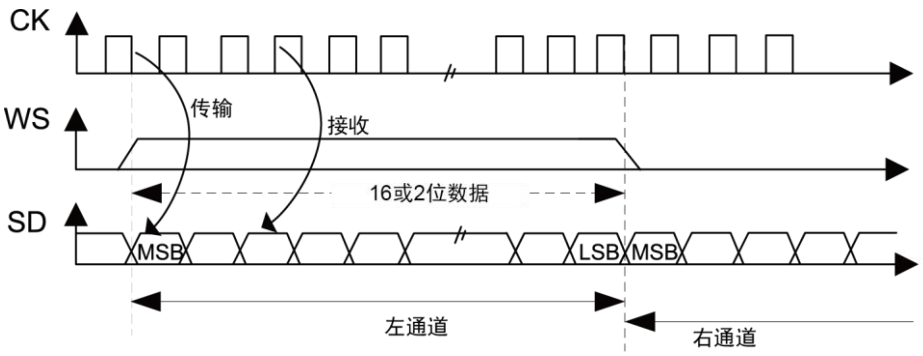
接收时，接收到第一个半字（高 16 位），则硬件将 RXNE 标志置 1，并在中断使能的情况下触发中断。

这样，就延长了两个写入或读取操作之间的时间间隔，从而可防止出现下溢或上溢情况（具体取决于数据传输方向）。

MSB 对齐标准

此标准同时生成 WS 信号和第一个数据位（即 MSBit）。

图 172. MSB 对齐的 16 位或 32 位全精度长度，CPOL = 0



发送方在时钟信号的下降沿改变数据；接收方在上升沿读取数据。

图 173. MSB 对齐的 24 位帧长度，CPOL = 0

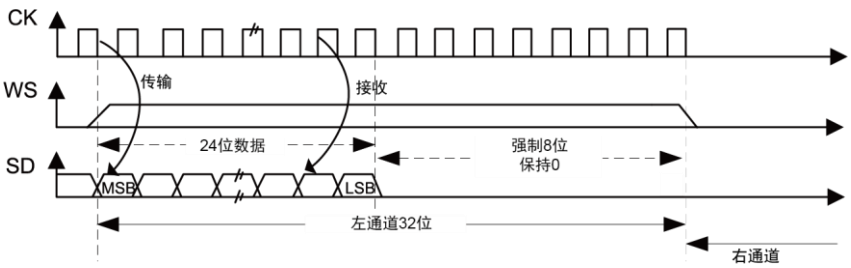
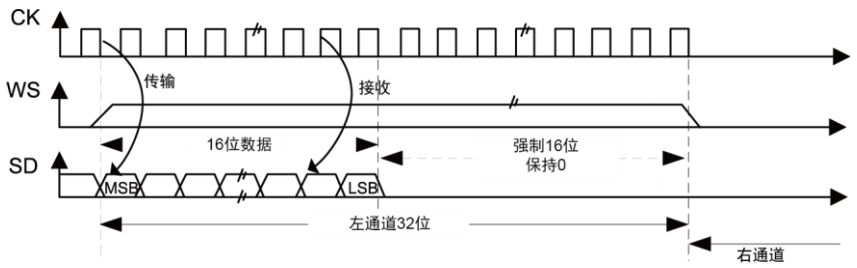


图 174. 扩展为 32 位数据包帧的 MSB 对齐的 16 位，CPOL = 0



LSB 对齐标准

该标准与 MSB 对齐标准类似（对于 16 位和 32 位全精度帧格式，没有任何不同）。

图 175. LSB 对齐的 16 位或 32 位全精度长度，CPOL = 0

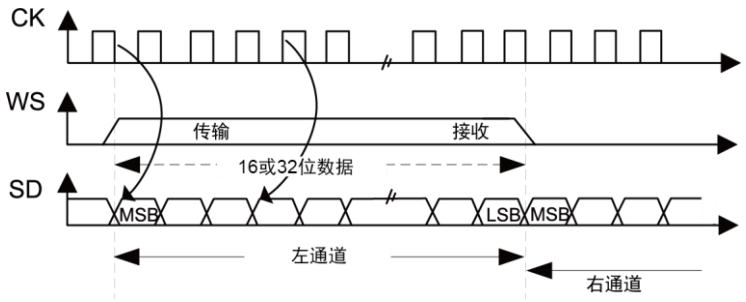
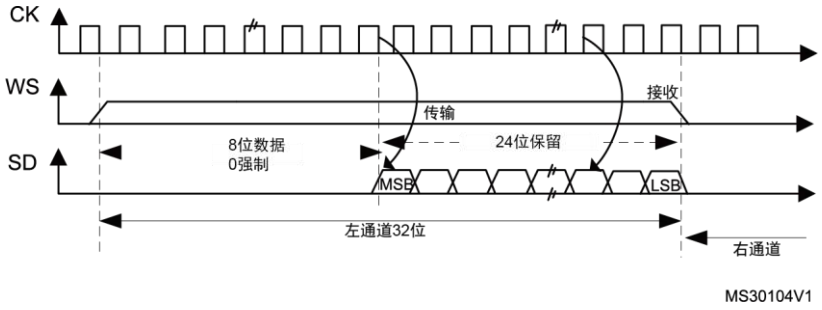


图 176. LSB 对齐的 24 位帧长度，CPOL = 0

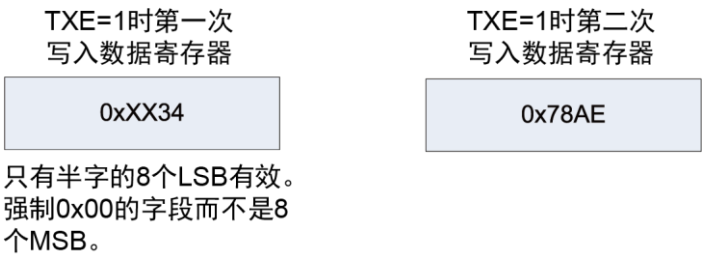


MS30104V1

- 在发送模式下：

  如果必须发送数据 0x3478AE，则需要通过软件或 DMA 对 SPIx_DR 寄存器执行两次写操作。操作如下所示。

图 177. 发送 0x3478AE 所需的操作



- 在接收模式下：
  如果接收到数据 0x3478AE，则在每个 RXNE 事件时需要对 SPIx_DR 寄存器执行两次连续的读取操作。

图 178. 接收 0x3478AE 时所需的操作

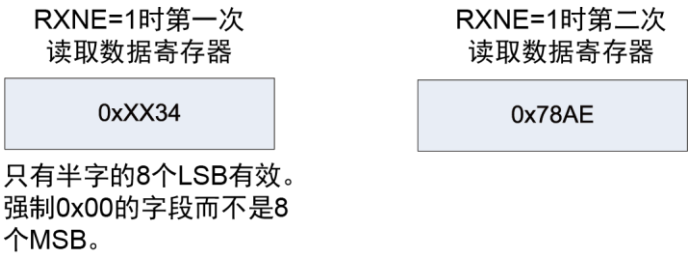
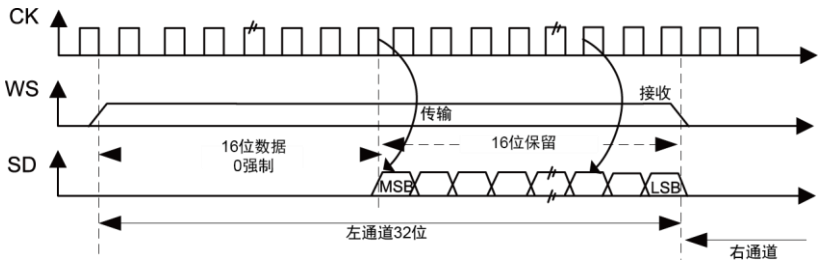


图 179. 扩展为 32 位数据包帧的 LSB 对齐的 16 位，CPOL = 0



如果在 I2S 配置阶段选择将 16 位数据帧扩展到 32 位通道帧，则只需要访问一次 SPIx_DR 寄存器。扩展到 32 位中高半字 （16 位 MSB）被硬件置为 0x0000。在这种情况下，其对应于半字 MSB。

如果要发送的数据或已接收的数据是 0x76A3（0x0000 76A3 扩展到 32 位），则需要如“图 180. 16 位数据帧扩展到 32 位通道帧的示例（2）”所示的操作

图 180. 16 位数据帧扩展到 32 位通道帧的示例（2）



在发送模式下，发生 TXE 事件时，应用程序需要写入要发送的数据（此例中，为 0x76A3）。首先发送 0x000 字段（扩展到 32 位）。有效数据（0x76A3）发送到 SD 后，TXE 标志会被再次置 1。

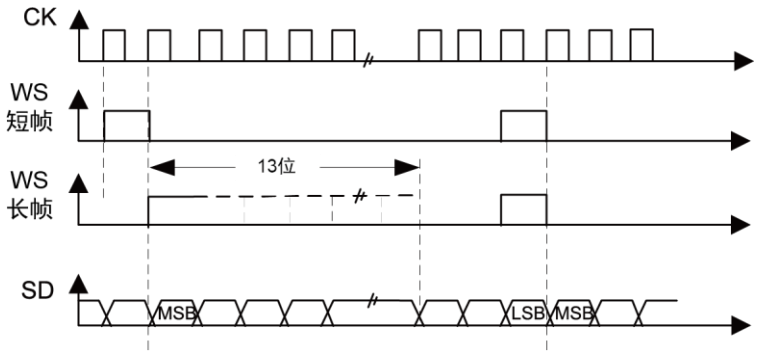
在接收模式下，当接收到有效半字后（而非 0x0000 字段），即会置位 RXNE。

这样，就延长了两个写入或读取操作之间的时间间隔，以防止出现下溢或上溢情况。

PCM 标准

对于 PCM 标准，无需使用通道信息。可使用两种 PCM 模式（短帧和长帧），并且可使用 SPIx_I2SCFGR 寄存器中的 PCMSYNC 位来配置。

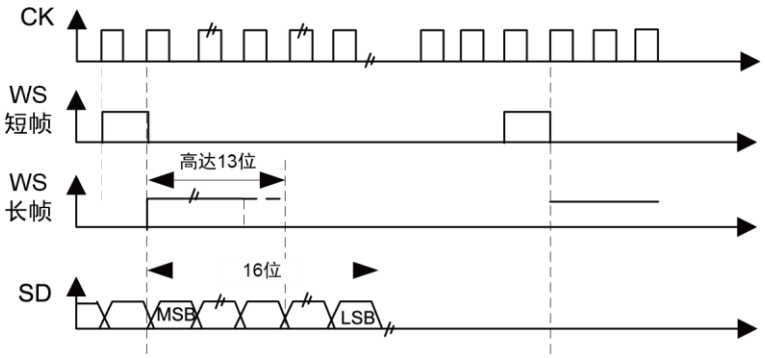
图 181. PCM 标准波形（16 位）



对于长帧同步，在主模式下会将 WS 信号持续 13 个周期。

对于短帧同步，WS 同步信号的持续时间仅为一个周期。

图 182. PCM 标准波形（16 位扩展到 32 位数据包帧）



注意： *对于两种模式（主/从模式）和两种同步（短/长同步），即使在从模式下，也需要指定两组连续数据（以及两个同步信号）之间位的个数（SPIx_I2SCFGR 寄存器中的 DATLEN 位和 CHLEN 位）。*

22.7.3 时钟发生器

I^2S 比特率决定了 I^2S 数据线路上的数据流和 I^2S 时钟信号频率。

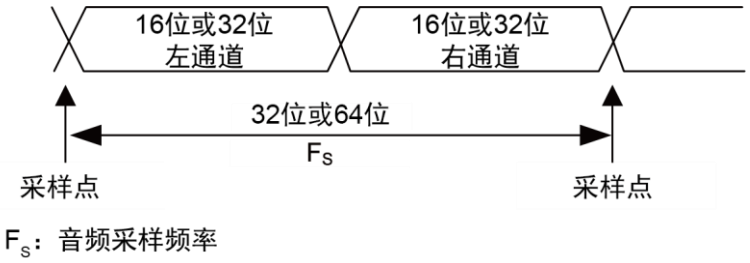
I^2S 比特率 = 每通道比特位数 × 通道数 × 采样音频频率。

I^2S 比特率（左和右通道）的计算如下：

I^2S 比特率 = 16 × 2 × 16 位音频的 f_S

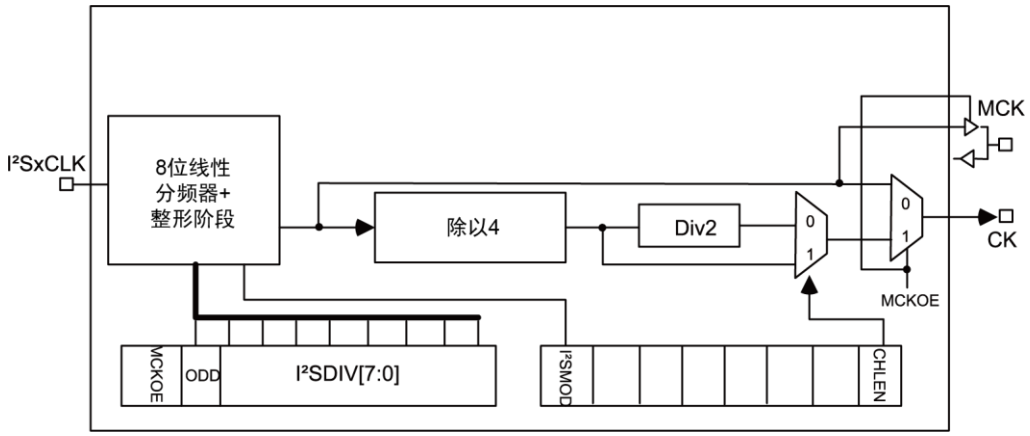
I^2S 比特率 = 32 × 2 × 32 位音频的 f_S。

图 183. 音频采样频率定义



配置主模式时，需要正确地对线性分频器进行设置，以便采用所需的音频频率进行通信。

图 184. I^2S 时钟发生器架构



其中的 x 可以是 2 或 3。

图 184. I^2S 时钟发生器架构显示了通信时钟架构。I2Sx 时钟始终是 32 MHz 频率时钟。

注意：此外，必须确保 I2SxCLK 频率高于或等于 SPI/I2S 模块使用的 APB 时钟频率。如不符合这一条件，SPI/I2S 将不工作。

音频采样频率可以是 96 kHz、48 kHz、44.1 kHz、32 kHz、22.05 kHz、16 kHz、11.025 kHz 或 8 kHz（或此范围内的任何其它值）。为达到所需频率，需要根据以下公式对线性分频器进行编程：

输出主时钟（SPIx_I2SPR 寄存器中的 MCKOE 置 1）时：

f_S = I2SxCLK/[256 *（(2*I2SDIV）+ODD）]，无论通道帧宽度是多少（16 位或 32 位宽度）。

禁止主时钟输出（MCKOE 位清零）时：

f_S = I2SxCLK/[(16*2）*((2*I2SDIV）+ODD）]（通道帧宽度为 16 位时）。

f_S = I2SxCLK/[(32*2）*((2*I2SDIV）+ODD）]（通道帧宽度为 32 位时）。

表 78. I2SCLK = 32 MHz 时的音频频率精度提供了不同时钟配置的精度值示例。

注意： *还可以采用其它配置以达到更好的时钟精度。*

表 78. I2SCLK = 32 MHz 时的音频频率精度

I2SCLK（MHz）	数据长度	I2SDIV	I2SODD	MCK	目标 fs（Hz）	实际 fs（Hz）	误差
32	16	5	0	无	96000	100000	4.1667%
32	32	2	1	无	96000	100000	4.1667%
32	16	10	1	无	48000	47619.0476	0.7936%
32	32	5	0	无	48000	50000	4.167%
32	16	11	1	无	44100	43478.261	1.410%
32	32	8	1	无	44100	45454.545	3.0715%
32	16	15	1	无	32000	32258.0645	0.806%
32	32	8	0	无	32000	31250	2.344%
32	16	22	1	无	22050	22222.22	0.781%
32	32	11	1	无	22050	21739.1304	1.410%
32	16	31	1	无	16000	15873.0159	0.794%
32	32	15	1	无	16000	16129.032	0.806%
32	16	45	1	无	11025	10989.011	0.326%
32	32	22	1	无	11025	11111.111	0.781%
32	16	62	1	无	8000	8000	0%
32	32	47	0	无	8000	7936.508	0.794%
32	16	1	1	有	48000	41666.667	13.194%
32	32	1	1	有	48000	41666.667	13.194%
32	16	1	1	有	44100	41666.667	5.518%
32	32	1	1	有	44100	41666.667	5.518%
32	16	2	0	有	32000	31250	2.3438%
32	32	2	0	有	32000	31250	2.3438%
32	16	3	0	有	22050	20833.333	5.5178%
32	32	3	0	有	22050	20833.333	5.5178%
32	16	4	0	有	16000	15625	2.3438%
32	32	4	0	有	16000	15625	2.3438%
32	16	5	1	有	11025	11363.6364	3.0715%
32	32	5	1	有	11025	11363.6364	3.0715%
32	16	8	0	有	8000	7812.5	2.344%
32	32	8	0	有	8000	7812.5	2.344%

表 79. I2SCLK = 16 MHz 时的音频频率精度

I2SCLK（MHz）	数据长度	I2SDIV	I2SODD	MCK	目标 fs（Hz）	实际 fs（Hz）	误差
16	16	2	1	无	96000	100000	4.1667%
16	32	2	0	无	96000	62500	34.890%
16	16	4	1	无	48000	50000	4.167%
16	32	2	1	无	48000	50000	4.167%
16	16	5	1	无	44100	45454.545	3.0715%
16	32	3	0	无	44100	41666.67	5.518%
16	16	8	0	无	32000	31250	2.344%
16	32	4	0	无	32000	31250	2.344%
16	16	11	1	无	22050	21739.13	1.410%
16	32	5	1	无	22050	22727.27	3.071%
16	16	15	1	无	16000	16129.032	0.806%
16	32	15	1	无	16000	15625	2.344%
16	16	22	1	无	11025	11111.111	0.781%
16	32	11	1	无	11025	10869.57	1.409%
16	16	31	1	无	8000	7936.51	0.794%
16	32	15	1	无	8000	8064.52	0.806%
16	16	N/A	N/A	有	48000	N/A	
16	32	N/A	N/A	有	48000	N/A	
16	16	N/A	N/A	有	44100	N/A	
16	32	N/A	N/A	有	44100	N/A	
16	16	N/A	N/A	有	32000	N/A	
16	32	N/A	N/A	有	32000	N/A	
16	16	3	0	有	22050	N/A	
16	32	3	0	有	22050	N/A	
16	16	2	0	有	16000	15625	2.3438%
16	32	2	0	有	16000	15625	2.3438%
16	16	3	0	有	11025	10416.67	5.518%
16	32	3	0	有	11025	10416.67	5.518%
16	16	4	0	有	8000	7812.5	2.344%
16	32	4	0	有	8000	7812.5	2.344%

22.7.4 I2S 主模式

I^2S 可配置为主模式。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。是否输出主时钟（MCK）由 SPIx_I2SPR 寄存器中的 MCKOE 位控制。

步骤

1. 选择 SPIx_I2SPR 寄存器的 I2SDIV[7:0]位，以定义串行时钟波特率，从而达到适当的音频采样频率。SPIx_I2SPR 寄存器的 ODD 位也需要设置。
2. 选择 CKPOL 位，定义通信时钟的稳定电平。如需为外部 DAC/ADC 音频元件提供主时钟 MCK，则置位 SPIx_I2SPR 寄存器中的 MCKOE 位（应基于 MCK 输出的状态计算 I2SDIV 和 ODD 值，请参见第 22.7.3 节“时钟发生器”了解更多信息）。

3. 置位 SPIx_I2SCFGR 寄存器中的 I2SMOD 位以激活 I2S 功能，通过 I2SSTD[1:0]和 PCMSYNC 位选择 I2S 标准，通过 DATLEN[1:0]位选择数据长度，并通过配置 CHLEN 位选择每个通道的比特位数。另外，通过 SPIx_I2SCFGR 寄存器中的 I2SCFG[1:0]位选择 I2S 主模式和方向（发射器或接收器）。
4. 如果需要，通过对 SPIx_CR2 寄存器执行写操作来选择所有可能的中断源和 DMA 功能。
5. SPIx_I2SCFGR 寄存器的 I2SE 位必须置 1。

WS 和 CK 配置为输出模式。如果 SPIx_I2SPR 的 MCKOE 位置 1，则 MCK 也是输出。

发送序列

将半字写入发送缓冲区后，发送序列随即开始。

假设写入 Tx 缓冲区的第一个数据对应于左通道数据。数据从发送缓冲区传输到移位寄存器时，TXE 置 1，并且必须将对应于右通道的数据写入发送缓冲区。CHSIDE 标志指示将发送的数据对应的通道。TXE 标志置 1 时，CHSIDE 标志有意义，因为该标志在 TXE 变为高电平时进行更新。

一个完整帧表示先进行左通道数据发送再进行右通道数据发送。不存在仅发送左通道的部分帧。

首位发送期间，数据按半字并行加载到 16 位移位寄存器中，然后以串行方式移位并输出到 MOSI/SD 引脚（MSB 在前）。每次数据从发送缓冲区传输到移位寄存器后，TXE 标志都将置 1，如果 SPIx_CR2 寄存器的 TXEIE 位置 1，将产生中断。

有关取决于所选 I²S 标准模式的写操作的详细信息，请参见第 22.7.2 节“支持的音频协议”。

为确保连续进行音频数据发送，必须在当前数据发送结束前将下一个要发送的数据写入 SPIx_DR 寄存器。

如需通过将 I2SE 清零来关闭 I²S，必须等到 TXE = 1 且 BSY = 0。

接收序列

工作模式与发送模式的相同，但第 3 点除外，配置时应通过 I2SCFG[1:0]位设置主设备接收模式。

无论数据或通道长度如何，音频数据始终按 16 位数据包进行接收。这意味着，每当接收缓冲区满时，RXNE 标志即置 1，并且如果 SPIx_CR2 寄存器的 RXNEIE 位置 1，还将产生中断。所接收的右通道或左通道的音频值可通过一次或两次接收操作进入接收缓冲区，具体取决于数据和通道长度配置。

读取 SPIx_DR 寄存器即会使 RXNE 位清零。

CHSIDE 在每次接收后进行更新。它由 I²S 单元生成的 WS 信号触发。

有关取决于所选 I²S 标准模式的读操作的详细信息，请参见第 22.7.2 节“支持的音频协议”。

如果在先前收到的数据尚未读取时又接收到新数据，将产生上溢错误并将 OVR 标志置 1。如果 SPIx_CR2 寄存器的 ERRIE 位置 1，将产生中断以指示该错误。

如需关闭 I²S，需执行特定的操作以确保 I²S 正确地完成了传输周期而没有发起新的数据传输。该序列取决于数据和通道长度的配置，以及所选的音频协议模式。在以下情况下：

- 32 位通道长度上扩展的 16 位数据长度（DATLEN = 00 且 CHLEN = 1），使用 LSB 对齐模式（I2SSTD = 10）
  1. 等待倒数第二个 RXNE = 1（n -1）
  2. 然后等待 17 个 I2S 时钟周期（使用软件循环）
  3. 禁用 I²S（I2SE=0）
- 32 位通道长度上扩展的 16 位数据长度（DATLEN = 00 且 CHLEN = 1），使用 MSB 对齐、I²S 或 PCM 模式（分别为 I2SSTD = 00、I2SSTD = 01 或 I2SSTD = 11）
  1. 等待最后一个 RXNE
  2. 然后等待 1 个 I²S 时钟周期（使用软件循环）
  3. 禁用 I²S（I2SE=0）

- 对于 DATLEN 和 CHLEN 的所有其他组合，无论通过 I²S STD 位选择了哪种音频模式，都应执行以下序列以关闭 I²S：
  1. 等待倒数第二个 RXNE = 1（n -1）
  2. 然后等待 1 个 I²S 时钟周期（使用软件循环）
  3. 禁用 I²S（I2SE =0）

注意： *传输期间，BSY 标志保持低电平。*

22.7.5 I2S 从模式

对于从设备配置，I²S 可配置为发送或接收模式。工作模式遵循的规则大多与 I²S 主设备配置的规则相同。在从模式下，没有要通过 I²S 接口生成的时钟。时钟和 WS 信号是来自连接到 I²S 接口的外部主设备的输入。这样，用户便不需要配置时钟。

应遵循如下配置步骤：

1. 置位 SPIx_I2SCFGR 寄存器中的 I2SMOD 位以选择 I²S 模式，通过 I2SSTD[1:0]位选择 I²S 标准，通过 DATLEN[1:0]位选择数据长度，并通过配置 CHLEN 位选择帧的每个通道的比特位数。另外，通过 SPIx_I2SCFGR 寄存器中的 I2SCFG[1:0]位选择从设备的模式（发送或接收）。
2. 如果需要，通过对 SPIx_CR2 寄存器执行写操作来选择所有可能的中断源和 DMA 功能。
3. SPIx_I2SCFGR 寄存器的 I2SE 位必须置 1。

发送序列

当外部主设备发送时钟并且通过 NSS_WS 信号请求传输数据时，发送序列开始。必须首先使能从设备，然后外部主设备才能开始通信。在主设备发起通信之前，必须加载 I2S 数据寄存器。

对于 I²S、MSB 对齐和 LSB 对齐模式，要写入数据寄存器的第一个数据项对应于左通道的数据。通信开始时，数据从发送缓冲区传输到移位寄存器。TXE 标志随后置 1，以请求将右通道数据写入 I²S 数据寄存器。

CHSIDE 标志指示将发送的数据对应的通道。与主发送模式相比，在从模式下，CHSIDE 由来自外部主设备的 WS 信号触发。这意味着，从设备需要首先为发送第一个数据做好准备，然后主设备才能产生时钟。WS 置位意味着首先发送左通道数据。

注意： *必须要在主设备发出的第一个时钟出现在 CK 线上至少 2 个 PCLK 周期之前置位 I2SE。*

首位发送期间，数据按半字从内部总线并行加载到 16 位移位寄存器中，然后以串行方式移位并输出到 MOSI/SD 引脚（MSB 在前）。每次数据从发送缓冲区传输到移位寄存器后，TXE 标志都将置 1，如果 SPIx_CR2 寄存器的 TXEIE 位置 1，将产生中断。

请注意，仅当 TXE 标志为 1 时，才可以尝试向发送缓冲区写入数据。

有关取决于所选 I²S 标准模式的写操作的详细信息，请参见第 22.7.2 节“支持的音频协议”。

为确保连续进行音频数据发送，必须在当前数据发送结束前将下一个要发送数据写入 SPIx_DR 寄存器。如果在数据尚未写入 SPIx_DR 寄存器时下一个数据通信的首个时钟边沿到来，下溢标志将置 1 并可能产生中断。通过这种方式，软件可以获知所传输的数据不正确。如果 SPIx_CR2 寄存器的 ERRIE 位置 1，则当 SPIx_SR 寄存器中的 UDR 标志变为 1 时，将产生中断。在这种情况下，必须关闭 I²S 并重启从左通道开始的数据传输。

如需通过将 I2SE 位清零来关闭 I²S，必须等到 TXE = 1 且 BSY = 0。

接收序列

工作模式与发送模式的相同，但第 1 点除外，配置时应通过 SPIx_I2SCFGR 寄存器中的 I2SCFG[1:0]位设置主设备接收模式。

无论数据长度或通道长度如何，音频数据始终按 16 位数据包进行接收。这意味着，每当接收缓冲区填满时，SPIx_SR 寄存器中的 RXNE 标志即置 1，并且如果 SPIx_CR2 寄存器的 RXNEIE 位置 1，还将产生中断。所接收的右通道或左通道的音频值可能通过一次或两次接收操作进入接收缓冲区，具体取决于数据长度和通道长度配置。

每次接收要从 SPIx_DR 寄存器读取的数据时，CHSIDE 标志都将更新。该标志由外部主设备所管理的外部 WS 线路触发。

读取 SPIx_DR 寄存器即会使 RXNE 位清零。

有关取决于所选 I²S 标准模式的读操作的详细信息，请参见第 22.7.2 节“支持的音频协议”。

如果在先前收到的数据尚未读取时又接收到新数据，将产生上溢错误，OVR 标志将置 1。如果 SPIx_CR2 寄存器的 ERRIE 位置 1，将产生中断以指示该错误。

如需在接收模式下关闭 I2S，必须在接收到最后一个 RXNE = 1 后立即将 I2SE 清零。

注意： *外部主设备应能够通过音频通道以 16 位或 32 位数据包发送/接收数据。*

22.7.6 I2S 错误标志

下溢标志（UDR）

I²S 单元有三个错误标志。

在从发送模式下，如果在软件尚未将任何值加载到 SPIx_DR 之前出现第一个数据发送时钟，此标志将置 1。当 SPIx_I2SCFGR 寄存器中的 I2SMOD 位置 1 时，此标志可用。如果 SPIx_CR2 寄存器中的 ERRIE 位置 1，可产生中断。

UDR 位通过 SPIx_SR 寄存器上的读操作进行清零。

上溢标志（OVR）

如果在尚未从 SPIx_DR 寄存器读取上一个数据时又接收到新数据，此标志将置 1。因此，传入的数据将丢失。如果 SPIx_CR2 寄存器中的 ERRIE 位置 1，可产生中断。

这种情况下，将不会用接收到的新数据更新接收缓冲区的内容。对 SPIx_DR 寄存器执行的读操作将返回先前正确接收的数据。主设备后续发送的所有其它半字都将丢失。

要将 OVR 位清零，应首先对 SPIx_DR 寄存器执行读操作，然后再对 SPIx_SR 寄存器进行读访问。

帧错误标志（FRE）

仅当 I2S 配置为从模式时，此标志才可由硬件置 1。如果外部主设备没有按照从设备期望的那样改变 WS 线，则此标志将置 1。如果失去同步，要从此状态中恢复并将外部主设备与 I2S 从设备重新同步，需执行下列步骤：

1. 禁用 I²S。
2. 在 WS 线上检测到正确的电平时将其重新使能（WS 在 I2S 模式下为高电平，在 MSB 对齐、LSB 对齐或 PCM 模式下为低电平）。

主设备与从设备之间的同步失效可能是由于 SCK 通信时钟或 WS 帧同步信号线上存在噪音干扰。如果 ERRIE 位置 1，可产生错误中断。读取状态寄存器时，同步失效标志（FRE）由软件清零。

22.7.7 DMA 特性

在 I²S 模式下，DMA 的工作方式与在 SPI 模式下完全相同。除了在 I²S 模式下 CRC 特性不可用（因为没有数据传输保护系统）之外，没有任何差异。

22.8 I2S 中断

表 80. I²S 中断请求提供了 I2S 中断的列表。

表 80. I²S（中断请求）

中断事件	事件标志	使能控制位
发送缓冲区为空	TXE	TXEIE
接收缓冲区非空	RXNE	RXNEIE
上溢错误	OVR	ERRIE
下溢错误	UDR	
帧错误	FRE	

22.9 SPI 和 I²S 寄存器

外设寄存器可支持半字（16 位）或字（32 位）访问。此外，SPI_DR 可支持 8 位访问。

22.9.1 SPIx 控制寄存器 1（SPIx_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
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。该位通过一条共用的双向数据线来支持半双工通信。双向模式激活时，使 RXONLY 位保持清零状态。 0：选择了 2 线单向数据模式 1：选择了 1 线双向数据模式
	注意：该位不适用于 I2S 模式。
位 14	**BIDIOE**：双向通信模式下的输出使能。 该位结合 BIDIMODE 位，用于选择双向模式下的传输方向。 0：禁止输出（只接收模式） 1：使能输出（只发送模式） 注意：在主模式下，使用 MOSI 引脚；在从模式下，使用 MISO 引脚。 该位不适用于 I2S 模式。
位 13	**CRCEN**：硬件 CRC 计算使能。 0：禁止 CRC 计算 1：使能 CRC 计算 注意：为确保正确操作，只应在禁止 SPI（SPE = “0”）时对此位执行写操作。 该位不适用于 I2S 模式。
位 12	**CRCNEXT**：下一个发送 CRC。 0：下一个发送值来自发送缓冲区 1：下一个发送值来自发送 CRC 寄存器 注意：向 SPIx_DR 寄存器写入最后一个数据后，必须立即对该位执行写操作。 该位不适用于 I2S 模式。
位 11	**CRCL**：CRC 长度。 此位由软件置 1 和清零，用以选择 CRC 长度。 0：8 位 CRC 长度。

	1：16 位 CRC 长度 注意：为确保正确操作，只应在禁止 SPI（SPE = “0”）时对此位执行写操作。 该位不适用于 I2S 模式。
位 10	**RXONLY**：只接收模式使能。 该位用于使能通过一条双向线专门接收数据的单工通信。当只接收模式激活时，将 BIDIMODE 位保持清零状态。此位也适用于多从模式系统，在此类系统中，不会访问特定从设备，也不会损坏访问的从设备的输出。 0：全双工（发送和接收） 1：禁止输出（只接收模式）
	注意：该位不适用于 I2S 模式。
位 9	**SSM**：软件从设备管理。 当 SSM 位置 1 时，NSS 引脚输入替换为 SSI 位的值。 0：禁止软件从设备管理 1：使能软件从设备管理 注意：该位不适用于 I^2S 模式和 SPI TI 模式。
位 8	**SSI**：内部从选择。 仅当 SSM 位置 1 时，此位才有效。此位的值将作用到 NSS 引脚上，并忽略 NSS 引脚的 I/O 值。 注意：该位不适用于 I^2S 模式和 SPI TI 模式。
位 7	**LSBFIRST**：帧格式。 0：发送/接收数据时 MSB 在前 1：发送/接收数据时 LSB 在前 注意：1. 当通信正在进行时，不应更改此位。 2. 该位不适用于 I^2S 模式和 SPI TI 模式。
位 6	**SPE**：使能 SPI。 0：禁止外设 1：使能外设 注意：在禁用 SPI 时，按照禁用 SPI 的程序中所述的步骤操作。 该位不适用于 I2S 模式。
位 5:3	**BR[2:0]**：波特率控制。 000：$f_{PCLK}/2$ 001：$f_{PCLK}/4$ 010：$f_{PCLK}/8$ 011：$f_{PCLK}/16$ 100：$f_{PCLK}/32$ 101：$f_{PCLK}/64$ 110：$f_{PCLK}/128$ 111：$f_{PCLK}/256$ 注意：当通信正在进行时，不应更改这些位。 该位不适用于 I2S 模式。
位 2	**MSTR**：主设备选择。 0：从设备配置 1：主设备配置 注意：当通信正在进行时，不应更改此位。

	该位不适用于 I2S 模式。
Bit1	**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 1 注意：当通信正在进行时，不应更改此位。 该位不适用于 I²S 模式和 SPI TI 模式。
位 0	**CPHA**：时钟相位。 0：从第一个时钟边沿开始采样数据 1：从第二个时钟边沿开始采样数据 注意：当通信正在进行时，不应更改此位。 该位不适用于 I²S 模式和 SPI TI 模式。

22.9.2 SPIx 控制寄存器 2（SPIx_CR2）

偏移地址：0x04

复位值：0x0700

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	LDMA_TX	LDMA_RX	FRXT H	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，必须保持复位值。
位 14	**LDMA_TX**：发送时的最后一次 DMA 传输。 此位用于数据封装模式，用于定义通过 DMA 发送的数据总数为奇数还是偶数。只有 SPIx_CR2 寄存器中的 TXDMAEN 位置 1 且使用封装模式（数据长度 =< 8 位，对 SPIx_DR 的写访问为 16 位宽）时，该位才有意义。当 SPI 禁止时（SPIx_CR1 寄存器中的 SPE = 0），必须对其执行写操作。 0：待传输数据数量为偶数 1：待传输数据数量为奇数 注意：如果 CRCEN 位置 1，请参见禁用 SPI 的步骤。 该位不适用于 I2S 模式。
位 13	**LDMA_RX**：接收时的最后一次 DMA 传输。 此位用于数据封装模式，用于定义通过 DMA 接收的数据总数为奇数还是偶数。只有 SPIx_CR2 寄存器中的 RXDMAEN 位置 1 且使用封装模式（数据长度 =< 8 位，对 SPIx_DR 的写访问为 16 位宽）时，该位才有意义。当 SPI 禁止时（SPIx_CR1 寄存器中的 SPE = 0），必须对其执行写操作。 0：待传输数据数量为偶数 1：待传输数据数量为奇数 注意：如果 CRCEN 位置 1，请参见禁用 SPI 的步骤。 该位不适用于 I2S 模式。
位 12	**FRXTH**：FIFO 接收阈值。 根据对 FIFO 的读访问（16 位或 8 位）将 FRXTH 置位。 该位用于设置触发 RXNE 事件的 RXFIFO 阈值。 0：如果 FIFO 占用水平大于或等于 1/2（16 位），将生成 RXNE 事件 1：如果 FIFO 占用水平大于或等于 1/4（8 位），将生成 RXNE 事件

位 12	注意：该位不适用于 I²S 模式。
位 11:8	**DS [3:0]**：数据大小。 这些位用于配置 SPI 传输的数据长度： 0000：未使用 0001：未使用 0010：未使用 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 如果软件尝试写入其中一个“未使用”值，这些位将被强制设为“0111”（8 位）。
	注意：该位不适用于 I2S 模式。
位 7	**TXEIE**：发送缓冲区空中断使能。 0：屏蔽 TXE 中断 1：使能 TXE 中断。TXE 标志置 1 时产生中断请求。
位 6	**RXNEIE**：接收缓冲区非空中断使能。 0：屏蔽 RXNE 中断 1：使能 RXNE 中断。RXNE 标志置 1 时产生中断请求。
位 5	**ERRIE**：错误中断使能。 此位用于在出现错误条件（SPI 模式下的 CRCERR、OVR 和 MODF；TI 模式下的 FRE；以及 I2S 模式下的 UDR、OVR 和 FRE）时控制中断的生成。 0：屏蔽错误中断 1：错误中断使能
位 4	**FRF**：帧格式。 0：SPI Motorola 模式 1：SPI TI 模式 注意：只有在禁止 SPI（SPE=0）后才能对此位执行写操作。 该位不适用于 I2S 模式。
位 3	**NSSP**：NSS 脉冲管理。 此位仅用于主模式。连续传输时，该位允许 SPI 在两个连续数据间生成 NSS 脉冲。单次数据传输时，该位强制 NSS 引脚在传输后变为高电平。 如果 CPHA =“1”或 FRF =“1”，该位无意义。 0：无 NSS 脉冲 1：生成 NSS 脉冲

	注意： 1. 只有在禁止 SPI（SPE=0）后才能对此位执行写操作。 2. 该位不适用于 I²S 模式和 SPI TI 模式。
位 2	**SSOE**：SS 输出使能。 0：在主模式下禁止 SS 输出，SPI 接口可在多主模式配置下工作 1：在主模式下以及使能 SPI 接口后使能 SS 输出。SPI 接口不能在多主模式环境下工作。 注意：该位不适用于 I²S 模式和 SPI TI 模式。
位 1	**TXDMAEN**：发送缓冲区 DMA 使能。 当此位置 1 时，每当 TXE 标志置 1，即产生 DMA 请求。 0：关闭发送缓冲区 DMA 1：使能发送缓冲区 DMA
位 0	**RXDMAEN**：接收缓冲区 DMA 使能。 当此位置 1 时，每当 RXNE 标志置 1，即产生 DMA 请求。 0：关闭接收缓冲区 DMA 1：使能接收缓冲区 DMA

22.9.3 SPI 状态寄存器（SPIx_SR）

偏移地址：0x08

复位值：0x0002

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	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，必须保持复位值。
位 12:11	**FTLVL[1:0]**：FIFO 发送水平。 这些位将由硬件置 1 和清零。 00：FIFO 为空 01：1/4 FIFO 10：1/2 FIFO 11：FIFO 为满（当 FIFO 阈值大于 1/2 时即视为满） 注意：这些位不适用于 I2S 模式。
位 10:9	**FRLVL[1:0]**：FIFO 接收水平。 这些位将由硬件置 1 和清零。 00：FIFO 为空 01：1/4 FIFO 10：1/2 FIFO 11：FIFO 已满 注意：使能 CRC 计算时，这些位不适用于 I2S 模式和 SPI 仅接收模式。
位 8	**FRE**：帧格式错误。 在 TI 从模式和 I2S 从模式下，此标志供 SPI 使用。请参见第 22.5.10 节“SPI 错误标志”和第 22.7.6 节“I2S 错误标志”。

	此标志由硬件置 1，在读取 SPIx_SR 时由软件复位。 0：无帧格式错误 1：发生帧格式错误
位 7	**BSY**：忙标志。 0：SPI（或 I2S）不忙 1：SPI（或 I2S）忙于通信或者发送缓冲区不为空。此标志由硬件置 1 和清零。 注意：使用 BSY 标志时须小心：参见第 22.5.9 节“SPI 状态标志”。
位 6	**OVR**：上溢标志。 0：未发生上溢 1：发生上溢 此标志由硬件置 1，由软件序列复位。参见第 22.7.6 节“I2S 错误标志”了解软件系列。
位 5	**MODF**：模式故障。 0：未发生模式故障 1：发生模式故障 此标志由硬件置 1，由软件序列复位。
位 4	**CRCERR**：CRC 错误。 0：接收到的 CRC 值与 SPIx_RXCRCR 值匹配 1：接收到的 CRC 值与 SPIx_RXCRCR 值不匹配 此标志由硬件置 1，通过软件写入 0 来清零。
位 3	**UDR**：下溢标志。 0：未发生下溢 1：发生下溢 此标志由硬件置 1，由软件序列复位。参见第 22.7.6 节“I2S 错误标志”了解软件系列。 注意：该位不适用于 SPI 模式。
位 2	**CHSIDE**：通道侧。 0：发送或接收左通道信息 1：发送或接收右通道信息 注意：该位不适用于 SPI 模式。该位在 PCM 模式下无意义。
位 1	**TXE**：发送缓冲空。 0：Tx 缓冲区中无更多空间（软件不应向 Tx 缓冲区写入数据）。 1：Tx 缓冲区中有至少一个空间（软件可以向 Tx 缓冲区写入数据）。
位 0	**RXNE**：接收缓冲区非空。 0：Rx 缓冲区为空 1：接收缓冲区非空

22.9.4 SPI 数据寄存器（SPIx_DR）

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
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。 已接收或者要发送的数据。 数据寄存器用于连接 Rx 和 Tx FIFO。在读取数据寄存器时，在对数据寄存器的写操作访问 TxFIFO 的同时访问 RxFIFO（参见第 22.5.8 节“数据发送和接收过程”）。 注意：数据始终右对齐。写入寄存器时将忽略未使用位，读取寄存器时会将未使用位读为 0。Rx 阈值设置必须始终与当前使用的读访问相符。

22.9.5 SPI CRC 多项式寄存器（SPIx_CRCPR）

偏移地址：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。 此寄存器包含用于 CRC 计算的多项式。 CRC 多项式（0007h）是此寄存器的复位值。可根据需要配置另一个多项式。

注意： *多项式值只应为奇数。不支持任何偶数值。*

22.9.6 SPI 接收 CRC 寄存器（SPIx_RXCRCR）

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
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]**：接收 CRC 寄存器。 使能 CRC 计算后，RxCRC[15:0]位将包含后续接收字节在计算后所得到的 CRC 值。当 SPIx_CR1 寄存器中的 CRCEN 位写入 1 时，此寄存器复位。CRC 通过 SPIx_CRCPR 寄存器中编程的多项式连续计算。 数据帧格式设置为 8 位数据（SPIx_CR1 的 CRCL 位清零）时，仅考虑 8 个 LSB 位。CRC 计算依据任意 CRC8 标准进行。 选择 16 位数据帧格式（SPIx_CR1 寄存器的 CRCL 位置 1）时，考虑此寄存器的全部 16 个位。CRC 计算依据任意 CRC16 标准进行。 当 BSY 标志置 1 时，读取此寄存器可能返回一个不正确的值。

22.9.7 SPI 发送 CRC 寄存器（SPIx_TXCRCR）

偏移地址：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]**：发送 CRC 寄存器。 使能 CRC 计算后，TxCRC[7:0]位将包含后续发送字节在计算后所得到的 CRC 值。当 SPIx_CR1 寄存器中的 CRCEN 位写入 1 时，此寄存器复位。CRC 通过 SPIx_CRCPR 寄存器中编程的多项式连续计算。 数据帧格式设置为 8 位数据（SPIx_CR1 的 CRCL 位清零）时，仅考虑 8 个 LSB 位。CRC 计算依据任意 CRC8 标准进行。 选择 16 位数据帧格式（SPIx_CR1 寄存器的 CRCL 位置 1）时，考虑此寄存器的全部 16 个位。CRC 计算依据任意 CRC16 标准进行。 注意：当 BSY 标志置 1 时，读取此寄存器可能返回一个不正确的值。这些位不适用于 I2S 模式。

22.9.8 SPIx_I2S 配置寄存器（SPIx_I2SCFGR）

偏移地址：0x1C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	ASTR TEN	I2SMOD	I2SE	I2SCFG		PCMSYNC	Res.	I2SSTD		CKPOL	DATLEN		CHLEN
			rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw

位 15:13	保留：由硬件强制为零。
位 12	**ASTRTEN**：异步启动使能。 0：异步启动禁用。在从模式下使能 I^2S 后，若接收到 I^2S 时钟并且在 WS 信号上检测到适当的边沿（取决于所选的协议），I^2S 从设备将启动传输。 1：异步启动使能。在从模式下使能 I^2S 后，若从主设备接收到 I^2S 时钟，I^2S 从设备将立即启动传输，而无需在 WS 信号上检测到预期的边沿。 注意：使用 I^2S Philips 标准时，适当的边沿为 WS 信号的下降沿；使用其它标准时，则为上升沿。
位 11	**I2SMOD**：I^2S 模式选择。 0：选择 SPI 模式 1：选择 I^2S 模式 注意：应在 SPI 禁止时配置此位。
位 10	**I2SE**：I^2S 使能。 0：I^2S 外设禁用 1：I^2S 外设使能 注意：该位不适用于 SPI 模式。
位 9:8	**I2SCFG**：I^2S 配置模式。 00：从模式 - 发送 01：从模式 - 接收 10：主模式 - 发送 11：主模式 - 接收 注意：应在 I^2S 禁止时配置这些位。 这些位不适用于 SPI 模式。

位 7	**PCMSYNC**：PCM 帧同步。 0：短帧同步 1：长帧同步 注意：只有在 I2SSTD = 11（使用 PCM 标准）时，此位才有意义。 该位不适用于 SPI 模式。
位 6	保留：由硬件强制为零。
位 5:4	**I2SSTD**：I²S 标准选择。 00：I²S Philips 标准。 01：MSB 对齐标准（左对齐） 10：LSB 对齐标准（右对齐） 11：PCM 标准 请参见第 22.7.2 节“支持的音频协议”了解关于 I²S 标准的更多信息。 注意：为确保正确操作，在 I²S 禁用时应配置这些位。 这些位不适用于 SPI 模式。
位 3	**CKPOL**：稳定状态时钟极性。 0：I²S 时钟稳定状态为低电平 1：I²S 时钟稳定状态为高电平 注意：为确保正确操作，在 I²S 禁用时应配置此位。 该位不适用于 SPI 模式。
位 2:1	**DATLEN**：传输的数据长度。 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 注意：为确保正确操作，在 I²S 禁用时应配置这些位。 这些位不适用于 SPI 模式。
位 0	**CHLEN**：通道长度（每个音频通道的位数）。 0：16 位宽度 1：32 位宽度 只有在 DATLEN = 00 时，此位的写操作才有意义，否则无论写入何值，通道长度始终由硬件固定为 32 位。 注意：为确保正确操作，在 I²S 禁用时应配置此位。 该位不适用于 SPI 模式。

22.9.9 SPIx_I2S 预分频器寄存器（SPIx_I2SPR）

偏移地址：0x20

复位值：0x0002

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	MCKOE	ODD	I2SDIV							
						rw	rw	rw							

位 15:10	保留：由硬件强制为零。
位 9	**MCKOE**：主时钟输出使能。 0：禁止主时钟输出 1：使能主时钟输出 注意：应在 I2S 禁止时配置此位。仅在 I2S 处于主模式时使用此位。 该位不适用于 SPI 模式。
位 8	**ODD**：预分频器的奇数因子。 0：实际分频值 = I2SDIV *2 1：实际分频值 =（I2SDIV * 2）+1。请参见第 22.7.3 节“时钟发生器”。 注意：应在 I2S 禁止时配置此位。仅在 I2S 处于主模式时使用此位。 该位不适用于 SPI 模式。
位 7:0	**I2SDIV**：I^2S 线性预分频器。 I2SDIV [7:0] = 0 或 I2SDIV [7:0] = 1 为禁用值。请参见第 22.7.3 节“时钟发生器”。 注意：应在 I^2S 禁止时配置这些位。仅在 I2S 处于主模式时使用这些位。 这些位不适用于 SPI 模式。

22.10 SPI/I2S 寄存器映射

偏移	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
0x00	**SPIx_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BIDIMODE	BIDIOE	CRCEN	CRCNEXT	CRCL	RXONLY	SSM	SSI	LSBFIRST	SPE	BR[2:0]			MSTR	CPOL	CPHA
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	**SPIx_CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LDMA_TX	LDMA_RX	FRXTH	DS[3:0]				TXEIE	RXNEIE	ERRIE	FRF	NSSP	SSOE	TXDMAEN	RXDMAEN
	复位值																		0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
0x08	SPIx_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FTLVL[1:0]		FRLVL[1:0]		FRE	BSY	OVR	MODF	CRCERR	UDR	CHSIDE	TXE	RXNE
	复位值																				0	0	0	0	0	0	0	0	0	0	0	1	0
0x0C	**SPIx_DR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DR[15:0]															
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	**SPIx_CRCPR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRCPOLY[15:0]															
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

偏移	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
0x14	**SPIx_RXCRCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RxCRC[15:0]															
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	**SPIx_TXCRCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TxCRC[15:0]															
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	**SPIx_I2SCFGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ASTRTEN	I2SMOD	I2SE	I2SCFG		PCMSYNC	Res.	I2SSTD		CKPOL		DATLEN		CHLEN
	复位值																				0	0	0	0	0	0		0	0	0	0	0	0
0x20	**SPIx_I2SPR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCKOE	ODD	I2SDIV							
	复位值																							0	0	0	0	0	0	0	0	1	0

请参见“表 3. BlueNRG-LP 存储器映射和外设寄存器边界地址”。

23 调试支持（DBG）

23.1 SWD 调试特性

BlueNRG-LP 器件 JTAG ID[31:0]如下：

- 0000 0010000000011110 00000100000 1
- （0x0201_E041）

BlueNRG-LP 的 Cortex-M0+子系统内置四个断点和两个观察点。

BlueNRG-LP 器件内置：

- ARM 串行线调试端口，使串行线调试（2 线）能够连接到 CPU（默认为上电复位后）。

注意： *当器件进入 DEEPSTOP 模式时，SWD 调试端口不通电。因此，调试访问会被禁用且无法通过 SWD 通道访问芯片。一个可能的恢复选项是通过 PA10 引脚激活内部的嵌入式 UART 自举程序（只需在硬件复位期间强制 PA10 为高电平）。*

24 射频子系统（MR_BLE）

BlueNRG-LP 内部的射频子系统（亦即 MR_BLE）负责管理蓝牙特性。

如需 MR_BLE/射频子系统的相关信息，请参见“BlueNRG-LP 无线电控制器”参考手册（RM0480）。

表 81. MR_BLE/BlueNRG-LP 的依赖关系提供了关于 BlueNRG-LP 器件内部 IP 集成的依赖关系的信息。

表 81. MR_BLE/BlueNRG-LP 的依赖关系

参数/特性	BlueNRG-LP 内部实现
支持的特性	
支持 Bluetooth®标准 5.2	有
支持 802.15.4 标准	无
基址	
GlobalStatMach RAM 表基址	0x2000 00C0
RRM 寄存器基址	0x6000 1400
无线电控制器寄存器基址	0x6000 1000
BLE APB 寄存器基址	0x6000 0000
WAKEUP 寄存器基址	0x6000 1800
中断映射	
irq_BLE_int1	BLE_TXRX/线路 18
irq_BLE_int2	BLE_AES/线路 19
irq_BLE_int3	BLE_DBG/线路 20
Irq_mr_syst（亦即 irq_radio_control）	RADIO_CTRL/线路 21
irq_rrm	MR_BLE/线路 22
irq_radio_fsm	MR_BLE/线路 22
irq_wakeup_cpu	CPU_WKUP/线路 23
irq_wakeup_ble	BLE_WKUP/线路 24
PWRC 连接	
睡眠模式请求	只支持模式 0
RCC 连接	
高速时钟	通过 RCC.APB2ENR 寄存器（MRBLEEN 位）实现软件使能/禁用。 通过 RCC.APB2ENR 寄存器（CLKBLEDIV 位字段）管理时钟频率。
慢速时钟	通过 RCC.RCC_CR 寄存器实现软件使能/禁用。 通过 RCC.RCC_CFGR 寄存器选择时钟源。 注：此时钟源为 BlueNRG-LP 器件中所有使用慢速时钟的 IP 所共用
MR_BLE 复位	通过 RCC.APB2RSTR 寄存器实现软件单独复位使能/禁用

关于系统时钟快于 MR_BLE 时钟时中断清零的警告

当系统时钟频率快于 MR_BLE 时钟时，软件必须注意 MR_BLE 中断清零的管理。

实际上，在该配置下，从 CPU 到 MR_BLE 标志寄存器的 AHB 至 APB 传输经过 APB 下变频器（负责在存在频率差异的情况下保证合适的 AHB 协议）。

此模块会增加一些传输延时，AHB 总线上的写操作 HREADY（忙碌）信息没有升为高电平。

因此，当中断在 MR_BLE 内部仍然有效时，在软件层面上“清零标志写操作”之后的指令仍然可以运行（写操作尚未内部传播）。

软件必须注意，在写操作后，可能存在至多 10 个周期的有效间隙。这意味着不得在中断处理程序刚结束时执行清零标志操作。

此限制涉及 MR_BLE 的所有中断线路：

- 无线电控制器中断（慢速时钟测量）= 线路 21
- RRM 和无线电 FSM 中断 = 线路 22
- CPU 唤醒中断 = 线路 23
- BLE 唤醒中断 = 线路 24
- BLE 发送/接收中断 = 线路 18
- BLE AES 中断 = 线路 19
- BLE 调试中断 = 线路 20

一个选项是在中断处理程序的早期将标志清零以屏蔽系统延时。最安全的选项是在写操作后读取标志寄存器。这可以保证写操作已被推入 MR_BLE 寄存器内部。

版本历史

表 82. 文档版本历史

日期	版本	变更
2020 年 7 月 6 日	1	初始版本。
2021 年 7 月 5 日	2	更新了表 5. GPIO 复用选项 AF0 - AF2 和表 6. GPIO 复用选项 AF3 - AF4。 更新了第 12 节“模数转换器（ADC）”及其所有后续章节。 增加了对 AF5、AF6 和 AF7 GPIO 复用选项的说明。删除了对 JTAG 调试支持的引用。 增加了 TX_SEQUENCE 和 RX_SEQUENCE 的缺失的复用功能和相关注释。 更新了对 LPMS/关断的说明。 增加了缺失的唤醒源（PB0 至 PB11）。 更新了第 22 节“串行外设接口/集成电路内置音频总线（SPI/I2S）”及其所有后续章节。 将 AF 表中 USART_TX、USART_RX、LPUART_RX 和 LPUART_TX 的类型从 I 或 O 更新为 I/O。 更新了第 20.8 节“USART 寄存器映射”。更新了第 9.4.6 节“SIZE 寄存器” 增加了第 9.5.7 节“页面写保护示例”和第 24 节“射频子系统（MR_BLE）”。
2022 年 4 月 1 日	3	更新了第 5.4.2 节“DEEPSTOP 模式”、第 7.4.4 节“GPIO 端口上/下拉寄存器（GPIOx_PUPDR）（x = A, B）”、“图 82. 独立看门狗框图”和第 23.1 节“SWD 调试特性”，并对整个文档进行了少量文本修改。

目录

1 文档约定 2

1.1 寄存器相关缩写词列表 2

1.2 术语表 2

1.3 缩略语 3

2 系统和存储器概述 5

2.1 系统架构 5

2.1.1 S0：CPU（Cortex®-M0+）S 总线 6

2.1.2 S1：DMA 总线 6

2.1.3 S2：射频系统总线 6

2.1.4 总线矩阵 7

2.2 存储器组织结构 7

2.2.1 引言 7

2.2.2 存储器映射和寄存器边界地址 8

2.3 ARM® Cortex®-M0+ 10

2.3.1 CPU 存储器重映射 10

2.3.2 中断 11

3 AHB 上/下变频器 13

3.1 AHB 上/下变频器说明 13

4 I/O 工作模式 15

5 电源控制器（PWRC） 21

5.1 特性 21

5.2 电源域 21

5.3 电源电压监控器 22

5.3.1 上电复位 POR/下电复位（PDR）/欠压复位（BOR） 22

5.3.2 电源电压检测（PVD） 23

5.4 工作模式 23

5.4.1 RUN 模式 23

5.4.2 DEEPSTOP 模式 24

5.4.3 SHUTDOWN 模式 26

5.4.4 工作模式转换管理 28

5.5 SMPS 降压稳压器 28
5.6 PWRC 寄存器说明 31
5.6.1 控制寄存器 1（PWRC_CR1） 31
5.6.2 控制寄存器 2（PWRC_CR2） 32
5.6.3 控制寄存器 3（PWRC_CR3） 33
5.6.4 控制寄存器 4（PWRC_CR4） 34
5.6.5 状态寄存器 1（PWRC_SR1） 35
5.6.6 状态寄存器 2（PWRC_SR2） 37
5.6.7 控制寄存器 5（PWRC_CR5） 38
5.6.8 I/O 端口 A 上拉控制寄存器（PWRC_PUCRA） 39
5.6.9 I/O 端口 A 下拉控制寄存器（PWRC_PDCRA） 40
5.6.10 I/O 端口 B 上拉控制寄存器（PWRC_PUCRB） 43
5.6.11 I/O 端口 B 下拉控制寄存器（PWRC_PDCRB） 44
5.6.12 控制寄存器 6（PWRC_CR6） 45
5.6.13 控制寄存器 7（PWRC_CR7） 47
5.6.14 状态寄存器 3（PWRC_SR3） 48
5.6.15 I/O DEEPSTOP 驱动配置寄存器（PWRC_IOxCFG） 50
5.6.16 调试寄存器（PWRC_DBGR） 52
5.6.17 扩展状态和复位寄存器（PWRC_EXTSRR） 52
5.7 PWRC 寄存器映射 53
5.8 编程模型 55
5.8.1 复位原因管理 55
5.8.2 SMPS 输出电平重新编程 56
6 复位和时钟控制器（RCC） 57
6.1 复位管理 57
6.1.1 概述 57
6.1.2 电源复位 58
6.1.3 看门狗复位 58
6.1.4 LOCKUP 复位 58
6.1.5 系统复位请求 58
6.1.6 退出 DEEPSTOP 58
6.2 时钟管理 58

6.2.1 系统时钟详情 59
6.2.2 外设时钟详情 59
6.2.3 慢速时钟频率详情 62
6.3 使用 MR_BLE 时的系统频率切换 62
6.4 外部引脚上的时钟观测 62
6.5 其他情况 63
6.5.1 IO BOOSTER 63
6.6 RCC 寄存器说明 63
6.6.1 时钟源控制寄存器（RCC_CR） 64
6.6.2 时钟配置寄存器（RCC_CFGR） 66
6.6.3 时钟中断使能寄存器（RCC_CIER） 68
6.6.4 时钟中断标志寄存器（RCC_CIFR） 69
6.6.5 时钟切换指令寄存器（RCC_CSCMDR） 70
6.6.6 AHB0 宏单元复位寄存器（RCC_AHBRSTR） 71
6.6.7 APB0 宏单元复位寄存器（RCC_APB0RSTR） 72
6.6.8 APB1 宏单元复位寄存器（RCC_APB1RSTR） 73
6.6.9 APB2 宏单元复位寄存器（RCC_APB2RSTR） 74
6.6.10 AHB0 宏单元时钟使能寄存器（RCC_AHBENR） 74
6.6.11 APB0 宏单元时钟使能寄存器（RCC_APB0ENR） 75
6.6.12 APB1 宏单元时钟使能寄存器（RCC_APB1ENR） 76
6.6.13 APB2 宏单元时钟使能寄存器（RCC_APB2ENR） 78
6.6.14 V33 复位状态寄存器（RCC_CSR） 79
6.6.15 RF 软件高速外部寄存器（RCC_RFSWHSECR） 79
6.6.16 Rf 高速外部寄存器（RCC_RFHSECR） 81
6.7 RCC 寄存器映射 82
6.8 编程模型 87
6.8.1 将系统时钟切换到 PLL64M 时钟树 87
6.8.2 使用直接的 HSE（高速外部时钟）代替 RC64MPLL 模块 87
6.8.3 在 MR_BLE 使能时更改系统时钟频率 87
7 通用 I/O（GPIO） 88
7.1 引言 88
7.2 GPIO 主要特性 88

7.3 GPIO 功能说明 88
7.3.1 通用 I/O（GPIO） 90
7.3.2 I/O 引脚复用功能复用器和映射 90
7.3.3 I/O 端口控制寄存器 91
7.3.4 I/O 端口数据寄存器 91
7.3.5 I/O 数据位操作 91
7.3.6 GPIO 锁定机制 92
7.3.7 I/O 复用功能输入/输出 92
7.3.8 外部中断/唤醒线路 92
7.3.9 输入配置 92
7.3.10 输出配置 93
7.3.11 复用功能配置 94
7.3.12 模拟配置 94
7.3.13 使用 LSE 振荡器引脚作为 GPIO 95
7.4 GPIO 寄存器 96
7.4.1 GPIO 端口模式寄存器（GPIOx_MODER）（x = A, B） 96
7.4.2 GPIO 端口输出类型寄存器（GPIOx_OTYPER）（x = A, B） 97
7.4.3 GPIO 端口输出速度寄存器（GPIOx_OSPEEDR）（x = A, B） 97
7.4.4 GPIO 端口上/下拉寄存器（GPIOx_PUPDR）（x = A, B） 98
7.4.5 GPIO 端口输入数据寄存器（GPIOx_IDR）（x = A, B） 98
7.4.6 GPIO 端口输出数据寄存器（GPIOx_ODR）（x = A, B） 99
7.4.7 GPIO 端口位置位/复位寄存器（GPIOx_BSRR）（x = A, B） 99
7.4.8 GPIO 端口配置锁定寄存器（GPIOx_LCKR）（x = A, B） 100
7.4.9 GPIO 复用功能低寄存器（GPIOx_AFRL）（x = A, B） 101
7.4.10 GPIO 复用功能高寄存器（GPIOx_AFRH）（x = A, B） 101
7.5 GPIO 寄存器映射 102
8 系统控制器（SYSCFG） 104
8.1 SYSCFG 主要特性 104
8.2 系统控制器寄存器说明 104
8.2.1 晶片 ID 寄存器（DIE_ID） 104
8.2.2 JTAG ID 寄存器（JTAG_ID） 105
8.2.3 I2C 超快速模式加引脚功能控制寄存器（I2C_FMP_CTRL） 106

8.2.4 I/O 中断检测类型寄存器（IO_DTR） 107
8.2.5 I/O 中断边沿寄存器（IO_IBER） 108
8.2.6 I/O 中断极性事件寄存器 (IO_IEVR) 109
8.2.7 I/O 中断使能寄存器（IO_IER） 110
8.2.8 I/O 中断状态和清零寄存器（IO_ISCR） 111
8.2.9 电源控制器中断使能寄存器（PWRC_IER） 112
8.2.10 电源控制器中断状态和清零寄存器（PWRC_ISCR） 113
8.2.11 I/O 模拟开关控制寄存器（GPIO_SWA_CTRL） 114
8.2.12 MR_BLE RX 或 TX 序列信息检测类型寄存器（BLERXTX_DTR） 115
8.2.13 MR_BLE RX 或 TX 序列信息检测类型寄存器（BLERXTX_IBER） 116
8.2.14 MR_BLE RX 或 TX 序列信息检测事件寄存器（BLERXTX_IEVR） 117
8.2.15 MR_BLE RX 或 TX 序列信息检测中断使能寄存器（BLERXTX_IER） 117
8.2.16 MR_BLE RX 或 TX 序列信息检测状态和清零寄存器（BLERXTX_ISCR） 118
8.3 系统控制器寄存器映射 119
9 嵌入式 Flash 存储器 121
9.1 Flash 主要特性 121
9.2 说明 121
9.3 Flash 控制器寄存器映射 122
9.4 Flash 控制器寄存器说明 123
9.4.1 指令寄存器（COMMAND） 123
9.4.2 配置寄存器（CONFIG） 124
9.4.3 中断状态寄存器（IRQSTAT） 125
9.4.4 中断屏蔽寄存器（IRQMASK） 126
9.4.5 原始状态寄存器（IRQRAW） 127
9.4.6 SIZE 寄存器 128
9.4.7 寄存器地址（ADDRESS） 129
9.4.8 线性反馈移位寄存器（LFSRVAL） 129
9.4.9 主 Flash 页面保护寄存器（PAGEPROTx） 130
9.4.10 数据寄存器（DATA0-DATA3） 131
9.5 编程模型 132
9.5.1 概述 132
9.5.2 读取功能示例 132

9.5.3 擦除功能示例 134
9.5.4 Write 功能示例 134
9.5.5 使能保护示例 135
9.5.6 OTP 功能示例 135
9.5.7 页面写保护示例 136
10 DMA 控制器（DMA） 137
10.1 DMA 简介 137
10.2 DMA 主要特性 137
10.3 DMA 功能说明 137
10.3.1 DMA 传输 137
10.3.2 仲裁器 138
10.3.3 DMA 通道 138
10.3.4 可编程数据宽度、数据对齐和字节序 139
10.3.5 错误管理 140
10.3.6 中断 140
10.3.7 DMA 请求映射 140
10.4 DMA 寄存器 141
10.4.1 DMA 中断状态寄存器（DMA_ISR） 141
10.4.2 DMA 中断标志清零寄存器（DMA_IFCR） 142
10.4.3 DMA 通道 x 配置寄存器（DMA_CCRx）（x = 1..8，其中的 x = 通道编号） 142
10.4.4 DMA 通道 x 数据数量寄存器（DMA_CNDTRx）（x = 1..8，其中的 x = 通道编号） 144
10.4.5 DMA 通道 x 外设地址寄存器（DMA_CPARx）（x = 1..8，其中的 x = 通道编号） .. 144
10.4.6 DMA 通道 x 存储器地址寄存器（DMA_CMARx）（x = 1..8，其中的 x = 通道编号） 145
10.5 DMA 寄存器映射 146
11 DMA 请求复用器（DMAMUX） 149
11.1 引言 149
11.2 DMAMUX 主要特性 149
11.3 DMAMUX 实现 149
11.3.1 DMAMUX 实例化 149
11.3.2 DMAMUX 映射 150
11.4 DMAMUX 功能说明 150

11.4.1 DMAMUX 框图 150
11.4.2 DMAMUX 通道 151
11.4.3 DMAMUX 请求线路复用器 151
11.5 DMAMUX 寄存器 151
11.5.1 DMAMUX 请求线路复用器通道 x 配置寄存器（DMAMUX_CxCR） 151
11.6 DMAMUX 寄存器映射 152
12 模数转换器（ADC） 154
12.1 特性 154
12.2 ADC 简介 154
12.2.1 可编程增益放大器（PGA） 155
12.2.2 温度传感器子系统 157
12.2.3 电池传感器 157
12.2.4 ADC 输入模式转换 157
12.2.5 校准点 158
12.2.6 稳定状态输入阻抗 158
12.2.7 输入信号采样瞬态响应 158
12.2.8 抽取滤波器（DF） 159
12.2.9 下采样器（DS） 162
12.3 中断 162
12.4 DMA 接口 162
12.5 ADC 模式 163
12.5.1 模拟音频模式 163
12.5.2 ADC 模式 165
12.5.3 数字音频模式 166
12.5.4 完全模式 167
12.5.5 偶发模式 167
12.5.6 并发功能 168
12.6 ADC 寄存器说明 169
12.6.1 版本寄存器（VERSION_ID） 169
12.6.2 ADC 配置寄存器（CONF） 169
12.6.3 ADC 控制寄存器（CTRL） 171
12.6.4 ADC 偶发模式控制寄存器（OCM_CTRL） 171

12.6.5 ADC PGA 配置寄存器（PGA_CONF） 173
12.6.6 ADC 输入电压开关选择寄存器（SWITCH） 174
12.6.7 抽取滤波器配置寄存器（DF_CONF） 176
12.6.8 下采样器配置寄存器（DS_CONF） 178
12.6.9 ADC 序列编程 1 寄存器（SEQ_1） 179
12.6.10 ADC 序列编程 2 寄存器（SEQ_2） 180
12.6.11 ADC 增益和偏移校正 1 寄存器（COMP_1） 181
12.6.12 ADC 增益和偏移校正 2 寄存器（COMP_2） 181
12.6.13 ADC 增益和偏移校正 3 寄存器（COMP_3） 182
12.6.14 ADC 增益和偏移校正 4 寄存器（COMP_4） 182
12.6.15 ADC 增益和偏移选择寄存器（COMP_SEL） 182
12.6.16 ADC 看门狗阈值寄存器（WD_TH） 185
12.6.17 ADC 看门狗配置寄存器（WD_CONF） 185
12.6.18 下采样器数据输出寄存器（DS_DATAOUT） 186
12.6.19 抽取滤波器数据输出寄存器（DF_DATAOUT） 186
12.6.20 ADC 中断状态寄存器（IRQ_STATUS） 186
12.6.21 ADC 中断使能寄存器（IRQ_ENABLE） 188
12.6.22 ADC 定时器配置寄存器（TIMER_CONF） 189
12.7 ADC 寄存器映射 190
13 随机数发生器（RNG） 194
13.1 特性 194
13.2 RNG 寄存器映射 194
13.3 RNG 寄存器说明 195
14 公钥加速器（PKA） 197
14.1 特性 197
14.2 PKA 寄存器映射 197
14.3 PKA 寄存器说明 198
14.4 编程模型 199
14.4.1 基本操作序列 199
14.4.2 PKA_RAM 中的数据位置 200
15 循环冗余校验计算单元（CRC） 201

15.1 引言 201
15.2 CRC 主要特性 201
15.3 CRC 功能说明 201
15.3.1 CRC 框图 201
15.3.2 CRC 操作 201
15.4 CRC 寄存器 203
15.4.1 数据寄存器（CRC_DR） 203
15.4.2 独立数据寄存器（CRC_IDR） 203
15.4.3 控制寄存器（CRC_CR） 203
15.4.4 CRC 初始值（CRC_INIT） 205
15.4.5 CRC 多项式（CRC_POL） 205
15.5 CRC 寄存器映射 205
16 高级控制定时器（TIM1） 206
16.1 TIM1 introduction 206
16.2 TIM1 主要特性 206
16.3 TIM1 功能说明 208
16.3.1 时基单元 208
16.3.2 计数器模式 209
16.3.3 重复计数器 217
16.3.4 外部触发输入 218
16.3.5 时钟选择 219
16.3.6 捕获/比较通道 221
16.3.7 输入捕获模式 223
16.3.8 PWM 输入模式 224
16.3.9 强制输出模式 225
16.3.10 输出比较模式 225
16.3.11 PWM 模式 226
16.3.12 不对称 PWM 模式 228
16.3.13 组合 PWM 模式 229
16.3.14 组合三相 PWM 模式 230
16.3.15 互补输出和死区时间插入 231
16.3.16 使用断路功能 232

16.3.17 发生外部事件时清除 OCxREF 信号 235
16.3.18 单脉冲模式 237
16.3.19 可再触发单脉冲模式（OPM） 239
16.3.20 编码器接口模式 239
16.3.21 UIF 位重映射 241
16.3.22 定时器输入异或功能 241
16.4 TIM1 寄存器 243
16.4.1 TIM1 控制寄存器 1（TIMx_CR1） 243
16.4.2 TIM1 控制寄存器 2（TIMx_CR2） 244
16.4.3 TIM1 从模式控制寄存器（TIMx_SMCR） 245
16.4.4 TIM1 中断使能寄存器（TIMx_DIER） 247
16.4.5 TIM1 状态寄存器（TIMx_SR） 248
16.4.6 TIM1 事件生成寄存器（TIMx_EGR） 249
16.4.7 TIM1 捕获/比较模式寄存器 1（TIMx_CCMR1） 250
16.4.8 TIM1 捕获/比较模式寄存器 2（TIMx_CCMR2） 254
16.4.9 TIM1 捕获/比较使能寄存器（TIMx_CCER） 255
16.4.10 TIM1 计数器（TIMx_CNT） 259
16.4.11 TIM1 预分频器（TIMx_PSC） 259
16.4.12 TIM1 自动重载寄存器（TIMx_ARR） 259
16.4.13 TIM1 重复计数器寄存器（TIMx_RCR） 260
16.4.14 TIM1 捕获/比较寄存器 1（TIMx_CCR1） 260
16.4.15 TIM1 捕获/比较寄存器 2（TIMx_CCR2） 260
16.4.16 TIM1 捕获/比较寄存器 3（TIMx_CCR3） 261
16.4.17 TIM1 捕获/比较寄存器 4（TIMx_CCR4） 261
16.4.18 TIM1 断路和死区时间寄存器（TIMx_BDTR） 262
16.4.19 TIM1 捕获/比较模式寄存器 3（TIMx_CCMR3） 265
16.4.20 TIM1 捕获/比较寄存器 5（TIMx_CCR5） 266
16.4.21 TIM1 捕获/比较寄存器 6（TIMx_CCR6） 266
16.4.22 TIM1 复用功能选项寄存器 1（TIMx_AF1） 267
16.4.23 TIM1 复用功能选项寄存器 2（TIMx_AF2） 268
16.5 TIM1 寄存器映射 270
17 实时时钟（RTC） 274

17.1 引言 274
17.2 RTC 主要特性 274
17.3 RTC 功能说明 275
17.3.1 RTC 框图 275
17.3.2 时钟和预分频器 275
17.3.3 实时时钟和日历 276
17.3.4 可编程闹铃 276
17.3.5 周期性自动唤醒 276
17.3.6 RTC 初始化和配置 277
17.3.7 读取日历 278
17.3.8 复位 RTC 278
17.3.9 RTC 同步 279
17.3.10 RTC 精密数字校准 279
17.3.11 校准时钟输出 280
17.3.12 闹钟输出 281
17.4 RTC 低功耗模式 281
17.5 RTC 中断 281
17.6 RTC 寄存器 282
17.6.1 RTC 时间寄存器（RTC_TR） 282
17.6.2 RTC 日期寄存器（RTC_DR） 282
17.6.3 RTC 控制寄存器（RTC_CR） 283
17.6.4 RTC 初始化和状态寄存器（RTC_ISR） 285
17.6.5 RTC 预分频器寄存器（RTC_PRER） 287
17.6.6 RTC 唤醒定时器寄存器（RTC_WUTR） 287
17.6.7 RTC 闹钟 A 寄存器（RTC_ALRMAR） 288
17.6.8 RTC 写保护寄存器（RTC_WPR） 289
17.6.9 RTC 亚秒寄存器（RTC_SSR） 289
17.6.10 RTC 平移控制寄存器（RTC_SHIFTR） 289
17.6.11 RTC 校准寄存器（RTC_CALR） 290
17.6.12 RTC 闹钟 A 亚秒寄存器（RTC_ALRMASSR） 291
17.6.13 RTC 备份寄存器（RTC_BKPxR） 292
17.7 RTC 寄存器映射 293

18 独立看门狗（IWDG） 297

18.1 引言 297

18.2 IWDG 主要特性 297

18.3 IWDG 功能说明 297

18.3.1 窗口选项 297

18.3.2 寄存器访问保护 298

18.3.3 调试模式 298

18.4 IWDG 寄存器 299

18.4.1 密钥寄存器（IWDG_KR） 299

18.4.2 预分频器寄存器（IWDG_PR） 299

18.4.3 重载寄存器（IWDG_RLR） 300

18.4.4 状态寄存器（IWDG_SR） 300

18.4.5 窗口寄存器（IWDG_WINR） 301

18.5 IWDG 寄存器映射 301

19 内部集成电路（I²C）接口 303

19.1 引言 303

19.2 I²C 主要特性 303

19.3 I²C 实现 304

19.4 I²C 功能说明 304

19.4.1 I²C 框图 304

19.4.2 I²C 时钟要求 305

19.4.3 模式选择 305

19.4.4 I²C 初始化 306

19.4.5 软件复位 309

19.4.6 数据传输 309

19.4.7 I²C 从模式 311

19.4.8 I²C 主模式 317

19.4.9 I²C_TIMINGR 寄存器配置示例 327

19.4.10 SMBus 特性 328

19.4.11 SMBus 初始化 330

19.4.12 SMBus:I2C_TIMEOUTR 寄存器配置示例 331

19.4.13 SMBus 从模式 332
19.4.14 错误条件 338
19.4.15 DMA 请求 339
19.5 I²C 中断 340
19.6 I²C 寄存器 341
19.6.1 控制寄存器 1（I2C_CR1） 341
19.6.2 控制寄存器 2（I2C_CR2） 343
19.6.3 设备自身地址 1 寄存器（I2C_OAR1） 346
19.6.4 设备自身地址 2 寄存器（I2C_OAR2） 346
19.6.5 时序寄存器（I2C_TIMINGR） 347
19.6.6 超时寄存器（I2C_TIMEOUTR） 348
19.6.7 中断和状态寄存器（I2C_ISR） 349
19.6.8 中断清零寄存器（I2C_ICR） 351
19.6.9 PEC 寄存器（I2C_PECR） 352
19.6.10 接收数据寄存器（I2C_RXDR） 352
19.6.11 发送数据寄存器（I2C_TXDR） 353
19.7 I²C 寄存器映射 354
20 通用同步/异步收发器（USART） 356
20.1 USART 简介 356
20.2 USART 主要特性 356
20.3 USART 扩展特性 357
20.4 USART 实现 357
20.5 USART 功能说明 358
20.5.1 USART 字符说明 359
20.5.2 FIFO 和阈值 360
20.5.3 发射器 361
20.5.4 接收器 363
20.5.5 波特率生成 368
20.5.6 USART 接收器对时钟偏差的容差 369
20.5.7 自动波特率检测 370
20.5.8 多处理器通信 371
20.5.9 Modbus 通信 372

20.5.10 奇偶校验控制 372
20.5.11 LIN（局域互连网络）模式 373
20.5.12 USART 同步模式 375
20.5.13 单线半双工通信 378
20.5.14 接收器超时 379
20.5.15 智能卡模式 379
20.5.16 IrDA SIR ENDEC 模块 382
20.5.17 使用 DMA 进行连续通信 384
20.5.18 RS232 硬件流控制和 RS485 驱动器使能 386
20.6 USART 中断 388
20.7 USART 寄存器 389
20.7.1 控制寄存器 1（USARTx_CR1） 389
20.7.2 控制寄存器 2（USARTx_CR2） 392
20.7.3 控制寄存器 3（USARTx_CR3） 395
20.7.4 波特率寄存器（USARTx_BRR） 398
20.7.5 保护时间和预分频器寄存器（USARTx_GTPR） 399
20.7.6 接收器超时寄存器（USARTx_RTOR） 400
20.7.7 请求寄存器（USARTx_RQR） 401
20.7.8 中断和状态寄存器（USARTx _ISR） 401
20.7.9 中断标志清零寄存器（USART_ICR） 406
20.7.10 接收数据寄存器（USART_RDR） 407
20.7.11 发送数据寄存器（USART_TDR） 408
20.7.12 预分频器寄存器（USARTx_PRESC） 408
20.8 USART 寄存器映射 409
21 通用异步收发器（LPUART） 412
21.1 LPUART 简介 412
21.2 LPUART 主要特性 412
21.3 LPUART 功能说明 413
21.3.1 LPUART 字符说明 414
21.3.2 FIFO 和阈值 415
21.3.3 发射器 416
21.3.4 接收器 418

21.3.5 波特率生成 421
21.3.6 多处理器通信 421
21.3.7 奇偶校验控制 423
21.3.8 单线半双工通信 423
21.3.9 使用 DMA 进行连续通信 424
21.3.10 RS232 硬件流控制和 RS485 驱动器使能 426
21.4 LPUART 中断 428
21.5 LPUART 寄存器 428
21.5.1 控制寄存器 1（LPUART_CR1） 428
21.5.2 控制寄存器 2（LPUART_CR2） 431
21.5.3 控制寄存器 3（LPUART_CR3） 432
21.5.4 波特率寄存器（LPUART_BRR） 434
21.5.5 请求寄存器（LPUART_RQR） 435
21.5.6 中断和状态寄存器（LPUART_ISR） 436
21.5.7 中断标志清零寄存器（LPUART_ICR） 439
21.5.8 接收数据寄存器（LPUART_RDR） 440
21.5.9 发送数据寄存器（LPUART_TDR） 440
21.5.10 预分频器寄存器（LPUART_PRESC） 441
21.6 LPUART 寄存器映射 441
22 串行外设接口/集成电路内置音频总线（SPI/I2S） 444
22.1 引言 444
22.2 SPI 主要特性 444
22.3 I2S 主要特性 444
22.4 SPI/I2S 实现 445
22.5 SPI 功能说明 445
22.5.1 概述 445
22.5.2 一个主设备和一个从设备之间的通信 446
22.5.3 标准多从设备通信 448
22.5.4 从设备选择（NSS）引脚管理 449
22.5.5 通信格式 450
22.5.6 SPI 配置 452
22.5.7 使能 SPI 的步骤 453

22.5.8 数据发送和接收过程 453
22.5.9 SPI 状态标志 460
22.5.10 SPI 错误标志 461
22.5.11 NSS 脉冲模式 462
22.5.12 TI 模式 462
22.5.13 CRC 计算 463
22.6 SPI 中断 464
22.7 I2S 功能说明 464
22.7.1 I2S 概述 464
22.7.2 支持的音频协议 466
22.7.3 时钟发生器 472
22.7.4 I2S 主模式 474
22.7.5 I2S 从模式 476
22.7.6 I2S 错误标志 477
22.7.7 DMA 特性 477
22.8 I2S 中断 477
22.9 SPI 和 I^2S 寄存器 478
22.9.1 SPIx 控制寄存器 1（SPIx_CR1） 478
22.9.2 SPIx 控制寄存器 2（SPIx_CR2） 480
22.9.3 SPI 状态寄存器（SPIx_SR） 482
22.9.4 SPI 数据寄存器（SPIx_DR） 483
22.9.5 SPI CRC 多项式寄存器（SPIx_CRCPR） 484
22.9.6 SPI 接收 CRC 寄存器（SPIx_RXCRCR） 484
22.9.7 SPI 发送 CRC 寄存器（SPIx_TXCRCR） 484
22.9.8 SPIx_I2S 配置寄存器（SPIx_I2SCFGR） 485
22.9.9 SPIx_I2S 预分频器寄存器（SPIx_I2SPR） 486
22.10 SPI/I2S 寄存器映射 487
23 调试支持（DBG） 489
23.1 SWD 调试特性 489
24 射频子系统（MR_BLE） 490
版本历史 492

表格索引

表 1. 寄存器相关缩写词列表 2
表 2. 缩略语 3
表 3. BlueNRG-LP 存储器映射和外设寄存器边界地址 9
表 4. 地址重映射取决于 REMAP 位 10
表 5. 中断向量 11
表 6. GPIO 复用选项 AF0 - AF2 16
表 7. GPIO 复用选项 AF3 - AF4 18
表 8. I/O 模拟特性映射 20
表 9. SMPS BOM 信息 29
表 10. PWRC 寄存器映射 53
表 11. 标志与 CPU 重启原因 55
表 12. 唤醒原因标志 55
表 13. CPU 与 MR_BLE 时钟依赖关系 61
表 14. RCC 寄存器映射和复位值 82
表 15. 端口位配置 90
表 16. GPIO 寄存器映射和复位值 102
表 17. SYSCFG 寄存器映射和复位值 119
表 18. Flash 存储器节区地址 121
表 19. Flash APB 寄存器 122
表 20. 客户可用的指令列表 123
表 21. Flash 大小信息 128
表 22. 系统存储器保护 135
表 23. 可编程数据宽度和字节序行为（当 PINC=MINC=1 且 NDT=4 时） 139
表 24. DMA 中断请求 140
表 25. DMA 寄存器映射和复位值 146
表 26. DMAMUX 实例化 149
表 27. DMAMUX 映射 150
表 28. DMAMUX 寄存器映射和复位值 152
表 29. PGA 参数 156
表 30. 校准点 158
表 31. 模拟模式下 ADC 输入频率为 1 MHz 时的输出数据速率 160
表 32. 有数字麦克风输入的 CIC 滤波器输出频率 161
表 33. 不同 pdm_rate 的 CIC/完全链的最小抽取因子 161
表 34. ADC 中断请求 162
表 35. ADC 模式总结 163
表 36. ADC 寄存器映射和复位值 190
表 37. RNG 寄存器列表 194
表 38. RNG_CR 寄存器说明 195
表 39. RNG_SR 寄存器说明 195
表 40. RNG_VAL 寄存器说明 196
表 41. PKA 寄存器列表 197
表 42. PKA_CSR 寄存器说明 198
表 43. PKA_ISR 寄存器说明 198
表 44. PKA_ISR 寄存器说明 199
表 45. KA_IEN 寄存器说明 199
表 46. ECC 标量乘法数据位置 200
表 47. CRC 寄存器映射和复位值 205
表 48. 不同 BRK/BK2 输入的定时器输出行为 235
表 49. 不同编码器信号的计数方向 240
表 50. 具有断路功能的互补通道 OCx 和 OCxN 的输出控制位 258
表 51. TIM1 寄存器映射和复位值 270
表 52. RTC 寄存器映射和复位值 293
表 53. IWDG 寄存器映射 302
表 54. BlueNRG-LP I2C 实现 304
表 55. I^2C-SMBUS 规范数据建立时间和保持时间 308
表 56. I2C 可配置表 311

表 57. I²C-SMBUS 规范时钟时序 319
表 58. fI2CCLK = 16 MHz 时的时序设置示例 327
表 59. SMBus 超时规范 329
表 60. 具有 PEC 配置的 SMBUS 331
表 61. TIMEOUTA 设置的示例（$t_{TIMEOUT}$ 最大值 = 25 ms） 331
表 62. TIMEOUTB 设置的示例 332
表 63. TIMEOUTA 设置的示例（t_{IDLE} 最大值 = 50 µs） 332
表 64. I²C 中断请求 340
表 65. I²C 寄存器映射 354
表 66. USART/LPUART 特性 357
表 67. 通过采样数据进行噪声检测 367
表 68. BRR [3:0] = 0000 时的 USART 接收器容差（高密度器件） 369
表 69. BRR [3:0] ≠ 0000 时的 USART 接收器容差（高密度器件） 369
表 70. 帧格式 373
表 71. USART 中断请求 388
表 72. USART 寄存器映射 409
表 73. 帧格式 423
表 74. LPUART 中断请求 428
表 75. LPUART 寄存器映射和复位值 441
表 76. BlueNRG-LP SPI 实现 445
表 77. SPI 中断请求 464
表 78. I2SCLK = 32 MHz 时的音频频率精度 473
表 79. I2SCLK = 16 MHz 时的音频频率精度 474
表 80. I²S（中断请求） 478
表 81. MR_BLE/BlueNRG-LP 的依赖关系 490
表 82. 文档版本历史 492

图片目录

图 1. BlueNRG-LP 系统架构 6
图 2. 存储器映射 8
图 3. AHB 上/下变频器 13
图 4. 电源域总览 22
图 5. 上电复位/下电复位波形 22
图 6. RUN 模式下的功率调节器和 SMPS 配置 24
图 7. DEEPSTOP 模式下的功率调节器和 SMPS 配置 26
图 8. SHUTDOWN 模式下的功率调节器和 SMPS 配置 27
图 9. 用于工作模式转换的 PWRC 状态机 28
图 10. 电源配置 29
图 11. PWRC SMPS 状态机总览 29
图 12. 复位生成 57
图 13. 外设时钟树总览 60
图 14. LCO/MCO 输出时钟 63
图 15. 混合模拟/数字 5 V 容限 I/O 端口位的基本结构 89
图 16. 仅数字 5 V 容限 I/O 端口位的基本结构 89
图 17. 输入浮空/上拉/下拉配置 93
图 18. 输出配置 93
图 19. 复用功能配置 94
图 20. 高阻态模拟配置 95
图 21. DMAMUX 框图 150
图 22. ADC 顶层图 155
图 23. 麦克风设置 156
图 24. ADC 采样时间 Tsw 和采样周期 Ts 159
图 25. 模拟源电阻的作用 159
图 26. 简化的抽取滤波器框图 160
图 27. CRC 计算单元框图 201
图 28. 高级控制定时器框图 207
图 29. 预分频器分频由 1 变为 2 时的计数器时序图 208
图 30. 预分频器分频由 1 变为 4 时的计数器时序图 209
图 31. 计数器时序图，1 分频内部时钟 210
图 32. 计数器时序图，2 分频内部时钟 210
图 33. 计数器时序图，4 分频内部时钟 210
图 34. 计数器时序图，N 分频内部时钟 211
图 35. 计数器时序图，ARPE=0 时更新事件（TIMx_ARR 未预装载） 211
图 36. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 已预装载） 212
图 37. 计数器时序图，1 分频内部时钟 213
图 38. 计数器时序图，2 分频内部时钟 213
图 39. 计数器时序图，4 分频内部时钟 213
图 40. 计数器时序图，N 分频内部时钟 214
图 41. 计数器时序图，未使用重复计数器时更新事件 214
图 42. 计数器时序图，1 分频内部时钟，TIMx_ARR = 0x6 215
图 43. 计数器时序图，2 分频内部时钟 215
图 44. 计数器时序图，4 分频内部时钟，TIMx_ARR=0x36 216
图 45. 计数器时序图，N 分频内部时钟 216
图 46. 计数器时序图，ARPE=1 时的更新事件（计数器下溢） 216
图 47. 计数器时序图，ARPE=1 时的更新事件（计数器上溢） 217
图 48. 不同模式和 TIMx_RCR 寄存器设置下的更新频率示例 218
图 49. 外部触发输入模块 218
图 50. 正常模式下的控制电路，1 分频内部时钟 219
图 51. TI2 外部时钟连接示例 219
图 52. 外部时钟模式 1 下的控制电路 220
图 53. 外部触发输入模块 220
图 54. 外部时钟模式 2 下的控制电路 221
图 55. 捕获/比较通道（例如：通道 1 输入阶段） 221
图 56. 捕获/比较通道 1 主电路 222

图 57. 捕获/比较通道的输出阶段（通道 1，通道 2 和 3 同前） 222
图 58. 捕获/比较通道的输出阶段（通道 4） 223
图 59. 捕获/比较通道的输出阶段（通道 5，通道 6 同前） 223
图 60. PWM 输入模式时序 225
图 61. 输出比较模式，翻转 OC1 226
图 62. 边沿对齐模式的 PWM 波形（ARR=8） 227
图 63. 中心对齐模式 PWM 波形（ARR=8） 228
图 64. 50%占空比时产生的 2 个相移 PWM 信号 229
图 65. 通道 1 和通道 3 上的组合 PWM 模式 230
图 66. 三相组合 PWM 信号（每个周期多个触发脉冲） 231
图 67. 插入了死区时间的互补输出 232
图 68. 延时大于负脉冲的死区时间波形 232
图 69. 延时大于正脉冲的死区时间波形 232
图 70. Break 和 break2 电路系统总览 233
图 71. 响应 BRK 上的断路事件的不同输出行为（OSSI = 1） 234
图 72. 跟随 BRK 和 BRK2 引脚断言的 PWM 输出状态（OSSI=1） 235
图 73. 跟随 BRK 断言的 PWM 输出状态（OSSI=0） 235
图 74. 清除 TIMx 的 OCxREF 236
图 75. COM 事件生成 6 步 PWM 的示例（OSSR=1） 237
图 76. 单脉冲模式示例 238
图 77. 可再触发单脉冲模式 239
图 78. 编码器接口模式下的计数器工作示例 240
图 79. TI1FP1 极性反相时的编码器接口模式示例 241
图 80. 测量 3 个信号上边沿之间的时间间隔 242
图 81. RTC 框图 275
图 82. 独立看门狗框图 298
图 83. I^2C 框图 305
图 84. I^2C 总线协议 306
图 85. 建立和保持时序 307
图 86. I^2C 初始化流程图 309
图 87. 数据接收 310
图 88. 数据发送 310
图 89. 从设备初始化流程图 313
图 90. I2C 从发射器的传输序列流程图（NOSTRETCH=0） 314
图 91. I2C 从发射器的传输序列流程图（NOSTRETCH=1） 315
图 92. I2C 从发射器的传输总线图 315
图 93. 从接收器的传输序列流程图（NOSTRETCH=0） 316
图 94. 从接收器的传输序列流程图（NOSTRETCH=1） 317
图 95. I2C 从接收器的传输总线图 317
图 96. 主时钟生成 319
图 97. 主模式初始化流程图 321
图 98. 10 位地址读访问（HEAD10R=1） 321
图 99. I2C 主发射器的传输序列流程图（N 255 字节） 322
图 100. I2C 主发射器的传输序列流程图（N>255 字节） 323
图 101. I2C 主发射器的传输总线图 324
图 102. I2C 主接收器的传输序列流程图（N>255 字节） 325
图 103. I2C 主接收器的传输序列流程图（N>255 字节） 326
图 104. I^2C 主接收器的传输总线图 327
图 105. $t_{LOW:SEXT}$ 和 $t_{LOW:MEXT}$ 的超时间隔 330
图 106. SMBus 从发射器的传输序列流程图（N 字节 + PEC） 333
图 107. SMBus 从发射器的传输总线图（SBC=1） 333
图 108. SMBus 从接收器的传输序列流程图（N 字节 + PEC） 335
图 109. SMBus 从接收器的总线传输图（SBC=1） 336
图 110. SMBus 主发射器的总线传输图 337
图 111. SMBus 主接收器的总线传输图 338
图 112. I^2C 中断映射图 341
图 113. USART 框图 359
图 114. 字长编程 360
图 115. 可配置的停止位 362

图 116. 发送时的 TC/TXE 行为 363
图 117. 16 倍或 8 倍过采样时的起始位检测 364
图 118. usart_ker_ck 时钟分频器框图 366
图 119. 16 倍过采样时的数据采样 367
图 120. 8 倍过采样时的数据采样 367
图 121. 使用空闲线路检测的静默模式 371
图 122. 使用地址标记检测的静默模式 372
图 123. LIN 模式下的断路检测（11 位断路长度 - LBDL 位置 1） 374
图 124. LIN 模式下的断路检测与数据拆包错误检测 375
图 125. 同步主设备发送的 USART 示例 376
图 126. USART 数据时钟时序图 M=0 376
图 127. USART 数据时钟时序图（M 位 = 01） 377
图 128. RX 数据建立/保持时间 377
图 129. ISO 7816-3 异步协议 379
图 130. 使用 1.5 个停止位的奇偶校验错误检测 381
图 131. IrDA SIR ENDEC - 框图 384
图 132. IrDA 数据调制（3/16） - 正常模式 384
图 133. 使用 DMA 发送 385
图 134. 使用 DMA 接收 386
图 135. 2 个 USART 间的硬件流控制 386
图 136. RS232 RTS 流控制 387
图 137. RS232 CTS 流控制 387
图 138. LPUART 框图 414
图 139. LPUART 字长编程 415
图 140. 可配置的停止位 417
图 141. 发送时的 TC/TXE 行为） 418
图 142. lpuart_ker_ck 时钟分频器框图 420
图 143. 使用空闲线路检测的静默模式 422
图 144. 使用地址标记检测的静默模式 422
图 145. 使用 DMA 发送 425
图 146. 使用 DMA 接收 426
图 147. 2 个 LPUART 间的硬件流控制 426
图 148. RS232 RTS 流控制 427
图 149. RS232 RTS 流控制 427
图 150. SPI 框图 446
图 151. 全双工单个主设备/单个从设备应用 447
图 152. 半双工单个主设备/单个从设备应用 447
图 153. 单工单一主设备/单一从设备应用（在只发送模式下为主设备/在只接收模式下为从设备） 448
图 154. 主设备和三个独立从设备 449
图 155. 硬件/软件从设备选择管理 450
图 156. 数据时钟时序图 451
图 157. 数据长度不等于 8 位或 16 位时的数据对齐 452
图 158. 将数据封装到 FIFO 中以进行发送和接收 455
图 159. 主设备全双工通信 457
图 160. 从设备全双工通信 458
图 161. 采用 CRC 的主设备全双工通信 459
图 162. 封装模式下的主设备全双工通信 460
图 163. Motorola SPI 主模式下的 NSSP 脉冲生成 462
图 164. TI 模式传输 463
图 165. I^2S 框图 465
图 166. I^2S Philips 协议波形（16/32 位全精度，CPOL = 0） 466
图 167. I^2S Philips 标准波形（24 位帧，CPOL = 0） 466
图 168. 发送 0x8EAA33 467
图 169. 接收 0x8EAA33 467
图 170. I^2S Philips 标准（16 位扩展到 32 位数据包帧，CPOL = 0） 467
图 171. 16 位数据帧扩展到 32 位通道帧的示例 468
图 172. MSB 对齐的 16 位或 32 位全精度长度，CPOL = 0 468
图 173. MSB 对齐的 24 位帧长度，CPOL = 0 468
图 174. 扩展为 32 位数据包帧的 MSB 对齐的 16 位，CPOL = 0 469

图 175. LSB 对齐的 16 位或 32 位全精度长度，CPOL = 0 469
图 176. LSB 对齐的 24 位帧长度，CPOL = 0 469
图 177. 发送 0x3478AE 所需的操作 470
图 178. 接收 0x3478AE 时所需的操作 470
图 179. 扩展为 32 位数据包帧的 LSB 对齐的 16 位，CPOL = 0 470
图 180. 16 位数据帧扩展到 32 位通道帧的示例（2） 471
图 181. PCM 标准波形（16 位） 471
图 182. PCM 标准波形（16 位扩展到 32 位数据包帧） 471
图 183. 音频采样频率定义 472
图 184. I^2S 时钟发生器架构 472

重要通知 - 仔细阅读

意法半导体公司及其子公司（“意法半导体”）保留随时对 ST 产品和/或本文档进行变更、更正、增强、修改和改进的权利，恕不另行通知。买方在订货之前应获取关于意法半导体产品的最新信息。ST 产品的销售依照订单确认时的相关 ST 销售条款。

买方自行负责对意法半导体产品的选择和使用，意法半导体概不承担与应用协助或买方产品设计相关的任何责任。

意法半导体不对任何知识产权进行任何明示或默示的授权或许可。

转售的意法半导体产品如有不同于此处提供的信息的规定，将导致意法半导体针对该产品授予的任何保证失效。

ST 及 ST 标识是意法半导体公司的商标。若需意法半导体商标的更多信息，请参考 www.st.com/trademarks。其他所有产品或服务名称是其各自所有者的财产。

本文档中的信息取代本文档所有早期版本中提供的信息。

© 2023 STMicroelectronics - 保留所有权利