CIU32L051x8

Cortex-M0+ 32-bit MCUs

Reference manual

RM1006

北京中电华大电子设计有限责任公司
CEC Huada Electronic Design Co.,Ltd

Declaration

The right to change the manual without notice to the user is reserved by CEC Huada Electronic Design Co., Ltd. Although we have reviewed the content of this manual, it is inevitable that it may still contain errors. As a result, we will periodically review the content of this manual and make necessary modification in the next release of the document. We recommend that you obtain the latest version from CEC Huada Electronic before finalizing your design.

Contents

List of tables

List of figures

1 Documentation conventions

1.1 List of abbreviations for registers

read/write (rw)	Software can read and write to this bit.
read only (r)	Software can only read this bit.
write only (w)	Software can only write to this bit. Reading this bit returns the reset value.
read/clear write 1 (rc_w1)	Software can read as well as clear this bit by writing 1. Writing 0 has no effect on the bit value.
read/clear write 0 (rc_w0)	Software can read as well as clear this bit by writing 0. Writing 1 has no effect on the bit value.
read/clear by read (rc_r)	Software can read this bit. Reading this bit automatically clears it to 0. Writing this bit has no effect on the bit value.
read/set by read (rs_r)	Software can read this bit. Reading this bit automatically sets it to 1. Writing this bit has no effect on the bit value.
read/set (rs)	Software can read as well as set this bit. Writing 0 has no effect on the bit value.
read/write once (rwo)	Software can only write once to this bit and can also read it at any time. Only a reset can return the bit to its reset value.
read-only write trigger (rt_w1)	Software can read this bit. Writing 1 triggers an event but has no effect on the bit value.
Reserved (Res.)	Reserved bit, reading this bit returns the reset value. Writing this bit has no effect on the bit value.

1.2 Glossary

The brief definition of acronyms and abbreviations used in this document:

Sector	512 bytes Flash memory region
Word	Data of 32-bit length

Half-word	Data of 16-bit length
Byte	Data of 8-bit length
Option Bytes	Product configuration bits stored in the Flash memory
OBL	Option byte loader
AHB	Advanced high-performance bus
APB	Advanced peripheral bus
Run	Run mode
Sleep	Sleep mode
Stop	Stop mode

2 Introduction

The CIU32L051 ultra-low-power secure MCU is based on ARM Cortex-M0+ core. It is available in multiple packages, including LQFP64/48, QFN32 and SSOP24. The maximum frequency is up to 48 MHz. It supports independent backup power supply. Abundant peripherals are built in, including LCD, ADC, internal reference voltage source VREFBUF, ultra-low-power comparer, multiple LPUART/U(S)ART/I2C/SPI, RTC, multiple timers, and AES algorithm coprocessor.

Application scenarios of CIU32L051 ultra-low-power security MCUs:

- Smart fire protection
- Portable medical devices
- Smart household appliances
- Other low-power consumption scenarios powered by batteries

3 Pinouts, pin description and alternate functions

3.1 Pinouts

The devices house in LQFP64, LQFP48, QFN32, and SSOP24 packages.

Figure 3-1 CIU32L051R8T6-LQFP64 pinout

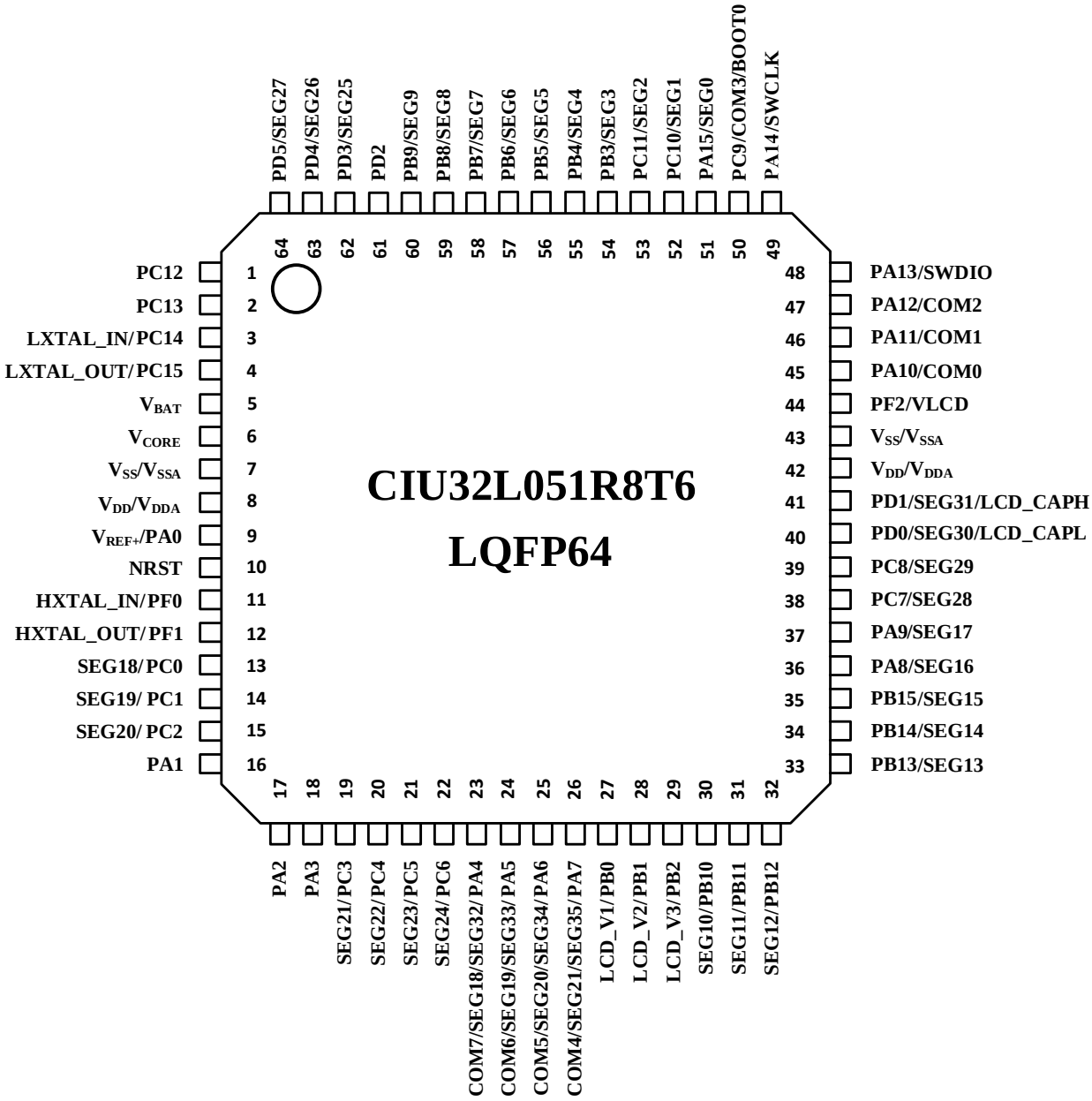


Figure 3-2 CIU32L051C8T6-LQFP48 pinout

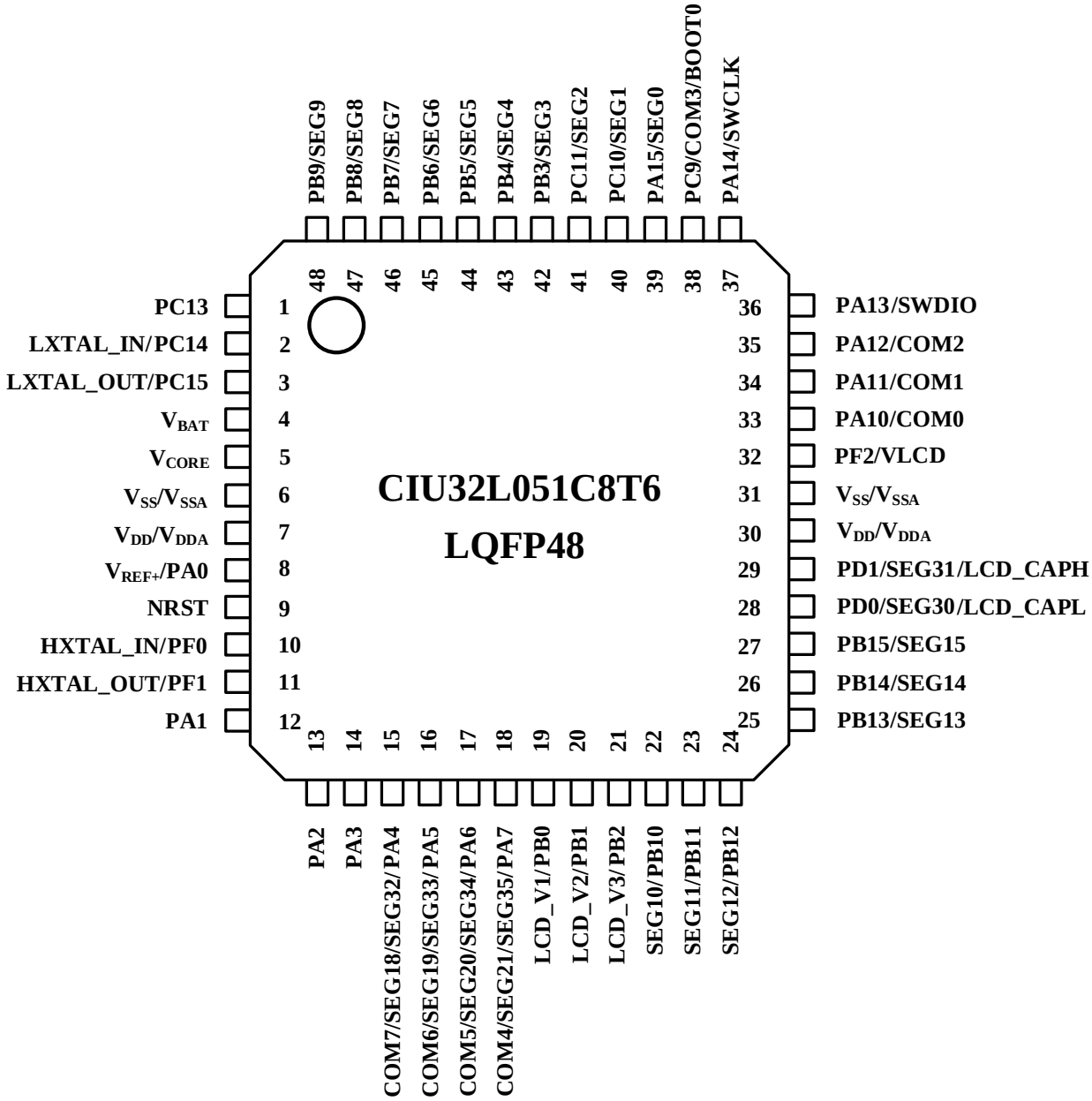
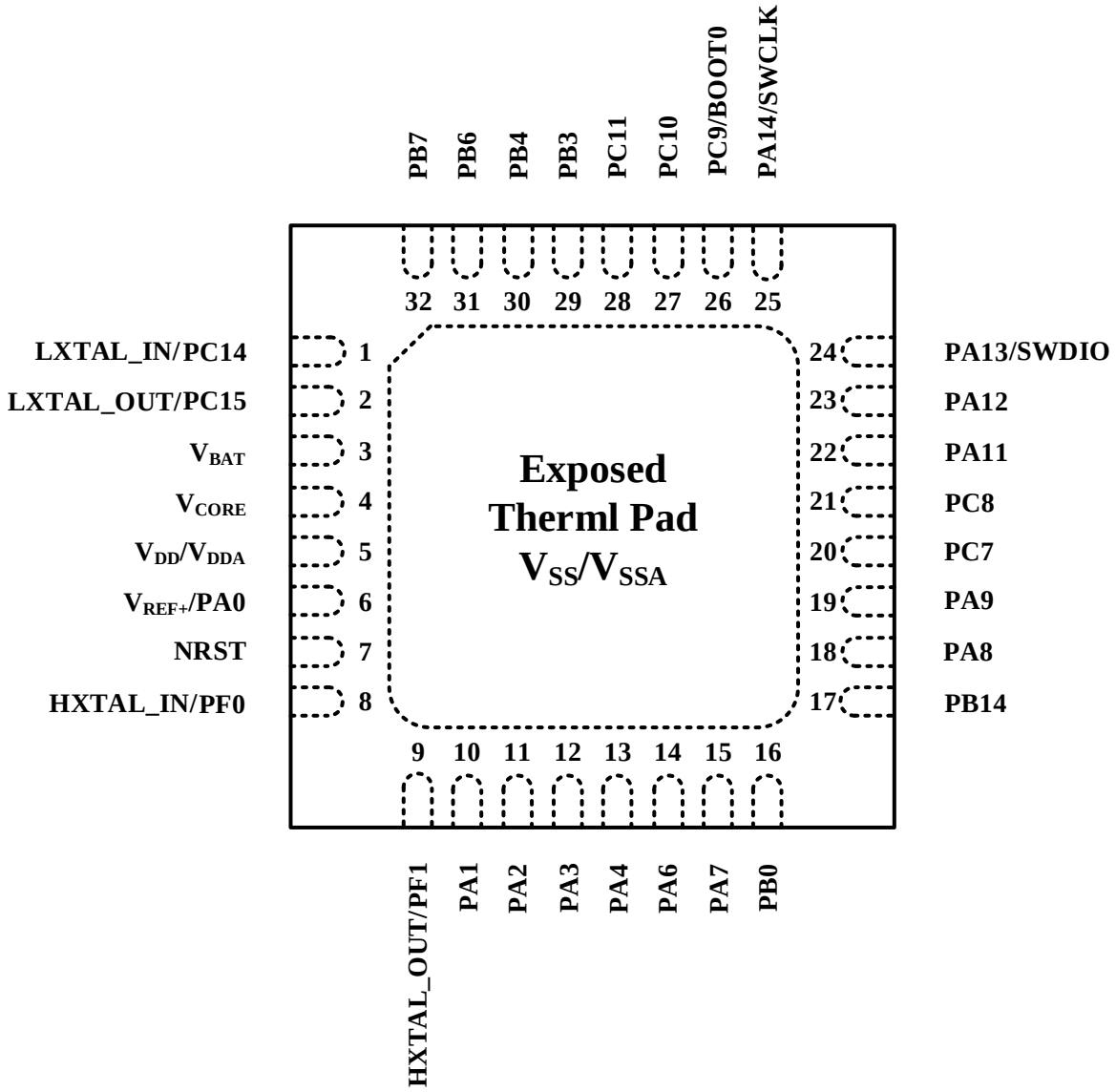
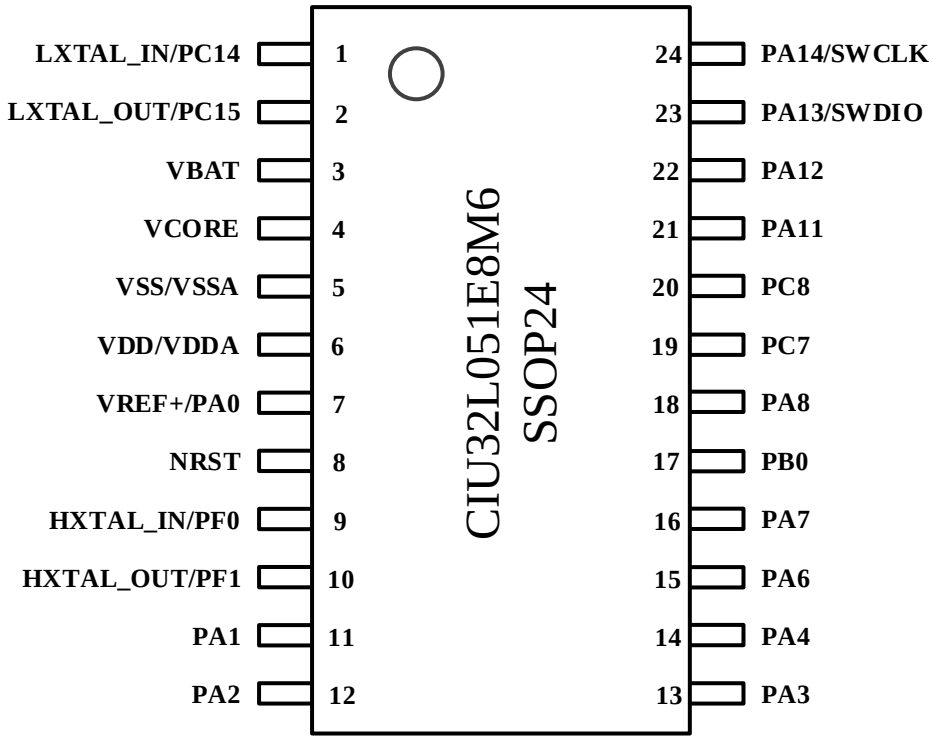


Figure 3-3 CIU32L051K8U6-QFN32 pinout



Note: Exposed Thermal Pad is V_{SS}/V_{SSA}, and must be connected to the Ground of PCB board for QFN32.

Figure 3-4 CIU32L051E8M6-SSOP24 pinout



3.2 Pin assignment and description

Table 3-1 Pin assignment and description

Pin number				Pin name	Pin type	Driving capability	Additional functions	Alternate functions
LQFP64	LQFP48	QFN32	SSOP24					
1	-	-	-	PC12	I/O	Medium	ADC_IN19	UART3_TX TIM5_CH3 TIM4_CH2 IR_OUT
2	1	-	-	PC13	I/O	Low	TAMP_IN RTC_OUT	-
3	2	1	1	PC14	I/O	Low	LXTAL_IN	-
4	3	2	2	PC15	I/O	Low	LXTAL_OUT	-
5	4	3	3	V_{BAT}	P	-	-	-
6	5	4	4	V_{CORE}	P	-	-	-
7	6	-	5	V_{SS}/V_{SSA}	G	-	-	-
8	7	5	6	V_{DD}/V_{DDA}	P	-	-	-
9	8	6	7	V_{REF+}/PA0	I/O	Medium	-	SPI2_SCK USART1_CTS UART4_TX LPTIM1_OUT COMP1_OUT
10	9	7	8	NRST	I	Medium	NRST	-
11	10	8	9	PF0	I/O	Medium	HXTAL_IN	TIM5_CH3 UART2_TX UART4_RTS LPUART2_CTS LPUART1_CTS I2C1_SCL
12	11	9	10	PF1	I/O	Medium	HXTAL_OUT	TIM5_CH4 UART2_RX UART4_CTS LPUART2_RTS LPUART1_RTS I2C1_SDA
13	-	-	-	PC0	I/O	Medium	-	SPI2_SCK TIM4_CH1

Pin number				Pin name	Pin type	Driving capability	Additional functions	Alternate functions
LQFP64	LQFP48	QFN32	SSOP24					
								LCD_SEG18 LPTIM1_OUT LPUART1_TX
14	-	-	-	PC1	I/O	Medium	-	SPI2_MISO TIM4_CH2 LCD_SEG19 LPTIM1_IN2 LPUART1_RX I2C1_SDA
15	-	-	-	PC2	I/O	Medium	-	SPI2_MOSI TIM4_CH3 LCD_SEG20 LPTIM1_IN1 I2C1_SCL
16	12	10	11	PA1	I/O	Medium	COMP2_INP ADC_IN0	SPI1_SCK USART1_RX TIM4_CH4 UART4_RX LPUART2_CTS
17	13	11	12	PA2	I/O	High/Medium configurable	COMP2_INM ADC_IN1	SPI1_MOSI USART1_TX TIM4_CH1 MCO LPUART1_TX COMP2_OUT
18	14	12	13	PA3	I/O	Medium	ADC_IN2	SPI2_MISO USART1_RTS_DE_CK TIM4_CH2 UART4_TX MCO LPUART1_RX
19	-	-	-	PC3	I/O	Medium	COMP1_INM	USART1_TX LCD_SEG21 UART3_CTS LPTIM1_ETR

Pin number				Pin name	Pin type	Driving capability	Additional functions	Alternate functions
LQFP64	LQFP48	QFN32	SSOP24					
20	-	-	-	PC4	I/O	Medium	COMP1_INP	USART1_RX LCD_SEG22 UART3_RTS UART3_RX
21	-	-	-	PC5	I/O	Medium	-	TIM3_CH1 USART1_RTS_DE_CK LCD_SEG23 UART3_TX LPUART1_TX
22	-	-	-	PC6	I/O	Medium	-	TIM3_CH2 USART1_CTS LCD_SEG24 UART3_RX LPUART1_RX
23	15	13	14	PA4	I/O	Medium	ADC_IN3 COMP1_INM	SPI1_NSS SPI2_MOSI TIM4_CH3 SEG32/SEG18/COM7 LPUART2_TX I2C1_SDA
24	16	-	-	PA5	I/O	Medium	ADC_IN4 COMP1_INP	SPI1_SCK IR_OUT TIM4_CH4 SEG33/SEG19/COM6 UART3_RTS LPUART2_RX I2C1_SCL COMP1_OUT
25	17	14	15	PA6	I/O	Medium	ADC_IN5	SPI1_MISO TIM3_CH1 SEG34/SEG20/COM5 UART3_CTS TIM5_CH1 LPUART1_CTS LPTIM1_IN1

Pin number				Pin name	Pin type	Driving capability	Additional functions	Alternate functions
LQFP64	LQFP48	QFN32	SSOP24					
26	18	15	16	PA7	I/O	Medium	ADC_IN6	SPI1_MOSI TIM3_CH2 SEG35/SEG21/COM4 TIM4_CH1 LPTIM1_IN2 COMP2_OUT
27	19	16	17	PB0	I/O	Medium	ADC_IN7 LCD_V1	SPI1_NSS TIM3_CH3 UART3_RX LPTIM1_OUT COMP1_OUT
28	20	-	-	PB1	I/O	Medium	COMP1_INM ADC_IN8 LCD_V2	TIM3_CH4 UART3_RTS LPUART1_RTS
29	21	-	-	PB2	I/O	Medium	COMP1_INP ADC_IN9 LCD_V3	SPI2_MISO UART3_TX LPTIM1_OUT
30	22	-	-	PB10	I/O	Medium	ADC_IN10	SPI2_MOSI LCD_SEG10 UART3_TX LPUART1_RX COMP1_OUT
31	23	-	-	PB11	I/O	Medium	ADC_IN11	SPI2_SCK LCD_SEG11 UART3_RX LPUART1_TX COMP2_OUT
32	24	-	-	PB12	I/O	Medium	ADC_IN15	SPI2_NSS LCD_SEG12 LPUART1_RTS
33	25	-	-	PB13	I/O	Medium	ADC_IN16	SPI2_SCK TIM4_ETR LCD_SEG13 UART3_CTS LPUART1_CTS

Pin number				Pin name	Pin type	Driving capability	Additional functions	Alternate functions
LQFP64	LQFP48	QFN32	SSOP24					
34	26	17	-	PB14	I/O	Medium	-	SPI2_MISO LCD_SEG14 UART3_RTS TIM4_CH1 LPUART2_CTS
35	27	-	-	PB15	I/O	Medium	-	SPI2_MOSI TIM4_CH2 LCD_SEG15 LPUART2_RTS COMP1_OUT
36	-	18	18	PA8	I/O	Medium	COMP1_INP	MCO SPI2_NSS UART3_TX LCD_SEG16 TIM4_CH3 LPTIM1_IN1 LPUART2_TX
37	-	19	-	PA9	I/O	Medium	COMP1_INM	MCO USART1_CTS UART3_RX LCD_SEG17 SPI2_MISO TIM4_CH4 LPUART2_RX LPTIM1_IN2
38	-	20	19	PC7	I/O	Medium	-	SPI1_NSS TIM3_CH3 TIM4_CH3 LCD_SEG28 LPUART2_RX UART3_RTS I2C1_SCL
39	-	21	20	PC8	I/O	Medium	-	SPI1_SCK TIM3_CH4 TIM4_CH4

Pin number				Pin name	Pin type	Driving capability	Additional functions	Alternate functions
LQFP64	LQFP48	QFN32	SSOP24					
								LCD_SEG29 LPUART2_TX LPTIM1_OUT UART3_CTS I2C1_SDA
40	28	-	-	PD0	I/O	Medium	LCD_CAPL	SPI1_MOSI SPI2_NSS TIM5_CH1 LCD_SEG30 LPTIM1_IN1 UART3_RX
41	29	-	-	PD1	I/O	Medium	LCD_CAPH	SPI1_MISO SPI2_SCK TIM5_ETR LCD_SEG31 LPTIM1_IN2 UART3_TX
42	30	-	-	V_{DD}/V_{DDA}	P	-	-	-
43	31	-	-	V_{SS}/V_{SSA}	G	-	-	-
44	32	-	-	PF2	I/O	Medium	VLCD	SPI2_MISO USART1_CTS TIM4_ETR TIM5_CH1 LPTIM1_ETR
45	33	-	-	PA10	I/O	Medium	-	SPI2_MOSI USART1_RTS_DE_CK TIM4_CH4 LCD_COM0 TIM5_CH2 TIM5_CH1 TIM5_CH3
46	34	22	21	PA11	I/O	Medium	-	SPI1_MISO USART1_TX LCD_COM1 TIM5_CH3

Pin number				Pin name	Pin type	Driving capability	Additional functions	Alternate functions
LQFP64	LQFP48	QFN32	SSOP24					
								TIM5_CH4 COMP1_OUT
47	35	23	22	PA12	I/O	Medium	-	SPI1_MOSI USART1_RX TIM4_ETR LCD_COM2 TIM5_CH4 COMP2_OUT
48	36	24	23	PA13	I/O	Medium	-	SWDIO USART1_TX IR_OUT LPUART2_TX
49	37	25	24	PA14	I/O	Medium	-	SWCLK USART1_RX LPUART2_RX
50	38	26	-	PC9	I/O	Medium	BOOT0	LCD_COM3
51	39	-	-	PA15	I/O	Medium	-	SPI1_NSS LCD_SEG0 UART4_RTS UART3_RTS
52	40	27	-	PC10	I/O	Medium	-	SPI2_MOSI UART3_RTS LCD_SEG1 TIM3_ETR
53	41	28	-	PC11	I/O	Medium	-	SPI2_MISO UART2_CTS LCD_SEG2 LPUART2_CTS
54	42	29	-	PB3	I/O	Medium	-	SPI1_SCK UART2_RTS TIM4_CH4 LCD_SEG3 LPUART2_RTS
55	43	30	-	PB4	I/O	Medium	-	SPI1_MISO UART2_CTS

Pin number				Pin name	Pin type	Driving capability	Additional functions	Alternate functions
LQFP64	LQFP48	QFN32	SSOP24					
								TIM3_CH1 LCD_SEG4 TIM5_ETR
56	44	-	-	PB5	I/O	Medium	-	SPI1_MOSI TIM3_CH2 LCD_SEG5 LPTIM1_IN1 COMP2_OUT
57	45	31	-	PB6	I/O	Medium	-	UART2_TX TIM5_CH3 LCD_SEG6 TIM5_CH2 LPTIM1_ETR
58	46	32	-	PB7	I/O	Medium	-	UART2_RX LCD_SEG7 UART4_CTS LPTIM1_IN2
59	47	-	-	PB8	I/O	Medium	-	MCO TIM4_ETR LCD_SEG8 TIM5_CH1 I2C1_SCL
60	48	-	-	PB9	I/O	Medium	-	IR_OUT LCD_SEG9 TIM5_CH2 I2C1_SDA
61	-	-	-	PD2	I/O	High/Medium configurable	-	MCO USART1_RTS_DE_CK SPI1_NSS UART2_RTS LPTIM1_IN1 IR_OUT
62	-	-	-	PD3	I/O	Medium	-	SPI1_SCK USART1_TX LCD_SEG25

Pin number				Pin name	Pin type	Driving capability	Additional functions	Alternate functions
LQFP64	LQFP48	QFN32	SSOP24					
								UART2_TX LPTIM1_IN2 TIM4_ETR TIM5_ETR
63	-	-	-	PD4	I/O	Medium	COMP2_INM	SPI1_MOSI UART4_TX TIM5_CH4 LCD_SEG26 UART2_RX LPTIM1_ETR USART1_RX TIM4_CH3
64	-	-	-	PD5	I/O	Medium	COMP2_INP	SPI1_MISO UART4_RX COMP2_OUT LCD_SEG27 UART2_RTS LPTIM1_OUT USART1_CTS TIM5_CH4

Table 3-2 Port alternate function mapping

PORT	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7
PA0	SPI2_SCK	USART1_CTS	-	-	UART4_TX	LPTIM1_OUT	-	COMP1_OUT
PA1	SPI1_SCK	USART1_RX	TIM4_CH4	-	UART4_RX	-	LPUART2_CTS	-
PA2	SPI1_MOSI	USART1_TX	TIM4_CH1	-	-	MCO	LPUART1_TX	COMP2_OUT
PA3	SPI2_MISO	USART1_RTS_DE_CK	TIM4_CH2	-	UART4_TX	MCO	LPUART1_RX	-
PA4	SPI1_NSS	SPI2_MOSI	TIM4_CH3	SEG32/SEG18/COM7	-	LPUART2_TX	I2C1_SDA	-
PA5	SPI1_SCK	IR_OUT	TIM4_CH4	SEG33/SEG19/COM6	UART3_RTS	LPUART2_RX	I2C1_SCL	COMP1_OUT
PA6	SPI1_MISO	TIM3_CH1	-	SEG34/SEG20/COM5	UART3_CTS	TIM5_CH1	LPUART1_CTS	LPTIM1_IN1
PA7	SPI1_MOSI	TIM3_CH2	-	SEG35/SEG21/COM4	TIM4_CH1	-	LPTIM1_IN2	COMP2_OUT
PA8	MCO	SPI2_NSS	UART3_TX	LCD_SEG16	TIM4_CH3	LPTIM1_IN1	LPUART2_TX	-
PA9	MCO	USART1_CTS	UART3_RX	LCD_SEG17	SPI2_MISO	TIM4_CH4	LPUART2_RX	LPTIM1_IN2
PA10	SPI2_MOSI	USART1_RTS_DE_CK	TIM4_CH4	LCD_COM0	TIM5_CH2	TIM5_CH1	TIM5_CH3	-
PA11	SPI1_MISO	USART1_TX	-	LCD_COM1	TIM5_CH3	-	TIM5_CH4	COMP1_OUT
PA12	SPI1_MOSI	USART1_RX	TIM4_ETR	LCD_COM2	TIM5_CH4	-	-	COMP2_OUT

PORT	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7
PA13	SWDIO	USART1_TX	IR_OUT	-	-	-	-	LPUART2_TX
PA14	SWCLK	USART1_RX	-	-	-	-	-	LPUART2_RX
PA15	SPI1_NSS	-	-	LCD_SEG0	UART4_RTS	UART3_RTS	-	-
PB0	SPI1_NSS	TIM3_CH3	-	-	UART3_RX	LPTIM1_OUT	-	COMP1_OUT
PB1	-	TIM3_CH4	-	-	UART3_RTS	-	LPUART1_RTS	-
PB2	SPI2_MISO	-	-	-	UART3_TX	LPTIM1_OUT	-	-
PB3	SPI1_SCK	UART2_RTS	TIM4_CH4	LCD_SEG3	-	-	LPUART2_RTS	-
PB4	SPI1_MISO	UART2_CTS	TIM3_CH1	LCD_SEG4	-	TIM5_ETR	-	-
PB5	SPI1_MOSI	-	TIM3_CH2	LCD_SEG5	-	LPTIM1_IN1	-	COMP2_OUT
PB6	-	UART2_TX	TIM5_CH3	LCD_SEG6	TIM5_CH2	LPTIM1_ETR	-	-
PB7	-	UART2_RX	-	LCD_SEG7	UART4_CTS	LPTIM1_IN2	-	-
PB8	MCO	-	TIM4_ETR	LCD_SEG8	-	TIM5_CH1	I2C1_SCL	-
PB9	-	IR_OUT	-	LCD_SEG9	-	TIM5_CH2	I2C1_SDA	-
PB10	SPI2_MOSI	-	-	LCD_SEG10	UART3_TX	-	LPUART1_RX	COMP1_OUT
PB11	SPI2_SCK	-	-	LCD_SEG11	UART3_RX	-	LPUART1_TX	COMP2_OUT
PB12	SPI2_NSS	-	-	LCD_SEG12	-	-	LPUART1_RTS	-
PB13	SPI2_SCK	TIM4_ETR	-	LCD_SEG13	UART3_CTS	-	LPUART1_CTS	-
PB14	SPI2_MISO	-	-	LCD_SEG14	UART3_RTS	TIM4_CH1	LPUART2_CTS	-
PB15	SPI2_MOSI	TIM4_CH2	-	LCD_SEG15	-	-	LPUART2_RTS	COMP1_OUT
PC0	-	SPI2_SCK	TIM4_CH1	LCD_SEG18	-	LPTIM1_OUT	LPUART1_TX	-

PORT	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7
PC1	-	SPI2_MISO	TIM4_CH2	LCD_SEG19	-	LPTIM1_IN2	LPUART1_RX	I2C1_SDA
PC2	-	SPI2_MOSI	TIM4_CH3	LCD_SEG20	-	LPTIM1_IN1	-	I2C1_SCL
PC3	-	USART1_TX	-	LCD_SEG21	UART3_CTS	-	LPTIM1_ETR	-
PC4	-	USART1_RX	-	LCD_SEG22	UART3_RTS	UART3_RX	-	-
PC5	-	TIM3_CH1	USART1_RTS_DE_CK	LCD_SEG23	UART3_TX	-	LPUART1_TX	-
PC6	-	TIM3_CH2	USART1_CTS	LCD_SEG24	UART3_RX	-	LPUART1_RX	-
PC7	SPI1_NSS	TIM3_CH3	TIM4_CH3	LCD_SEG28	LPUART2_RX	-	UART3_RTS	I2C1_SCL
PC8	SPI1_SCK	TIM3_CH4	TIM4_CH4	LCD_SEG29	LPUART2_TX	LPTIM1_OUT	UART3_CTS	I2C1_SDA
PC9	-	-	-	LCD_COM3	-	-	-	-
PC10	SPI2_MOSI	UART3_RTS	-	LCD_SEG1	TIM3_ETR	-	-	-
PC11	SPI2_MISO	UART2_CTS	-	LCD_SEG2	-	-	LPUART2_CTS	
PC12	-	UART3_TX	TIM5_CH3	-	TIM4_CH2	IR_OUT	-	-
PC13	-	-	-	-	-	-	-	-
PC14	-	-	-	-	-	-	-	-
PC15	-	-	-	-	-	-	-	-
PD0	SPI1_MOSI	SPI2_NSS	TIM5_CH1	LCD_SEG30	-	LPTIM1_IN1	-	UART3_RX
PD1	SPI1_MISO	SPI2_SCK	TIM5_ETR	LCD_SEG31	-	LPTIM1_IN2	-	UART3_TX
PD2	MCO	USART1_RTS_DE_CK	SPI1_NSS	-	UART2_RTS	LPTIM1_IN1	-	IR_OUT

PORT	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7
PD3	SPI1_SCK	USART1_TX	-	LCD_SEG25	UART2_TX	LPTIM1_IN2	TIM4_ETR	TIM5_ETR
PD4	SPI1_MOSI	UART4_TX	TIM5_CH4	LCD_SEG26	UART2_RX	LPTIM1_ETR	USART1_RX	TIM4_CH3
PD5	SPI1_MISO	UART4_RX	COMP2_OUT	LCD_SEG27	UART2_RTS	LPTIM1_OUT	USART1_CTS	TIM5_CH4
PF0	TIM5_CH3	UART2_TX	UART4_RTS	-	LPUART2_CTS	-	LPUART1_CTS	I2C1_SCL
PF1	TIM5_CH4	UART2_RX	UART4_CTS	-	LPUART2_RTS	-	LPUART1_RTS	I2C1_SDA
PF2	SPI2_MISO	USART1_CTS	TIM4_ETR	-	TIM5_CH1	LPTIM1_ETR	-	-

4 System and memory architecture

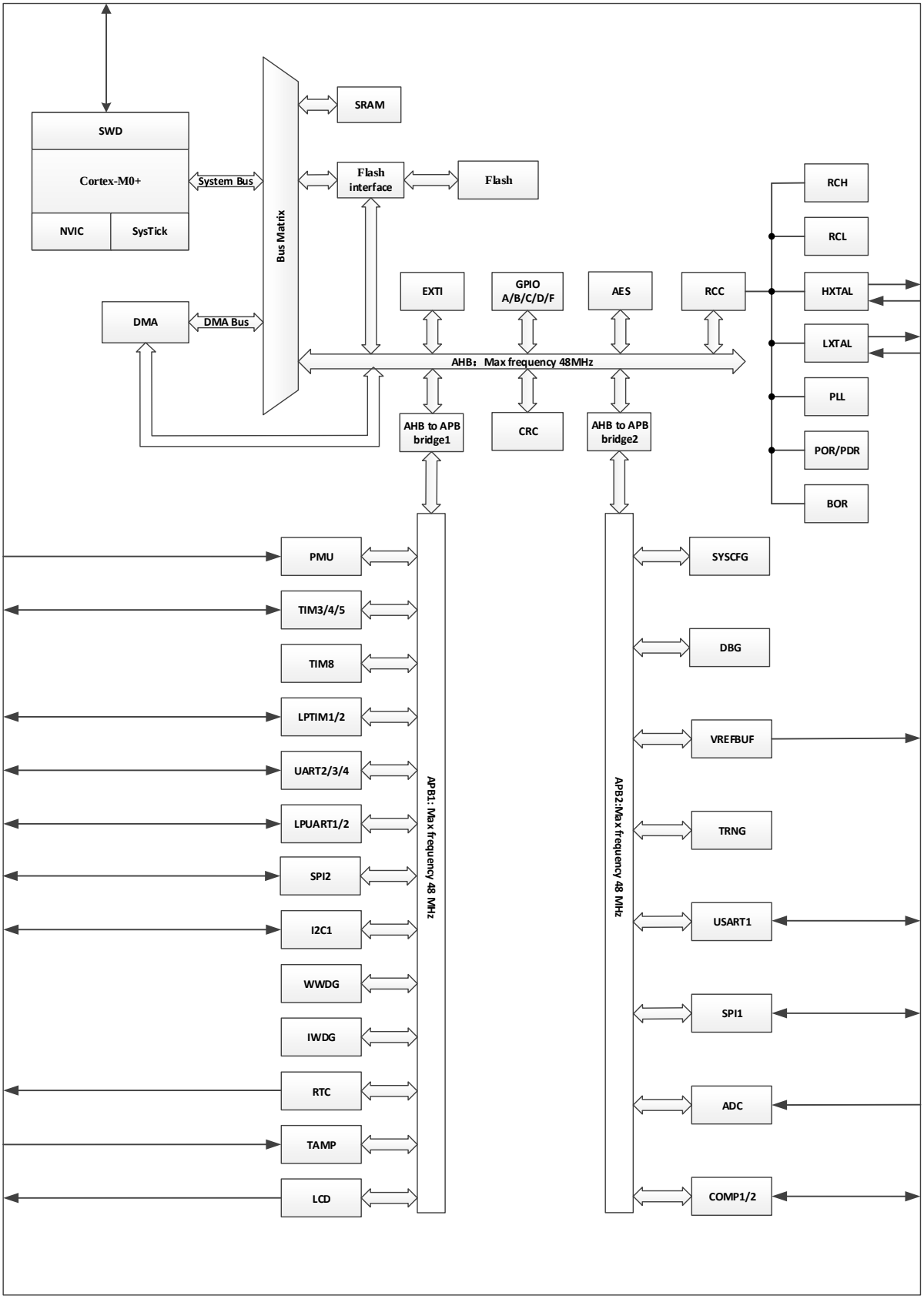
4.1 System architecture

The main system consists of:

- Two masters:
  - Cortex-M0+ core
  - DMA
- Three slaves:
  - Internal SRAM
  - Internal Flash memory
  - AHB: AHB with AHB-to-APB bridge that connects all the APB peripherals

These are interconnected using a multilayer AHB bus architecture as shown in the following:

Figure 4-1 System architecture



4.1.1 System bus

This bus connects the system bus of the Cortex-M0+ core to Bus Matrix, the core performs instruction fetching, data operations, and access to AHB/APB peripherals through this bus.

4.1.2 DMA bus

This bus connects the AHB bus of DMA to Bus Matrix, the DMA accesses the Flash, SRAM, and AHB/APB peripherals through this bus.

4.1.3 Bus matrix

The bus matrix manages the access arbitration between the core system bus and the DMA master bus. The bus matrix is composed of masters (CPU, DMA) and slaves (Flash interface, SRAM and AHB-to-APB bridge).

4.1.4 AHB to APB bridge

The AHB-to-APB bridge provides full synchronous connections between the AHB and the APB bus. The maximum frequency of APB1 and APB2 is up to 48 MHz.

4.2 Memory

4.2.1 SRAM

The devices have 16 Kbytes of embedded SRAM. The SRAM can be accessed by bytes, half-words (16 bits) or full words (32 bits), at maximum system clock frequency without wait state and thus by both CPU and DMA.

4.2.2 Flash

The Flash memory is composed of five distinct physical areas:

- User flash area: Up to 64 Kbytes. It contains the application program and user data if necessary.
- System memory area: Containing the Bootloader code.
- Option bytes area: Storing option bytes for peripherals and memory protection configuration.
- OTP area: One-time programmable area, size is 1 Kbytes.

- Engineer area: Storing product information and factory calibration parameters.

The Flash interface implements instruction access and data access based on the AHB protocol. It also implements the logic necessary to carry out the Flash memory operations (Program/Erase) controlled through the Flash registers.

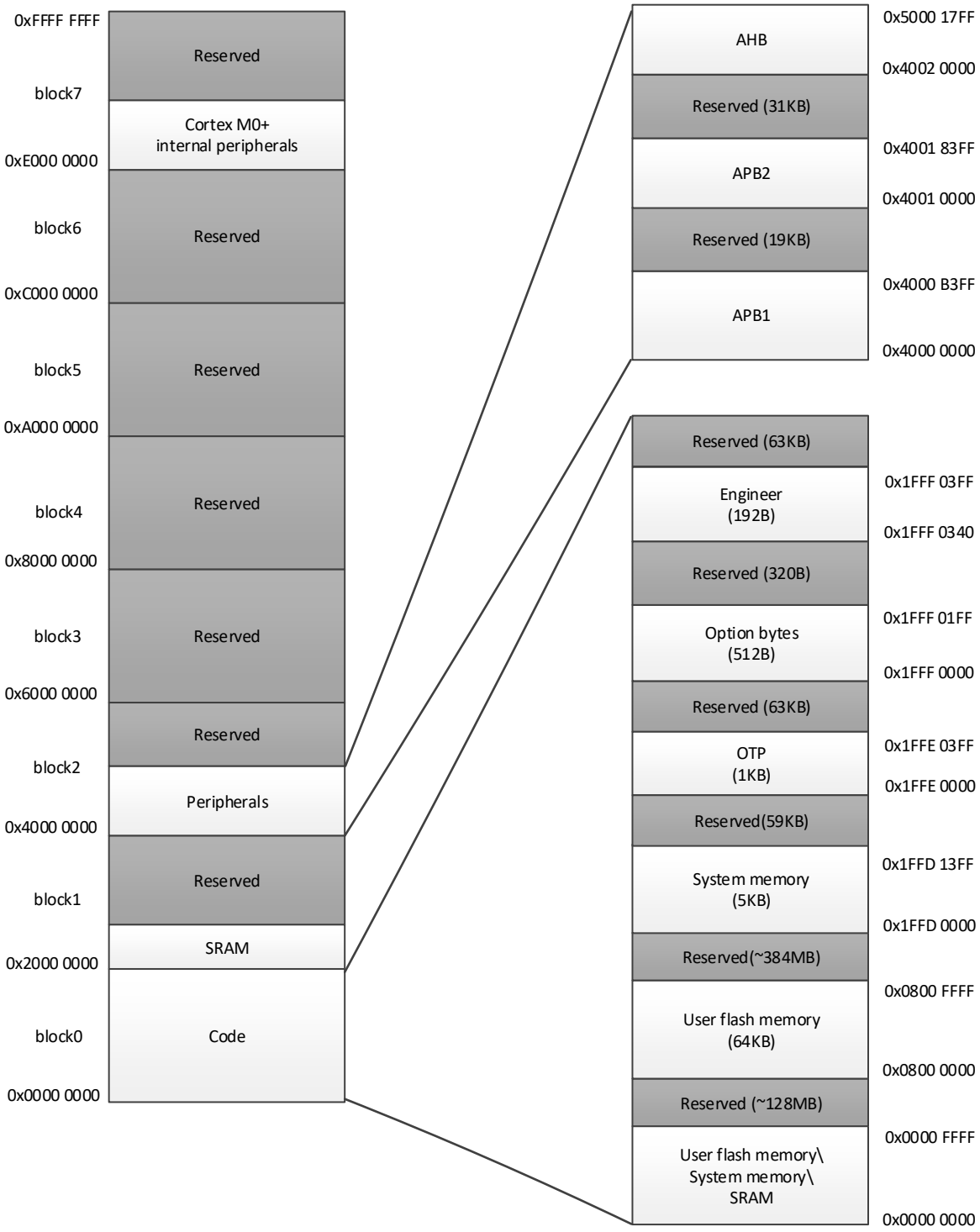
4.3 Memory map

Program memory, data memory, registers and I/O ports are organized within the same linear 4-Gbyte address space.

The bytes are coded in memory in Little Endian format. The lowest numbered byte in a word is considered the word's least significant byte and the highest numbered byte the most significant.

The addressable memory space is divided into eight main blocks, of 512 Mbytes each.

Figure 4-2 Memory map



All the memory map areas that are not allocated to on-chip memories and peripherals are considered “Reserved”. For the detailed mapping of available memory and register areas, refer to the following tables.

Table 4-1 Memory boundary addresses

Type	Boundary address	Size	Memory area
SRAM	0x2000 4000 - 0x3FFF FFFF	~512 MB	Reserved
	0x2000 0000 - 0x2000 3FFF	16 KB	SRAM
Flash	0x1FFF 0400 - 0x1FFF FFFF	63 KB	Reserved
	0x1FFF 0340 - 0x1FFF 03FF	192 B	Engineer
	0x1FFF 0200 - 0x1FFF 033F	320 B	Reserved
	0x1FFF 0000 - 0x1FFF 01FF	512 B	Option bytes
	0x1FFE 0400 - 0x1FFE FFFF	63 KB	Reserved
	0x1FFE 0000 - 0x1FFE 03FF	1 KB	OTP
	0x1FFD 1400 - 0x1FFD FFFF	59 KB	Reserved
	0x1FFD 0000 - 0x1FFD 13FF	5 KB	System memory
	0x0801 0000 - 0x1FFC FFFF	~384 MB	Reserved
	0x0800 0000 - 0x0800 FFFF	64 KB	User flash memory

Table 4-2 Peripheral register boundary addresses

Bus	Boundary address	Size	Peripheral
CPU	0xE000 0000 - 0xE00F FFFF	1 MB	Cortex-M0+ internal peripherals
AHB	0x5000 1800 - 0x5FFF FFFF	~256 MB	Reserved
	0x5000 1400 - 0x5000 17FF	1 KB	GPIOF
	0x5000 1000 - 0x5000 13FF	1 KB	Reserved
	0x5000 0C00 - 0x5000 0FFF	1 KB	GPIOD
	0x5000 0800 - 0x5000 0BFF	1 KB	GPIOC
	0x5000 0400 - 0x5000 07FF	1 KB	GPIOB
	0x5000 0000 - 0x5000 03FF	1 KB	GPIOA
	0x4002 6400 - 0x4FFF FFFF	~256 MB	Reserved
	0x4002 6000 - 0x4002 63FF	1 KB	AES
	0x4002 3400 - 0x4002 5FFF	11 KB	Reserved
	0x4002 3000 - 0x4002 33FF	1 KB	CRC
	0x4002 2400 - 0x4002 2FFF	3 KB	Reserved
	0x4002 2000 - 0x4002 23FF	1 KB	FLASH
	0x4002 1C00 - 0x4002 1FFF	1 KB	Reserved
	0x4002 1800 - 0x4002 1BFF	1 KB	EXTI
	0x4002 1400 - 0x4002 17FF	1 KB	Reserved

Bus	Boundary address	Size	Peripheral
	0x4002 1000 - 0x4002 13FF	1 KB	RCC
	0x4002 0400 - 0x4002 0FFF	3 KB	Reserved
	0x4002 0000 - 0x4002 03FF	1 KB	DMA
APB2	0x4001 8400 - 0x4001 FFFF	31 KB	Reserved
	0x4001 8000 - 0x4001 83FF	1 KB	TRNG
	0x4001 5C00 - 0x4001 7FFF	9 KB	Reserved
	0x4001 5800 - 0x4001 5BFF	1 KB	DBG
	0x4001 3C00 - 0x4001 57FF	7 KB	Reserved
	0x4001 3800 - 0x4001 3BFF	1 KB	USART1
	0x4001 3400 - 0x4001 37FF	1 KB	Reserved
	0x4001 3000 - 0x4001 33FF	1 KB	SPI1
	0x4001 2800 - 0x4001 2FFF	2 KB	Reserved
	0x4001 2400 - 0x4001 27FF	1 KB	ADC
	0x4001 0400 - 0x4001 23FF	8 KB	Reserved
	0x4001 0200 - 0x4001 03FF	1 KB	COMP1/2
	0x4001 01B0 - 0x4001 01FF		VREFBUF
	0x4001 0000 - 0x4001 01AF		SYSCFG
APB1	0x4000 B400- 0x4000 FFFF	19 KB	Reserved
	0x4000 B000- 0x4000 B3FF	1 KB	TAMP (+Backup registers)
	0x4000 9400- 0x4000 AFFF	7 KB	Reserved
	0x4000 9000 - 0x4000 93FF	1 KB	LPTIM2
	0x4000 8800 - 0x4000 8FFF	2 KB	Reserved
	0x4000 8400 - 0x4000 87FF	1 KB	LPUART2
	0x4000 8000 - 0x4000 83FF	1 KB	LPUART1
	0x4000 7C00 - 0x4000 7FFF	1 KB	LPTIM1
	0x4000 7400 - 0x4000 7BFF	2 KB	Reserved
	0x4000 7000 - 0x4000 73FF	1 KB	PMU
	0x4000 5800 - 0x4000 6FFF	6 KB	Reserved
	0x4000 5400 - 0x4000 57FF	1 KB	I2C1
	0x4000 5000 - 0x4000 53FF	1 KB	Reserved
	0x4000 4C00 - 0x4000 4FFF	1 KB	UART4
	0x4000 4800 - 0x4000 4BFF	1 KB	UART3

Bus	Boundary address	Size	Peripheral
	0x4000 4400 - 0x4000 47FF	1 KB	UART2
	0x4000 3C00 - 0x4000 43FF	2 KB	Reserved
	0x4000 3800 - 0x4000 3BFF	1 KB	SPI2
	0x4000 3400 - 0x4000 37FF	1 KB	Reserved
	0x4000 3000 - 0x4000 33FF	1 KB	IWDG
	0x4000 2C00 - 0x4000 2FFF	1 KB	WWDG
	0x4000 2800 - 0x4000 2BFF	1 KB	RTC
	0x4000 2400 - 0x4000 27FF	1 KB	LCD
	0x4000 1400 - 0x4000 23FF	4 KB	Reserved
	0x4000 1000 - 0x4000 13FF	1 KB	TIM8
	0x4000 0C00 - 0x4000 0FFF	1 KB	TIM5
	0x4000 0800 - 0x4000 0BFF	1 KB	TIM4
	0x4000 0400 - 0x4000 07FF	1 KB	TIM3
	0x4000 0000 - 0x4000 03FF	1 KB	Reserved

4.4 Boot configuration

Different boot modes can be selected through the BOOT0 pin and option bytes (RDP, BOOT_SEL, BOOT0_SW, and BOOT1_SW), as shown in the following table.

Table 4-3 Boot modes

RDP	BOOT_SEL	BOOT0 pin(1)	BOOT0_SW bit	BOOT1_SW bit	Boot mode
0	0	0	x	x	User flash
		1	x	1	Bootloader
		1	x	0	SRAM
	1	x	1	x	User flash
		x	0	1	Bootloader
		x	0	0	SRAM
1	0	0	x	x	User flash
		1	x	x	Bootloader
	1	x	1	x	User flash
		x	0	x	Bootloader
2	x	x	x	x	User flash

1. When the BOOT0 pin is multiplexed as LCD function, it is recommended to use BOOT_SEL, BOOT0_SW, and BOOT1_SW to configure the boot mode, in case the boot mode is selected incorrectly during the system reset.

After reset, according to the selected boot mode, the corresponding memory space is remapped to address 0x0000 0000, the CPU fetches the top-of-stack value from address 0x0000 0000, then starts code execution from the boot memory at 0x0000 0004.

To change the boot mode by modifying the option bytes, the modified boot mode will not take effect immediately. The boot mode can be made effective through the following methods:

- POR/PDR reset
- Option byte loading

Depending on the selected boot mode, user flash memory, Bootloader or SRAM is accessible as follows:

- Boot from user flash memory: The user flash memory is aliased in the boot memory space (0x0000 0000), but still accessible from its original memory space (0x0800 0000). In other words, the user flash memory contents can be accessed starting from address 0x0000 0000 or 0x0800 0000.
- Boot from Bootloader: The Bootloader is aliased in the boot memory space (0x0000 0000), but still accessible from its original memory space (0x1FFD 0000).
- Boot from the embedded SRAM: The SRAM is aliased in the boot memory space (0x0000 0000), but it is still accessible from its original memory space (0x2000 0000).

When RDP is level 2, the device will force a unique entry point in the user flash memory for boot, regardless of the other boot mode configuration bits.

4.4.1 Memory remap

Once the boot mode is selected, the application software can access the remapped memory space through the address 0x0000 0000, and can also modify the memory accessible in the remap area. This modification is performed by programming the MEM_MODE[1:0] bits in *System control register (SYSCFG_CR)*.

4.4.2 Bootloader

The embedded Bootloader is located in System memory, it is used to reprogram the

user flash memory using the following serial interface:

- USART1: PA11/PA12

For further details, refer to the Bootloader User Manual (UM1000).

5 Embedded Flash memory (FLASH)

5.1 Introduction

The Flash memory is connected to the AHB bus and managed by the Flash controller. It implements instruction fetch, read, program, and erase operations on the memory, while incorporating secure access mechanisms.

5.2 Flash main features

- Up to 64 Kbytes main memory, page size: 512 bytes
- 32-bit wide data write, 8/16/32-bit wide data read
- Supports page erase and mass erase
- 256-byte fast programming
- Three read protection levels actived by option
- Two write protection areas selected by option

5.3 Flash functional description

5.3.1 Flash memory organization

The Flash memory is organized as 32-bit-wide memory cells that can be used for storing both code and data constants. The data width of write is 32-bit, and the data width of read is 8/16/32-bits.

The Flash memory is organized as follows:

- User flash: up to 64 Kbytes, used for storing user code and data, containing 128 pages of 512 bytes
- System memory: up to 5 Kbytes, containing the bootloader code
- Option bytes: used for storing peripheral and memory protection configurations
- OTP: 1 Kbytes OTP (one-time programmable) for user data
- Engineer: contains the device information and factory calibration data

The memory organization is shown in the following table.

Table 5-1 Flash memory organization(1)

Area	Addresses	Size (bytes)	Notes
User flash	0x0800 0000 – 0x0800 01FF	512	Page 0
	…	…	…
	0x0800 FE00 – 0x0800 FFFF	512	Page 127
System memory	0x1FFD 0000 – 0x1FFD 13FF	5K	System memory
OTP	0x1FFE 0000 – 0x1FFE 03FF	1K	One-time programmable area
Option bytes	0x1FFF 0000 – 0x1FFF 01FF	512	Option bytes
Engineer	0x1FFF 0340 – 0x1FFF 03FF	192	Device information and factory calibration data

5.3.2 Flash read access latency

The LATENCY bit in the FLASH_ACR register is used to configure the number of wait states for Flash read access. The relationship between the HCLK clock frequency and the Flash wait states for read access is shown in the following table.

Table 5-2 Number of wait states according to HCLK frequency

Frequency of HCLK	Wait states	LATENCY configuration
HCLK ≤ 16 MHz	0 HCLK	LATENCY = 0
16 MHz < HCLK ≤ 32 MHz	1 HCLK	LATENCY = 1
HCLK > 32 MHz	2 HCLK	LATENCY = 2

When changing the HCLK, the following software sequences must be applied in order to tune the number of wait states needed to access the Flash memory:

- Increasing the HCLK frequency:
  1) Configure the LATENCY bits of the *FLASH access control register (FLASH_ACR)* to increase the number of wait states for Flash read access.
  2) Read the LATENCY bits to check if the new value of wait states have been configured successfully.
  3) When increasing the HCLK frequency, configure the SYSW[2:0] bits of the *Clock configuration register (RCC_CFG)* to switch to a higher frequency clock source, or configure the HPRE[2:0] bits to reduce the system clock prescaler value.
  4) Confirm that the system clock has been successfully switched.
- Decreasing the CPU frequency:

1) When decreasing the HCLK frequency, configure the SYSW[2:0] bits of the *Clock configuration register (RCC_CFG)* to switch to a lower frequency clock source, or configure the HPRE[2:0] bits to increase the system clock prescaler value.
2) Confirm that the system clock has been successfully switched.
3) Configure the LATENCY bits of the *FLASH access control register (FLASH_ACR)* to reduce the number of wait states for Flash read access.
4) Read the LATENCY bits to check if the new value of wait states have been configured successfully.

5.3.3 Unlocking the Flash memory

To prevent accidental modification of the Flash, protection measures have been added. Writting a key to a specific register to unlock the configuration permissions for related functions.

Unlocking the Flash control register

After reset, the FLASH_CR register is locked. The following sequence unlocks this register:

1) Write KEY1 = 0xE57A 1A85 to the FLASH_CRKEY register.
2) Write KEY2 = 0x7C6E 8391 to the FLASH_CRKEY register.
3) Confirm the FLASH_CR register is unlocked by checking the LOCK bit of the FLASH_CR register is cleared.

All bits except OBL_LAUNCH and OPT_ERASE in the FLASH_CR can be written when it is unlocked.

Any wrong sequence will lock the FLASH_CR register until the next system reset. In the case of a wrong key sequence, a bus error is detected.

The FLASH_CR register can be locked again by software by setting the LOCK bit or reset.

Unlocking the Option bytes

After reset, the Flash option bytes are locked. The OBL_LAUNCH and OPT_ERASE bits in the FLASH_CR register are also write-protected. To update the option bytes, an unlock operation must be performed first.

The following sequence is used to unlock this register:

1) Unlock the FLASH_CR with the LOCK clearing sequence.
2) Write OPTKEY1 = 0x6A89 4D7B to the FLASH_OPTKEY register.
3) Write OPTKEY2 = 0x7C31 1F5A to the FLASH_OPTKEY register.
4) Confirm the option bytes is unlocked by checking the OPTLOCK bit of the FLASH_CR register is cleared.

The OBL_LAUNCH and OPT_ERASE bits in FLASH_CR can be written when Option bytes are unlocked.

Any wrong sequence will lock up the Flash memory option registers until the next system reset. In the case of a wrong key sequence, a bus error is detected.

The option bytes can be locked again by software by setting the OPTLOCK or reset.

If the LOCK bit is set by software, the OPTLOCK bit is automatically set as well and the option bytes are locked again.

5.3.4 User flash erase operations

The user flash erase operation can be performed at following levels:

- Page erase (512 bytes)
- Mass erase

User flash page erase sequences

To erase a page, follow the procedure below:

1) Check clock of Flash registers is enabled by checking the FLASHEN bit of the *AHB peripheral clock enable register (RCC_AHBEN)*.
2) Check that no Flash memory operation is ongoing by checking the BSY bit of the FLASH_SR register.
3) Check and clear all error flags in the FLASH_SR register due to a previous programming.
4) Unlock the FLASH_CR register by clearing the LOCK bit.
5) Write the ER_MODE bits to 01 in the FLASH_CR register to enable page erase mode.
6) Select the page to erase (PNB) in the FLASH_CR register.
7) Set the ERASE bit of the FLASH_CR register to start erase operation. Then the BSY bit is set automatically.
8) Wait until the BSY bit of the FLASH_SR register is cleared and then the ERASE

bit is cleared automatically.

9) Repeat steps 6 to 8 to erase different pages.
10) Write the ER_MODE bits to 00 if there is not erase request anymore.
11) Set the LOCK bit of the FLASH_CR register to lock the FLASH_CR register.

User flash mass erase sequences

To perform a mass erase to erase the whole user flash, follow the procedure below:

1) Check clock of Flash registers is enabled by checking the FLASHEN bit of the *AHB peripheral clock enable register (RCC_AHBEN)*.
2) Check that no Flash memory operation is ongoing by checking the BSY bit of the FLASH_SR register.
3) Check and clear all error flags in the FLASH_SR register due to a previous programming.
4) Unlock the FLASH_CR register by clearing the LOCK bit.
5) Write the ER_MODE bits to 11 in the FLASH_CR register to enable mass erase mode.
6) Set the ERASE bit of the FLASH_CR register to start erase operation. Then the BSY bit is set automatically.
7) Wait until the BSY bit of the FLASH_SR register is cleared and then the ERASE bit is cleared automatically.
8) Write the ER_MODE bits to 00 if there is not erase request anymore.
9) Set the LOCK bit of the FLASH_CR register to lock the FLASH_CR register.

5.3.5 User flash standard program operation

The Flash memory is programmed 32 bits at a time.

User flash standard programming sequences

The Flash memory programming sequence is as follows:

1) Check clock of Flash registers is enabled by checking the FLASHEN bit of the *AHB peripheral clock enable register (RCC_AHBEN)*.
2) Check that no Flash memory operation is ongoing by checking the BSY bit of the FLASH_SR register.
3) Check and clear all error flags in the FLASH_SR register due to a previous programming.

4) Unlock the FLASH_CR register by clearing the LOCK bit.
5) Set the PG_MODE bit of the FLASH_CR register to enable program mode.
6) Write a word (32 bits) at the desired memory address, then the BSY bit is set automatically.
7) Wait until the BSY bit of the FLASH_SR register is cleared.
8) Repeat steps 6 to 7 to write different addresses.
9) Clear the PG_MODE bit of the FLASH_CR register if there no more programming request anymore.
10) Set the LOCK bit of the FLASH_CR register to lock the FLASH_CR register.

5.3.6 User flash fast program operation

Fast programming user flash is executed in the RAM. This mode allows programming a half page (256 bytes), so the lower 8-bit of the write address is 0.

User flash fast programming sequences

The Flash memory programming sequence in fast mode is as follows:

1) Check clock of Flash registers is enabled by checking the FLASHEN bit of the *AHB peripheral clock enable register (RCC_AHBEN)*.
2) Check that no Flash memory operation is ongoing by checking the BSY bit of the FLASH_SR register.
3) Check and clear all error flags in the FLASH_SR register due to a previous programming.
4) Unlock the FLASH_CR register by clearing the LOCK bit.
5) Set the PG_MODE bit and the FSTPG_MODE bit of the FLASH_CR register to enable fast program mode.
6) Write a word (32bits) at the desired memory address, then the BSY bit is set automatically.
7) Wait until the BSY bit of the FLASH_SR register is cleared.
8) Repeat steps 6 to 7 to write 64 words, then the FSTPG_MODE bit is cleared automatically to exit fast program mode.
9) If an error occurs, the FSTERR bit is set automatically, an interrupt is generated if OPERRIE is enabled.
10) Clear the PG_MODE bit of the FLASH_CR register if there no more programming request anymore.

11) Set the LOCK bit of the FLASH_CR register to lock the FLASH_CR register.

5.4 Option bytes description

5.4.1 Option bytes organization

The option bytes are stored in the Option bytes area of the Flash memory. The option bytes are configured by the end user depending on the application requirements.

To guarantee the accuracy, a word is split up in option bytes. The lower 16 bits store the option bytes, while the higher 16 bits store the complement of the option bytes.

The organization of these bytes is shown in the following table.

Table 5-3 Organization of option bytes

Addresses	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1FFF 0000	Reserved					BOR_LEVEL		BOR_EN	RDP							
0x1FFF 0004	Reserved	BOOT_SEL	BOOT0_SW	BOOT1_SW	Reserved			VBAT_MODE_EN	Reserved		IWDG_STOP	Reserved				RST_STOP
0x1FFF 0008	Reserved							WRP1A_START								
0x1FFF 000C	Reserved							WRP1A_END								
0x1FFF 0010	Reserved							WRP1B_START								
0x1FFF 0014	Reserved							WRP1B_END								

After reset, option bytes are loaded into internal option byte registers automatically. Refer to the detailed registers descriptions for the function of each control bit as following:

- *FLASH option register 1 (FLASH_OPTR1)*
- *FLASH option register 2 (FLASH_OPTR2)*
- *FLASH WRP area A start address register (FLASH_WRP1AS)*

- *FLASH WRP area A end address register (FLASH_WRP1AE)*
- *FLASH WRP area B start address register (FLASH_WRP1BS)*
- *FLASH WRP area B end address register (FLASH_WRP1BE)*

5.4.2 Option bytes erase and program operation

Option bytes erase sequences

1) Check clock of Flash registers is enabled by checking the FLASHEN bit of the *AHB peripheral clock enable register (RCC_AHBEN)*.
2) Check that no Flash memory operation is ongoing by checking the BSY bit of the FLASH_SR register.
3) Check and clear all error flags in the FLASH_SR register due to a previous programming.
4) Unlock the FLASH_CR register by clearing the LOCK bit.
5) Set the OPT_ERASE bit of the FLASH_CR register to start option bytes erase operation. Then the BSY bit is set automatically.
6) Wait until the BSY bit of the FLASH_SR register is cleared and then the OPT_ERASE bit is cleared automatically.
7) Set the OPTLOCK bit of the FLASH_CR register to lock option bytes.
8) Set the LOCK bit of the FLASH_CR register to lock the FLASH_CR register if needed.

Option bytes programming sequences

Option bytes can only be written with 32 bits. The complementary values are automatically computed and written into the upper 16-bit upon setting the OPTSTRT bit.

The option bytes programming sequence is as follows:

1) Check clock of Flash registers is enabled by checking the FLASHEN bit of the *AHB peripheral clock enable register (RCC_AHBEN)*.
2) Check that no Flash memory operation is ongoing by checking the BSY bit of the FLASH_SR register.
3) Check and clear all error flags in the FLASH_SR register due to a previous programming
4) Unlock the option bytes by clearing the OPTLOCK bit.

5) Set the PG_MODE bit of the FLASH_CR register to enable program mode.
6) Write a word to option byte address sequentially. The complementary value is automatically computed and written into the upper 16-bit. The BSY bit is set automatically.
7) Wait until the BSY bit of the FLASH_SR register is cleared.
8) Repeat steps 6 to 7 to write the all option bytes.
9) Clear the PG_MODE bit of the FLASH_CR register.
10) Set the OPTLOCK bit of the FLASH_CR register to lock the option bytes.
11) Set the LOCK bit of the FLASH_CR register to lock the FLASH_CR register if needed.

Option bytes programming operation will be ignored and the data is not written yet if it's locked, and PROGERR is set.

After programming, all new options are updated into the option bytes, but not applied to the system. A read from the option registers will still return the last loaded option byte values, the new options will have effect on the system only after they are loaded.

5.4.3 Option bytes loading

To make the option bytes in the Option bytes area effective, option bytes loading is performed in two cases:

- POR/PDR
- When OBL_LAUNCH bit of the FLASH_CR register is set

Option byte loader automatically verifies the option block and their complements. Reset state is held if there is any error occurs.

After the option byte loading takes effect, option byte loader stores the data into internal option registers. A read from the option registers will always return the values after they are loaded.

5.5 OTP

OTP program operation is the same as User flash. The OTP data (32 bits) can be written only once. Refer to: *User flash standard program operation*.

The program operation is aborted if OTP cell is written again, and the PROGERR bit of the FLASH_SR register is set.

5.6 Engineer

Engineer area, which is factory-programmed, contains the device information and factory calibration data. It can only be read by the software.

Table 5-4 Device information

Addresses	Size (words)	Description
0x1FFF 0340	3	96-bit UID
0x1FFF 03A4	1	Chip type: 0x00000003: CIU32L05
0x1FFF 03A8	1	Size of Flash memory (Kbytes) 0x00000040：64
0x1FFF 03AC	1	Size of SRAM (Kbytes) 0x00000010：16

Table 5-5 Factory calibration parameter

Addresses	Bit[31:16]	Bit[15:0]	Description
0x1FFF 03C0	~BGR_CAL	BGR_CAL	BGR factory calibration value
0x1FFF 03C4	~TS_CAL_25	TS_CAL_25	TS 25°C factory calibration value
0x1FFF 03C8	~TS_CAL_85	TS_CAL_85	TS 85°C factory calibration value
0x1FFF 03CC	~RCHCAL	RCHCAL	RCH factory calibration value
0x1FFF 03D0	~RCLCAL	RCLCAL	RCL factory calibration value
0x1FFF 03D4	~VREFBUF_CAL_20	VREFBUF_CAL_20	VREFBUF 2.048V factory calibration value
0x1FFF 03D8	~VREFBUF_CAL_25	VREFBUF_CAL_25	VREFBUF 2.5V factory calibration value
0x1FFF 03DC	~VREFBUF_CAL_30	VREFBUF_CAL_30	VREFBUF 3.0V factory calibration value

5.7 Flash memory protection

Flash memory can be protected with the read protection RDP and the write protection WRP.

- Flash read protection (RDP): There are different levels to be set to protect the Flash memory.
- Flash write protection (WRP): When WRP is active, it cannot be erased or programmed. Only fetching and reading the Flash memory are possible.

5.7.1 Flash read protection (RDP)

There are three levels of read protection.

RDP 0:

- Boot from User flash/SRAM
  - User flash: Fetch, read, program and erase operations are possible
  - System memory: Fetch and read operations are possible
  - Option bytes: Read, program and erase operations are possible
  - OTP: Read and program operations are possible
  - Backup registers: Read and write operations are possible
- Boot from Bootloader
  - User flash: Fetch, read, program and erase operations are possible
  - System memory: Fetch and read operations are possible
  - Option bytes: Read, program and erase operations are possible
  - OTP: Read and program operations are possible
  - Backup registers: Inaccessible
- Debug mode
  - User flash: Fetch, read, program and erase operations are possible
  - System memory: Fetch and read operations are possible
  - Option bytes: Read, program and erase operations are possible
  - OTP: Read and program operations are possible
  - Backup registers: Read and write operations are possible

Note: The CPU debug port is disabled when boot form Bootloader.

RDP 1:

- Boot from User flash
  - User flash: Fetch, read, program and erase operations are possible

- System memory: Fetch and read operations are possible
- Option bytes: Read, program and erase operations are possible
- OTP: Read and program operations are possible
- Backup registers: Read and write operations are possible

- Boot from Bootloader
  - User flash: Fetch operation is possible
  - System memory: Fetch and read operations are possible
  - Option bytes: Read, program and erase operations are possible
  - OTP: Read and program operations are possible
  - Backup registers: Be totally inaccessible
- The boot from SRAM is not available
- The CPU debug port is not available

RDP 2:

- Boot from User flash
  - User flash: Fetch, read, program and erase operations are possible
  - System memory: Fetch and read operations are possible
  - Option bytes: Read operation is possible
  - OTP: Read and program operations are possible
  - Backup registers: Read and write operations are possible
- The boot from Bootloader is not available
- The boot from SRAM is not available
- The CPU debug port is not available

An unavailable access to the Flash memory will be aborted and generate a bus error and a Hard Fault interrupt.

In this level, erase and program operations to Option bytes are not available. If the OPT_ERASE bit of the FLASH_CR register is set or there is a write operation to

Option bytes, the erase and program operations will be ignored and a bus error and a Hard Fault interrupt are generated.

Set RDP level

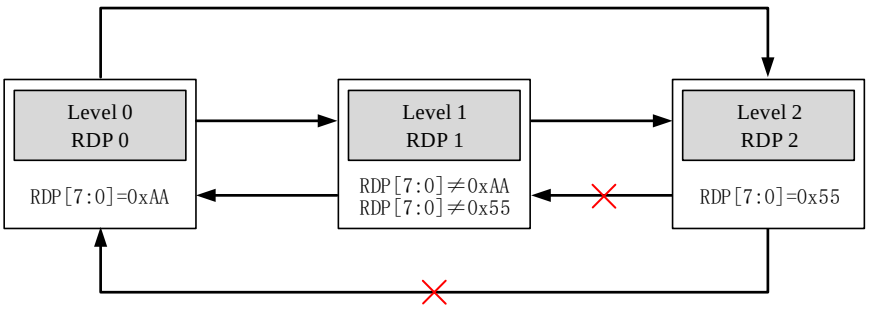
The RDP[7:0] bits in the FLASH_OPTR1 register are used to configure the RDP level. The protection level is validated when the Option bytes are reloaded. Refer to *Option bytes decription*.

Table 5-6 Flash memory read protection status

RDP level	RDP[7:0] value
RDP0	0xAA (default)
RDP1	Any values except 0xAA and 0x55
RDP2	0x55

The read protection level can change from RDP0 to RDP1, or from RDP0/1 to RDP2. RDP level increase (Level 0 to Level 1, 1 to 2, or 0 to 2) does not cause any change to the memories.

Figure 5-1 Changing read protection (RDP) level



RDP level decrease sequences

To decrease the RDP level, follow the procedure below:

1) Check that no Flash memory operation is ongoing by checking the BSY bit of the FLASH_SR register.
2) Check and clear all error flags in the FLASH_SR register due to a previous programming.
3) Unlock the option bytes by clearing the OPTLOCK bit of the FLASH_CR register.
4) Erase Option bytes and wait until the BSY bit of the FLASH_SR register is cleared.

5) Set the PG_MODE bit of the FLASH_CR register to enable program mode.
6) Change the value of the RDP byte to 0xAA in Option bytes area and the BSY bit is set automatically.
7) Wait until the BSY bit of the FLASH_SR register is cleared.

To avoid reading or modifying the data in the memory, the following operations is executed automatically when decreasing RDP level:

- Mass erase the User flash
- The RDP byte is updated to 0xAA, the other bits in OPTR1 are updated to the value of the option register
- OPTR2 is updated to the value of the option register
- WRP is disabled:

  WRP1y_START[6:0] is changed to 0x7F (y = A or B)

  WRP1y_END[6:0] is changed to 0x00
- Clear the backup registers

5.7.2 Flash write protection

WRP protects the User flash area. When WRP is active, only fetching and reading the area are possible. If an erase/program operation to a write-protected part of the Flash memory is attempted, the write protection error flag (WRPERR) of the FLASH_SR register is set and the operation is ignored.

Two write-protected (WRP) areas can be defined in the User flash area. The granularity is one page. WRP areas 1A and 1B can be independent or overlapping.

Each area is defined by a start page offset and an end page offset related to the physical Flash memory base address. These offsets are defined in Option bytes FLASH_WRP1yS and FLASH_WRP1yE. WRP is validated when the Option bytes are reloaded. Refer to *Option bytes decription*.

Table 5-7 WRP protection

WRP registers values (y = A or B)	WRP-protected area
WRP1y_START > WRP1y_END	None (unprotected)
WRP1y_START = WRP1y_END	Page WRP1y
WRP1y_START < WRP1y_END	Pages from WRP1y_START to WRP1y_END

The WRP area is defined from the address:

User flash base address (0x0800 0000) + WRP1y_START × 0x200

to the address:

User flash base address (0x0800 0000) + (WRP1y_END + 1) × 0x200 - 1

For example, to protect by WRP from the address 0x0800 3E00 to the address 0x0800 81FF:

1) Address 0x0800 3E00 is in Page 31, WRP1A_START = 31
2) Address 0x0800 81FF is in Page 64, WRP1A_END = 64

Modification of user options to clear WRP, to increase or to decrease the size of the WRP-protected areas when WRP is active. To validate the new WRP options, the option bytes must be reloaded by setting the OBL_LAUNCH bit in the FLASH_CR register.

WRP is deactivated when the RDP change from Level 1 to Level 0 and the User flash is mass erased.

5.8 Flash operation errors

The bits PROGERR, FSTERR, and WRPERR may be set during operating the User flash and Option bytes area. No erase and program operations start when the error flags are set (PROGERR, FSTERR, and WRPERR).

5.8.1 Program/erase error flag PROGERR

In case of the following errors, the Flash memory operation (programming or erasing) is aborted and the PROGERR bit is set.

- Program the flash memory when PG_MODE bit is 0.
- Not 32-bit data is written.
- A write access to OTP words which have been programmed.

- ER_MODE[1:0] is 00 or the operation area set by PNB is not allocated in flash memory when ERASE bit is set.
- Any two of the bits ERASE/OPT_ERASE/OBL_LAUNCH are set simultaneously.
- A write access to the Option bytes when they are locked (OPTLOCK = 1) or not erased.

5.8.2 Write protection error flag WRPERR

Following operations are aborted and WRPERR bit is set.

- An address to be erased belongs to a write-protected area
- An address to be programmed belongs to a write-protected area

5.8.3 Fast programming error flag FSTERR

Following errors can be detected in fast program mode. In case of error, FSTERR bit is set and FSTPG_MODE/BSY bits are cleared by hardware to exit the fast mode.

- Program the flash memory when PG_MODE bit is 0
- An address to be programmed belongs to a write-protected area or the address does not belong to User flash
- Not 32-bit data is written
- Read the flash.
- The upper 24-bits of the program addresses are not the same
- The interval between two words programming operations exceeds 96 μs

5.9 Flash interrupts

There is end of operation interrupt and abnormal operation interrupts.

Table 5-8 Flash interrupt requests

Interrupt event	Event flag	Interrupt enable control bit	Event flag/interrupt clearing method
End of operation	EOP	EOPIE	Write EOP = 1
Operation error	WRPERR/ PROGERR/ FSTERR	OPERRIE	Write WRPERR/ PROGERR/FSTERR = 1

5.10 Flash registers

The Flash registers can only be accessed by words (32-bit).

Table 5-9 Flash base address

Peripheral	Base address
Flash	0x4002 2000

5.10.1 Flash access control register (FLASH_ACR)

Address offset: 0x00

Reset value: 0x0000 0000

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

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

Bits	Name	Description
31:2	Reserved	Must be kept at reset value
1:0	LATENCY[1:0]	Flash memory access latency 00: Zero wait states 01: One wait state 10: Two wait states 11: Reserved (keep the latest configuration)

5.10.2 Flash key register (FLASH_CRKEY)

Address offset: 0x08

Reset value: 0x0000 0000

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

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

Bits	Name	Description
31:0	CRKEY[31:0]	To unlock the FLASH control register (FLASH_CR) Unlock the FLASH_CR register except OBL_LAUNCH and OPT_ERASE bits. Refer to the unlock sequences: *Unlocking Flash control register*

5.10.3 Flash option key register (FLASH_OPTKEY)

Address offset: 0x0C

Reset value: 0x0000 0000

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

Bits	Name	Description
31:0	OPTKEY[31:0]	To unlock the option bytes and related control bits The unlock range includes: OBL_LAUNCH and OPT_ERASE bits. Refer to the unlock sequences: *Unlocking Option bytes*

5.10.4 Flash status register (FLASH_SR)

Address offset: 0x10

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.							EOP	Res.							BSY
							rc_w1								r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.											WRP ERR	PROG ERR	FST ERR	Res.	
											rc_w1	rc_w1	rc_w1		

Bits	Name	Description
31:25	Reserved	Must be kept at reset value

24	EOP	End of operation Cleared by writing 1, and write 0 is invalid. Set by hardware when one or more Flash memory operation (program or erase) has been completed successfully. 0: Flash operation is in progress 1: Flash operation is finished
23:17	Reserved	must be kept at reset value
16	BSY	Flash operation status flag This bit is set at the beginning of a Flash memory operation, and reset when the operation finishes 0: Flash is idle 1: Flash is busy
15:5	Reserved	Must be kept at reset value
4	WRPERR	Write protection error Cleared by writing 1, and write 0 is invalid. 0: Operate correctly 1: WRP error Refer to: *Write protection error flag WRPERR*
3	PROGERR	Program/erase error Cleared by writing 1, and write 0 is invalid. 0: Operate correctly 1: Program or erase error Refer to: *Program/erase error flag PROGERR*
2	FSTERR	Fast program error Cleared by writing 1, and write 0 is invalid. 0: Operate correctly 1: Fast program error Refer to: *Fast programming error flag FSTERR*
1:0	Reserved	Must be kept at reset value

5.10.5 Flash control register (FLASH_CR)

Address offset: 0x14

Reset value: 0xC000 0000

Note: This register cannot be modified when BSY in FLASH status register (FLASH_SR) is set.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	OPT LOCK	Res.		OBL_ LAUNCH	Res.	OPERRIE	EOPIE	Res.							
rs	rs			rs		rw	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	PNB[6:0]							Res.		FSTPG_ MODE	OPT_ ERASE	ERASE	ER_MODE[1:0]		PG_ MODE
	rw	rw	rw	rw	rw	rw	rw			rs	rs	rs	rw	rw	rw

Bits	Name	Description
31	LOCK	FLASH_CR Lock When set, the FLASH_CR register is locked. It is cleared by hardware after detecting the unlock sequence. Refer to the unlock sequences: *Unlocking Flash control register*
30	OPTLOCK	Options Lock When set, OBL_LAUNCH and OPT_ERASE bits in FLASH_CR register and so option page is locked. This bit is cleared by hardware after detecting the unlock sequence. Refer to the unlock sequences: *Unlocking Option bytes*
29:28	Reserved	Must be kept at reset value
27	OBL_LAUNCH	Forces the option byte loading 0: Option byte loading complete 1: Option byte loading requested and system reset is generated
26	Reserved	Must be kept at reset value
25	OPERRIE	Program/erase error interrupt enable This bit enables the interrupt generation when the WRPERR/ PROGERR/FSTERR bits of the FLASH_SR register are set. 0: Disable

		1: Enable
24	EOPIE	End of program/erase operation interrupt enable This bit enables the interrupt generation when the EOP bit of the FLASH_SR register is set. 0: Disable 1: Enable
23:15	Reserved	Must be kept at reset value
14:8	PNB[6:0]	Page number selection 0x000: Page 0 0x001: Page 1 ⋮ 0x7E: Page 126 0x7F: Page 127
7:6	Reserved	Must be kept at reset value
5	FSTPG_MODE	Fast program mode Cleared by hardware when fast program is finished or an error occurred. 0: Fast program disabled 1: Fast program enabled
4	OPT_ERASE	To start Option bytes erase This bit triggers an options erase when set, and is cleared automatically when erase is finished.
3	ERASE	To start Flash erase This bit triggers an options erase when set, and is cleared automatically when erase is finished.
2:1	ER_MODE[1:0]	Flash erase mode 00: Erase disabled 01/10: Page erase enabled 11: Mass erase enabled

0	PG_MODE	Flash program mode 0: Flash memory program disabled 1: Flash memory program enabled

5.10.6 Flash option register 1 (FLASH_OPTR1)

Address offset: 0x20

Reset value: bit: 00000000 00000000 00000XXX XXXXXXXX

Factory default value: 0x0000 00AA

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					BOR_LEVEL[1:0]		BOR_EN	RDP[7:0]							
					r	r	r	r	r	r	r	r	r	r	r

Bits	Name	Description
31:11	Reserved	Must be kept at reset value
10:9	BOR_LEVEL[1:0]	BOR threshold 00: Level0 rising/falling (2.1/2.0V) 01: Level1 rising/falling (2.3/2.2V) 10: Level2 rising/falling (2.6/2.5V) 11: Level3 rising/falling (2.9/2.8V)
8	BOR_EN	BOR enable 0: BOR disabled 1: BOR enabled
7:0	RDP[7:0]	RDP level 0xAA: Level0 (RDP0) 0x55: Level2 (RDP2) Others: Level1 (RDP1)

5.10.7 Flash option register 2 (FLASH_OPTR2)

Address offset: 0x24

Reset value: bit: 00000000 00000000 0XXX000X 00X0000X

Factory default value: 0x0000 3121

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	BOOT_SEL	BOOT0_SW	BOOT1_SW	Res.			VBAT_MODE_EN	Res.		IWDG_STOP	Res.				RST_STOP
	r	r	r				r			r					r

Bits	Name	Description
31:15	Reserved	Must be kept at reset value
14	BOOT_SEL	Boot configuration, refer to: *Boot configuration*
13	BOOT0_SW	Boot configuration, refer to: *Boot configuration*
12	BOOT1_SW	Boot configuration, refer to: *Boot configuration*
11:9	Reserved	Must be kept at reset value
8	VBAT_MODE_EN	V_{BAT} mode enable 0: Disable 1: Enable
7:6	Reserved	Must be kept at reset value
5	IWDG_STOP	Independent watchdog counter freeze in Stop mode 0: Independent watchdog counter is frozen in Stop mode 1: Independent watchdog counter is running in Stop mode
4:1	Reserved	Must be kept at reset value
0	RST_STOP	0: Reset generated when entering the Stop mode 1: No reset generated when entering the Stop mode

5.10.8 Flash WRP area A start address register (FLASH_WRP1AS)

Address offset: 0x38

Reset value: bit: 00000000 00000000 00000000 0XXXXXXX

Factory default value: 0x0000 007F

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.									WRP1A_START[6:0]						
									r	r	r	r	r	r	r

Bits	Name	Description
31:7	Reserved	Must be kept at reset value
6:0	WRP1A_START[6:0]	WRP1A start offset The WRP areas are defined related to the physical Flash memory base address (0x0800 0000). If the flash is remapped to 0x0000 0000, the remapped areas and the original areas are all protected. WRP1A start address: User flash base address + WRP1A_START[6:0] × 0x200

5.10.9 Flash WRP area A end address register (FLASH_WRP1AE)

Address offset: 0x3C

Reset value: bit: 00000000 00000000 00000000 0XXXXXXX

Factory default value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.									WRP1A_END[6:0]						
									r	r	r	r	r	r	r

Bits	Name	Description
31:7	Reserved	Must be kept at reset value

6:0	WRP1A_END[6:0]	WRP1A end offset The WRP areas are defined related to the physical Flash memory base address (0x0800 0000). If the flash is remapped to 0x0000 0000, the remapped areas and the original areas are all protected. WRP1A end address: User flash base address + (WRP1A_END[6:0] + 1) × 0x200 – 1

5.10.10 Flash WRP area B start address register (FLASH_WRP1BS)

Address offset: 0x40

Reset value: bit: 00000000 00000000 00000000 0XXXXXXX

Factory default value: 0x0000 007F

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.									WRP1B_START[6:0]						
									r	r	r	r	r	r	r

Bits	Name	Description
31:7	Reserved	Must be kept at reset value
6:0	WRP1B_START[6:0]	WRP1B start offset The WRP areas are defined related to the physical Flash memory base address (0x0800 0000). If the flash is remapped to 0x0000 0000, the remapped areas and the original areas are all protected. WRP1B start address: User flash base address + WRP1B_START[6:0] × 0x200

5.10.11 Flash WRP area B end address register (FLASH_WRP1BE)

Address offset: 0x44

Reset value: bit: 00000000 00000000 00000000 0XXXXXXX

Factory default value: 0x0000 0000

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

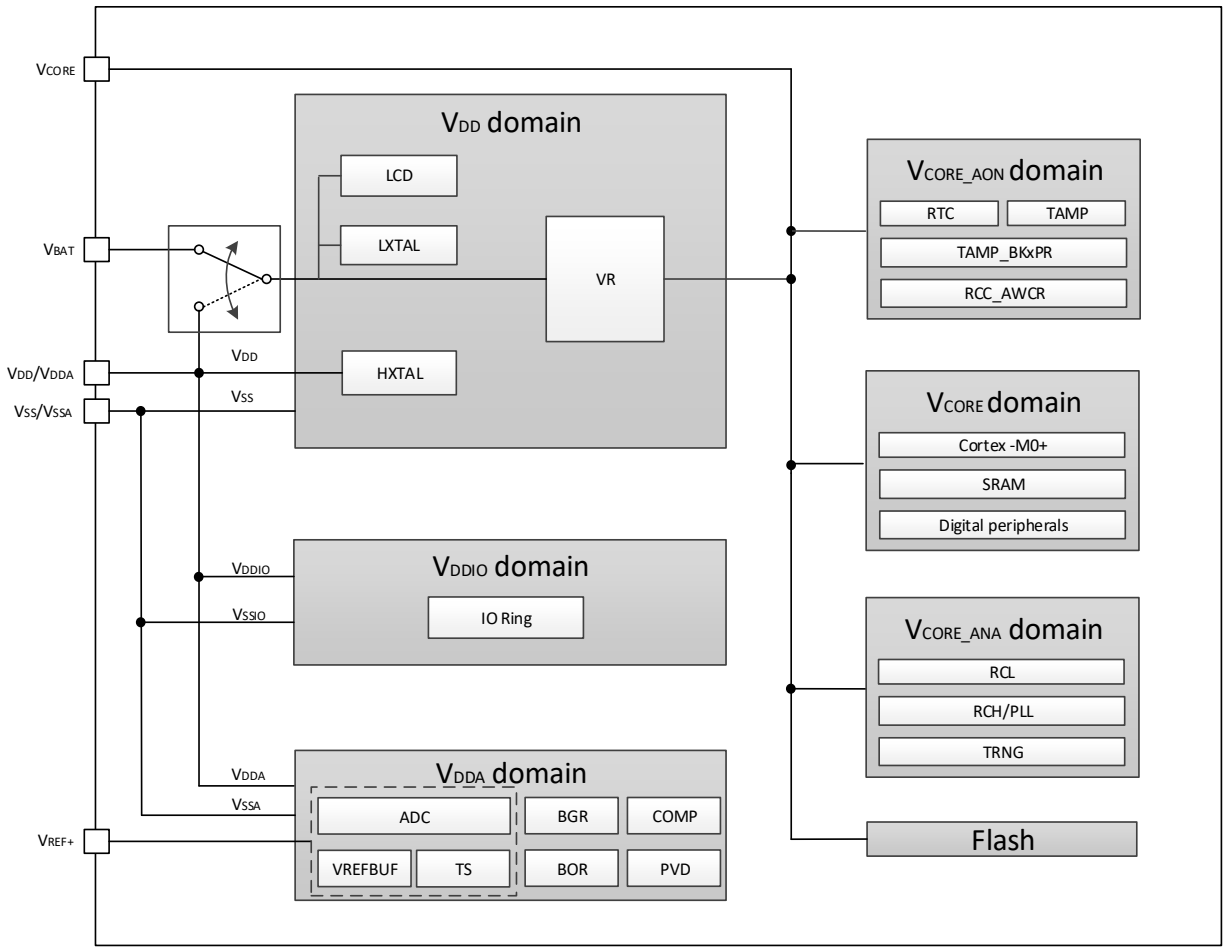
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.									WRP1B_END[6:0]						
									r	r	r	r	r	r	r

Bits	Name	Description
31:9	Reserved	Must be kept at reset value
6:0	WRP1B_END[6:0]	WRP1B end offset The WRP areas are defined related to the physical Flash memory base address (0x0800 0000). If the flash is remapped to 0x0000 0000, the remapped areas and the original areas are all protected. WRP1A end address: User flash base address + (WRP1B_END[6:0] + 1) × 0x200 – 1

6 Power management unit (PMU)

6.1 Power supplies

Figure 6-1 Power supply overview



Several different power supplies are provided to specific peripherals:

- V_{DD}/V_{DDA} = 1.8 V to 5.5 V. V_{DD}/V_{DDA} is the external power supply, supporting Power-On Reset/Power-Down Reset (POR/PDR) and Brown-Out Reset (BOR). It is provided externally through V_{DD}/V_{DDA} pin. See details in *Power supply supervisor.*

- V_{BAT} is the backup power supply. When using the backup power supply, V_{BAT} mode must be enabled (set VBAT_MODE_EN in the *Flash option register 2 (FLASH_OPTR2)* to 1). The input voltage for the V_{BAT} pin should be in the range of 1.55 V to 4.2 V. If the backup power supply is not used, V_{BAT} mode must be disabled (set VBAT_MODE_EN in the *Flash option register 2 (FLASH_OPTR2)* to 0). In this case, the V_{BAT} pin can be shorted to the V_{DD}/V_{DDA} pins, and the

input voltage is in the range of 1.8 V to 5.5 V.

- V_{REF+} is the input reference voltage for the ADC and 6-bit DAC, or the output of the VREFBUF.
- V_{CORE} pin requires an external capacitor. It is recommended to use a 1 μF+0.1 μF capacitor.

6.1.1 V_{REF+}

When ADC, VREFBUF and the 6-bit DAC are not using V_{REF+}, V_{REF+} can be used as a GPIO (PA0).

V_{REF+} input reference voltage

To ensure a better accuracy on low-voltage inputs and outputs, the user can connect to V_{REF+} a separate reference voltage lower than V_{DDA}. V_{REF+} is the highest voltage, represented by the full scale value, for an analog input (ADC) or output (DAC) signal.

In this mode, the requirement is that $V_{REF+} \leq V_{DDA}$.

V_{REF+} output reference voltage

The VREFBUF can also provide the voltage to external components through V_{REF+} pin. VREFBUF supports multiple voltage outputs, which are determined by the VRS[1:0] bits in the *VREFBUF control and status register (VREFBUF_CSR)*.

6.1.2 V_{BAT}

The V_{BAT} pin can be connected to an optional backup voltage supply. When V_{DD} is turned off, the power for the regulator switches from V_{DD} to V_{BAT}. The V_{BAT} pin powers the RTC, TAMP, LXTAL, backup registers, RCC_AWCR register, and some I/O pins (PC13/PC14/PC15 pins) in the V_{CORE_AON} domain. Conversely, the power for the regulator is V_{DD}.

When V_{BAT} is supplying power, only the PC13/PC14/PC15 pins are available, see details in *V_{BAT} mode.*

The V_{BAT} is determined by the VBAT_MODE_EN bit in the *Flash option register 2 (FLASH_OPTR2)*. The connection of V_{BAT} pin should be consistent with the V_{BAT} mode configuration:

- Non-V_{BAT} mode: The V_{BAT} pin should be shorted to the V_{DD}/V_{DDA} pins, and VBAT_MODE_EN must be configured to 0.

- V_{BAT} mode: The V_{BAT} pin connects to a backup power supply, and VBAT_MODE_EN must be configured to 1.

6.2 Power supply supervisor

6.2.1 Power-on reset (POR) / power-down reset (PDR)

The device features an integrated power-on reset (POR) / power-down reset (PDR), The POR/PDR is active in all power modes.

The reset temporization $t_{RSTTEMPO}$ starts when V_{DD} crosses POR threshold. During power-down, when V_{DD} drops below the PDR threshold, the device is put under reset immediately.

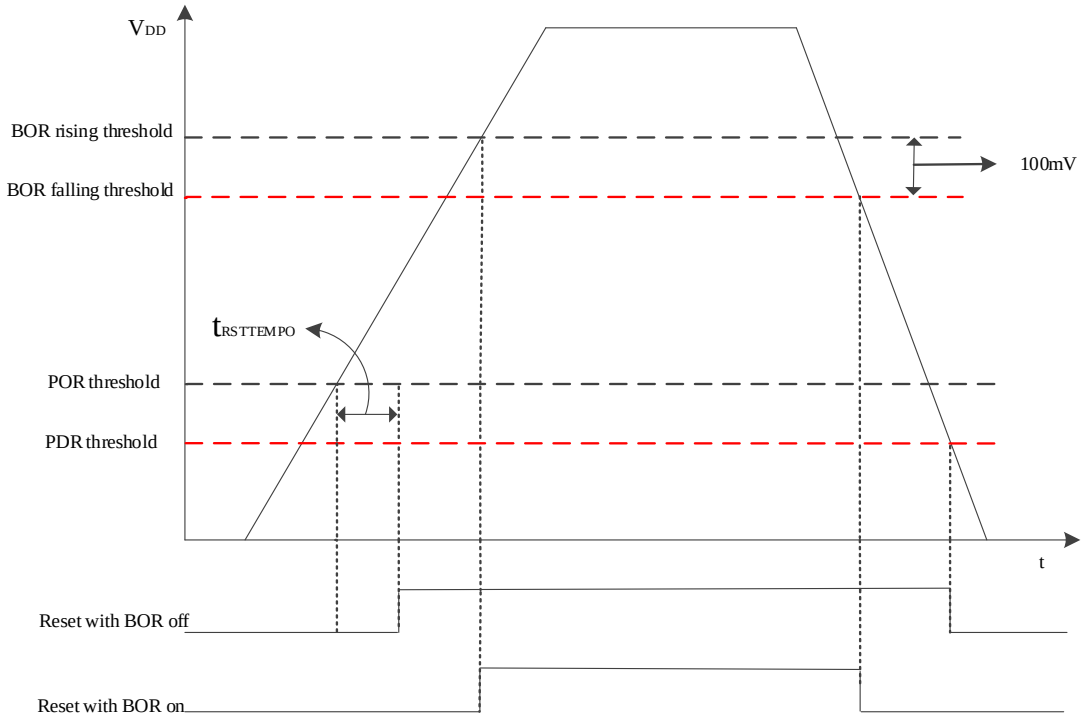
6.2.2 Brown-out reset (BOR)

The BOR can be enabled or disabled through option bytes, with 4 selectable threshold levels. See details in *Flash option register 1 (FLASH_OPTR1).*

Each BOR threshold level sets both the rising and falling thresholds.

- When BOR is enabled, the reset is released when the V_{DD} is higher than the configured BOR rising threshold; the reset is triggered when the V_{DD} is lower than the BOR falling threshold.
- When BOR is disabled, the reset is released when the V_{DD} is higher than the POR threshold.

Figure 6-2 POR/PDR and BOR thresholds



6.2.3 Programmable voltage detector (PVD)

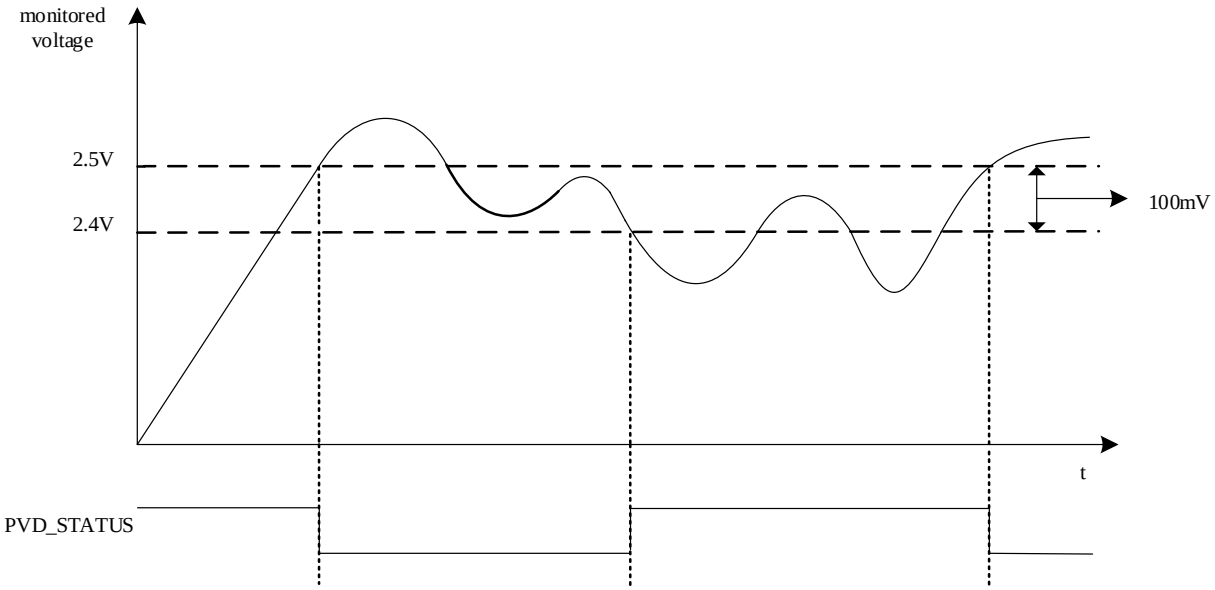
The PVD can be used to monitor the V_{DD} power supply by comparing it to the selected thresholds.

When a PVD alarm is triggered, which can generate an interrupt if enabled. The trigger conditions are as follows:

- The monitored voltage is higher than the threshold.
- The monitored voltage is lower than the threshold.
- The monitored voltage drops from higher than the threshold to lower than the threshold.
- The monitored voltage rises from lower than the threshold to higher than the threshold.

PVD thresholds have multiple levels that are determined by the PVD_LEVEL[2:0] bits in the *Power control register 2 (PMU_CR2)*. PVD has a hysteresis function, so when a PVD threshold level is selected, the lower threshold voltage is 100mV lower than the upper threshold voltage. See the following figure (PVD_LEVEL[2:0] is 010).

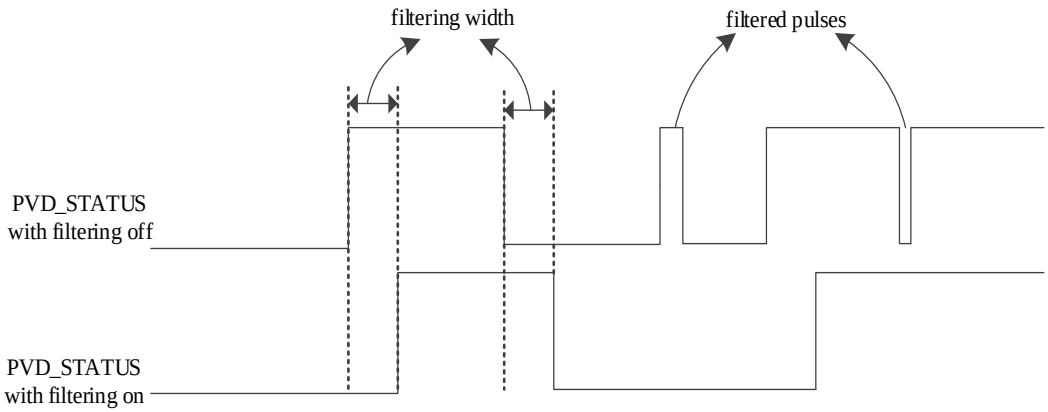
Figure 6-3 PVD threshold



PVD supports digital filtering, and the filter time is determined by the FILTER_TIME[2:0] bits in the *Power control register 2 (PMU_CR2)*. Digital filtering is enabled by setting the FILTER_EN bit in the PMU_CR2 register to 1. When filtering is enabled, PVD_STATUS reflects the filtered PVD state.

Since the filter time is based on the PCLK, in Stop mode, where PCLK has stopped, the digital filtering function cannot work. Therefore, the digital filtering function should be disabled before entering Stop mode. Otherwise, the PVD will not be able to wake up the system.

Figure 6-4 PVD filtering



When the PVD_LOCK bit in the *SYSCFG secure control register (SYSCFG_SECCR)* is set to 1, the PVD configuration is locked, and any changes to the PVD configuration are ignored. This means that write operations to the PMU_CR2 register are ignored until the next reset.

After PVD is enabled (PVD_EN is set to 1), the PVD threshold level, alarm trigger conditions, and filter configuration cannot be modified. After enabling PVD, the PVD output status becomes valid only after a minimum delay of 40μs. During this period, any alarm events should be ignored (clear PVD_INTF to 0).

6.2.4 Internal bandgap reference voltage

The internal bandgap reference voltage (BGR, typical voltage 1.2 V) has various applications, as follows:

- Provide a reference voltage for TS (Temperature Sensor) and VREFBUF
- Used as an internal input channel for ADC
- Used as the inverting input signal for COMP

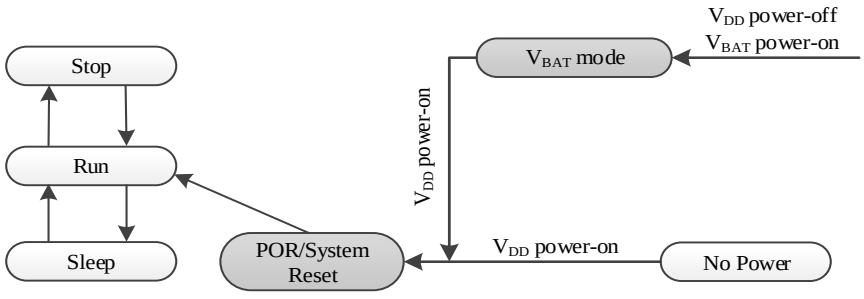
BGR is enabled by setting BGR_EN to 1. After enabling BGR, it is necessary to wait for BGR to stabilize (the t_{START} startup stabilization time is specified in the datasheet) before the associated functions can be used.

If BGR is not used in Stop mode, BGR should be disabled before entering low-power mode to reduce system power consumption. After exiting Stop mode, if BGR is needed, it must be re-enabled and allowed to stabilize.

6.3 Low-power modes

By default, the microcontroller is in Run mode after a system or a POR/PDR reset, with the system clock source being RCHSYS (16 MHz). Several low-power modes are available to save power when the CPU does not need to be kept running. Additionally, system power consumption in Run mode can be reduced by slowing down the system clocks and gating the clocks to the APB and AHB peripherals when they are unused.

Figure 6-5 Low-power modes state diagram



The device features 3 low-power modes. Differences between various power modes and available peripherals are shown in the table below.

Table 6-1 Low-power mode summary

Mode name	Mode description	Entry	Wakeup source	Wakeup system clock
Sleep	CPU clock OFF no effect on other clocks or analog clock sources	Set SLEEPDEEP to 0 WFI or Return from ISR	Any NVIC interrupt	Same as before entering Sleep mode
		SLEEPDEEP set to 0 WFE	Wakeup event	
Stop	All clocks OFF except RCL and LXTAL RCH, HXTAL, PLL, and Flash are power-off	Set SLEEPDEEP to 1 WFI or Return from ISR	Interrupts from peripherals connected to EXTI	RCHSYS (The RCH prescaler value is determined by the RCHDIV[1:0] bits)
		Set SLEEPDEEP to 1 WFE	Wakeup events from peripherals connected to EXTI	
V_{BAT}	All clocks OFF except LXTAL Only V_{CORE_AON} domain remains active	V_{DD} power-off V_{BAT} pin connected to backup power supply or battery	V_{DD} power-on	RCHSYS (16M)

Table 6-2 Functionalities depending on the working mode(1)

Peripheral	Run	Sleep	Stop		V_{BAT}
			functionality	wakeup capability	
CPU	√	×	×	×	×
Flash	√	√	×	×	×
SRAM	√	√	√	×	×
Backup Registers	√	√	√	×	√
BOR	○	○	○	○	×
POR/PDR	√	√	√	√	√
PVD	○	○	○	○	×
RCH	○	○	○(2)	×	×
RCL	○	○	○	×	×
PLL	○	○	×	×	×
HXTAL	○	○	×	×	×
LXTAL	○	○	○	×	○
HXTAL CSS	○	○	×	×	×
LXTAL CSS	○	○	○	○	×

Peripheral	Run	Sleep	Stop		V_{BAT}
			functionality	wakeup capability	
DMA	○	○	×	×	×
USART1	○	○	×	×	×
UART2/3/4	○	○	×	×	×
LPUART1/2	○	○	○ (3)	○ (3)	×
I2C1	○	○	○ (4)	○ (4)	×
SPI1/2	○	○	×	×	×
TIM3/4/5/8	○	○	×	×	×
LPTIM1/2	○	○	○	○	×
IWDG	○	○	○	○	×
WWDG	○	○	×	×	×
SysTick	○	○	×	×	×
ADC	○	○	×	×	×
VREFBUF	○	○	○	×	×
BGR	○	○	○	×	×
COMP1/2	○	○	○	○	×
TS	○	○	×	×	×
GPIO	○	○	○	○	×
LCD	○	○	○	×	×
TRNG	○	○	×	×	×
CRC	○	○	×	×	×
AES	○	○	×	×	×
TAMP	○	○	○	○	○
RTC	○	○	○	○	○

1. Legend: √-Enable, ×-Not available, ○-Optional.
2. RCH can be kept enabled in Stop mode by setting the RCH_AWON bit to 1 in the *Clock control register 1 (RCC_CSR1)*.
3. LPUART1/2 can wake up system in Stop mode. See details in *EXTI line connections*.
4. When an address matches, a wake-up event is generated by I2C1, which can wake up system in Stop mode.

6.3.1 Run mode

By default, the microcontroller is in Run mode after a system or a POR/PDR reset, with the system clock source being RCHSYS (16 MHz). The maximum supported clock frequency is 48 MHz, and all peripherals are available.

The device supports the low-power Run mode, where the system clock source can be selected as either RCL (32 KHz) or LXTAL (32.768 KHz).

To further reduce the power consumption in this mode, proceed as follows:

- Configure the prescaler to reduce the clock frequencies of SYSCLK, HCLK, PCLK1, and PCLK2.
- Disable the clocks of unused peripherals.
- Enter Sleep or Stop mode.

6.3.2 Sleep mode

When entering Sleep mode, only the CPU clock is disabled, and all peripherals can function normally. After exiting Sleep mode, the system enters Run mode, and the system clock source remains unchanged.

Table 6-3 Sleep mode summary

Characteristic	Description
Mode entry	1. Set SLEEPDEEP to 0 2. WFI
	1. Set SLEEPDEEP to 0 2. No wakeup event is pending 3. WFE
	1. Set SLEEPDEEP to 0 2. Set SLEEPONEXIT to 1 3. Return from the lowest priority ISR
Mode exit	If WFI or Return from ISR was used for entry: – Any interrupt (enabled)
	If WFE was used for entry and SEVONPEND = 0: – Any interrupt (enabled) – Wakeup events from EXTI
	If WFE was used for entry and SEVONPEND = 1: – Interrupt even when disabled in NVIC – Wakeup events from EXTI
Wakeup latency	None

6.3.3 Stop mode

The Stop mode is based on the Cortex-M0+ Deepsleep mode. In Stop mode, all clocks

in the V_{CORE} domain (include CPU and peripheral) are stopped. The PLL, RCH and the HXTAL are disabled. RCL and LXTAL can function normally based on the configuration.

Some peripherals with the wakeup capability (LPUART1/2 and I2C1) can switch on the RCH to detect wakeup event, and switch off the RCH after receiving the frame if it is not a wakeup frame.

In Stop mode, SRAM and register contents are preserved, and the Flash is in PowerDown state. If Flash memory programming or erasing is ongoing, the Stop mode entry is delayed until the memory access is finished.

Peripherals available in Stop mode are listed in the table: *Functionalities depending on the working mode*.

To further reduce the power consumption in this mode, proceed as follows:

- Enable ultra-low power configuration (set ULP_EN in the *Power control register 3 (PMU_CR3)*), which allows PDR/BOR to periodic sample supply voltage.
- In Stop mode, if BGR is not used, disable BGR before entering the low power mode to reduce system power consumption. After exiting Stop mode, enable BGR again if BGR is needed, and wait for it to stabilize (the t_{START} startup stabilization time is specified in the datasheet).
- Although ADC and TS (temperature sensor) cannot work in Stop mode, they still consume power. Therefore, these peripherals should be switched off before entering Stop mode.

After exiting Stop mode, the device enters Run mode, where the system clock source is RCHSYS, and the clock division is determined by the RCHDIV[1:0] bits.

Table 6-4 Stop mode summary

Characteristic	Description
Mode entry	1. Configure the RCHDIV[1:0] bits to select the system clock after exiting Stop mode 2. Set SLEEPDEEP to 1 3. WFI
	1. Configure the RCHDIV[1:0] bits to select the system clock after exiting Stop mode

	2. Set SLEEPDEEP to 1 3. No wakeup event is pending 4. WFE
	1. Configure the RCHDIV[1:0] bits to select the system clock after exiting Stop mode 2. Set SLEEPDEEP to 1 3. Set SLEEPONEXIT to 1 4. Return from the lowest priority ISR
Mode exit	If WFI or Return from ISR was used for entry: – Any EXTI line configured in interrupt mode (the corresponding interrupt vector must be enabled in the NVIC). See details in *EXTI line connections.*
	If WFE was used for entry and SEVONPEND = 0: – Any EXTI line configured in interrupt mode (the corresponding interrupt vector must be enabled in the NVIC). See details in *EXTI line connections.* – Events from EXTI
	If WFE was used for entry and SEVONPEND = 1: – Any EXTI line configured in interrupt mode (even if the corresponding EXTI interrupt vector is disabled in the NVIC). See details in *EXTI line connections.* – Events from EXTI
Wakeup latency	RCH wakeup time + Flash wakeup time

6.3.4 V_{BAT} mode

V_{BAT} mode enable and disable are determined by the VBAT_MODE_EN bit in the *Flash option register 2 (FLASH_OPTR2)*. When V_{BAT} mode is enabled, the V_{BAT} pin is connected to the backup power supply.

In V_{BAT} mode, all clock sources except LXTAL are disabled. In the V_{CORE_AON} domain, RTC, TAMP, LXTAL, backup registers, the RCC_AWCR register, and POR/PDR are available, while all other peripherals are powered off.

Only PC13, PC14, PC15 are available in V_{BAT} mode.

- RTC output pin: PC13
- TAMP external tamper detection pin: PC13
- LXTAL related pins: PC14, PC15

Entry and exit V_{BAT} mode

When V_{DD} is powered off and the V_{BAT} pin has a power supply, the system enters V_{BAT} mode. When V_{DD} power is restored, the system exits V_{BAT} mode.

6.4 PMU registers

The PMU registers can only be accessed by words (32-bit).

Table 6-5 PMU base address

Peripheral	Base address
PMU	0x4000 7000

6.4.1 Power control register 1 (PMU_CR1)

Address offset: 0x00

Reset value: 0x0000 0000

Note: The register can be reset by POR/PDR and system reset.

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

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

Bits	Name	Description
31:16	Reserved	Must be kept at reset value
15	BGR_EN	Bandgap reference enable 0: Disable 1: Enable
14:9	Reserved	Must be kept at reset value
8	VAON_WEN	V_{CORE_AON} domain write enable (backup registers, TAMP registers and RCC_AWCR register): 0: Write V_{CORE_AON} domain registers disabled 1: Write V_{CORE_AON} domain registers enabled
7:0	Reserved	Must be kept at reset value

6.4.2 Power control register 2 (PMU_CR2)

Address offset: 0x04

Reset value: 0x0000 0000

Note: The register can be reset by POR/PDR and system reset.

When PVD is enabled (PVD_EN is 1), the PVD_LEVEL, HT_EN, LT_EN, FT_EN, RT_EN, FILTER_EN, and FILTER_TIME cannot be modified. After the PVD_LOCK bit in the SYSCFG secure control register (SYSCFG_SECCR) is written with 1, the PVD configuration cannot be changed, and writing to this register will be ignored.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PVD_IE	HT_EN	LT_EN	FT_EN	RT_EN	FILTER_EN	FILTER_TIME[2:0]			Res.			PVD_LEVEL[2:0]			PVD_EN
rw	rw	rw	rw	rw	rw	rw	rw	rw				rw	rw	rw	rw

Bits	Name	Description
31:16	Reserved	Must be kept at reset value
15	PVD_IE	Programmable voltage detector interrupt enable 0: Disable 1: Enable
14	HT_EN	Enable alarm when detected voltage is higher than the threshold 0: Disable 1: Enable
13	LT_EN	Enable alarm when detected voltage is lower than the threshold 0: Disable 1: Enable
12	FT_EN	Enable alarm when detected voltage falls below the threshold 0: Disable 1: Enable

11	RT_EN	Enable alarm when detected voltage rises above the threshold 0: Disable 1: Enable
10	FILTER_EN	Programmable voltage detector digital filter enable Before entering Stop mode, the PVD digital filter should be disabled 0: Disable 1: Enable
9:7	FILTER_TIME[2:0]	Programmable voltage detector digital filter time 000: The filter period is 2 PCLK cycles 001: The filter period is 4 PCLK cycles 010: The filter period is 8 PCLK cycles 011: The filter period is 16 PCLK cycles 100: The filter period is 32 PCLK cycles 101: The filter period is 64 PCLK cycles 110: The filter period is 128 PCLK cycles 111: The filter period is 256 PCLK cycles
6:4	Reserved	Must be kept at reset value
3:1	PVD_LEVEL[2:0]	Programmable voltage detector threshold selection 000: Level0 rising/falling (2.1/2.0V) 001: Level1 rising/falling (2.3/2.2V) 010: Level2 rising/falling (2.5/2.4V) 011: Level3 rising/falling (2.6/2.5V) 100: Level4 rising/falling (2.7/2.6V) 101: Level5 rising/falling (2.9/2.8V) 110: Level6 rising/falling (3.0/2.9V) 111: Level7 rising/falling (3.1/3.0V)
0	PVD_EN	Programmable voltage detector enable 0: Disable 1: Enable

6.4.3 Power control register 3 (PMU_CR3)

Address offset: 0x08

Reset value: 0x0000 0000

Note: The register can be reset by POR/PDR and system reset.

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

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

Bits	Name	Description
31:10	Reserved	Must be kept at reset value
9	ULP_EN	Ultra-low-power enable When set, the supply voltage is sampled for PDR/BOR reset condition only periodically and not continuously in Stop mode, in order to save power. When exiting Stop, the sampling of supply voltage automatically becomes continuous. 0: Disable 1: Enable
8:0	Reserved	Must be kept at reset value

6.4.4 Power status register (PMU_SR)

Address offset: 0x14

Reset value: 0x0000 0000

Note: The register can be reset by POR/PDR and system reset.

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			PVD_ INTF	PVD_ STATUS	Res.										
			r	r											

Bits	Name	Description
31:13	Reserved	Must be kept at reset value
12	PVD_INTF	Programmable voltage detector interrupt flag When a PVD event occurs, PVD_INTF is set. PVD_INTF can be cleared by setting the PVD_INTF_CLR bit in the PMU_CLR register or by clearing the PVD_EN bit (disable PVD).
11	PVD_STATUS	Programmable voltage detector status (if the filter is enabled, PVD_STATUS indicates the filtered result) 0: Detected voltage is higher than threshold 1: Detected voltage is lower than threshold
10:0	Reserved	Must be kept at reset value

6.4.5 Power status clear register (PMU_CLR)

Address offset: 0x18

Reset value: 0x0000 0000

Note: The register can be reset by POR/PDR and system reset.

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

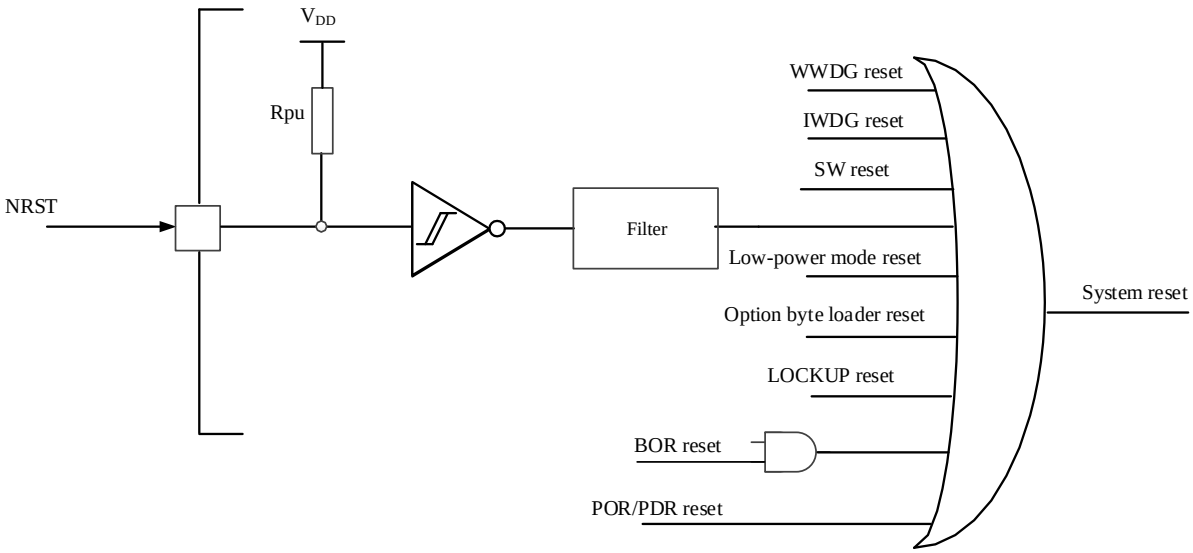
Bits	Name	Description
31:13	Reserved	Must be kept at reset value
12	PVD_INTF_CLR	Clear the PVD_INTF Writing 1 to the PVD_INTF_CLR bit will clear the PVD_INTF
11:0	Reserved	Must be kept at reset value

7 Reset and clock control (RCC)

7.1 Reset

There are three types of reset, defined as POR/PDR reset, system reset and V_{CORE_AON} domain reset.

Figure 7-1 Block diagram of the reset circuit



7.1.1 POR/PDR reset

- POR/PDR reset set all V_{CORE} domain registers to their reset values, including the option registers, IWDG registers, and DBG-related registers.
- POR/PDR reset also set certain bits of the RCC_AWCR register and the RTC-related registers to their reset values under the V_{CORE_AON} domain.

7.1.2 System reset

System reset resets the registers in the V_{CORE} domain (excluding the option registers in V_{CORE} domain, the RCC calibration registers, and the DBG-related registers), certain bits of RCC_CSR2 register, and the partial RTC registers in the V_{CORE_AON} domain.

System reset is generated when one of the following events occurs:

- Low level on the NRST pin
- Window watchdog event (WWDG reset)
- Independent watchdog event (IWDG reset)

- Software reset (SW reset)
- Low-power mode reset
- Option byte loader reset
- LOCKUP reset
- BOR reset

The reset source can be identified by checking the reset flags in the RCC_CSR2 register.

NRST pin

System reset is generated when the NRST pin detects a low level. The reset pin has a built-in pull-up resistor and an integrated glitch filter circuit. To ensure a reliable reset, the NRST pin low level hold time should be greater than 500μs. Refer to *Figure: Block diagram of the reset circuit*.

Software reset

The SYSRESETERQ bit in Cortex-M0+ Application interrupt and reset control register must be set to force a software reset on the device.

Low-power mode reset

To prevent that critical applications mistakenly enter a low-power mode, low-power mode reset is available. If enabled in option bytes, the reset is generated when entering Stop mode.

This type of reset is enabled by resetting RST_STOP bit in user option bytes. In this case, whenever a Stop mode entry sequence is successfully executed, the device is reset instead of entering Stop mode. Refer to *Flash option register 2 (FLASH_OPTR2).*

Option byte loader reset

The option byte loader reset is generated when the OBL_LAUNCH bit is set in the *Flash control register (FLASH_CR)*. This bit is used to launch the option byte loading by software.

For more information about user option bytes, refer to *Embedded Flash memory (Flash).*

BOR reset

BOR reset can be enabled or disabled through the BOR_EN bit in the *Flash option*

register 1 (FLASH_OPTR1). For detailed information on BOR reset, refer to *Brown-out reset*.

7.1.3 V_{CORE_AON} domain reset

A V_{CORE_AON} domain reset is generated when one of the following events occurs:

- Software reset, triggered by setting the AW_RST bit in the RCC_AWCR register, which resets RTC, TAMP (including the backup registers), and certain bits of the RCC_AWCR register.
- V_{DD}/V_{DDA} power on, if V_{DD}/V_{DDA} and V_{BAT} have previously been power off.

7.2 Clocks

The device provides the following clock sources producing primary clocks:

- RCH: a high-speed fully-integrated RC oscillator producing RCH clock (about 16 MHz)
- HXTAL: a high-speed oscillator with external crystal/ceramic resonator or external clock source, producing HXTAL clock (4 to 32 MHz)
- RCL: a low-speed fully-integrated RC oscillator producing RCL clock (about 32 KHz)
- LXTAL: a low-speed oscillator with external crystal/ceramic resonator or external clock source, producing LXTAL clock (accurate 32.768 KHz or external clock up to 1 MHz).

The RCHSYS (16 MHz) is used as system clock source after startup from reset.

The clocks of AHB, APB1 and APB2 can be configured with prescaler. The maximum frequency is 48 MHz.

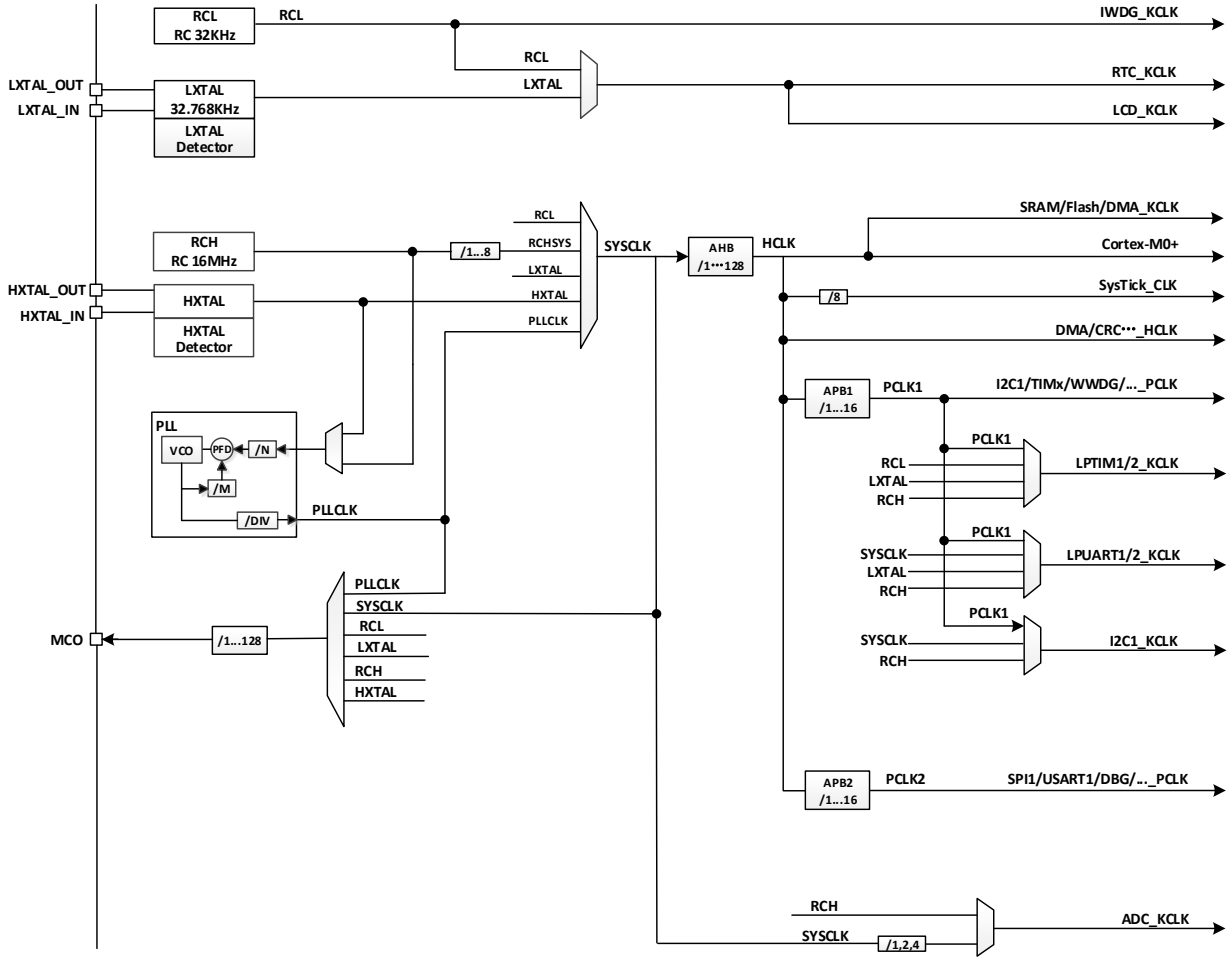
The following table shows the relationship of different clocks:

Table 7-1 Clock relationship

Output clock	Input clock	Description
RCHSYS	RCH	A clock derived from RCH through division by a factor programmable from 1 to 8
PLLCLK	RCH, HXTAL	A clock output from the PLL block
SYSCLK	LXTAL, RCL, HXTAL,	System clock, selected through the register

Output clock	Input clock	Description
	PLLCLK, RCHSYS	
HCLK	SYSCLK	A clock derived from SYSCLK through division by a factor programmable from 1 to 128
PCLK1	HCLK	A clock derived from HCLK through division by a factor programmable from 1 to 16
PCLK2	HCLK	A clock derived from HCLK through division by a factor programmable from 1 to 16

Figure 7-2 Clock tree



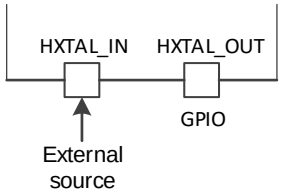
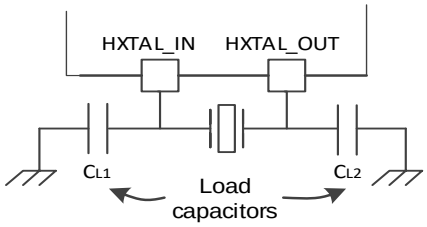
7.2.1 HXTAL clock

The high speed external clock signal (HXTAL) can be generated from two possible clock sources:

- External crystal/ceramic resonator
- User external clock (bypass mode)

The resonator and the load capacitors have to be placed as close as possible to the oscillator pins in order to minimize output distortion and startup stabilization time. The loading capacitance values must be adjusted according to the selected oscillator.

Table 7-2 HXTAL clock sources

Clock source	Hardware configuration
External clock	
Crystal/Ceramic resonators	

External crystal/ceramic resonator

The 4 to 32 MHz external oscillator has the advantage of producing a very accurate rate on the main clock.

The HXTAL can be switched on and off using the HXTALON bit in the RCC_CSR1 register.

The stabilization time of the HXTAL clock can be configured through the HXTAL_STAB_SEL[1:0] bits in the RCC_STABC register. These bits can only be modified when the HXTALON bit is 0.

The HXTALRDY flag in the RCC_CSR1 register indicates if the HXTAL oscillator is stable or not. An interrupt can be generated if enabled in the RCC_IER register.

To achieve the optimal balance between startup time and low power consumption, and improve compatibility with different crystals, the drive capability of the oscillator circuit can be adjusted using the HXTAL_DRV[1:0] bits in the RCC_CSR1 register.

HXTAL clock configuration is shown as below:

Table 7-3 HXTAL clock configuration

HXTALON	HXTALBYP	Status
0	x	HXTAL disabled
1	0	HXTAL enabled with external crystal mode
1	1	HXTAL enabled with bypass mode

External source (bypass mode)

In this mode, an external clock source must be provided. It can have a frequency of up

to 32 MHz.

This mode is selected by setting the HXTALON and HXTALBYP bits. The external clock signal (square, sinus or triangle) with 40-60% duty cycle depending on the frequency must drive the HXTAL_IN pin. The HXTAL_OUT pin can be used as a GPIO.

7.2.2 RCH clock

The RCH clock signal is generated from an internal 16 MHz RC oscillator, which can be used as the system clock or the reference clock of PLL.

RCH requires only 1.3 µs (typical) to stabilize from startup, with an accuracy of ±2% over the full voltage and temperature range, and does not require an external crystal resonator.

RCH can be enabled or disabled through the RCHON bit in the RCC_CSR1 register. Setting the RCH_AWON bit in the RCC_CSR1 register to 1 ensures that RCH remains enabled after entering Stop mode.

The RCHRDY flag in the RCC_CSR1 register indicates whether RCH is stable. If the interrupt is enabled in the RCC_IER register, an interrupt will be generated when RCH stabilizes.

RCH can be fine-tuned through the RCH_CAL[6:0] bits in the RCC_RCHCAL register. Calibration can be achieved using the capture function of TIM3. For more details, refer to *Internal/external clock measurement with TIM3*.

The factory calibration value for RCH is stored at address 0x1FFF03CC. For more details, refer to *Factory calibration parameter*.

7.2.3 PLL

The reference clock source for the internal PLL can be provided by either the RCH or HXTAL clock, with an input clock range of 4 MHz to 32 MHz. The output clock frequency range is 6 MHz to 48 MHz. For more details, refer to *Figure: Clock tree* and the RCC_PLLCFG register.

The PLL configuration (selection of the input clock, multiplication factor and division factor) must be done before enabling the PLL. Once the PLL is enabled, these parameters cannot be changed.

To modify the PLL configuration, proceed as follows:

1) Gate the PLL by setting PLLEN to 0 in the RCC_PLLCFG register.
2) Disable the PLL by setting PLLON to 0 in the RCC_CSR1 register.
3) Wait until PLLRDY is cleared. The PLL is now fully stopped.
4) Change the desired parameter.
5) Enable the PLL again by setting PLLON to 1.
6) Wait until PLLRDY is 1. The PLL is now ready.
7) Enable the PLL output by configuring PLLPEN in the RCC_PLLCFG register.

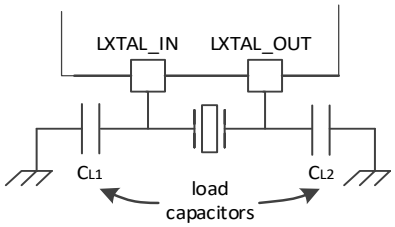
PLLEN cannot be cleared if PLLCLK is used as system clock.

7.2.4 LXTAL clock

The low speed external clock signal (LXTAL) can be generated from external crystal/ceramic resonator.

The resonator and the load capacitors have to be placed as close as possible to the oscillator pins in order to minimize output distortion and startup stabilization time. The loading capacitance values must be adjusted according to the selected oscillator.

Table 7-4 LXTAL clock sources

Clock source	Hardware configuration
Crystal/Ceramic resonators	

The frequency of LXTAL crystal is 32.768 KHz. It has the advantage of providing a low-power but highly accurate clock source to the real-time clock peripheral (RTC) for clock/calendar or other timing functions.

The LXTAL can be switched on and off using the LXTALON bit in the RCC_AWCR register.

The stabilization time of the LXTAL clock can be configured through the LXTAL_STAB_SEL[1:0] bits in the RCC_AWCR register. These bits can only be modified when the LXTALON bit is 0.

The LXTALRDY flag in the RCC_AWCR register indicates if the LXTAL oscillator

is stable or not. An interrupt can be generated if enabled in the RCC_IER register.

To achieve the optimal balance between startup time and low power consumption, and improve compatibility with different crystals, the drive capability of the oscillator circuit can be adjusted using the LXTAL_DRV_MODE and LXTAL_DRV[1:0] bits in the RCC_AWCR register.

LXTAL clock configuration is shown as below:

Table 7-5 LXTAL clock configuration

LXTALON	Status
0	LXTAL disabled
1	LXTAL enabled

7.2.5 RCL clock

The RCL acts as a low-power clock source that can be kept running in Stop mode for the IWDG and RTC. The clock frequency is 32 KHz.

RCL can be enabled in several ways:

- Enabled using RCLON bit in the RCC_CSR2 register
- Enabled when IWDG is enabled
- Enabled when RCL is used as the RTC clock source and RTC is enabled (RTC_EN bit in the RCC_AWCR register is set to 1)

The RCLRDY flag in the RCC_CSR2 register indicates if the RCL is stable or not. An interrupt can be generated if enabled in the RCC_IER register.

RCL can be tuned through the RCL_CAL[4:0] bits in the RCC_RCLCAL register. Calibration can be achieved using the capture function of TIM3. For more details, refer to *Internal/external clock measurement with TIM3*.

The factory calibration value for RCL is stored at address 0x1FFF03D0. For more details, refer to *Factory calibration parameter*.

7.2.6 System clock

One of the following clocks can be selected as system clock (SYSCLK):

- RCHSYS
- HXTAL

- PLLCLK
- RCL
- LXTAL

The system clock maximum frequency is 48 MHz. Upon system reset, the RCHSYS (16 MHz) clock derived from RCH oscillator is selected as system clock.

A switch from one clock source to another occurs only if the target clock source is ready. If a clock source which is not yet ready is selected, the switch occurs when the clock source becomes ready. SYSWS bits in the RCC_CFG register indicate which clock is ready and which clock is currently used as a system clock.

When a clock source is used directly or through the PLL as a system clock, it is not possible to stop it.

7.2.7 HXTAL clock security system (HXTAL CSS)

A clock security system on HXTAL can be enabled by setting the HXTAL_CSSON bit in the RCC_CSR1 register. After enabling, the CSS becomes active once the HXTAL is stable.

If a failure is detected on the HXTAL clock:

- The HXTAL CSS function is automatically disabled (HXTAL_CSSON cleared to 0).
- The HXTAL oscillator is automatically disabled (HXTALON cleared to 0).
- If HXTAL is selected directly or indirectly (PLLCLK selected for SYSCLK and HXTAL selected as PLL input) as system clock, the system clock switches automatically to RCHSYS (automatically enabled by hardware), and the division of RCH is determined by the RCHDIV[1:0] bits.
- The HXTAL_CSSF bit in the RCC_ISR register is set to 1, and an HXTAL CSS interrupt is generated. This interrupt is connected to the Cortex-M0+ NMI (Non-Maskable Interrupt). Therefore, the HXTAL_CSSC bit in the RCC_ICR register must be set to 1 to clear the HXTAL CSS interrupt in the NMI ISR. Before clearing HXTAL_CSSF to 0, the HXTAL clock cannot be re-enabled.
- If the HXTAL clock is the reference clock of the PLL, the PLL will be disabled too.

7.2.8 LXTAL clock security system (LXTAL CSS)

A clock security system on LXTAL can be enabled by setting the LXTAL_CSSON bit in the RCC_AWCR register. After enabling, the CSS can only be disabled by a V_{CORE_AON} domain reset or after the LXTAL failure is detected.

When LXTAL_CSSON is 1, the LXTAL CSS becomes active once the LXTALRDY bit, RCLON bit, and RCLRDY bit are set to 1.

The LXTAL CSS is available in all modes except V_{BAT}. The LXTAL CSS function is not affected during system reset.

If a failure is detected on the LXTAL clock:

- The LXTAL_CSSF bit is automatically set to 1 by hardware in the RCC_ISR register.
- The LXTAL_CSSD bit is automatically set to 1 by hardware in the RCC_AWCR register.
- The LXTAL oscillator is automatically disabled (LXTALON cleared to 0).
- A LXTAL CSS interrupt is generated which is connected to the Cortex-M0+ NMI (Non-Maskable Interrupt). Therefore, the LXTAL CSS must be disabled (clearing LXTAL_CSSON, the LXTAL_CSSD bit will be automatically cleared to 0) and LXTAL_CSSC bit in the RCC_ICR register must be set to 1 to clear the LXTAL CSS interrupt in the NMI ISR. Before clearing LXTAL_CSSD to 0, the LXTAL clock cannot be re-enabled.
- When LXTAL is used as the RTC clock source, LXTAL no longer provides the clock to the RTC, but the RTC registers are not affected. At this point, software can change the RTC clock source or take any appropriate action to secure the application.

If LXTAL is used as system clock, and a failure of LXTAL clock is detected, the system clock switches automatically to RCL.

When the LXTAL CSS is enabled, the LXTAL frequency must be greater than 30 KHz, in order to avoid false alarm.

In Stop mode, a LXTAL clock failure generates a wakeup.

7.2.9 Peripheral asynchronous clock selection

All peripheral clocks are provided by their respective bus clocks (HCLK, PCLK1, and

PCLK2), but some peripherals have their own kernel clocks configured independently.

Table 7-6 Peripheral asynchronous clock sources

module	asynchronous clock sources
IWDG	RCL
ADC	SYSCLK divided by 1, 2, or 4, RCH
LPUART1/2	PCLK1, LXTAL, RCH, SYSCLK
LPTIM1/2	PCLK1, LXTAL, RCH, RCL
I2C1	PCLK1, RCH, SYSCLK
RTC	RCL, LXTAL
LCD	RCL, LXTAL

7.2.10 IWDG clock

If the IWDG is started by software access, the RCL oscillator is forced ON and cannot be disabled. After the RCL oscillator temporization, the clock is provided to the IWDG.

7.2.11 ADC clock

The ADC asynchronous clock is derived from the system clock, or from the RCH. The system clock can be divided by 1, 2, or 4. The maximum frequency of ADC is 16 MHz. For more clock information refer to *Analog-to-digital converter (ADC): ADC clock* and *Peripherals independent clock configuration register (RCC_CLKSEL)*.

7.2.12 RTC and LCD clock

The RTC and LCD use the same clock source, which is selected through the RTCSEL[1:0] bits in the RCC_AWCR register:

- LXTAL external low speed clock
- RCL RC oscillator clock

After selecting the RTC/LCD clock source, the clock source can only be changed in the following two cases:

- V_{CORE_AON} domain reset
- LXTAL acts as the clock source and occurs the LXTAL CSS failure

When LXTAL is used as the RTC clock source, the RTC is available in Stop and V_{BAT} modes. When RCL is used as the RTC clock source, the RTC is available in Stop mode.

When LXTAL or RCL is used as the LCD clock source, the LCD is available in Stop

mode.

7.2.13 Clock-out capability

The microcontroller clock output (MCO) capability allows the clock to be output onto the external MCO pin. One of the following can be selected as the MCO clock:

- RCH
- RCL
- HXTAL
- LXTAL
- SYSCLK
- PLLCLK

The selection is controlled by the MCOSEL[2:0] bits of the RCC_CFG register. The selected clock can be divided with a factor programmable through the MCOPRE[2:0] field of the RCC_CFG register.

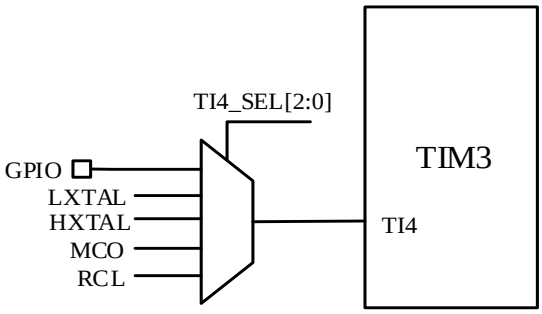
7.2.14 Internal/external clock measurement with TIM3

It is possible to indirectly measure the frequency of all on-board clock sources with the TIM3 channel 4 input capture. RCH clock can be input and measured through the MCO. The measurement accuracy can be improved by configuring the capture prescaler of TIM3. The larger the prescaler value, the higher the measurement accuracy.

The different input capture sources can be selected by configuring the TI4_SEL[2:0] bits in the *TIM3_TISEL register*.

- GPIO (TIM3_CH4)
- LXTAL
- HXTAL
- MCO
- RCL

Figure 7-3 Frequency measurement with TIM3 in capture mode



7.3 Low-power modes

- AHB and APB peripheral clocks can be disabled by software.
- Sleep mode stops the CPU clock, while all other clocks remain the same as in Run mode.
- Stop mode stops all the clocks in the V_{CORE} domain and disables the PLL, RCH and HXTAL oscillator. When RCH_AWON is configured as 1, the RCH clock remains enabled even when the device is in Stop mode.
- The LPUART1/2 peripherals can also operate with the clock from the RCH or LXTAL oscillator (LXTALON set) when the system is in Stop mode.
- The I2C1 peripheral can also operate with the clock from the RCH when the system is in Stop mode.

The Stop mode can be overridden for debugging by setting the DBG_STOP bit in the *DBG configuration register (DBG_CR)* before entering Stop mode to provide a clock to the debug interface.

7.4 RCC registers

The RCC registers can only be accessed by words (32-bit).

Table 7-7 RCC base address

Peripheral	Base address
RCC	0x4002 1000

7.4.1 Clock control register 1 (RCC_CSR1)

Address offset: 0x00

Reset value: 0x0010 0500

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.						PLL RDY	PLL ON	Res.		HXTAL_DRV [1:0]		HXTAL_ CSSON	HXTAL RDY	HXTAL BYP	HXTAL ON
						r	rw			rw	rw	rs	r	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			RCHDIV[1:0]		RCH RDY	RCH_ AWON	RCH ON	Res.							
			rw	rw	r	rw	rw								

Bits	Name	Description
31:26	Reserved	Must be kept at reset value
25	PLLRDY	PLL clock ready flag When PLLON is set to 0, PLLRDY will be automatically cleared. 0: PLL is not ready 1: PLL is ready
24	PLLON	PLL enable Cleared by hardware when entering Stop mode. This bit cannot be reset if the PLL clock is used as the system clock. 0: Disable 1: Enable
23:22	Reserved	Must be kept at reset value
21:20	HXTAL_DRV[1:0]	HXTAL oscillator drive capability 00: Low driving capability 01: Medium-low driving capability

		10: Medium-high driving capability 11: High driving capability
19	HXTAL_CSSON	HXTAL CSS enable Set by software to enable the clock security system. When HXTAL_CSSON is set, the clock detector is enabled by hardware when the HXTAL oscillator is ready, and disabled by hardware if a HXTAL clock failure is detected. This bit is set only and is cleared by reset. 0: Disable 1: Enable
18	HXTALRDY	HXTAL clock ready flag When HXTALON is set to 0, HXTAL_RDY will be automatically cleared. 0: HXTAL oscillator is not ready 1: HXTAL oscillator is ready
17	HXTALBYP	HXTAL crystal oscillator bypass The HXTALBYP bit can be written only if the HXTALON and HXTALRDY are both 0. 0: HXTAL crystal oscillator is not bypassed 1: HXTAL crystal oscillator is bypassed with external clock
16	HXTALON	HXTAL clock enable Cleared by hardware to stop the HXTAL oscillator when entering Stop mode. This bit cannot be reset if the HXTAL oscillator is used directly or indirectly as the system clock. 0: Disable 1: Enable
15:13	Reserved	Must be kept at reset value
12:11	RCHDIV[1:0]	RCH clock division factor This bit field controlled by software sets division factor of the RCH clock divider to produce RCHSYS clock. 00: 1

		01: 2 10: 4 11: 8
10	RCHRDY	RCH clock ready flag When RCHON is set to 0, RCH_RDY will be automatically cleared. 0: RCH oscillator is not ready 1: RCH oscillator is ready
9	RCH_AWON	RCH always enable for peripheral kernels Set and cleared by software to force RCH ON even in Stop mode. The RCH can only feed LPUART1/2 and I2C1 peripherals configured with RCH as kernel clock. Keeping the RCH ON in Stop mode allows avoiding to slow down the communication speed because of the RCH startup time. This bit has no effect on RCHON value. 0: No effect on RCH oscillator 1: RCH oscillator is forced ON even in Stop mode.
8	RCHON	RCH clock enable Cleared by hardware to stop the RCH oscillator when entering Stop mode. 0: Disable 1: Enable
7:0	Reserved	Must be kept at reset value

7.4.2 Clock stable configuration register (RCC_STABC)

Address offset: 0x04

Reset value: 0x0000 0003

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

Bits	Name	Description
31:2	Reserved	Must be kept at reset value
1:0	HXTAL_STAB_SEL[1:0]	Number of cycles for HXTAL clock startup stabilization 00: 256 01: 1024 10: 4096 11: 16384 *Note: The bit can be modified only when the HXTALON bit is set to 0.*

7.4.3 Clock configuration register (RCC_CFG)

Address offset: 0x08

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	MCOPRE[2:0]			Res.	MCOSEL[2:0]			Res.					P2PRE[2:0]		
	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.	P1PRE[2:0]			Res.	HPRE[2:0]			Res.		SYSWS[2:0]			SYSW[2:0]		
	rw	rw	rw		rw	rw	rw			r	r	r	rw	rw	rw

Bits	Name	Description
31	Reserved	Must be kept at reset value
30:28	MCOPRE[2:0]	Microcontroller clock output prescaler 000: 1 001: 2 010: 4 011: 8 100: 16 101: 32 110: 64 111: 128
27	Reserved	Must be kept at reset value

26:24	MCOSEL[2:0]	Microcontroller clock output 000: MCO output disabled 001: SYSCLK 010: Reserved (MCO output disabled) 011: RCH 100: HXTAL 101: PLLCLK 110: RCL 111: LXTAL
23:19	Reserved	Must be kept at reset value
18:16	P2PRE[2:0]	APB2 prescaler To produce PCLK2 clock, it sets the division factor of HCLK clock as follows: 0xx: 1 100: 2 101: 4 110: 8 111: 16
15	Reserved	Must be kept at reset value
14:12	P1PRE[2:0]	APB1 prescaler To produce PCLK1 clock, it sets the division factor of HCLK clock as follows: 0xx: 1 100: 2 101: 4 110: 8 111: 16 *Caution: When using APB1 peripherals to wake up device in Stop mode, the division factor should not be greater than 2.*
11	Reserved	Must be kept at reset value
10:8	HPRE[2:0]	AHB prescaler

		To produce HCLK clock, it sets the division factor of SYSCLK clock as follows: 000: 1 001: 2 010: 4 011: 8 100: 16 101: 32 110: 64 111: 128
7:6	Reserved	Must be kept at reset value
5:3	SYSWS[2:0]	System clock switch status 000: RCHSYS 001: HXTAL 010: PLLCLK 011: RCL 100: LXTAL
2:0	SYSW[2:0]	System clock switch 000: RCHSYS 001: HXTAL 010: PLLCLK 011: RCL 100: LXTAL Others: Reserved (RCHSYS)

7.4.4 PLL configuration register (RCC_PLLCFG)

Address offset: 0x0C

Reset value: 0x0002 0260

Note: This register is used to configure the PLL clock outputs according to the formulas:

$f_{VCO} = f_{PLLIN} \times (M/N)$

$f_{PLLCLK} = f_{VCO}/DIV$

where f_{PLLIN} is PLL input clock frequency, f_{VCO} is PLL VCO frequency, f_{PLLCLK} is

PLLCLK output frequency.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.													PLLDIV[1:0]		PLL EN
													rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					PLLN[2:0]			PLLM[4:0]					Res.		PLL SRC
					rw	rw	rw	rw	rw	rw	rw	rw			rw

Bits	Name	Description
31:19	Reserved	Must be kept at reset value
18:17	PLLDIV[1:0]	PLL VCO division factor 00: 1 01: 2 10: 4 11: 8 *Note: The bit field can be written only when the PLL is disabled.*
16	PLLEN	PLLCLK clock output enable When entering Stop mode, PLLEN is automatically cleared by hardware. 0: Disable 1: Enable
15:11	Reserved	Must be kept at reset value
10:8	PLLN[2:0]	Division factor N of the PLL input clock 000: 8 001: 1 010: 2 011: 3 100: 4 101: 5 110: 6 111: 7 *Caution: The bit field can be written only when the PLL is disabled.* *The software must set these bits so that the PLL input*

frequency after the /N divider is between 4 and 8 MHz.

7:3	PLLM[4:0]	PLL frequency multiplication factor M 00000: 32 00001: 1 00010: 2 00011: 3 ⋮ 11110: 30 11111: 31 *Caution: The bit field can be written only when the PLL is disabled. The software must set these bits so that the PLL output frequency is between 50 and 100 MHz.*
2:1	Reserved	Must be kept at reset value
0	PLLSRC	PLL input clock source 0: RCH 1: HXTAL *Caution: The bit field can be written only when the PLL is disabled. The PLL input clock frequency is between 4 and 32 MHz.*

7.4.5 Clock interrupt enable register (RCC_IER)

Address offset: 0x10

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.										PLL_RDYIE	HXTAL_RDYIE	RCH_RDYIE	Res.	LXTAL_RDYIE	RCL_RDYIE
										rw	rw	rw		rw	rw

Bits	Name	Description
31:6	Reserved	Must be kept at reset value
5	PLL_RDYIE	PLL ready interrupt enable

		0: Disable 1: Enable
4	HXTAL_RDYIE	HXTAL ready interrupt enable 0: Disable 1: Enable
3	RCH_RDYIE	RCH ready interrupt enable 0: Disable 1: Enable
2	Reserved	Must be kept at reset value
1	LXTAL_RDYIE	LXTAL ready interrupt enable 0: Disable 1: Enable
0	RCL_RDYIE	RCL ready interrupt enable 0: Disable 1: Enable

7.4.6 Clock interrupt flag register (RCC_ISR)

Address offset: 0x14

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.						LXTAL_CSSF	HXTAL_CSSF	Res.		PLL_RDYF	HXTAL_RDYF	RCH_RDYF	Res.	LXTAL_RDYF	RCL_RDYF
						r	r			r	r	r		r	r

Bits	Name	Description
31:10	Reserved	Must be kept at reset value
9	LXTAL_CSSF	LXTAL CSS interrupt flag Set by hardware when a failure is detected in the LXTAL oscillator.

		Cleared by software by setting the LXTAL_CSSC bit. 0: No LXTAL CSS interrupt 1: LXTAL CSS interrupt
8	HXTAL_CSSF	HXTAL CSS interrupt flag Set by hardware when a failure is detected in the HXTAL oscillator. Cleared by software by setting the HXTAL_CSSC bit. 0: No HXTAL CSS interrupt 1: HXTAL CSS interrupt
7:6	Reserved	Must be kept at reset value
5	PLL_RDYF	PLL ready interrupt flag Set by hardware when the PLL is ready and PLL_RDYIE is set. Cleared by software setting the PLL_RDYC bit. 0: No PLL clock ready interrupt 1: PLL clock ready interrupt
4	HXTAL_RDYF	HXTAL ready interrupt flag Set by hardware when the HXTAL is ready and HXTAL_RDYIE is set. Cleared by software setting the HXTAL_RDYC bit. 0: No HXTAL clock ready interrupt 1: HXTAL clock ready interrupt
3	RCH_RDYF	RCH ready interrupt flag Set by hardware when the RCH is ready and RCH_RDYIE is set. Cleared by software setting the RCH_RDYC bit. 0: No RCH clock ready interrupt 1: RCH clock ready interrupt
2	Reserved	Must be kept at reset value
1	LXTAL_RDYF	LXTAL ready interrupt flag Set by hardware when the LXTAL is ready and LXTAL_RDYIE is set. Cleared by software setting the LXTAL_RDYC bit. 0: No LXTAL clock ready interrupt 1: LXTAL clock ready interrupt

0	RCL_RDYF	RCL ready interrupt flag Set by hardware when the RCL is ready and RCL_RDYIE is set. Cleared by software setting the RCL_RDYC bit. 0: No RCL clock ready interrupt 1: RCL clock ready interrupt

7.4.7 Clock interrupt clear register (RCC_ICR)

Address offset: 0x18

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.						LXTAL_CSSC	HXTAL_CSSC	Res.		PLL_RDYC	HXTAL_RDYC	RCH_RDYC	Res.	LXTAL_RDYC	RCL_RDYC
						w	w			w	w	w		w	w

Bits	Name	Description
31:10	Reserved	Must be kept at reset value
9	LXTAL_CSSC	LXTAL CSS interrupt clear 0: No effect 1: Clear LXTAL_CSSF flag
8	HXTAL_CSSC	HXTAL CSS interrupt clear 0: No effect 1: Clear HXTAL_CSSF flag
7:6	Reserved	Must be kept at reset value
5	PLL_RDYC	PLL ready interrupt clear 0: No effect 1: Clear PLL_RDYF flag
4	HXTAL_RDYC	HXTAL ready interrupt clear 0: No effect

		1: Clear HXTAL_RDYF flag
3	RCH_RDYC	RCH ready interrupt clear 0: No effect 1: Clear RCH_RDYF flag
2	Reserved	Must be kept at reset value
1	LXTAL_RDYC	LXTAL ready interrupt clear 0: No effect 1: Clear LXTAL_RDYF flag
0	RCL_RDYC	RCL ready interrupt clear 0: No effect 1: Clear RCL_RDYF flag

7.4.8 I/O port reset register (RCC_IOPRST)

Address offset: 0x1C

Reset value: 0x0000 0000

Note: The software reset flow for the I/O port is as follows: Set the corresponding bit to 1 to reset the GPIO port, clear the bit to return the GPIO port to its normal mode.

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.										GPIOF_RST	Res.	GPIOD_RST	GPIOC_RST	GPIOB_RST	GPIOA_RST
										rw		rw	rw	rw	rw

Bits	Name	Description
31:6	Reserved	Must be kept at reset value
5	GPIOF_RST	GPIOF reset 0: No effect 1: Reset GPIOF
4	Reserved	Must be kept at reset value

3	GPIOD_RST	GPIOD reset 0: No effect 1: Reset GPIOD
2	GPIOC_RST	GPIOC reset 0: No effect 1: Reset GPIOC
1	GPIOB_RST	GPIOB reset 0: No effect 1: Reset GPIOB
0	GPIOA_RST	GPIOA reset 0: No effect 1: Reset GPIOA

7.4.9 AHB peripheral reset register (RCC_AHBRST)

Address offset: 0x20

Reset value: 0x0000 0000

Note: The software reset flow for the peripherals are as follows: Set the corresponding bit to 1 to reset the peripheral, clear the bit to return the peripheral to its normal mode.

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

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

Bits	Name	Description
31:17	Reserved	Must be kept at reset value
16	AES_RST	AES reset 0: No effect 1: Reset AES

15:13	Reserved	Must be kept at reset value
12	CRC_RST	CRC reset 0: No effect 1: Reset CRC
11:1	Reserved	Must be kept at reset value
0	DMA_RST	DMA reset 0: No effect 1: Reset DMA

7.4.10 APB1 peripheral reset register (RCC_APB1RST)

Address offset: 0x24

Reset value: 0x0000 0000

Note: The software reset flow for the peripherals are as follows: Set the corresponding bit to 1 to reset the peripheral, clear the bit to return the peripheral to its normal mode.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPTIM1_RST	LPTIM2_RST	Res.						LPUART2_RST	Res.	I2C1_RST	LPUART1_RST	UART4_RST.	UART3_RST	UART2_RST	Res.
rw	rw							rw		rw	rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SPI2_RST	Res.					LCD_RST	Res.	TIM8_RST	Res.		TIM5_RST	TIM4_RST	TIM3_RST	Res.
	rw						rw		rw			rw	rw	rw	

Bits	Name	Description
31	LPTIM1_RST	LPTIM1 reset 0: No effect 1: Reset LPTIM1
30	LPTIM2_RST	LPTIM2 reset 0: No effect 1: Reset LPTIM2
29:24	Reserved	Must be kept at reset value
23	LPUART2_RST	LPUAR2 reset

		0: No effect 1: Reset LPUART2
22	Reserved	Must be kept at reset value
21	I2C1_RST	I2C1 reset 0: No effect 1: Reset I2C1
20	LPUART1_RST	LPUAR1 reset 0: No effect 1: Reset LPUART1
19	UART4_RST	UART4 reset 0: No effect 1: Reset UART4
18	UART3_RST	UART3 reset 0: No effect 1: Reset UART3
17	UART2_RST	UART2 reset 0: No effect 1: Reset UART2
16:15	Reserved	Must be kept at reset value
14	SPI2_RST	SPI2 reset 0: No effect 1: Reset SPI2
13:9	Reserved	Must be kept at reset value
8	LCD_RST	LCD reset 0: No effect 1: Reset LCD

7	Reserved	Must be kept at reset value
6	TIM8_RST	TIM8 reset 0: No effect 1: Reset TIM8
5:4	Reserved	Must be kept at reset value
3	TIM5_RST	TIM5 reset 0: No effect 1: Reset TIM5
2	TIM4_RST	TIM4 reset 0: No effect 1: Reset TIM4
1	TIM3_RST	TIM3 reset 0: No effect 1: Reset TIM3
0	Reserved	Must be kept at reset value

7.4.11 APB2 peripheral reset register (RCC_APB2RST)

Address offset: 0x28

Reset value: 0x0000 0000

Note: The software reset flow for the peripherals are as follows: Set the corresponding bit to 1 to reset the peripheral, clear the bit to return the peripheral to its normal mode.

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	USART1_RST	Res.	SPI1_RST	Res.							TRNG_RST	Res.			SYSCFG_RST
	rw		rw								rw				rw

Bits	Name	Description
31:28	Reserved	Must be kept at reset value

27	DBG_RST	DBG reset 0: No effect 1: Reset DBG
26:21	Reserved	Must be kept at reset value
20	ADC_RST	ADC reset 0: No effect 1: Reset ADC
19:15	Reserved	Must be kept at reset value
14	USART1_RST	USART1 reset 0: No effect 1: Reset USART1
13	Reserved	Must be kept at reset value
12	SPI1_RST	SPI1 reset 0: No effect 1: Reset SPI1
11:5	Reserved	Must be kept at reset value
4	TRNG_RST	TRNG reset 0: No effect 1: Reset TRNG
3:1	Reserved	Must be kept at reset value
0	SYSCFG_RST	SYSCFG, COMP, and VREFBUF reset 0: No effect 1: Reset SYSCFG, COMP, and VREFBUF *Note: SYSCFG_SECCR register cannot be reset by SYSCFG_RST bit.*

7.4.12 I/O port clock enable register (RCC_IOPEN)

Address offset: 0x2C

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.										GPIOFEN	Res.	GPIODEN	GPIOCEN	GPIOBEN	GPIOAEN
										rw		rw	rw	rw	rw

Bits	Name	Description
31:6	Reserved	Must be kept at reset value
5	GPIOFEN	GPIOF clock enable 0: Disable 1: Enable
4	Reserved	Must be kept at reset value
3	GPIODEN	GPIOD clock enable 0: Disable 1: Enable
2	GPIOCEN	GPIOC clock enable 0: Disable 1: Enable
1	GPIOBEN	GPIOB clock enable 0: Disable 1: Enable
0	GPIOAEN	GPIOA clock enable 0: Disable 1: Enable

7.4.13 AHB peripheral clock enable register (RCC_AHBEN)

Address offset: 0x30

Reset value: 0x0000 0100

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

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

Bits	Name	Description
31:17	Reserved	Must be kept at reset value
16	AESEN	AES clock enable 0: Disable 1: Enable
15:13	Reserved	Must be kept at reset value
12	CRCEN	CRC clock enable 0: Disable 1: Enable
11:9	Reserved	Must be kept at reset value
8	FLASHEN	Flash registers clock enable This bit is only used to control the access clock for the Flash registers. The Flash memory clock remains always on. 0: Disable 1: Enable *Note: When the Flash is busy, this bit cannot be cleared.*
7:1	Reserved	Must be kept at reset value
0	DMAEN	DMA clock enable 0: Disable

1: Enable

7.4.14 APB1 peripheral clock enable register (RCC_APB1EN)

Address offset: 0x34

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPTIM1 EN	LPTIM2 EN	Res.	PMU EN	Res.				LPUART2 EN	Res.	I2C1 EN	LPUART1 EN	UART4 EN	UART3 EN	UART2 EN	Res.
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.	SPI2 EN	Res.		WWDG EN	RTCAPB EN	Res.	LCDEN	Res.	TIM8 EN	Res.		TIM5 EN	TIM4 EN	TIM3 EN	Res.
	rw			rw	rw		rw		rw			rw	rw	rw	

Bits	Name	Description
31	LPTIM1EN	LPTIM1 clock enable 0: Disable 1: Enable
30	LPTIM2EN	LPTIM2 clock enable 0: Disable 1: Enable
29	Reserved	Must be kept at reset value
28	PMUEN	PMU clock enable 0: Disable 1: Enable
27:24	Reserved	Must be kept at reset value
23	LPUART2EN	LPUART2 clock enable 0: Disable 1: Enable
22	Reserved	Must be kept at reset value
21	I2C1EN	I2C1 clock enable

		0: Disable 1: Enable
20	LPUART1EN	LPUART1 clock enable 0: Disable 1: Enable
19	UART4EN	UART4 clock enable 0: Disable 1: Enable
18	UART3EN	UART3 clock enable 0: Disable 1: Enable
17	UART2EN	UART2 clock enable 0: Disable 1: Enable
16:15	Reserved	Must be kept at reset value
14	SPI2EN	SPI2 clock enable 0: Disable 1: Enable
13:12	Reserved	Must be kept at reset value
11	WWDGEN	WWDG clock enable Set by software to enable the window watchdog clock. Cleared by hardware system reset. 0: Disable 1: Enable
10	RTCAPBEN	RTC APB clock enable 0: Disable 1: Enable

9	Reserved	Must be kept at reset value
8	LCDEN	LCD clock enable 0: Disable 1: Enable
7	Reserved	Must be kept at reset value
6	TIM8EN	TIM8 clock enable 0: Disable 1: Enable
5:4	Reserved	Must be kept at reset value
3	TIM5EN	TIM5 clock enable 0: Disable 1: Enable
2	TIM4EN	TIM4 clock enable 0: Disable 1: Enable
1	TIM3EN	TIM3 clock enable 0: Disable 1: Enable
0	Reserved	Must be kept at reset value

7.4.15 APB2 peripheral clock enable register (RCC_APB2EN)

Address offset: 0x38

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	USART1 EN	Res.	SPI1 EN	Res.							TRNG EN	Res.			SYSCFG EN

	rw		rw								rw				rw

Bits	Name	Description
31:28	Reserved	Must be kept at reset value
27	DBGEN	DBG clock enable 0: Disable 1: Enable
26:21	Reserved	Must be kept at reset value
20	ADCEN	ADC clock enable 0: Disable 1: Enable
19:15	Reserved	Must be kept at reset value
14	USART1EN	USART1 clock enable 0: Disable 1: Enable
13	Reserved	Must be kept at reset value
12	SPI1EN	SPI1 clock enable 0: Disable 1: Enable
11:5	Reserved	Must be kept at reset value
4	TRNGEN	TRNG clock enable 0: Disable 1: Enable
3:1	Reserved	Must be kept at reset value
0	SYSCFGEN	SYSCFG, COMP, and VREFBUF clock enable 0: Disable

1: Enable

7.4.16 Peripherals independent clock configuration register (RCC_CLKSEL)

Address offset: 0x3C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADC_SEL [1:0]		Res.								LPTIM2_SEL [1:0]		LPTIM1_SEL [1:0]		Res.	
rw	rw									rw	rw	rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		I2C1_SEL [1:0]		LPUART1_SEL [1:0]		LPUART2_SEL [1:0]		Res.							
		rw	rw	rw	rw	rw	rw								

Bits	Name	Description
31:30	ADC_SEL[1:0]	ADC clock source selection ADC clock frequency should not greater than 16 MHz. 00: SYSCLK 01: SYSCLK divided by 2 10: SYSCLK divided by 4 11: RCH
29:22	Reserved	Must be kept at reset value
21:20	LPTIM2_SEL[1:0]	LPTIM2 clock source selection 00: PCLK 01: RCL 10: RCH 11: LXTAL
19:18	LPTIM1_SEL[1:0]	LPTIM1 clock source selection 00: PCLK 01: RCL 10: RCH 11: LXTAL
17:14	Reserved	Must be kept at reset value

13:12	I2C1_SEL[1:0]	I2C1 clock source selection 00: PCLK 01: SYSCLK 10: RCH 11: Reserved (PCLK)
11:10	LPUART1_SEL[1:0]	LPUART1 clock source selection 00: PCLK 01: SYSCLK 10: RCH 11: LXTAL
9:8	LPUART2_SEL[1:0]	LPUART2 clock source selection 00: PCLK 01: SYSCLK 10: RCH 11: LXTAL
7:0	Reserved	Must be kept at reset value

7.4.17 V_{CORE_AON} domain control register (RCC_AWCR)

Address offset: 0x40

Reset value: 0x0020 0008

Note: This register is in the V_{CORE_AON} domain. POR/PDR can reset bits 5, 6, and 16. V_{CORE_AON} domain reset can reset bits 0, 1, 2, 3, 4, 5, 6, 8, 9, 15, 20, and 21.

After reset, these bits are write-protected and the VAON_WEN bit in the Power control register 1 (PMU_CR1) has to be set before these can be modified.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.										LXTAL_STAB_SEL[1:0]		Res.			AW_RST
										rw	rw				rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTC EN	Res.					RTCSEL[1:0]		Res.	LXTAL_CSSD	LXTAL_CSSON	LXTAL_DRV[1:0]		LXTAL RDY	LXTAL DRV_MODE	LXTAL ON
rw						rw	rw		r	rw	rw	rw	r	rw	rw

Bits	Name	Description

31:22	Reserved	Must be kept at reset value
21:20	LXTAL_STAB_SEL[1:0]	Number of cycles for LXTAL clock startup stabilization 00: 256 01: 1024 10: 4096 11: 16384 *Note: The bit can be modified only when the LXTALON bit is set to 0.*
19:17	Reserved	Must be kept at reset value
16	AW_RST	V_{CORE_AON} domain software reset 0: No effect 1: Reset V_{CORE_AON} domain
15	RTCEN	RTC kernel clock enable 0: Disable 1: Enable
14:10	Reserved	Must be kept at reset value
9:8	RTCSEL[1:0]	RTC/LCD clock source selection 00: No clock 01: LXTAL 1x: RCL *Note: Once the RTC clock source is selected, it cannot be changed anymore unless the following two cases occur, which will automatically clear this bit field and allow the clock source to be reconfigured:* – *V_{CORE_AON} domain software reset* – *LXTAL clock is selected as the clock source and a LXTAL CSS failure is detected* *For more information about clock configuration of RTC and LCD, refer to RTC and LCD clock.*
7	Reserved	Must be kept at reset value

6	LXTAL_CSSD	LXTAL CSS failure detection 0: No failure detected 1: Failure detected *Note: If LXTAL_CSSON bit is cleared, this bit will be automatically cleared.*
5	LXTAL_CSSON	LXTAL CSS enable 0: Disable 1: Enable *Note: Once enabled, this bit cannot be disabled, except after a LXTAL failure detection (LXTAL_CSSD =1). In that case the software must disable the LXTAL_CSSON bit.*
4:3	LXTAL_DRV[1:0]	LXTAL oscillator drive capability 00: Low driving capability 01: Medium-low driving capability 10: Medium-high driving capability 11: High driving capability *Note: To improve crystal compatibility, it is recommended to configure the settings to enhanced mode (LXTAL_DRV_MODE = 1) and medium-high driving capability (LXTAL_DRV[1:0] = 10).*
2	LXTALRDY	LXTAL oscillator ready When LXTALON is set to 0, LXTALRDY will be automatically cleared. 0: LXTAL oscillator is not ready 1: LXTAL oscillator is ready
1	LXTAL_DRV_MODE	LXTAL oscillator drive mode 0: Normal mode 1: Enhanced mode *Note: To improve crystal compatibility, it is recommended to configure the settings to enhanced mode (LXTAL_DRV_MODE = 1) and medium-high driving capability (LXTAL_DRV[1:0] = 10).*

0	LXTALON	LXTAL enable 0: Disable 1: Enable

7.4.18 Clock control register 2 (RCC_CSR2)

Address offset: 0x44

Reset value: 0xXX00 0000

Note: This register is only reset by POR/PDR reset. Bits 8 and 16 can also be reset by system reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPM _RSTF	WWDG _RSTF	IWDG _RSTF	SW_ RSTF	PMU_ RSTF	NRST _ RSTF	OBL_ RSTF	LOCKUP _RSTF	Res.							RMVF
r	r	r	r	r	r	r	r								rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.							LOCKUP_ RSTEN	Res						RCL RDY	RCL ON
							rw							r	rw

Bits	Name	Description
31	LPM_RSTF	Low-power reset flag Set by hardware when a reset occurs due to illegal Stop mode entry. Cleared by setting the RMVF bit. 0: No illegal mode reset occurred 1: Illegal mode reset occurred
30	WWDG_RSTF	Window watchdog reset flag Set by hardware when a window watchdog reset occurs. Cleared by setting the RMVF bit. 0: No window watchdog reset occurred 1: Window watchdog reset occurred
29	IWDG_RSTF	Independent window watchdog reset flag Set by hardware when an independent watchdog reset domain occurs. Cleared by setting the RMVF bit. 0: No independent watchdog reset occurred 1: Independent watchdog reset occurred
28	SW_RSTF	Software reset flag

		Set by hardware when a software reset occurs. Cleared by setting the RMVF bit. 0: No software reset occurred 1: Software reset occurred
27	PMU_RSTF	POR/PDR and BOR flag Set by hardware when a POR/PDR and BOR occurs. Cleared by setting the RMVF bit. 0: No POR or BOR occurred 1: POR or BOR occurred
26	NRST_RSTF	NRST reset flag Set by hardware when a reset from the NRST pin occurs. Cleared by setting the RMVF bit. 0: No reset from NRST pin occurred 1: Reset from NRST pin occurred
25	OBL_RSTF	Option byte loader reset flag Set by hardware when a reset from the option byte loading occurs. Cleared by setting the RMVF bit. 0: No reset from option byte loading occurred 1: Reset from option byte loading occurred
24	LOCKUP_RSTF	LOCKUP reset flag Set by hardware when a reset from the LOCKUP failure occurs with LOCKUP enabled (LOCKUP_RSTEN set to 1). Cleared by setting the RMVF bit. 0: No reset from LOCKUP failure occurred 1: Reset from LOCKUP failure occurred
23:17	Reserved	Must be kept at reset value
16	RMVF	Remove reset flags Set by software to clear the reset flags, and this bit is also cleared to 0 simultaneously. 0: No effect 1: Clear reset flags

15:9	Reserved	Must be kept at reset value
8	LOCKUP_RSTEN	LOCKUP reset flag 0: Disable 1: Enable
7:2	Reserved	Must be kept at reset value
1	RCLRDY	RCL oscillator ready When RCLON is set to 0, RCLRDY will be automatically cleared. 0: RCL oscillator is not ready 1: RCL oscillator is ready
0	RCLON	RCL enable 0: Disable 1: Enable

7.4.19 RCL calibration register (RCC_RCLCAL)

Address offset: 0x50

Reset value: 0x0000 00XX

Note: This register is only reset by POR/PDR reset.

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

Bits	Name	Description
31:5	Reserved	Must be kept at reset value
4:0	RCL_CAL[4:0]	RCL clock calibration value

7.4.20 RCH calibration register (RCC_RCHCAL)

Address offset: 0x54

Reset value: 0x0000 00XX

Note: The register is only reset when both V_{DD} and V_{BAT} are powered off.

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

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

Bits	Name	Description
31:7	Reserved	Must be kept at reset value
6:0	RCH_CAL[6:0]	RCH clock calibration value

7.4.21 V_{REFBUF} calibration register (RCC_VREFBUFCAL)

Address offset: 0x10

Reset value: 0x0000 00XX

Note: This register is only reset by POR/PDR reset.

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

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

Bits	Name	Description
31:7	Reserved	Must be kept at reset value
6:0	VREFBUF_CAL[6:0]	V_{REFBUF} voltage calibration value

8 General-purpose I/Os (GPIO)

8.1 Introduction

Each general-purpose I/O port has three 32-bit configuration registers (GPIOx_MODE, GPIOx_OTYPE and GPIOx_PUPD), two 32-bit data registers (GPIOx_IDR and GPIOx_ODR), a 32-bit set/reset register (GPIOx_BSR), a 32-bit reset register (GPIOx_BR) and a 32-bit driving capability register (GPIOx_HDCFG). In addition, all GPIOs have two 32-bit alternate function selection registers (GPIOx_AFH and GPIOx_AFL).

8.2 GPIO main features

- Anti-backflow for each I/O
- High driving capability for two I/Os
- GPIO ports in several modes
  - Input mode
  - Output mode
  - Alternate function mode
  - Analog mode
- Pull-up/down configuration
- Push-pull or open drain configuration
- Highly flexible pin multiplexing allows the use of I/O pins as GPIOs or as one of several peripheral functions
- Bit set and reset register (GPIOx_BSR) for bitwise write access to the GPIOx_ODR register

8.3 GPIO functional description

Each port bit of the GPIO ports can be individually configured by software in several modes:

- Input floating
- Input pull-up

- Input pull-down
- Output open-drain with pull-up or pull-down capability
- Output push-pull with pull-up or pull-down capability
- Alternate function push-pull with pull-up or pull-down capability
- Alternate function open-drain with pull-up or pull-down capability
- Analog function

Figure 8-1 shows the basic structure of a standard I/O port bit. Table 8-1 gives the possible port bit configurations.

Figure 8-1 Basic structure of an I/O port bit

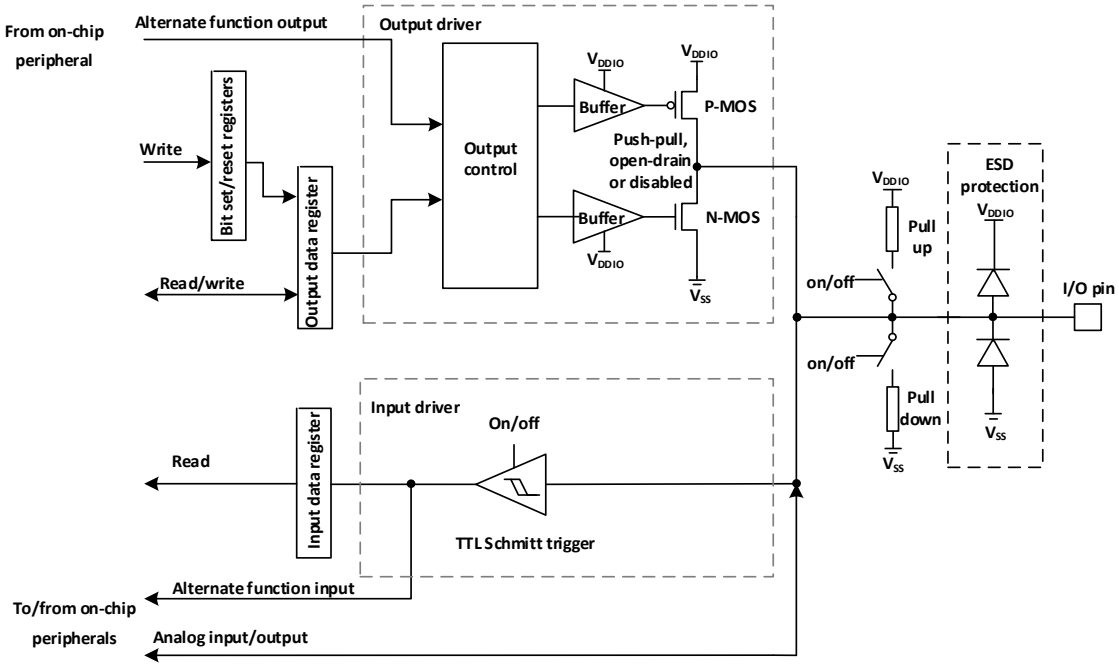


Table 8-1 Port bit configuration table(1)

I/O configuration		I/O registers configuration			
		MODE[1:0]	OTYPE	PUPD[1:0]	
Input	Floating	00	-	0	0
	Pull-up		-	0	1
	Pull-down		-	1	0
Output	Push-pull	01	0	0	0
	Push-pull + Pull-up		0	0	1
	Push-pull + Pull-down		0	1	0
	Open-drain		1	0	0

I/O configuration		I/O registers configuration			
		MODE[1:0]	OTYPE	PUPD[1:0]	
	Open-drain + Pull-up		1	0	1
	Open-drain + Pull-down		1	1	0
Alternate function	Push-pull	10	0	0	0
	Push-pull + Pull-up		0	0	1
	Push-pull + Pull-down		0	1	0
	Open-drain		1	0	0
	Open-drain + Pull-up		1	0	1
	Open-drain + Pull-down		1	1	0
Analog	Input/Output	11	-	Disabled by hardware	
			-		
			-		

1. “-”: invalid.

8.3.1 General-purpose I/O (GPIO)

During and just after reset, except PC9, PA13 and PA14, all of the I/O ports are configured in analog mode.

- PC9:
  - During reset, BOOT0 pin in pull-down
  - After reset, pin in analog mode
- PA14:
  - During and after reset, SWCLK in pull-down
- PA13:
  - During and after reset, SWDIO in pull-up

8.3.2 I/O pin alternate function multiplexer and mapping

Each I/O pin has GPIO function, peripheral alternate function and additional functions.

GPIO

Configure the desired I/O as output or input in the GPIOx_MODE register.

Peripheral alternate function

Each peripheral has alternate functions mapped onto different I/O pins. The device I/O pins are connected to on-board peripherals/modules through the multiplexer that allows only one peripheral alternate function connected to an I/O pin at a time. In this way, there can be no conflict between peripherals available on the same I/O pin.

- After reset, the multiplexer selection is alternate function 0 (AF0) that can be configured through the GPIOx_AFL and GPIOx_AFH registers.
- Peripheral alternate function
  - Each I/O pin has a multiplexer with up to eight alternate function inputs (AF0 to AF7) that can be configured through the GPIOx_AFL and GPIOx_AFH registers. Refer to *Port alternate function mapping* for more details.
  - Select the type, pull-up/pull-down and driving capability via the GPIOx_OTYPE, GPIOx_PUPD and GPIOx_HDCFG registers, respectively.
  - The I/Os are configured in alternate function mode via the GPIOx_MODE register.

Additional functions

For the additional functions, configure the required function in the related registers. These functions have priority over the configuration in the standard GPIO registers.

- Analog functions
  - ADC and COMP: Configure the required I/O in analog mode through the GPIOx_MODE register, and enable ADC or COMP by configuring their own control registers.
- Other functions
  - RTC: The I/O is capable of digital signals output by the RTC configuration. For details about I/O control by the RTC, refer to RTC chapter.
  - TAMP: The I/O is capable of digital signals input by the TAMP configuration. For details about I/O control by the TAMP, refer to TAMP chapter.
  - External clock source:

    The external high speed clock mode (HXTAL) is selected by setting the

HXTALBYP and HXTALON bits in the *Clock control register 1 (RCC_CSR1)*. When the clock source is an external oscillator, the desired I/Os (HXTAL_IN and HXTAL_OUT) are used as analog pins. When the clock source is an external clock source, only the HXTAL_IN pin is reserved for clock input as an analog pin and the HXTAL_OUT pin can still be used as normal GPIO.

The external low speed clock mode (LXTAL) is selected by setting the LXTALON bit in the *V_{CORE_AON} domain contol register (RCC_AWCR)*. The desired I/Os (LXTAL_IN and HXTAL_OUT) are used as analog pins.

8.3.3 I/O port control registers

Each of the GPIO ports has three 32-bit memory-mapped control registers (GPIOx_MODE, GPIOx_OTYPE, GPIOx_PUPD) to configure I/Os.

- The GPIOx_MODE register is used to select the I/O mode (input, output, alternate function, analog).
- The GPIOx_OTYPE register is used to select the output type (push- pull or open-drain).
- The GPIOx_PUPD register is used to select the pull- up/pull-down whatever the I/O direction.

8.3.4 I/O port data registers

Each GPIO has two 32-bit memory-mapped data registers: input and output data registers (GPIOx_IDR and GPIOx_ODR).

- The GPIOx_ODR register stores the data to be output, it is read/write accessible.
- The data input through the I/O are stored into the input data register (GPIOx_IDR), a read-only register.

8.3.5 I/O data bitwise handling

The bit set/reset register (GPIOx_BSR) is a 32-bit register which allows the application to set and reset each individual bit in the output data register (GPIOx_ODR).

- To each bit in the GPIOx_ODR register, correspond two control bits in the GPIOx_BSRR register: BS(i) and BR(i). When written to 1, bit BS(i) sets the corresponding ODR(i) bit. When written to 1, bit BR(i) resets the ODR(i) corresponding bit.

- Writing any bit to 0 in the GPIOx_BSR register does not have any effect on the corresponding bit in the GPIOx_ODR register.
- If there is an attempt to both set and reset a bit in the GPIOx_BSRR register, the set action takes priority.

Using the GPIOx_BSR register to change the values of individual bits in the GPIOx_ODR register is a “one-shot” effect. The bits of the GPIOx_ODR register can always be accessed directly instead of read-write process. When programming the GPIOx_ODR register at bit level, the process will not be interrupted, and there is no need for the software to disable interrupts.

8.3.6 External interrupt/wakeup lines

All ports have external interrupt capability, which are configured and managed by EXTI. Refer to *Extended interrupt and event controller (EXTI).* When there are multiple I/O pins connected via EXTI lines, the I/O pins should be selected on different EXTI lines through the *EXTI external interrupt selection register (EXTI_EXTICRx)*. For example: PA0, PB0, PC0… are on the same EXTI line, and PA1, PB1, PC1… are on the same EXTI line.

8.3.7 Input configuration

Figure 8-2 Input floating/pull up/pull down configurations

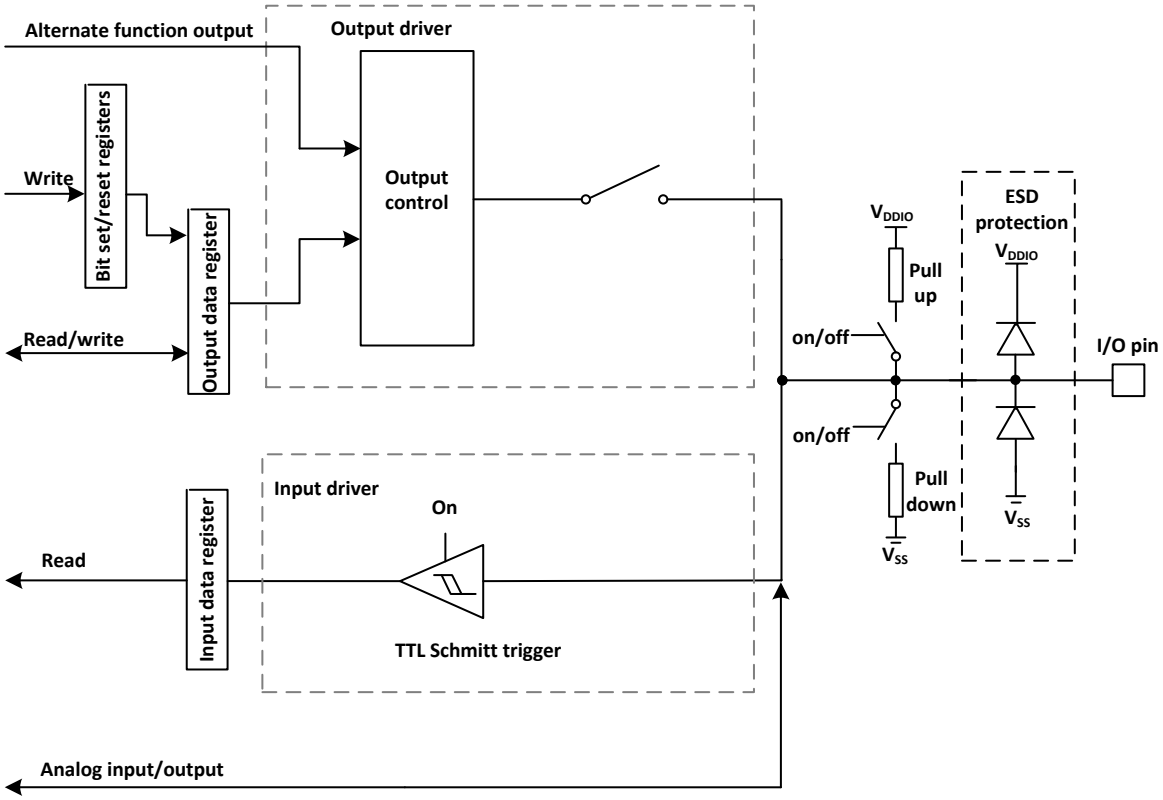


Figure 8-2 shows the input configuration of the I/O port bit. When the I/O port is programmed as input:

- The Schmitt trigger input is activated
- The pull-up and pull-down resistors are activated depending on the value in the GPIOx_PUPD register
- The output buffer is disabled
- A read access to the input data register provides the I/O state

8.3.8 Output configuration

Figure 8-3 Output configuration

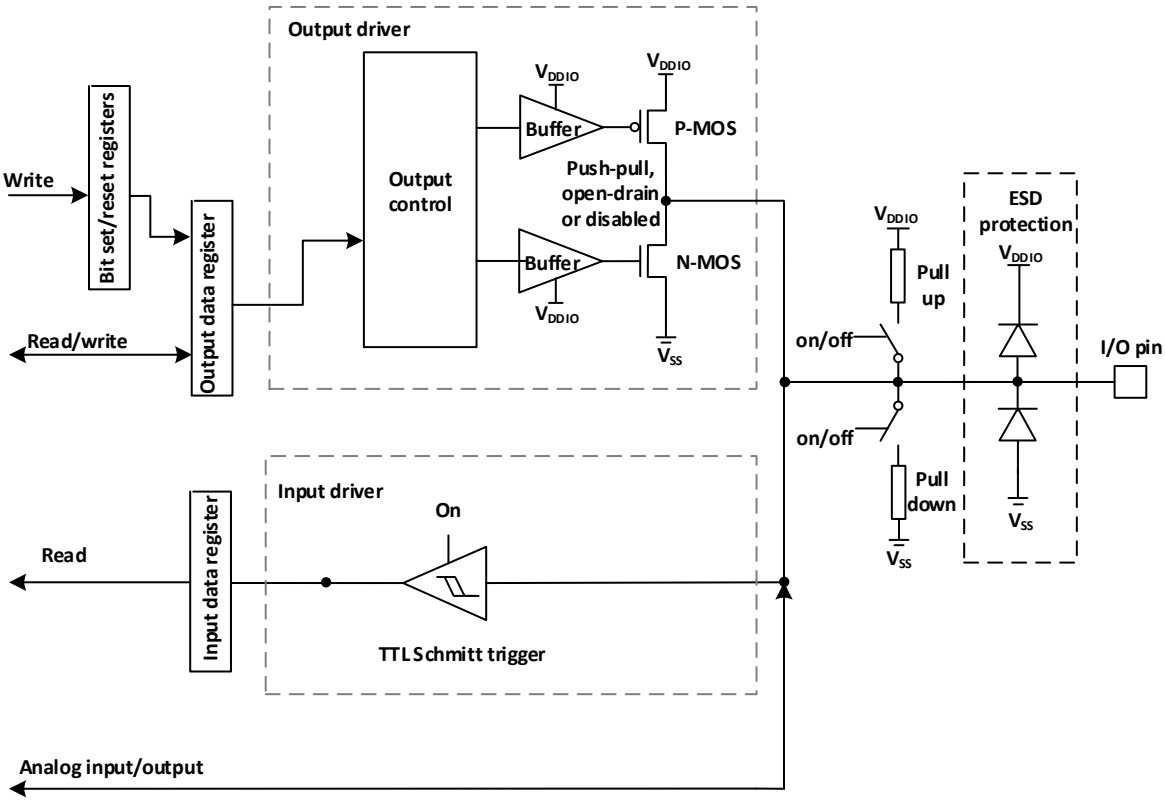


Figure 8-3 shows the output configuration of the I/O port bit. When the I/O port is programmed as output:

- The output buffer is enabled:
  - Open drain mode: A “0” in the output data register activates the N-MOS whereas a “1” in the output data register leaves the port in Hi-Z (the P-MOS is never activated)
  - Push-pull mode: A “0” in the output data register activates the N-MOS

whereas a “1” in the output data register activates the P-MOS

- A read access to the output data register gets the last written value
- The Schmitt trigger input is activated
- The pull-up and pull-down resistors are activated depending on the value in the GPIOx_PUPD register
- A read access to the input data register gets the I/O state

8.3.9 Alternate function configuration

Figure 8-4 Alternate function configuration

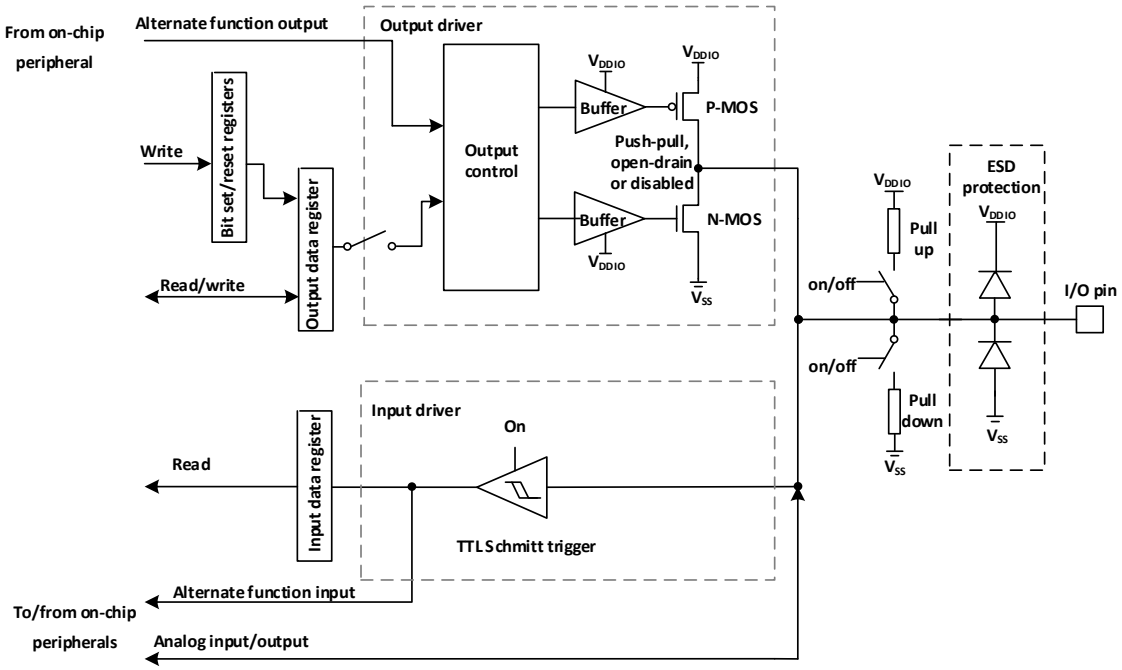


Figure 8-4 shows the alternate function configuration of the I/O port bit. When the I/O port is programmed as alternate function:

- The Schmitt trigger input is activated
- The weak pull-up and pull-down resistors are activated or not depending on the value in the GPIOx_PUPD register
- A read access to the input data register gets the I/O state
- The output buffer can be configured in open-drain or push-pull mode
- The output buffer is driven by the signals coming from the peripheral (transmitter enable and data)

8.3.10 I/O alternate function input/output

Two registers are provided to select one of the alternate function inputs/outputs available for each I/O. With these registers, the user can connect an alternate function to some other pin as required by the application. This means that a number of possible peripheral functions are multiplexed on each GPIO using the GPIOx_AFH and GPIOx_AFL alternate function registers. The application can thus select any one of the possible functions for each I/O. Refer to *Port alternate function mapping* for more details.

8.3.11 Analog configuration

Figure 8-5 High impedance-analog configuration

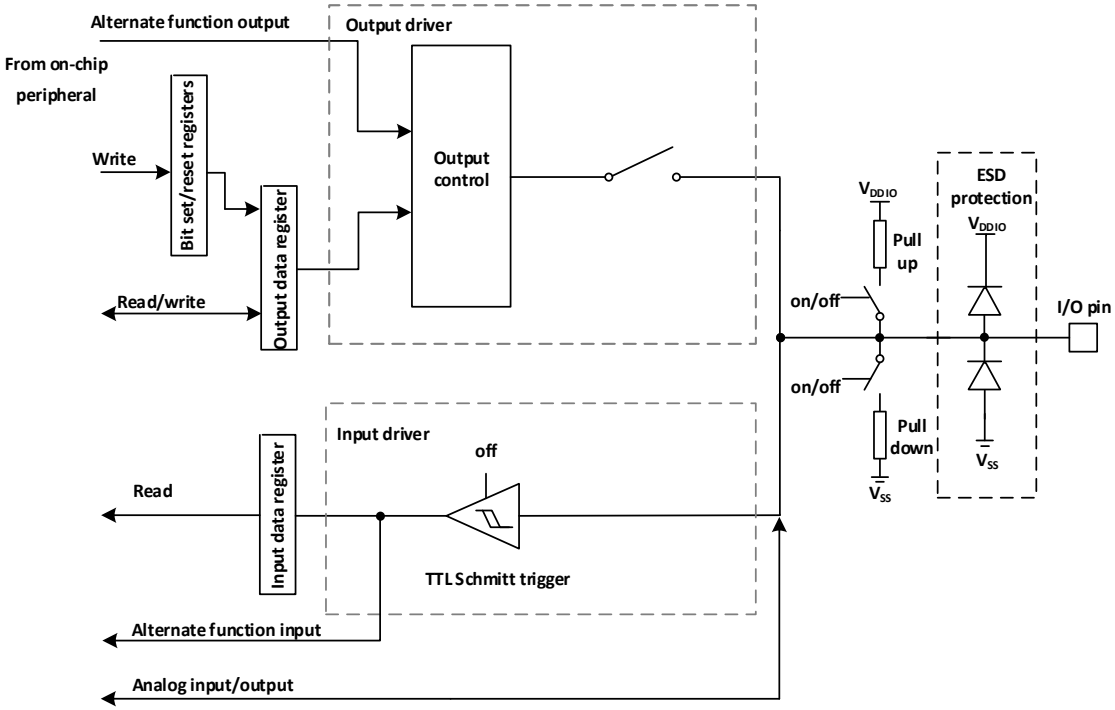


Figure 8-5 shows the high-impedance, analog-input configuration of the I/O port bits. When the I/O port is programmed as analog configuration:

- The output buffer is disabled
- The Schmitt trigger input is deactivated, providing zero consumption for every analog value of the I/O pin. The output of the Schmitt trigger is forced to a constant value “0”.
- The weak pull-up and pull-down resistors are disabled by hardware
- Read access to the input data register gets the value “0”

8.3.12 I/O driving capability

The driving capability for the I/Os can be divided into three categories: the low driving capability, the medium driving capability and the configurable driving capability (high driving or medium driving). When the I/O port is setting for driving capability:

- The PA2 and PD2 pins can be configured in high driving or medium driving mode
- PC13, PC14 and PC15 pins for V_{BAT} power are in low driving mode
- The other pins are in medium driving mode

8.3.13 Using the GPIO pins in the V_{CORE_AON} domain

The PC13/PC14/PC15's GPIO functionality is lost when the core supply domain is powered off (when the device enters V_{BAT} mode). In this case, the PC13/PC14/PC15 are in an analog input mode. The PC13 can be configured to the RTC function, for details about I/O control by the RTC, refer to *Real-time clock (RTC)*. The PC14/PC15 can be configured to LXTAL function by the *RCC_AWCR* register in the V_{CORE_AON} domain.

8.4 GPIO registers

The GPIO registers can only be accessed by words (32-bit).

Table 8-2 GPIO base address

Peripheral	Base address
GPIOA	0x5000 0000
GPIOB	0x5000 0400
GPIOC	0x5000 0800
GPIOD	0x5000 0C00
GPIOF	0x5000 1400

8.4.1 GPIO port mode register (GPIOx_MODE) (x = A to F)

Address offset: 0x00

Reset value: 0xEBFF FFFF (for port A)

Reset value: 0x0000 003F (for port F)

Reset value: 0xFFFF FFFF (for other ports)

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

Bits	Name	Description
31:0	MODEy[1:0]	Port x configuration I/O pin y (x = A to F, y = 0 to 15) These bits are written to configure the I/O mode. 00: Input mode 01: General purpose output mode 10: Alternate function mode 11: Analog mode

8.4.2 GPIO port output type register (GPIOx_OTYPE) (x = A to F)

Address offset: 0x04

Reset value: 0x0000 0000

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

Bits	Name	Description
31:16	Reserved	Must be kept at reset value
15:0	OTy[15:0]	Port x configuration I/O pin y (x = A to F, y = 0 to 15) These bits are written to configure the I/O output type. 0: Output push-pull 1: Output open-drain

8.4.3 GPIO port pull-up/pull-down register (GPIOx_PUPD) (x = A to F)

Address offset: 0x0C

Reset value: 0x2400 0000 (for port A)

Reset value: 0x0000 0000 (for others)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUPD15[1:0]		PUPD 14[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

Bits	Name	Description
31:0	PUPDy[1:0]	Port x configuration I/O pin y (x = A to F, y = 0 to 15) These bits are written to configure the I/O pull-up or pull-down. 00: No pull-up, pull-down 01: Pull-up 10: Pull-down 11: Reserved *Note: Writing reserved value will not change the status depending on the previous valid configuration.*

8.4.4 GPIO port input data register (GPIOx_IDR) (x = A to F)

Address offset: 0x10

Reset value: 0x0000 XXXX

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

Bits	Name	Description
31:16	Reserved	Must be kept at reset value
15:0	IDy[15:0]	Port x input data I/O pin y (x = A to F, y = 0 to 15) These bits are read-only. They contain the input value of the corresponding I/O port.

8.4.5 GPIO port output data register (GPIOx_ODR) (x = A to F)

Address offset: 0x14

Reset value: 0x0000 0000

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

Bits	Name	Description
31:16	Reserved	Must be kept at reset value
15:0	ODy	Port x output data I/O pin y (x = A to F, y = 0 to 15) These bits can be read and written by software.

8.4.6 GPIO port bit set/reset register (GPIOx_BSR) (x = A to F)

Address offset: 0x18

Reset value: 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

Bits	Name	Description
31:16	BRy	Port x reset I/O pin y (x = A to F, y = 0 to 15) These bits are write-only. A read to these bits gets the value 0x0000. 0: No action on the corresponding ODx bit 1: Resets the corresponding ODx bit *Note: If both BSy and BRy are set, BSy has priority.*
15:0	BSy	Port x set I/O pin y (x = A to F, y = 0 to 15) These bits are write-only. A read to these bits gets the value 0x0000. 0: No action on the corresponding ODx bit 1: Sets the corresponding ODx bit

8.4.7 GPIO alternate function low register (GPIOx_AFL) (x = A to F)

Address offset: 0x20

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	AFSEL7[2:0]			Res.	AFSEL6[2:0]			Res.	AFSEL5[2:0]			Res.	AFSEL4[2:0]		
	rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	AFSEL3[2:0]			Res.	AFSEL2[2:0]			Res.	AFSEL1[2:0]			Res.	AFSEL0[2:0]		
	rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw

Bits	Name	Description
31:0	AFSELy[2:0]	Alternate function selection for port x pin y (x = A to F, y = 0 to 7) These bits are written to configure alternate function I/Os. 000: AF0 001: AF1 010: AF2

011: AF3

100: AF4

101: AF5

110: AF6

111: AF7

Note: Writing reserved bits has no effect.

8.4.8 GPIO alternate function high register (GPIOx_AFH) (x = A to F)

Address offset: 0x24

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	AFSEL15[2:0]			Res.	AFSEL14[2:0]			Res.	AFSEL13[2:0]			Res.	AFSEL12[2:0]		
	rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	AFSEL11[2:0]			Res.	AFSEL10[2:0]			Res.	AFSEL9[2:0]			Res.	AFSEL8[2:0]		
	rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw

Bits	Name	Description
31:0	AFSELy[2:0]	Alternate function selection for port x pin y (x = A to F, y = 8 to 15) These bits are written to configure alternate function I/Os. 000: AF0 001: AF1 010: AF2 011: AF3 100: AF4 101: AF5 110: AF6 111: AF7 *Note: Writing reserved bits has no effect.*

8.4.9 GPIO port bit reset register (GPIOx_BR) (x = A to F)

Address offset: 0x28

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits	Name	Description
31:16	Reserved	Must be kept at reset value
15:0	BRy	Port x reset IO pin y (x = A to F, y = 0 to 15) These bits are write-only. A read to these bits gets the value 0x0000. 0: No action on the corresponding ODx bit 1: Resets the corresponding ODx bit

8.4.10 GPIO port driving capability register (GPIOx_HDCFG) (x = A or D)

Address offset: 0x2C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HD15	HD14	HD13	HD12	HD11	HD10	HD9	HD8	HD7	HD6	HD5	HD4	HD3	HD2	HD1	HD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Name	Description
31:16	Reserved	Must be kept at reset value
15:0	HDy	Port x configuration I/O pin y (x = A or D, y = 0 to 15) These bits are written to configure the I/O driving capability. 0: High driving capability disabled 1: High driving capability enabled *Note: Only the driving capabilities of PA2 pin in the PortA port and PD2 pin in the PortD port can be configured, the other pins cannot be configured.*

9 System configuration controller (SYSCFG)

9.1 Introduction

The main purposes of the system configuration controller are the following:

- Configuring the IRTIM modulation signal and its output polarity
- Configuring the 6-bit DAC
- Remapping the memory

9.2 SYSCFG registers

The SYSCFG register can only be accessed by words (32-bit).

Table 9-1 SYSCFG base address

Peripheral	Base address
SYSCFG	0x4001 0000

9.2.1 SYSCFG control register (SYSCFG_CR)

Address offset: 0x00

Reset value: 0x0000 000X

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.						6BIT_DAC_REF	6BIT_DAC_EN	Res.		6BIT_DAC_DIV[5:0]					
						rw	rw			rw	rw	rw	rw	rw	rw

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

Bits	Name	Description
31:26	Reserved	Must be kept at reset value
25	6BIT_DAC_REF	6-bit DAC input reference voltage source select 0: VREFBUF or V_{REF+} 1: V_{DDA}
24	6BIT_DAC_EN	6-bit DAC enable 0: Disable 1: Enable *Note: After enabling the 6-bit DAC, it is necessary to wait t_{STAB} for output stable (The t_{STAB} please refer datasheet)*
23:22	Reserved	Must be kept at reset value
21:16	6BIT_DAC_DIV[5:0]	6-bit DAC voltage divider of input voltage source 000000: 1/64 000001: 2/64 000010: 3/64

		000011: 4/64 ⋮ 111110: 63/64 111111: 64/64
15:5	Reserved	Must be kept at reset value
4:3	IR_MODE[1:0]	IRTIM modulation envelope signal selection Refer to *Infrared interface (IRTIM)* 00: TIM5_OC1 01: USART1_TX 10: UART4_TX 11: Reserved (keep the latest configuration)
2	IR_POL	IR output polarity selection 0: Output of IRTIM (IR_OUT) is not inverted 1: Output of IRTIM (IR_OUT) is inverted
1:0	MEM_MODE[1:0]	Memory mapping selection bits These bits control the memory internal mapping at address 0x0000 0000. After reset these bits take on the value selected by the actual boot mode configuration. Refer to Table: *Boot moods*. 0x: User Flash memory mapped at 0x0000 0000 10: System memory mapped at 0x0000 0000 11: SRAM mapped at 0x0000 0000

9.2.2 SYSCFG secure control register (SYSCFG_SECCR)

Address offset: 0x04

Reset value: 0x0000 0000

Note: This register cannot be reset when the SYSCFG_RST bit in the APB2 peripheral reset register (RCC_APB2RST) is set.

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

Res.										PVD_LOCK	Res.				
										rs					

Bits	Name	Description
31:6	Reserved	Must be kept at reset value
5	PVD_LOCK	PVD lock enable bit Set by software and cleared by a system reset. Write 0 is invalid. It locks the whole content of the PMU_CR2 register. 0: PMU_CR2 register is unlocked and all bits can be read or written 1: PMU_CR2 register is locked and all bits are read only
4:0	Reserved	Must be kept at reset value

10 Direct memory access controller (DMA)

10.1 Introduction

The direct memory access (DMA) controller is a bus master and system peripheral. The DMA is used to perform programmable data transfers between memory-mapped peripherals and/or memories, upon the control of an off-loaded CPU.

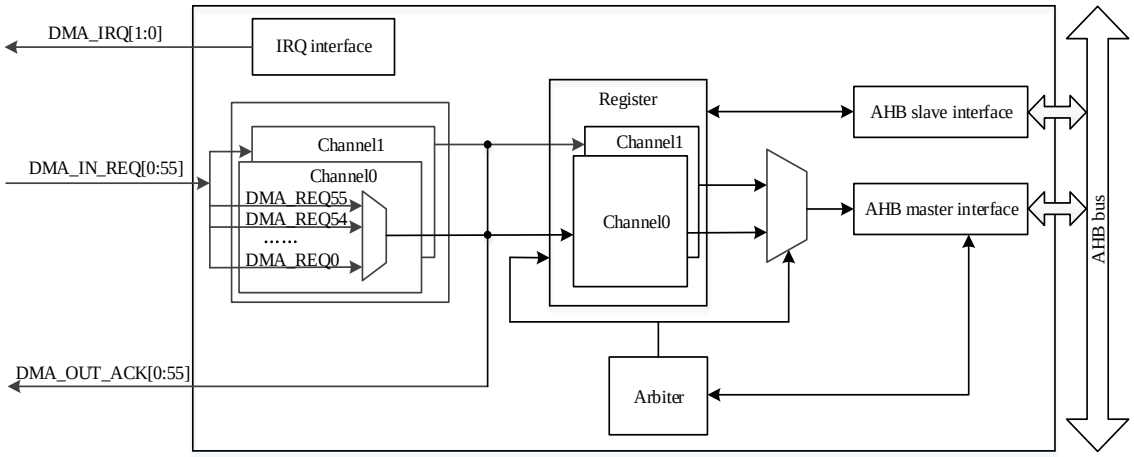
10.2 DMA main features

- Two independent DMA channels, with an integrated arbiter to handle channel requests between the channels
- Two types of channel request triggers: peripheral trigger and software trigger
- Four DMA transfer modes: memory-to-memory, memory-to-peripheral, peripheral-to-memory, peripheral-to-peripheral data transfers
- Two DMA transfer types: block transfer and burst transfer
- Supports the address incremented or non-incremented
- Programmable transfer sizes: byte, half-word, or word
- Programmable number of data to be transferred: 0 to 65535
- Supports the circular mode
- Three event flags (transfer complete, half transfer, and transfer error) in a single interrupt request for each channel

10.3 DMA functional description

10.3.1 DMA block diagram

Figure 10-1 DMA block diagram



10.3.2 DMA signals

Table 10-1 DMA internal signals

Signal name	Signal type	Description
DMA_IN_REQ[0:55]	Input	DMA channel requests
DMA_OUT_ACK[0:55]	Output	DMA channel acknowledges
DMA_IRQ[1:0]	Output	DMA channel interrupts

10.3.3 DMA channel requests

All channel requests supported by DMA channel are shown in table below.

Table 10-2 DMA channel request signals

Request ID	Request source	Request signal	
		Channel 0	Channel 1
0	Software trigger	Software trigger of DMA request	
1	ADC	ADC	
2	SPI1	SPI1_RX	SPI1_TX
3	SPI2	SPI2_RX	SPI2_TX
4	I2C1	I2C1_RX	I2C1_TX
5	Reserved		
6	USART1	USART1_RX	USART1_TX

Request ID	Request source	Request signal	
		Channel 0	Channel 1
7	UART2	UART2_RX	UART2_TX
8	UART3	UART3_RX	UART3_TX
9	UART4	UART4_RX	UART4_TX
10	LPUART1	LPUART1_RX	LPUART1_TX
11	LPUART2	LPUART2_RX	LPUART2_TX
12~14	Reserved		
15	TIM3	TIM3_CC1	TIM3_CC2
16		TIM3_CC3	TIM3_CC4
17		TIM3_TRIG	TIM3_UP
18	TIM4	TIM4_CC1	TIM4_CC2
19		TIM4_CC3	TIM4_CC4
20		TIM4_TRIG	TIM4_UP
21	TIM5	TIM5_CC1	TIM5_CC2
22		TIM5_CC3	TIM5_CC4
23		TIM5_TRIG	TIM5_UP
24	TIM8	TIM8_UP	Reserved
25	Reserved		
26	AES	AES_IN	AES_OUT
27	LPTIM1	LPTIM1_CMPM	LPTIM1_ARRM
28	LPTIM2	Reserved	LPTIM2_ARRM

There are two types of DMA channel request generation:

- Peripheral trigger channel request: REQ_ID[4:0] bit field of the DMA_CCx register is set to 1~28
- Software trigger channel request: REQ_ID[4:0] bit field of the DMA_CCx register is set to 0

10.3.4 DMA transfers

After DMA channel enabled, it first reads a single data (8/16/32 bits) from the source address, and then writes the single data (8/16/32 bits) to the destination address. The NDT[15:0] bit field of the DMA_CNDTRx register is decremented after each transfer, indicating the number of times to be transferred. If no transfer error occurs, the transfer

process will be repeated until the value of the DMA_CNDTRx register is decremented to zero.

Transfer modes

There are four transfer modes based on the address types of source and destination address:

- Memory-to-memory

  When both the source and destination addresses are memory locations, and select software trigger for the channel request. Once the channel is enabled, the transfer immediately initiates and continues until the value in the DMA_CNDTRx register decrements to zero, at which point the transfer stops.

Note: The circular mode can not be used in memory-to-memory mode.

- Memory-to-peripheral or peripheral-to-memory

  In this transfer mode, once the channel is enabled, the transfer is triggered by the peripheral request and continues until the value in the DMA_CNDTRx register decrements to zero, at which point the transfer stops.

- Peripheral-to-peripheral

  When both the source and destination addresses are peripherals locations, and a channel request is triggered by peripheral. The another peripheral acts as passive access peripheral for this channel. When the channel request is software-triggered, the transfer immediately starts upon channel enablement, similar to the memory-to-memory transfer mode.

Transfer types

Block transfer type and Burst transfer type can be selected through the TYPE bit in the DMA_CCx register. The channel request can be triggered by peripheral or software.

Table 10-3 DMA transfer types

Function Comparison	Software trigger Block transfer	Software trigger Burst transfer	Peripheral trigger Block transfer	Peripheral trigger Burst transfer
Start transfer condition	Channel enable	Channel enable	Peripheral trigger channel request	Peripheral trigger channel request
CNDTR register value	Pending transfer data number	Pending transfer data number	Pending handle request number	Pending transfer data number
Single request transfer	NDT(1)	NDT	1	NDT

number				
Total transfer number(2)	NDT	NDT	NDT	NDT
Channel pause and stop	Support	Not Support	Support	Not Support
Transfer mode	Memory-to-memory	Memory-to-memory	Memory-to-peripheral or peripheral-to-memory	Memory-to-peripheral or peripheral-to-memory

1. NDT refers to the value written into NDT[15:0] of the CNDTR register.
2. Total transfer number refers to the total data number of DMA transfer completed when the value written to the trigger CNDTR is NDT.

10.3.5 Programmable data sizes

The transfer sizes of single data (byte, half-word, or word) is determined by the SIZE[1:0] fields of the DMA_CCx register.

Note: Transfer address and transfer size width must be aligned, otherwise the hardware will automatically align the address, resulting unexpected transfer.

Table 10-4 DMA address alignment

SIZE[1:0]	Data size	Source and destination address
00	Byte	Unlimited
01	Half word	Address bit 0 is set to 0
10	Word	Address bit 0 and bit 1 are set to 0

10.3.6 Address increment

The SINC and DINC bits in the DMA_CCx register determine whether the source or destination address pointer is incremented after each transfer, as shown in table below.

Table 10-5 DMA address increment

SINC	DINC	Source address	Destination address
0	0	Address fixed	Address fixed
0	1	Address fixed	Address incremented
1	0	Address incremented	Address fixed
1	1	Address incremented	Address incremented

When SINC or DINC is set to 1, the source address pointer or destination address pointer adds 1, 2, or 4 to the transfer address, depending on the data bit width configured by SIZE[1:0].

The DMA_CSARx and DMA_CDARx registers indicate the current transfer address.

10.3.7 Circular mode

Non circular mode

When the DMA channel is configured in non-circular mode (CIR = 0), after the last data transfer is complete (the DMA_CNDTRx register reaches zero), the channel remains enabled but does not handle any channel requests. If the same source and destination address are still used for next transfer, simply disable the DMA channel first, change the value of DMA_CNDTRx register, and then enable the DMA channel again.

Circular mode

When the DMA channel is configured in circular mode (CIRC = 1), the DMA_CNDTRx register will automatically reload the initial value, and the address registers are reloaded with the starting addresses from the DMA_CSARx and DMA_CDARx registers after the last data transfer is complete.

Circular mode can be used to handle loop buffers. For example, the data from a 10-channel ADC circular scan is saved in a fixed buffer area in SRAM. When the ADC completes the data acquisition for all 10 channels and returns to the first channel, the DMA channel's starting address also automatically resets to the initial address. When the ADC requests another transfer, the data from the same channels will overwrite the previous data.

10.3.8 Channel configuration procedure

The following sequence is needed to configure a DMA channel x:

1) Configure the parameters listed below in the DMA_CCx register:
   - the transfer data width (SIZE[1:0])
   - the source address increment mode (SINC)
   - the destination address increment mode (DINC)
   - the transfer type of DMA (TYPE)
   - the channel request ID (REQ_ID[4:0])
   - the circular mode (CIRC)
   - transfer complete interrupt enable (TFIE), half transfer interrupt enable

(THIE), and transfer error interrupt enable (TEIE)

2) Set the transfer number in the DMA_CNDTRx register.

3) Set the transfer source address in the DMA_CSARx register.

4) Set the transfer destination address in the DMA_CDARx register.

5) Set the EN bit of the DMA_CCx register to enable DMA channel.

10.3.9 Suspend and stop a channel

When DMA channel is set to the Burst transfer type, ongoing transfers can not be paused or stopped. When DMA channel is set to the Block transfer type, the transfer can be paused or stopped.

- Suspend and resume a channel

  Before the DMA channel transfer number decreases to zero, clear the EN bit in the DMA_CCx register, and do not change the configuration of channel, the channel will be suspended. Set the EN bit of the DMA_CCx register to resume channel transfer.

- Stop and restart transfer

  Before the DMA channel transfer number decreased to zero, if DMA channel request is triggered by software, once the EN bit of the DMA_CCx register is cleared, the channel transfer will stop. If restarting this channel for other transfer, it needs reconfigure any parameters in the DMA_CCx, DMA_CSARx, DMA_CDARx, or DMA_CNDTRx registers and enable the channel to restart transfer again.

  If channel request is triggered by peripheral, disable the peripheral request first, and then disable the DMA channel (EN = 0) to ensure that there is no pending channel request. If restarting this channel for other transfer, it needs reconfigure any parameters in the DMA_CCx, DMA_CSARx, DMA_CDARx, or DMA_CNDTRx registers, and enable the channel again. Waiting for the peripheral trigger a new channel request, and then the channel transfer is restarted.

  To ensure the integrity of data transfer, when DMA channel is disabled during DMA transfer, the software need to wait for a certain delay to ensure that the DMA is reconfigured after the completion of a single transfer. The delay of different transfer mode is as follows:

- DMA transfer with peripheral trigger channel request: delay of 3 PCLK cycles and 2 HCLK cycles
- DMA transfer between memories other than Flash write operations: delay of 6 HCLK cycles
- DMA transfer with Flash write operation: wait for the *Flash status register (FLASH_SR)* BSY bit to clear 0 before reconfiguring DMA

Note: The DMA_CCx, DMA_CSARx, DMA_CDARx, and DMA_CNDTRx registers maintain the state value when channel is disabled (EN = 0).

10.3.10 Error management

A DMA transfer error is generated when reading from or writing to a reserved address space, and the bit TEx of the DMA_ISR register is set. At the same time, the faulty channel x is automatically disabled through a hardware clear of its EN bit in the corresponding DMA_CCx register. The EN bit of the DMA_CCx register can not be set again by software (channel x reactivated) until the TEx bit of the DMA_ISR register is cleared (by setting the TECFx bit of the DMA_ICR register).

When the software is notified with a transfer error over a channel which involves a peripheral, the software has first to stop this peripheral in DMA mode, in order to disable any pending or future DMA request. Clear TEx and then reconfigure both DMA and peripheral in DMA mode for a new transfer.

10.4 DMA interrupts

An interrupt can be generated on a half transfer, transfer complete or transfer error for each DMA channel x. Each event can be configured to enable or disable interrupts. Please refer to the table below for details of DMA interrupts.

Table 10-6 DMA interrupt requests

Interrupt event	Event flag	Enable bit	Clear flag method
Transfer complete on channelx	TFx	TFIE	Write TFCFx = 1 or GCFx = 1
Half transfer on channelx	THx	THIE	Write THCFx = 1 or GCFx = 1
Transfer error on channelx	TEx	TEIE	Write TECFx = 1 or GCFx = 1
Half transfer or transfer complete or transfer error on channel x	Gx	-	Write GCFx = 1 or TFCFx = 1, THCFx = 1, and TECFx = 1

10.5 DMA registers

The DMA register can only be accessed by words (32-bit).

Table 10-7 DMA base address

Peripheral	Base address
DMA	0x4002 0000

10.5.1 DMA interrupt status register (DMA_ISR)

Address offset: 0x00

Reset Value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								TE1	TH1	TF1	G1	TE0	TH0	TF0	G0
								r	r	r	r	r	r	r	r

Bits	Name	Description
31:8	Reserved	Must be kept at reset value
7	TE1	Transfer error (TE) flag for channel 1 0: No TE event 1: A TE event occurred
6	TH1	Half transfer (TH) flag for channel 1 0: No TH event 1: A TH event occurred
5	TF1	Transfer complete (TF) flag for channel 1 0: No TF event 1: A TF event occurred
4	G1	Global interrupt flag for channel 1 0: No TE, TH or TF event

		1: A TE, HT or TC event occurred
3	TE0	Transfer error (TE) flag for channel 0 0: No TE event 1: A TE event occurred
2	TH0	Half transfer (TH) flag for channel 0 0: No TH event 1: A TH event occurred
1	TF0	Transfer complete (TF) flag for channel 0 0: No TF event 1: A TF event occurred
0	G0	Global interrupt flag for channel 0 0: No TE, TH or TF event 1: A TE, HT or TC event occurred

10.5.2 DMA interrupt flag clear register (DMA_ICR)

Address offset: 0x04

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								TECF1	THCF1	TFCF1	GCF1	TECF0	THCF0	TFCF0	GCF0
								w	w	w	w	w	w	w	w

Bits	Name	Description
31:8	Reserved	Must be kept at reset value
7	TECF1	Transfer error flag clear for channel 1
6	THCF1	Half transfer flag clear for channel 1

5	TFCF1	Transfer complete flag clear for channel 1
4	GCF1	Global interrupt flag clear for channel 1
3	TECF0	Transfer error flag clear for channel 0
2	THCF0	Half transfer flag clear for channel 0
1	TFCF0	Transfer complete flag clear for channel 0
0	GCF0	Global interrupt flag clear for channel 0

10.5.3 DMA channel x configuration register (DMA_CCx)

Address offset: 0x08 + 0x14 × x, (x = 0 to 1)

Reset value: 0x0000 0000

Note: After channel (EN = 1) enabled, this register is read-only in all bit field except for TEIE, THIE, and TFIE.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.											REQ_ID[4:0]				
											rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			TYPE	CIRC	Res.			SIZE[1:0]		SINC	DINC	TEIE	THIE	TFIE	EN
			rw	rw				rw	rw	rw	rw	rw	rw	rw	rw

Bits	Name	Description
31:21	Reserved	Must be kept at reset value
20:16	REQ_ID[4:0]	Channel request ID Refer to Table: *DMA channel request signals*
15:13	Reserved	Must be kept at reset value
12	TYPE	DMA transfer type

		0: Block transfer 1: Burst transfer
11	CIRC	Circular mode 0: Disable 1: Enable
10:8	Reserved	Must be kept at reset value
7:6	SIZE[1:0]	Transfer data width 00: 8 bits 01: 16 bits 10: 32 bits 11: Reserved (keep the latest configuration)
5	SINC	Source address increment mode 0: Disable 1: Enable
4	DINC	Destination address increment mode 0: Disable 1: Enable
3	TEIE	Transfer error interrupt enable 0: Disable 1: Enable
2	THIE	Half transfer interrupt enable 0: Disable 1: Enable
1	TFIE	Transfer complete interrupt enable 0: Disable 1: Enable

0	EN	Channel enable 0: Disable 1: Enable

10.5.4 DMA channel x number of data to transfer register (DMA_CNDTRx)

Address offset: 0x0C + 0x14 × x, (x = 0 to 1)

Reset value: 0x0000 0000

Note: The register is read only when the channel is enabled (EN = 1).

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

Bits	Name	Description
31:16	Reserved	Must be kept at reset value
15:0	NDT[15:0]	Number of data to transfer (0 to 2^{16} - 1) When channel is enabled, this field will decrease by 1 after each DMA transfer, indicating the remaining amount of data items to transfer. If this field is zero, no transfer can be served whatever the channel status (enabled or not).

10.5.5 DMA channel x source address register (DMA_CSARx)

Address offset: 0x10 + 0x14 × x, (x = 0 to 1)

Reset value: 0x0000 0000

Note: The register can only write address in the area of User flash, SRAM, or Peripheral registers, and the register is read only when the channel is enabled (EN = 1).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SA[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
SA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Name	Description
31:0	SA[31:0]	Source address When SIZE[1:0] = 01 (16 bits), bit 0 of SA[31:0] is ignored. Access is automatically aligned to a half-word address. When SIZE[1:0] = 10 (32 bits), bit 1 and 0 of SA[31:0] are ignored. Access is automatically aligned to a word address.

10.5.6 DMA channel x destination address register (DMA_CDARx)

Address offset: 0x14 + 0x14 × x, (x = 0 to 1)

Reset value: 0x0000 0000

Note: The register can only write address in the area of User flash, SRAM, or Peripheral registers, and the register is read only when the channel is enabled (EN = 1).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DA[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
DA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Name	Description
31:0	DA[31:0]	Destination address When SIZE[1:0] = 01 (16 bits), bit 0 of DA[31:0] is ignored. Access is automatically aligned to a half-word address. When SIZE[1:0] = 10 (32 bits), bit 1 and 0 of DA[31:0] are ignored. Access is automatically aligned to a word address.

11 Nested vectored interrupt controller (NVIC)

11.1 Introduction

The Nested vectored interrupt controller (NVIC) implements efficient exception handling and interrupt management. All interrupts are managed by the NVIC.

11.2 NVIC main features

- 32 maskable interrupt channels and 16 interrupts of the Cortex-M0+
- Low-latency exception and interrupt handling
- 4 programmable priority levels, "0" represents the highest priority
- The numbers in the interrupt vector table represent hardware priorities. If two interrupts with the same software priority occur simultaneously, the interrupt with the lower numerical value in the hardware priority field has higher priority

The NVIC and the processor core interface are closely coupled, which enables low-latency interrupt processing and efficient processing of late arriving higher priority interrupts. When an exception occurs, the NVIC automatically saves the processor state onto the stack. Upon completion of the interrupt service routine (ISR), the state is restored from the stack.

11.3 Interrupt and exception vectors

Table 11-1 Vector table(1)

Position	Priority	Type of priority	Acronym	Description	Address
-	-	-	-	Reserved	0x0000_0000
-	-3	fixed	Reset	Reset	0x0000_0004
-	-2	fixed	NMI_Handler	Non maskable interrupt. HXTAL CSS and LXTAL CSS interrupts are linked to the NMI vector.	0x0000_0008
-	-1	fixed	HardFault_Handler	All class of fault	0x0000_000C
-	-	-	Reserved	Reserved	0x0000_0010 0x0000_0014 ~ 0x0000_0028
-	3	settable	SVC_Handler	System service call via SVC instruction	0x0000_002C
-	-	-	Reserved	Reserved	0x0000_0030 0x0000_0034

Position	Priority	Type of priority	Acronym	Description	Address
-	5	settable	PendSV_Handler	Pendable request for system service	0x0000_0038
-	6	settable	SysTick_Handler	System tick timer	0x0000_003C
0	7	settable	WWDG	Window watchdog interrupt	0x0000_0040
1	8	settable	PVD	Power voltage detector interrupt	0x0000_0044
2	9	settable	RTC / TAMP	RTC and TAMP global interrupts	0x0000_0048
3	10	settable	FLASH	Flash global interrupt	0x0000_004C
4	11	settable	RCC	RCC global interrupt	0x0000_0050
5	12	settable	EXTI[1:0]	EXTI line 0 & 1 interrupt	0x0000_0054
6	13	settable	EXTI[3:2]	EXTI line 2 & 3 interrupt	0x0000_0058
7	14	settable	EXTI[15:4]	EXTI line 4 to 15 interrupt	0x0000_005C
8	15	settable	LCD	LCD global interrupt	0x0000_0060
9	16	settable	DMA_Channel0	DMA channel 0 interrupt	0x0000_0064
10	17	settable	DMA_Channel1	DMA channel 1 interrupt	0x0000_0068
11	18	-	Reserved	Reserved	0x0000_006C
12	19	settable	ADC / COMP	ADC interrupt COMP interrupt (EXTI 16 & 17)	0x0000_0070
13	20	-	Reserved	Reserved	0x0000_0074
14	21	-	Reserved	Reserved	0x0000_0078
15	22	settable	TIM3	TIM3 global interrupt	0x0000_007C
16	23	settable	TIM4	TIM4 global interrupt	0x0000_0080
17	24	settable	TIM5	TIM5 global interrupt	0x0000_0084
18	25	settable	TIM8	TIM8 global interrupt	0x0000_0088
19	26	settable	AES	AES global interrupt	0x0000_008C
20	27	settable	LPTIM1	LPTIM1 global interrupt	0x0000_0090
21	28	settable	I2C1	I2C1 global interrupt	0x0000_0094
22	29	-	Reserved	Reserved	0x0000_0098
23	30	settable	SPI1	SPI1 global interrupt	0x0000_009C
24	31	settable	SPI2	SPI2 global interrupt	0x0000_00A0
25	32	settable	USART1	USART1 global interrupt	0x0000_00A4
26	33	settable	UART2	UART2 global interrupt	0x0000_00A8
27	34	settable	UART3 / UART4	UART3 and 4 global interrupts	0x0000_00AC
28	35	settable	LPUART1	LPUART1 global interrupt	0x0000_00B0
29	36	settable	TRNG	TRNG global interrupt	0x0000_00B4
30	37	settable	LPTIM2	LPTIM2 global interrupt	0x0000_00B8
31	38	settable	LPUART2	LPUART2 global interrupt	0x0000_00BC

1. The grayed cells correspond to the Cortex-M0+ interrupts.

12 Extended interrupt and event controller (EXTI)

12.1 Introduction

The Extended interrupt and event controller (EXTI) manages the CPU and system wakeup through configurable and direct lines.

The EXTI also includes the EXTI I/O port mux, which allows any port to be configured as a wake-up source.

12.2 EXTI main features

- System wakeup upon event on any input
- Configurable lines (from I/Os, or peripherals not having an associated interrupt pending status bit)
  - Selectable active trigger edge
  - Individual interrupt and event generation mask, used for conditioning the CPU wakeup, interrupt and event generation
- Direct lines (from peripherals having an associated flag and interrupt pending status bit)
  - Individual interrupt and event generation mask for conditioning the CPU wakeup and event generation
- I/O port selector, and any port can wake up the system

12.3 EXTI block diagram

The inputs to EXTI are divided into two types: configurable lines and direct lines. The outputs are divided into event outputs and interrupt outputs.

The internal implementation of the EXTI is described in the *EXTI function description*.

Figure 12-1 EXTI block diagram

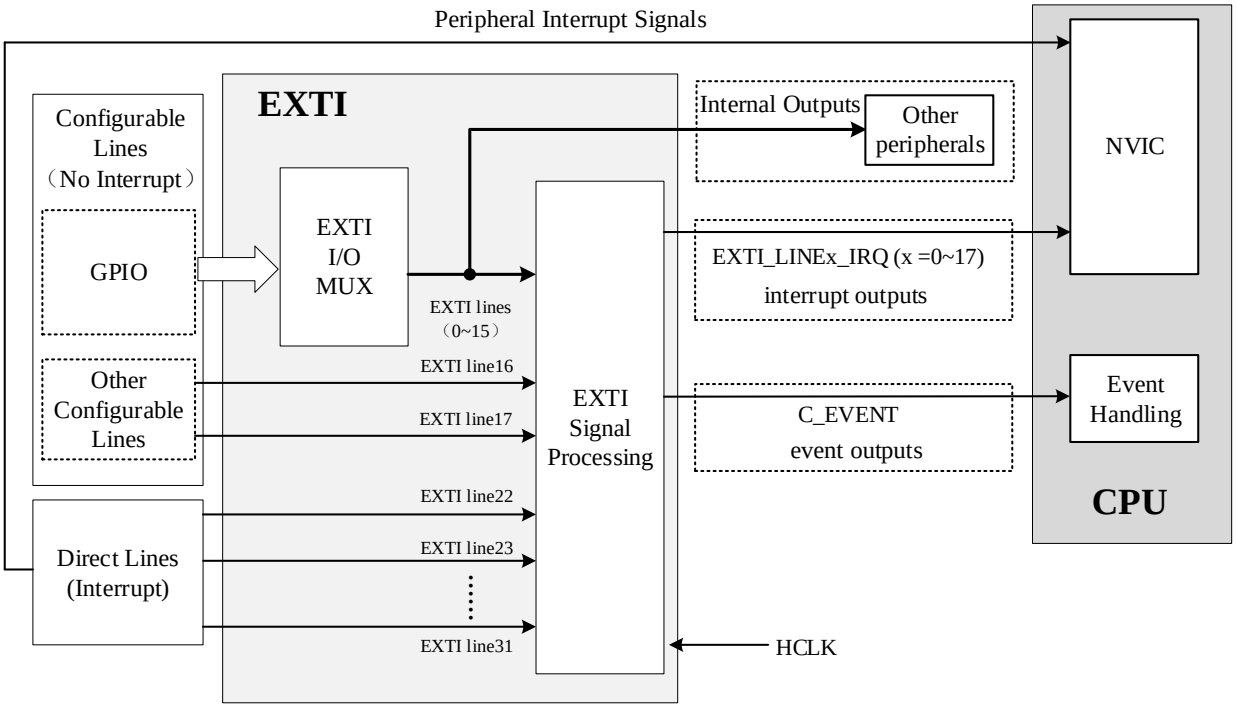


Table 12-1 EXTI internal signal

Signal name	I/O	Description
EXTI_LINEx_IRQ	Output	Interrupt signals generated by configurable lines (x = 0~17)
C_EVENT	Output	Event output signals to the CPU, synchronized with the CPU clock
HCLK	Input	EXTI clock, same as AHB bus clock

12.4 Main signal connections and functions of EXTI

EXTI provides interrupt pending flags for configurable lines. A pending flag set to 1 indicates that an interrupt is pending and needs to be cleared by software writing a "1" to the flag. The interrupt pending flags of EXTI can trigger CPU interrupts.

The EXTI module implements an I/O multiplexer. All GPIO ports input to the EXTI multiplexer, allowing to select a port to wake up the system via a configurable event. The selected I/O can wake up the CPU from low-power mode.

Peripherals that already implement wake-up events and interrupts serve as direct lines for EXTI.

The EXTI manages events from various channels and routes them to the CPU event handling module.

Events and interrupts generated by EXTI can wake up the CPU from low-power mode. The main ways to wake up the CPU are as follows:

- When using WFE (Wait for event) to enter low-power mode, the event signal input to the CPU event handling module can wake up the CPU from low-power mode, and the interrupt signal can also wake up the CPU from low-power mode.
- When using WFI (Wait for interrupt) to enter low-power mode, the interrupt signal can wake up the CPU from low-power mode.

For more details, refer to *Low-power modes*.

12.5 EXTI lines and signals table

Table 12-2 EXTI line connections

EXTI line	Line source	Type	Peripheral wake-up sources
0~15	GPIO	Configurable	GPIO input signals
16	COMP1	Configurable	COMP1 output
17	COMP2	Configurable	COMP2 output
22	LPUART2	Direct	When UEWK = 1, the following interrupt requests support wake-up when the interrupt is enabled: character match interrupt, receive data register not empty interrupt, and wakeup from low-power mode interrupt.
23	LPTIM2	Direct	The following interrupt requests support wake-up when the interrupt is enabled: auto-reload match interrupt.
24	PVD	Direct	PVD alarm event
25	RTC	Direct	RTC interrupt request
26	TAMP	Direct	TAMP interrupt request
27	I2C1	Direct	When WUPEN = 1, the address match event can be used for wake-up without enabling an interrupt.
29	LPUART1	Direct	When UEWK = 1, the following interrupt requests support wake-up when the interrupt is enabled: character match interrupt, receive data register not empty interrupt, and wakeup from low-power mode interrupt.
30	LPTIM1	Direct	The following interrupt requests support wake-up when the interrupt is enabled: compare match interrupt, auto-reload match interrupt, external trigger interrupt, counter direction change interrupt. When timeout enable (TIMEOUT = 1), the compare match event can be used for wake-up without enabling an interrupt.
31	LXTAL CSS	Direct	LXTAL clock failure interrupt request

12.6 EXTI functional description

The enable of configurable lines is controlled by EXTI, and the corresponding lines are enabled by configuring the EXTI_RTSR register or the EXTI_FTSR register. The enable of direct lines is controlled in the peripheral. The wake-up with interrupt mask register (EXTI_IMR) or the wake-up with event mask register (EXTI_EMR) determines whether an enabled channel can wake up the system. For more details, see the table below:

Table 12-3 EXTI masking functionality

Configuration of wake-up mask registers		Effect of wake-up mask register configuration		
Wakeup with interrupt mask register IMR.IMn	**Wakeup with event mask register EMR.EMn**	**Configurable line pending register PIR.PIFn**	**Interrupt output**	**Event output**
0	0	No	Masked	Masked
	1	No	Masked	Yes
1	0	Yes	Yes	Masked
	1	Yes	Yes	Yes

Unmasked events (EXTI_EMR.EMn = 1) or unmasked interrupts (EXTI_IMR.IMn = 1) generated by configurable lines and direct lines can be used as wake-up sources to wake up the CPU.

12.6.1 Configurable line

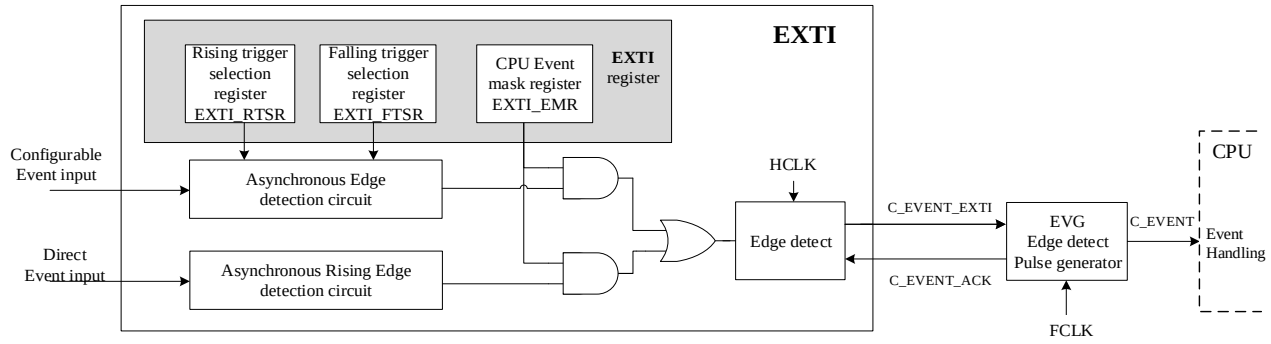
The trigger edge for configurable events can be selected as rising edge, falling edge, or both edges through the EXTI_RTSR and EXTI_FTSR registers.

Event output of configurable lines

The masking of event outputs from configurable lines is managed by the EXTI_EMR register, and the results are then routed to the CPU event handling module, which triggers the event response.

For more details, see the diagram below:

Figure 12-2 Event output block diagram for configurable lines and direct lines

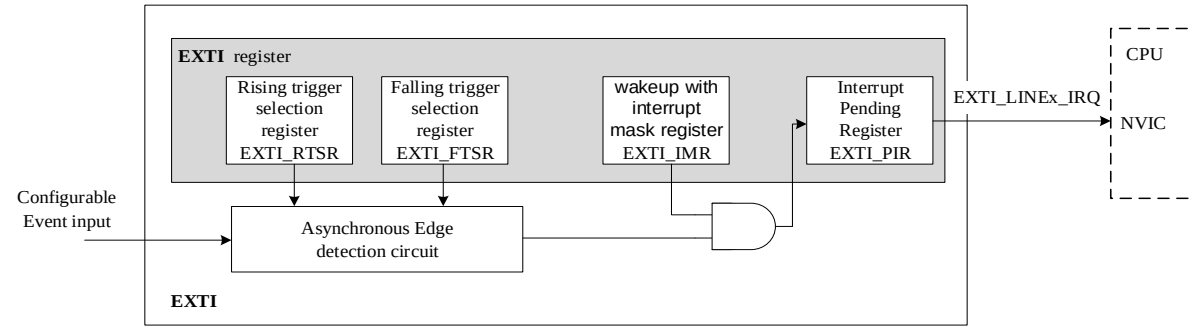


Interrupt output of configurable lines

When a configurable line detects a trigger edge, if the interrupt is not masked (EXTI_IMR.IMn = 1), the corresponding PIFn bit in the interrupt pending register (EXTI_PIR) is set, generating an interrupt request signal to the NVIC. This wakes up the CPU and triggers a CPU interrupt. The PIFn bit must be cleared by writing a “1” to it in software to clear the interrupt request.

The interrupt pending register (EXTI_PIR) only responds to unmasked (IMn = 1) interrupt requests from configurable lines. This is independent of the configuration of the EXTI_EMR register. For more details, see the diagram below:

Figure 12-3 Interrupt output block diagram for configurable lines



12.6.2 Direct line

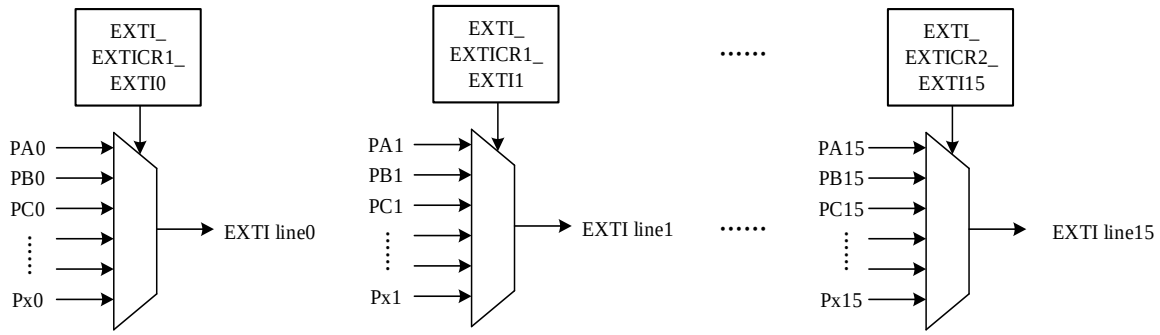
Events or interrupts from direct lines can wake up the system. The ability of events or interrupts to wake up the system is set through the EXTI_EMR register and EXTI_IMR register.

Interrupts from direct lines are implemented in the peripheral and are independent of EXTI. The logic for the event output of direct lines see: *Figure Event output block diagram for configurable lines and direct lines*.

12.6.3 EXTI GPIO mux

The EXTI mux allows selecting GPIOs as interrupts and wakeup. The GPIOs are connected via 16 EXTI mux lines to the first 16 EXTI events as configurable event. The selection of GPIO port as EXTI mux output is controlled through the EXTI_EXTICR1 register and EXTI_EXTICR2 register.

Figure 12-4 EXTI GPIO mux



The EXTIs mux outputs are available as output signals from the EXTI, to trigger other functional blocks. The EXTI mux outputs are available independently of mask setting through the EXTI_IMR register and EXTI_EMR register. For more details, refer to the relevant peripheral module sections: *ADC pins and internal signals*.

12.7 EXTI registers

The EXTI registers can only be accessed by words (32-bit).

Table 12-4 EXTI base address

Peripheral	Base address
EXTI	0x4002 1800

12.7.1 EXTI rising trigger selection register (EXTI_RTSR)

Address offset: 0x00

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.														RT17	RT16
														rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RT15	RT14	RT13	RT12	RT11	RT10	RT9	RT8	RT7	RT6	RT5	RT4	RT3	RT2	RT1	RT0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Name	Description
31:18	Reserved	Must be kept at reset value
17:0	RTx	Rising trigger event configuration bit of configurable line (x = 0~17) Each bit enables/disables the rising edge trigger for the event and interrupt on the corresponding line. 0: Disable 1: Enable

12.7.2 EXTI falling trigger selection register (EXTI_FTSR)

Address offset: 0x04

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.														FT17	FT16
														rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FT15	FT14	FT13	FT12	FT11	FT10	FT9	FT8	FT7	FT6	FT5	FT4	FT3	FT2	FT1	FT0

rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Name	Description
31:18	Reserved	Must be kept at reset value
17:0	FTx	Falling trigger event configuration bit of configurable line (x = 0~17) Each bit enables/disables the falling edge trigger for the event and interrupt on the corresponding line. 0: Disable 1: Enable

12.7.3 EXTI interrupt pending register (EXTI_PIR)

Address offset: 0x08

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.														PIF17	PIF16
														rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIF15	PIF14	PIF13	PIF12	PIF11	PIF10	PIF9	PIF8	PIF7	PIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0
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

Bits	Name	Description
31:18	Reserved	Must be kept at reset value
17:0	PIF	When a trigger interrupt is detected and pending on EXTI configurable line x (x = 0~17), it indicates that the interrupt is pending. Each bit is set to 1 when a rising or falling edge interrupt is detected on the corresponding line. Each bit is cleared by writing 1 into it. 0: No interrupt pending request detected 1: Interrupt request is pending

12.7.4 EXTI external interrupt selection register 1 (EXTI_EXTICR1)

Address offset: 0x50

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	EXTI_7[2:0]			Res.	EXTI_6[2:0]			Res.	EXTI_5[2:0]			Res.	EXTI_4[2:0]		
	rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	EXTI_3[2:0]			Res.	EXTI_2[2:0]			Res.	EXTI_1[2:0]			Res.	EXTI_0[2:0]		
	rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw

Bits	Name	Description
31	Reserved	Must be kept at reset value
30:28	EXTI_7[2:0]	EXTI_7 GPIO port selection These bits are written by software to select the source input for EXTI_7 external interrupt. 000: PA7 pin 001: PB7 pin 010: PC7 pin Others: Reserved
27	Reserved	Must be kept at reset value
26:24	EXTI_6[2:0]	EXTI_6 GPIO port selection These bits are written by software to select the source input for EXTI_6 external interrupt. 000: PA6 pin 001: PB6 pin 010: PC6 pin Others: Reserved
23	Reserved	Must be kept at reset value
22:20	EXTI_5[2:0]	EXTI_5 GPIO port selection These bits are written by software to select the source input for EXTI_5 external interrupt. 000: PA5 pin 001: PB5 pin 010: PC5 pin

		011: PD5 pin Others: Reserved
19	Reserved	Must be kept at reset value
18:16	EXTI_4[2:0]	EXTI_4 GPIO port selection These bits are written by software to select the source input for EXTI_4 external interrupt. 000: PA4 pin 001: PB4 pin 010: PC4 pin 011: PD4 pin Others: Reserved
15	Reserved	Must be kept at reset value
14:12	EXTI_3[2:0]	EXTI_3 GPIO port selection These bits are written by software to select the source input for EXTI_3 external interrupt. 000: PA3 pin 001: PB3 pin 010: PC3 pin 011: PD3 pin Others: Reserved
11	Reserved	Must be kept at reset value
10:8	EXTI_2[2:0]	EXTI_2 GPIO port selection These bits are written by software to select the source input for EXTI_2 external interrupt. 000: PA2 pin 001: PB2 pin 010: PC2 pin 011: PD2 pin 101: PF2 pin Others: Reserved

7	Reserved	Must be kept at reset value
6:4	EXTI_1[2:0]	EXTI_1 GPIO port selection These bits are written by software to select the source input for EXTI_1 external interrupt. 000: PA1 pin 001: PB1 pin 010: PC1 pin 011: PD1 pin 101: PF1 pin Others: Reserved
3	Reserved	Must be kept at reset value
2:0	EXTI_0[2:0]	EXTI_0 GPIO port selection These bits are written by software to select the source input for EXTI_0 external interrupt. 000: PA0 pin 001: PB0 pin 010: PC0 pin 011: PD0 pin 101: PF0 pin Others: Reserved

12.7.5 EXTI external interrupt selection register 2 (EXTI_EXTICR2)

Address offset: 0x54

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	EXTI_15[2:0]			Res.	EXTI_14[2:0]			Res.	EXTI_13[2:0]			Res.	EXTI_12[2:0]		
	rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	EXTI_11[2:0]			Res.	EXTI_10[2:0]			Res.	EXTI_9[2:0]			Res.	EXTI_8[2:0]		
	rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw

Bits	Name	Description
31	Reserved	Must be kept at reset value

30:28	EXTI_15[2:0]	EXTI_15 GPIO port selection These bits are written by software to select the source input for EXTI_15 external interrupt. 000: PA15 pin 001: PB15 pin 010: PC15 pin Others: Reserved
27	Reserved	Must be kept at reset value
26:24	EXTI_14[2:0]	EXTI_14 GPIO port selection These bits are written by software to select the source input for EXTI_14 external interrupt. 000: PA14 pin 001: PB14 pin 010: PC14 pin Others: Reserved
23	Reserved	Must be kept at reset value
22:20	EXTI_13[2:0]	EXTI_13 GPIO port selection These bits are written by software to select the source input for EXTI_13 external interrupt. 000: PA13 pin 001: PB13 pin 010: PC13 pin Others: Reserved
19	Reserved	Must be kept at reset value
18:16	EXTI_12[2:0]	EXTI_12 GPIO port selection These bits are written by software to select the source input for EXTI_12 external interrupt. 000: PA12 pin 001: PB12 pin 010: PC12 pin

		Others: Reserved
15	Reserved	Must be kept at reset value
14:12	EXTI_11[2:0]	EXTI_11 GPIO port selection These bits are written by software to select the source input for EXTI_11 external interrupt. 000: PA11 pin 001: PB11 pin 010: PC11 pin Others: Reserved
10:8	EXTI_10[2:0]	EXTI_10 GPIO port selection These bits are written by software to select the source input for EXTI_10 external interrupt. 000: PA10 pin 001: PB10 pin 010: PC10 pin Others: Reserved
7	Reserved	Must be kept at reset value
6:4	EXTI_9[2:0]	EXTI_9 GPIO port selection These bits are written by software to select the source input for EXTI_9 external interrupt. 000: PA9 pin 001: PB9 pin 010: PC9 pin Others: Reserved
3	Reserved	Must be kept at reset value
2:0	EXTI_8[2:0]	EXTI_8 GPIO port selection These bits are written by software to select the source input for EXTI_8 external interrupt. 000: PA8 pin 001: PB8 pin

010: PC8 pin

Others: Reserved

12.7.6 EXTI wakeup with interrupt mask register (EXTI_IMR)

Address offset: 0x70

Reset value: 0xEFC0 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IM31	IM30	IM29	Res.	IM27	IM26	IM25	IM24	IM23	IM22	Res.				IM17	IM16
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
IM15	IM14	IM13	IM12	IM11	IM10	IM9	IM8	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	Rw	rw	rw	rw	rw

Bits	Name	Description
31:29	IMx	CPU wakeup with interrupt mask on line x (x = 29~31) Setting/clearing each bit unmasks/masks the CPU wakeup with interrupt. 0: Disable 1: Enable
28	Reserved	Must be kept at reset value
27:22	IMx	CPU wakeup with interrupt mask on line x (x = 22~27) Setting/clearing each bit unmasks/masks the CPU wakeup with interrupt. 0: Disable 1: Enable
21:18	Reserved	Must be kept at reset value
17:0	IMx	CPU wakeup with interrupt mask on line x (x = 0~17) Setting/clearing each bit unmasks/masks the CPU wakeup with interrupt. 0: Disable 1: Enable

12.7.7 EXTI wakeup with event mask register (EXTI_EMR)

Address offset: 0x74

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EM31	EM30	EM29	Res.	EM27	EM26	EM25	EM24	EM23	EM22	Res.				EM17	EM16
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
EM15	EM14	EM13	EM12	EM11	EM10	EM9	EM8	EM7	EM6	EM5	EM4	EM3	EM2	EM1	EM0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	Rw	rw	rw	rw	rw

Bits	Name	Description
31:29	EMx	CPU wakeup with event generation mask on line x (x = 29~31) Setting/clearing each bit unmasks/masks the CPU wakeup with event generation on the corresponding line. 0: Disable 1: Enable
28	Reserved	Must be kept at reset value
27:22	EMx	CPU wakeup with event generation mask on line x (x = 22~27) Setting/clearing each bit unmasks/masks the CPU wakeup with event generation on the corresponding line. 0: Disable 1: Enable
21:18	Reserved	Must be kept at reset value
17:0	EMx	CPU wakeup with event generation mask on line x (x = 0~17) Setting/clearing each bit unmasks/masks the CPU wakeup with event generation on the corresponding line. 0: Disable 1: Enable

13 Cyclic redundancy check calculation unit (CRC)

13.1 Introduction

The CRC (cyclic redundancy check) calculation unit is used to get a CRC code of an input data from initial value and a generator polynomial. CRC-based techniques are used to verify data transmission or storage integrity.

13.2 CRC main features

- Supports CRC-16 and CRC-32 polynomials
  - Uses CRC-16 polynomial: 0x1021

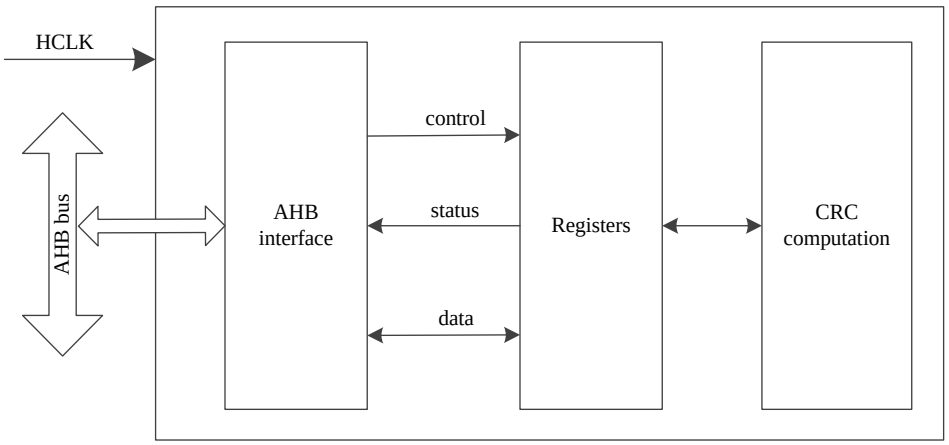
    $X^{16}+X^{12}+X^{5}+1$
  - Uses CRC-32 polynomial: 0x4C11DB7

    $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^{8}+X^{7}+X^{5}+X^{4}+X^{2}+X+1$
- Supports calculation mode and check mode
- CRC computation done in 1 AHB clock cycle for the 8-bit data size

13.3 CRC functional description

13.3.1 CRC block diagram

Figure 13-1 CRC calculation unit block diagram



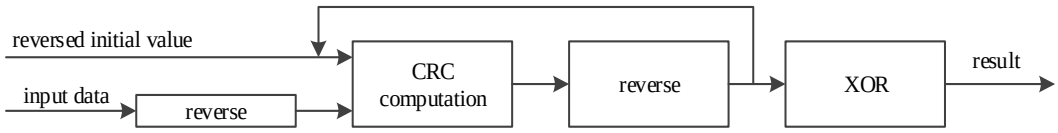
13.3.2 CRC operation

The CRC data register (CRC_DR) is used to input new data. The CRC result data

register (CRC_RDR) holds the CRC initial value and the result of the CRC calculation. The CRC initial value needs to be bitwise reversed when being written to the CRC_RDR register. Refer to *Output data reverse block diagram*.

Each write operation to the CRC_DR register creates a combination of the initial value of the CRC_RDR register (or the value after reversing the data generated from the previous calculation) after reversing and the new data in the CRC_DR register. The data generated by the calculation is reversed and then XORed with 0xFFFFFFFF (or 0xFFFF). The result of the XOR operation is output to the CRC_RDR register. Refer to the diagram below.

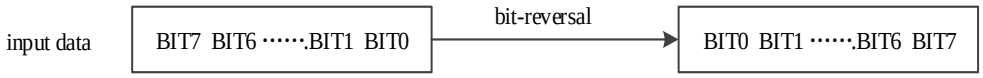
Figure 13-2 CRC calculation unit block diagram



Input data is sequentially input into the CRC_DR register in 8-bit data format. The 8-bit data calculation can be completed in 1 AHB clock cycle, the next data can be immediately written to the CRC_DR register without any wait time.

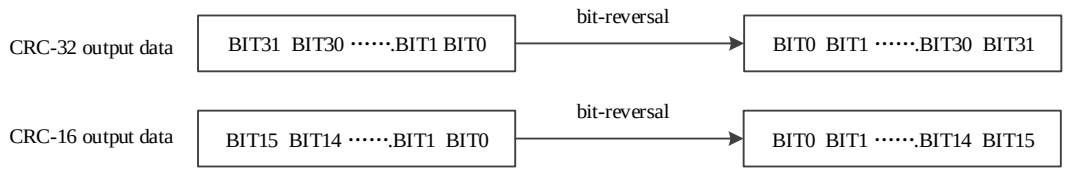
The input data should be reversed. For example: input data 0x11, 0x22, 0x33, and 0x44 are converted into 0x88, 0x44, 0xCC, and 0x22. Refer to the diagram below.

Figure 13-3 Input data reverse block diagram



CRC output data is bit-reversed and then XORed with 0xFFFFFFFF (or 0xFFFF). For example: CRC-32 output data 0x11223344, after bit-reversal the data becomes 0x22CC4488, and after XORing the data becomes 0xDD33BB77. Refer to the diagram below.

Figure 13-4 Output data reverse block diagram



CRC supports calculation mode and check mode. CRC calculation mode involves

inputting the original data into the CRC to obtain the CRC calculation result. CRC check mode involves inputting the original data to be verified and the CRC value of the original data to be verified into the CRC, to check whether the original data matches with the CRC value. The operation steps for both modes are as follows:

- Calculation mode

  1) Set polynomial size through the POLY_SIZE bit in the CRC_CSR register. 0 corresponds to CRC-16 polynomial size, 1 corresponds to CRC-32 polynomial size.
  2) Reverse the CRC initial value and write to the CRC_RDR register.
  3) Set the data to the CRC_DR register sequentially in 8-bit width.
  4) Read the result of the CRC_RDR register.

- Check mode

  1) Set polynomial size through the POLY_SIZE bit in the CRC_CSR register. 0 corresponds to CRC-16 polynomial size, 1 corresponds to CRC-32 polynomial size.
  2) Reverse the CRC initial value and write to the CRC_RDR register.
  3) Write the original data to be verified to the CRC_DR register sequentially in 8-bit width.
  4) Write the CRC value of the original data to be verified into the CRC_DR register in the order from the least significant byte to the most significant byte.
  5) Read the CHK_FLAG bit in the CRC_CSR register to determine whether the CRC check is correct. 1 indicates the current CRC check is correct. Otherwise, the current CRC check is incorrect.

13.4 CRC registers

The CRC registers can only be accessed by words (32-bit).

Table 13-1 CRC base address

Peripheral	Base address
CRC	0x4002 3000

13.4.1 CRC control status register (CRC_CSR)

Address offset: 0x00

Reset value: 0x0000 0001

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

Bits	Name	Description
31:2	Reserved	Must be kept at reset value
1	CHK_FLAG	CRC check result 0: Check failed 1: Check passed
0	POLY_SIZE	Polynomial size 0: 16 bit polynomial 1: 32 bit polynomial

13.4.2 CRC result data register (CRC_RDR)

Address offset: 0x04

Reset value: 0x0000 0000

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

RESULT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Name	Description
31:0	RESULT[31:0]	CRC calculation result data register bits Read RESULT[15:0] for CRC-16 calculation result. Read RESULT[31:0] for CRC-32 calculation result. Write reversed initial value to RESULT[15:0] for CRC-16 initializing. Write reversed initial value to RESULT[31:0] for CRC-32 initializing.

13.4.3 CRC data register (CRC_DR)

Address offset: 0x80

Reset value: 0x0000 0000

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

Bits	Name	Description
31:8	Reserved	Must be kept at reset value
7:0	DATA[7:0]	Data register bits This register is used to write new data to the CRC calculator.

14 Analog-to-digital converter (ADC)

14.1 Introduction

The 12-bit ADC is a successive approximation analog-to-digital converter (SAR ADC). It has up to 18 multiplexed channels allowing it to measure signals from 15 external and 3 internal sources. The conversion mode of channel sequence can be set to single scan, continuous scan, discontinuous mode. The result of the ADC is stored in a 12-bit data register.

The analog watchdog feature allows the application to detect if the input voltage goes outside the user-defined higher or lower thresholds.

14.2 ADC main features

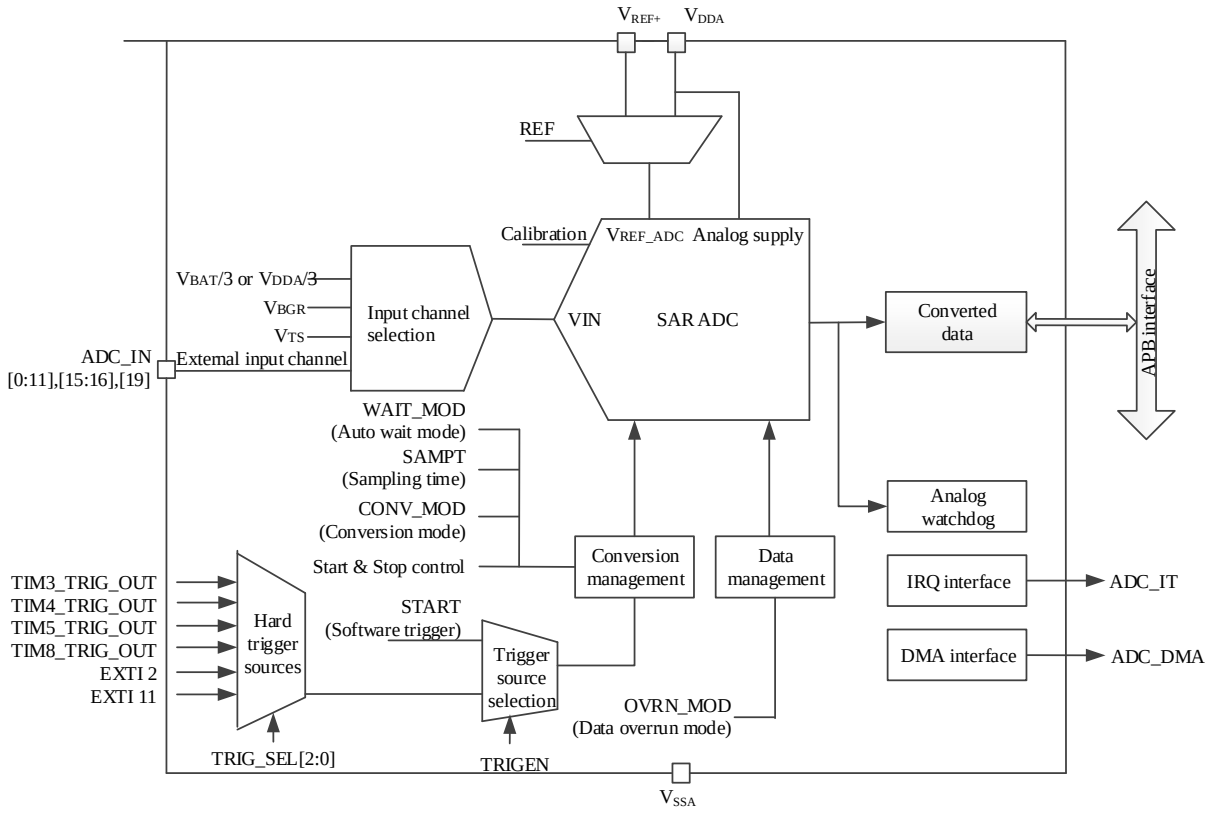
- ADC supply requirements: 1.8 V ~ 5.5 V
- Reference voltage: V_{REF_ADC}
  - V_{DDA}
  - V_{REF+} reference voltage pin input
  - VREFBUF: 2.048 V/2.5 V/3.0 V
- ADC input range: 0 ~ V_{REF_ADC}
- Up to 1 Msps for 12-bit resolution
- Analog input channels
  - 15 external channels inputs, support high impedance signals
  - 3 internal channels inputs: V_{TS}, V_{BGR}, $V_{DDA}/3$ or $V_{BAT}/3$
- Conversion modes
  - Single scan mode
  - Continuous scan mode
  - Discontinuous mode
- Start-of-conversion can be initiated:
  - By software

- By hardware triggers with configurable polarity (timer events or GPIO input events)

- Programmable sampling time
- Auto wait mode
- DMA support
- Analog watchdog

14.3 ADC functional description

Figure 14-1 ADC block diagram



14.3.1 ADC pins and internal signals

Table 14-1 ADC input pins

Name	Signal type	Remarks
V_{DDA}	Input, analog power supply	Analog power supply for the ADC
	Input, analog reference	The reference voltage for the ADC
V_{SSA}	Input, analog supply ground	Ground for analog power supply
V_{REF+}	Input, analog reference	The reference voltage for the ADC

Name	Signal type	Remarks
ADC_INx	Analog input signals	15 external analog input channels

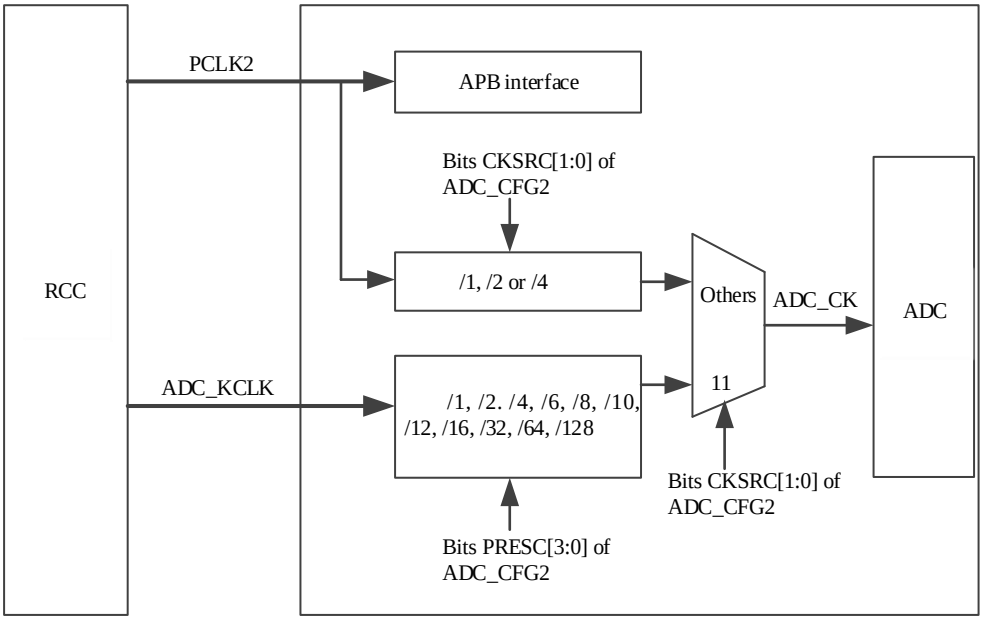
Table 14-2 ADC internal input signals

Name	Signal type	Description
TIMx_TRIG_OUT, EXTI channel 2/11	Input	ADC conversion triggers
V_{TS}	Input	Internal temperature sensor output voltage
V_{BGR}	Input	Internal BGR output voltage
$V_{DDA}/3$	Input	V_{DDA} pin input voltage divided by 3
$V_{BAT}/3$	Input	V_{BAT} pin input voltage divided by 3

14.3.2 ADC clock

The ADC employs a dual clock-domain architecture, where the clock source can be selected from either the APB2 clock (PCLK2) or the ADC asynchronous clock (ADC_KCLK). The ADC clock architecture is as shown in the figure below:

Figure 14-2 ADC clock scheme



- The ADC asynchronous clock (ADC_KCLK) is selected by setting the CKSRC[1:0] bit field of the ADC_CFG2 register to '11'. The clock operates independently from PCLK2, and its frequency can be divided by a prescaler (1, 2, 4, 6, 8, 10, 12, 16, 32, 64, 128) when programming PRESC[3:0] bits in the ADC_CFG2 register.

- The ADC clock can be derived from the APB clock (PCLK2) of the ADC bus interface, divided by a programmable factor (1, 2 or 4) according to bits CKSRC[1:0] of the ADC_CFG2 register. The PCLK2 is selected concurrently with the configuration of the prescaler factor.

The advantage of selecting the ADC_KCLK is that it reduces the PCLK2 frequency for chip-level low-power operation, while allowing ADC_KCLK to operate at its maximum supported frequency to maintain optimal ADC performance.

The advantage of selecting the PCLK2 lies in the elimination of cross-clock domain synchronization requirements. In contrast, when the ADC_KCLK is selected, a resynchronization delay exists between trigger events and ADC sampling instants due to cross-clock domain transitions. This delay exhibits a variable range dependent on the current frequency and phase relationship between PCLK2 and ADC_KCLK.

Caution: The ADC clock (ADC_CK) requires a 50% duty cycle (typical value, with a permissible range of 40% to 60%). Consequently, when the CKSRC[1:0] bit field is set to '00' to select the undivided PCLK2 as the ADC clock source, user must ensure that PCLK2 maintains a 50% duty cycle. This necessitates the use of a system clock with a 50% duty cycle and disabling any frequency division of the system clock by the AHB and APB prescalers.

The relationship between the ADC clock frequency and the operating voltage is defined as follows:

- When 2.4 V < V_{DDA} ≤ 5.5 V, 600 KHz ≤ f_{ADC_CK} ≤ 16 MHz
- When 1.8 V ≤ V_{DDA} ≤ 2.4 V, 600 KHz ≤ f_{ADC_CK} ≤ 8 MHz

14.3.3 On/off control

ADEN and ADDIS bits in the ADC_CR register are used to enable or disable the ADC.

Follow this procedure to enable the ADC:

1) Set ADEN = 1 to enable the ADC.
2) Wait a stabilization time before ADC starts, the specific time is as follows:
   - When ADC_CK ≥ 6 MHz, stabilization time is 2.5μs.
   - When ADC_CK < 6 MHz, stabilization time is seventeen ADC_CK cycles.

Follow this procedure to disable the ADC:

1) Check START = 0 in the ADC_CR register to ensure that no conversion is ongoing.
   If START = 1, stop any ongoing conversion by writing '1' to STOP bit in the ADC_CR register and waiting until this bit is read at '0'.
2) Set ADDIS = 1 in the ADC_CR register.
3) Wait until ADEN = 0, indicating that the ADC is fully disabled.

14.3.4 Calibration

ADC calibration function removes the offset error which may vary from chip to chip due to process variation.

Calibration should be performed before starting A/D conversion. Enable ADC and wait a stabilization time, set CALEN in the ADC_CR register to start a calibration. During the procedure, the CALEN is kept to '1'. It is cleared by hardware when calibration is complete, then EOCAL flag is set.

After calibration is complete, the calibration factor can be read from the ADC_CALFACT register or ADC_DR register. This calibration factor will continue to be applied to the ADC and retained even if the ADC is disabled (ADEN = 0). The calibration factor will be cleared only in the following scenarios:

- The power supply is removed from the ADC (for example when the product is power-off or enters V_{BAT} mode)
- The ADC peripheral reset or system reset occurs

The calibration factor can be written to the ADC_CALFACT register when the ADC is enabled but not actively converting. Upon the next conversion initiation, the calibration factor will be automatically loaded into the ADC. The loading process of the calibration factor will not cause any delay to the start of the conversion.

The application can store the calibration factor (e.g., in Flash memory) and load the previously saved value into the ADC_CALFACT register when restarting the ADC.

When the ADC operating conditions change (V_{DDA} changes are the main contributor to ADC offset variations and temperature change to a lesser extend), it is recommended to re-run a calibration cycle.

ADC calibration procedure:

1) Set ADEN = 1, wait a stabilization time.
2) Set CALEN = 1 to start calibration.

3) Wait until EOCAL = 1. If the interrupt is enabled by setting the EOCALIE bit in the ADC_IER register, an interrupt can be generated.
4) Read the calibration factor from DATA[5:0] bits in the ADC_DR register or ADC_CALFACT register.

The procedure of loading the calibration factor by software:

1) Check ADEN = 1, and START = 0.
2) Write to the ADC_CALFACT register with the saved calibration factor.
3) The calibration factor is used as soon as a new conversion is launched.

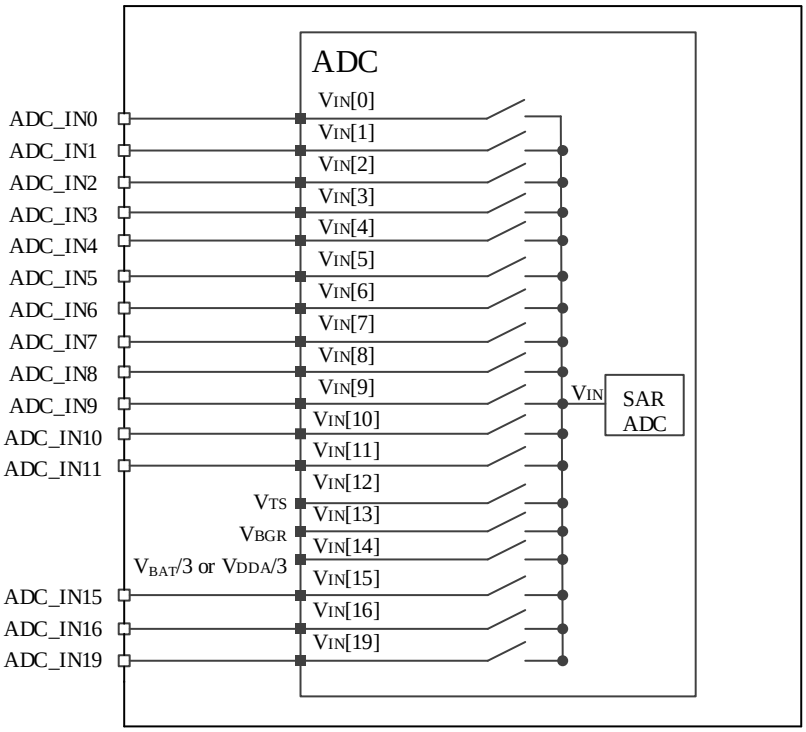
14.3.5 Conversion channel

There are up to 18 multiplexed channels:

- 15 external analog inputs from GPIO pins (ADC_INx)
- 3 internal analog inputs (V_{BGR}, V_{TS}, $V_{BAT}/3$ or $V_{DDA}/3$):
  - The temperature sensor is connected to channel ADC_IN[12].
  - The internal voltage reference V_{BGR} is connected to channel ADC_IN[13].
  - $V_{BAT}/3$ or $V_{DDA}/3$ channel is connected to ADC_IN[14].

The internal source ADC_IN[12] refer to *Figure: Temperature sensor channel block diagram*, the internal source ADC_IN[13] refer to *Figure: V_{BGR} channel block diagram*, the internal source ADC_IN[14] refer to *Figure: V_{BAT} and V_{DDA} channel block diagram*.

Figure 14-3 ADC input channels



Channel selection

ADC can convert a single channel or a sequence of channels. The sequence of the channels to be converted can be programmed in the ADC_CHCFG register.

- Sequence length: configured through CHNx bits.
- Sequence direction: configured through SDIR bit, SDIR = 0: forward scan, from the lowest to the highest channel number. SDIR = 1: backward scan, from the highest to the lowest channel number.

Example: Channel 0, 3, 7, 10 are selected to convert:

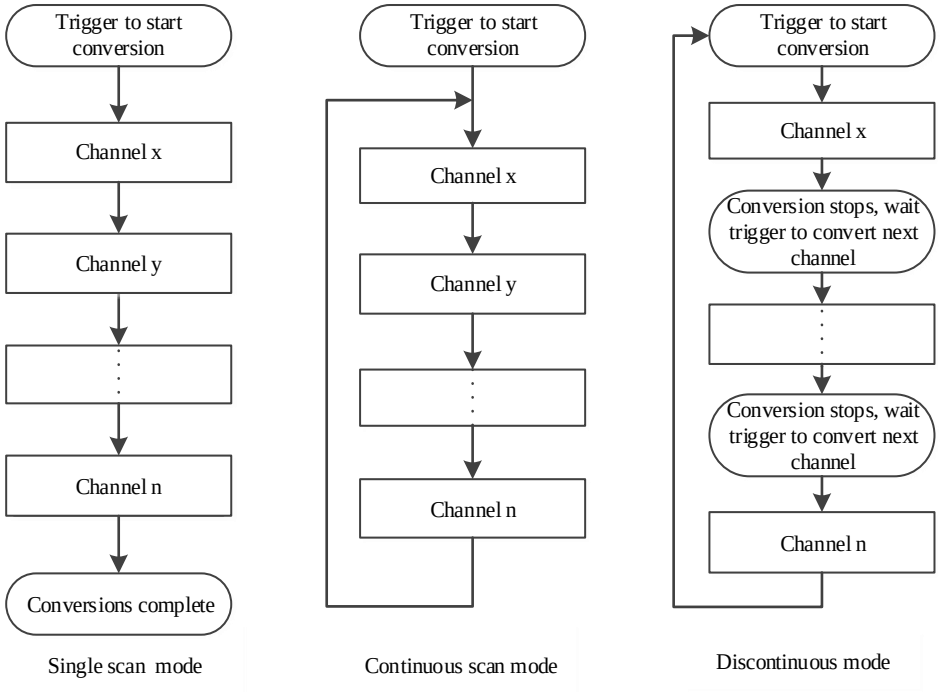
- Set CHN0, CHN3, CHN7, CHN10 bits in the ADC_CHCFG register to enable channel 0, 3, 7, 10. According to configuration of SDIR bit, sequence direction is channel 0, 3, 7, 10 (SDIR = 0) or channel 10, 7, 3,0 (SDIR = 1).

Write operations to the SDIR bit in the ADC_CFG1 register and the CHCFG_MOD bit in the ADC_CHCFG register can only be performed when START = 0.

14.3.6 Conversion modes

Single scan mode, continuous scan mode, and discontinuous mode can be selected to perform a conversion or a sequence of conversions.

Figure 14-4 ADC conversion modes



Single scan mode

In this conversion mode, the ADC performs a single sequence of conversions, converting the configured channels once when valid soft or hard trigger event occurs. After each conversion is complete, the ADC stops until a new external trigger event occurs or the START bit is set again. This mode is selected when CONV_MOD[1:0] bits in the ADC_CFG1 register is configured to '00'.

Conversion is started by either:

- Software trigger, setting the START bit in the ADC_CR register, conversion starts immediately.
- Hardware trigger, after setting the START bit in the ADC_CR register, then waiting until hard trigger event occurs, conversion starts immediately.

Inside the sequence, after each conversion is complete:

- The converted data are stored in the 12-bit ADC_DR register.
- The EOC (end of conversion) flag is set.
- An interrupt is generated if the EOCIE bit is set.

After the sequence of conversions is complete:

- The EOS (end of sequence) flag is set.
- An interrupt is generated if the EOSIE bit is set.

Then the ADC stops until a new external trigger event occurs or the START bit is set again.

Continuous scan mode

In this conversion mode, ADC performs a sequence of conversions, converting all the configured channels once and then automatically re-starts and continuously performs the same sequence of conversions. This mode is selected when CONV_MOD[1:0] bits in the ADC_CFG1 register is configured to '01'.

Conversion is started by either:

- Software trigger, setting the START bit in the ADC_CR register, conversion starts immediately.
- Hardware trigger, after setting the START bit in the ADC_CR register, then waiting until hard trigger event occurs, conversion starts immediately.

Inside the sequence, after each conversion is complete:

- The converted data are stored in the 12-bit ADC_DR register.
- The EOC (end of conversion) flag is set.
- An interrupt is generated if the EOCIE bit is set.

After the sequence of conversions is complete:

- The EOS (end of sequence) flag is set.
- An interrupt is generated if the EOSIE bit is set.

Discontinuous mode

In this mode, a hardware or software trigger event is required to start each conversion defined in the sequence. After the conversion of current channel is complete, the ADC stops until a new external trigger event occurs or the START bit is set again, then ADC starts the conversion of next channel. After the sequence of conversions is complete, ADC performs the same conversion sequence. This mode is selected when CONV_MOD[1:0] bits in the ADC_CFG1 register is configured to '10'.

Example: Channel 0, 3, 7, 10 are selected to convert:

- 1st trigger: Channel 0 is converted, the EOC flag is set when conversion is complete.
- 2nd trigger: Channel 3 is converted, the EOC flag is set when conversion is complete.
- 3rd trigger: Channel 7 is converted, the EOC flag is set when conversion is complete.
- 4th trigger: Channel 10 is converted, both EOC and EOS flag are set when conversion is complete.
- 5th trigger: Channel 0 is converted, the EOC flag is set when conversion is complete.
- 6^{th} trigger: Channel 3 is converted, the EOC flag is set when conversion is complete.
- ...

14.3.7 Start/Stop conversions

Starting conversions

A conversion or a sequence of conversion can be triggered either by software or by an external event:

- Software trigger is selected by writing TRIGEN[1:0] bits in the ADC_CFG1 register to '00', a conversion or a sequence of conversion starts after setting START bit in the ADC_CR register to '1'.
- Hardware trigger is selected by writing TRIGEN[1:0] bits in the ADC_CFG1 register not equal to '00', once software has set START to '1', then external events are able to trigger a conversion or a sequence of conversion with the selected polarity.

Any triggers which occur while a conversion is ongoing are ignored. If START = 0, any hardware triggers which occur are ignored.

The software is allowed to change trigger source and polarity only when START bit is '0'. The correspondence between TRIGEN[1:0] bits values and the trigger polarity is

shown in the following table.

Table 14-3 Trigger source and polarity

Trigger source and polarity	TRIGEN[1:0]
Hardware trigger detection disabled (conversion is started by software)	00
Hardware trigger detection on the rising edge	01
Hardware trigger detection on the falling edge	10
Hardware trigger detection on both the rising and falling edge	11

External hardware trigger sources can be selected by setting TRIG_SEL[2:0] bits in the ADC_CFG1 register. The software is allowed to change external hardware trigger sources only when START bit is '0'.

Table 14-4 External hardware trigger sources

Name	Source	TRIG_SEL[2:0]
TRG0	Reserved	000
TRG1	Reserved	001
TRG2	TIM3_TRIG_OUT	010
TRG3	TIM4_TRIG_OUT	011
TRG4	TIM5_TRIG_OUT	100
TRG5	TIM8_TRIG_OUT	101
TRG6	EXTI 2	110
TRG7	EXTI 11	110

The START bit is also used to indicate whether an ADC operation is currently ongoing. It is possible to re-configure the ADC while START = 0, indicating that the ADC is idle.

In the following situations, the START bit is cleared by hardware:

- In all cases (CONV_MOD[1:0] = XX, TRIGEN[1:0] = XX), after execution of the STOP procedure invoked by software.
- According to the configuration of conversion mode and trigger source, the START bit is cleared by hardware at different timing.
  - In single scan mode with software trigger (CONV_MOD[1:0] = 00, TRIGEN[1:0] = 00), at the end of conversion sequence (EOS = 1).

- In discontinuous mode with software trigger (CONV_MOD[1:0] = 10, TRIGEN[1:0] = 00), at the end of conversion (EOC = 1).

In continuous scan mode (CONV_MOD[1:0] = 01), the START bit is not cleared by hardware when the EOS flag is set, because the sequence is automatically relaunched.

In single scan mode (CONV_MOD[1:0] = 00) and discontinuous mode (CONV_MOD[1:0] = 10), when hardware trigger is selected (TRIGEN[1:0] ≠ 01), START bit is not cleared by hardware when the EOS flag is set. This avoids the need for software having to set the START bit again and ensures the next trigger event is not missed.

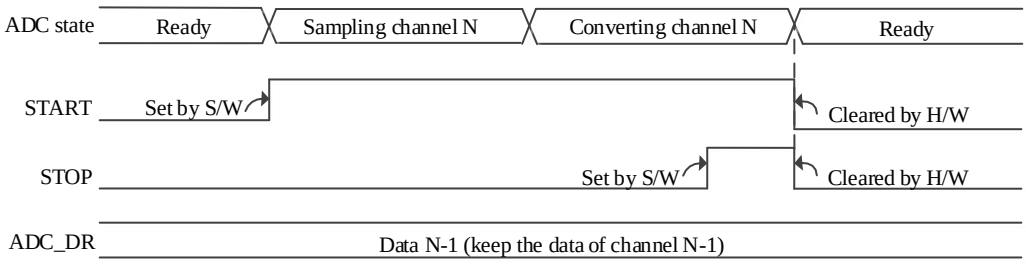
Stopping conversions

Any ongoing conversion can be aborted by setting the STOP bit in the ADC_CR register to '1'. This action resets the ADC module and places it in idle mode, preparing the system for subsequent operations.

When the STOP bit is set by software, any ongoing conversion is aborted and the result is discarded (the ADC_DR register is not updated with the current conversion). The scan sequence is also aborted and reset (meaning that restarting the ADC would re-start a new sequence).

Once this procedure is complete, the STOP and START bits are both cleared by hardware, and the software must wait until START = 0 before starting new conversions.

Figure 14-5 Stopping an ongoing conversion



End of conversion, end of sampling phase, end of conversion sequence

The EOC flag in the ADC_ISR register is set when each end of conversion event occurs, and a new conversion data result is available in the ADC_DR register. An interrupt can be generated if the EOCIE bit is set in the ADC_IER register. The EOC flag is cleared by software either by writing '1' to it, or by reading the ADC_DR register.

The EOSAMP flag in the ADC_ISR register is set when the end of sampling phase

occurs. An interrupt can be generated if the EOSAMPIE bit is set in the ADC_IER register. The EOSAMP flag is cleared by software by writing ‘1’ to it.

The EOS flag in the ADC_ISR register is set when each end of sequence event occurs, and the last data result of a conversion sequence is available in the ADC_DR register. An interrupt can be generated if the EOSIE bit is set in the ADC_IER register. The EOS flag is cleared by software by writing ‘1’ to it.

14.3.8 Example of conversion timing

Channel 0, 3, 7, 10 are configured to be converted.

Figure 14-6 Single scan mode, software trigger

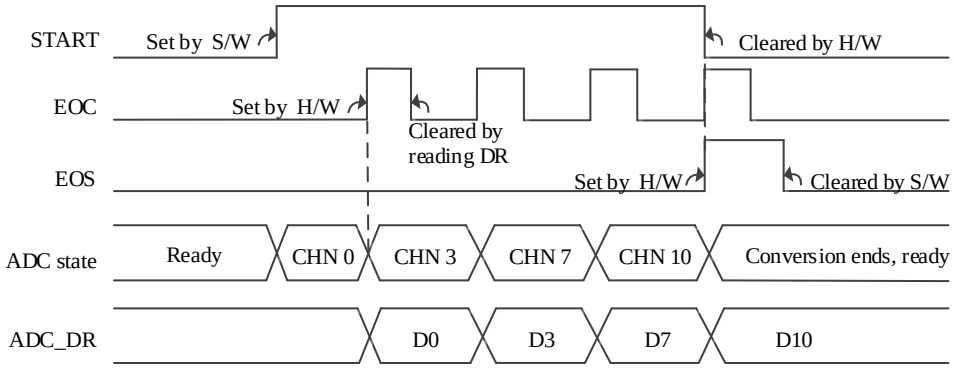


Figure 14-7 Continuous scan mode, software trigger

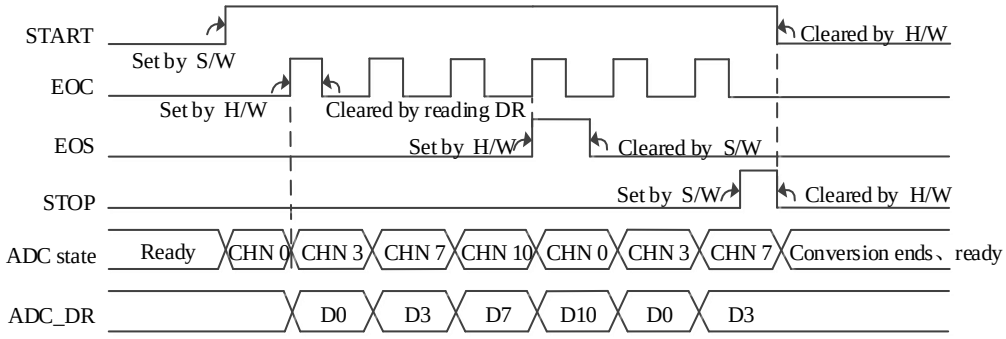


Figure 14-8 Single scan mode, hardware trigger

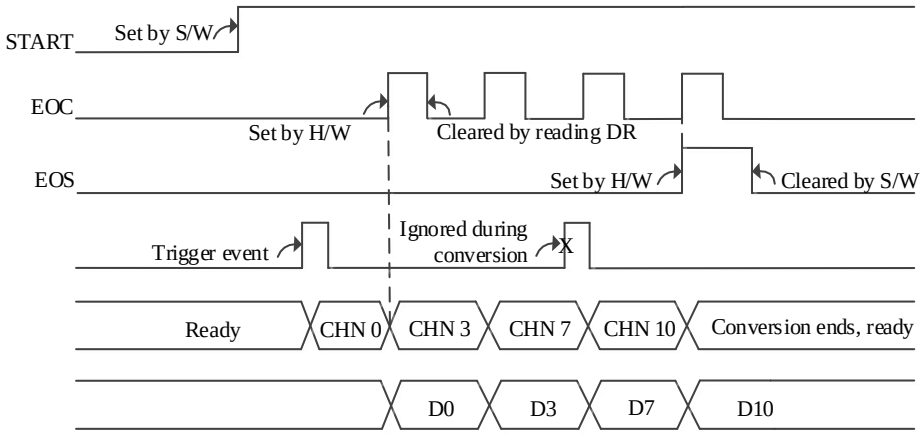
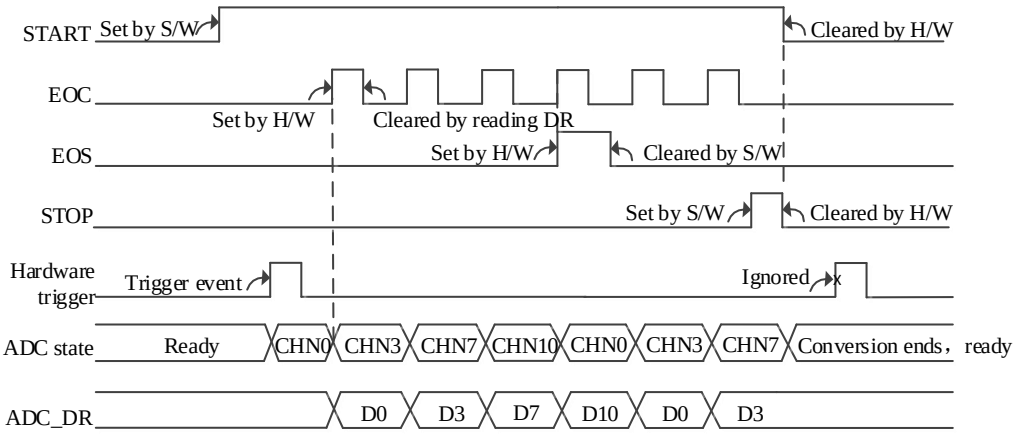


Figure 14-9 Continuous scan mode, hardware trigger



14.3.9 Low frequency trigger mode

After ADC is stabled or last conversion is complete, ADC is ready to start a new conversion. It is necessary to start conversion within time of t_{IDLE}, otherwise corrupted convert data may be obtained. (refer to datasheet for the value of t_{IDLE})

Low frequency trigger mode is used to reset the internal status of ADC, when the time interval between ADC ready and the beginning of conversion exceeds the max value of t_{IDLE}. Whenever software trigger or hardware trigger occurs, the internal status of ADC is reset by hardware automatically.

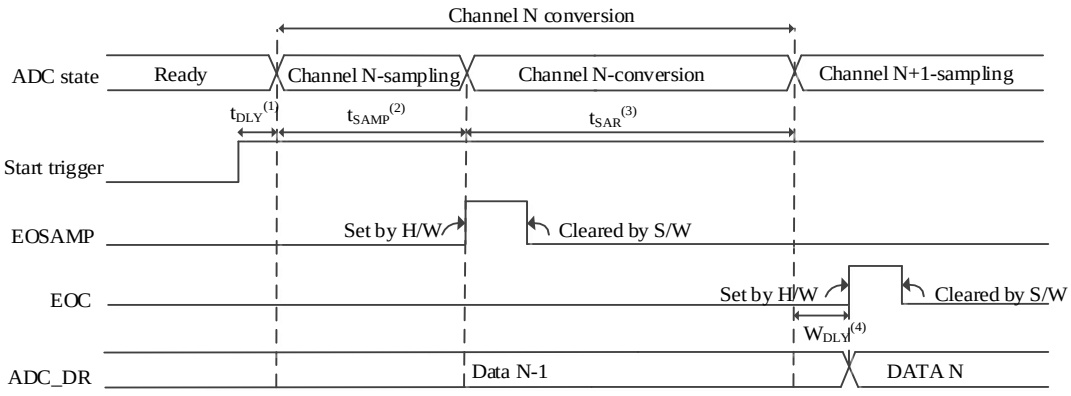
Enabling the low frequency trigger mode by setting the LFTRG bit in the ADC_CFG2 register, is useful to meet the needs of application which the max value of t_{IDLE} is exceeded.

14.3.10 Programmable sampling time

The ADC analog-to-digital conversion process consists of two phases: voltage sampling and successive approximation register (SAR) voltage calculation. Therefore, the total ADC conversion time (i.e., the duration from conversion start to completion) is the sum of the configured sampling time (t_{SAMP}) and the SAR conversion time (t_{SAR}).

The sampling time (t_{SAMP}) is user-configurable based on input signal characteristics, while the SAR conversion time (t_{SAR}) is determined by the ADC resolution.

Figure 14-10 ADC analog to digital conversion timing



1. t_{DLY}: trigger latency (refer to datasheet for more details)
2. t_{SAMP}: sampling time, configured by the ADC_SAMPT register.
3. t_{SAR}: the successive approximation time
4. W_{DLY}: the ADC_DR register write latency (refer to datasheet for more details)

Before starting a conversion, the ADC needs to establish a direct connection between the voltage source to be measured and the embedded sampling capacitor of the ADC. This sampling time must be enough for the input voltage source to charge the sample and hold capacitor to the input voltage level. Having a programmable sampling time allows the conversion speed to be trimmed according to the input resistance of the input voltage source.

The ADC samples the input voltage for a number of ADC clock cycles that can be configured via SAMPT1[3:0] and SAMPT2[3:0] bits in the ADC_SAMPT register.

Each channel can choose one out of two sampling times configured in SAMPT1[3:0] and SAMPT2[3:0] bits, through SAMPT_SELx bits in the ADC_SAMPT register.

The total conversion time is calculated as follows:

$t_{CONV} = t_{SAMP} + t_{SAR} = t_{SAMP} + 13 \times t_{ADC\_CK}$

Note: t_{SAR} always equals to 13 ADC_CK clock cycles.

Example: With ADC_CK = 16 MHz and sampling time of 3 ADC_CK clock cycles:

$t_{CONV} = 3 + 13 = 16\ t_{ADC\_CK} = 1\ \mu s$

14.3.11 Data overrun

A data overrun event occurs when the converted data was not read in time by the CPU or the DMA, before the data from a new conversion is available.

When a new conversion completes, the OVRN flag is set in the ADC_ISR register if the EOC flag is still at '1' at the time. An interrupt can be generated if the OVRNIE bit is set in the ADC_IER register. The OVRN flag is cleared by software by writing '1' to it.

When a data overrun occurs, the ADC keeps operating and can continue to convert unless the software stops and resets the conversion sequence by setting the STOP bit in the ADC_CR register.

Programming the OVRN_MOD bit in the ADC_CFGR1 register to select if the data is preserved or overwritten when a data overrun event occurs:

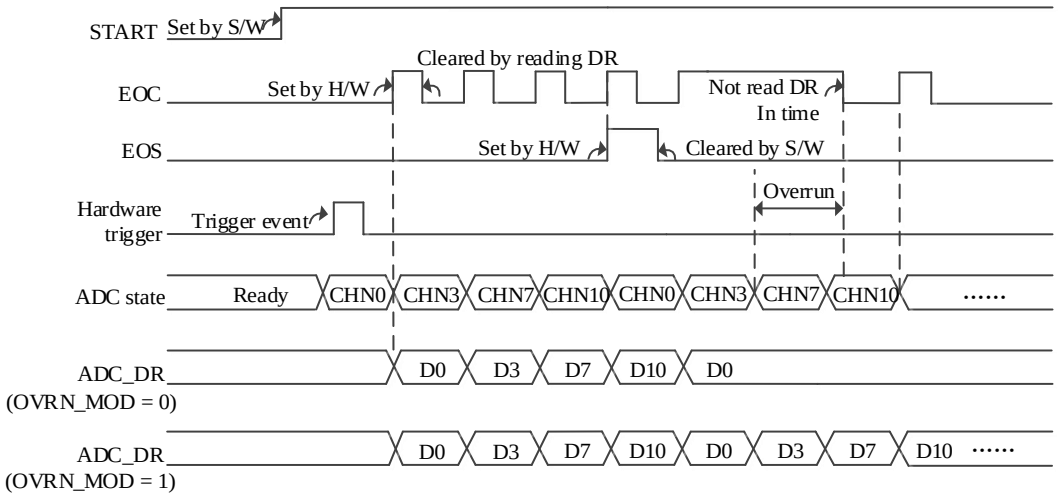
- OVRN_MOD = 0

  A data overrun event preserves the data register from being overwritten, the old data is maintained and the new conversion is discarded. If OVRN remains at '1', further conversions can be performed but the resulting data is discarded.

- OVRN_MOD = 1

  A data overrun event overwrites the data register with the last conversion result and the previous unread data is lost. If OVRN remains at '1', further conversions can be performed and the ADC_DR register always contains the data from the latest conversion.

Figure 14-11 Data overrun timing(1)



1. Channel 0, 3, 7, 10 are converted, upwards scan, continuous scan mode, hardware trigger.

14.3.12 Data management

Managing converted data without using the DMA

If the conversions are slow enough, the conversion sequence can be handled by software, there is no need to use DMA. In this case the software must use the EOC flag and EOC interrupt to handle each data result. Each time a conversion is complete, the EOC bit is set in the ADC_ISR register and ADC_DR register can be read. The OVRN_MOD bit in the ADC_CFGR1 register should be configured to '0' to manage overrun events as an error.

In the case of letting ADC convert one or more channels without reading the data after each conversion, the OVRN_MOD bit should be configured at '1' and the OVRN flag should be ignored by the software. Then an overrun event does not prevent the ADC from continuing to convert and the ADC_DR register always contains the latest conversion data.

Managing converted data using the DMA

Since all converted channel data are stored in a single data register, it is efficient to use DMA when converting more than one channel, and avoids losing the conversion data stored in the ADC_DR register.

When DMA request is enabled by setting the DMAEN bit in the ADC_CFG1 register, a DMA request is generated after each conversion, DMA transfers the converted data

from the ADC_DR register to the destination location set by the software.

Despite this, if an overrun occurs when OVRN_MOD = 1 because the DMA could not serve the DMA transfer request in time, the ADC stops generating DMA requests and the data corresponding to the new conversion is not transferred by the DMA. All the data transferred to the destination (for example SRAM) can be considered as invalid.

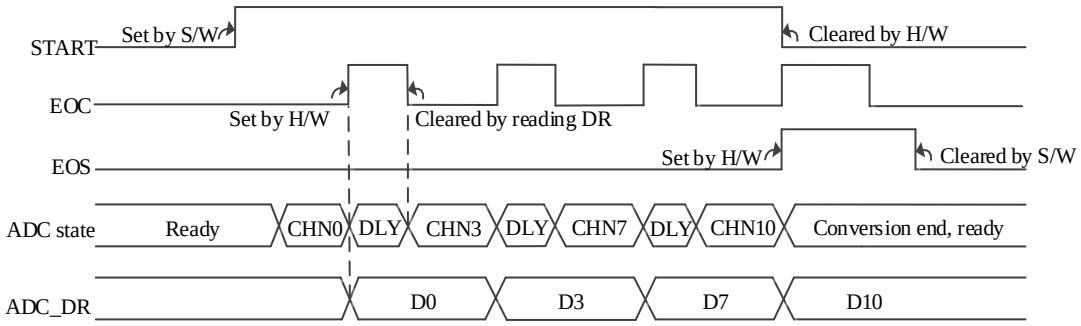
14.3.13 Auto wait mode

Auto wait mode can be used to simplify the software as well as optimizing the performance of applications clocked at low frequency where there might be a risk of ADC data overrun occurring. This is a way to automatically adapt the speed of the ADC to the speed of the system that reads the data.

When auto wait mode is enabled by setting WAIT_MOD bit in the ADC_CFG1 register, a new conversion can start only if the previous data has been treated, once the ADC_DR register has been read or if the EOC bit has been cleared by the software.

Caution: Any hardware triggers which occur while a conversion is ongoing or during the wait time preceding the read access are ignored.

Figure 14-12 Auto wait mode(1)



1. Channel 0, 3, 7, 10 are converted, upwards scan, single scan mode, software trigger.

14.3.14 Analog watchdog

Introduction

The ADC analog watchdog is used to monitor whether the input voltage of a channel is within the configured threshold range.

By setting the CHNx (x = 0~16, 19) bit in the ADC_AWDG1CR register, the analog watchdog monitors the input voltage of the selected conversion channel. When the

CHNx (x = 0~16, 19) bit is set to '1', the corresponding analog watchdog monitoring channel is enabled.

The analog watchdog monitors the input voltage by comparing the complete 12-bit raw conversion data against the configured thresholds. The upper and lower monitoring voltage thresholds are set in the AWDG_HT[11:0] and AWDG_LT[11:0] bit fields of the ADC_AWDG1TR register, respectively.

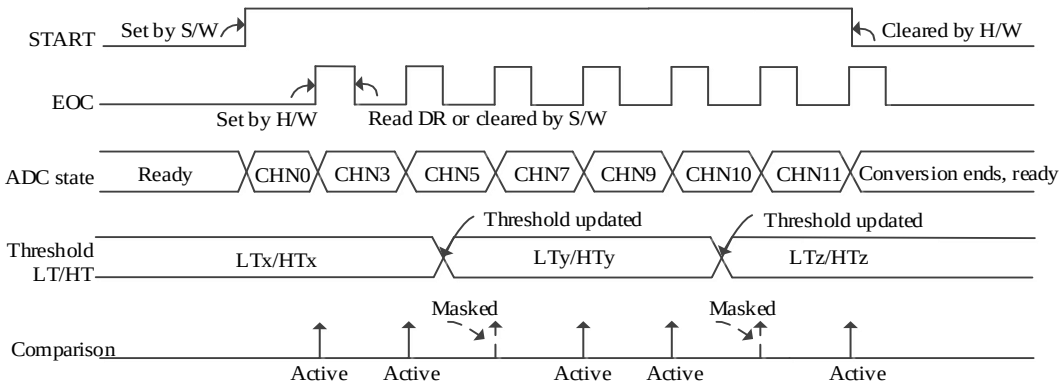
The AWDG1 flag is set if the analog voltage converted by the ADC is below a lower threshold or above a higher threshold. An interrupt can be generated when AWDG1IE bit in the ADC_IER register has set to '1'. The AWDG1 flag is cleared by software by writing '1' to it.

Analog watchdog thresholds control

AWDG_LT[11:0] and AWDG_HT[11:0] can be changed during an analog-to-digital conversion.

If thresholds are programmed during the ADC guarded channel conversion, the watchdog function is masked for this conversion. This mask is cleared when starting a new conversion, and the resulting new AWDG threshold is applied starting the next ADC conversion result.

Figure 14-13 Analog watchdog threshold update(1)



1. Channel 0, 3, 5, 7, 9, 10, 11 are converted, upwards scan, single scan mode, software trigger. Input channel selected to be monitored: 0, 3, 5, 7, 9, 10, 11.

14.3.15 Temperature sensor

The temperature sensor equipped in device can be used to measure the junction temperature (T_J) of the device. It is internally connected to the ADC_VIN[12] input channel to convert the sensor's output voltage to a digital value.

The temperature sensor outputs voltage changes linearly with temperature. The offset of this line varies from chip to chip due to process variation. To improve the accuracy of the temperature sensor measurement, calibration values are stored in flash memory for each device during production, accessible in read-only mode. The calibration conditions and memory address are shown in the following table:

Table 14-5 Temperature sensor calibration values

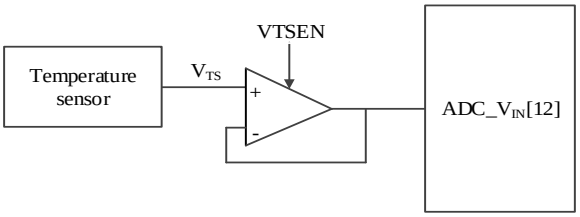
Calibration value name	Calibration conditions	Memory address
TS_{CAL_25}	Temperature: 25 °C (± 2 °C), $V_{DDA} = V_{REF+} = 3.3$ V	0x1FFF03C4
TS_{CAL_85}	Temperature: 85 °C (±2 °C), $V_{DDA} = V_{REF+} = 3.3$ V	0x1FFF03C8

Where:

- TS_{CAL_25} is the temperature sensor calibration value which is the actual output value converted by ADC at 25 °C.
- TS_{CAL_85} is the temperature sensor calibration value which is the actual output value converted by ADC at 85 °C.

The user can enable conversion of ADC_VIN[12] (V_{TS}) by setting the VTSEN bit in the ADC_CFG2 register. The sampling time of temperature sensor needs to exceed the value of t_{SAMP} (refer to the datasheet for detail). To prevent any unwanted consumption on the temperature sensor, it is recommended to disable the temperature sensor when not needed for ADC conversion by clearing the VTSEN bit.

Figure 14-14 Temperature sensor channel block diagram



Before using the temperature sensor, it is essential to enable BGR and wait for its stabilization time, refer to *Internal bandgap reference voltage*.

Reading the temperature by ADC conversion:

1) Select the ADC_VIN[12] input channel.
2) Select an appropriate sampling time t_{SAMP}.
3) Set the VTSEN bit in the ADC_ CFG2 register to wake up the temperature sensor

from power down state and wait for its stabilization time.

4) Start the ADC conversion by setting the START bit or by hardware trigger.

5) Wait the end of conversion, read the resulting V_{TS} data (TS_{DATA}) in the ADC_DR register.

6) Calculate the temperature using the following formula:

$$\text{Temperature}(℃) = \frac{85℃-25℃}{TS_{CAL\_85}-TS_{CAL\_25}} \times (TS_{DATA} - TS_{CAL\_25}) + 25℃$$

TS_{DATA} is the actual output value converted by ADC at current temperature.

14.3.16 Internal bandgap reference voltage detection

The embedded internal bandgap reference voltage (V_{BGR}) is internally connected to ADC_VIN[13] input channel, can provide a stable voltage output for the ADC.V_{BGR} of each device varies due to process variation, the calibration is performed and calibration values are stored in flash memory for each device during production, accessible in read-only mode. V_{BGR} calibration value can be used to evaluate the actual V_{REF_ADC} voltage level. The calibration conditions and memory address are shown in the following table.

Before using V_{BGR}, it is essential to enable BGR and wait for its stabilization time, refer to *Internal bandgap reference voltage*.

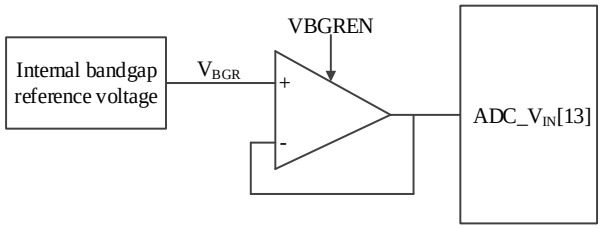
Table 14-6 BGR calibration value

Calibration value name	Calibration conditions	Memory address
BGR_CAL	Temperature: 25 °C (± 2 °C), V_{DDA} = V_{REF_ADC} = 3.3 V	0x1FFF03C0

BGR_CAL is the actual V_{BGR} output value converted by ADC at V_{DDA} = V_{REF_ADC} = 3.3 V.

The user can enable conversion of ADC_VIN[13] (V_{BGR}) by setting the VBGREN bit in the ADC_CFG2 register.

Figure 14-15 V_{BGR} channel block diagram



Calculating the actual ADC VREF+ voltage using V_{BGR}

The V_{REF_ADC} voltage may be subject to variation or not precisely known. The embedded internal voltage reference (V_{BGR}) and its calibration data acquired by the ADC during the manufacturing process at V_{REF_ADC} = 3 V can be used to evaluate the actual V_{REF_ADC} voltage level.

The following formula gives the actual V_{REF_ADC} voltage supplying the device:

$$V_{REF\_ADC} = 3.3\ V \times V_{BGR\_CAL}/V_{BGR\_DATA}$$

Where:

- V_{BGR_CAL} is calibration value of V_{BGR}.
- V_{BGR_DATA} is the actual V_{BGR} output value converted by ADC at current V_{REF_ADC}.

Converting a supply-relative ADC measurement to an absolute voltage value

The ADC is designed to deliver a digital value corresponding to the ratio between the analog power supply V_{REF_ADC} and the voltage applied on the converted channel.

For applications where V_{REF_ADC} is known, the user can use the following formula to get this absolute value of the voltage input from converted channel:

$$V_{CHANNELX} = \frac{V_{REF\_ADC}}{FULL\_SCALE} \times ADC\_DATAx$$

For applications where V_{REF_ADC} value is not known, the user can use the internal reference voltage V_{BGR} to calculate the actual V_{REF_ADC} value, V_{REF_ADC} can be replaced by the expression of V_{BGR}, resulting in the following formula:

$$V_{CHANNELX} = \frac{3.3\ V \times V_{BGR\_CAL} \times ADC\_DATAx}{V_{BGR\_DATA} \times FULL\_SCALE}$$

Where:

- V_{BGR_CAL} is calibration value of V_{BGR}.
- ADC_DATAx is the value measured by the ADC on channel-x at current V_{REF_ADC}.
- V_{BGR_DATA} is the actual V_{BGR} output value converted by the ADC at current V_{REF_ADC}.
- FULL_SCALE is maximum digital value of ADC output. For example, with 12-

bit resolution, it is 2^{12} - 1 = 4095.

14.3.17 V_{BAT} and V_{DDA} voltage monitoring

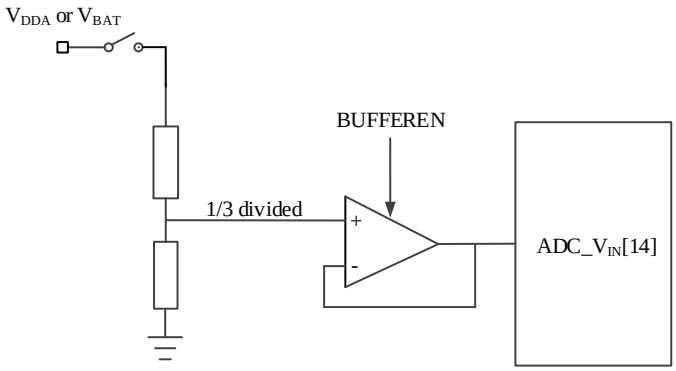
ADC can be used to measure the voltage of V_{BAT} and V_{DDA} pin, the V_{BAT} pin or V_{DDA} pin is internally connected to a bridge divider, which is automatically enabled when the VBAT_DIV3 bit or the VDDA_DIV3 bit is set, to connect VBAT/3 or VDDA/3 to the ADC_V_{IN}[14] input channel.

Setting the VBAT_DIV3 bit or the VDDA_DIV3 bit in the ADC_CFG2 register enables V_{BAT} input or V_{DDA} input. Due to the high output impedance of the bridge divider resistor string, the internal buffer is automatically enabled by the hardware to enhance the driving for ADC input. In order to reduce the power consumption of the bridge divider, it is recommended to disable V_{BAT} input or V_{DDA} input by writing '0' to the VBAT_DIV3 and the VDDA_DIV3, when V_{BAT} and V_{DDA} voltage monitoring are not needed.

V_{BAT} input and V_{DDA} input use the same internal channel of ADC, the user needs to follow this procedure to enable only one input simultaneously:

1) Clear the VBAT_DIV3 and the VDDA_DIV3 bits.
2) Set the VBAT_DIV3 or the VDDA_DIV3 bit, to enable V_{BAT} input or V_{DDA} input exclusively.

Figure 14-16 V_{BAT} and V_{DDA} channel block diagram



14.3.18 ADC interrupts

An interrupt can be generated by any of the following events:

- End of calibration (EOCAL flag)
- End of sampling phase (EOSAMP flag)

- End of any conversion (EOC flag)
- End of a sequence of conversions (EOS flag)
- When an analog watchdog detection occurs (AWDG1 flag)
- When a data overrun occurs (OVRN flag)

Table 14-7 ADC interrupts

Interrupt event	Event flag	Enable bit
End of calibration	EOCAL	EOCALIE
End of sampling phase	EOSAMP	EOSAMPIE
End of any conversion	EOC	EOCIE
End of a sequence of conversions	EOS	EOSIE
Analog watchdog detection	AWDG1	AWDG1IE
Data overrun	OVRN	OVRNIE

14.4 ADC registers

The ADC registers can only be accessed by words (32-bit).

Table 14-8 ADC base address

Peripheral	Base address
ADC	0x4001 2400

14.4.1 ADC control register (ADC_CR)

Address offset: 0x00

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.											STOP	Res.	START	ADDIS	ADEN
											rs		rs	rs	rs

Bits	Name	Description
31	CALEN	ADC calibration enable Set by software to start the calibration of the ADC, and cleared by hardware after calibration is complete. 0: Calibration complete or not in calibration status 1: Write 1 to start calibration. Read at 1 means that a calibration is in progress. *Note: The software is allowed to set CALEN only when the ADC is ready to start conversion (wait a stabilization time after setting ADEN = 1), and START = 0, STOP = 0, ADDIS = 0.*
30:5	Reserved	Must be kept at reset value
4	STOP	Stop ADC conversion Set by software to stop and discard an ongoing conversion, and cleared by hardware when the conversion is effectively discarded and the ADC is ready to accept a new start conversion command. 0: No ADC stop conversion command ongoing 1: Write 1 to stop the ADC. Read 1 means that a STOP command is

		in progress. *Note: Setting STOP to 1 is only effective when START = 1 and ADDIS = 0.*
3	Reserved	Must be kept at reset value
2	START	Start ADC conversion Set by software to start ADC conversion. Depending on the configuration of TRIGEN[1:0] bits in ADC_CFG1 register, a conversion either starts immediately (software trigger configured) or once a hardware trigger event occurs (hardware trigger configured). It is cleared by hardware or software, refer to *Start/Stop conversions* for detail. 0: No ADC conversion is ongoing 1: Write 1 to start the ADC. Read 1 means that the ADC is converting. *Note: The software is allowed to set START only when ADEN = 1 and ADDIS = 0.*
1	ADDIS	Disable ADC Set by software to disable the ADC and put it into power-down state, and cleared by hardware once the ADC is effectively disabled (ADEN is also cleared by hardware at this time). 0: No ADDIS command ongoing 1: Write 1 to disable the ADC. Read 1 means that an ADDIS command is in progress. *Note: Setting ADDIS to 1 is only effective when ADEN = 1 and START = 0.*
0	ADEN	Enable ADC Set by software to enable the ADC, and cleared by hardware when the ADC is disabled, after setting ADDIS = 1. The software has to wait a stabilization time until the ADC is effectively ready to operate after setting ADEN = 1. 0: ADC is disabled 1: Write 1 to enable the ADC *Note: The software is allowed to set ADEN only when all bits of*

ADC_CR registers are 0 (CALEN = 0, STOP = 0, START = 0, ADDIS = 0 and ADEN = 0).

14.4.2 ADC configuration register 1 (ADC_CFG1)

Address offset: 0x04

Reset value: 0x0000 0000

Note: The software is allowed to write this register only when START bit is 0.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.													TRIG_SEL[2:0]		
													rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	WAIT_MOD	CONV_MOD[1:0]		OVRN_MOD	TRIGEN[1:0]		Res.						SDIR	Res.	DMAEN
	rw	rw	rw	rw	rw	rw							rw		rw

Bits	Name	Description
31:19	Reserved	Must be kept at reset value
18:16	TRIG_SEL[2:0]	External trigger selection 010: TIM3_TRIG_OUT 011: TIM4_TRIG_OUT 100: TIM5_TRIG_OUT 101: TIM8_TRIG_OUT 110: EXTI channel 2 111: EXTI channel 11 Others: Reserved (TIM3_TRIG_OUT)
15	Reserved	Must be kept at reset value
14	WAIT_MOD	Auto wait mode enable 0: Disable 1: Enable
13:12	CONV_MOD[1:0]	Conversion mode selection 00: Single scan mode 01: Continuous scan mode 10: Discontinuous mode

		11: Reserved (Single scan mode)
11	OVRN_MOD	Data overrun management mode 0: When an overrun is detected, the ADC_DR register is preserved with the old data. 1: When an overrun is detected, the ADC_DR register is overwritten with the last conversion result.
10:9	TRIGEN[1:0]	External trigger enable and polarity selection Select the external trigger polarity and enable the trigger. 00: Hardware trigger detection disabled (The software trigger is enabled to start the conversion) 01: Hardware trigger detection on the rising edge 10: Hardware trigger detection on the falling edge 11: Hardware trigger detection on both the rising and falling edges
8:3	Reserved	Must be kept at reset value
2	SDIR	Scan sequence direction 0: Upward scan (from CHN0 to CHN19) 1: Backward scan (from CHN19 to CHN0)
1	Reserved	Must be kept at reset value
0	DMAEN	DMA request enable 0: Disable 1: Enable

14.4.3 ADC configuration register 2 (ADC_CFG2)

Address offset: 0x08

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CKSRC[1:0]		Res.		PRESC[3:0]				Res.				VDDA_DIV3	VBAT_DIV3	VTSEN	VBGREN
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.							LFTRG	Res.							REF

							rw								rw

Bits	Name	Description
31:30	CKSRC[1:0]	ADC clock selection 00: PCLK2 01: PCLK2/2 10: PCLK2/4 11: ADC_KCLK, refer to *Peripheral asynchronous clock selection*. *Note: The software is allowed to write these bits only when the ADC is disabled.*
29:28	Reserved	Must be kept at reset value
27:24	PRESC[3:0]	Prescaler of asynchronous clock ADC_KCLK 0000: ADC_KCLK not divided 0001: ADC_KCLK divided by 2 0010: ADC_KCLK divided by 4 0011: ADC_KCLK divided by 6 0100: ADC_KCLK divided by 8 0101: ADC_KCLK divided by 10 0110: ADC_KCLK divided by 12 0111: ADC_KCLK divided by 16 1000: ADC_KCLK divided by 32 1001: ADC_KCLK divided by 64 1010: ADC_KCLK divided by 128 Others: Reserved (ADC_KCLK not divided) *Note: The software is allowed to write these bits only when the ADC is disabled.*
23:20	Reserved	Must be kept at reset value
19	VDDA_DIV3	V_{DDA}/3 input enable 0: Disable 1: Enable *Note: Enabling V_{BAT}/3 input and V_{DDA}/3 input simultaneously is prohibited for they use the same internal ADC input channel.*

The software is allowed to write this bit only when START bit is 0.

18	VBAT_DIV3	V_{BAT}/3 input enable 0: Disable 1: Enable *Note: Enabling V_{BAT}/3 input and V_{DDA}/3 input simultaneously is prohibited for they use the same internal ADC input channel.* *The software is allowed to write this bit only when START bit is 0.*
17	VTSEN	Temperature sensor input enable 0: Disable 1: Enable *Note: The software is allowed to write this bit only when START bit is 0.*
16	VBGREN	V_{BGR} input enable 0: Disable 1: Enable *Note: The software is allowed to write this bit only when START bit is 0.*
15:9	Reserved	Must be kept at reset value
8	LFTRG	Low frequency trigger mode enable 0: Disable 1: Enable
7:1	Reserved	Must be kept at reset value
0	REF	ADC reference voltage V_{REF_ADC} selection 0: External reference voltage V_{REF+} or VREFBUF 1: V_{DDA} *Note: When the REF bit is 0 and VREFBUF is disabled, ADC reference voltage is from input of VREF+ pin. When the REF*

bit is 0 and VREFBUF is enabled, ADC reference voltage is from VREFBUF, refer to Voltage reference buffer.
The software is allowed to write this bit only when the ADC is disabled.

14.4.4 ADC interrupt and status register (ADC_ISR)

Address offset: 0x10

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				EOCAL	Res.			AWDG1	Res.		OVRN	EOS	EOC	EOSAMP	Res.
				rc_w1				rc_w1			rc_w1	rc_w1	rc_w1	rc_w1	

Bits	Name	Description
31:12	Reserved	Must be kept at reset value
11	EOCAL	End of calibration flag Set by hardware when calibration is complete, and cleared by software writing 1 to it. 0: Calibration is not complete 1: Calibration is complete
10:8	Reserved	Must be kept at reset value
7	AWDG1	Analog watchdog flag Set by hardware when the converted voltage crosses the values programmed in the ADC_AWDG1TR register, and cleared by software writing 1 to it. 0: No analog watchdog event occurred 1: Analog watchdog event occurred
6:5	Reserved	Must be kept at reset value
4	OVRN	Data overrun flag

		Set by hardware when a new conversion has completed while the EOC flag was already set, and cleared by software writing 1 to it. 0: No data overrun event occurred 1: Data overrun event occurred
3	EOS	End of conversion sequence flag Set by hardware at the end of the conversion of a sequence of channels selected by the ADC_CHCFG register, and cleared by writing 1 to it. 0: Conversion sequence is not complete 1: Conversion sequence is complete
2	EOC	End of conversion flag Set by hardware at the end of each conversion of a channel when a new data result is available in the ADC_DR register, and cleared by software writing 1 to it or by reading the ADC_DR register. 0: Channel conversion is not complete 1: Channel conversion is complete
1	EOSAMP	End of sampling flag Set by hardware during the conversion, at the end of the sampling phase, and cleared by software writing 1 to it. 0: The sampling is not complete 1: The sampling is complete
0	Reserved	Must be kept at reset value

14.4.5 ADC interrupt enable register (ADC_IER)

Address offset: 0x14

Reset value: 0x0000 0000

Note: The software is allowed to write this register only when START bit is 0.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				EOCALIE	Res.			AWDG1IE	Res.		OVRNIE	EOSIE	EOCIE	EOSAMPIE	Res.

				rw				rw			rw	rw	rw	rw	

Bits	Name	Description
31:12	Reserved	Must be kept at reset value
11	EOCALIE	End of calibration interrupt enable 0: Disable 1: Enable, the interrupt is generated when EOCAL bit is set
10:8	Reserved	Must be kept at reset value
7	AWDG1IE	Analog watchdog interrupt enable 0: Disable 1: Enable, the interrupt is generated when AWDG1 bit is set
6:5	Reserved	Must be kept at reset value
4	OVRNIE	Data overrun interrupt enable 0: Disable 1: Enable, the interrupt is generated when OVRN bit is set
3	EOSIE	End of conversion sequence interrupt enable 0: Disable 1: Enable, the interrupt is generated when EOS bit is set
2	EOCIE	End of conversion interrupt enable 0: Disable 1: Enable, the interrupt is generated when EOC bit is set
1	EOSAMPIE	End of sampling interrupt enable 0: Disable 1: Enable, the interrupt is generated when EOSAMP bit is set
0	Reserved	Must be kept at reset value

14.4.6 ADC sampling time register (ADC_SAMPT)

Address offset: 0x18

Reset value: 0x0000 0000

Note: The software is allowed to write this register only when START bit is 0.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.				SAMPT_SEL19	Res.		SAMPT_SEL16	SAMPT_SEL15	SAMPT_SEL14	SAMPT_SEL13	SAMPT_SEL12	SAMPT_SEL11	SAMPT_SEL10	SAMPT_SEL9	SAMT_SEL8
				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
SAMPT_SEL7	SAMPT_SEL6	SAMPT_SEL5	SAMPT_SEL4	SAMPT_SEL3	SAMPT_SEL2	SAMPT_SEL1	SAMPT_SEL0	SAMPT2[3:0]				SAMPT1[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Name	Description
31:28	Reserved	Must be kept at reset value
27	SAMPT_SEL[19]	Channel-19 sampling time selection 0: Setting of SAMPT1[3:0] bits 1: Setting of SAMPT2[3:0] bits
26:25	Reserved	Must be kept at reset value
24:8	SAMPT_SEL[16:0]	Channel-x sampling time selection 0: Setting of SAMPT1[3:0] bits 1: Setting of SAMPT2[3:0] bits
7:4	SAMPT2[3:0]	Sampling time selection 2 0001: 3 ADC_CK clock cycles 0010: 7 ADC_CK clock cycles 0011: 12 ADC_CK clock cycles 0100: 19 ADC_CK clock cycles 0101: 39 ADC_CK clock cycles 0110: 79 ADC_CK clock cycles 0111: 119 ADC_CK clock cycles 1000: 159 ADC_CK clock cycles 1001: 239 ADC_CK clock cycles 1010: 319 ADC_CK clock cycles 1011: 479 ADC_CK clock cycles 1100: 639 ADC_CK clock cycles 1101: 959 ADC_CK clock cycles

1110: 1279 ADC_CK clock cycles

1111: 1919 ADC_CK clock cycles

Others: Reserved (3 ADC_CK clock cycles)

3:0	SAMPT1[3:0]	Sampling time selection 1 0001: 3 ADC_CK clock cycles 0010: 7 ADC_CK clock cycles 0011: 12 ADC_CK clock cycles 0100: 19 ADC_CK clock cycles 0101: 39 ADC_CK clock cycles 0110: 79 ADC_CK clock cycles 0111: 119 ADC_CK clock cycles 1000: 159 ADC_CK clock cycles 1001: 239 ADC_CK clock cycles 1010: 319 ADC_CK clock cycles 1011: 479 ADC_CK clock cycles 1100: 639 ADC_CK clock cycles 1101: 959 ADC_CK clock cycles 1110: 1279 ADC_CK clock cycles 1111: 1919 ADC_CK clock cycles Others: Reserved (3 ADC_CK clock cycles)

14.4.7 ADC channel selection register (ADC_CHCFG)

Address offset: 0x1C

Reset value: 0x0000 0000

Note: The software is allowed to write this register only when START bit is 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.												CHN19	Res.		CHN16
												rw			rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHN15	CHN14	CHN13	CHN12	CHN11	CHN10	CHN9	CHN8	CHN7	CHN6	CHN5	CHN4	CHN3	CHN2	CHN1	CHN0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Name	Description
31:20	Reserved	Must be kept at reset value

19	CHN[19]	Conversion channel-19 selection 0: Input channel-19 is not selected for conversion 1: Input channel-19 is selected for conversion
18:17	Reserved	Must be kept at reset value
16:0	CHN[16:0]	Conversion channel-x selection 0: Input channel-x is not selected for conversion 1: Input channel-x is selected for conversion

14.4.8 ADC analog watchdog configuration register (ADC_AWDG1CR)

Address offset: 0x20

Reset value: 0x0000 0000

Note: The software is allowed to write this register only when START bit is 0.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.												CHN19	Res.		CHN16
												rw			rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHN15	CHN14	CHN13	CHN12	CHN11	CHN10	CHN9	CHN8	CHN7	CHN6	CHN5	CHN4	CHN3	CHN2	CHN1	CHN0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Name	Description
31:20	Reserved	Must be kept at reset value
19	CHN[19]	Analog watchdog channel selection 0: Channel-19 is not monitored by AWDG1 1: Channel-19 is monitored by AWDG1
18:17	Reserved	Must be kept at reset value
16:0	CHN[16:0]	Analog watchdog channel selection 0: Channel-x is not monitored by AWDG1 1: Channel-x is monitored by AWDG1

14.4.9 ADC analog watchdog threshold register (ADC_AWDG1TR)

Address offset: 0x28

Reset value: 0x0FFF 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.				AWDG_HT[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

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

Bits	Name	Description
31:28	Reserved	Must be kept at reset value
27:16	AWDG_HT[11:0]	ADC analog watchdog higher threshold
15:12	Reserved	Must be kept at reset value
11:0	AWDG_LT[11:0]	ADC analog watchdog lower threshold

14.4.10 ADC calibration factor register (ADC_CALFACT)

Address offset: 0x30

Reset value: 0x0000 0000

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

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

Bits	Name	Description
31:6	Reserved	Must be kept at reset value
5:0	CALFACT[5:0]	Calibration factor These bits are written by hardware or by software. – Once a calibration is complete, they are updated by hardware with the calibration factors. – Software can configure a new calibration factor, which is applied once a new calibration is launched.

Note: The software is allowed to set CALFACT only when ADEN = 1 and START = 0. After a calibration is complete, calibration factor is also stored to DATA[5:0] in the ADC_DR register.

14.4.11 ADC data register (ADC_DR)

Address offset: 0x40

Reset value: 0x0000 0000

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

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

Bits	Name	Description
31:12	Reserved	Must be kept at reset value
11:0	DATA[11:0]	Converted data

15 Voltage reference buffer (VREFBUF)

15.1 Introduction

The device embeds a voltage reference buffer (VREFBUF) as a high-precision reference voltage source, which operates stably across the entire temperature range. The VREFBUF is factory-calibrated, and it automatically loads the factory calibration values into the calibration register upon power-up. The output voltage of the VREFBUF remains unaffected by variations in V_{DDA}, providing a high-precision reference voltage for the ADC and supplying the input voltage for the 6-bit DAC (which offers a 64-level voltage divider reference source).

15.2 VREFBUF main features

- Supports three voltages
  - 2.048 V (2.4 V ≤ V_{DDA} ≤ 5.5 V)
  - 2.5 V (2.8 V ≤ V_{DDA} ≤ 5.5 V)
  - 3.0 V (3.3 V ≤ V_{DDA} ≤ 5.5 V)
- Supports 2 mA driving capability and 1 Msps ADC sampling rate
- Provides high-precision voltage reference for external components

15.3 VREFBUF functional description

An external capacitor (1μF + 0.1μF) must be connected to the V_{REF+} pin before using the VREFBUF. Then enable the BGR and wait for the BGR to stabilize. See *Internal bandgap reference voltage*.

After enabling the VREFBUF by setting EN bit in the VREFBUF_CSR register, the user must wait for the stabilization time (t_{STAB}) (refer to the data sheet for stabilization specifications), meaning that the voltage reference output has reached its expected value.

VREFBUF has been factory-calibrated for output voltages of 2.048 V, 2.5 V, and 3.0 V. Before enabling VREFBUF, the corresponding calibration values must be read from the calibration parameter storage area and written to the *VREFBUF calibration register (RCC_VREFBUFCAL)*. The calibration value for 2.048 V is automatically loaded during power-on reset. The calibration values are stored in the calibration parameter storage area, as shown in the following table:

Table 15-1 VREFBUF calibration parameter storage address

VREFBUF voltage reference	Calibration temperature and voltage conditions	Storage address
2.048 V	Temperature: 25 °C (± 2 °C) V_{DD} = 3.3 V	0x1FFF03D4
2.5 V		0x1FFF03D8
3.0 V		0x1FFF03DC

15.4 VREFBUF registers

The VREFBUF registers can only be accessed by words (32-bit).

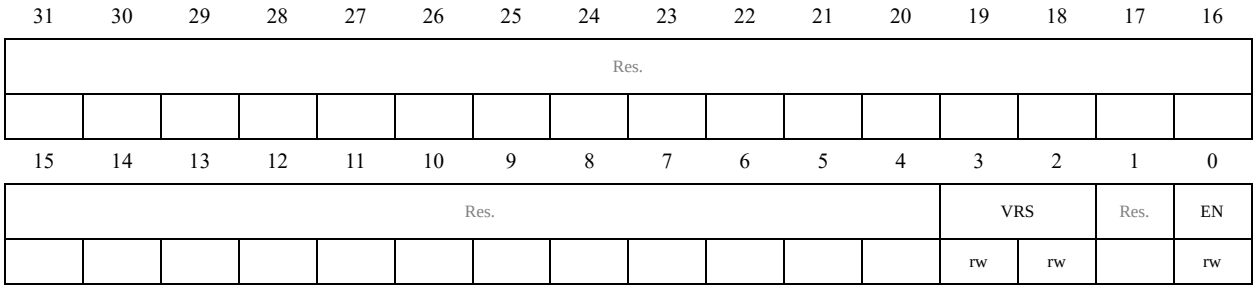
Table 15-2 VREFBUF base address

Peripheral	Base address
VREFBUF	0x4001 01B0

15.4.1 VREFBUF control and status register (VREFBUF_CSR)

Address offset: 0x00

Reset value: 0x0000 0000



Bits	Name	Description
31:4	Reserved	Must be kept at reset value
3:2	VRS[1:0]	VREFBUF voltage reference scale This bit selects the value generated by the VREFBUF. 00: 2.048 V 01: 2.5 V 10: 3.0 V 11: Reserved (default: 2.048 V)
1	Reserved	Must be kept at reset value
0	EN	VREFBUF enable 0: VREFBUF disable 1: VREFBUF enable

15.4.2 VREFBUF calibration control register (VREFBUF_CAL)

Refer to *VREFBUF calibration register (RCC_VREFBUFCAL)*.

16 Comparator (COMP)

16.1 Introduction

The device integrates two ultra-low-power comparators (COMP1/COMP2), supporting independent operation, window comparator configuration, and combination with timers. This enables flexible applications including:

- Wake-up from low-power mode triggered by an analog signal
- Analog signal voltage comparison

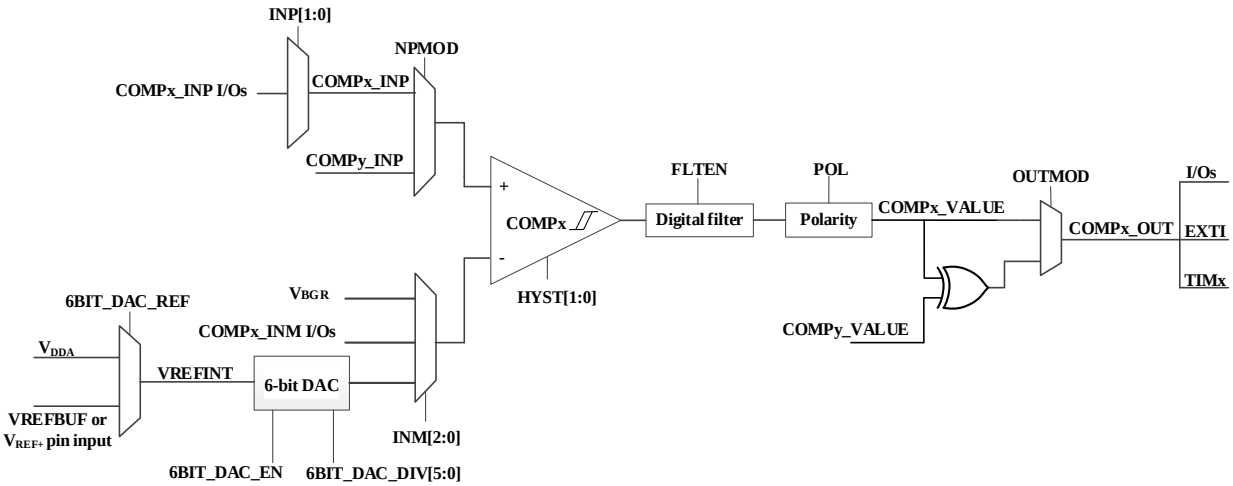
16.2 COMP main features

- Supports rail-to-rail inputs
- Programmable speed / consumption, with a minimum consumption of 500 nA
- Programmable hysteresis
- Configurable plus and minus inputs for flexible voltage selection:
  - Multiplexed I/O pins
  - Internal bandgap reference voltage: V_{BGR}
  - 6-bit DAC (64-level voltage divider reference source)
- Supports wakeup from Stop mode
- The outputs can be redirected to I/Os or to timer inputs (including TIM3/4/5 and LPTIM1) for triggering
- Configurable digital filter
- Window comparator

16.3 COMP functional description

16.3.1 COMP block diagram

Figure 16-1 COMP block diagram (1)



1. In the diagram, x/y taking 1/2 represents COMP1/2, and x/y are not equal. The register for 6-bit DAC is described in the *System Control Register (SYSCFG_CR)*.

16.3.2 COMP pins and internal signals

The I/Os used as comparators inputs must be configured in analog mode in the GPIOs registers. Two non-inverting inputs can be exchanged internally by enabling INPMOD bit in the COMP1_CSR and the COMP2_CSR register.

The comparator output can be connected to the I/Os using the alternate function channel, and can also be internally redirected to a variety of timers (including TIM3, TIM4, TIM5 and LPTIM1) for the purposes of input capture, etc. The comparator output can be simultaneously connected to GPIOs and other peripherals.

Table 16-1 COMP1 non-inverting input assignment

COMP1 non-inverting input	COMP1_INP[1:0]
PC4	00
PA5	01
PB2	10
PA8	11

Table 16-2 COMP1 inverting input assignment

COMP1 inverting input	COMP1_INM[2:0]
V_{BGR}	001

6-bit DAC	010
PC3	011
PA4	100
PB1	101
PA9	110

Table 16-3 COMP2 non-inverting input assignment

COMP2 non-inverting input	COMP2_INP
PA1	0
PD5	1

Table 16-4 COMP2 inverting input assignment

COMP2 inverting input	COMP2_INM[2:0]
V_{BGR}	001
6-bit DAC	010
PA2	011
PD4	100

When selecting V_{BGR} or the 6-bit DAC as the inverting input signal for the comparator, set EN bit to 1 in the COMPx_CSR register and wait for the comparator to stabilize.

16.3.3 COMP enable and disable control

After enabling the comparator by setting EN bit to 1 in the COMPx_CSR register, wait for the designated stabilization time (t_{START}) (refer to the data sheet for stabilization specifications). Clear the EN bit in the COMPx_CSR register to disable the comparator.

16.3.4 Window comparator

Comparator 1 and comparator 2 can be utilized to create window comparator, aiming to monitor the analog voltage if it is within specified voltage range defined by lower and upper threshold. The monitored analog voltage is connected to the non-inverting inputs of comparators connected together and the upper and lower threshold voltages are connected to the inverting inputs of the comparators.

When using the window comparison function, set the INPMOD bit of one COMPy register to 1 to select the non-inverting input signal corresponding to the COMPx_INP configuration, thereby ensuring the non-inverting inputs for both comparators connect internally together and select the same input signal. When the INPMOD bit for

COMP1/2 are both set to 1, the non-inverting inputs of COMP1/2 are swapped.

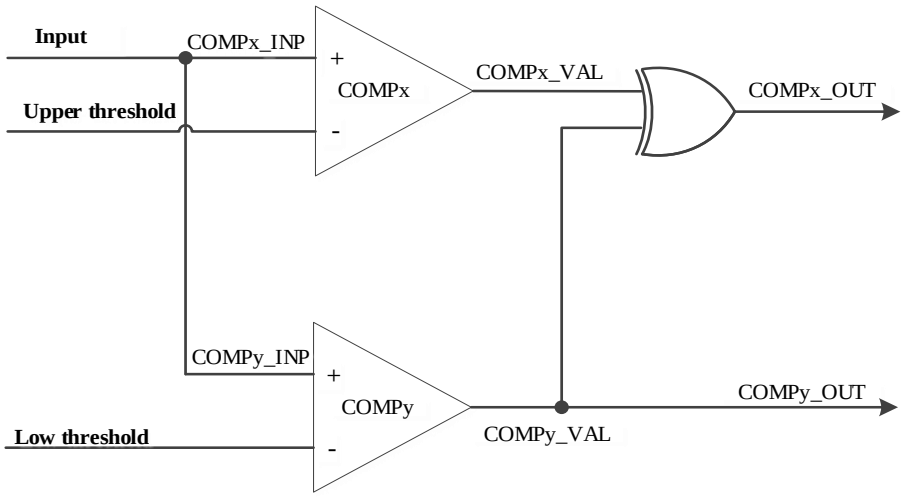
Additionally, configuring the OUTMOD bit of the COMPx register to 1 allows the output states of the two comparators (filtered and polarity control results, labeled as COMPx_VAL and COMPy_VAL) to be XOR-computed. The comparator output signal COMPx_OUT is the XOR value of COMPx_VAL and COMPy_VAL.

The comparator output mode configuration (OUTMOD bit) is not strictly tied to the window comparator mode. Specifically:

- In window comparator mode: The outputs of the two comparators can still remain independent, allowing software to process each output signal separately.
- In non-window comparator modes: The two comparators' results can also be configured to produce an XOR-combined output for software processing.

The window mode is illustrated in the following figure:

Figure 16-2 Window mode (1)



1. For COMPx, the bit INPMOD = 0, and the bit OUTMOD = 1. For COMPy, the bit INPMOD = 1, and the bit OUTMOD = 0.

16.3.5 Polarity and filter

The comparator supports polarity selection and digital filtering function.

The comparator output polarity can be selected through the POL bit, and the output state VAL is as follows:

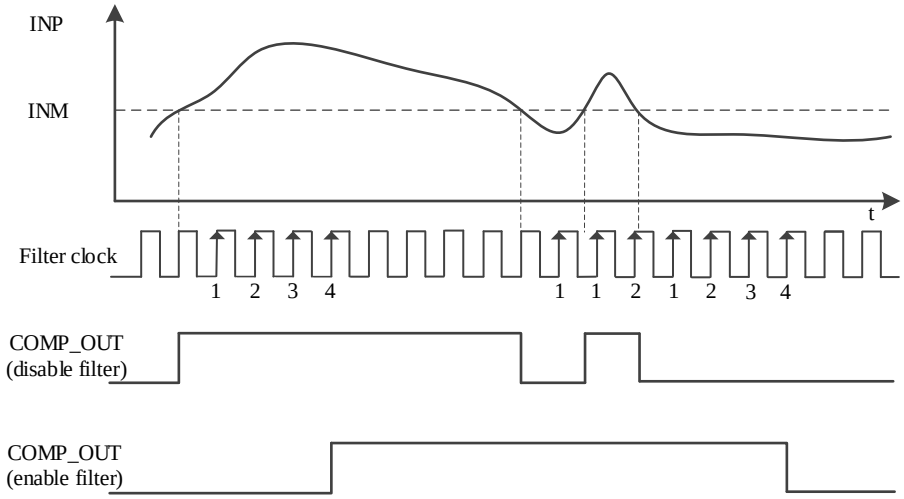
- The comparator output state in not inverted when the POL bit is set to 0

- If the non-inverting input voltage is greater than the inverting input voltage, the output state VAL is 1, otherwise, it is 0.

- The comparator output state in inverted when the POL bit is set to 1
  - If the non-inverting input voltage is greater than the inverting input voltage, the output state VAL is 0, otherwise, it is 1.

The comparator has a digital filtering function to filter out system noise by setting the FLTEN bit to 1. The filter clock source only supports PCLK2 clock, and the filter time is selected by the FLTIME[2:0] bit field. The diagram for setting the filter time to 4 clock cycles is shown below:

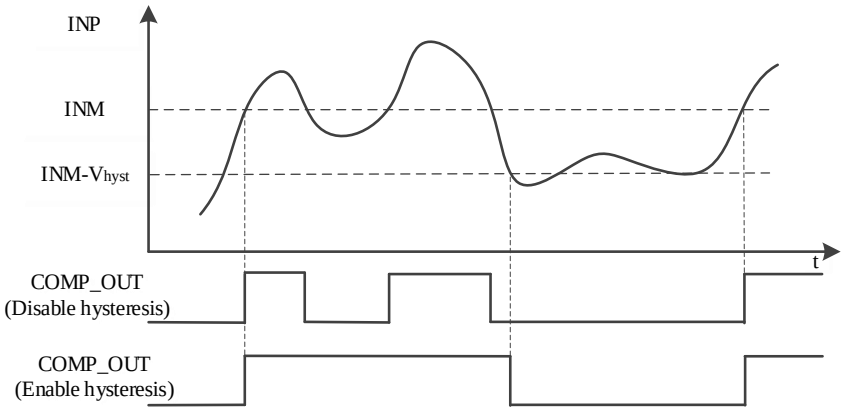
Figure 16-3 Comparator digital filtering



16.3.6 Hysteresis

The comparator includes a programmable hysteresis to avoid spurious output transitions in case of noisy signals. The hysteresis levels of 10 mV, 20 mV, and 30 mV can be set through the HYST[1:0] bit field. The hysteresis can be disabled if it is not needed (for instance when exiting from low-power mode) to be able to force the hysteresis value using external components. Refer to the following diagram for the comparator hysteresis:

Figure 16-4 Comparator hysteresis



16.3.7 COMP power and speed modes

The comparator response time is defined as the duration from a voltage change at either the non-inverting or inverting input signals to the output of a valid comparison result. This response speed correlates directly with power consumption - faster response requires higher power. The comparator's speed and power modes can be configured via the SPEED[1:0] bit field in its control register, as specified in the table below:

Table 16-5 Response speed and power modes

SPEED[1:0]	Response speed (μs)	Power (μA)
00: High speed (full power)	0.1	22.5
01: Medium speed (medium power)	0.15	12
10: Low speed (low power)	1	1.7
11: Ultra-low speed (ultra-low power)	3.5	0.5

16.3.8 Comparator lock mechanism

The comparators can be used for safety purposes, such as over-current or thermal protection. For applications having specific functional safety requirements, it is necessary to insure that the comparator programming cannot be altered in case of spurious register access or program counter corruption.

For this purpose, the COMPx_CSR register can be write-protected (read-only).

Once the programming is completed, the COMPx LOCK bit can be set to 1. This causes the whole register to become read-only, including the COMPx LOCK bit. The write protection can only be reset by system reset and cannot be reset by COMP

peripheral reset in SYSCFG_RST.

16.4 COMP interrupts

The comparator outputs are internally connected to the extended interrupts and events controller. Each comparator has its own EXTI line and can generate either interrupts or events. The same mechanism is used to exit from low-power modes. Refer to *section: Extended interrupt and event controller*.

Table 16-6 COMP interrupts

Interrupt event	Interrupt flag	Enable control bit	Exit from Sleep	Exit from Stop
COMP1 output	COMP1_OUT	through EXTI	yes	yes
COMP2 output	COMP2_OUT	through EXTI	yes	yes

16.5 COMP registers

The COMP registers can only be accessed by words (32-bit).

Table 16-7 COMP base address

Peripheral	Base address
COMP	0x4001 0200

16.5.1 Comparator 1 control and status register (COMP1_CSR)

Address offset: 0x00

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	VAL	Res.	FLTEN	FLTIME[2:0]			Res.					SPEED[1:0]		HYST[1:0]	
rs	r		rw	rw	rw	rw						rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POL	OUTMOD	INPMOD	Res.			INP[1:0]		Res.	INM[2:0]			Res.			EN
rw	rw	rw				rw	rw		rw	rw	rw				rw

Bits	Name	Description
31	LOCK	COMP1_CSR register lock This bit is set by software and cleared only by a system reset. It locks the whole content of the register COMP1_CSR[31:0] 0: The COMP1_CSR[31:0] register read/write bits can be written by software 1: The COMP1_CSR[31:0] register bits can be read out but not written by software
30	VAL	Comparator 1 output status This bit is read-only. It reflects the level of the comparator 1 output after the polarity selector and digital filter.
29	Reserved	Must be kept at reset value
28	FLTEN	Comparator 1 digital filter enable 0: Disable 1: Enable

27:25	FLTIME[2:0]	Digital filter time selection (PCLK2 clock) 000: 2 clock cycles 001: 4 clock cycles 010: 8 clock cycles 011: 16 clock cycles 100: 32 clock cycles 101: 64 clock cycles 110: 128 clock cycles 111: 256 clock cycles
24:20	Reserved	Must be kept at reset value
19:18	SPEED[1:0]	Comparator 1 speed and power mode selector 00: High speed and full power 01: Medium speed and medium power 10: Low speed and low power 11: Ultra-low speed and ultra-low power
17:16	HYST[1:0]	Comparator 1 hysteresis selector 00: None 01: Low hysteresis voltage with 10 mV 10: Medium hysteresis voltage with 20 mV 11: High hysteresis voltage with 30 mV
15	POL	Comparator 1 polarity selector 0: Non-inverted 1: Inverted
14	OUTMOD	Comparator 1 output selector 0: COMP1_VAL 1: COMP1_VAL XOR COMP2_VAL
13	INPMOD	Comparator 1 non-inverting input selector for window mode 0: Signal selected with INP[1:0] bit field of this register 1: COMP2_INP signal of the comparator 2 (required for window mode)

12:10	Reserved	Must be kept at reset value
9:8	INP[1:0]	Comparator 1 signal selector for non-inverting input 00: PC4 01: PA5 10: PB2 11: PA8
7	Reserved	Must be kept at reset value
6:4	INM[2:0]	Comparator 1 signal selector for inverting input 001: V_{BGR} 010: 6-bit DAC 011: PC3 100: PA4 101: PB1 110: PA9 Others: Reserved *Note: Configure for 6-bit DAC see system control register SYSCFG_CR*
3:1	Reserved	Must be kept at reset value
0	EN	Comparator 1 enable bit 0: Disable 1: Enable

16.5.2 Comparator 2 control and status register (COMP2_CSR)

Address offset: 0x04

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	VAL	Res.	FLTEN	FLTIME[2:0]			Res.					SPEED[1:0]		HYST[1:0]	
rs	r		rw	rw	rw	rw						rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POL	OUTMOD	INPMOD	Res.				INP	Res.	INM[2:0]			Res.			EN
rw	rw	rw					rw		rw	rw	rw				rw

Bits	Name	Description
31	LOCK	COMP2_CSR register lock This bit is set by software and cleared only by a system reset. It locks the whole content of the register COMP2_CSR[31:0] 0: The COMP2_CSR[31:0] register read/write bits can be written by software 1: The COMP2_CSR[31:0] register bits can be read out but not written by software
30	VAL	Comparator 2 output status This bit is read-only. It reflects the level of the comparator 2 output after the polarity selector and digital filter.
29	Reserved	Must be kept at reset value
28	FLTEN	Comparator 2 digital filter enable 0: Disable 1: Enable
27:25	FLTIME[2:0]	Digital filter time selection (PCLK2 clock) 000: 2 clock cycles 001: 4 clock cycles 010: 8 clock cycles 011: 16 clock cycles 100: 32 clock cycles 101: 64 clock cycles 110: 128 clock cycles 111: 256 clock cycles
24:20	Reserved	Must be kept at reset value
19:18	SPEED[1:0]	Comparator 2 speed and power mode selector 00: High speed and full power 01: Medium speed and medium power 10: Low speed and low power 11: Ultra-low speed and ultra-low power

17:16	HYST[1:0]	Comparator 2 hysteresis selector 00: None 01: Low hysteresis voltage with 10 mV 10: Medium hysteresis voltage with 20 mV 11: High hysteresis voltage with 30 mV
15	POL	Comparator 2 polarity selector 0: Non-inverted 1: Inverted
14	OUTMOD	Comparator 2 output selector 0: COMP2_VAL 1: COMP2_VAL XOR COMP1_VAL
13	INPMOD	Comparator 2 non-inverting input selector for window mode 0: Signal selected with INP[1:0] bit field of this register 1: COMP1_INP signal of the comparator 1 (required for window mode)
12:9	Reserved	Must be kept at reset value
8	INP	Comparator 2 signal selector for non-inverting input 0: PA1 1: PD5
7	Reserved	Must be kept at reset value
6:4	INM[2:0]	Comparator 2 signal selector for inverting input INM 001: V_{BGR} 010: 6-bit DAC 011: PA2 100: PD4 Others: Reserved *Note: Configure for 6-bit DAC see system control register SYSCFG_CR*
3:1	Reserved	Must be kept at reset value

0	EN	Comparator 2 enable bit 0: Disable 1: Enable

17 Liquid crystal display controller (LCD)

17.1 Introduction

The LCD controller is a digital controller/driver for monochrome passive liquid crystal display (LCD) with up to 8 common terminals and up to 44 segment terminals to drive 144 (4 COMs x 36 SEGs) or 204 (6 COMs x 34 SEGs) or 256 (8COMs x 32 SEGs) LCD picture elements (pixels). The exact number of terminals depends on the device pinout as described in the datasheet.

17.2 LCD main features

- Supports 1/4, 1/6 and 1/8 duty
- Supports 1/3 and 1/4 bias
- Drive mode:
  - Charge pump mode: a step-up converter features high drive capability, with V_{LCD} boost voltage that can exceed V_{DD} and remains independent of V_{DD} variations. The V_{LCD} supports multiple configurable voltage levels, reaching up to 5.25 V. Even if the V_{DD} voltage drops or fluctuates, it ensures optimal display performance under dynamic conditions.
  - Internal resistor division mode: an internal resistive network offers 16-level adjustable contrasts, dynamic switching between high and low drive capability, eliminates external capacitor requirements, and boosts IO utilization efficiency.
  - External capacitor division mode: the drive voltage is derived through a V_{DD} divider configuration, offering superior display performance compared to the internal resistor division mode.
- Blink capability with software adjustable blink mode and blink frequency
- Programmable deadtime between frames
- Frame rate rang: 30-100 Hz (64 Hz typical)
- Supports LCD antipolar-polarization technology
- Supports waveform B

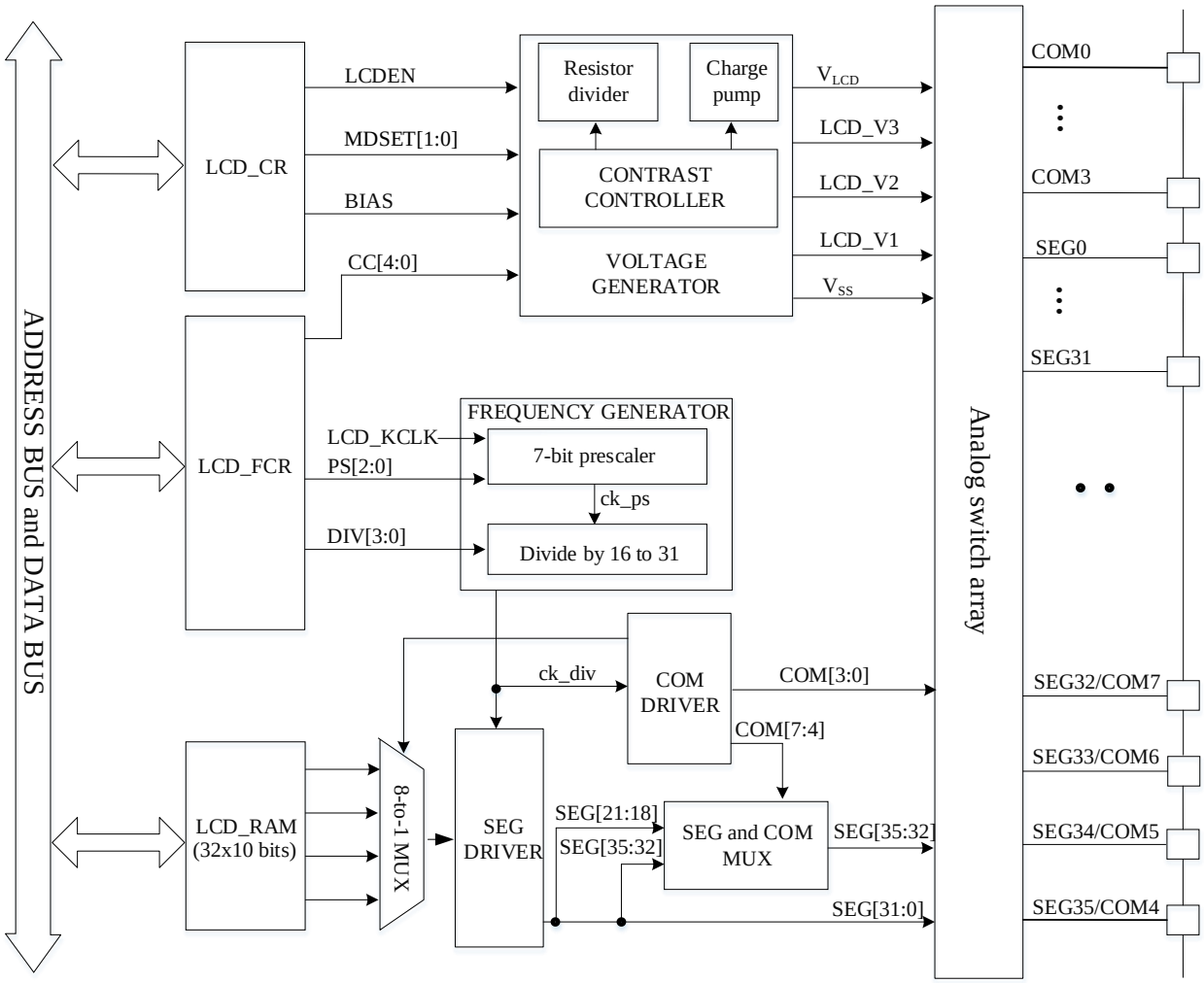
- Supports low-power modes: the LCD controller can be displayed in Sleep and Stop modes

17.3 LCD functional description

17.3.1 General description

The LCD controller has five main blocks, including frequency generator, voltage generator, common driver, segment driver, common and segment mux (see Figure 17-1).

Figure 17-1 LCD controller block diagram

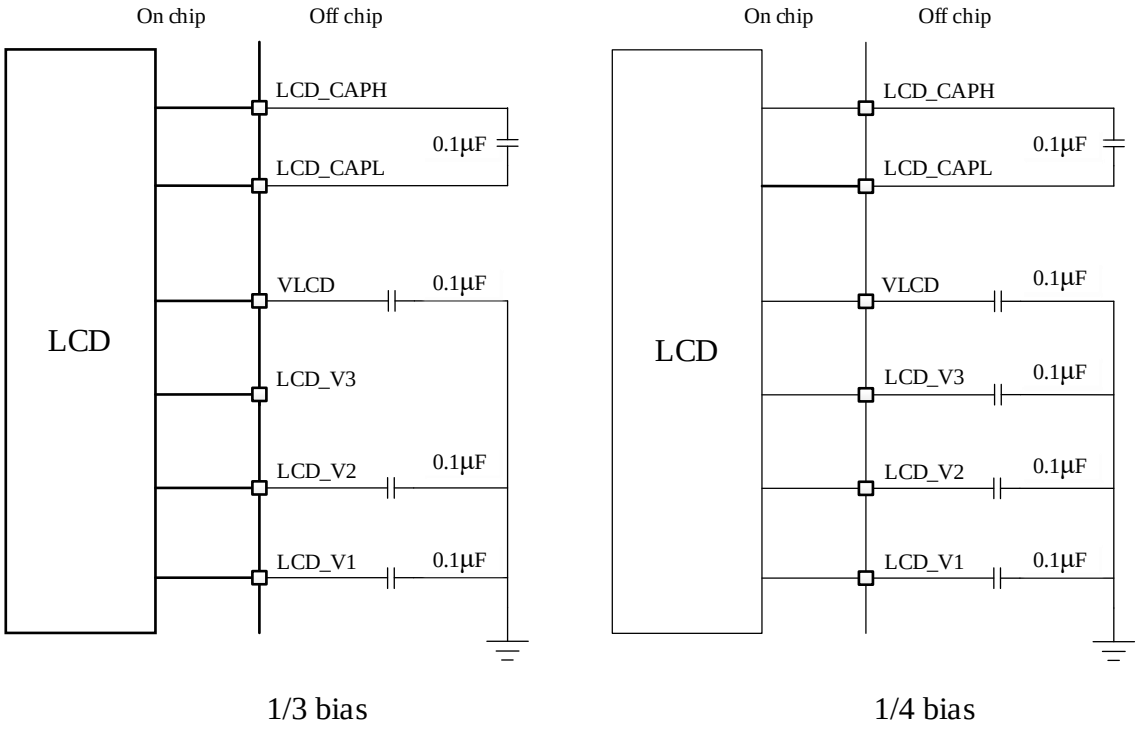


17.3.2 Charge pump mode

In charge pump mode (MDSET[1:0] = 00), a robust drive capability is provided. The boost voltage of V_{LCD} can exceed V_{DD} and remains independent of V_{DD} fluctuations, ensuring stable waveform generation under dynamic voltage conditions. Even if

battery voltage drops or V_{DD} voltage fluctuates, this mode maintains consistent drive performance, preventing display quality degradation in battery-powered applications. The boost voltage of V_{LCD} can reach up to 5.25 V, supporting enhanced display clarity for 8 COM and 6 COM LCD modules while meeting stringent voltage requirements.

Figure 17-2 LCD charge pump mode



As shown in the diagram, external 0.1 µF capacitors should be added between LCD_CAPH and LCD_CAPL pins, and between each of the following pins and V_{SS}: LCD_V1, LCD_V2, LCD_V3, and VLCD. When operating in 1/3 bias mode, LCD_V3 pin can be utilized as a GPIO.

The steps for pin configuration:

The LCD_CAPH/LCD_CAPL and GPIOs are automatically connected internally and require no configuration. The LCD_PAD_EN[3:0] bit field enables connections between LCD_V1/LCD_V2/LCD_V3/VLCD pins and the GPIOs. Refer to the *LCD control register (LCD_CR)* section for register mapping details.

The charge pump clock divider balances LCD power consumption and display performance by adjusting the CPDIV[2:0] bit field in the *LCD control register (LCD_CR)*. To minimize power consumption while maintaining display quality, set the charge pump clock divider ratio to no less than 16.

To ensure stable drive voltage and avoid unexpected voltages affecting display quality

during charge pump startup, follow this sequence:

- During initialization, clear the SCOC bit in the LCD_FCR register. This connects the COM and SEG pins to V_{SS}.
- After enabling the LCD, wait for the designated stabilization time (t_{STAB}) (refer to the data sheet for voltage stabilization specifications).
- Set the SCOC bit to restore normal COM and SEG signal output.

V_{LCD} voltage level configuration

The CC[4:0] bit field within the LCD frame control register (LCD_FCR) is used to configure the V_{LCD} voltage level in charge pump mode. Refer to the following table for the voltage corresponding to each CC[4:0] value.

Table 17-1 V_{LCD} voltage levels of charge pump mode

CC[4:0]	V_{LCD} voltage at different bias settings (V)	
	1/3 bias	1/4 bias
00000	2.55	2.60
00001	2.70	2.80
00010	2.85	3.00
00011	3.00	3.20
00100	3.15	3.40
00101	3.30	3.60
00110	3.45	3.80
00111	3.60	4.00
01000	3.75	4.20
01001	3.90	4.40
01010	4.05	4.60
01011	4.20	4.80
01100	4.35	5.00
01101	4.50	5.20
01110	4.65	-
01111	4.80	-
10000	4.95	-
10001	5.10	-

CC[4:0]	V_{LCD} voltage at different bias settings (V)	
	1/3 bias	1/4 bias
10010	5.25	-

Note: V_{LCD} voltage is clamped to 5.25 V under 1/3 bias when CC[4:0] value surpassed 10010
V_{LCD} voltage is clamped to 5.20 V under 1/4 bias when CC[4:0] value surpassed 01101

17.3.3 Internal resistor division mode

In internal resistor division mode (MDSET[1:0] = 10), no external capacitors are required while maximizing I/O utilization.

Drive capability configuration

The HD bit and PON[3:0] bits in the LCD_FCR register can be used to configure the drive capability described below:

Table 17-2 LCD drive selection

HD	PON[3:0]	LCD drive selection
HD = 0	PON[3:0] = 0	low drive mode
HD = 0	PON[3:0] ≠ 0	dynamic switching between high-drive and low-drive mode
HD = 1	-	high drive mode

Contrast control

When operating in high-drive mode, the CC[3:0] bit field within the *LCD frame control register (LCD_FCR)* enables VLCD voltage level selection to adjust display contrast. Refer to the voltage levels mapped in the following Table.

Table 17-3 V_{LCD} voltage levels of internal resistor division mode (only for high-drive mode)

CC[3:0]	V_{LCD} voltage levels at different V_{DD} supply (V)				
	5	4.5	3.6	3.3	3.0
0000	2.74	2.47	1.97	1.81	1.64
0001	2.83	2.55	2.04	1.87	1.70
0010	2.92	2.63	2.10	1.93	1.75
0011	3.01	2.71	2.17	1.99	1.81
0100	3.12	2.81	2.25	2.06	1.87
0101	3.23	2.91	2.33	2.13	1.94
0110	3.35	3.02	2.41	2.21	2.01

CC[3:0]	V_{LCD} voltage levels at different V_{DD} supply (V)				
	5	4.5	3.6	3.3	3.0
0111	3.47	3.12	2.50	2.29	2.08
1000	3.61	3.25	2.60	2.38	2.17
1001	3.76	3.38	2.71	2.48	2.26
1010	3.93	3.54	2.83	2.59	2.35
1011	4.10	3.69	2.95	2.71	2.46
1100	4.30	3.87	3.10	2.83	2.58
1101	4.51	4.06	3.25	2.98	2.71
1110	4.75	4.28	3.42	3.14	2.85
1111	5.00	4.50	3.60	3.30	3.00

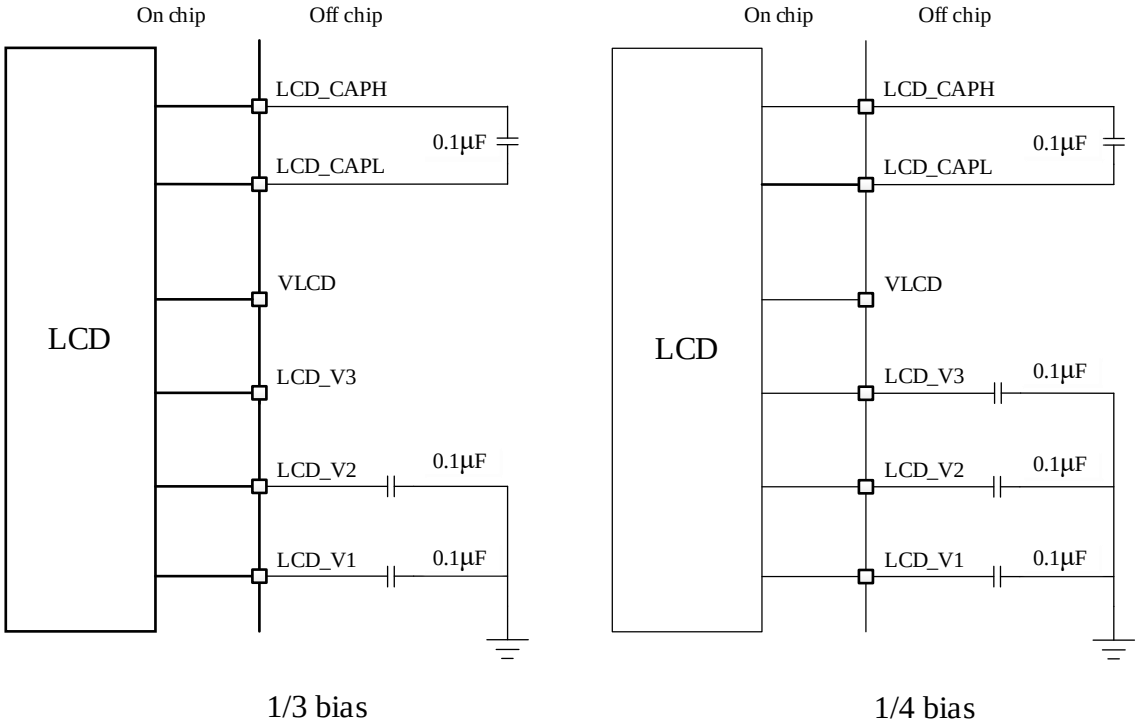
Note: The lower 4 bits (CC[3:0]) of the CC[4:0] bit field are active in this mode.

17.3.4 External capacitor division mode

In external capacitor division mode (MDSET[1:0] = 01), the intermediate voltages on LCD_V1, LCD_V2 and LCD_V3 pins are derived directly from the V_{DD} voltage divider, offering superior display performance compared to the internal resistor division mode.

External 0.1 µF capacitors should be added between LCD_CAPH and LCD_CAPL pins, and between each of the following pins and V_{SS}: LCD_V1, LCD_V2 and LCD_V3. The VLCD pin can be utilized as a GPIO. When operating in 1/3 bias mode, LCD_V3 pin can also be utilized as a GPIO. Refer to the diagram below:

Figure 17-3 LCD external capacitor division mode



The charge pump clock divider in the CPDIV[2:0] bit field of the LCD_CR register enables users to configure an optimal divider value to balance LCD power consumption and display performance based on application requirements.

17.3.5 Frequency generator

The frequency generator consists of a prescaler (7-bit ripple counter) and a 16 to 31 clock divider (see *Figure LCD controller block diagram)*.

Frequency generator clock sources

LCD_KCLK is the same as RTCCLK. Refer to the RTC/LCD clock description in the RCC *Section: RTC and LCD clock*.

- 32 KHz low speed external RC (LXTAL)
- 32 KHz low speed internal RC (RCL)

Frequency generator output

The frequency generator outputs the prescaler clock (ck_ps) and the divider clock (ck_div) based on the input clock (LCD_KCLK), thereby providing the LCD controller with the required clock frequencies.

The relation between the input clock frequency (f_{LCD_KCLK}) and its output clock

frequency f_{ck_ps} and f_{ck_div} are:

$$f_{ck\_ps} = \frac{f_{LCD\_KCLK}}{2^{PS[2:0]}}$$

$$f_{ck\_div} = \frac{f_{ck\_ps}}{(16 + \mathrm{DIV})} = \frac{f_{LCD\_KCLK}}{2^{PS[2:0]}(16 + \mathrm{DIV}[3:0])}$$

- ck_ps
  - The prescaler clock - ck_ps, used for pulse duration.
- ck_div
  - Serves as the phrase clock for LCD controller's drive waveform timing, see *section: LCD drive waveform.*
  - Generates the frame frequency for LCD display.
  - Produces the blink frequency for LCD controller's blink display function, see *section: Blink display.*

Frame frequency

The frame frequency must be selected to be within a range of around ~30 Hz to ~100 Hz, with a typical value of 64 Hz. The relation between the frame frequency f_{frame} and frequency generator's output clock frequency f_{ck_div} is:

$$f_{frame} = f_{ck\_div} \times \mathrm{duty}$$

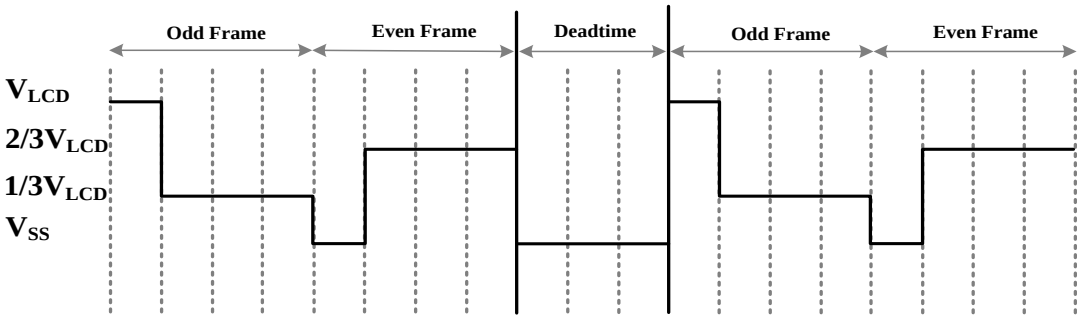
Table 17-4 Example of frame rate calculation

F_{FRAME}	LCD_KCLK	4 COM		6 COM		8 COM	
		PS[2:0]	DIV[3:0]	PS[2:0]	DIV[3:0]	PS[2:0]	DIV[3:0]
51 Hz	32.768 KHz	3	3	2	11	2	3
57 Hz	32.768 KHz	3	2	2	8	2	2
64 Hz	32.768 KHz	3	0	2	5	2	0
68 Hz	32.768 KHz	2	14	2	4	1	14
76 Hz	32.768 KHz	2	11	2	2	1	11

17.3.6 Deadtime

The DEAD[2:0] bits in the LCD_FCR register can be used to program a time of up to seven phase periods. During the deadtime the COM and SEG values are put to V_{SS}, see *LCD frame control register (LCD_FCR).*

Figure 17-4 Deadtime



17.3.7 Blink display

The LCD controller also implements a programmable blink feature to allow some pixels to continuously switch on at a specific frequency. The blink mode can be configured by the BLINK[1:0] bits in the LCD_FCR register. The blink frequency can be selected from eight different values using the BLINKF[2:0] bits in the LCD_FCR register. (see *Section: LCD frame control register (LCD_FCR)*)

- Blink mode
  - Blink mode 1: BLINK[1:0] = 01, blink enabled on SEG[0], COM[0] (1 pixel)
  - Blink mode 2: BLINK[1:0] = 10, blink enabled on SEG[0], all COMs (up to 8 pixels depending on the programmed duty)
  - Blink mode 3: BLINK[1:0] = 11, blink enabled on all SEGs and all COMs (all pixels)
- Blink frequency
  - The typical blink frequency is in the range of 0.25 Hz, 0.5 Hz, 1 Hz, 2 Hz or 4 Hz.
  - The blink frequency (f_{blink}) is defined as:

$$f_{blink} = \frac{f_{ck\_div}}{2^{[BLINKF+3]}}$$

17.3.8 COM and SEG multiplexing

The maximum number of output pins depends on the package pin count.

Table 17-5 LCD COM/SEG output pins versus package

Package pin count	SEG output pins	COM output pins	pixels

64 pins	SEG[35:0]	COM[3:0]	36×4
48 pins	SEG[21:0]	COM[3:0]	22×4

COM and SEG multiplexing of 64 pins package

All the possible ways of multiplexing the COM and SEG functions are described in Table 17-7.

Table 17-6 Remapping capability for 64 pins package (1)

Configuration bits		Remapping capability	Output pin	Function
DUTY	MUX_SEG			
1/8	-	32x8	COM[7:4]	COM[7:4]
			SEG[31:0]	SEG[31:0]
			COM[3:0]	COM[3:0]
1/6	1	34x6	COM[5:4]	COM[5:4]
			COM[7:6]	SEG[33:32]
			SEG[31:0]	SEG[31:0]
			COM[3:0]	COM[3:0]
1/4	1	36x4	COM[7:4]	SEG[35:32]
			SEG[31:0]	SEG[31:0]
			COM[3:0]	COM[3:0]

1. '–' considered invalid configuration.

COM and SEG multiplexing of 48 pins package

All the possible ways of multiplexing the COM and SEG functions are described in Table 17-8.

Table 17-7 Remapping capability for 48 pins package (1)

Configuration bits		Remapping capability	Output pin	Function
DUTY	MUX_SEG			
1/8	-	18x8	COM[7:4]	COM[7:4]
			SEG[31:30]	SEG[31:30]
			SEG[15:0]	SEG[15:0]
			COM[3:0]	COM[3:0]
1/6	1	20x6	COM[5:4]	COM[5:4]
			SEG[31:30]	SEG[31:30]
			COM[7:6]	SEG[19:18]
			SEG[15:0]	SEG[15:0]

Configuration bits		Remapping capability	Output pin	Function
DUTY	MUX_SEG			
			COM[3:0]	COM[3:0]
1/4	1	22x4	SEG[31:30]	SEG[31:30]
			COM[7:4]	SEG[21:18]
			SEG[15:0]	SEG[15:0]
			COM[3:0]	COM[3:0]

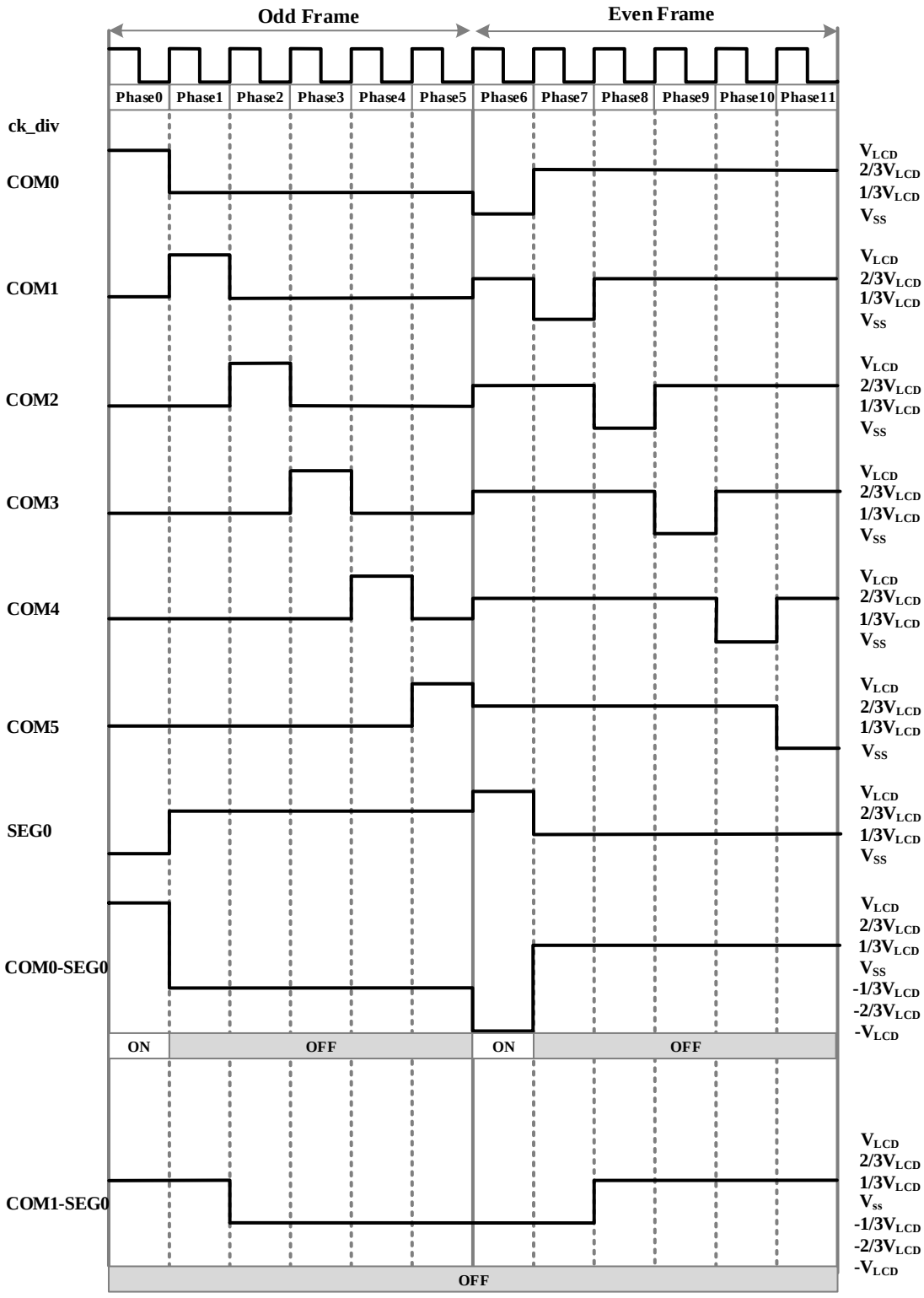
1. '–' considered invalid configuration.

17.3.9 LCD drive waveform

The LCD controller supports waveform B, with examples of commonly used configurations provided in this chapter:

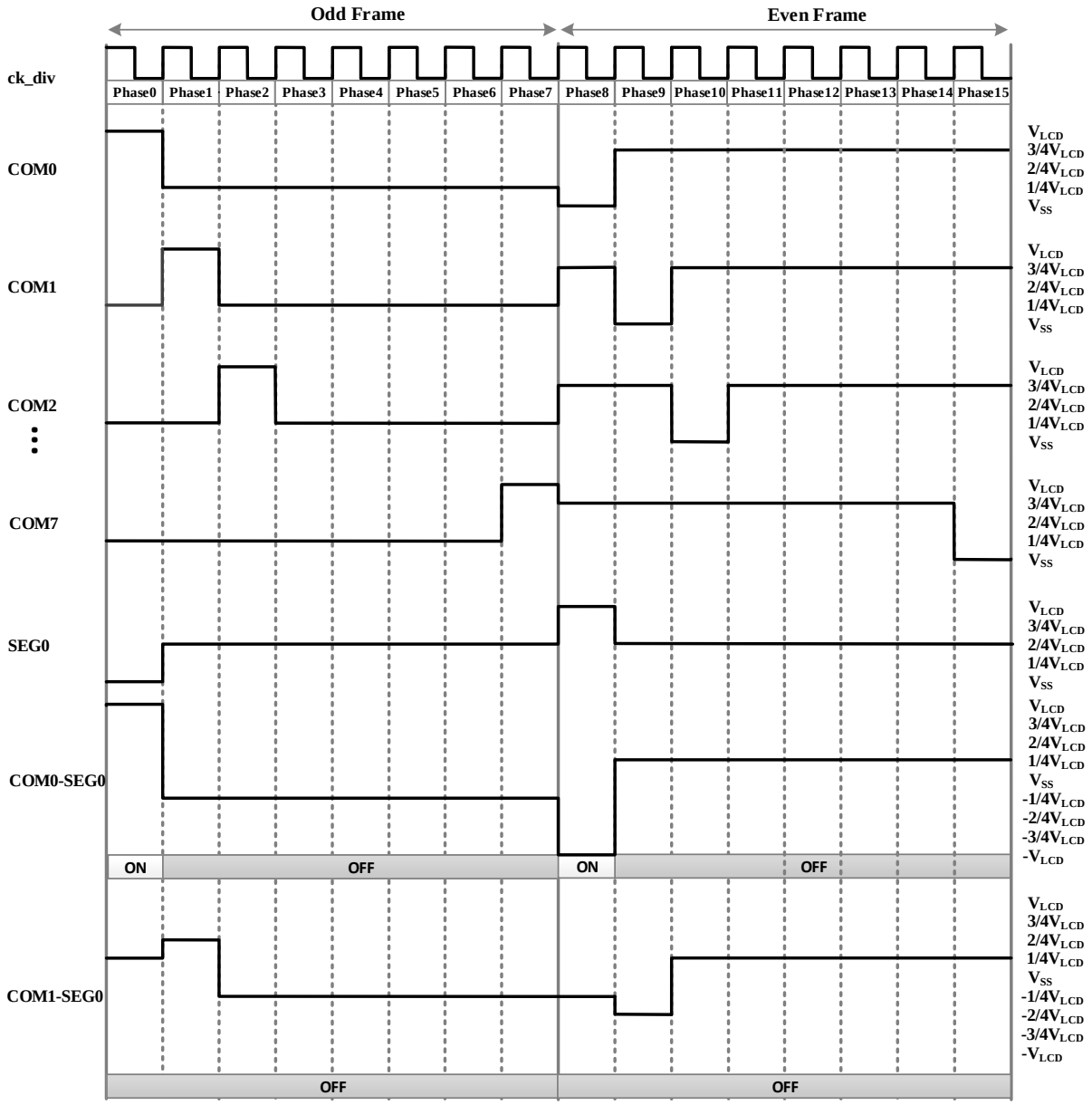
Waveform of 1/6 duty, 1/3 bias

Figure 17-5 1/6 duty, 1/3 bias



Waveform of 1/8 duty, 1/4 bias

Figure 17-6 1/8 duty, 1/4 bias



17.3.10 LCD_RAM register map

Depending on the duty configuration, the following tables illustrate the correspondence between COM pins, SEG pins, and LCD_RAM register bits.

Table 17-8 LCD_RAM register map of 4 COM

LCD_RAM	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
LCD_RAM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0
	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
LCD_RAM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1
	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
LCD_RAM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2
	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
LCD_RAM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3

LCD_RAM	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
	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
LCD_RAM4					COM2	COM2	COM2	COM2									COM1	COM1	COM1	COM1									COM0	COM0	COM0	COM0
					SEG35	SEG34	SEG33	SEG32									SEG35	SEG34	SEG33	SEG32									SEG35	SEG34	SEG33	SEG32
LCD_RAM5																									COM3	COM3	COM3	COM3				
																									SEG35	SEG34	SEG33	SEG32				

Table 17-9 LCD_RAM register map of 6 COM

LCD_RAM	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
LCD_RAM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0
	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
LCD_RAM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1
	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
LCD_RAM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2
	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
LCD_RAM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3

LCD_RAM	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
	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
LCD_RAM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4
	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
LCD_RAM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5
	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
LCD_RAM6	COM3	COM3									COM2	COM2									COM1	COM1									COM0	COM0
	SEG33	SEG32									SEG33	SEG32									SEG33	SEG32									SEG33	SEG32

LCD_RAM	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
LCD_RAM7													COM5	COM5									COM4	COM4								
													SEG33	SEG32									SEG33	SEG32								

Table 17-10 LCD_RAM register map of 8 COM

LCD_RAM	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
LCD_RAM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0	COM0
	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
LCD_RAM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1	COM1
	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
LCD_RAM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2	COM2
	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
LCD_RAM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3	COM3

LCD_RAM	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
	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
LCD_RAM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4	COM4
	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
LCD_RAM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5	COM5
	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
LCD_RAM6	COM6	COM6	COM6	COM6	COM6	COM6	COM6	COM6	COM6	COM6	COM6	COM6	COM6	COM6	COM6	COM6	COM6	COM6	COM6	COM6	COM6	COM6	COM6	COM6	COM6	COM6	COM6	COM6	COM6	COM6	COM6	COM6
	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0

LCD_RAM	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
LCD_RAM7	COM7	COM7	COM7	COM7	COM7	COM7	COM7	COM7	COM7	COM7	COM7	COM7	COM7	COM7	COM7	COM7	COM7	COM7	COM7	COM7	COM7	COM7	COM7	COM7	COM7	COM7	COM7	COM7	COM7	COM7	COM7	COM7
	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0

17.4 LCD interrupts

Table 17-11 LCD interrupts

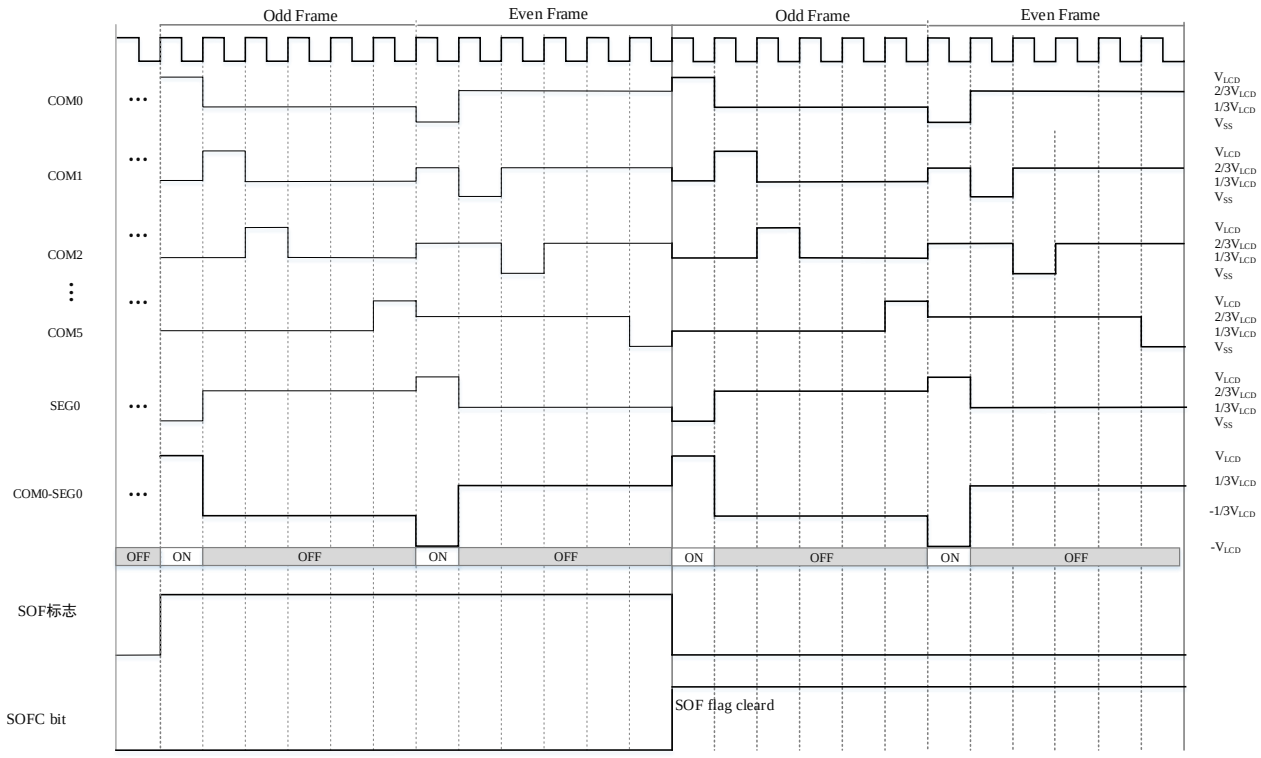
Interrupt event	Event flag	Interrupt enable control bit	Event flag/interrupt clearing method
Start of frame (SOF)	SOF	SOFIE	Write SOFC = 1

Start of frame (SOF)

The LCD start of frame flag is set by hardware at the beginning of a new frame, at the same time as the display data is updated. The SOF interrupt is executed if the SOFIE (start of frame interrupt enable) bit is set. SOF is cleared by writing the SOFC bit to 1 in the LCD_CLR register when executing the corresponding interrupt handing vector. See *Section: LCD status regsister (LCD_SR)* and *LCD clear register (LCD_CLR).*

As illustrated in Figure 17-7 (using a 1/6-duty and 1/3-bias drive waveform), the SOF flag is generated at the start of the odd-numbered frames.

Figure 17-7 Example of SOF (1/6 duty, 1/3 bias)



17.5 LCD display flowchart

Figure 17-8 Flowchart example of charge pump mode

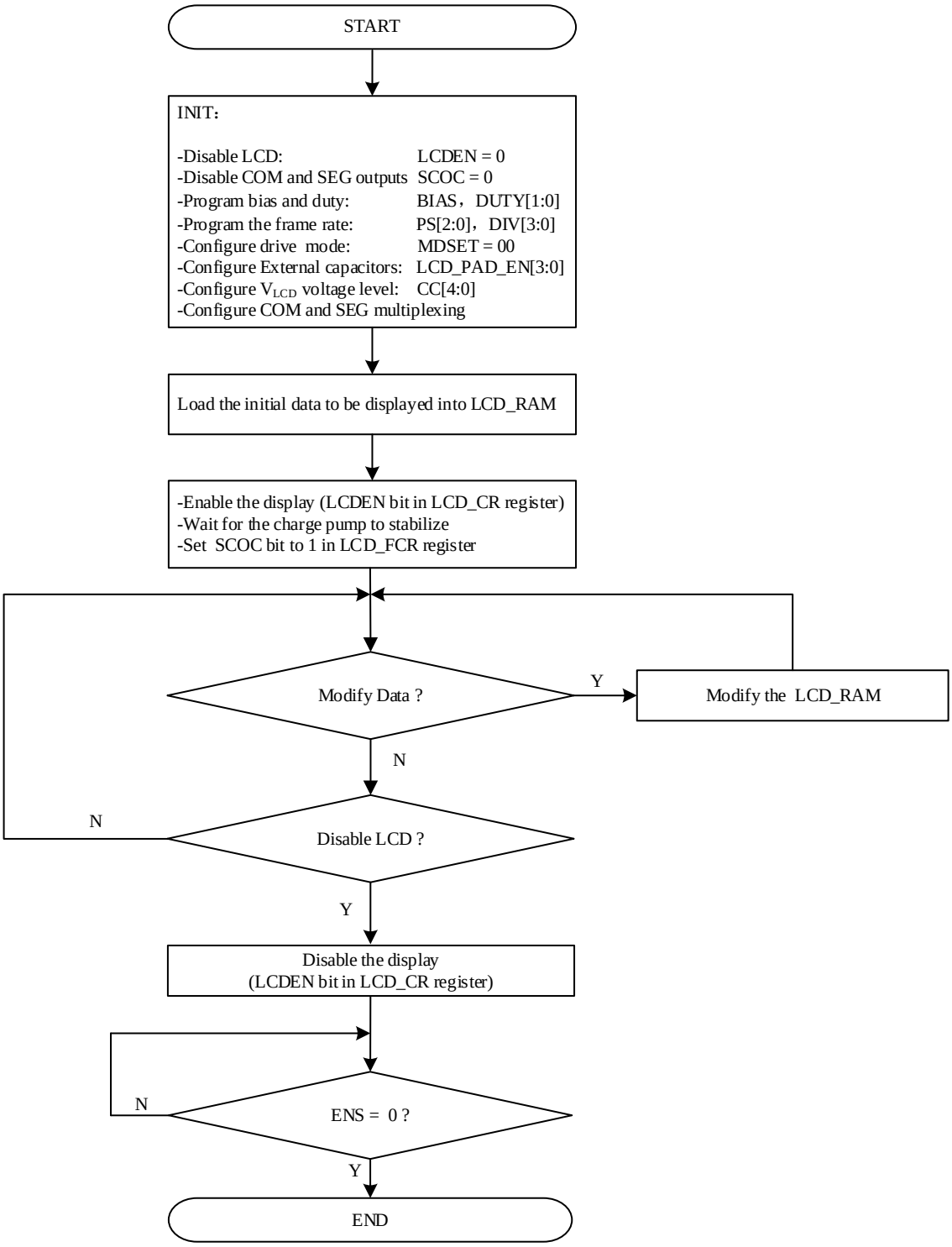


Figure 17-9 Flowchart example of external capacitor division mode

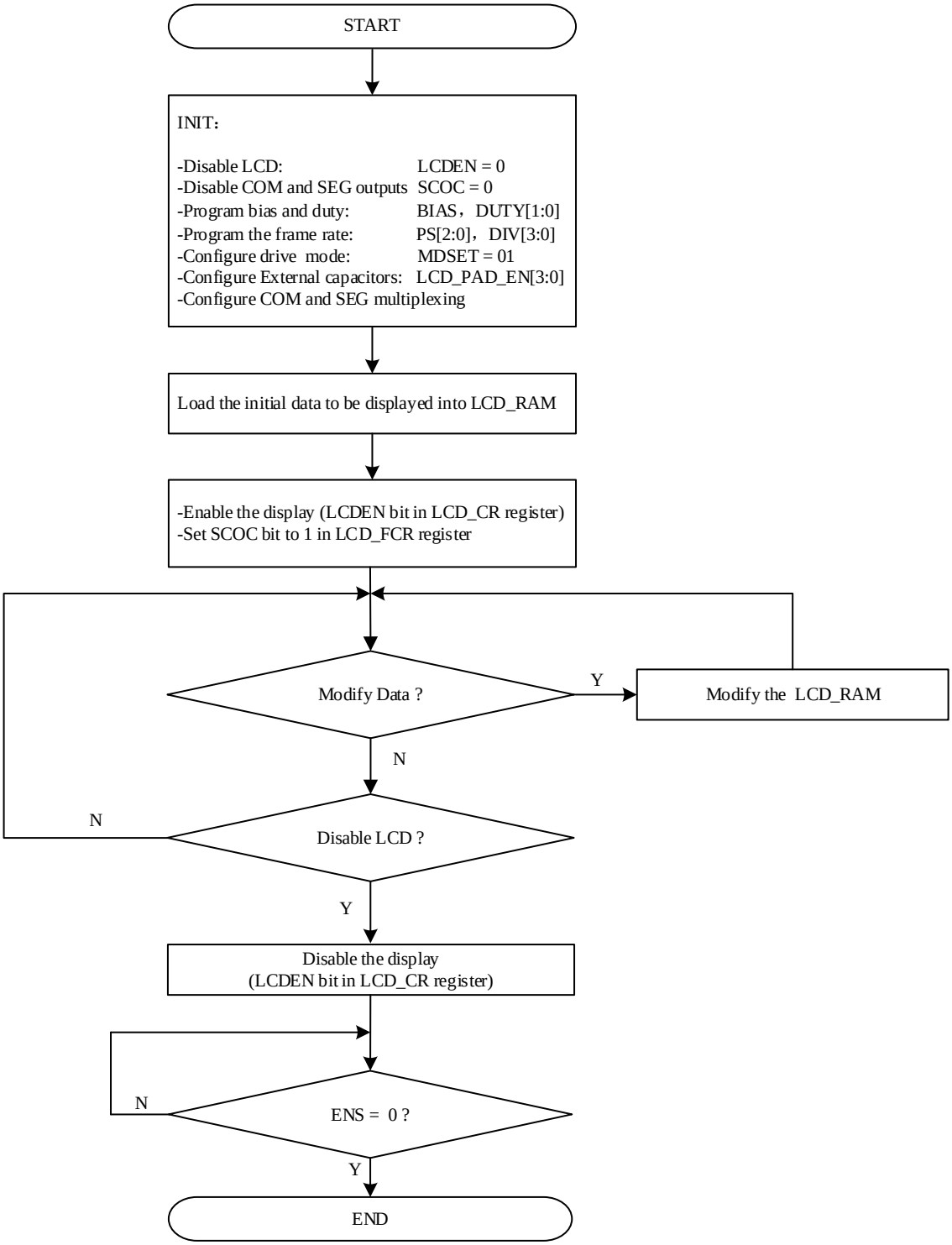
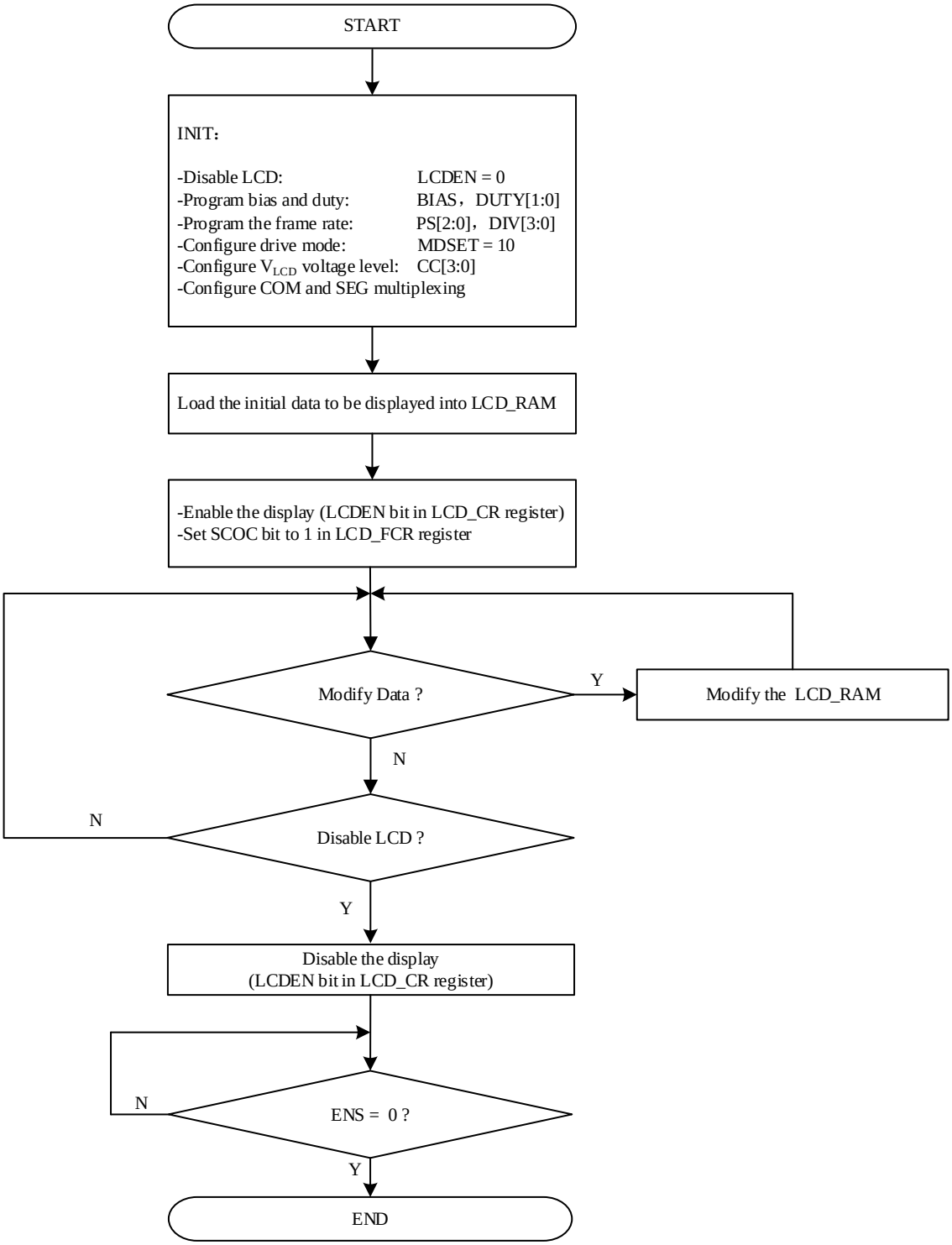


Figure 17-10 Flowchart example of internal resistor division mode



17.6 LCD Low-power modes

The LCD controller can be displayed in Sleep and Stop modes or can be fully disabled to reduce power consumption.

Table 17-12 LCD behavior in low-power modes

Mode	LCD clock source	Description
Sleep	LXTAL/RCL	The LCD is still active
Stop		

17.7 LCD registers

The LCD registers can only be accessed by words (32-bit).

Table 17-13 LCD base address

Peripheral	Base address
LCD	0x4000 2400

17.7.1 LCD control register (LCD_CR)

Address offset: 0x00

Reset value: 0x0000 0000

Note: The LCD_CR register is write-protected when the LCD is enabled (ENS bit in the LCD_SR register to 1).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.													CPDIV[2:0]		
													rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		MDSET[1:0]		LCD_PAD_EN[3:0]				MUX_SEG	Res.	BIAS	Res.	DUTY[1:0]		Res.	LCDEN
		rw	rw	rw	rw	rw	rw	rw		rw		rw	rw		rw

Bits	Name	Description
31:19	Reserved	Must be kept at reset value
18:16	CPDIV[2:0]	Charge pump clock divider 000: LCD_KCLK 001: LCD_KCLK /2 010: LCD_KCLK /4 ⋮ 110: LCD_KCLK /64 111: LCD_KCLK /128 *Note: To minimize power consumption while maintaining display quality, set the charge pump clock divider ratio to no less than 16.*
15:14	Reserved	Must be kept at reset value
13:12	MDSET[1:0]	Drive mode selection

		00: Charge pump mode 01: External capacitor division mode 10: Internal resistor division mode 11: Reserved
11	LCD_PAD_EN[3]	The VLCD pin connects to PF2 enable 0: Disable 1: Enable
10	LCD_PAD_EN[2]	The LCD_V3 pin connects to PB2 enable 0: Disable 1: Enable
9	LCD_PAD_EN[1]	The LCD_V2 pin connects to PB1 enable 0: Disable 1: Enable
8	LCD_PAD_EN[0]	The LCD_V1 pin connects to PB0 enable 0: Disable 1: Enable
7	MUX_SEG	Mux segment enable This bit is used to enable SEG pin remapping, see *Section COM and SEG multiplexing.* 0: SEG pin multiplexing disabled 1: SEG pin multiplexing enabled *Note: Set this bit to enable SEG pin multiplexing under 1/3-duty and 1/6-duty.*
6	Reserved	Must be kept at reset value
5	BIAS	Bias selector 0: Bias 1/3 1: Bias 1/4
4	Reserved	Must be kept at reset value
3:2	DUTY[1:0]	Duty selection

		00: 1/4 duty 01: 1/6 duty 10: 1/8 duty 11: Reserved (default value: 1/4 duty)
1	Reserved	Must be kept at reset value
0	LCDEN	LCD controller enable 0: LCD controller disabled 1: LCD controller enabled

17.7.2 LCD frame control register (LCD_FCR)

Address offset: 0x04

Reset value: 0x0000 0000

Note: The LCD_FCR register (except for SCOC and ANTI_POLAR bits that can be updated any time) is write-protected when the LCD is enabled (ENS bit in the LCD_SR register to 1).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CC[4:0]					ANTI_POLAR	SCOC	PS[2:0]			DIV[3:0]				BLINK[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
BLINKF[2:0]			Res.				DEAD[2:0]			PON[3:0]				SOFIE	HD
rw	rw	rw					rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Name	Description
31:27	CC[4:0]	Contrast control – MDSET[1:0] = 00: Refer to *Table: V_{LCD} voltage levels of charge pump mode* for configuration. – MDSET[1:0] = 10: Refer to *Table: V_{LCD} voltage level of internal resister division mode (only for high drive)* for configuration.
26	ANTI_POLAR	Antipolar selection when the LCD is disabled 0: All COM and SEG pins are floating 1: All COM and SEG pins are driven to Vss

25	SCOC	COM and SEG output enable The new value of SCOC bit is applied only after LCD controller is enabled. 0: All COM and SEG pins are driven to Vss 1: All COM and SEG pins are driven by LCD_RAM registers *Note: In charge pump mode, after enabling the LCD controller, set the SCOC bit to 1 and wait for the charge pump voltage to stabilize.*
24:22	PS[2:0]	PS 7-bit prescaler These bits are written by software to define the division factor of the PS 7-bit prescaler. ck_ps = LCD_KCLK/$2^{PS[2:0]}$. see *Section: Frequency generator*. 000: ck_ps = LCD_KCLK 001: ck_ps = LCD_KCLK/2 010: ck_ps = LCD_KCLK/4 ⋮ 111: ck_ps = LCD_KCLK/128
21:18	DIV[3:0]	DIV clock divider These bits are written by software to define the division factor of the DIV divider. See *Section: Frequency generator.* 0000: ck_div = ck_ps/16 0001: ck_div = ck_ps/17 0010: ck_div = ck_ps/18 ⋮ 1111: ck_div = ck_ps/31
17:16	BLINK[1:0]	Blink mode selection 00: Blink disabled 01: Blink enabled on SEG[0], COM[0] (1 pixel) 10: Blink enabled on SEG[0], all COMs (up to 8 pixels depending on the programmed duty) 11: Blink enabled on all SEGs and all COMs (all pixels)
15:13	BLINKF[2:0]	Blink frequency selection 000: $f_{ck\_div}/8$ 001: f_{ck_div} /16

		010: f_{ck_div} /32 011: f_{ck_div} /64 100: f_{ck_div} /128 101: f_{ck_div} /256 110: f_{ck_div} /512 111: f_{ck_div} /1024
12:9	Reserved	Must be kept at reset value
8:6	DEAD[2:0]	Deadtime duration These bits are written to configure the length of the deadtime between frames. During the deadtime the COM and SEG voltage levels are held at 0 V to reduce the contrast without modifying the frame rate. 000: No deadtime 001: 1 phase period deadtime 010: 2 phase period deadtime ⋮ 111: 7 phase period deadtime
5:2	PON[3:0]	Pulse ON duration These bits are written to define the duration of high drive by the ck_ps pulse, used for switching between high/low-drive modes in the internal resistor division configuration. A short ck_ps pulse will lead to lower power consumption, but displays with low drive may need a longer pulse to achieve satisfactory contrast. 0000: 0 0001: 1/ck_ps 0010: 2/ck_ps ⋮ 1111: 15/ck_ps
1	SOFIE	Start of frame interrupt enable 0: LCD Start of Frame interrupt disabled 1: LCD Start of Frame interrupt enabled
0	HD	High drive enable 0: Permanent high drive disabled

1: Permanent high drive enabled

17.7.3 LCD status register (LCD_SR)

Address offset: 0x08

Reset value: 0x0000 0000

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

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

Bits	Name	Description
31:2	Reserved	Must be kept at reset value
1	SOF	Start of frame flag This bit is set by hardware at the beginning of a new frame, at the same time as the display data is updated. It is cleared by writing a 1 to the SOFC bit in the LCD_CLR register. The bit clear has priority over the set. 0: No event 1: Start of Frame event occurred. An LCD SOF interrupt is generated if the SOFIE bit is set.
0	ENS	LCD enabled status 0: LCD Controller disabled 1: LCD Controller enabled *Note: The ENS bit is set immediately when the LCDEN bit in the LCD_CR goes from 0 to 1. It is cleared by hardware when the LCDEN bit goes from 1 to 0.*

17.7.4 LCD clear register (LCD_CLR)

Address offset: 0x0C

Reset value: 0x0000 0000

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

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

Bits	Name	Description
31:2	Reserved	Must be kept at reset value
1	SOFC	Start of frame flag clear 0: No effect 1: Clear SOF flag
0	Reserved	Must be kept at reset value

17.7.5 LCD display memory (LCD_RAM)

Address offset: 0x14 to 0x38

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEGMENT_DATA[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
SEGMENT_DATA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Name	Description
31:0	SEGMENT_DATA[31:0]	Each bit corresponds to one pixel of the LCD display 0: Pixel inactive 1: Pixel active *Note: The correspondence between COM pins, SEG pins, and LCD_RAM register bits is illustrated in Section: LCD_RAM register map.*

18 True random number generator (TRNG)

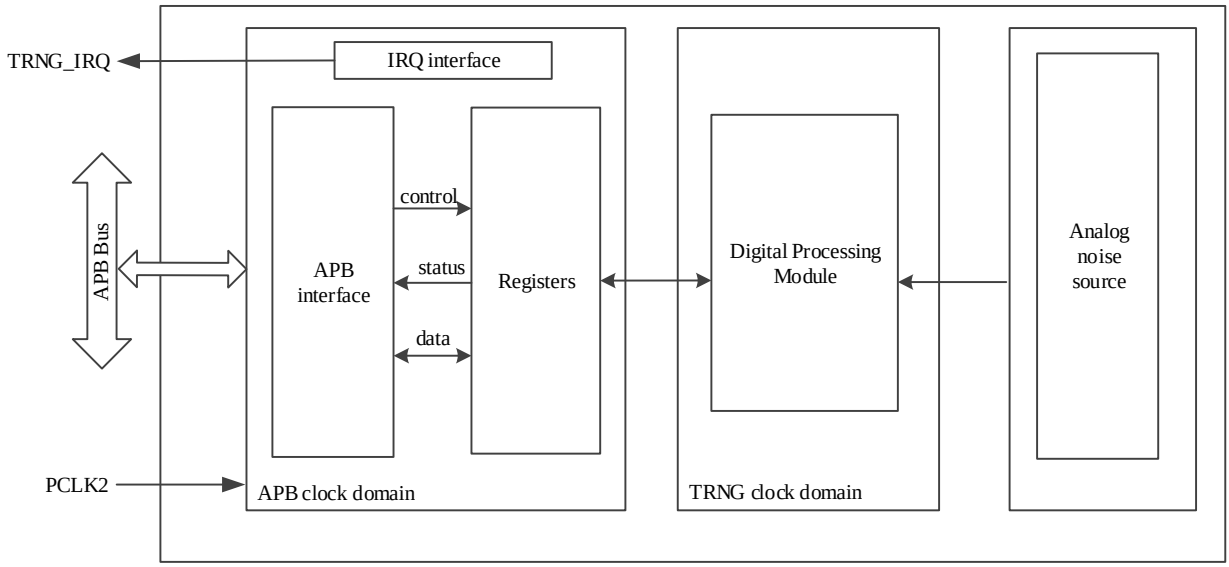
18.1 Introduction

The TRNG delivers 32-bit true random numbers, it is composed of an entropy source (analog) and a digital processing component.

18.2 TRNG main features

18.2.1 TRNG block diagram

Figure 18-1 TRNG block diagram



18.2.2 TRNG internal signals

Table 18-1 TRNG internal signals

Signal name	Signal type	Description
PCLK2	Input	APB2 clock
TRNG_IRQ	Output	TRNG global interrupt request

18.2.3 Random number generation

The TRNG consists of an analog true random source and a digital post-processing module. The entropy generated by the analog source is sampled and the processed through the digital module to produce cryptographically secure random numbers.

TRNG operation

The TRNG_DR register provides 32-bit random number output. After enabling the

TRNG clock, the system should poll the RDY flag in the TRNG_SR register. When this flag is set to 1, the valid random number can be retrieved from the TRNG_DR register through a single read operation.

Implementation procedure:

1) Enable TRNG clock, by setting the TRNGEN bit to 1 in the RCC_APB2EN register.
2) Select TRNG mode by configuring the TRNG_SEL bit in the TRNG_CR1 register to set the desired entropy generation method.
3) Poll the RDY flag in the TRNG_SR register until the flag is asserted.
4) Read the random number from the TRNG_DR register.
5) Repeat steps 3~4 when more random numbers need to be generated, otherwise proceed to step 6.
6) Disable TRNG clock by setting the TRNGEN bit to "0" in the RCC_APB2EN register.

18.3 TRNG interrupts

The relevant control bits for interrupts are shown in the following table:

Table 18-2 TRNG interrupt requests

Interrupt event	Event flag	Enable control bit	Interrupt clear method
Ready flag	RDY	RDYIE	Read data from the TRNG_DR register

18.4 TRNG registers

The TRNG registers can only be accessed by words (32-bit).

Table 18-3 TRNG base address

Peripheral	Base address
TRNG	0x4001 8000

18.4.1 TRNG control register 1 (TRNG_CR1)

Address offset: 0x00

Reset value: 0x0000 0000

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

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

Bits	Name	Description
31:5	Reserved	Must be kept at reset value
4	TYPE_SEL	Random number type selection 0: True random number 1: Pseudo random number
3:1	Reserved	Must be kept at reset value
0	RDYIE	Data ready interrupt enable 0: Disable 1: Enable

18.4.2 TRNG status register (TRNG_SR)

Address offset: 0x08

Reset value: 0x0000 0000

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

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

Bits	Name	Description
31:1	Reserved	Must be kept at reset value
0	RDY	Data Ready RDY bit is set to 0 when read. 0: No random data is available 1: Random data is available

18.4.3 TRNG data register (TRNG_DR)

Address offset: 0x10

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OUT[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
OUT[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits	Name	Description
31:0	OUT[31:0]	32-bit random value

19 AES hardware accelerator (AES)

19.1 Introduction

The AES hardware accelerator (AES) encrypts or decrypts data. It is defined in Federal information processing standards (FIPS).

The AES supports ECB mode, for key sizes of 128 bits.

The AES supports DMA single transfers for incoming and outgoing data (two DMA channels required).

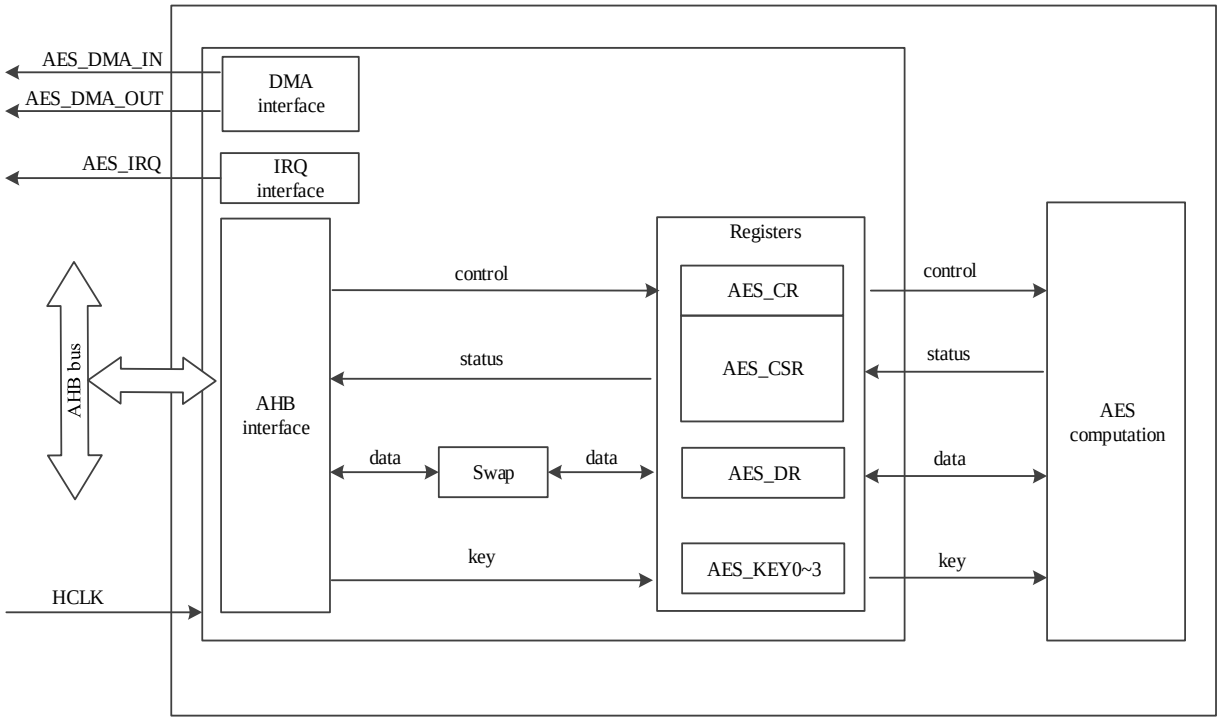
19.2 AES main features

- Support Electronic codebook (ECB) mode
- Support for cipher key lengths of 128-bit
- Data-swapping logic to support 1-, 8- or 16-bit data
- Supports DMA transfer

19.3 AES functional description

19.3.1 AES block diagram

Figure 19-1 AES block diagram



19.3.2 AES internal signals

AES peripheral internal signals as shown in the following table:

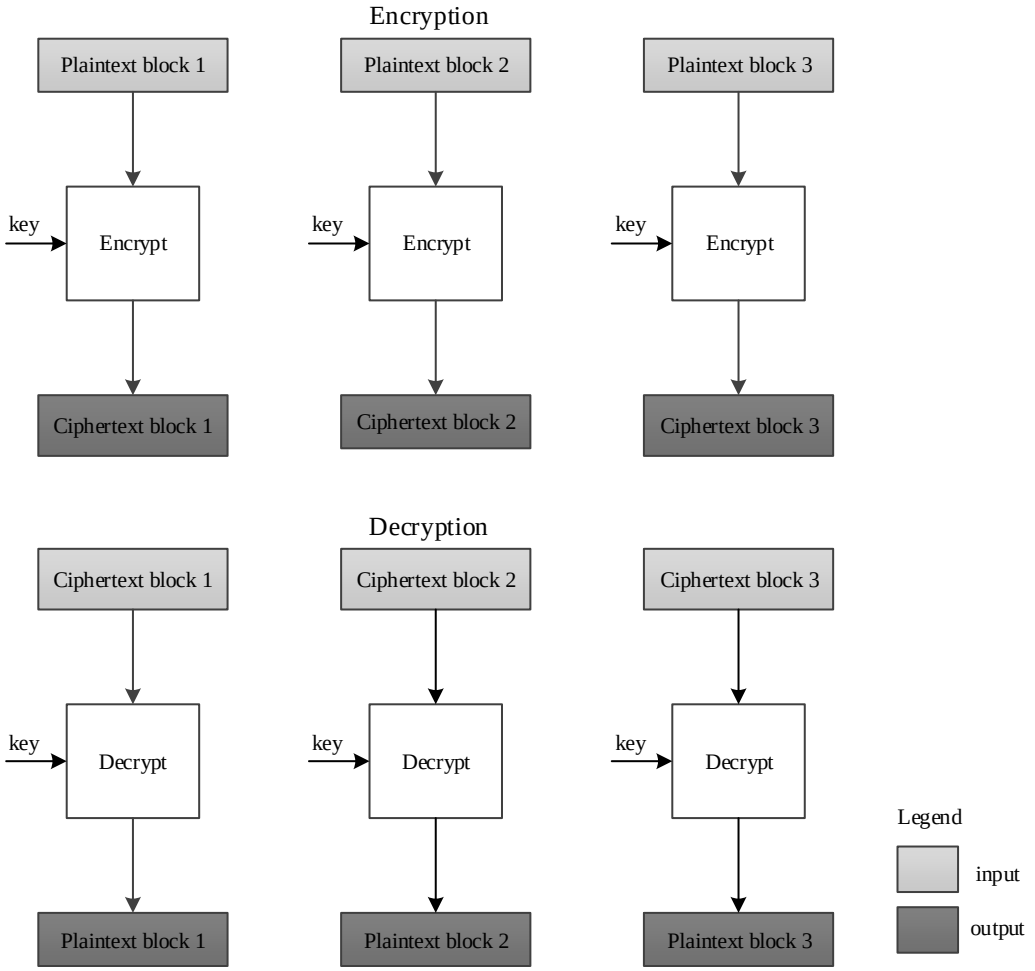
Table 19-1 AES internal input/output signals

Signal name	Signal type	Description
HCLK	Input	AHB bus clock
AES_IRQ	Output	AES interrupt request
AES_DMA_IN	Input/output	Input DMA single request/acknowledge
AES_DMA_OUT	Input/output	Output DMA single request/acknowledge

19.3.3 AES cryptographic unit

The AES implements 128-bit block cipher operations in ECB mode, supporting both encryption and decryption with 128-bit key length.

Figure 19-2 ECB encryption and decryption principle

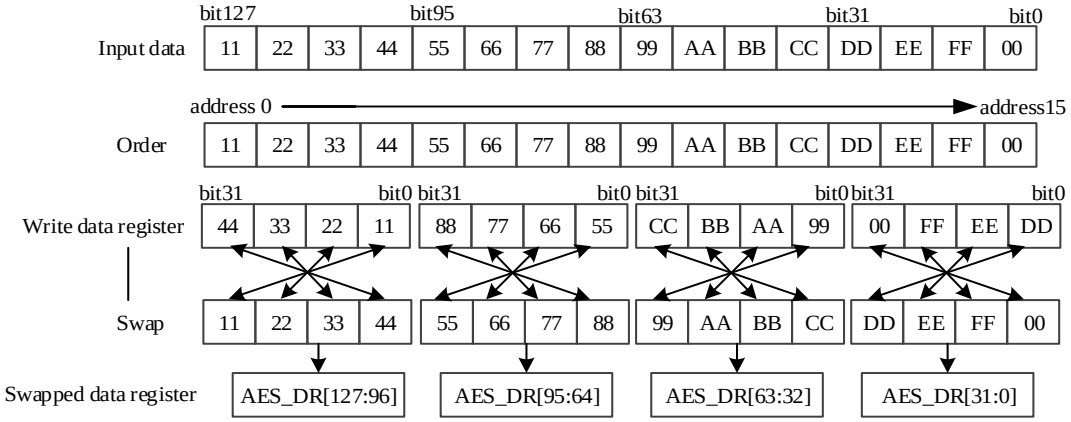


ECB divides message into 128-bit data blocks and each block is encrypted or decrypted separately.

19.3.4 Data swapping

The AES peripheral can be configured to perform a bit-, a byte-, a half-word-, or no swapping on the input data word in the AES_DR register. The data swap type is selected through the DATATYPE[1:0] bit field of the AES_CR register. This bit field can be configured based on the bit order of the input data, for example: for a 128-bit data block with data 0x112233445566778899AABBCCDDEEFF00, setting the DATA_TYPE[1:0] bit field to “10” swaps the data by byte, and the data storage order is shown in the following figure.

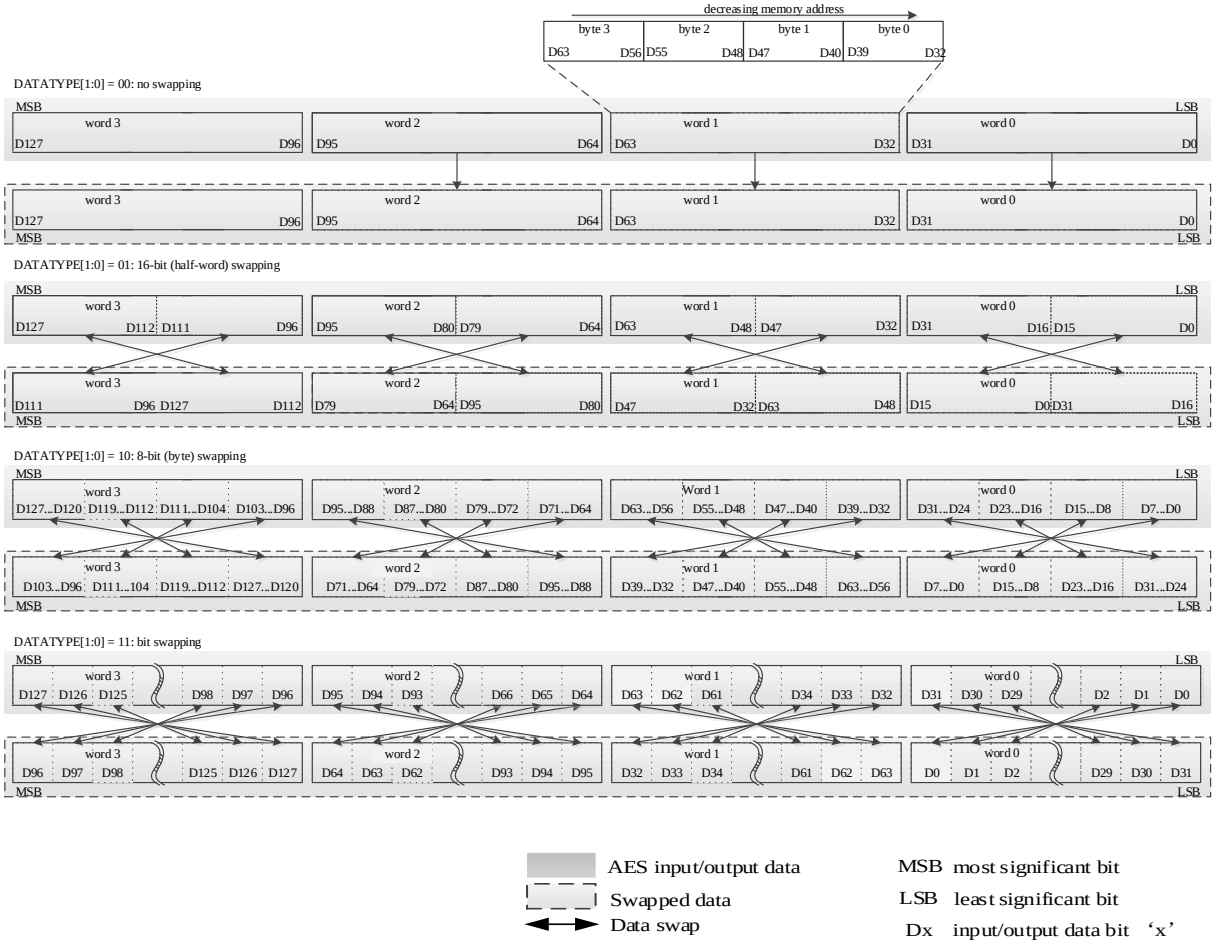
Figure 19-3 Data Storage



During encryption and decryption operations, the hardware exchanges data according to the data exchange type and inputs it into the AES_DR data register, then performs the operation. After the operation is completed, the result data is output to the AES_DR data register. When reading the AES_DR data register, the data is exchanged and then output.

For different data swap types, Figure19-4 shows the construction of data P127..0, from the input/output data entered through the AES_DR register.

Figure 19-4 128-bit block construction with respect to data swap



19.3.5 Operation description

The 128-bit data block is input into the AES through the AES_DR data register, and the result data generated after the encryption/decryption operation is output to the AES_DR data register. During the encryption/decryption operation, reading this register yields 0. Data in the AES data register is stored in little-endian format within the word, and in big-endian format between words. The data order of the 128-bit data corresponding to the data register is shown in the following table.

Table 19-2 Data register data order

AES_DR			
bit[127:96]	bit[95:64]	bit[63:32]	bit[31:0]

The 128-bit key is input into the AES through four 32-bit AES_KEY0~3 key registers. The key registers are not affected by the data swap function controlled by the DATA_TYPE[1:0] bit field of the AES_CR control register, and the key registers are write-only, not readable. Data is stored in little-endian format both within and between the key registers. The data order of the key registers corresponding to the 128-bit key

is shown in the following table.

Table 19-3 Key register data order

AES_KEYR3	AES_KEYR2	AES_KEYR1	AES_KEYR0
bit[127:96]	bit[95:64]	bit[63:32]	bit[31:0]

The MOD_SEL bit in the AES_CR register is used to select whether the AES operation is for encryption or decryption. If the AES operation is switched from encryption to decryption or from decryption to encryption, the key must be reloaded; otherwise, it will result in incorrect operation results.

After performing four write operations on the AES_DR register, the AES data encryption and decryption operations begin, and the BUSY bit in the AES_CSR register is set to 1. Upon completion of the operations, the DONE bit in the AES_CSR register control status register is set to 1, and the BUSY bit is automatically cleared to 0. If the CCIE bit in the AES_CSR register is set to 1, setting the DONE bit to 1 will trigger an operation completion interrupt. The DONE bit is cleared by software upon setting the CCFC bit or reading the AES_DR register 4 times.

AES can perform encryption and decryption of data via DMA. When encrypting or decrypting data in DMA mode, it is necessary to activate both the DMA input and output channels, See *DMA data transfer*.

AES encryption and decryption must be initialized, through the following sequence:

1) Enable the AES clock.
2) Configure the AES mode, by programming the MODE_SEL bit field of the AES_CR register.
3) Configure the data type with the DATA_TYPE[1:0] bit field in the AES_CR register.

Data can be processed through flag polling or interrupt from the AES peripheral, through the following sequence:

1) Initialize AES.
2) Write a symmetric key into the AES_KEYR0~3 registers.
3) Write four input data words into the AES_DR register.
4) Wait until the status flag DONE is set in the AES_CSR register, then read the four data words from the AES_DR register.
5) Repeat the step 4) until the payload is entirely processed.
6) If a key change is required, proceed to step 2); otherwise, continue operation and

proceed to step 3).

7) Disable the AES clock.

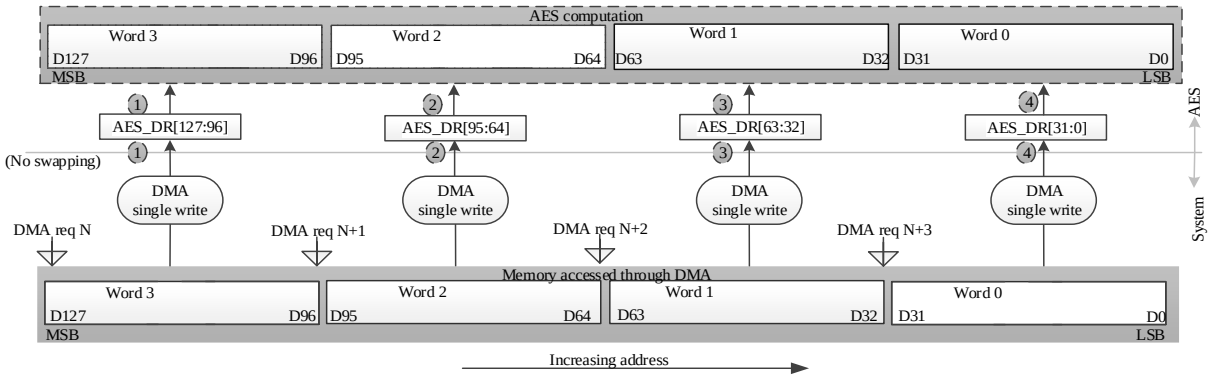
19.3.6 DMA data transfer

The AES peripheral provides an interface to connect to the DMA (direct memory access) controller. The DMA operation is controlled through the DMAIN_EN or DMAOUT_EN bit of the AES_CR register.

Data input using DMA

Setting the DMAIN_EN bit of the AES_CSR register enables DMA writing into AES. Transmitting a 128-bit (4-word) input data block will generate four DMA requests, as shown in the following figure:

Figure 19-5 DMA transfer of a 128-bit data block during input phase



DMA input channel configuration is shown in the following table:

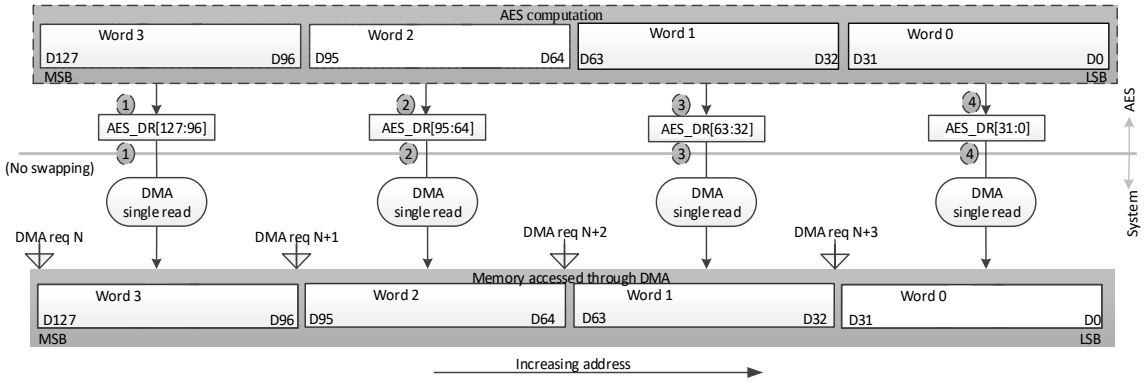
Table 19-4 DMA channel configuration for memory-to-AES data transfer

Item	Configuration
Transfer type	Block
Transfer size	Message length: a multiple of 128 bits
Source/Destination transfer width	32-bit words
Source address increment	After each 32-bit transfer
Destination address increment	No increment
Source address	Memory
Destination address	Fixed address of the AES_DR register

Data output using DMA

Setting the DMAOUT_EN bit of the AES_CSR register enables DMA writing into AES. Transmitting a 128-bit (4-word) output data block will generate four DMA requests, as shown in the following figure:

Figure 19-6 DMA transfer of a 128-bit data block during output phase



DMA output channel configuration, see table below:

Table 19-5 DMA channel configuration for AES-to-memory data transfer

Item	Configuration
Transfer type	Block
Transfer size	Message length: a multiple of 128 bits
Source/Destination transfer width	32-bit words
Source address increment	No increment
Destination address increment	After each 32-bit transfer
Source address	Fixed address of the AES_DR register
Destination address	Memory

DMA operation

DMA transfer through flag polling or interrupt from the AES peripheral, through the following sequence:

1) Initialize the DMA data input channel and enable it, See *Direct memory access controller (DMA)*.
2) Initialize the DMA data output channel and enable it, See *Direct memory access controller (DMA)*.
3) Initialize AES.
4) Write a symmetric key into the AES_KEYR0~3 registers.
5) Enable AES DMA transfer by setting the DMAIN_EN and MAOUT_EN bit to "1" in the AES_CSR register.
6) Wait until the status flag TF1 is set in the DMA_SR register.
7) Disable AES DMA transfer by setting the DMAIN_EN and MAOUT_EN bit to "0" in the AES_CSR register.
8) Disable data input and data output DMA channel.

9) Disable the AES clock.

19.4 AES interrupts

The AES interrupt generated when computation is completed (DONE bit).

Table 19-6 AES interrupt requests

Interrupt event	Event flag	Enable bit	Interrupt clear method
Computation completed	DONE	CCIE	Setting the CCFC bit in the AES_CSR register or read the AES_DR register 4 times

19.5 AES registers

The AES registers can only be accessed by words (32-bit).

Table 19-7 AES base address

Peripheral	Base address
AES	0x4002 6000

19.5.1 AES control register (AES_CR)

Address offset: 0x00

Reset value: 0x0000 0000

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

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

Bits	Name	Description
31:8	Reserved	Must be kept at reset value
7	MOD_SEL	AES operating mode 0: Encryption 1: Decryption
6:5	DATA_TYPE[1:0]	Data type selection This bit field defines the format of data written in the AES_DR register or read from the AES_DR register. 00: None 01: Half-word (16-bit) 10: Byte (8-bit) 11: Bit
4:0	Reserved	Must be kept at reset value

19.5.2 AES control status register (AES_CSR)

Address offset: 0x04

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					CCFC	Res.	CCIE	BUSY	DMAOUT_EN	DMAIN_EN	DONE	Res.			
					w		rw	r	rw	rw	r				

Bits	Name	Description
31:11	Reserved	Must be kept at reset value
10	CCFC	Computation completed clear flag DONE bit will be cleared when the flag is set to 1.
9	Reserved	Must be kept at reset value
8	CCIE	Computation completed interrupt enable This bit enables or disables the AES interrupt generation when DONE bit is set. 0: Disable 1: Enable
7	BUSY	Busy This flag indicates whether AES is idle or busy during encryption phase. The flag is set by hardware; it is cleared when computation completed. 0: Idle 1: Busy
6	DMAOUT_EN	DMA output enable This bit enables/disables data transferring with DMA, in the output phase. 0: Disable 1: Enable
5	DMAIN_EN	DMA input enable This bit enables/disables data transferring with DMA, in the input

		phase. 0: Disable 1: Enable
4	DONE	Computation completed flag The flag is set by hardware. Upon the flag setting, an interrupt is generated if enabled through the CCIE bit. It is cleared by software upon setting the CCFC bit or reading the AES_DR register 4 times. 0: Not completed 1: Completed
3:0	Reserved	Must be kept at reset value

19.5.3 AES key register 0 (AES_KEYR0)

Address offset: 0x10

Reset value: 0x0000 0000

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

Bits	Name	Description
31:0	KEY[31:0]	Cryptographic key, bits[31:0], write only.

19.5.4 AES key register 1 (AES_KEYR1)

Address offset: 0x14

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[63:48]															
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
KEY[47:32]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits	Name	Description
31:0	KEY[63:32]	Cryptographic key, bits[63:32], write only.

19.5.5 AES key register 2 (AES_KEYR2)

Address offset: 0x18

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[95:80]															
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
KEY[79:64]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits	Name	Description
31:0	KEY[95:64]	Cryptographic key, bits[95:64], write only.

19.5.6 AES key register 3 (AES_KEYR3)

Address offset: 0x1C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[127:112]															
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
KEY[111:96]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits	Name	Description
31:0	KEY[127:96]	Cryptographic key, bits[127:96], write only.

19.5.7 AES data register (AES_DR)

Address offset: 0x30

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[x+31:x+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
DATA[x+15:x]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Name	Description
31:0	DATA[x+31:0]	128-bit input/output data Write four input data words into the AES_DR register. Wait until the status flag DONE is set in the AES_CSR register, then read the four data words from the AES_DR register. Upon each write or read, the data are handled by the data swap block according to the DATA_TYPE[1:0] bit field. The substitution for “x”, from the first to the fourth write operation, is: 96, 64, 32, and 0. In other words, data from the first to the fourth write operation are: AES_DR[127:96], AES_DR[95:64], AES_DR[63:32], and AES_DR[31:0].

20 General-purpose timers (TIM3/4/5)

20.1 Introduction

The general-purpose timers consist of a 16-bit auto-reload counter driven by a programmable prescaler. They may be used for a variety of purposes, including measuring the pulse lengths of input signals (input capture) or generating output waveforms (output compare, one-pulse and PWM).

The timers are linked together internally for timer synchronization.

20.2 TIM3/4/5 main features

- 16-bit up, down, up/down auto-reload counter
- 16-bit programmable prescaler
- Up to 4 independent channels for:
  - Input capture
  - Output compare
  - PWM generation
  - One-pulse mode output
- Triggered by external signal, the timers can be started, stopped and initialized
- ADC can be triggered periodically and several timers can be connected
- Continuous transmission can be achieved through the DMA burst mode
- Encoder mode

20.3 TIM3/4/5 functional description

20.3.1 TIM3/4/5 Block diagram

Figure 20-1 TIMx block diagram (x = 3)

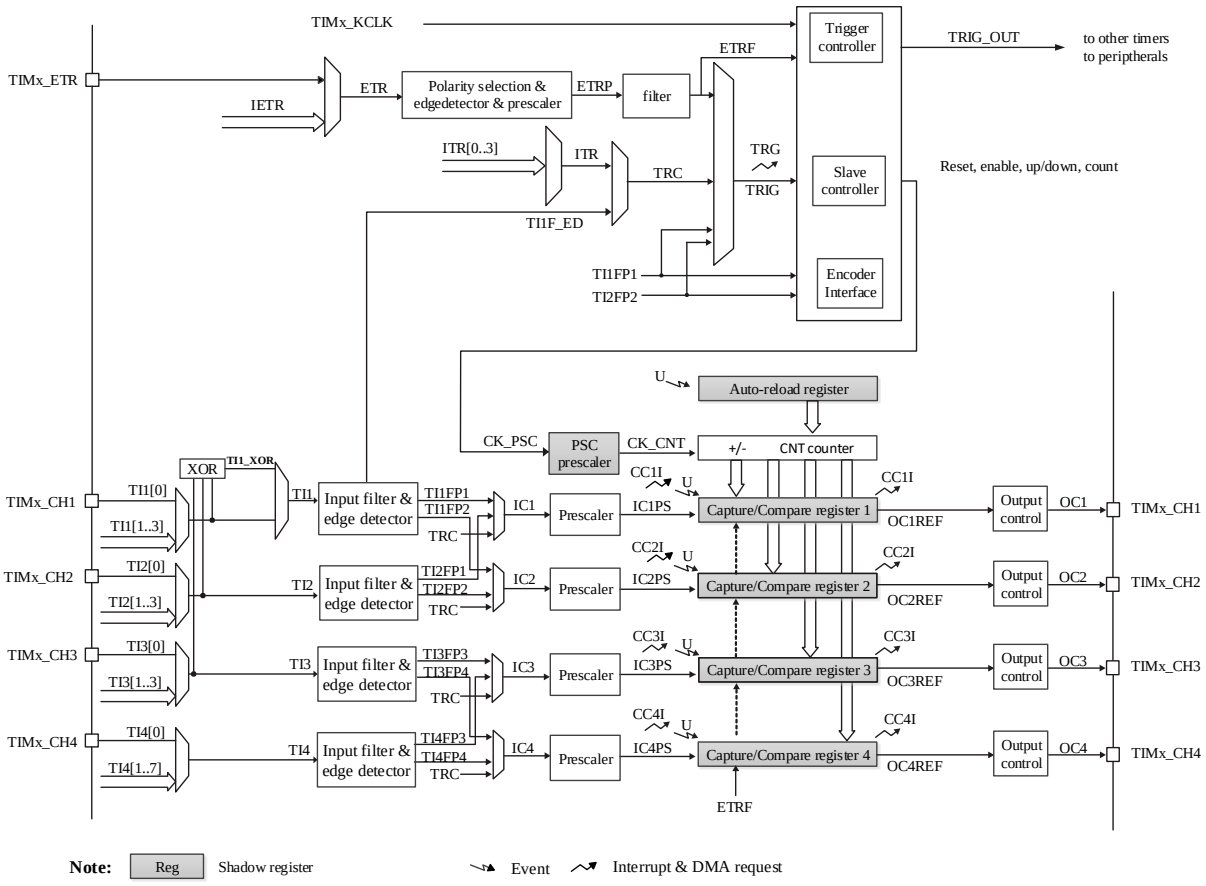
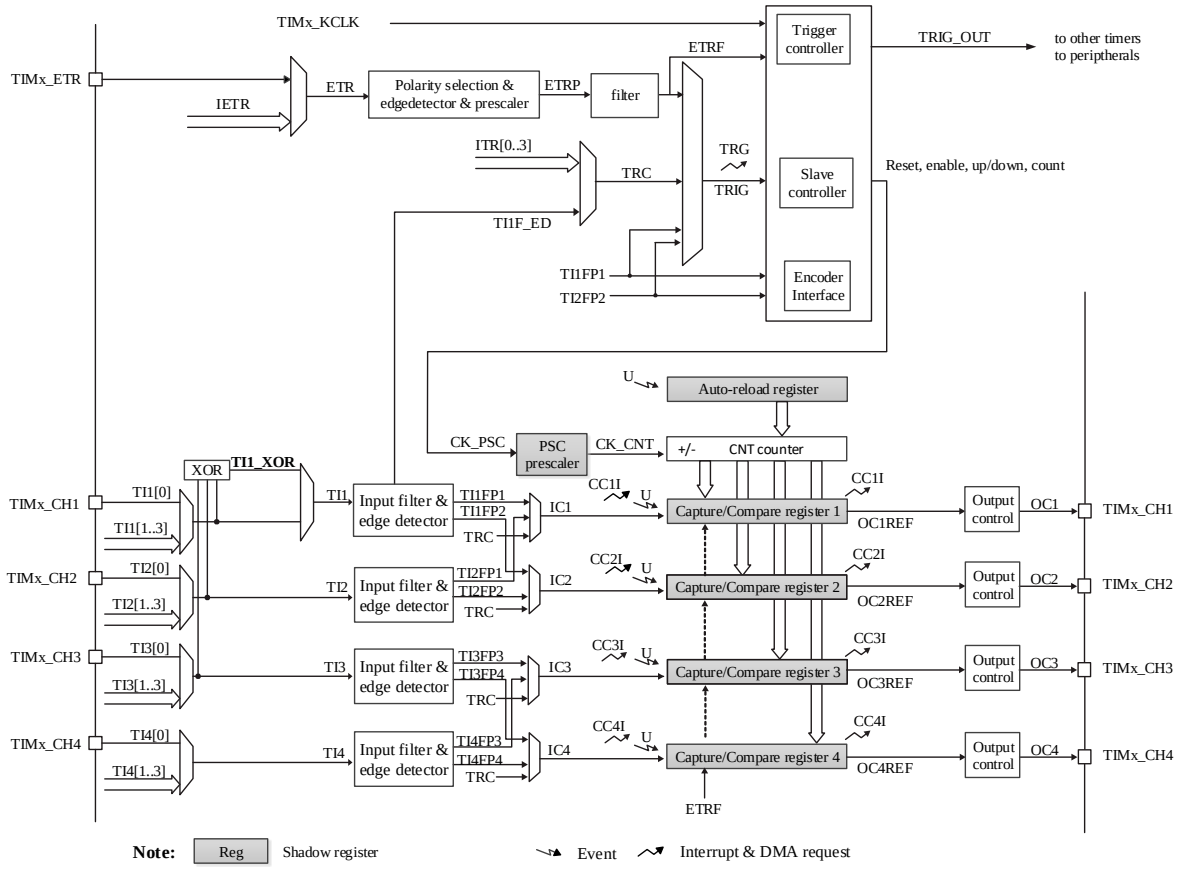


Figure 20-2 TIMx block diagram (x = 4, 5)



20.3.2 TIM3/4/5 pins and internal signals

Table 20-1 TIMx internal signals

Signal name	Signal type	Description
ITR[0…3]	Inputs	Internal trigger input
IETR	Inputs	Internal ETR trigger input
TRIG_OUT	Outputs	Internal trigger output

20.3.3 Prescaler

The prescaler can divide the counter clock frequency by any factor between 1 and 65536. The PSC[15:0] bits in the TIMx_PSC register are used to configure the prescaler value. The actual prescaler factor is PSC[15:0] + 1.

It can be changed on the fly as the TIMx_PSC register is buffered. The new prescaler ratio is taken into account at the next update event.

The following figures show some examples of the counter behavior when the prescaler ratio is changed on the fly.

Figure 20-3 Counter timing diagram with prescaler division change from 1 to 2

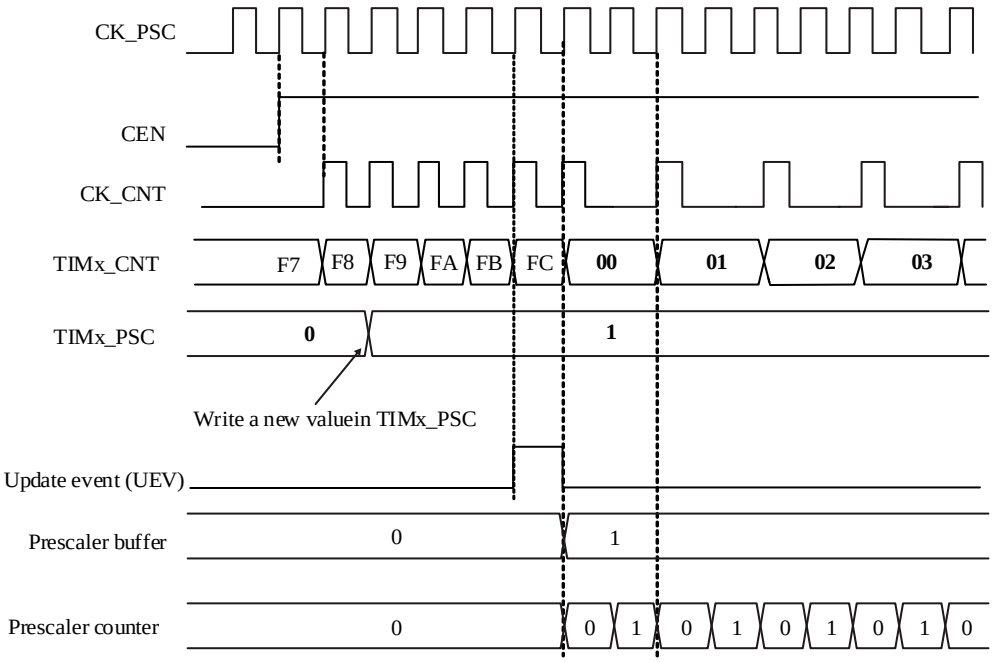
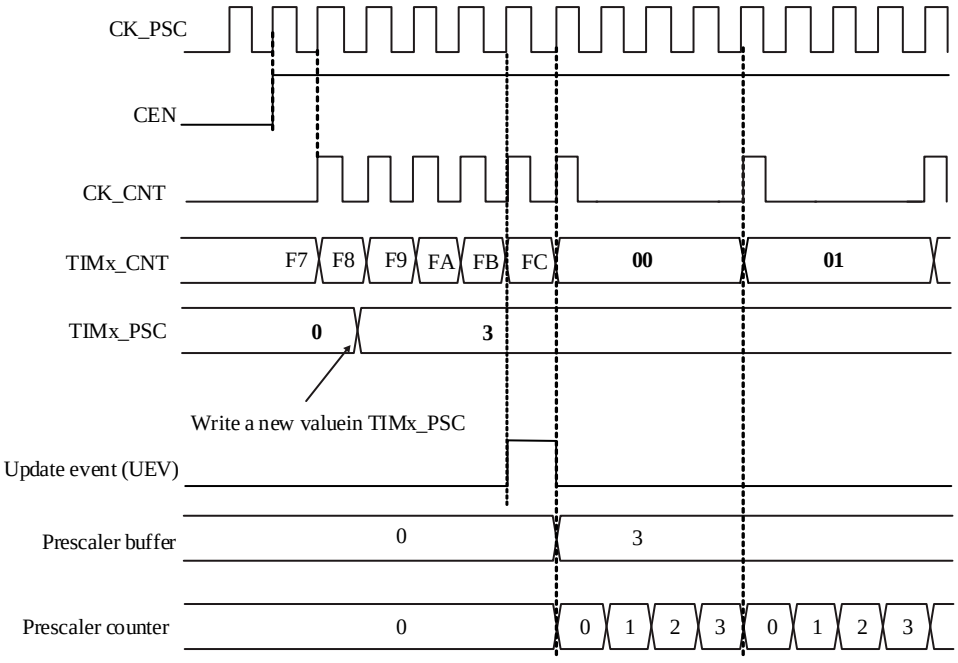


Figure 20-4 Counter timing diagram with prescaler division change from 1 to 4



20.3.4 Counter modes

TIMx has an internal 16-bit counter. The counter can count up, down or both up and down.

The counter configuration supports dynamic modification, and the TIMx_CNT register, the TIMx_ARR register and the TIMx_PSC register can be read and written while the counter is running.

The ARR[15:0] bit field in the TIMx_ARR register can be used to configure the auto-reload value, which has a buffering function (shadow register).

If the APRE bit in the TIMx_CR1 register is '0', the update of the shadow register is not affected by the updte event, and the value of the TIMx_ARR register is updated immediately. If the APRE bit is set, the value in the TIMx_ARR register is updated to the shadow register when a UEV event occurs.

The UEV event can be disabled by software by setting the UDIS bit in the TIMx_CR1 register. This avoids updating the shadow registers while writing new values into the preload registers.

If the URS (update request selection) bit in the TIMx_CR1 register is set, setting the UG bit generates an update event UEV, and both the counter and the pre-division counter will be initialized, but the UIF flag is not set.

Select the update interrupt (UEV) sources by the URS bit:

- 0: Any of the following events generate an UEV if enabled:
  - Counter overflow/underflow
  - Setting the UG big in the TIMx_EVTG register
  - Update generation through the slave mode controller
- 1: Only counter overflow/underflow generates an UEV if enabled.

Upcounting mode

In upcounting mode, the counter counts from 0 to the auto-reload value (contents of the TIMx_ARR register), then generates a counter overflow event, the UIF flag of the TIMx_SR register will be automatically set to 1, and the counter will be cleared to 0 and starts counting again.

When an update event occurs, all the registers are updated and the update flag (UIF bit in the TIMx_SR register) is set:

- The auto-reload shadow register is updated with the preload value (TIMx_ARR register)
- The buffer of the prescaler is reloaded with the preload value (content of the TIMx_PSC register)

The following figures show some examples of the counter behavior for different clock frequencies when TIMx_ARR = 0x36:

Figure 20-5 Counting timing diagram, overflow event when the internal clock divided by 1

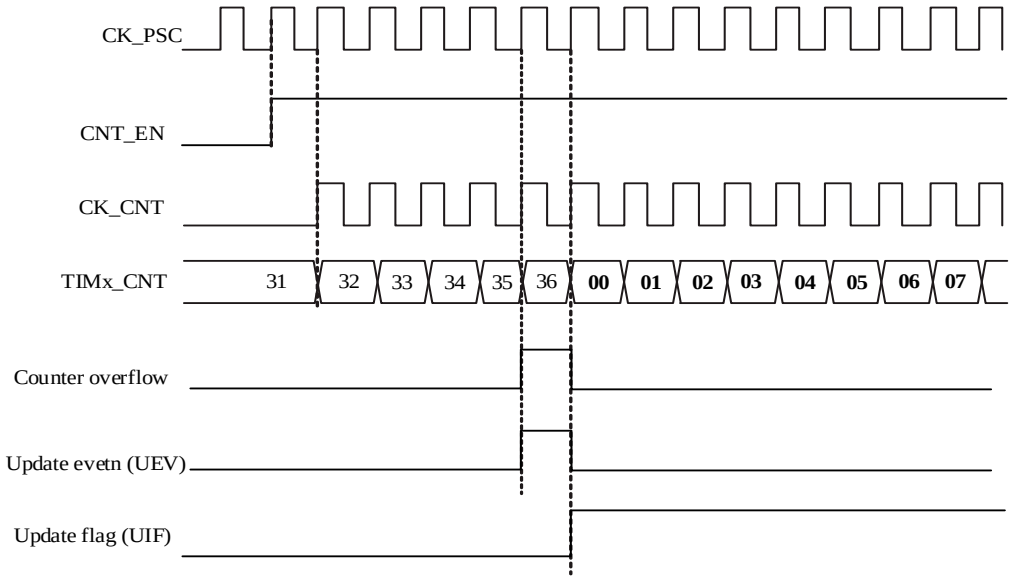


Figure 20-6 Counting timing diagram, overflow event when the internal clock divided by 4

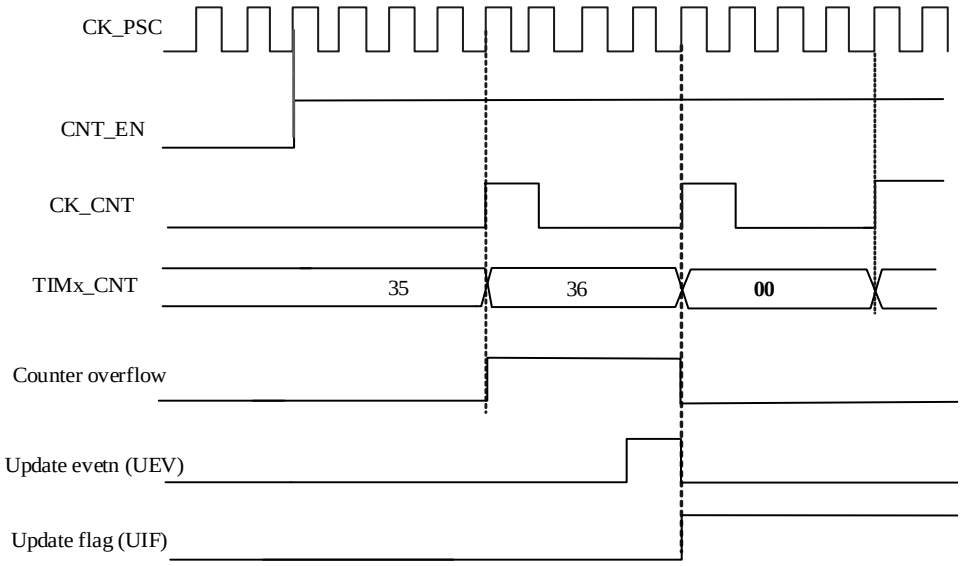


Figure 20-7 Counter timing diagram, update event when ARPE = 0

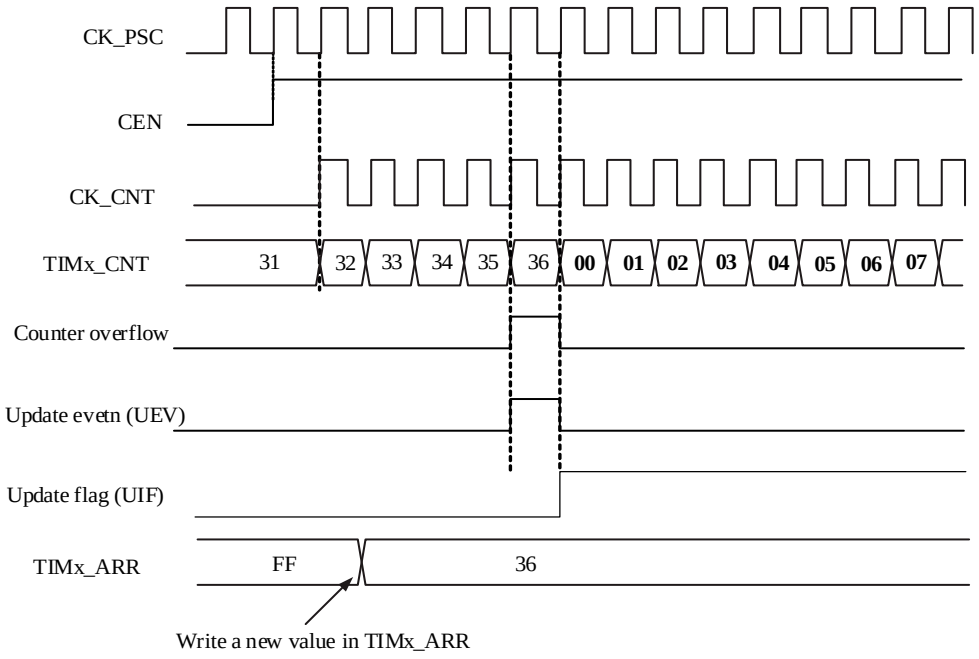
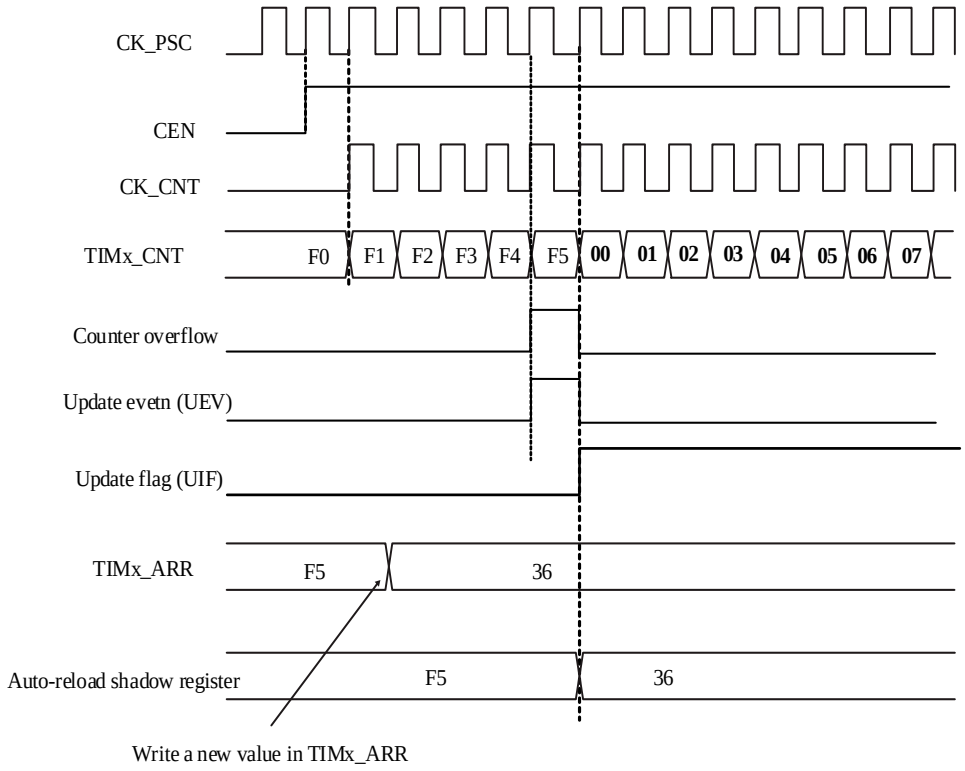


Figure 20-8 Counter timing diagram, update event when ARPE = 1



Downcounting mode

In downcounting mode, the counter counts from the auto-reload value (contents of the TIMx_ARR register) down to 0, then generates a counter underflow event, the UIF flag of the TIMx_SR register will be automatically set to 1, and the counter will be cleared to 0 and starts counting again.

When an update event occurs, all the registers are updated and the update flag (UIF bit in the TIMx_SR register) is set:

- The buffer of the prescaler is reloaded with the preload value (content of the TIMx_PSC register)
- The auto-reload active register is updated with the preload value (content of the TIMx_ARR register). Note that the auto-reload is updated before the counter is reloaded, so that the next period is the expected one.

The following figures show some examples of the counter behavior for different clock frequencies when TIMx_ARR = 0x36:

Figure 20-9 Counting timing diagram, underflow event when the internal clock divided by 1

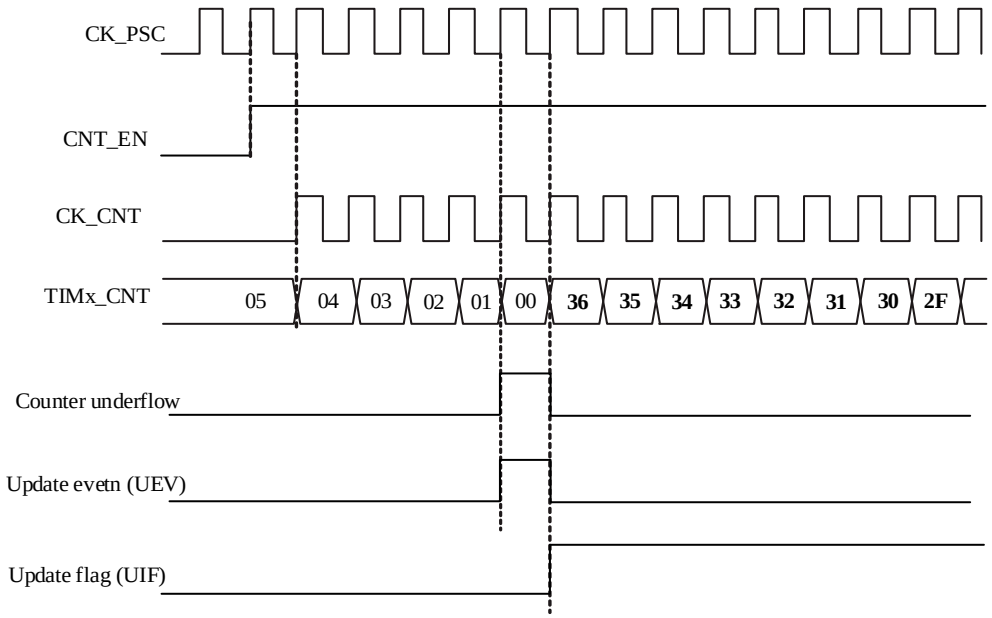


Figure 20-10 Counting timing diagram, underflow event when the internal clock divided by 4

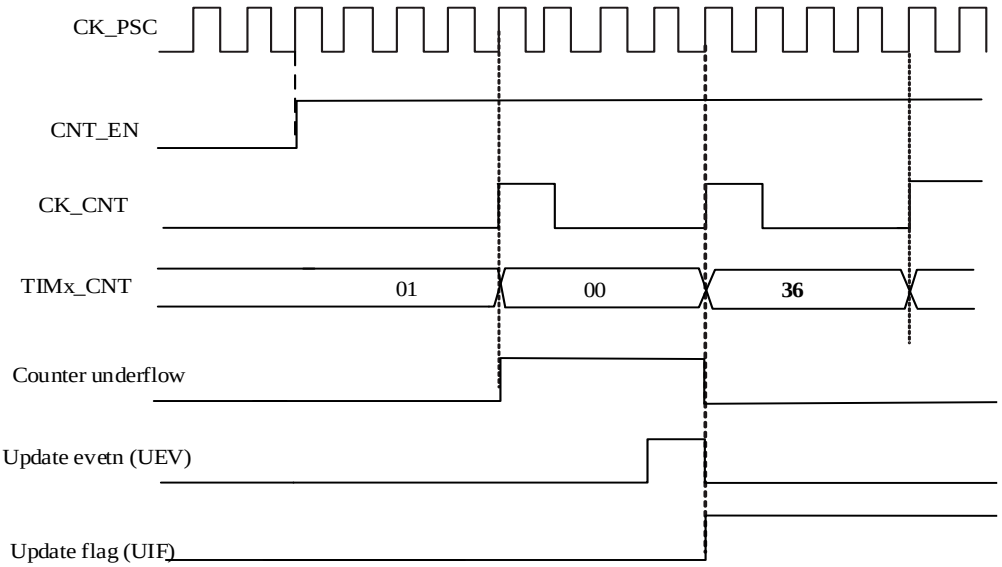
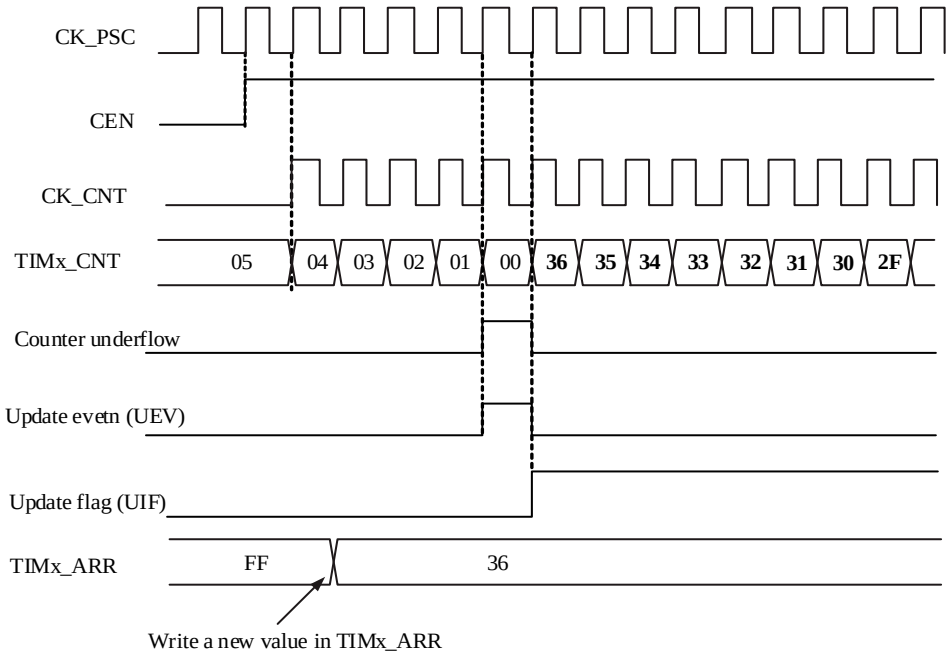


Figure 20-11 Counter timing diagram, update event when ARPE = 0



Center-aligned mode (up/down counting)

In center-aligned mode, the counting process consists of the following stages:

1) The counter counts from 0 to the auto-reload value (content of the TIMx_ARR register) – 1, generates a counter overflow event.

2) The counter counts from the auto- reload value down to 1 and generates a counter underflow event.

3) Then it restarts counting from 0.

When the DIR bit in the TIMx_CR1 register is set to 1, the counter starts decrementing from the current value of the TIMx_CNT register.

Center-aligned mode is active when the CMS bits in the TIMx_CR1 register are not equal to '00'. In this mode, the DIR bit in the TIMx_CR1 register cannot be written. It is updated by hardware and gives the current direction of the counter.

- Center aligned mode 1 (CMS = “01”): The update event can be generated at each counter overflow and at each counter underflow. The Output compare interrupt flag of channels configured in output is set when: the counter counts down.
- Center aligned mode 2 (CMS = “10”): The update event can be generated at each counter overflow and at each counter underflow. The Output compare interrupt flag of channels configured in output is set when: the counter counts up.
- Center aligned mode 3 (CMS = “11”): The update event can be generated at each counter overflow and at each counter underflow. The Output compare interrupt flag of channels configured in output is set when: the counter counts up and down.

When an update event occurs, all the registers are updated and the update flag (UIF bit in the TIMx_SR register) is set (depending on the URS bit):

- The buffer of the prescaler is reloaded with the preload value (content of the TIMx_PSC register)
- The auto-reload active register is updated with the preload value (content of the TIMx_ARR register)

The following figures show some examples of the counter behavior for different clock frequencies when TIMx_ARR = 0x06:

Figure 20-12 Counter timing diagram, internal clock divided by 1

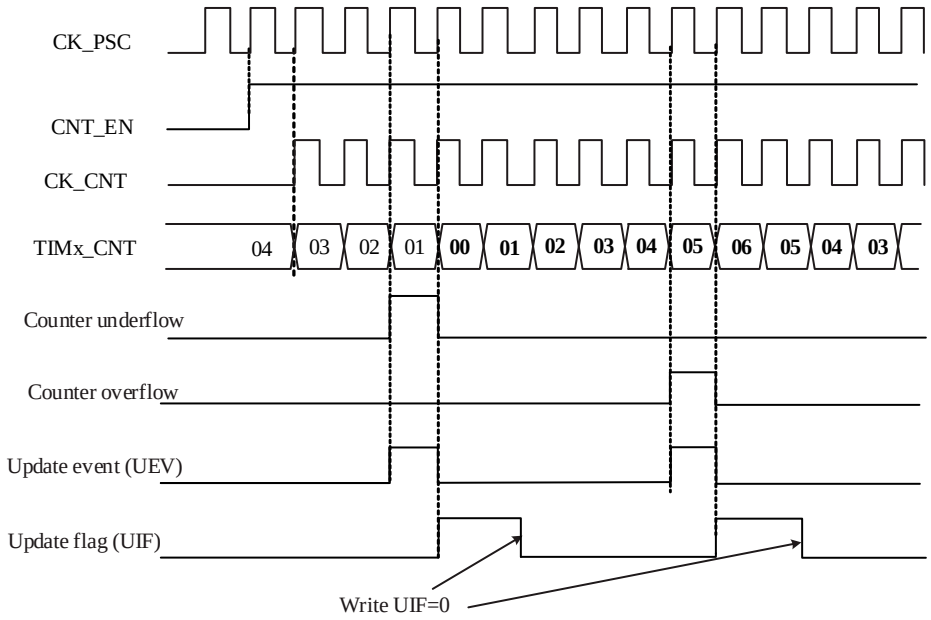


Figure 20-13 Counter timing diagram, internal clock divided by 2

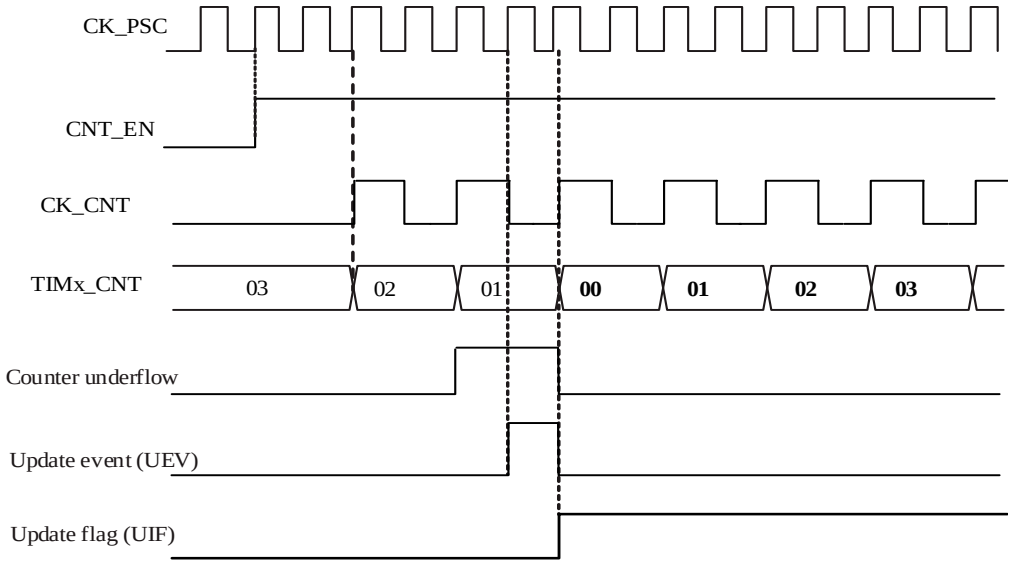


Figure 20-14 Counter timing diagram, update event with ARPE = 1 (counter underflow)

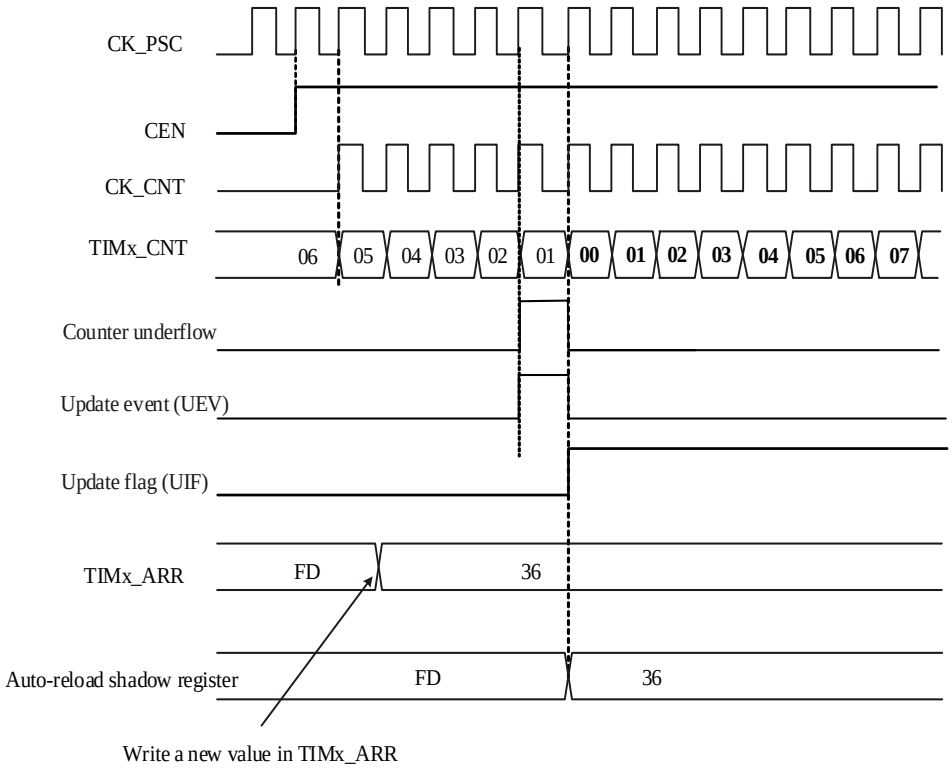
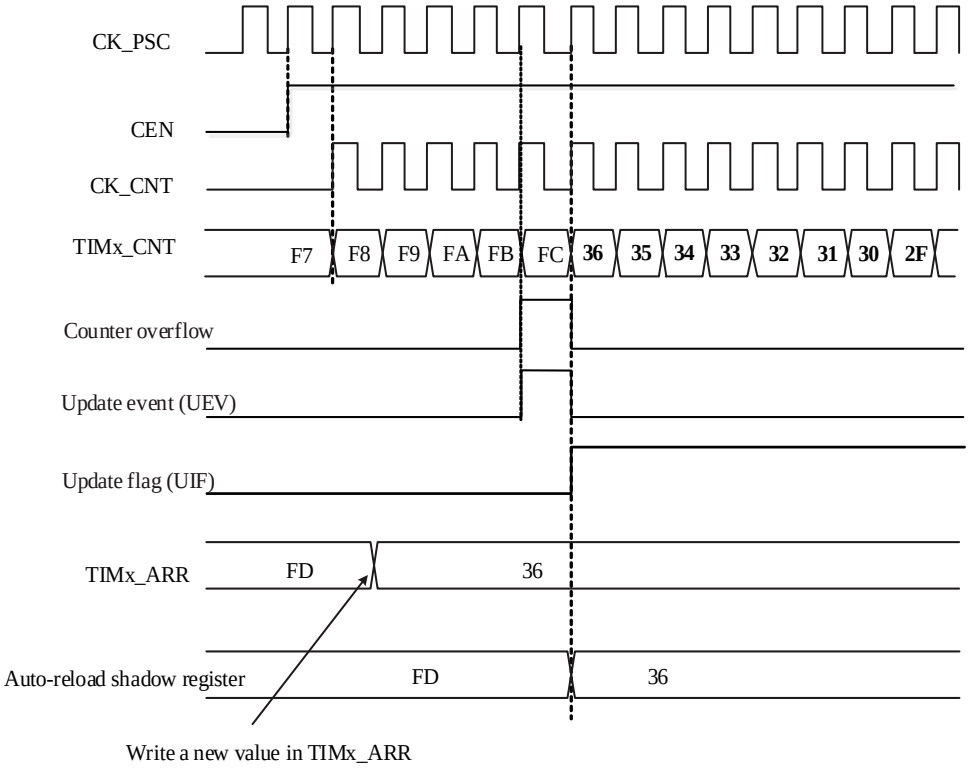


Figure 20-15 Counter timing diagram, update event with ARPE = 1 (counter overflow)



The safest way to use center-aligned mode is to generate an update by software (setting the UG bit in the TIMx_EGR register) just before starting the counter and not to write the counter while it is running.

20.3.5 External trigger input

The timer features an external trigger input ETR. The ETR input comes from multiple sources, the selection is done with the ETR_SEL[2:0] bit field in the TIMx_AF1 register. Refer to the following figure.

Figure 20-16 TIM3 ETR input sources

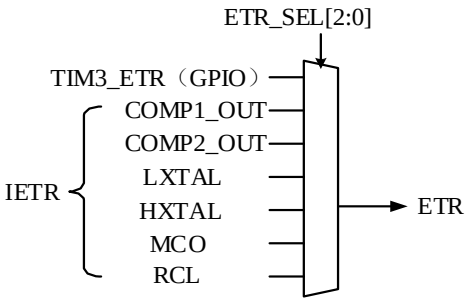
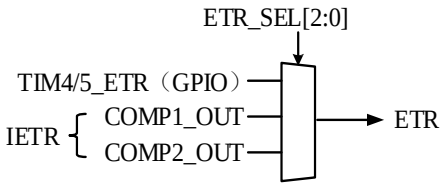


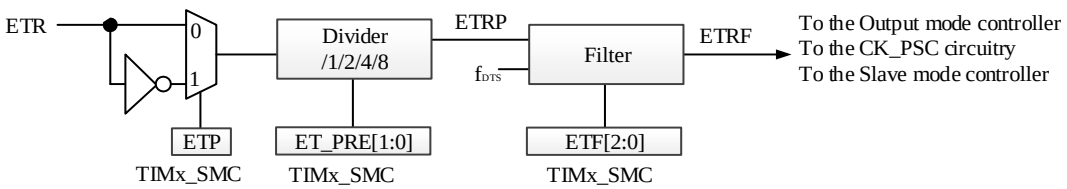
Figure 20-17 TIM4/5 ETR input sources



As shown in the following figure, the ETR input polarity is defined with the ETP bit in the TIMx_SMC register. The trigger can be prescaled with the divider programmed by the ET_PRE[1:0] bit field and digitally filtered with the ETF[2:0] bit field, ETRF is generated. It can be used as:

- Clock mode 2 (see *Clock selection*).
- Trigger for the slave mode (see *Timer and external trigger synchronization*).
- Clearing the OCxREF signal on an external event (see *Clearing the OCxREF signal*).

Figure 20-18 External trigger input block



20.3.6 Clock selection

The counter clock can be provided by the following clock sources:

- Internal clock: TIMx_KCLK
- Clock mode 1: TRIG trigger signal
- Clock mode 2: trigger input ETR
- Encoder mode

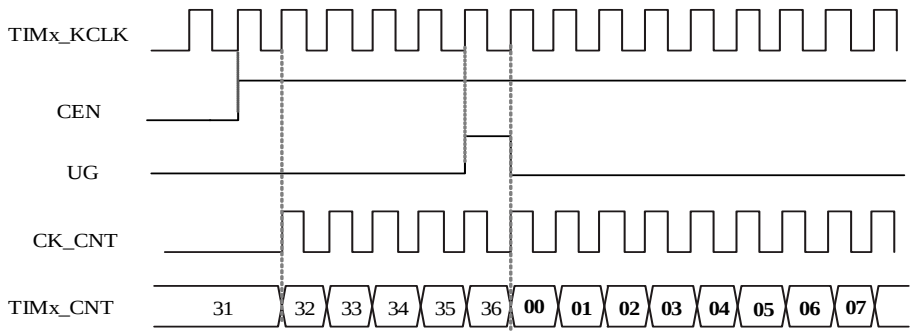
The above clock sources can be prescaled by configuring the value of the TIMx_PSC register and then used as the counter clock (CK_CNT).

Internal clock source

When the slave mode controller is disabled (SM_SEL = 000) in the TIMx_SMC register, the count clock source for the counter is TIMx_KCLK, which is form RCC (TIMx_PCLK).

The following figures shows the behavior of the control circuit and the upcounter in normal mode, without prescaler.

Figure 20-19 Control circuit in internal clock mode



Clock mode 1

This mode is selected when SM_SEL = 111 in the TIMx_SMC register. The counter can count at each rising or falling edge on a selected input (the CCxP and CCxNP bits in the TIMx_CCEN register).

The trigger input source is selected by writing TS[2:0] bit field in the TIMx_SMC register:

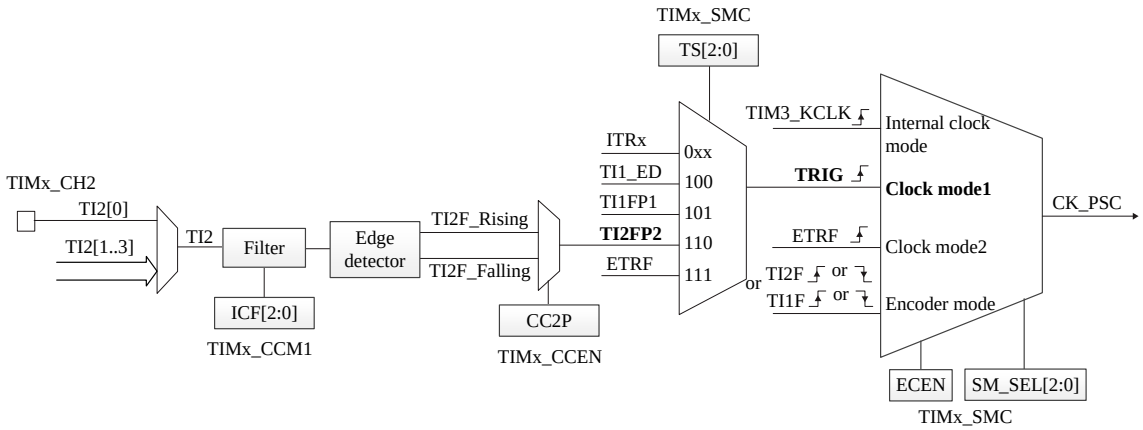
- ITRx (x = 0~3)
- TI1F_ED

- TI1FP1
- TI2FP2
- ETRF

ETRF and TIx signals are asynchronous with TIMx_KCLK, so the maximum frequency of ETRF and TIx is half of the TIMx_KCLK frequency.

When TI1F_ED is selected as the TRIG signal, its maximum frequency is one-quarter of the TIMx_KCLK frequency.

Figure 20-20 TI2 clock mode 1 connection example (rising edge polarity)



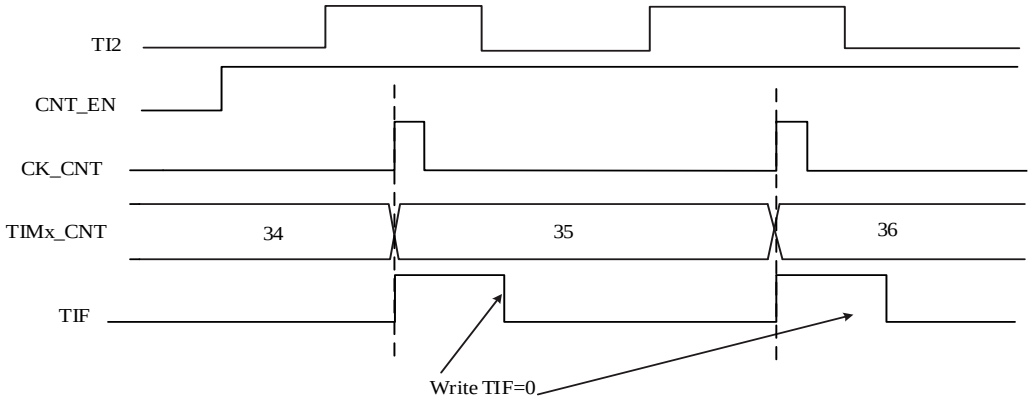
For example, to configure the upcounter to count in response to a rising edge on the TI2 input, use the following procedure:

1) Select the proper TI2x source: Configure the TI2SEL[1:0] = 00 in the TIMx_TISEL register.
2) Set the input filter duration: Configure the IC2F[2:0] bits in the TIMx_CCM1 register (if no filter is needed, keep IC2F[2:0] = 000).
3) Select rising edge polarity: Configure CC2P = 0 and CC2NP = 0 in the TIMx_CCEN register.
4) Configure the timer in clock mode 1: Write SM_SEL[2:0] = 111 in the TIMx_SMC register.
5) Select TI2 as the input source: Configure the TS[2:0] = 110 in the TIMx_SMC register.
6) Enable the counter: Write CEN = 1 in the TIMx_CR1 register.

When a rising edge occurs on TI2, the counter counts once and the TIF flag is set. The delay between the rising edge on TI2 and the actual clock of the counter is due to the

resynchronization circuit on TI2 input.

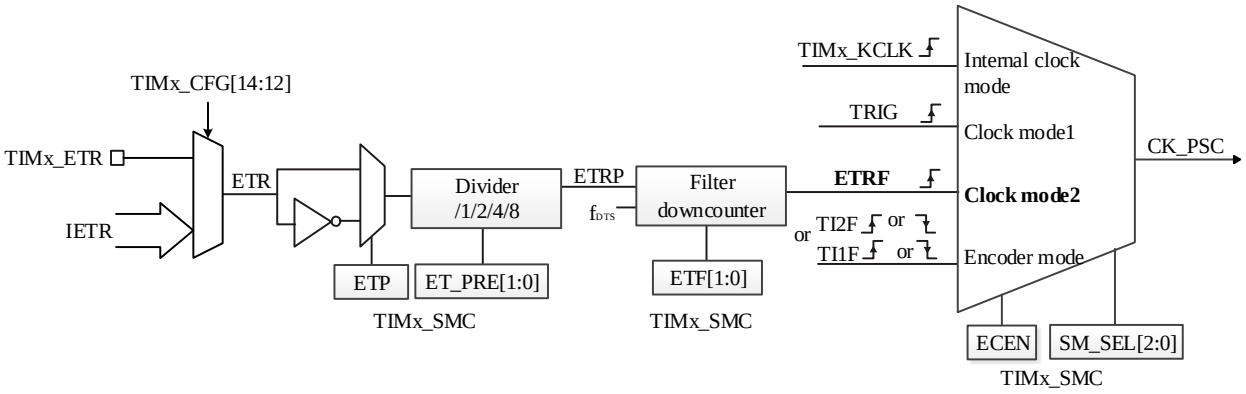
Figure 20-21 Control circuit in clock mode 1



Clock mode 2

This mode is selected by writing ECEN = 1 in the TIMx_SMC register. The counter can count at each rising or falling edge on the external trigger input ETR.

Figure 20-22 External trigger input block



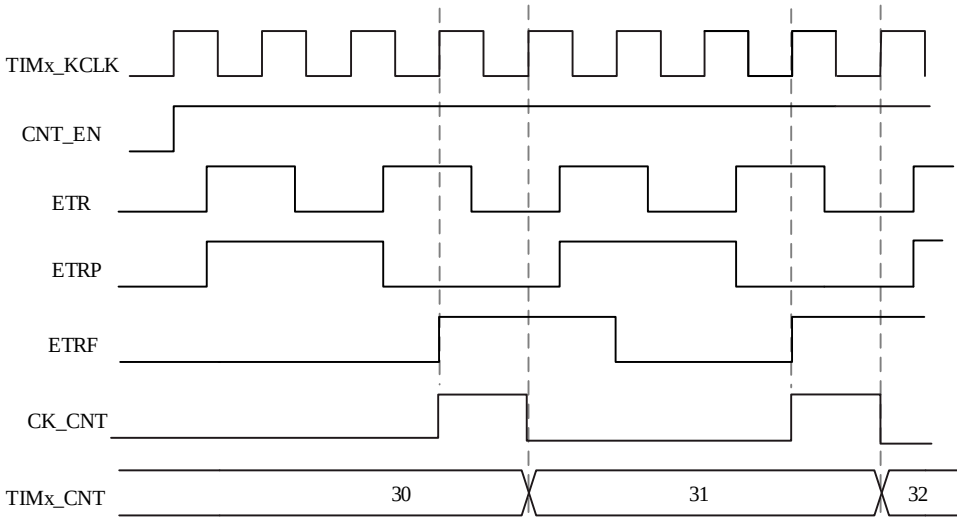
The delay between the rising edge on ETR and the actual clock of the counter is due to the resynchronization circuit on the ETRP signal. As a consequence, the maximum frequency which can be correctly captured by the counter is at most half of the TIMx_KCLK frequency.

For example, to configure the upcounter to count each 2 rising edges on ETR, use the following procedure:

1) Select the proper ETR source: Configure the ETR_SEL[2:0] = 000 in the TIMx_AF1 register.
2) Set the input filter duration: Configure the ETF[2:0] bits in the TIMx_CCM1 register (if no filter is needed, keep ETF[2:0] = 000).

3) Set the prescaler: Configure the ET_PRE[1:0] = 01 in the TIMx_SMC register.
4) Select rising edge polarity: Configure ETP = 0 in the TIMx_SMC register.
5) Enable clock mode 2: Write ECEN = 1 in the TIMx_SMC register.
6) Enable the counter: Write CEN = 1 in the TIMx_CR1 register.

Figure 20-23 Control circuit in clock mode 2



If the signal source of clock mode 1 and clock mode 2 is ETRF, and clock mode 2 takes effect at the same time, clock mode 2 has a higher priority.

Encoder mode

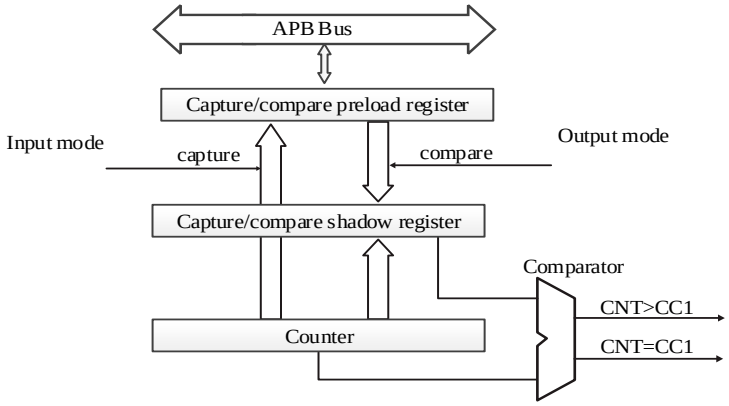
When the SM_SEL[2:0] bits in the TIMx_SMC register are configured to 001, 010, or 011, the counter can count on the rising or falling edges of TI1/TI2 inputs. For more details, see: *Encoder interface mode*.

20.3.7 Capture/Compare channels

Each Capture/Compare channel include:

- A capture/compare register (including a shadow register)
- An input stage for capture (with digital filter, multiplexing and prescaler)
- An output stage (with comparator and output control)

Figure 20-24 Capture/Compare channels



In capture mode, captures are actually done in the shadow register, which is copied into the preload register.

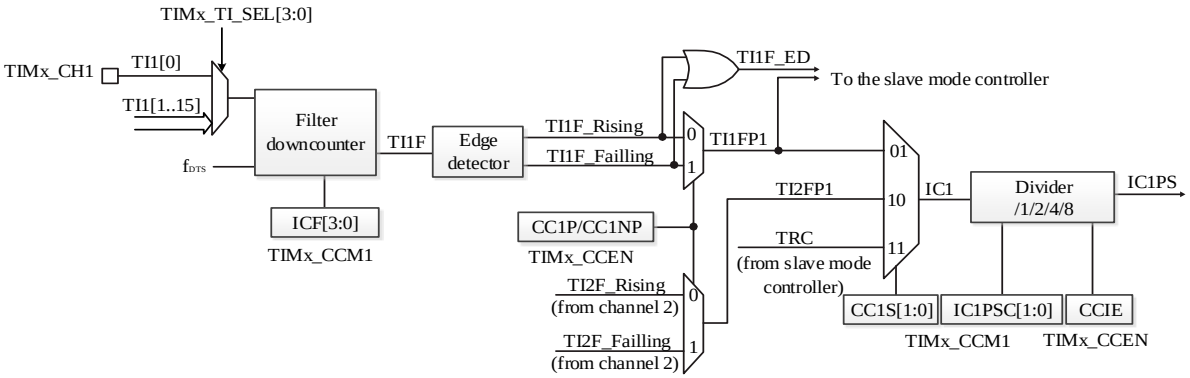
In compare mode, the content of the preload register is copied into the shadow register which is compared to the counter.

Input capture mode

The input sources for TIx can be configured in the TIMx_TISEL register. The input capture source can be configured using the CCxS[1:0] bit fields in the TIMx_CCMx register. For example, for channel 1, the options are TRC, TI1FP1, and TI2FP1. The input stage samples the corresponding TIx input to generate a filtered signal TIxF. Then, an edge detector with polarity selection generates a signal (ICx) which can be used as trigger input by the slave mode controller or as the capture command. It is prescaled before the capture register.

The following figure gives an overview of one Capture/Compare channel.

Figure 20-25 Capture/compare channel (channel 1 input stage)



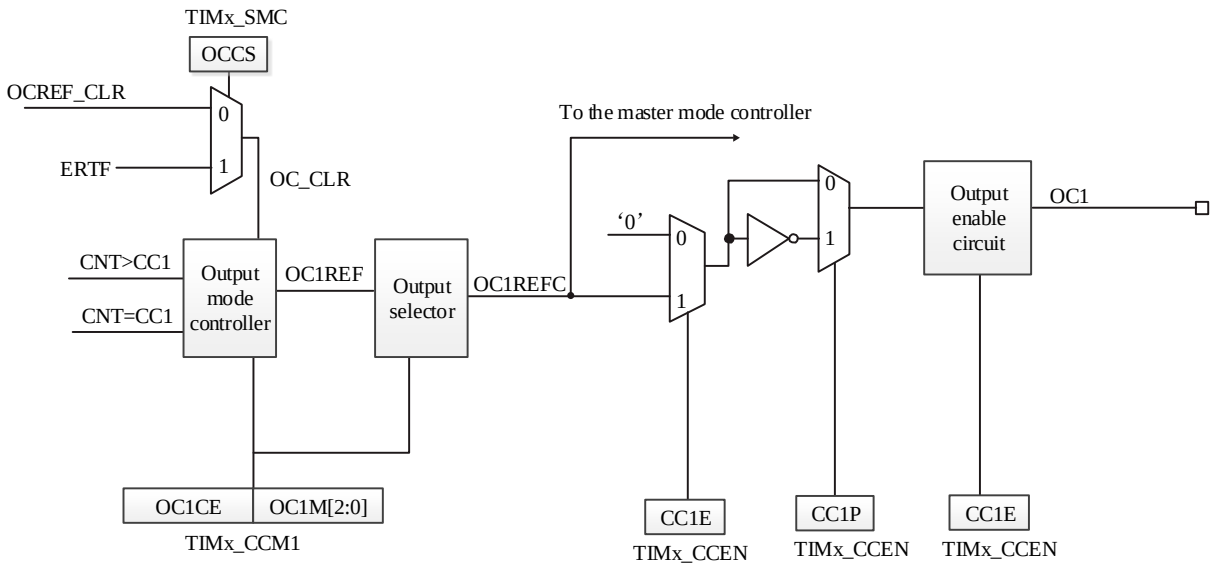
Output compare mode

Output channels 1 to 4 can be output to the external pins of the chip.

The output modes are divided as follows:

- Output compare mode, refer to *Output compare mode*.
- Force output mode, refer to *Forced output mode*.
- PWM mode, refer to *PWM mode*.
- One-pulse mode, refer to *One-pulse mode*.

Figure 20-26 Output stage of capture/compare channel (channel 1)



20.3.8 Input capture mode

In input capture mode, capture events can be generated in the following ways:

- A transition detected by the corresponding ICx signal.
- Setting the corresponding CCxG bit in the TIMx_EVTG register.

When an input capture occurs:

- The TIMx_CCx register gets the value of the counter on the active transition.
- CCxIF flag is set in the TIMx_SR register.
- An interrupt is generated depending on the CCxIE bit.
- A DMA request is generated depending on the CCx_DMAEN bit.

If a capture occurs while the CCxIF flag was already high, then the over-capture flag CCxOF (TIMx_SR register) is set, and the TIMx_CCx register gets the new value.

In input capture mode (CCxS[1:0] $\neq$ 00), The TIMx_CCx register is read-only and

cannot be programmed.

CCxIF can be cleared:

- Write CCxIF = 0.
- Read the captured data stored in the TIMx_CCx register.

The following example shows how to capture the counter value in the TIMx_CC1 when TI1 input rises. To do this, use the following procedure:

1) Select the active input: TIMx_CC1 must be linked to the TI1 input, so write the CC1S bits to 01 in the TIMx_CCM1 register.
2) Set the input filter: Set the IC1F[2:0] bits in the TIMx_CCM1 register to 011.
3) Select rising edge triggering for TI1: Set the CC1P bit in the TIMx_CCEN register to 0.
4) Enable capture: Set the CC1E bit in the TIMx_CCEN register to 1.

20.3.9 PWM input mode

This mode is a particular case of input capture mode. The procedure is the same except:

- Two ICx signals are mapped on the same TIx input.
- Two ICx signals are active on edges with opposite polarity.
- One of the two TIxFP signals is selected as trigger input and the slave mode controller is configured in reset mode. Refer to *Timer and external trigger synchronization*.

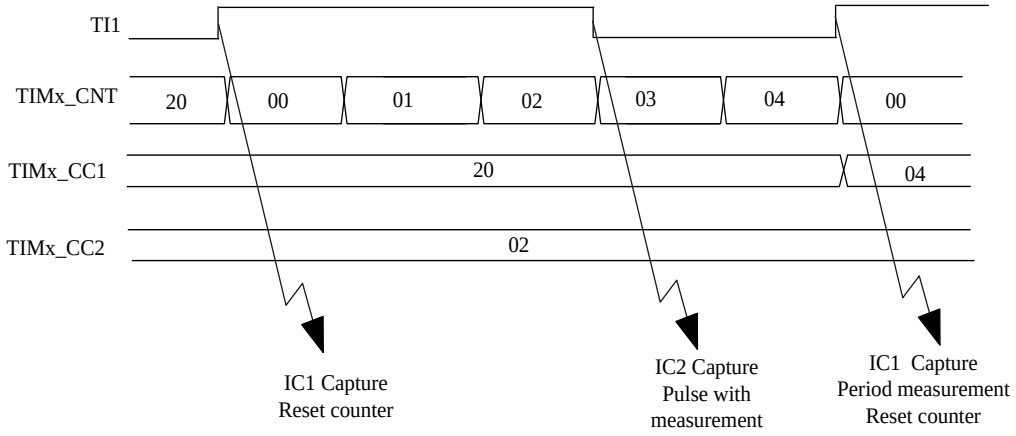
For example, the user can measure the period and the duty cycle of the PWM applied on TI1 using the following procedure:

1) Select the active polarity for TI1FP1: Configure the CC1NP and CC1P bits in the TIMx_CCEN register to 00 (rising edge active).
2) Map TI1FP1 to IC1 by setting the CC1S[1:0] bits in the TIMx_CCM1 register to 01.
3) Select the active polarity for TI1FP2: Set the CC2NP and CC2P bits in the TIMx_CCEN register to 10 (falling edge active).
4) Map TI1FP2 to IC2: Set the CC2S[1:0] bits in the TIMx_CCM1 register to 10.
5) Select the trigger input for the slave mode controller: Set the TS[2:0] bits in the TIMx_SMC register to 101 (select TI1FP1).
6) Configure the slave mode controller in reset mode: Write the SM_SEL[2:0] =

100 in the TIMx_SMC register.

7) Enable the capture: Write the CC1E and CC2E bits to ‘1’ in the TIMx_CCEN register.

Figure 20-27 PWM input mode timing



20.3.10 Output compare mode

This mode includes: match output active level, match output inactive level, and toggle mode. The preload function of the TIMx_CCx register can be enabled or disabled using the OCxPE bit in the TIMx_CCMx register. When a match is found between the capture/compare register and the counter, the output compare function:

- Sets a flag in the interrupt status register (CCxIF bit in the TIMx_SR register).
- The output polarity is determined by the CCxP bit in the TIMx_CCEN register.
- Generates an interrupt if the corresponding interrupt mask is set (CCxIE bit in the TIMx_DIER register).
- Sends a DMA request if the CC_DMASEL bit in the TIMx_CR2 register is 0 and the CCx_DMAEN bit in the TIMx_DIER register is 1.

1) Write the desired data in the TIMx_ARR and TIMx_CCx registers.

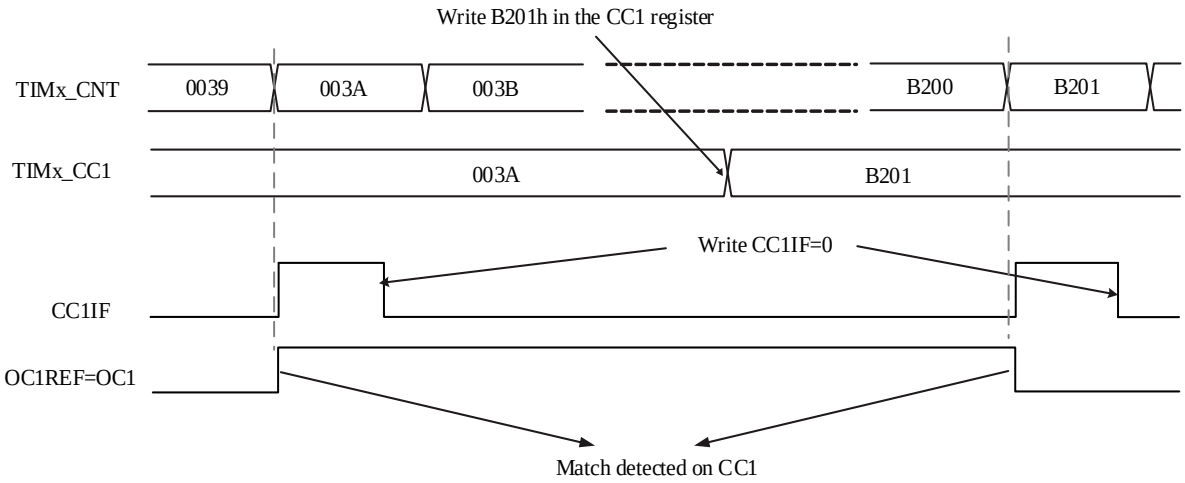
2) Select the output mode, for example:

   - Write OCxM[2:0] = 011 to toggle OCx output pin when CNT matches CCx.
   - Disable preload: Write OCxPE in the TIMx_CCMx register.
   - Select active high polarity: Write CCxP = 0 in the TIMx_CCEN register.
   - Enable the output: Write CCxE = 1 in the TIMx_CCEN register.

3) Enable the counter: Write CEN = 1 in the TIMx_CR1 register.

When the preload register is not enabled (OCxPE bit in the TIMx_CCMx register is 0), the value of the TIMx_CCx register can be changed in real-time to control the output waveform.

Figure 20-28 Compare output mode, toggle on OC1 (OCxPE = 0)



20.3.11 Forced output mode

In output mode, each output compare signal can be forced to active or inactive level, independently of any comparison between the output compare register and the counter.

- To force an output compare signal (OCxREF) to its active level, user just needs to write '101' in the OCxM bits in the corresponding TIMx_CCMx register. Thus OCxREF is forced high (OCxREF is always active high).
- To force an output compare signal (OCxREF) to its inactive level, user just needs to write '100' in the OCxM bits in the corresponding TIMx_CCMx register. Thus OCxREF is forced low.

OCx polarity is software programmable using the CCxP bit in the TIMx_CCEN register.

Anyway, the comparison between the TIMx_CCx shadow register and the counter is still performed and allows the flag to be set.

20.3.12 PWM mode

Pulse width modulation mode allows to generate a signal with a frequency determined by the value of the TIMx_ARR register and a duty cycle determined by the value of the TIMx_CCx register.

The PWM mode can be selected independently on each channel:

- Write OCxM[2:0] bits in the TIMx_CCMx register.
  - PWM mode 1 (OCxM[2:0] = 110): In upcounting, channel 1 is active as long as TIMx_CNT < TIMx_CC1 else inactive. In downcounting, channel 1 is inactive as long as TIMx_CNT > TIMx_CC1 else active.
  - PWM mode 2 (OCxM[2:0] = 111): In upcounting, channel 1 is inactive as long as TIMx_CNT < TIMx_CC1 else active. In downcounting, channel 1 is active as long as TIMx_CNT > TIMx_CC1 else inactive.
- Enabled the corresponding preload by writing OCxPE = 1 in the TIMx_CCMx register.
- Enabled the auto-reload preload register (in upcounting or center-aligned modes) by setting the ARPE bit in the TIMx_CR1 register.

As the preload registers are transferred to the shadow registers only when an update event occurs, before starting the counter, all registers must be initialized by setting the UG bit in the TIMx_EVTG register.

OCx polarity is software programmable using the CCxP bit in the TIMx_CCEN register.

The timer is able to generate PWM in edge-aligned mode or center-aligned mode depending on the CMS[1:0] bits in the TIMx_CR1 register.

PWM edge-aligned mode

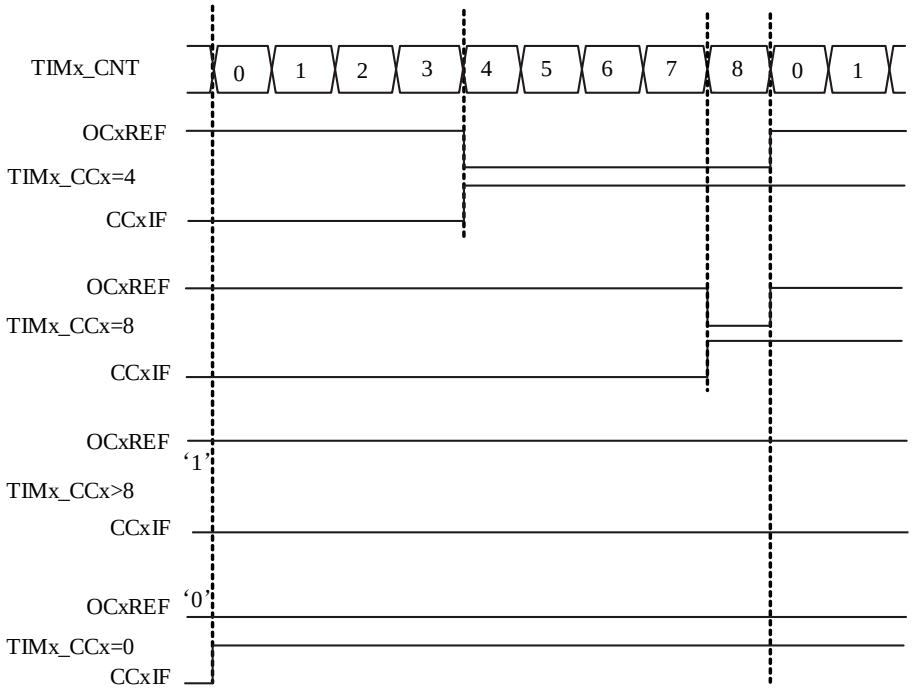
- Upcounting configuration

  Upcounting is active when the DIR bit in the TIMx_CR1 register is low. Refer to the *Upcounting mode*.

  In the following example, we consider PWM mode 1. The reference PWM signal OCxREF is high as long as TIMx_CNT < TIMx_CCx else it becomes low. If the compare value in TIMx_CCx is greater than the auto-reload value (in the TIMx_ARR register) then OCxREF is held at '1'. If the compare value is 0 then OCxREF is held at '0'.

  The following figures shows some edge-aligned PWM waveforms in an example where TIMx_ARR = 8.

Figure 20-29 Edge-aligned PWM waveforms in upcounting mode (ARR = 8)



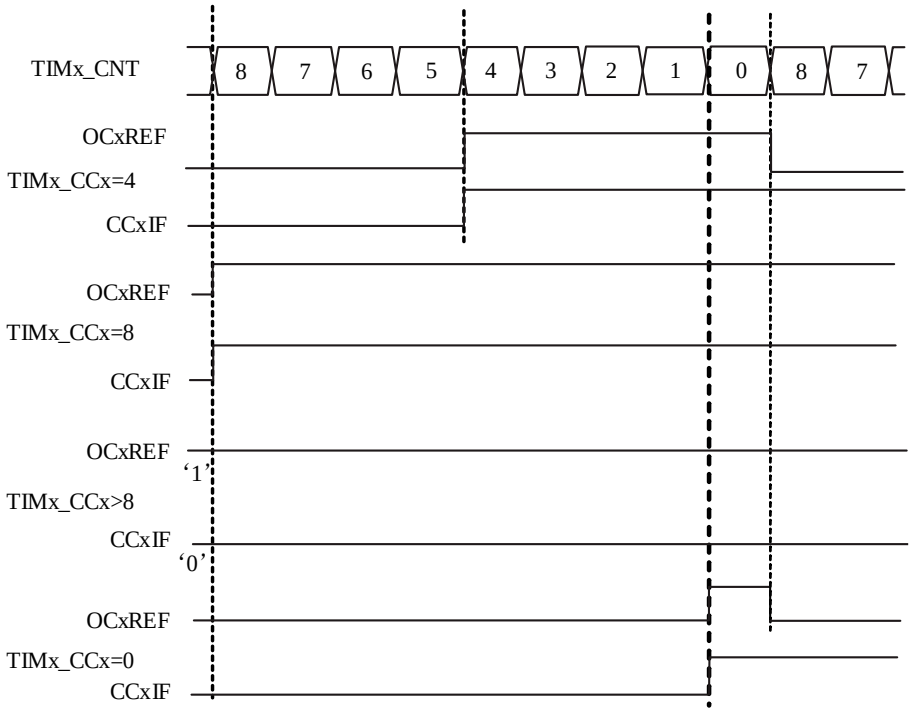
- Downcounting configuration

  Downcounting is active when DIR bit in the TIMx_CR1 register is high. Refer to the *Downcounting mode.*

  In PWM mode 1, the reference signal OCxREF is low as long as TIMx_CNT > TIMx_CCx else it becomes high. If the compare value in TIMx_CCx is greater than the auto-reload value in TIMx_ARR register, then OCxREF is held at ‘1’. 0% PWM is not possible in this mode.

  The following figures shows some edge-aligned PWM waveforms in an example where TIMx_ARR = 8.

Figure 20-30 Edge-aligned PWM waveforms in downcounting mode (ARR = 8)



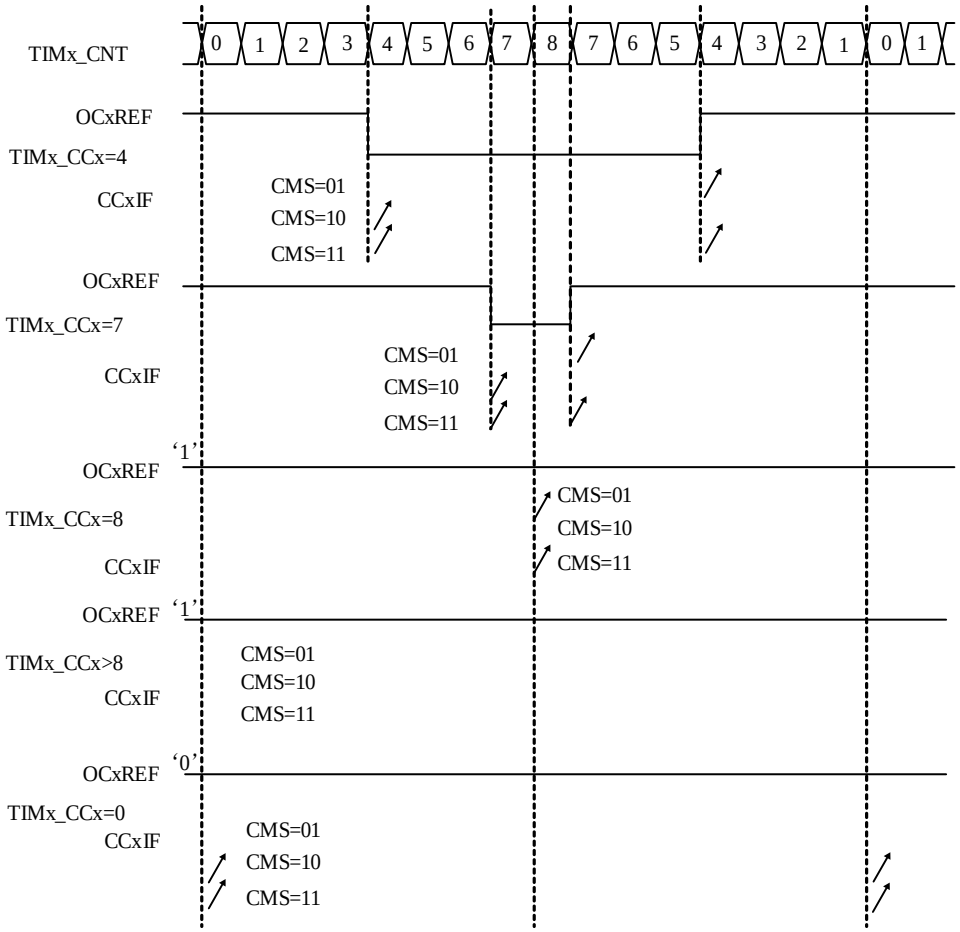
PWM center-aligned mode

Center-aligned mode is active when the CMS[1:0] bits in the TIMx_CR1 register are different from ‘00’.

If the CMS[1:0] bits are configured for center-aligned mode 3, the counter will generate both an overflow and an underflow in each PWM period, allowing the PWM duty cycle to be updated twice. For more details, see: *Center-aligned mode (increment/decrement counting)*.

For example: When the PWM mode is PWM mode 1 and the TIMx_ARR register is set to 8, the PWM waveform in center-aligned mode is as follows:

Figure 20-31 Center-aligned PWM waveforms (ARR = 8)



20.3.13 Clearing the OCxREF signal

The OCxREF signal of a given channel can be cleared when a high level is applied on the OC_CLR signal, OCxREF remains low until the next update event (UEV) occurs.

OC_CLR signal source can be selected by configuring the OCCS bit in the TIMx_SMC register:

- OCCS = 0: COMP1/COMP2 output.
  - Write the OCREF_CLR = 0: OCREF_CLR input can be selected to COMP1 output.
  - Write the OCREF_CLR = 1: OCREF_CLR input can be selected to COMP2 output.
- OCCS = 1: ETRF signal.

  Write the OCxCE = 1 in the TIMx_CCMx register, when a high level is detected on the ETRF input, OCxREF is immediately cleared to 0.

This function can only be used in output compare and PWM modes. It does not work

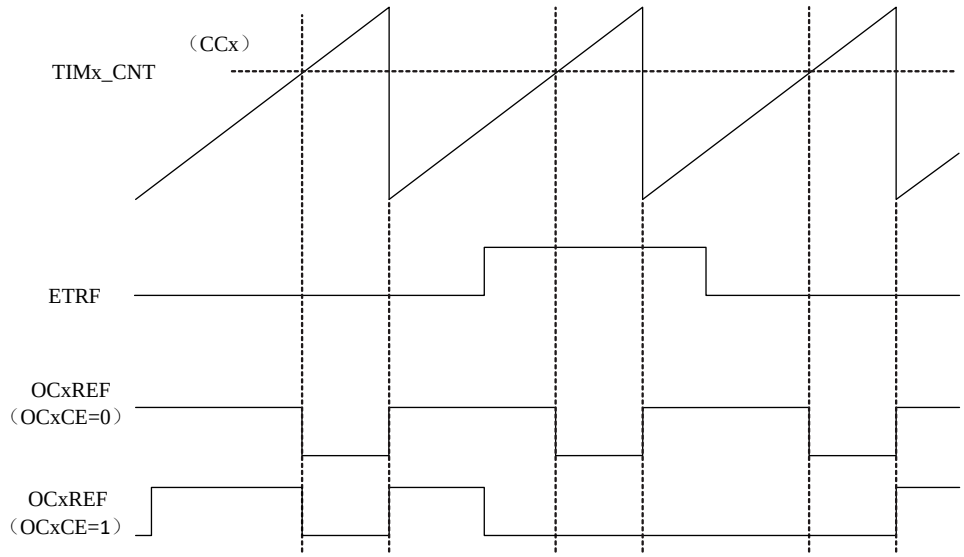
in forced output mode. For example, the OCxREF signal can be connected to the output of a comparator to control the current.

When ETRF is chosen, ETR must be configured as follows:

1) The external trigger prescaler should be kept off: bits ET_PRE[1:0] of the TIMx_SMC register set to ‘00’.
2) The external clock mode 2 must be disabled: bit ECEN of the TIMx_SMC register set to ‘0’.
3) The external trigger polarity and the external trigger filter can be configured according to the user needs.

The following figures shows the behavior of the OCxREF signal when the ETRF input becomes high, for both values of the enable bit OCxCE. In this example, the timer TIMx is programmed in PWM mode.

Figure 20-32 Clearing TIMx OCxREF



20.3.14 One-pulse mode

One-pulse mode (OPM) is a particular case of the previous modes. One-pulse mode is selected by setting the OPM bit in the TIMx_CR1 register. It allows the counter to be started in response to a stimulus and to generate a pulse with a programmable length after a programmable delay.

Starting the counter can be controlled through the slave mode controller. Generating the waveform can be done in output compare mode or PWM mode. This makes the counter stop automatically at the next update event.

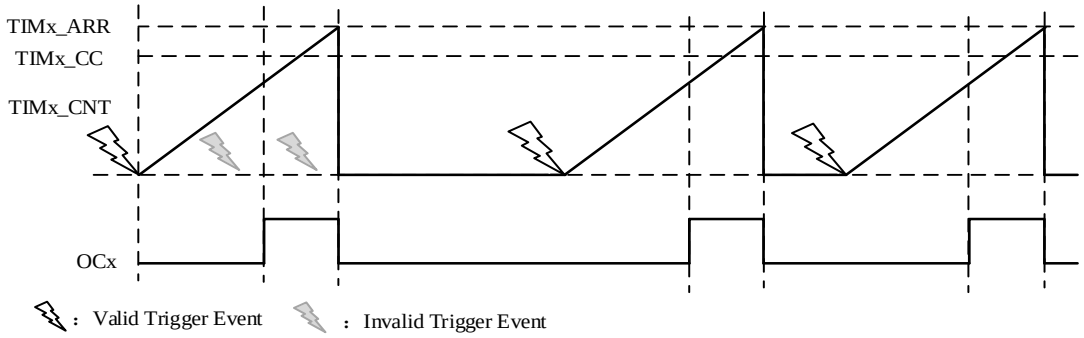
A pulse can be correctly generated only if the compare value is different from the

counter initial value. Before starting (when the timer is waiting for the trigger), the configuration must be:

- In upcounting: CNT < CCx ≤ ARR (in particular, 0 < CCx)
- In downcounting: CNT > CCx

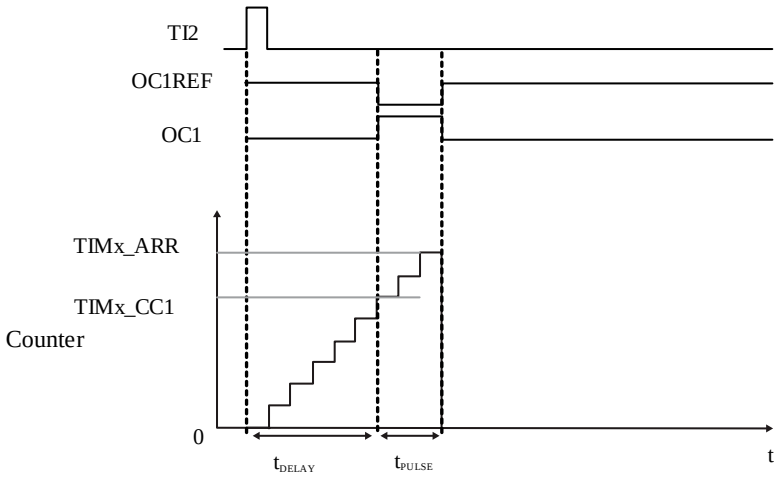
From the start of the counter until the value in the TIMx_CNT register equals the value in the TIMx_ARR register, any trigger events that occur will be discarded. See the following diagram:

Figure 20-33 One pulse mode output



For example, one may want to generate a positive pulse on OC1 with a length of t_{DELAY} and after a delay of t_{PULSE} as soon as a positive edge is detected on the TI2 input pin.

Figure 20-34 Example of one pulse mode



Let's use TI2FP2 as trigger 1:

1) Select the proper TI2x source with the TI2_SEL[2:0] bits in the TIMx_TISEL register.
2) Map TI2FP2 on TI2 by writing CC2S = 01 in the TIMx_CCM1 register.
3) TI2FP2 must detect a rising edge, write CC2P = 0 and CC2NP = 0 in the

TIMx_CCEN register.

4) Configure TI2FP2 as trigger for the slave mode controller (TRIG) by writing TS[2:0] = 110 in the TIMx_SMC register.

5) TI2FP2 is used to start the counter by writing SM_SEL[2:0] = 110 in the TIMx_SMC register (trigger mode).

The OPM waveform is defined by writing the compare registers (taking into account the clock frequency and the counter prescaler).

- t_{DELAY} is defined by the value written in the TIMx_CC1 register.
- t_{PULSE} is defined by the difference between the auto-reload value and the compare value (TIMx_ARR - TIMx_CC1).
- To build a waveform with a transition from 0 to 1 when a compare match occurs and a transition from 1 to 0 when the counter reaches the auto-reload value.
  - OC1 output polarity: CC1P bit is 0.
  - PWM mode 2 enable (writing OC1M[2:0] = 111 in the TIMx_CCM1 register)
  - Optionally the preload registers can be enabled by writing OC1PE = 1 in the TIMx_CCM1 register and ARPE in the TIMx_CR1 register.
  - Write the compare value in the TIMx_CC1 register
  - Write the auto-reload value in the TIMx_ARR register.
  - Generate an update by setting the UG bit and wait for external trigger event on TI2.

In our example, the DIR and CMS bits in the TIMx_CR1 register should be low.

OCx fast enable

In One-pulse mode, the edge detection on TIx input set the CEN bit which enables the counter. Then the comparison between the counter and the compare value makes the output toggle. But several clock cycles are needed for these operations, limiting the minimum delay t_{DELAY} min.

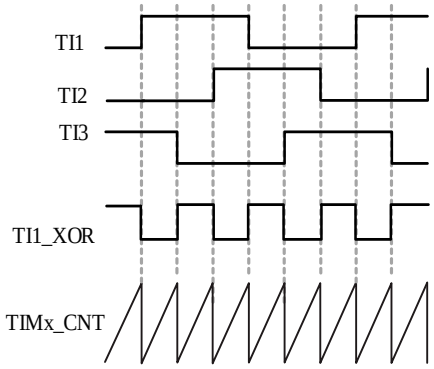
To output a waveform with the minimum delay, the OCxFE bit can be set in the TIMx_CCMx register. Then OCxREF (and OCx) is forced in response to the stimulus, without taking in account the comparison. Its new level is the same as if a compare match had occurred. OCxFE acts only if the channel is configured in PWM1 or PWM2 mode.

20.3.15 Timer input XOR function

The TI1_XOR_SEL bit in the TIMx_CR2 register, allows the input filter of channel 1 to be connected to the output of a XOR gate, combining the three input pins TI1, TI2, and TI3.

The XOR output can be used with all the timer input functions such as trigger or input capture. This makes it easier to measure the interval between edges on two input signals (see the following diagram).

Figure 20-35 Measure the time interval between edges of three signals



20.3.16 Encoder interface mode

Encoder interface mode has three counting methods:

- Encoder mode 1: When the SM_SEL[2:0] bits in the TIMx_SMC register are set to 001, the counter counts only on TI1 edges.
- Encoder mode 2: When the SM_SEL[2:0] bits in the TIMx_SMC register are set to 010, the counter counts only on TI2 edges.
- Encoder mode 3: When the SM_SEL[2:0] bits in the TIMx_SMC register are set to 011, the counter counts on both TI1 and TI2 edges.

Encoder interface mode acts simply as an external clock with direction selection. This means that the counter just counts continuously between 0 and the auto-reload value in the TIMx_ARR register (0 to ARR or ARR down to 0 depending on the direction). So the TIMx_ARR must be configured before starting. In the same way, the capture, compare, prescaler, trigger output features continue to work as normal.

The two inputs TI1 and TI2 are used to interface to an incremental encoder. Select the TI1 and TI2 polarity by programming the CC1P and CC2P bits in the TIMx_CCEN register. When needed, the input filter can be programmed as well (Configure using the IC1F[2:0] and IC2F[2:0] bit fields in the TIMx_CCM1 register). The counter is

clocked by each valid transition on TI1FP1 or TI2FP2 assuming that it is enabled (CEN bit in TIMx_CR1 register written to 1). The table summarizes the possible combinations, assuming TI1 and TI2 do not switch at the same time.

When both encoder interface mode and clock mode 2 are configured, encoder interface mode has higher priority.

Note: When enabling Encoder interface mode, the PSC prescaler must be set to 0.

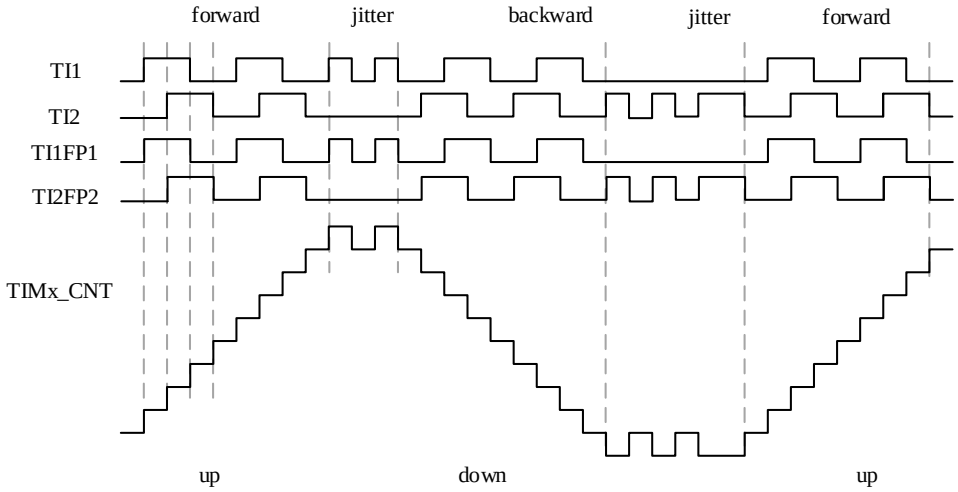
Table 20-2 Counting direction versus encoder signals

Active edge	Level on opposite signal (TI1FP1 for TI2, TI2FP2 for TI1)	TI1FP1 signal		TI2FP2 signal	
		Rising	Falling	Rising	Falling
Counting on TI1 only	High	Down	Up	No Count	No Count
	Low	Up	Down	No Count	No Count
Counting on TI2 only	High	No Count	No Count	Up	Down
	Low	No Count	No Count	Down	Up
Counting on TI1 and TI2	High	Down	Up	Up	Down
	Low	Up	Down	Down	Up

The following diagram gives an example of counter operation, showing count signal generation and direction control. It also shows how input jitter is compensated where both edges are selected. This might occur if the sensor is positioned near to one of the switching points. For this example, the configuration is the following:

1) Map TI1 to TI1FP1: Write 01 to the CC1S[1:0] bits in the TIMx_CCM1 register.
2) Map TI2 to TI2FP2: Write 01 to the CC2S[1:0] bits in the TIMx_CCM1 register.
3) TI1FP1 is not inverted, so TI1FP1 equals TI1: Clear the CC1P and CC1NP bits in the TIMx_CCEN register.
4) TI2FP2 is not inverted, so TI2FP2 equals TI2: Clear the CC2P and CC2NP bits in the TIMx_CCEN register.
5) Both rising and falling edges are active: Write 011 to the SM_SEL[2:0] bits in the TIMx_SMC register.
6) Enable the counter: Set the CEN bit in the TIMx_CR1 register to 1.

Figure 20-36 Example of counter operation in encoder interface mode



20.3.17 Timers and external trigger synchronization

The TIMx timers are linked together internally for timer synchronization or chaining.

Table 20-3 TIMx internal trigger connection

TIMx	ITR0 (TS = 000)	ITR1 (TS = 001)	ITR2 (TS = 010)	ITR3 (TS = 011)
TIM3	TIM5	Reserved	TIM4	Reserved
TIM4	Reserved	TIM3	TIM5	Reserved
TIM5	Reserved	TIM3	TIM4	Reserved

Reset mode

The counter and its prescaler can be reinitialized in response to an event on a trigger input.

If the URS bit from the TIMx_CR1 register is low, an update event UEV is generated. Then all the preloaded registers (TIMx_ARR, TIMx_CCx) are updated.

In the following example, the upcounter is cleared in response to a rising edge on TI1 input:

1) Map TI1 to IC1: Write 01 to the CC1S[1:0] bits in the TIMx_CCM1 register.
2) Configure TI1 to trigger on rising edges: Set the CC1P and CC1NP bits in the TIMx_CCEN register to 0.
3) Configure the input filter bandwidth (no filter is needed in this example, so keep the IC1F[2:0] bits at 000).
4) Configure reset mode: Write 100 to the SM_SEL[2:0] bits in the TIMx_SMC register.

5) Select TI1FP1 as the trigger source: Write 101 to the TS[2:0] bits in the TIMx_SMC register.

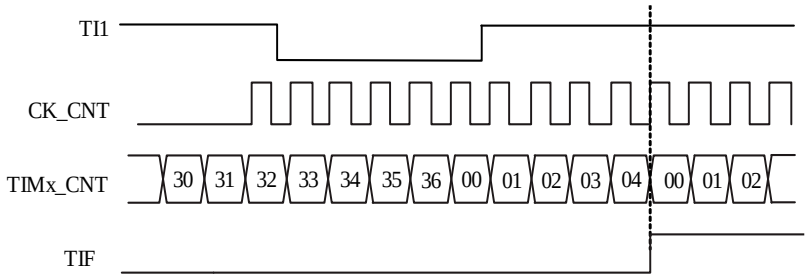
6) Enable the counter: Set the CEN bit in the TIMx_CR1 register to 1.

The counter starts counting on the internal clock, then behaves normally until TI1 rising edge. When TI1 rises, the counter is cleared and restarts from 0. In the meantime, the trigger flag is set (TIF bit in the TIMx_SR register) and an interrupt request, or a DMA request can be sent if enabled (depending on the TIE and TDMA_EN bits in TIMx_DIER register).

The delay between the rising edge on TI1 and the actual reset of the counter is due to the resynchronization circuit on TI1 input.

The following figure shows this behavior when the auto-reload register TIMx_ARR = 0x36.

Figure 20-37 Control circuit in reset mode



Gated mode

The counter can be enabled depending on the level of a selected input.

In the following example, the upcounter counts only when TI1 input is low:

1) Map TI1 to IC1: Write 01 to the CC1S[1:0] bits in the TIMx_CCM1 register.

2) Configure TI1 to trigger on low levels: Set the CC1P bit to 1 and the CC1NP bit to 0 in the TIMx_CCEN register.

3) Configure the input filter bandwidth (no filter is needed in this example, so keep the IC1F[2:0] bits at 000).

4) Configure the timer for gated mode: Write 101 to the SM_SEL[2:0] bits in the TIMx_SMC register.

5) Select TI1FP1 as the trigger source: Write 101 to the TS[2:0] bits in the TIMx_SMC register.

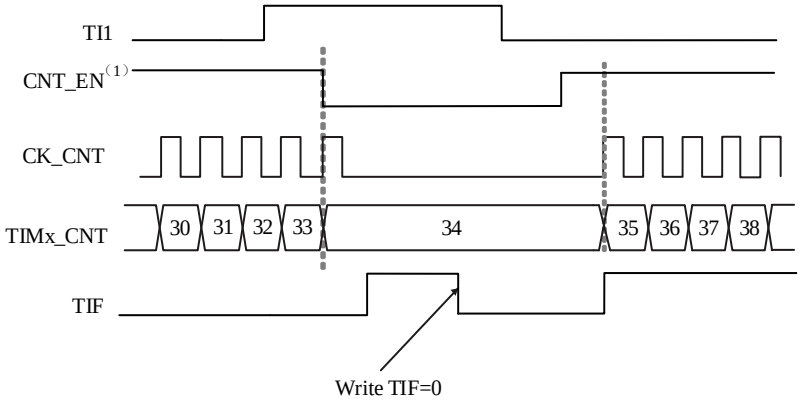
6) Enable the counter: Set the CEN bit in the TIMx_CR1 register to 1.

The counter starts counting on the internal clock as long as TI1 is low and stops as

soon as TI1 becomes high. The TIF flag in the TIMx_SR register is set both when the counter starts or stops.

The delay between the rising edge on TI1 and the actual stop of the counter is due to the resynchronization circuit on TI1 input.

Figure 20-38 Control circuit in gated mode



1. CNT_EN is an internal control signal of TIMx.

Trigger mode

The counter can start in response to an event on a selected input.

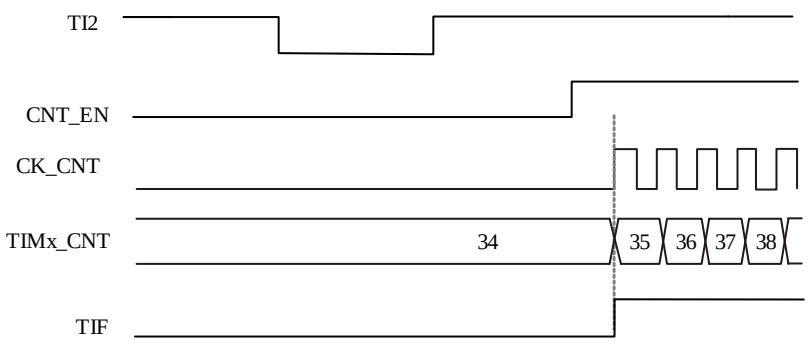
In the following example, the upcounter starts in response to a rising edge on TI2 input:

1) Map TI2 to IC2: Write 01 to the CC2S[1:0] bits in the TIMx_CCM1 register.
2) Configure TI2 to trigger on rising edges: Set the CC2P and CC2NP bits in the TIMx_CCEN register to 0.
3) Configure the input filter bandwidth (no filter is needed in this example, so keep the IC2F[2:0] bits at 000).
4) Configure the timer for trigger mode: Write 110 to the SM_SEL[2:0] bits in the TIMx_SMC register.
5) Select TI2FP2 as the trigger source: Write 110 to the TS[2:0] bits in the TIMx_SMC register.

When a rising edge occurs on TI2, the counter starts counting on the internal clock and the TIF flag is set in the TIMx_SR register.

The delay between the rising edge on TI2 and the actual start of the counter is due to the resynchronization circuit on TI2 input.

Figure 20-39 Control circuit in trigger mode



External clock mode 2 + trigger mode

The external clock mode 2 can be used in addition to another slave mode (except external clock mode 1 and encoder mode). In this case, the ETR signal is used as external clock input, and another input can be selected as trigger input when operating in reset mode, gated mode or trigger mode.

Note: In this mode, the ETR cannot be selected as the TRIG signal using the TS[2:0] bits in the TIMx_SMC register.

In the following example, the upcounter is incremented at each rising edge of the ETR signal as soon as a rising edge of TI1 occurs:

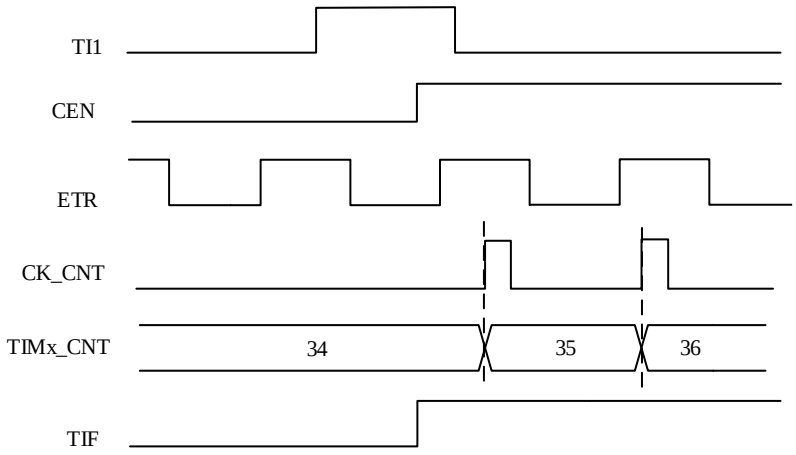
1) Configure ETR (set relevant bits in the TIMx_SMC register):
   - Configure the input filter bandwidth (no filter is needed in this example, so keep the ETF[2:0] bits at 000).
   - Set the prescaler value to 1: Write 00 to the ET_PRE[2:0] bits.
   - Configure ETR to trigger on rising edges: Clear the ETP bit to 0.
   - Enable Clock Mode 2: Set the ECEN bit to 1.
2) Configure TI1 (set relevant bits in the TIMx_CCM1 register):
   - Configure the input filter bandwidth (no filter is needed in this example, so keep the IC1F[2:0] bits at 000).
   - Map TI1 to IC1: Write 01 to the CC1S[1:0] bits.
3) Configure rising edge triggering: Clear the CC1P and CC1NP bits in the TIMx_CCEN register.
4) Configure trigger mode: Write 110 to the SM_SEL[2:0] bits in the TIMx_SMC register.

5) Set TI1FP1 as the trigger source: Write 101 to the TS[2:0] bits in the TIMx_SMC register.

A rising edge on TI1 enables the counter and sets the TIF flag in the TIMx_SR register. The counter then counts on ETR rising edges.

The delay between the rising edge of the ETR signal and the actual reset of the counter is due to the resynchronization circuit on ETRP input.

Figure 20-40 Control circuit in external clock mode 2 + trigger mode



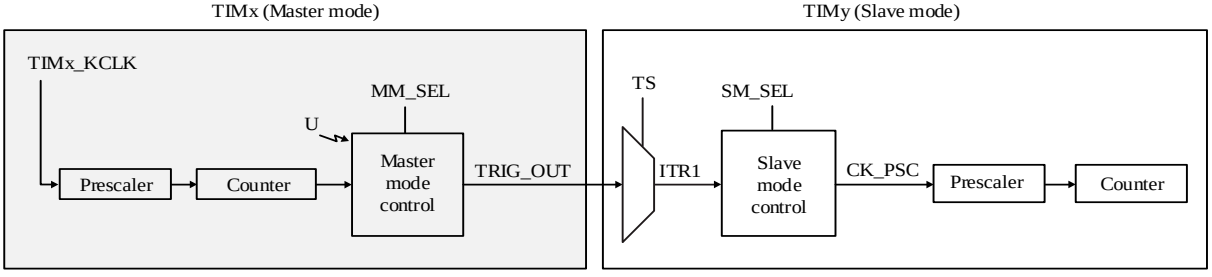
20.3.18 Timer synchronization

The TIMx timers are linked together internally for timer synchronization or chaining.

The following figures shows presents an overview of the trigger selection and the master mode selection blocks.

Note: The slave peripheral clock must be enabled first to receive trigger events from the master mode timer, and the slave peripheral clock must not be changed in real-time to avoid losing trigger signals.

Figure 20-41 Master/Slave timer example



Using TIMx as prescaler for TIMy

For example, TIMx can be configured to act as prescaler for TIMy:

1) Configure TIMx in master mode: Set the MM_SEL[2:0] bits in the TIMx_CR2 register to 010 to output a rising edge on TRIG_OUT each time an update event is generated.
2) Set TIMx TRIG_OUT to TIMy: Configure TIMy in slave mode and set ITRx as the internal trigger source. This can be configured using the TS[2:0] bits in the TIMy_SMC register.
3) Set the slave mode controller to Clock Mode 1: Write 111 to the SM_SEL[2:0] bits in the TIMy_SMC register, so that TIMy clock is provided by the rising edges of the periodic trigger signals from TIMx.
4) Enable the TIMy counter: Set the CEN bit in the TIMy_CR1 register to 1.

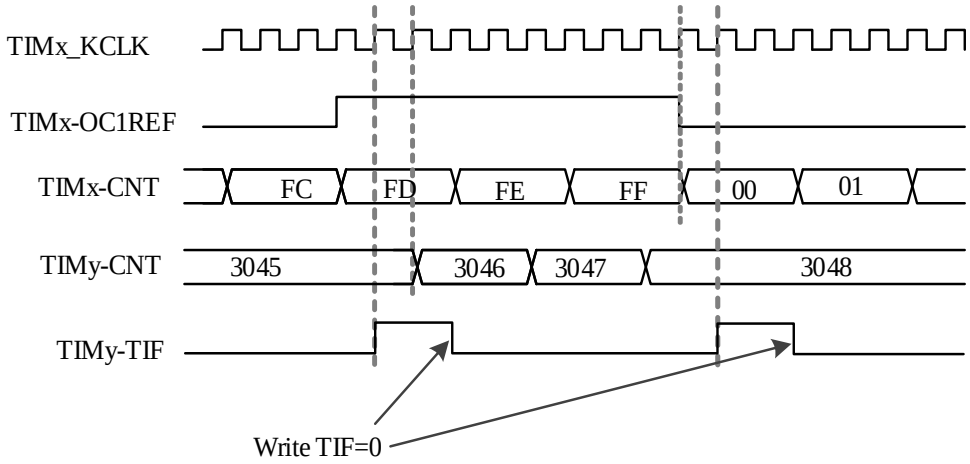
Enable the TIMx counter: Set the CEN bit in the TIMx_CR1 register to 1.

Using TIMx to enable TIMy

In this example, the enable of TIMx is controlled by the output compare 1 of TIMy. Refer to the diagram below. TIMy counts on the divided internal clock only when OC1REF of TIMx is high. Both counter clock frequencies are divided by 3 by the prescaler compared to CK_CNT.

1) Set TIMx to master mode and configure the output compare 1 reference signal (OC1REF) as the trigger output: Write 100 to the MM_SEL[2:0] bits in the TIMx_CR2 register.
2) Configure the TIMx OC1REF waveform: Set the TIMx_CCM1 register.
3) Set TIMy to receive the trigger input from TIMx: Write 0xx (ITR0 to ITR3) to the TS[2:0] bits in the TIMy_SMC register.
4) Set TIMy to gated mode: Write 101 to the SM_SEL[2:0] bits in the TIMy_SMC register.
5) Enable the TIMy counter: Set the CEN bit in the TIMy_CR1 register to 1.
6) Enable the TIMx counter: Set the CEN bit in the TIMx_CR1 register to 1.

Figure 20-42 Gating TIMy with OC1REF of TIMx



In the diagram above, the TIMy counter and prescaler are not initialized before being started. So they start counting from their current value. It is possible to start from a given value by resetting both timers before starting TIMx.

Starting 2 timers synchronously in response to an external trigger

In this example, the enable of TIMx is set when its TI1 input rises, and the enable of TIMy with the enable of TIMx. To ensure the counters are aligned, TIMx must be configured in Master/Slave mode (For TI1, TIMx operates in slave mode; for TIMy, TIMx operates in master mode):

1) Set TIMx trigger output TRIG_OUT: Write 001 to the MM_SEL[2:0] bits in the TIMx_CR2 register.

2) Set TIMx to receive the input trigger TRIG from TI1: Write 100 to the TS[2:0] bits in the TIMx_SMC register.

3) Set TIMx to trigger mode: Write 110 to the SM_SEL[2:0] bits in the TIMx_SMC register.

4) Set TIMx to master mode: Set the MS_MOD bit in the TIMx_SMC register to 1.

5) Set TIMy to receive the input trigger from TIMx: Write 000 to the TS[2:0] bits in the TIMy_SMC register.

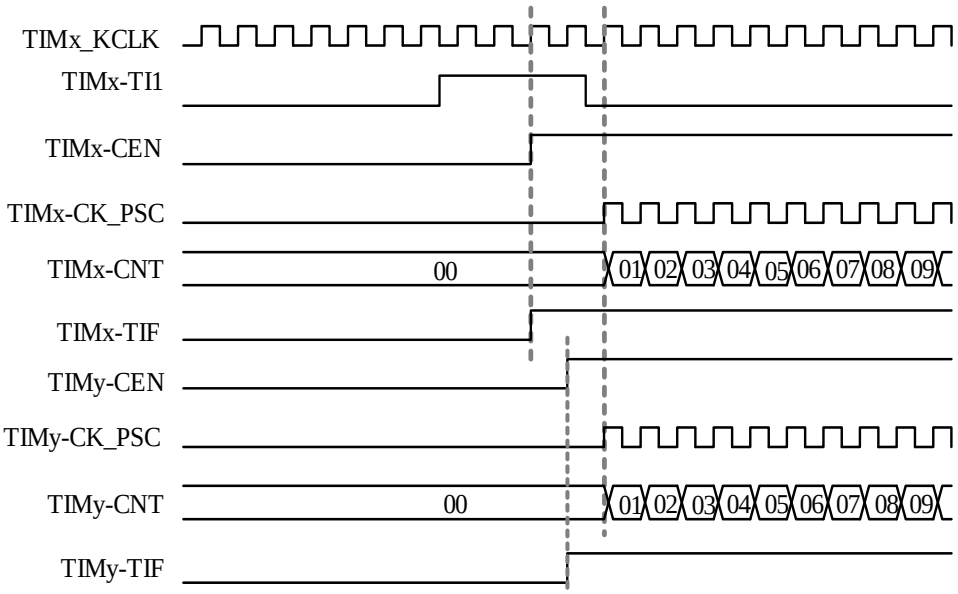
6) Set TIMy to trigger mode: Write 110 to the SM_SEL[2:0] bits in the TIMy_SMC register.

When a rising edge occurs on TI1, both counters starts counting synchronously on the

internal clock and both TIF flags are set.

In this example both timers are initialized before starting (by setting their respective UG bits). Both counters start from 0, but an offset can easily be inserted between them by writing any of the counter registers (TIMx_CNT).

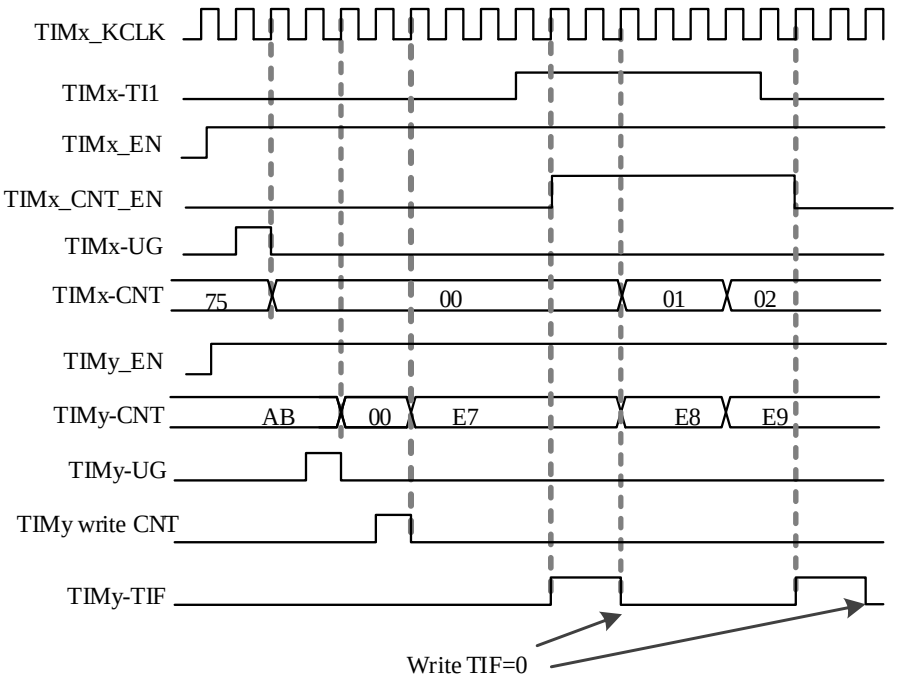
Figure 20-43 Triggering TIMx and TIMy with TIMx TI1 input



If the trigger mode of the TIMx slave mode controller is configured as gated mode, the counters start counting when TI1 is high and stop counting when TI1 is low.

In the following configuration, TIMx is in master mode and starts counting from 0. TIMy is in slave mode and starts counting from 0xE7. Both timers have the same prescaler ratio.

Figure 20-44 Gating TIMy with enable of TIMx



20.3.19 DMA burst mode

When DMA burst transfer is enabled, TIMx can generate DMA requests based on an event to write data to multiple registers of the timer or read values from multiple registers of the timer. For more details on DMA burst functionality, *see DMA Transfer*.

For example, when configuring DMA burst transfer, data can be sequentially updated to the TIMx_CCx registers (x = 1, 2, 3, 4) upon an update event. The specific steps are as follows (assuming the DMA clock is enabled):

1) Configure the DMA channel (refer to the table: *DMA Channel Request Signals*):

- Target address: TIMx_CC1 register.
- Source address: The SRAM buffer address containing the data to be transferred to the TIMx_CCx registers.
- Address increment: Configure the target address to increment as needed.
- Transfer quantity: 4.
- Enable DMA burst transfer: Set the TYPE bit in the DMA_CCx register to 1.
- Data Width: Word (32-bit).

2) Enable the update DMA request for TIMx: Set the UDMA_EN bit in the TIMx_DIER register to 1.

3) Enable TIMx: Set the CEN bit in the TIMx_CR1 register to 1.

4) Enable the DMA channel: Set the EN bit in the DMA_CCx register to 1.

20.3.20 Debug mode

When the microcontroller enters debug mode (Cortex-M0+ core halted), the counter either continues to work normally or stops, depending on TIMx_HOLD configuration bit in the *APB1 freeze register (DBG_APB1_FZ)*.

20.4 TIM3/4/5 interrupts

By setting the relevant enable bit in the TIMx_DIER register, the following events trigger an interrupt:

Table 20-4 Interrupt requests

Interrupt event	Event flag	Enable control bit	Interrupt clear method
Update event	UIF	UIE	UIF cleared by writing it to ‘0’
Capture/Compare event (1~4)	CCxIF	CCxIE	CCxIF cleared by writing it to ‘0’ or by reading the captured data stored in the TIMx_CCx register.
Trigger event	TIF	TIE	TIF cleared by writing it to ‘0’

20.5 TIM3/4/5 registers

The TIM3/4/5 registers can only be accessed by words (32-bit).

Table 20-5 TIMx base address (x = 3, 4, 5)

Peripheral	Base address
TIM3	0x4000 0400
TIM4	0x4000 0800
TIM5	0x4000 0C00

20.5.1 TIM control register 1 (TIMx_CR1)

Address offset: 0x00

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.						CLK_DIV[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Name	Description
31:10	Reserved	Must be kept at reset value
9:8	CLK_DIV[1:0]	Clock division This bit-field indicates the division ratio between the timer clock frequency and sampling clock (t_{DTS}) used by the digital filters 00: $f_{DTS} = f_{TIMx\_KCLK}$ 01: $f_{DTS} = f_{TIMx\_KCLK}/2$ 10: $f_{DTS} = f_{TIMx\_KCLK}/4$ 11: Reserved ($f_{DTS} = f_{TIMx\_KCLK}$)
7	ARPE	Auto-reload preload enable 0: Disable 1: Enable
6:5	CMS	Center-aligned mode selection

00: Edge-aligned mode. The counter counts up or down depending on the direction bit (DIR).
01: Center-aligned mode 1. The counter counts up and down alternatively. Output compare interrupt flags (CCxIF) are set only when the counter is counting down
10: Center-aligned mode 2. The counter counts up and down alternatively. Output compare interrupt flags (CCxIF) are set only when the counter is counting up
11: Center-aligned mode 3. The counter counts up and down alternatively. Output compare interrupt flags (CCxIF) are set only when the counter is counting down or up
Note: It is not allowed to switch from edge-aligned mode to center-aligned mode as long as the counter is enabled (CEN = 1)

4	DIR	Direction This bit is read only when the timer is configured in center-aligned mode or encoder mode. 0: Counter used as upcounter 1: Counter used as downcounter
3	OPM	One-pulse mode 0: Counter is not stopped at update event 1: Counter stops counting at the next update event (clearing the CEN bit)
2	URS	Update request source When UDIS is cleared: 0: Any of the following events generate an UEV: – Counter overflow/underflow – Setting the UG bit – Update generation through the slave mode controller 1: Only counter overflow/underflow generates an UEV *Note: Setting the UDIS bit to 1, if the UG bit is then set to 1, or if the counter overflows, the counter and the prescaler counter are reinitialized, but the UIF flag is not set.*
1	UDIS	Update disable

0: Enable: The URS bit determines the trigger source for the update event.

1: Disable

Note: Setting the UDIS bit to 1, if the UG bit is then set to 1, or if the counter overflows, the counter and the prescaler counter are reinitialized, but the UIF flag is not set.

0	CEN	Counter enable 0: Disable 1: Enable *Note: External clock, gated mode and encoder mode can work only if the CEN bit has been previously set by software. However trigger mode can set the CEN bit automatically by hardware.*

20.5.2 TIM control register 2 (TIMx_CR2)

Address offset: 0x04

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								TI1_XOR_SEL	MM_SEL[2:0]			CC_DMASEL	Res.		
								rw	rw	rw	rw	rw			

Bits	Name	Description
31:8	Reserved	Must be kept at reset value
7	TI1_XOR_SEL	TI1 selection 0: The TIMx_CH1 pin is connected to TI1 input 1: The TIMx_CH1, CH2 and CH3 pins are connected to the TI1 input (XOR combination)
6:4	MM_SEL[2:0]	Master mode selection These bits allow to select the information to be sent in master mode to slave timers for synchronization (TRIG_OUT). 000: Reset - The following signals can be used as trigger outputs

(TRIG_OUT):

- The UG bit in the TIMx_EVTG register
- Trigger input generation (when the slave mode controller is configured in reset mode), but the signal on TRIG_OUT will have a certain delay compared to the actual reset

001: Enable - The counter enable signal is used as trigger output (TRIG_OUT).
The counter enable signal is generated by a logic AND between CEN control bit in the TIMx_CR1 register and the trigger input when configured in gated mode. To ensure perfect synchronization between the current timer and the slave timer, the master/slave mode must be selected by setting the MS_MOD bit in the TIMx_SMC register to 1.

010: Update - The update event is selected as trigger output (TRIG_OUT)

011: Compare Pulse - The trigger output sends a positive pulse when the CC1IF flag is to be set (even if it was already high), as soon as a capture or a compare match occurred (TRIG_OUT)

100: Compare - OC1REF signal is used as trigger output (TRIG_OUT)

101: Compare - OC2REF signal is used as trigger output (TRIG_OUT)

110: Compare - OC3REF signal is used as trigger output (TRIG_OUT)

111: Compare - OC4REF signal is used as trigger output (TRIG_OUT)

3	CC_DMASEL	Capture/Compare DMA selection 0: CCx DMA requests sent when CCx event occurs 1: CCx DMA requests sent when update event occurs
2:0	Reserved	Must be kept at reset value

20.5.3 TIM slave mode control register (TIMx_SMC)

Address offset: 0x08

Reset value: 0x0000 0000

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

Bits	Name	Description
31:16	Reserved	Must be kept at reset value
15	ETP	ETR polarity 0: ETR is non-inverted, active at high level or rising edge 1: ETR is inverted, active at low level or falling edge
14	ECEN	Clock mode 2 enable It is possible to simultaneously use external clock mode 2 with the following slave modes: – Reset mode – Gated mode – Trigger mode If clock mode 1 and clock mode 2 are enabled at the same time, the clock mode 2 has a higher priority. 0: Disable 1: Enable
13:12	ET_PRE	ETR prescaler control ETRP frequency must be at most ½ of TIMx_KCLK frequency. 00: Prescaler off 01: ETRP frequency divided by 2 10: ETRP frequency divided by 4 11: ETRP frequency divided by 8
11	Reserved	Must be kept at reset value
10:8	ETF[2:0]	External trigger filter The sampling frequency of the ETRP signal and the digital filter

		width for ETRP. A valid level is only recognized if it is sampled for four consecutive sampling cycles. 000: No filter, sampling is done at f_{DTS} 001: $f_{SAMPLING} = f_{TIMx\_KCLK}$ 010: $f_{SAMPLING} = f_{DTS}/2$ 011: $f_{SAMPLING} = f_{DTS}/4$ 101: $f_{SAMPLING} = f_{DTS}/8$ 101: $f_{SAMPLING} = f_{DTS}/16$ 110: $f_{SAMPLING} = f_{DTS}/32$ 111: $f_{SAMPLING} = f_{DTS}/32$
7	MS_MOD	Master/Slave mode 0: No action 1: The effect of an event on the trigger input (TRIG) is delayed to allow a perfect synchronization between the current timer and its slaves (through TRIG_OUT). It is useful if we want to synchronize several timers on a single external event
6:4	TS[2:0]	Trigger selection 000: Internal Trigger 0 (ITR0) 001: Internal Trigger 1 (ITR1) 010: Internal Trigger 2 (ITR2) 011: Internal Trigger 3 (ITR3) 100: TI1 Edge Detector (TI1F_ED) 101: Filtered Timer Input 1 (TI1FP1) 110: Filtered Timer Input 2 (TI2FP2) 111: ETR input (ETRF) *Note: See Table: TIMx internal trigger connection for more details on ITRx meaning.* *These bits must be changed only when they are not used (when SM_SEL = 000) to avoid wrong edge detections at the transition.*
3	OCCS	OCREF clear selection 0: OCREF_CLR is connected to COMP1 or COMP2 output depending on the OCREF_CLR bit in the TIMx_CFG register 1: OCREF_CLR is connected to ETRF

2:0	SM_SEL[2:0]	Slave mode selection When external signals are selected the active edge of the trigger signal (TRIG) is linked to the polarity selected on the external input. 000: Slave mode disabled -- The PSC prescaler clock is provided by the internal clock. When the CEN bit in the TIMx_CR1 register is 1, counting is started 001: Encoder mode 1 -- Counter counts up/down on TI1FP1 edge depending on TI2FP2 level 010: Encoder mode 2 -- Counter counts up/down on TI2FP2 edge depending on TI1FP1 level 011: Encoder mode 3 -- Counter counts up/down on both TI1FP1 and TI2FP2 edges depending on the level of the other input 100: Reset mode -- Rising edge of the selected trigger input (TRIG) reinitializes the counter and generates an update of the registers 101: Gated mode -- The counter clock is enabled when the trigger input (TRIG) is high. The counter stops (but is not reset) as soon as the trigger becomes low. Both start and stop of the counter are controlled 110: Trigger Mode -- The counter starts at a rising edge of the trigger TRGI (but it is not reset). Only the start of the counter is controlled 111: Clock Mode 1 - Rising edges of the selected trigger (TRIG) clock the counter *Note: The gated mode must not be used if TI1F_ED is selected as the trigger input (TS = 0100). Indeed, TI1F_ED outputs 1 pulse for each transition on TI1F, whereas the gated mode checks the level of the trigger signal.*

20.5.4 TIM interrupt/DMA enable register (TIMx_DIER)

Address offset: 0x0C

Reset value: 0x0000 0000

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

Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDMA_EN	Res.	CC4_DMAEN	CC3_DMAEN	CC2_DMAEN	CC1_DMAEN	UDMA_EN	Res.	TIE	Res.	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	rw		rw	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw

Bits	Name	Description
31:15	Reserved	Must be kept at reset value
14	TDMA_EN	Trigger DMA request enable 0: Disable 1: Enable
13	Reserved	Must be kept at reset value
12	CC4_DMAEN	Capture/Compare 4 DMA request enable 0: Disable 1: Enable
11	CC3_DMAEN	Capture/Compare 3 DMA request enable 0: Disable 1: Enable
10	CC2_DMAEN	Capture/Compare 2 DMA request enable 0: Disable 1: Enable
9	CC1_DMAEN	Capture/Compare 1 DMA request enable 0: Disable 1: Enable
8	UDMA_EN	Update DMA request enable 0: Disable 1: Enable
7	Reserved	Must be kept at reset value
6	TIE	Trigger interrupt enable

		0: Disable 1: Enable
5	Reserved	Must be kept at reset value
4	CC4IE	Capture/Compare 4 interrupt enable 0: Disable 1: Enable
3	CC3IE	Capture/Compare 3 interrupt enable 0: Disable 1: Enable
2	CC2IE	Capture/Compare 2 interrupt enable 0: Disable 1: Enable
1	CC1IE	Capture/Compare 1 interrupt enable 0: Disable 1: Enable
0	UIE	Update interrupt enable 0: Disable 1: Enable

20.5.5 TIM status register (TIMx_SR)

Address offset: 0x10

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			CC4OF	CC3OF	CC2OF	CC1OF	Res.		TIF	Res.	CC4IF	CC3IF	CC2IF	CC1IF	UIF
			rc_w0	rc_w0	rc_w0	rc_w0			rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

Bits	Name	Description
31:13	Reserved	Must be kept at reset value

12	CC4OF	Capture/Compare 4 over capture flag Refer to CC1OF description
11	CC3OF	Capture/Compare 3 over capture flag Refer to CC1OF description
10	CC2OF	Capture/Compare 2 over capture flag Refer to CC1OF description
9	CC1OF	Capture/Compare 1 over capture flag This flag is cleared by software by writing it to '0'. 0: No over capture has been detected 1: The counter value has been captured in the TIMx_CC1 register while CC1IF flag was already set
8:7	Reserved	Must be kept at reset value
6	TIF	Trigger interrupt flag – Gated mode: The flag is set when the counter starts or stop – No gated mode: The flag is set when active edge detected on TRGI input when the slave mode controller is enabled This flag is cleared by software by writing it to '0'. 0: No trigger event occurred 1: Trigger interrupt pending
5	Reserved	Must be kept at reset value
4	CC4IF	Capture/Compare 4 interrupt flag Refer to CC1IF description
3	CC3IF	Capture/Compare 3 interrupt flag Refer to CC1IF description
2	CC2IF	Capture/Compare 2 interrupt flag Refer to CC1IF description

1	CC1IF	Capture/Compare 1 interrupt flag If channel CC1 is configured as output: This flag is set when the content of the counter TIMx_CNT matches the content of the TIMx_CC1 register. There are 3 possible options for flag setting in center-aligned mode, refer to the CMS bits in the TIMx_CR1 register for the full description. This flag is cleared by software by writing it to '0'. 0: No compare match 1: A compare match If channel CC1 is configured as input: This flag is set when counter value has been captured in the TIMx_CC1 register. It is cleared by software or by reading the TIMx_CC1 register. 0: No input capture occurred 1: An input capture occurred
0	UIF	Update interrupt flag 0: No update occurred 1: Update interrupt pending: – At overflow or underflow and if UDIS = 0 in the TIMx_CR1 register – Software using the UG bit in the TIMx_EVTG register, if URS = 0 and UDIS = 0 in the TIMx_CR1 register – The update event generated by the slave mode controller (reset mode), if URS = 0 and UDIS = 0 in the TIMx_CR1 register (see the TIMx_SMCR register description).

20.5.6 TIM event generation register (TIMx_EVTG)

Address offset: 0x14

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.									TG	Res.	CC4G	CC3G	CC2G	CC1G	UG
									w		w	w	w	w	w

Bits	Name	Description
31:7	Reserved	Must be kept at reset value
6	TG	Trigger generation This bit is set by software in order to generate an event.
5	Reserved	Must be kept at reset value
4	CC4G	Capture/compare 4 generation Refer to CC1G description
3	CC3G	Capture/compare 3 generation Refer to CC1G description
2	CC2G	Capture/compare 2 generation Refer to CC1G description
1	CC1G	Capture/compare 1 generation This bit is set by software in order to generate an event. If channel CC1 is configured as output: CC1IF flag is set If channel CC1 is configured as input: – The CC1IF flag is set. The CC1OF flag is set if the CC1IF flag was already high. – The current value of the counter is captured in the TIMx_CC1 register.
0	UG	Update generation This bit is set by software in order to generate an event. 0: No action 1: Re-initialize the counter and generates an update of the registers. The prescaler counter is also cleared (anyway the prescaler ratio is not affected).

20.5.7 TIM capture/compare mode register 1 (TIMx_CCM1) -- (Output compare mode)

Address offset: 0x18

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Name	Description
31:16	Reserved	Must be kept at reset value
15	OC2CE	Output compare 2 clear enable Refer to O1CE description
14:12	OC2M[2:0]	Output compare 2 mode Refer to OC1M[2:0] description
11	OC2PE	Output compare 2 preload enable Refer to OC1PE description
10	OC2FE	Output compare 2 fast enable Refer to OC1FE description
9:8	CC2S[1:0]	Capture/Compare 2 selection CC2S bits are writable only when the channel is off (CC2E = 0 in the TIMx_CCEN register). 00: CC2 channel is configured as output 01: CC2 channel is configured as input, IC2 is mapped on TI2 10: CC2 channel is configured as input, IC2 is mapped on TI1 11: CC2 channel is configured as input, IC2 is mapped on TRC. This mode is working only if an internal trigger input is selected through TS[2:0] bit (TIMx_SMC register)
7	OC1CE	Output compare 1 clear enable 0: OC1REF is not affected by OC_CLR signal 1: OC1REF is cleared as soon as a high level is detected on OC_CLR signal (OCCS bit in the TIMx_SMC register)
6:4	OC1M[2:0]	Output compare 1 mode

These bits define the behavior of the output reference signal OC1REF. The OC1REF signal determines OC1. OC1REF is active high.

000: Frozen -- The comparison between the output compare register TIMx_CC1 and the counter TIMx_CNT has no effect on the outputs. (this mode is used to generate a timing base)

001: Active level on match -- OC1REF signal is forced active level when the counter TIMx_CNT matches the TIMx_CC1 register (Before the match, the OC1REF signal is at an inactive level)

010: Inactive level on match -- OC1REF signal is forced inactive level when the counter TIMx_CNT matches the TIMx_CC1 register (Before the match, the OC1REF signal is at an inactive level)

011: Toggle -- OC1REF toggles when TIMx_CNT = TIMx_CC1

100: Force inactive level -- OC1REF is forced inactive level

101: Force active level -- OC1REF is forced active level

110: PWM mode 1:

- In upcounting, channel 1 is active as long as TIMx_CNT < TIMx_CC1 else inactive
- In downcounting, channel 1 is inactive as long as TIMx_CNT > TIMx_CC1 else active

111: PWM mode 2:

- In upcounting, channel 1 is inactive as long as TIMx_CNT < TIMx_CC1 else active
- In downcounting, channel 1 is active as long as TIMx_CNT > TIMx_CC1 else inactive

3	OC1PE	Output compare 1 preload enable 0: Disable: The TIMx_CC1 register can be written at any time, the new value is taken in account immediately 1: Enable: Read/Write operations access the preload register. The TIMx_CC1 register preload value is loaded in the active register at each update event
2	OC1FE	Output compare 1 fast enable This bit is used to accelerate the effect of an event on the trigger in input on the OC1 output. 0: Disable: OC1 behaves normally depending on counter and CC1

values even

1: Enable: An active edge on the trigger input acts like a compare match on OC1 output. Then, OC1 is set to the compare level independently from the result of the comparison. OC1FE acts only if the channel is configured in PWM1 or PWM2 mode.

1:0	CC1S[1:0]	Capture/Compare 1 selection CC1S bits are writable only when the channel is off (CC1E = 0 in the TIMx_CCEN register). 00: CC1 channel is configured as output 01: CC1 channel is configured as input, IC1 is mapped on TI1 10: CC1 channel is configured as input, IC1 is mapped on TI2 11: CC1 channel is configured as input, IC1 is mapped on TRC. This mode is working only if an internal trigger input is selected through TS[2:0] bits (TIMx_SMC register)

20.5.8 TIM capture/compare mode register 1 (TIMx_CCM1) -- (Input capture mode)

Address offset: 0x18

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	IC2F[2:0]			IC2PSC[1:0]		CC2S[1:0]		Res.	IC1F[2:0]			IC1PSC[1:0]		CC1S[1:0]	
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

Bits	Name	Description
31:15	Reserved	Must be kept at reset value
14:12	IC2F[2:0]	Input capture 2 filter Refer to IC1F[2:0] description
11:10	IC2PSC[1:0]	input compare 2 prescaler Refer to OC1PS[1:0] description *Note: This bit field is read-write only for TIM3, writing to it has no effect on TIM4 and TIM5.*

9:8	CC2S[1:0]	Capture/Compare 2 selection Refer to CC2S[1:0] bits description in the *TIM caputre/compare mode register1 (TIMx_CCM1) – (Output compare mode)*
7	Reserved	Must be kept at reset value
6:4	IC1F[2:0]	Input capture 1 filter This bit-field defines the frequency used to sample TI1. The digital filter is made of an event counter in which 4 consecutive events are needed to validate a transition on the output: 000: No filter, sampling is done at f_{DTS} 001: $f_{SAMPLING} = f_{TIMx\_KCLK}$ 010: $f_{SAMPLING} = f_{DTS}/2$ 011: $f_{SAMPLING} = f_{DTS}/4$ 100: $f_{SAMPLING} = f_{DTS}/8$ 101: $f_{SAMPLING} = f_{DTS}/16$ 110: $f_{SAMPLING} = f_{DTS}/32$ 111: $f_{SAMPLING} = f_{DTS}/32$
3:2	IC1PSC[1:0]	Input capture 1 prescaler The prescaler is reset as soon as CC1E = 0 (TIMx_CCEN register). 00: No prescaler, capture is done each time an edge is detected on the capture input 01: Capture is done once every 2 events 10: Capture is done once every 4 events 11: Capture is done once every 8 events *Note: This bit field is read-write only for TIM3, writing to it has no effect on TIM4 and TIM5.*
1:0	CC1S[1:0]	Capture/Compare 1selection Refer to CC1S[1:0] bits description in the *TIM caputre/compare mode register1 (TIMx_CCM1) – (Output compare mode)*

20.5.9 TIM capture/compare mode register 2 (TIMx_CCM2) -- (Output compare mode)

Address offset: 0x1C

Reset value: 0x0000 0000

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

Res.															
							rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M[2:0]			OC4PE	OC4FE	CC4S[1:0]		OC3CE	OC3M[2:0]			OC3PE	OC3FE	CC3S[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Name	Description
31:16	Reserved	Must be kept at reset value
15	OC4CE	Output compare 4 clear enable Refer to OC1CE description
14:12	OC4M[2:0]	Output compare 4 mode Refer to OC1M[1:0] description
11	OC4PE	Output compare 4 preload enable Refer to OC1PE description
10	OC4FE	Output compare 4 fast enable Refer to OC1FE description
9:8	CC4S[1:0]	Capture/Compare 4 selection CC4S bits are writable only when the channel is off (CC4E = 0 in TIMx_CCEN register). 00: CC4 channel is configured as output 01: CC4 channel is configured as input, IC4 is mapped on TI4 10: CC4 channel is configured as input, IC4 is mapped on TI3 11: CC4 channel is configured as input, IC4 is mapped on TRC. This mode is working only if an internal trigger input is selected through TS[2:0] bits (TIMx_SMC register)
7	OC3CE	Output compare 3 clear enable Refer to OC1CE description
6:4	OC3M[2:0]	Output compare 3 mode Refer to OC1M[1:0] description
3	OC3PE	Output compare 3 preload enable

		Refer to OC1PE description
2	OC3FE	Output compare 3 fast enable Refer to OC1FE description
1:0	CC3S[1:0]	Capture/Compare 3 selection CC3S bits are writable only when the channel is off (CC3E = 0 in TIMx_CCEN register). 00: CC3 channel is configured as output 01: CC3 channel is configured as input, IC3 is mapped on TI3 10: CC3 channel is configured as input, IC3 is mapped on TI4 11: CC3 channel is configured as input, IC3 is mapped on TRC. This mode is working only if an internal trigger input is selected through TS[2:0] bits (TIMx_SMC register)

20.5.10 TIM capture/compare mode register 2 (TIMx_CCM2) -- (Input capture mode)

Address offset: 0x1C

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	IC4F[2:0]			IC4PSC[1:0]		CC4S[1:0]		Res.	IC3F[2:0]			IC3PSC[1:0]		CC3S[1:0]	
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

Bits	Name	Description
31:15	Reserved	Must be kept at reset value
14:12	IC4F[2:0]	Input capture 4 filter Refer to IC1F[2:0] description
11:10	IC4PSC[1:0]	input compare 4 prescaler Refer to OC1PS[1:0] description *Note: This bit field is read-write only for TIM3, writing to it has no effect on TIM4 and TIM5.*
9:8	CC4S[1:0]	Capture/Compare 4 selection

		Refer to CC4S[1:0] bits description in the *TIM caputre/compare mode regiser 2 (TIMx_CCM2) -- (Output compare mode)*
7	Reserved	Must be kept at reset value
6:4	IC3F[2:0]	Input capture 3 filter Refer to IC1F[2:0] description
3:2	IC3PSC[1:0]	input compare 3 prescaler Refer to OC1PS[1:0] description *Note: This bit field is read-write only for TIM3, writing to it has no effect on TIM4 and TIM5.*
1:0	CC3S[1:0]	Capture/Compare 3 selection Refer to CC3S[1:0] bits description in the *TIM caputre/compare mode regiser 2 (TIMx_CCM2) -- (Output compare mode)*

20.5.11 TIM capture/compare enable register (TIMx_CCEN)

Address offset: 0x20

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4NP	Res.	CC4P	CC4E	CC3NP	Res.	CC3P	CC3E	CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E
rw		rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw

Bits	Name	Description
31:16	Reserved	Must be kept at reset value
15	CC4NP	Capture/Compare 4 complementary output polarity Refer to CC1NP description, valid only when configured as input
14	Reserved	Must be kept at reset value
13	CC4P	Capture/Compare 4 output Polarity Refer to CC1P description

12	CC4E	Capture/Compare 4 output enable. Refer to CC1E description
11	CC3NP	Capture/Compare 3 complementary output polarity Refer to CC1NP description, valid only when configured as input
10	Reserved	Must be kept at reset value
9	CC3P	Capture/Compare 3 output Polarity Refer to CC1P description
8	CC3E	Capture/Compare 3 output enable. Refer to CC1E description
7	CC2NP	Capture/Compare 2 complementary output polarity Refer to CC1NP description, valid only when configured as input
6	Reserved	Must be kept at reset value
5	CC2P	Capture/Compare 2 output Polarity Refer to CC1P description
4	CC2E	Capture/Compare 2 output enable Refer to CC1E description
3	CC1NP	Capture/Compare 1 complementary output polarity CC1 channel configured as output: CC1NP must be kept cleared in this case. CC1 channel configured as input: This bit is used in conjunction with CC1P to define TI1FP1/TI2FP1 polarity. Refer to CC1P description.
2	Reserved	Must be kept at reset value
1	CC1P	Capture/Compare 1 output polarity CC1 channel configured as output: 0: OC1 active high

1: OC1 active low

CC1 channel configured as input:

CC1NP/CC1P bits select the active polarity of TI1FP1 and TI2FP1 for trigger or capture operations:

- CC1NP = 0, CC1P = 0:
  TIxFP1 rising edge (capture or trigger operations in reset, external clock or trigger mode)
  TIxFP1 is not inverted (trigger operation in gated mode or encoder mode)
- CC1NP = 0, CC1P = 1:
  TIxFP1 falling edge (capture or trigger operations in reset, external clock or trigger mode)
  TIxFP1 is inverted (trigger operation in gated mode or encoder mode)
- CC1NP = 1, CC1P = 1:
  both TIxFP1 rising and falling edges (capture or trigger operations in reset, external clock or trigger mode)
  TIxFP1 is not inverted (trigger operation in gated mode). This configuration must not be used in encoder mode.
- CC1NP = 1, CC1P = 0: Reserved (default: CC1NP = 0, CC1P = 0)

0	CC1E	Capture/Compare 1 output enable CC1 channel configured as output: 0: Disable 1: Enable CC1 channel configured as input: Input capture enable 0: Disable 1: Enable

20.5.12 TIM counter register (TIMx_CNT)

Address offset: 0x24

Reset value: 0x0000 0000

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

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

Bits	Name	Description
31:16	Reserved	Must be kept at reset value
15:0	CNT[15:0]	Counter value

20.5.13 TIM prescaler register (TIMx_PSC)

Address offset: 0x28

Reset value: 0x0000 0000

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

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

Bits	Name	Description
31:16	Reserved	Must be kept at reset value
15:0	PSC[15:0]	Prescaler value The counter clock frequency CK_CNT is equal to f_{CK_PSC} / (PSC[15:0] + 1). PSC contains the value to be loaded in the active prescaler register at each update event.

20.5.14 TIM auto-reload register (TIMx_ARR)

Address offset: 0x2C

Reset value: 0x0000 FFFF

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

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

ARR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Name	Description
31:16	Reserved	Must be kept at reset value
15:0	ARR[15:0]	Auto-reload value The counter is blocked while the auto-reload value is null.

20.5.15 TIM capture/compare register 1 (TIMx_CC1)

Address offset: 0x34

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Name	Description
31:16	Reserved	Must be kept at reset value
15:0	CC1[15:0]	Capture/Compare 1 value If CC1 channel is configured as output: – It is loaded permanently if the preload feature is not selected in the TIMx_CCM1 register (bit OC1PE = 0) – The preload value is copied in the active capture/compare 1 register when an update event occurs (bit OC1PE = 1) If CC1 channel is configured as input: The TIMx_CC1 register is read-only and cannot be programmed. CC1 is the counter value transferred by the last input capture 1 event.

20.5.16 TIM capture/compare register 2 (TIMx_CC2)

Address offset: 0x38

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Name	Description
31:16	Reserved	Must be kept at reset value
15:0	CC2[15:0]	Capture/Compare 2 value If CC2 channel is configured as output: – It is loaded permanently if the preload feature is not selected in the TIMx_CCM1 register (bit OC2PE = 0) – The preload value is copied in the active capture/compare 2 register when an update event occurs (bit OC2PE = 1) If CC2 channel is configured as input: The TIMx_CC2 register is read-only and cannot be programmed. CC2 is the counter value transferred by the last input capture 2 event.

20.5.17 TIM capture/compare register 3 (TIMx_CC3)

Address offset: 0x3C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC3[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Name	Description
31:16	Reserved	Must be kept at reset value
15:0	CC3[15:0]	Capture/Compare 3 value If CC3 channel is configured as output:

- It is loaded permanently if the preload feature is not selected in the TIMx_CCM2 register (bit OC3PE = 0)
- The preload value is copied in the active capture/compare 3 register when an update event occurs (bit OC3PE = 1)

If CC3 channel is configured as input:

The TIMx_CC3 register is read-only and cannot be programmed.

CC3 is the counter value transferred by the last input capture 3 event.

20.5.18 TIM capture/compare register 4 (TIMx_CC4)

Address offset: 0x40

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Name	Description
31:16	Reserved	Must be kept at reset value
15:0	CC4[15:0]	Capture/Compare 4 value If CC4 channel is configured as output: - It is loaded permanently if the preload feature is not selected in the TIMx_CCM2 register (bit OC4PE = 0) - The preload value is copied in the active capture/compare 4 register when an update event occurs (bit OC4PE = 1) If CC4 channel is configured as input: The TIMx_CC4 register is read-only and cannot be programmed. CC4 is the counter value transferred by the last input capture 4 event.

20.5.19 TIM configuration register (TIMx_CFG)

Address offset: 0x50

Reset value: 0x0000 0000

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

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

Bits	Name	Description
31:1	Reserved	Must be kept at reset value
0	OCREF_CLR	OCxREF clear source selection 0: COMP1 output is connected to the OCREF_CLR input 1: COMP2 output is connected to the OCREF_CLR input

20.5.20 TIM3 alternate function register (TIM3_AF1)

Address offset: 0x60

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															ETR_SEL[2]
															rw

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

Bits	Name	Description
31:17	Reserved	Must be kept at reset value
16:14	ETR_SEL[2:0]	ETR input selection 000: GPIO 001: COMP1 output 010: COMP2 output 011: LXTAL 100: HXTAL 101: MCO 110: RCL

111: Reserved (GPIO)

13:0	Reserved	Must be kept at reset value

20.5.21 TIM4/5 alternate function register (TIMx_AF1)

Address offset: 0x60

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															ETR_SEL[2]
															rw

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

Bits	Name	Description
31:17	Reserved	Must be kept at reset value
16:14	ETR_SEL[2:0]	ETR input selection 000: GPIO 001: COMP1 output 010: COMP2 output Other: Reserved (GPIO)
13:0	Reserved	Must be kept at reset value

20.5.22 TIM3 input selection register (TIM3_TISEL)

Address offset: 0x68

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.					TI4_SEL[2:0]			Res.							
					rw	rw	rw								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.						TI2_SEL[1:0]		Res.						TI1_SEL[1:0]	
						rw	rw							rw	rw

Bits	Name	Description

31:27	Reserved	Must be kept at reset value
26:24	TI4_SEL[2:0]	TI4 input selection 000: TIM3_CH4 001: LXTAL 010: HXTAL 011: MCO 100: RCL Other: Reserved (TIM3_CH4)
23:10	Reserved	Must be kept at reset value
9:8	TI2_SEL[1:0]	TI2 input selection 00: TIM3_CH2 01: COMP2 output Other: Reserved (TIM3_CH2)
7:2	Reserved	Must be kept at reset value
1:0	TI1_SEL[1:0]	TI1 input selection 00: TIM3_CH1 01: COMP1 output Other: Reserved (TIM3_CH1)

20.5.23 TIM4/5 input selection register (TIMx_TISEL)

Address offset: 0x68

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.						TI2_SEL[1:0]		Res.						TI1_SEL[1:0]	
						rw	rw							rw	rw

Bits	Name	Description
31:10	Reserved	Must be kept at reset value

9:8	TI2_SEL[1:0]	TI2 input selection 00: TIMx_CH2 01: COMP2 output Other: Reserved (TIMx_CH2)
7:2	Reserved	Must be kept at reset value
1:0	TI1_SEL[1:0]	TI1 input selection 00: TIMx_CH1 01: COMP1 output Other: Reserved (TIMx_CH1)

21 Basic timer (TIM8)

21.1 Introduction

The basic timer consists of a 16-bit auto-reload counter driven by a programmable prescaler. The timer is internally connected to the ADC.

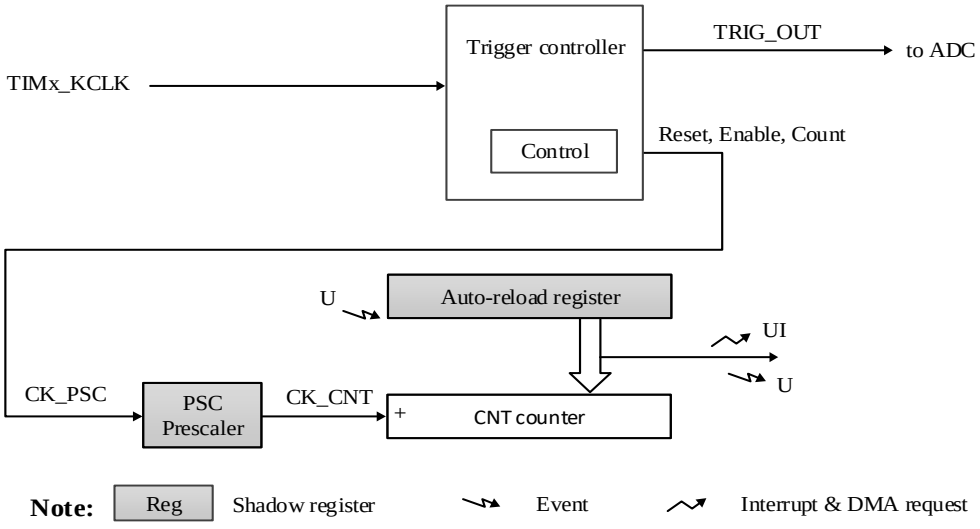
21.2 TIM8 main features

- 16-bit auto-reload upcounter
- 16-bit programmable prescaler
- ADC can be triggered periodically

21.3 TIM8 functional description

21.3.1 TIM8 block diagram

Figure 21-1 Basic timer block diagram (x = 8)



21.3.2 TIM8 internal signals

Table 21-1 TIMx internal signals

Signal name	Signal type	Description
TRIG_OUT	Output	Trig out signal

21.3.3 Prescaler

The prescaler can divide the counter clock frequency by any factor between 1 and 65536. The PSC[15:0] bits in the TIMx_PSC register are used to configure the

prescaler value. The actual prescaler factor is PSC[15:0] + 1.

It can be changed on the fly as the TIMx_PSC register is buffered. The new prescaler ratio is taken into account at the next update event.

The following figures show some examples of the counter behavior when the prescaler ratio is changed on the fly.

Figure 21-2 Counter timing diagram with prescaler division change from 1 to 2

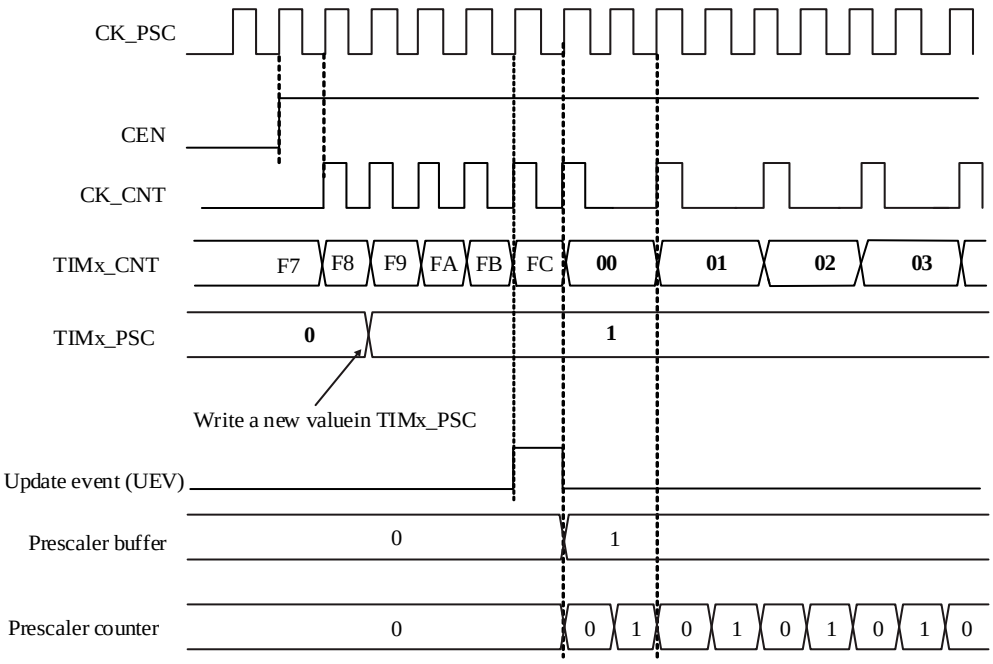
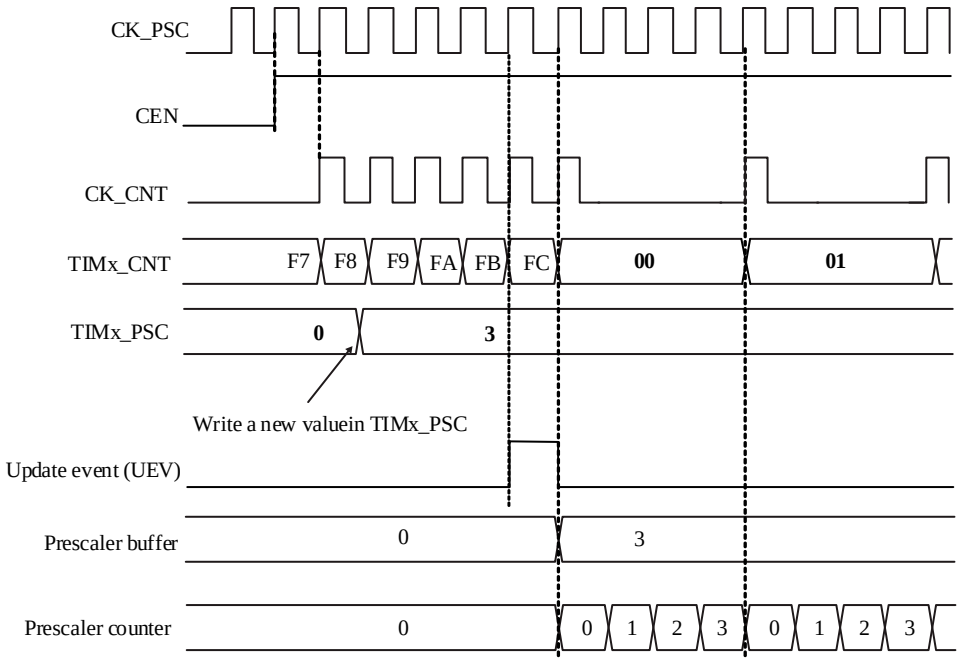


Figure 21-3 Counter timing diagram with prescaler division change from 1 to 4



21.3.4 Counter mode

TIM8 has an internal 16-bit counter. The upcounting mode, the counter counts from 0 to the auto-reload value (contents of the TIMx_ARR register), then generates a counter overflow event, the UIF flag of the TIMx_SR register will be automatically set to 1, and the counter will be cleared to 0 and starts counting again.

The counter configuration supports dynamic modification, and the TIMx_CNT register, TIMx_ARR register and TIMx_PSC register can be read and written while the counter is running.

The ARR[15:0] bit field in the TIMx_ARR register can be used to configure the auto-reload value, which has a buffering function (shadow register).

If the APRE bit in the TIMx_CR1 register is '0', the update of the shadow register is no affected by the updte event, and the value of the TIMx_ARR register is updated immediately. If the APRE bit is set, the value in the TIMx_ARR register is updated to the shadow register when a UEV event occurs.

The UEV event can be disabled by software by setting the UDIS bit in the TIMx_CR1 register. This avoids updating the shadow registers while writing new values into the preload registers.

If the URS (update request selection) bit in the TIMx_CR1 register is set, setting the UG bit generates an update event UEV, and both the counter and the pre-division counter will be initialized, but the UIF flag is not set.

Select the update interrupt (UEV) sources by the URS bit:

- 0: Any of the following events generate an UEV if enabled:
  - Counter overflow
  - Setting the UG big in TIMx_EVTG register
- 1: Only counter overflow generates an UEV if enabled.

When an update event occurs, all the registers are updated and the update flag (UIF bit in the TIMx_SR register) is set:

- The auto-reload shadow register is updated with the preload value (content of the TIMx_ARR register)
- The buffer of the prescaler is reloaded with the preload value (content of the TIMx_PSC register)

The following figures show some examples of the counter behavior for different clock frequencies when TIMx_ARR = 0x36:

Figure 21-4 Counting timing diagram, overflow event when the internal clock divided by 1

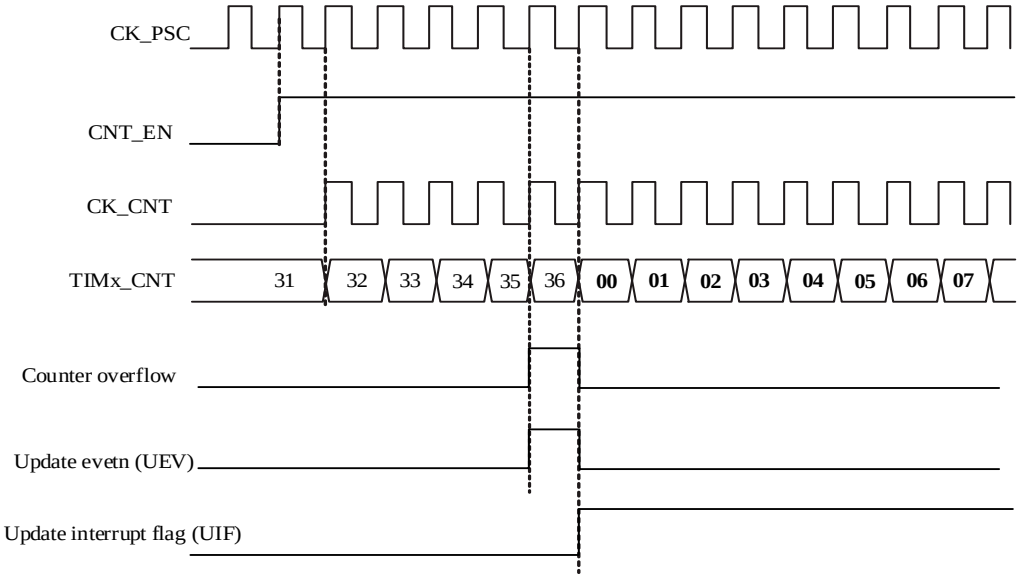


Figure 21-5 Counting timing diagram, overflow event when the internal clock divided by 4

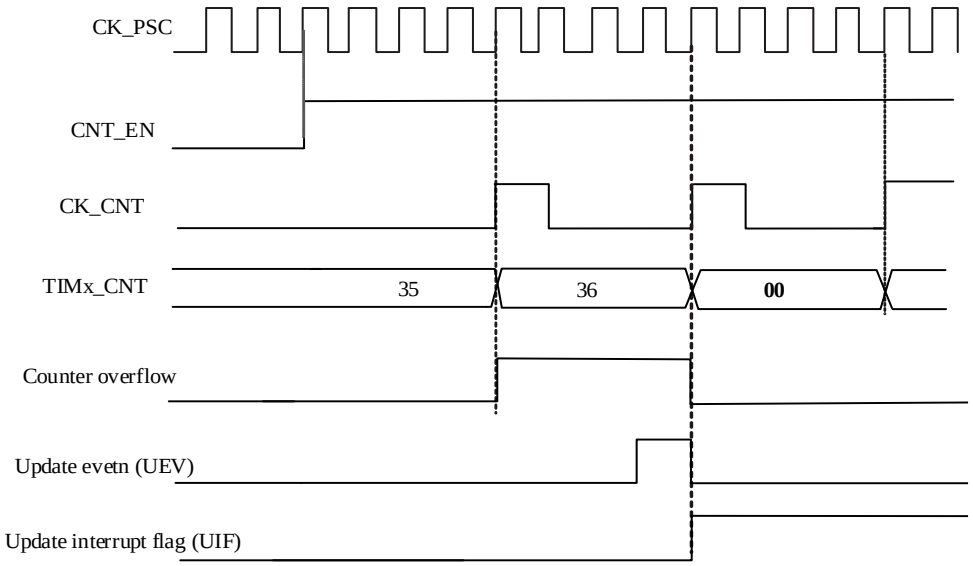


Figure 21-6 Counter timing diagram, update event when ARPE = 0

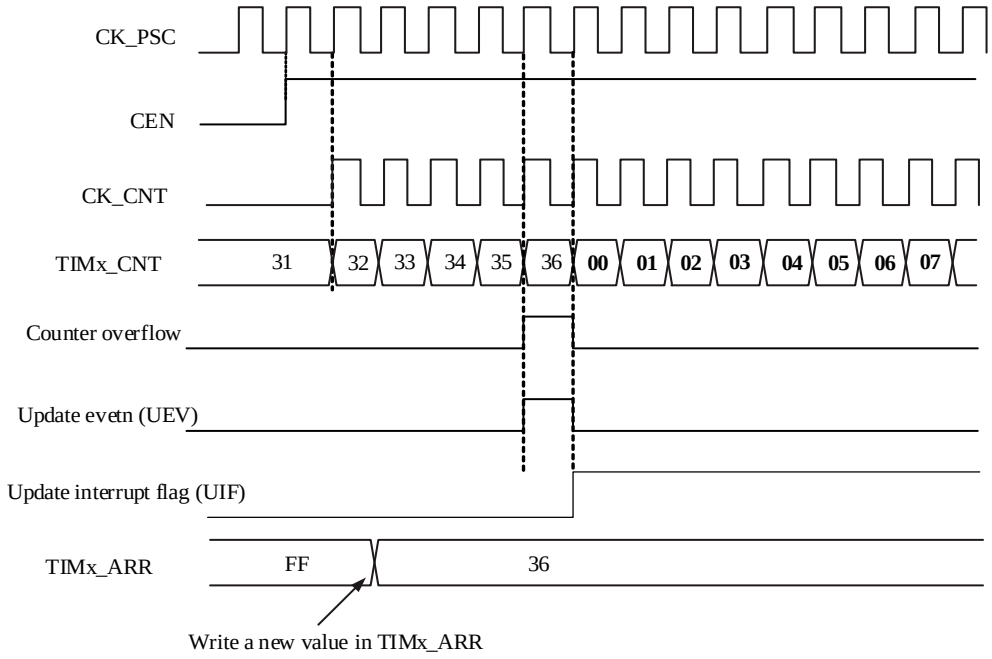
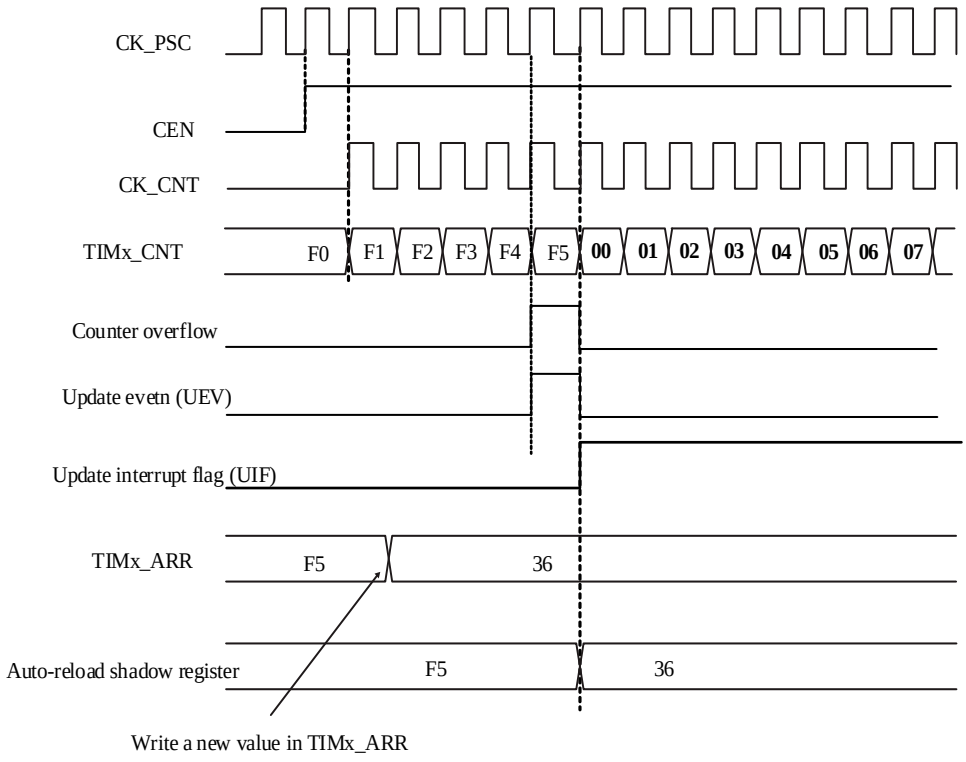


Figure 21-7 Counter timing diagram, update event when ARPE = 1

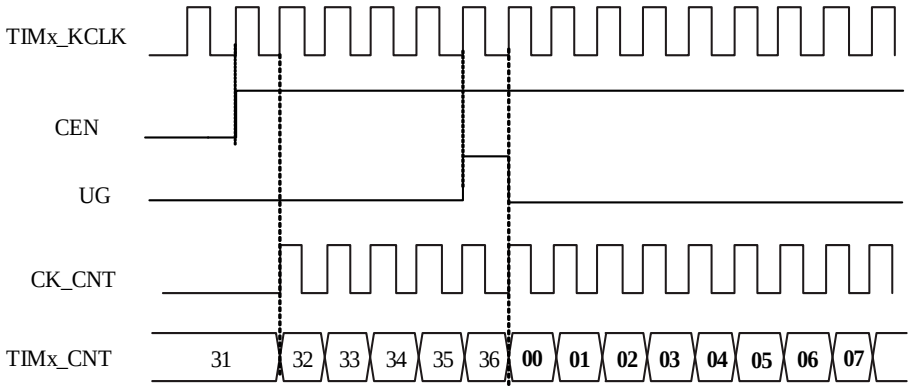


21.3.5 Clock selection

The counter clock is TIM8_KCLK, which comes from the TIMx_PCLK of the RCC.

The following figure shows the behavior of the control circuit.

Figure 21-8 Control circuit in normal mode, internal clock divided by 1



21.3.6 Debug mode

When the microcontroller enters debug mode (Cortex-M0+ core halted), the counter either continues to work normally or stops, depending on TIM8_HOLD configuration bit in the *APB1 freeze register (DBG_APB1_FZ)*.

21.4 TIM8 interrupts

By setting the relevant enable bit in the TIMx_DIER register, the following events trigger an interrupt:

Table 21-2 Interrupt requests

Interrupt event	Event flag	Enable control bit	Interrupt clear method
Update event	UIF	UIE	UIF can be cleared by writing it to ‘0’

21.5 TIM8 registers

The TIM8 registers can only be accessed by words (32-bit).

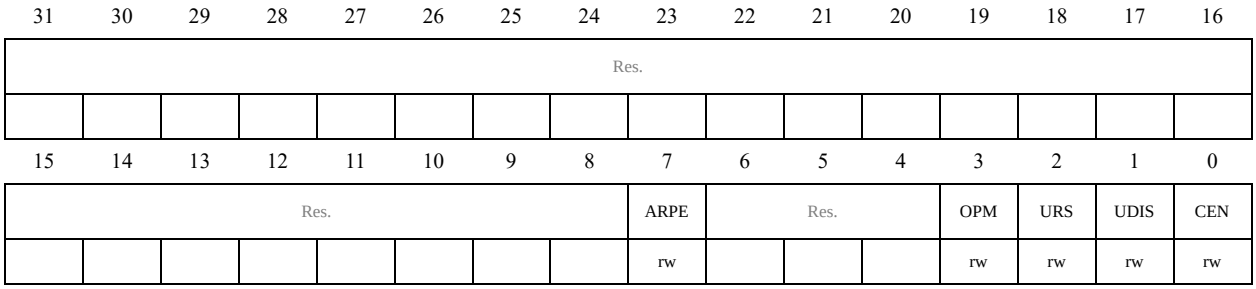
Table 21-3 TIM8 base address

Peripheral	Base address
TIM8	0x4000 1000

21.5.1 TIM control register 1 (TIMx_CR1)

Address offset: 0x00

Reset value: 0x0000 0000



Bits	Name	Description
31:8	Reserved	Must be kept at reset value
7	ARPE	Auto-reload preload enable 0: Disable 1: Enable
6:4	Reserved	Must be kept at reset value
3	OPM	One-pulse mode 0: Counter is not stopped at update event 1: Counter stops counting at the next update event (clearing the CEN bit)
2	URS	Update request source When UDIS is cleared: 0: Any of the following events generate an UEV: – Counter overflow – Setting the UG bit

1: Only counter overflow generates an UEV

Note: When the UDIS bit is 0 and the URS bit is set to 1, setting the UG bit generates an update event, initializing the counter and the prescaler counter, but does not set the UIF flag.

1	UDIS	Update disable 0: Enable: The URS bit determines the trigger source for the update event. 1: Disable *Note: Setting the UDIS bit to 1, if the UG bit is then set to 1, or if the counter overflows, the counter and the prescaler counter are reinitialized, but the UIF flag is not set.*
0	CEN	Counter enable 0: Disable 1: Enable

21.5.2 TIM control register 2 (TIMx_CR2)

Address offset: 0x04

Reset value: 0x0000 0000

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

Bits	Name	Description
31:7	Reserved	Must be kept at reset value
6:4	MM_SEL[2:0]	These bits are used to select the information to be sent in master mode to slave timers for synchronization (TRIG_OUT). 000: Reset - The following signals can be used as trigger outputs (TRIG_OUT): – The UG bit in the TIMx_EVTG register 001: Enable - the counter enable signal is used as trigger output

(TRIG_OUT). The counter enable signal is controlled by the CEN bit in the TIMx_CR1 register.

010: Update - The update event is selected as trigger output (TRIG_OUT)

Other: Reserved (default: Reset)

3:0	Reserved	Must be kept at reset value

21.5.3 TIM interrupt/DMA enable register (TIMx_DIER)

Address offset: 0x0C

Reset value: 0x0000 0000

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

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

Bits	Name	Description
31:9	Reserved	Must be kept at reset value
8	UDMA_EN	Update DMA request enable 0: Disable 1: Enable
7:1	Reserved	Must be kept at reset value
0	UIE	Update interrupt enable 0: Disable 1: Enable

21.5.4 TIM status register (TIMx_SR)

Address offset: 0x10

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.															UIF
															rc_w0

Bits	Name	Description
31:1	Reserved	Must be kept at reset value
0	UIF	Update interrupt flag This bit is set by hardware on an update event. 0: No update occurred 1: Update interrupt pending: – At overflow regarding the repetition counter value and if UDIS = 0 in the TIMx_CR1 register. – Software using the UG bit in the TIMx_EVTG register, if URS = 0 and UDIS = 0 in the TIMx_CR1 register.

21.5.5 TIM event generation register (TIMx_EVTG)

Address offset: 0x14

Reset value: 0x0000 0000

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

Bits	Name	Description
31:1	Reserved	Must be kept at reset value
0	UG	Update generation, it is automatically cleared after one TIMx_KCLK clock by hardware. 0: No action 1: Re-initialize the counter and generates an update of the registers. The prescaler counter is also cleared (anyway the prescaler ratio is not affected).

21.5.6 TIM counter register (TIMx_CNT)

Address offset: 0x24

Reset value: 0x0000 0000

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

Bits	Name	Description
31:16	Reserved	Must be kept at reset value
15:0	CNT[15:0]	Counter value

21.5.7 TIM prescaler register (TIMx_PSC)

Address offset: 0x28

Reset value: 0x0000 0000

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

Bits	Name	Description
31:16	Reserved	Must be kept at reset value
15:0	PSC[15:0]	Prescaler value The counter clock frequency CK_CNT is equal to f_{CK_PSC} / (PSC[15:0] + 1). PSC contains the value to be loaded in the active prescaler register at each update event.

21.5.8 TIM auto-reload register (TIMx_ARR)

Address offset: 0x2C

Reset value: 0x0000 FFFF

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

Bits	Name	Description
31:16	Reserved	Must be kept at reset value
15:0	ARR[15:0]	Auto-reload value The counter is blocked while the auto-reload value is null.

22 Low-power timer (LPTIM1/2)

22.1 Introduction

The LPTIMx is a 16-bit timer that can select PCLK1, LXTAL, RCH or RCL as its clock source. It can keep running in Stop mode and has a timed wake-up function.

22.2 LPTIM1/2 main features

- 16 bit upcounter
- 3-bit prescaler for the counter prescaler
- Up to 2 input channels and 1 output channel for:
  - PWM mode
  - One-pulse mode
- Selectable clock:
  - Internal clock sources: PCLK1, LXTAL, RCH or RCL
  - External clock source: LPTIMx_IN1 (LPTIM1 supports only)
- Programmable digital glitch filter
- Encoder mode
- Interrupts and timeout events can wake up from the low-power mode

22.3 LPTIMx functional description

22.3.1 LPTIMx block diagram

Figure 22-1 LPTIM1 block diagram

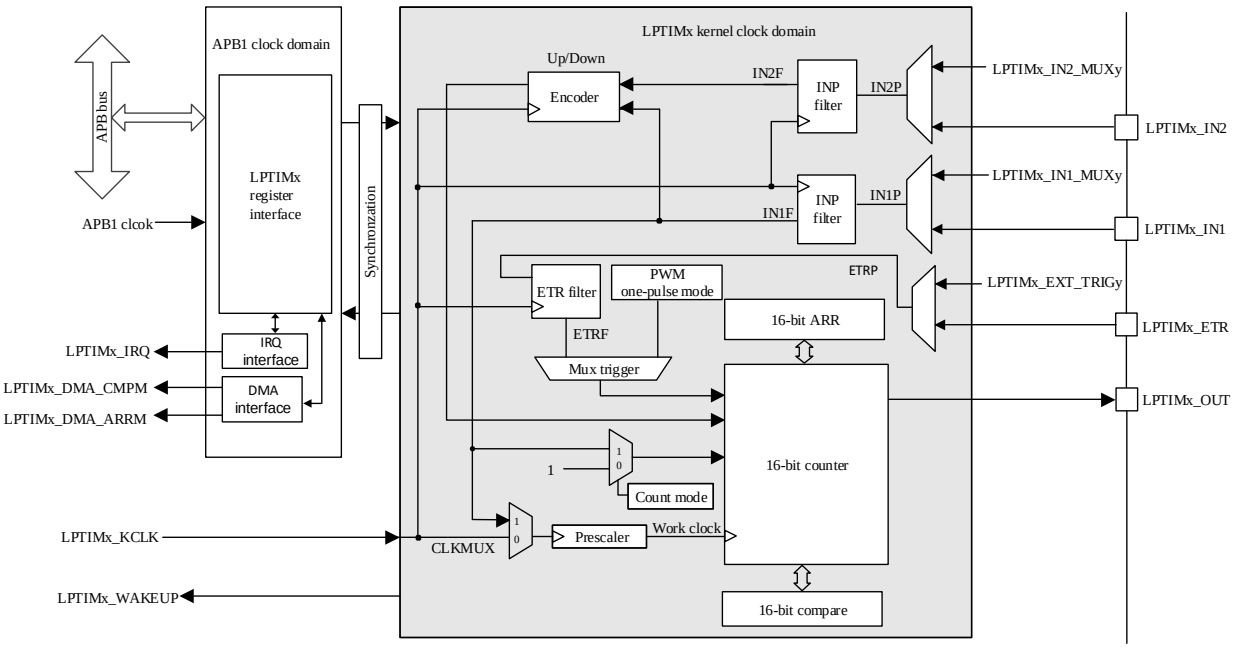
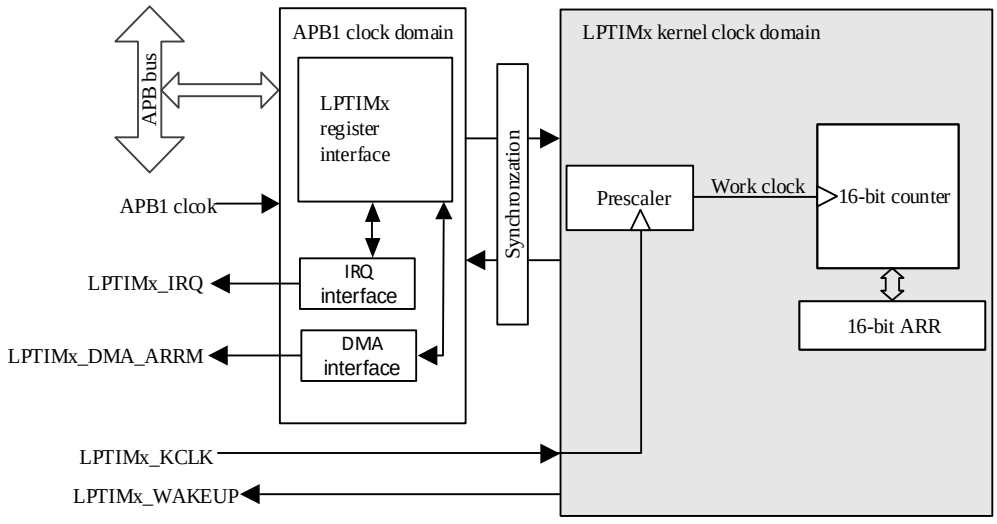


Figure 22-2 LPTIM2 block diagram



LPTIM2 has only 16-bit counter, prescaler and internal clock source.

22.3.2 LPTIM pins and internal signals

Table 22-1 LPTIMx input/output pins

Pin name	Signal type	Description
LPTIMx_IN1	Input	LPTIMx input 1 from GPIO pin on mux input 1

Pin name	Signal type	Description
LPTIMx_IN2	Input	LPTIMx input 2 from GPIO pin on mux input 2
LPTIMx_ETR	Input	LPTIMx external trigger GPIO pin
LPTIMx_OUT	Output	LPTIMx output GPIO pin

Table 22-2 LPTIMx internal signals

Signal name	Signal type	Description
LPTIMx_WAKEUP	Output	LPTIMx wakeup event
LPTIMx_IRQ	Output	LPTIMx global interrupt
LPTIMx_DMA_CMPM	Output	Compare match DMA request signal
LPTIMx_DMA_ARRM	Output	Auto-reload match DMA request signal
LPTIMx_IN1_MUXy	Input	Internal LPTIMx input 1 connected to mux input y, y = 0~1
LPTIMx_IN2_MUXy	Input	Internal LPTIMx input 2 connected to mux input y, y = 0~1
LPTIMx_EXT_TRIGy	Input	LPTIMx external trigger signal y, y = 0~7

22.3.3 LPTIM input and trigger mapping

Table 22-3 LPTIMx_IN1 input connection

LP_IN1_MUX	LPTIM1 input 1 connected to
LPTIMx_IN1_MUX0	GPIO pin as LPTIM1_IN1 alternate function
LPTIMx_IN1_MUX1	COMP1_OUT

Table 22-4 LPTIMx_IN2 input connection

LP_IN2_MUX	LPTIM1 input 2 connected to
LPTIMx_IN2_MUX0	GPIO pin as LPTIM1_IN2 alternate function
LPTIMx_IN2_MUX1	COMP2_OUT

Table 22-5 LPTIMx external trigger connection

TRIGSEL	External trigger
LPTIMx_EXT_TRIG0	LPTIMx_ETR GPIO pin
LPTIMx_EXT_TRIG1	RTC_ALARM_TRIG
LPTIMx_EXT_TRIG6	COMP1_OUT
LPTIMx_EXT_TRIG7	COMP2_OUT

22.3.4 Clock source and counter mode

The clock for LPTIMx can be provided by the following sources:

- Internal clock sources: LPTIMx_KCLK can be configured through the *Peripherals independent clock configuration register (RCC_CLKSEL)*, using the

LPTIMx_SEL[1:0] bit field to select LXTAL, PCLK1, RCH, or RCL as the LPTIMx clock source.

- External clock source: Clock source provided by the LPTIMx_IN1 input pin.

The LPTIMx counter can be clocked using an internal clock signal or an external clock signal injected on LPTIMx_IN1. Programming the CKSEL and COUNTMODE bits in the LPTIMx_CFG register allows controlling whether the LPTIMx will use an external clock source or an internal one.

Internal clock source

LPTIMx is clocked by an internal clock source: Set the CKSEL bit in the LPTIMx_CFG register to 0.

- COUNT_MODE = 0

  The LPTIMx is configured to be clocked by an internal clock source and the LPTIMx counter is configured to be updated following each internal clock pulse.

- COUNT_MODE = 1

  Count the external clock signal on the LPTIMx_IN1 pin. The counter can count on the rising edge, falling edge, or both edges of the external clock signal. The valid edges of the external clock signal are selected by the CKPOL[1:0] bits in the LPTIMx_CFG register.

  When counting on the rising or falling edge, the external clock signal frequency must be less than the internal clock frequency. When counting on both the rising and falling edges, the internal clock frequency should be at least four times the external clock signal frequency.

External clock source

LPTIMx is clocked by an external clock source: Set the CKSEL bit in the LPTIMx_CFG register to 1. The COUNT_MODE value is irrelevant in this configuration. This setup allows for timeout functionality or pulse counting in low-power modes.

For this configuration, the LPTIMx counter can be updated either on rising edges or falling edges of the LPTIMx_IN1 but not on both rising and falling edges.

Since the signal injected on the LPTIMx_IN1 is also used to clock the LPTIMx kernel logic, there is some initial latency (after the LPTIMx is enabled) before the counter is incremented. More precisely, the first four active edges on the LPTIMx_IN1 (after

LPTIMx is enable) are lost.

22.3.5 Glitch filter

LPTIMx supports digital filtering of input signals, with the filter clock provided by LPTIMx_KCLK.

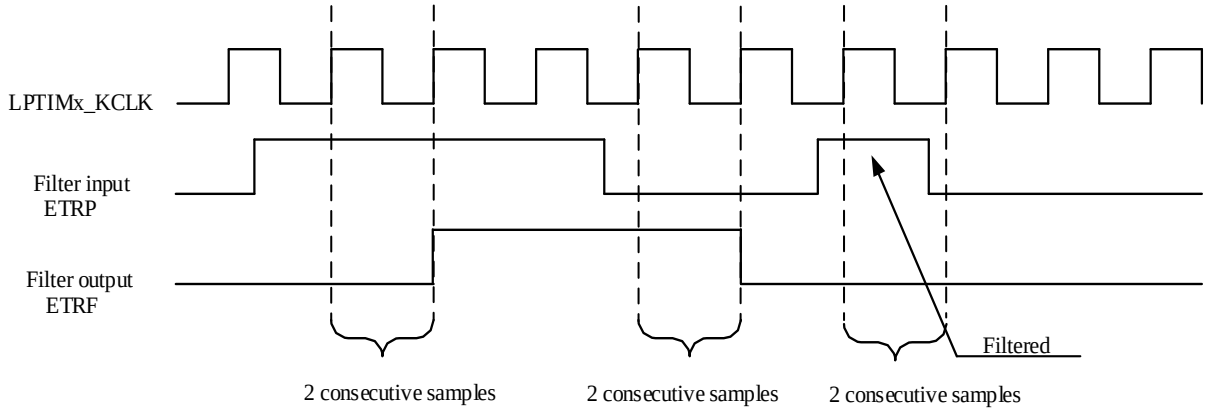
The digital filters are divided into two groups:

- The digital filters of IN1P, IN2P: The digital filters sensitivity is controlled by the CKFLT[1:0] bits. The configuration parameter CKFLT[1:0] bits apply to both IN1P and IN2P signals.
- The digital filters of ETRP: The digital filters sensitivity is controlled by the TRGFLT[1:0] bits.

When the LPTIMx is clocked by an external clock source, the CKFLT[1:0] bits must be set to 0 to disable the digital filter.

An example of glitch filter behavior in case of a 2 consecutive samples programmed.

Figure 22-3 Glitch filter timing diagram



22.3.6 Trigger multiplexer

The LPTIMx counter can be started by software trigger or by a valid edge on one of the eight trigger inputs. TRIG_EN[1:0] bits are used to determine the LPTIMx trigger source:

- Software trigger: When TRIG_EN[1:0] bits are equals ‘00’, the LPTIMx counter is started as soon as one of the CNTSTRT or SNGSTRT bit is set by software. Counting starts after two APB1 clocks and two LPTIMx_KCLK clocks.
- External trigger: When TRIG_EN[1:0] bits are different than ‘00’, TRIGSEL[2:0] bits are used to select which of the trigger inputs is used to start the counter. see

Table: LPTIMx external trigger connection. LPTIMx starts counting after detecting a valid edge on the external trigger input. The TRIG_EN[1:0] bits are used to select the valid edge of the external trigger input.

The external triggers are considered asynchronous signals for the LPTIMx. So after a trigger detection, a two-counter-clock period latency is needed before the timer starts running due to the synchronization. If a new trigger event occurs when the timer is already started it will be ignored (unless timeout function is enabled).

The timer must be enabled before setting the SNGSTRT/CNTSTRT bits. Any write on these bits when the timer is disabled will be discarded by hardware.

Before the external trigger takes effect, the CNTSTRT or SNGSTRT bit must be configured. Otherwise, detecting a valid edge on the external trigger input will start the PWM mode counting.

22.3.7 Output compare modes

Output compare modes is divided into:

- PWM mode: the timer is free running, starting from a trigger event and stopping only when disabled.
- One-shot mode: the timer is started from a trigger event and stops when reaching the ARR value.

In both modes, the counter can be started by an external trigger or by software.

SNGSTRT and CNTSTRT bits can only be set when the timer is enabled (The ENABLE bit is set to ‘1’). It is possible to change “on the fly” from One-shot mode to PWM mode.

If both the CNTSTRT and SNGSTRT bits are set to 1 (both PWM mode and single pulse mode are active), the timer operates in PWM mode.

The output waveform polarity is configured by the WAVPOL bit in the LPTIMx_CFG register.

- WAVPOL = 0: When LPTIMx_CNT < LPTIMx_CMP, the output is low, otherwise, the output is high.
- WAVPOL = 1: When LPTIMx_CNT < LPTIMx_CMP, the output is high, otherwise, the output is low.

PWM mode

To enable PWM mode counting, the CNTSTRT bit in the LPTIMx_CR register must be set to 1. Once set, the CNTSTRT bit is immediately automatically cleared by hardware. The CNTSTRT bit can be set to 1 only after the ENABLE bit in the LPTIMx_CR register has been set to 1.

If counting is started via software-triggered PWM mode (when the TRIG_EN[1:0] bit field in the LPTIMx_CFG register is configured as 00), setting the CNTSTRT bit will start the counter in PWM mode.

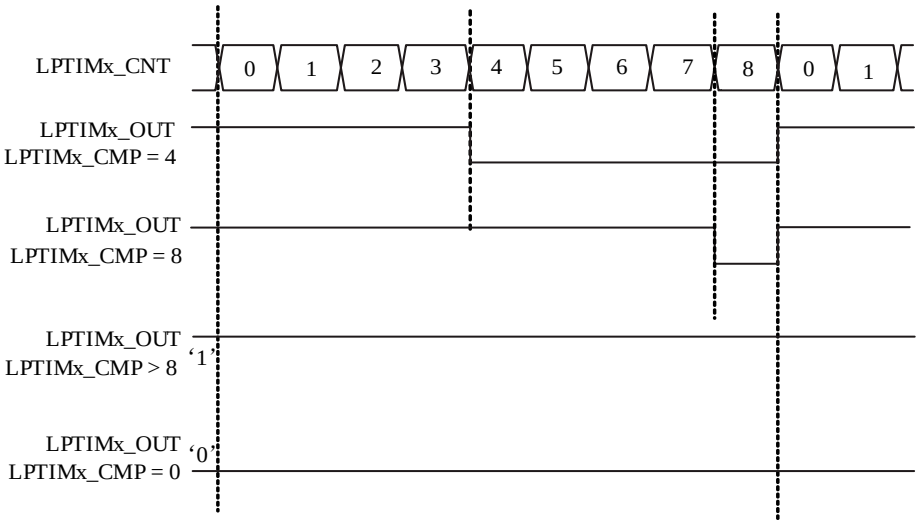
- When LPTIMx_CNT < LPTIMx_CMP, the output is inactive, otherwise, the output is active.
- When LPTIMx_CNT < LPTIMx_CMP, the output is active, otherwise, the output is inactive.

If the value of the LPTIMx_CNT register matches the value of the LPTIMx_CMP register, the CMPM bit in the LPTIMx_ISR register is set to 1. The CMPM bit is cleared by setting the CMPM_CF bit in the LPTIMx_ICR register to 1.

If the value of the LPTIMx_CNT register matches the value of the LPTIMx_ARR register, the ARRM bit in the LPTIMx_ISR register is set to 1. The ARRM bit is cleared by setting the ARRM_CF bit in the LPTIMx_ICR register to 1.

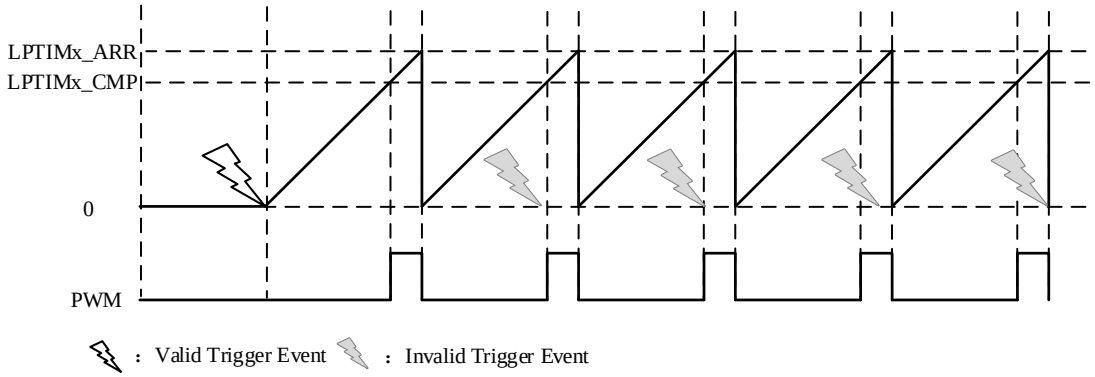
Output waveform in PWM mode:

Figure 22-4 Waveform in PWM mode (LPTIMx_ARR = 8, WAVPOL bit = 1)



If the counting is started by an external trigger (TRIG_EN[1:0] bits in the LPTIMx_CFG register is not 00), after setting CNTSTRT to 1, the counter will start counting in PWM mode when a valid external trigger event occurs. Any subsequent external trigger events will be discarded, as shown in the following figure.

Figure 22-5 Waveform in PWM mode



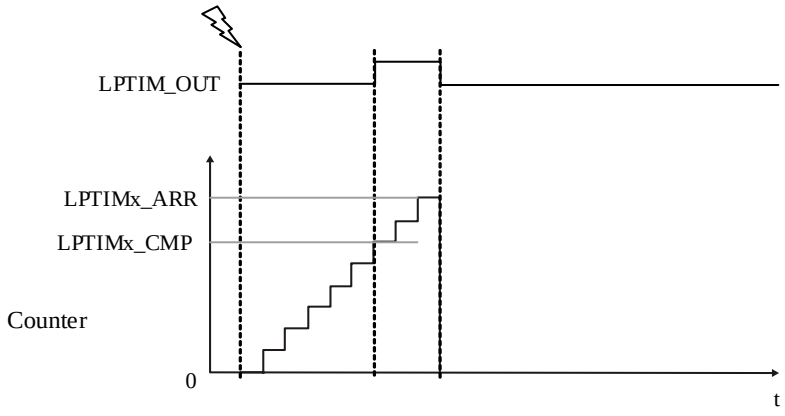
One-shot mode

To enable one-shot mode counting, the SNGSTRT bit in the LPTIMx_CR register must be set to 1. After SNGSTRT is set to 1, it is automatically cleared to 0 by hardware. The SNGSTRT bit can be set to 1 only after the ENABLE bit in the LPTIMx_CR register is set to 1.

If the counting is started by software trigger (the TRIG_EN[1:0] bits in the LPTIMx_CFG register is 00), setting SNGSTRT to 1 will start the counter in one-shot mode.

Waveform in one-shot mode:

Figure 22-6 Waveform in one-shot mode (WAVPOL = 1)

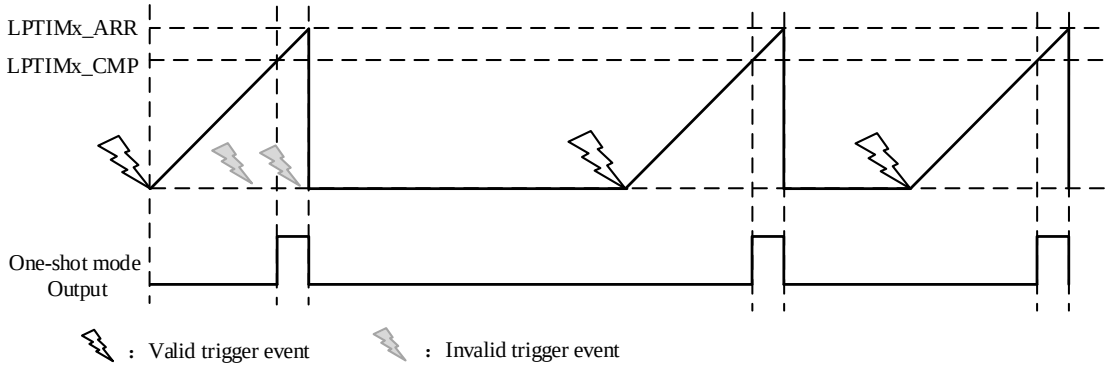


The output waveform in one-shot mode requires that the value in the LPTIMx_ARR register must be greater than the value in the LPTIMx_CMP register. If the LPTIMx_CMP compare register is set to a value greater than the LPTIMx_ARR register, the LPTIMx_CNT register will be reset to 0 when the count reaches the LPTIMx_ARR value.

If the counting is started by an external trigger (the TRIG_EN[1:0] bits in the LPTIMx_CFG register is not 00), after setting SNGSTRT to 1, the counter will start

counting to the ARR value when a valid external trigger event occurs. Any trigger events that occur during this period will be discarded, as shown in the following figure.

Figure 22-7 Waveform in one-shot mode



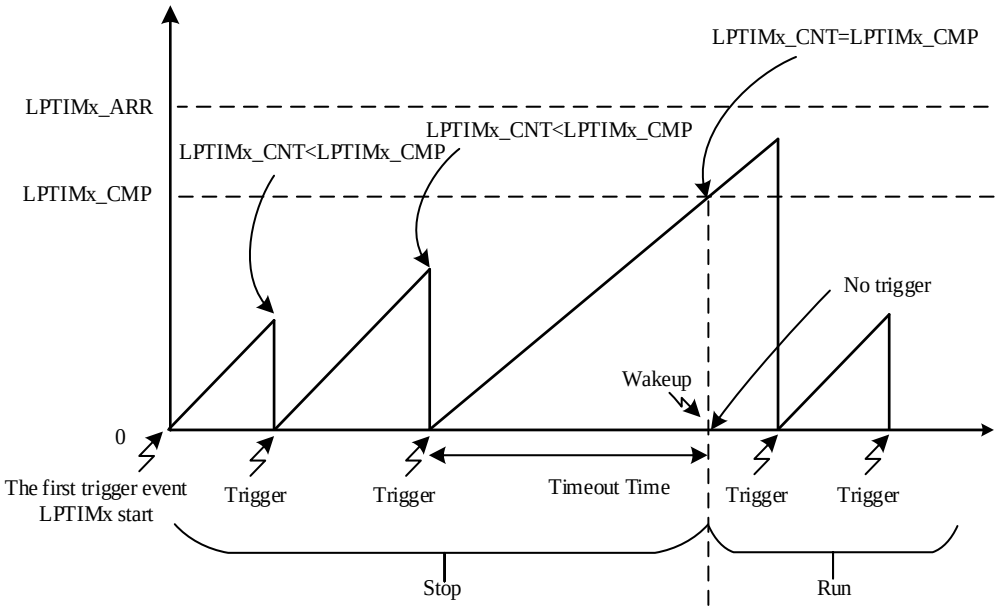
22.3.8 Timeout function

The TIMOUT bit in the LPTIMx_CFG register is set to 1, the timeout function is enabled. When the TIMOUT bit is set to 1, the first trigger event will start the timer, any successive trigger event will reset the counter and the timer will restart.

If no trigger event occurs within the wait time configured by LPTIMx_CMP, a compare match event will occur. This feature is supported only by LPTIM1.

The timeout timing sequence is shown in the following figure.

Figure 22-8 Timeout timming



22.3.9 LPTIM enable

Configure the ENABLE bit in the LPTIMx_CR register to enable or disable the LPTIMx.

- When the ENABLE bit is set to 1, the enable takes effect after two LPTIMx counter clock cycles. If ENABLE is set to 1 and then external triggering or software triggering (CNTSTRT bit or SNGSTRT bit set to 1) is configured, the software does not need to wait for the enable to take effect; counting starts automatically after two LPTIMx counter clock cycles.
- Clearing the ENABLE bit to 0 will reset the LPTIMx_CNT register after three LPTIMx_KCLK clock cycles. During this period, any attempt to start counting will be ignored.

The LPTIMx_CFG register must be modified only when the LPTIMx is disabled.

22.3.10 Encoder mode

Only LPTIM1 supports encoder mode. Setting the ENC bit in the LPTIMx_CFG register to 1 enables the quadrature encoder mode. Setting the CNTSTRT bit in the LPTIMx_CR register to 1 starts the encoder counting.

Encoder mode has three counting methods:

- Encoder mode 1: When the CKPOL[1:0] bits is 00, the counter counts on rising edges.
- Encoder mode 2: When the CKPOL[1:0] bits is 01, the counter counts on falling edges.
- Encoder mode 3: When the CKPOL[1:0] bits is 10, the counter counts on both rising and falling edges.

When LPTIMx is clocked by an internal clock source, the quadrature encoder mode can be used. This mode is equivalent to an external clock counter with direction selection. Therefore, the LPTIMx_ARR must be configured before starting. The clock signal for the counter is generated by two external input signals, LPTIMx_IN1 and LPTIMx_IN2, and the phase difference between these two signals determines the counting direction.

Changes in the counting direction are indicated by the DOWN and UP bits in the LPTIMx_ISR register. If the counting direction changes from incrementing to decrementing, the DOWN bit is set to 1. If the counting direction changes from

decrementing to incrementing, the UP bit is set to 1. The DOWN and UP bits can be cleared by setting the DOWN_CF and UP_CF bits in the LPTIMx_ICR register to 1. If the DOWN_IE or UP_IE bit in the LPTIMx_IER is set to 1 (interrupt enabled), a change in direction will generate an interrupt. The counter clock is provided by the rising edge, falling edge, or both edges of the LPTIMx_IN1 or LPTIMx_IN2 signals. The following table shows the relationship between the counting direction and the encoder signals.

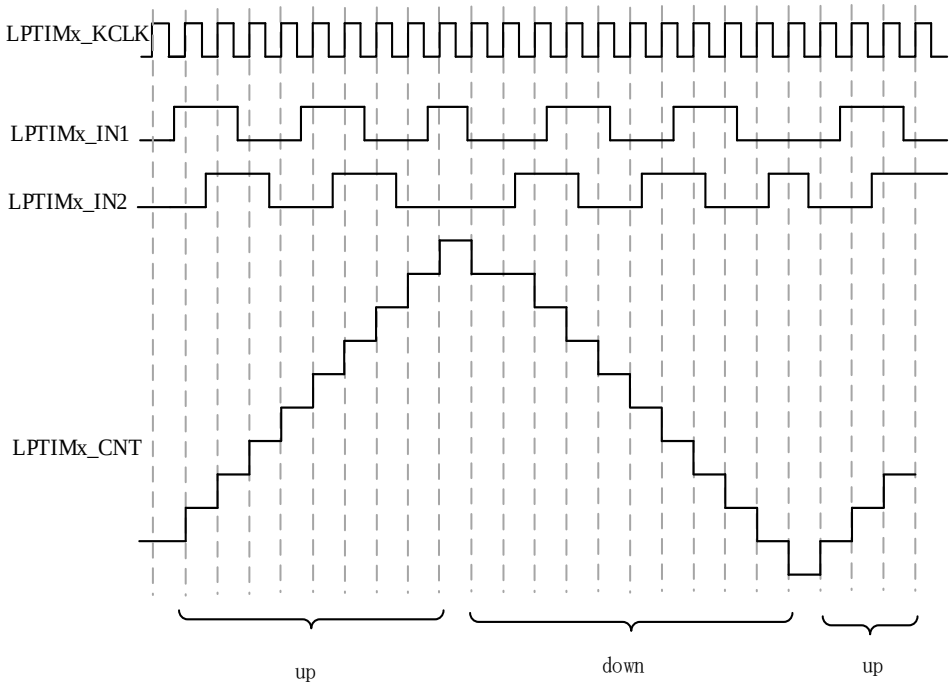
The signal frequency on the LPTIMx_IN1 and LPTIMx_IN2 inputs must not exceed one-fourth of the LPTIMx_KCLK clock frequency.

Table 22-6 Encoder counting scenarios

Active edge	**Level on opposite signal (LPTIMx_IN1 for LPTIMx_IN2,LPTIMx_IN2 for LPTIMx_IN1)**	**LPTIMx_IN1 signal**		**LPTIMx_IN2 signal**	
		Rising	**Falling**	**Rising**	**Falling**
Rising edge	High	Down	No count	Up	No count
	Low	Up	No count	Down	No count
Falling edge	High	No count	Up	No count	Down
	Low	No count	Down	No count	Up
Both edges	High	Down	Up	Up	Down
	Low	Up	Down	Down	Up

The following figure shows a counting sequence for encoder mode when both-edge sensitivity is configured.

Figure 22-9 Encoder mode counting sequence



22.3.11 Debug mode

When the microcontroller enters debug mode (Cortex-M0+ core halted), the counter either continues to work normally or stops, depending on LPTIMx_HOLD configuration bit in the *APB1 freeze register (DBG_APB1_FZ)*.

22.4 LPTIM low-power modes

Table 22-7 Effect of low-power modes on the LPTIMx

Mode	Description
Sleep	No effect. LPTIMx interrupts cause the device to exit Sleep mode.
Stop	When the clock driving LPTIMx is LXTAL, RCL internal clock, or an external input clock source, there is no effect. LPTIMx interrupts and compare match events in LPTIM1 timeout mode can wake up the device from Stop mode.

22.5 LPTIM interrupts

The following events generate an interrupt/wake-up event, if they are enabled through the LPTIMx_IER register:

Table 22-8 Interrupt requests

Interrupt event	Event flag	Enable control bit	Interrupt clear method
Compare match	CMPM	CMPM_IE	CMPM cleared by software by setting CMPM_CF bit
Auto-reload match	ARRM	ARRM_IE	ARRM cleared by software by setting ARRM_CF bit
External trigger event	EXTTRIG	EXTTRIG_IE	EXTTRIG cleared by software by setting EXTTRIG_CF bit
Counter direction change	DOWN, UP	DOWN_IE, UP_IE	DOWN cleared by software by setting DOWN_CF bit UP cleared by software by setting UP_CF bit

22.6 LPTIM registers

The LPTIM registers can only be accessed by words (32-bit).

Table 22-9 LPTIMx base address (x = 1, 2)

Peripheral	Base address
LPTIM1	0x4000 7C00
LPTIM2	0x4000 9000

22.6.1 LPTIM interrupt and status register (LPTIMx_ISR)

Address offset: 0x00

Reset value: 0x0000 0000

Note: LPTIM2 has only ARRM bit. Other bits are reserved.

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.									DOWN	UP	Res.		EXTTRIG	ARRM	CMPM
									rw	rw			rw	rw	rw

Bits	Name	Description
31:7	Reserved	Must be kept at reset value
6	DOWN	Counter direction change from up to down In encoder mode, DOWN bit is set by hardware to indicate that the counter direction has changed from up to down. 0: No change 1: Counter direction change up to down
5	UP	Counter direction change from down to up In encoder mode, UP bit is set by hardware to indicate that the counter direction has changed from down to up. 0: No change 1: Counter direction change down to up
4:3	Reserved	Must be kept at reset value

2	EXTTRIG	External trigger event EXTTRIG is set by hardware to indicate that a valid edge on the selected external trigger input has occurred. 0: No external trigger event occurred 1: External trigger event occurred
1	ARRM	Auto-reload match ARRM is set by hardware to indicate that the LPTIMx_CNT register value reached the LPTIMx_ARR register value. 0: No auto-reload match event occurred 1: Auto-reload match event occurred
0	CMPM	Compare match CMPM bit is set by hardware to indicate that LPTIMx_CNT register value reached the LPTIMx_CMP register value. 0: No comparison match event occurred 1: A comparison match event occurred

22.6.2 LPTIM interrupt clear register (LPTIMx_ICR)

Address offset: 0x04

Reset value: 0x0000 0000

Note: LPTIM2 has only ARRM_CF bit. Other bits are reserved.

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.									DOWN_CF	UP_CF	Res.		EXTTRIG_CF	ARRM_CF	CMPM_CF
									w	w			w	w	w

Bits	Name	Description
31:7	Reserved	Must be kept at reset value
6	DOWN_CF	Direction change to down clear flag Writing 1 to this bit clear the DOWN flag in the LPTTIMx_ISR register.

5	UP_CF	Direction change to up clear flag Writing 1 to this bit clear the UP flag in the LPTIMx_ISR register.
4:3	Reserved	Must be kept at reset value
2	EXTTRIG_CF	External trigger valid edge clear flag Writing 1 to this bit clear the EXTTRIG flag in the LPTIMx_ISR register.
1	ARRM_CF	Auto-reload match clear flag Writing 1 to this bit clear the ARRM flag in the LPTIMx_ISR register.
0	CMPM_CF	Compare match clear flag Writing 1 to this bit clear the CMPM flag in the LPTIMx_ISR register.

22.6.3 LPTIM interrupt enable register (LPTIMx_IER)

Address offset: 0x08

Reset value: 0x0000 0000

Note: LPTIM2 has only ARRM_IE and ARR_DMAEN bits. Other bits are reserved.

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		ARR_DMAEN	CMP_DMAEN	Res.					DOWN_IE	UP_IE	Res.		EXTTRIG_IE	ARRM_IE	CMPM_IE
		rw	rw						rw	rw			rw	rw	rw

Bits	Name	Description
31:14	Reserved	Must be kept at reset value
13	ARR_DMAEN	Auto-reload match DMA request enable 0: Disable 1: Enable

12	CMP_DMAEN	Compare match DMA request enable 0: Disable 1: Enable
11:7	Reserved	Must be kept at reset value
6	DOWN_IE	Direction change to down interrupt enable 0: Disable 1: Enable
5	UP_IE	Direction change to up interrupt enable 0: Disable 1: Enable
4:3	Reserved	Must be kept at reset value
2	EXTTRIG_IE	External trigger valid edge interrupt enable 0: Disable 1: Enable
1	ARRM_IE	Auto-reload match interrupt enable 0: Disable 1: Enable
0	CMPM_IE	Compare match interrupt enable 0: Disable 1: Enable

22.6.4 LPTIM configuration register (LPTIMx_CFG)

Address offset: 0x0C

Reset value: 0x0000 0000

Note: LPTIM2 has only PRESC bits. Other bits are reserved.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.							ENC	COUNT_MODE	Res.	WAVPOL	Res.	TIMEOUT	TRIG_EN[1:0]		Res.
							rw	rw		rw		rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRIG_SEL[2:0]			Res.	PRESC[2:0]			Res.	TRIG_FLT[1:0]		Res.	CKFLT[1:0]		CKPOL[1:0]		CKSEL
rw	rw	rw		rw	rw	rw		rw	rw		rw	rw	rw	rw	rw

Bits	Name	Description
31:25	Reserved	Must be kept at reset value
24	ENC	Encoder mode enable 0: Disable 1: Enable
23	COUNT_MODE	Counter mode enabled 0: The counter is incremented following each internal clock pulse 1: The counter is incremented following each valid clock pulse on the LPTIMx_IN1
22	Reserved	Must be kept at reset value
21	WAVPOL	Waveform shape polarity 0: The LPTIMx output reflects the compare results between LPTIMx_ARR and LPTIMx_CMP registers 1: The LPTIMx output reflects the inverse of the compare results between LPTIMx_ARR and LPTIMx_CMP registers
20	Reserved	Must be kept at reset value
19	TIMOUT	Timeout enable 0: A trigger event arriving when the timer is already started will be ignored 1: A trigger event arriving when the timer is already started will reset and restart the counter
18:17	TRIG_EN[1:0]	Trigger enable and polarity 00: Software trigger (counting start is initiated by software) 01: Rising edge is the active edge 10: Falling edge is the active edge 11: Both edges are active edges
16	Reserved	Must be kept at reset value
15:13	TRIG_SEL[2:0]	Trigger selector 000: LPTIMx_EXT_TRIG0, LPTIMx_ETR GPIO pin 001: LPTIMx_EXT_TRIG1, RTC_ALARM_TRIG 110: LPTIMx_EXT_TRIG6, COMP1_OUT 111: LPTIMx_EXT_TRIG7, COMP2_OUT Other: Reserved (invalid configuration)

12	Reserved	Must be kept at reset value
11:9	PRESC[2:0]	Clock prescaler 000: 1 001: 2 010: 4 011: 8 100: 16 101: 32 110: 64 111: 128
8	Reserved	Must be kept at reset value
7:6	TRIG_FLT[1:0]	Configurable digital filter for trigger The TRIG_FLT value sets the number of consecutive equal samples that should be detected when a level change occurs on an internal trigger before it is considered as a valid level transition. 00: Any trigger active level change is considered as a valid trigger 01: N = 2 10: N = 4 11: N = 8
5	Reserved	Must be kept at reset value
4:3	CKFLT[1:0]	Configurable digital filter for external clock The CKFLT value sets the number of consecutive equal samples that should be detected when a level change occurs on an external clock signal before it is considered as a valid level transition. 00: Any external clock signal level change is considered as a valid transition 01: N = 2 10: N = 4 11: N = 8
2:1	CKPOL[1:0]	Clock polarity If LPTIMx is clocked by an external clock source: 00: The rising edge is the active edge. 01: The falling edge is the active edge. 10: Both edges are active edges, LPTIMx must select an internal clock source, and the internal clock source frequency must be at least four times the external clock frequency.

		11: The rising edge is the active edge If the LPTIMx is configured in encoder mode: 00: The encoder sub-mode 1 is active, the rising edge is the active edge. 01: The encoder sub-mode 2 is active, the falling edge is the active edge. 10: The encoder sub-mode 3 is active, both edges are active edges. 11: The encoder sub-mode 1 is active, the rising edge is the active edge.
0	CKSEL	Clock selector 0: LPTIMx is clocked by internal clock source 1: LPTIMx is clocked by an external clock source through the LPTIMx_IN1

22.6.5 LPTIM control register (LPTIMx_CR)

Address offset: 0x10

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.													CNTSTRT	SNGSTRT	ENABLE
													rw	rw	rw

Bits	Name	Description
31:3	Reserved	Must be kept at reset value
2	CNTSTRT	PWM mode enable This bit is set by software and cleared by hardware.
1	SNGSTRT	On-pulse mode enable This bit is set by software and cleared by hardware.
0	ENABLE	LPTIMx enable 0: Disable 1: Enable

22.6.6 LPTIM compare register (LPTIMx_CMP)

Address offset: 0x14

Reset value: 0x0000 0000

Note: LPTIM1 supports only.

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

Bits	Name	Description
31:16	Reserved	Must be kept at reset value
15:0	CMP[15:0]	Compare value The LPTIMx_CMP register must only be modified when the LPTIMx is enabled (ENABLE bit set to ‘1’).

22.6.7 LPTIM auto reload register (LPTIMx_ARR)

Address offset: 0x18

Reset value: 0x0000 0001

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

Bits	Name	Description
31:16	Reserved	Must be kept at reset value
15:0	ARR[15:0]	Auto reload value The LPTIMx_ARR register must only be modified when the LPTIMx is enabled (ENABLE bit set to ‘1’).

22.6.8 LPTIM counter register (LPTIMx_CNT)

Address offset: 0x1C

Reset value: 0x0000 0000

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

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

Bits	Name	Description
31:16	Reserved	Must be kept at reset value
15:0	CNT[15:0]	Counter value

22.6.9 LPTIM configuration register 2 (LPTIMx_CFG2)

Address offset: 0x20

Reset value: 0x0000 0000

Note: LPTIM1 supports only.

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.										IN2_SEL[1:0]		Res.		IN1_SEL[1:0]	
										rw	rw			rw	rw

Bits	Name	Description
31:6	Reserved	Must be kept at reset value
5:4	IN2_SEL[1:0]	LPTIMx_IN2 source selection 00: LPTIMx_IN2_MUX0, LPTIMx_IN2 GPIO pin 01: LPTIMx_IN2_MUX1, COMP2_OUT Other: Reserved (LPTIMx_IN2 GPIO pin)
3:2	Reserved	Must be kept at reset value
1:0	IN1_SEL[1:0]	LPTIMx_IN1 source selection 00: LPTIMx_IN1_MUX0, LPTIMx_IN1 GPIO pin

01: LPTIMx_IN1_MUX1, COMP1_OUT

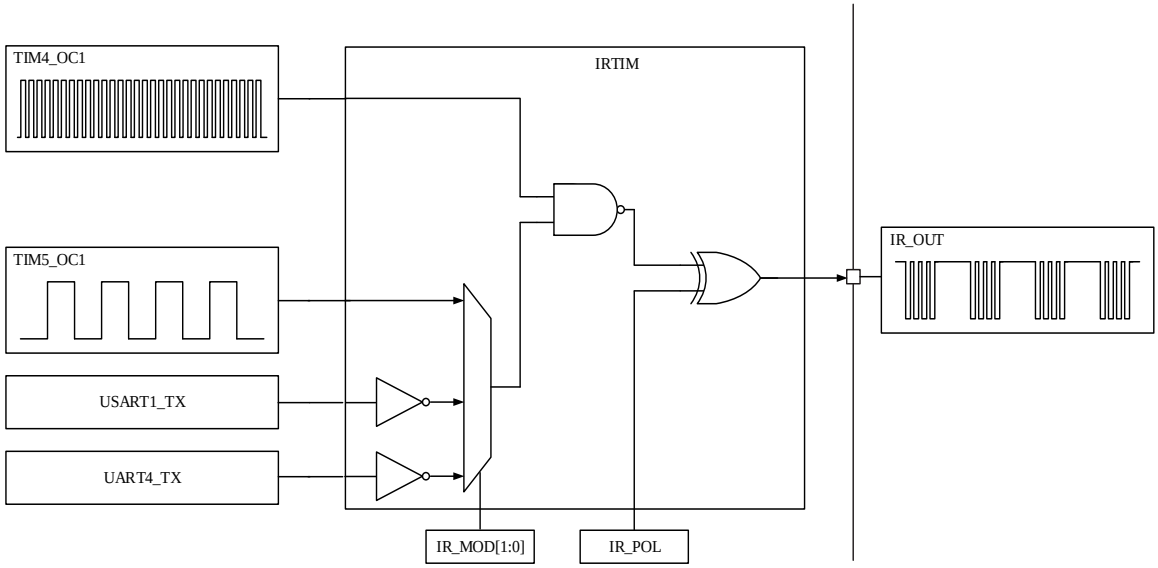
Other: Reserved (LPTIMx_IN1 GPIO pin)

23 Infrared interface (IRTIM)

An infrared interface (IRTIM) for remote control is available on the device. It can be used with an infrared LED to perform remote control functions.

It uses internal connections with USART1, USART4, TIM4 and TIM5 as shown in the following figure.

Figure 23-1 IRTIM internal hardware connections



By configuring different carriers and modulated signals, infrared control signals can be generated according to any infrared pulse modulation standard.

TIM4 (TIM4_OC1) is used to generate the high frequency carrier signal, while TIM5 (TIM5_OC1) or alternatively USART1 or USART4 generates the modulation envelope according to the setting of the IR_MODE[1:0] bits in the *SYSCFG configruration register (SYSCFG_CR)*.

The polarity of the output signal from IRTIM is controlled by the IR_POL bit in the *SYSCFG configruration register (SYSCFG_CR)* and could be inverted by setting of this bit.

The infrared function is output on the IR_OUT pin. The activation of this function is done through the GPIO register by enabling the related alternate function bit.

The high sink LED driver capability (PD2 pin) can be activated through *GPIO drive capability configuration register (GPIOx_HDCFG)* and used to sink the high current needed to directly control an infrared LED.

24 Independent watchdog (IWDG)

24.1 Introduction

The devices feature an embedded watchdog peripheral. The independent watchdog peripheral can detect and solve malfunctions, and triggers system reset when the counter reaches a given timeout value.

The independent watchdog (IWDG) is clocked by its own dedicated low-speed RC oscillator (RCL) and thus stays active even if the system clock fails, thereby enhancing system reliability.

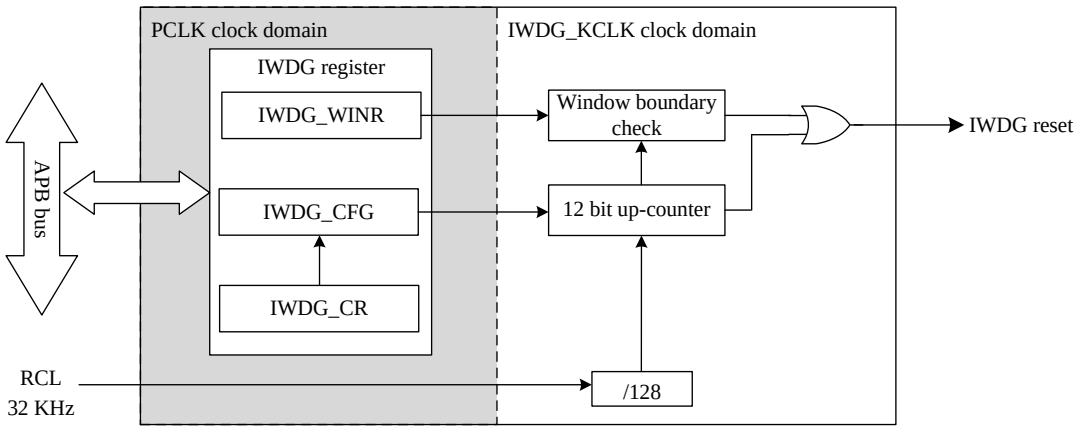
24.2 IWDG main features

- 12-bit up-counter
- Clocked by the independent RC oscillator (RCL), can operate in Stop mode
- Conditional reset:
  - Reset (if watchdog activated) when the up-counter overflows to its maximum value
  - Reset (if watchdog activated) if the up-counter is reloaded outside the window

24.3 IWDG functional description

24.3.1 Block diagram

Figure 24-1 IWDG block diagram



The IWDG remains functional in Stop mode.

When the independent watchdog is started by writing the value 0x0000 CCCC in the IWDG control register (IWDG_CR), the counter starts counting up from 0. When it reaches the overflow time a reset signal is generated (IWDG reset).

Whenever the key value 0x0000 AAAA is written in the IWDG control register (IWDG_CR), the count value is cleared and the watchdog reset is prevented.

The IWDG clock source (RCL) remains enabled and cannot be disabled once the watchdog is active. Enabling the IWDG automatically activates the RCL clock source.

24.3.2 Register access protection

Write access to IWDG configure register (IWDG_CFG) and IWDG window register (IWDG_WINR) is protected. To modify these registers, the user must first write the code 0x0000 5555 in the IWDG control register (IWDG_CR). A write access to this register with a different value breaks the sequence and register access is protected again.

24.3.3 Window function

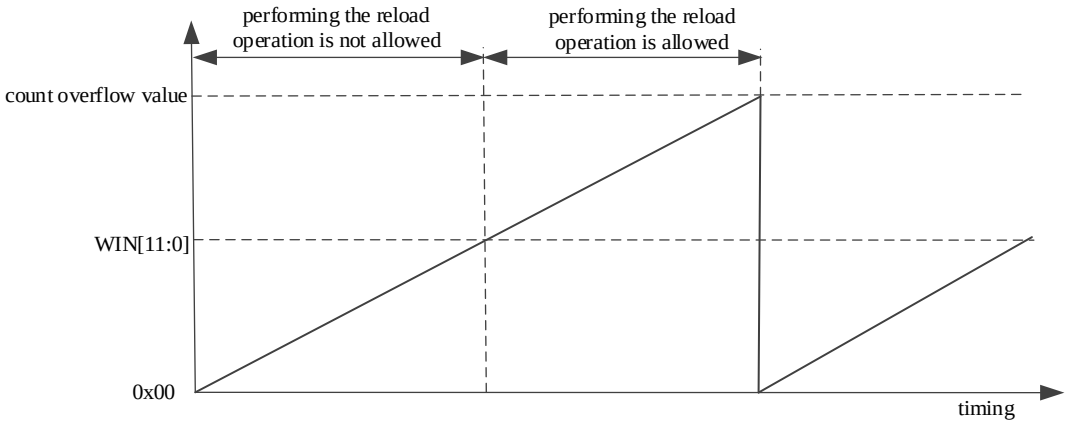
The IWDG can also work as a window watchdog by setting the appropriate window in the IWDG window register (IWDG_WINR).

The watchdog reload operation is considered valid only when the count value exceeds the value programmed in the IWDG window register (IWDG_WINR). Otherwise, it will trigger the IWDG reset. Upon reset, the IWDG window register (IWDG_WINR) defaults to 0x00 and the window option is disabled.

The IWDG window register (IWDG_WINR) can be refreshed in real-time. The current count value must be read and verified whether it is within the allowed range before performing the reload operation. To modify the window value after performing the reload operation, follow the steps below:

1) Refresh the counter value by writing 0x0000 AAAA in the IWDG control register (IWDG_CR).
2) Enable register access by writing 0x0000 5555 in the IWDG control register (IWDG_CR).
3) Verify that the value of IWDG count register (IWDG_CNT) is 0x00.
4) Write the new window value to the IWDG window register (IWDG_WINR).
5) Read the value of the window register (IWDG_WINR) to confirm the correct write operation.

Figure 24-2 IWDG window diagram



24.3.4 Overflow time setting

The IWDG internal prescaler is set to 128, offering 8 available presets for the divided counter overflow time. The relationship between the overflow time and the counting period is as follows:

$t_{IWDG} = t_{RCL} \times 128 \times N$

Table 24-1 IWDG overflow time table

RCL	Counting period (N)	Overflow time (ms)
32 KHz	32	128
	64	256
	128	512
	256	1024
	512	2048
	1024	4096
	2048	8192
	4096	16384

Real-time modification of the overflow time can be configured as follows:

1) Ensure the IWDG is enabled and operational.
2) Refresh the counter value by writing 0x0000 AAAA in the IWDG control register (IWDG_CR).
3) Enable register access by writing 0x0000 5555 in the IWDG control register (IWDG_CR).
4) Verify that the value of IWDG counter register (IWDG_CNT) is 0x00.
5) Modify the counter overflow time in the IWDG configure register (IWDG_CFG).

6) Read the selected overflow time value of the IWDG configure register (IWDG_CFG) to confirm the correct write operation.

24.3.5 Debug mode

When the device enters Debug mode (core halted), the IWDG counter either continues to work normally or stops, depending on the configuration of the IWDG_HOLD bit in the *APB1 freeze register (DBG_APB1_FZ)*.

24.4 IWDG low-power mode

Table 24-2 The effect to IWDG in Low-power mode

Mode	Description
Sleep	No impact
Stop	No impact. The IWDG continues counting or stops counting based on the IWDG_STOP bit in the *FLASH option register2 (FLASH_OPTR2)*.

24.5 IWDG registers

The IWDG registers can only be accessed by words (32-bit).

Table 24-3 IWDG base address

Peripheral	Base address
IWDG	0x4000 3000

24.5.1 IWDG control register (IWDG_CR)

Address offset: 0x00

Reset value: 0x0000 0000

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

Bits	Name	Description
31:16	Reserved	Must be kept at reset value
15:0	KEY[15:0]	Key value Write only, read 0x0000. These bits must be written by software at regular intervals with the key value 0x0000 AAAA, otherwise the watchdog generates a reset when the counter overflows. Writing the key value 0x0000 5555 to enable access to the IWDG configuration register (IWDG_CFG) and the IWDG window register (IWDG_WINR). Writing the key value 0x0000 CCCC starts the watchdog.

24.5.2 IWDG configuration register (IWDG_CFG)

Address offset: 0x04

Reset value: 0x0000 0000

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

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

Bits	Name	Description
31:3	Reserved	Must be kept at reset value
2:0	OVP[2:0]	Overflow time These bits are write access protected and used to select the counter overflow time. 000: 128ms 001: 256ms 010: 512ms 011: 1.024s 100: 2.048s 101: 4.096s 110: 8.192s 111: 16.384s

24.5.3 IWDG window register (IWDG_WINR)

Address offset: 0x10

Reset value: 0x0000 0000

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

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

Bits	Name	Description
31:12	Reserved	Must be kept at reset value
11:0	WIN[11:0]	Watchdog counter window value

24.5.4 IWDG count register (IWDG_CNT)

Address offset: 0x14

Reset value: 0x0000 0000

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

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

Bits	Name	Description
31:12	Reserved	Must be kept at reset value
11:0	CNT[11:0]	Watchdog counter value Since the counter clock operates asynchronously with the APB bus, software must read the counter value at least twice consecutively and only consider the result to be stable when consecutive reads yield identical values.

25 System window watchdog (WWDG)

25.1 Introduction

The system window watchdog (WWDG) is used to detect the occurrence of a software fault and trigger a system reset when the counter reaches its overflow condition.

The WWDG clock is prescaled from the APB1 clock and has a configurable time-window that can be programmed to detect abnormal operations.

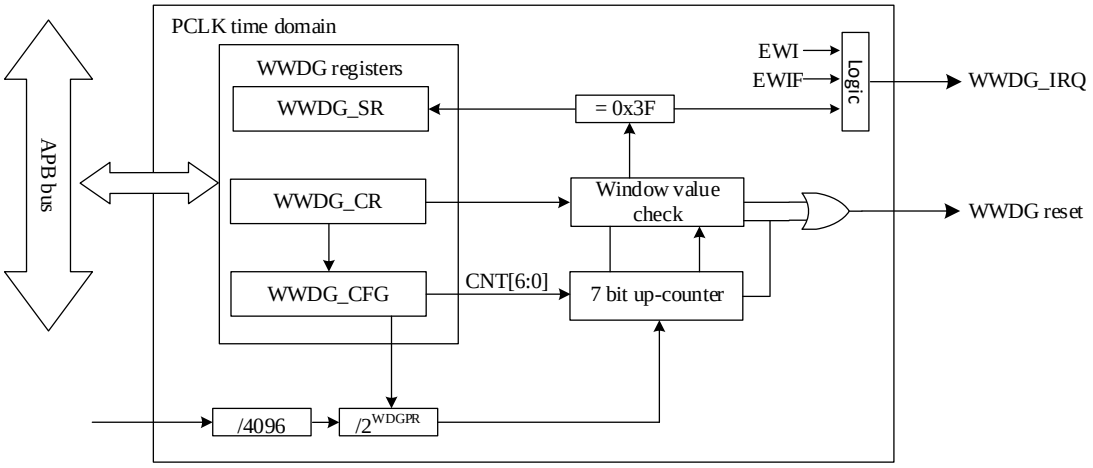
25.2 WWDG main features

- Up-counter
- Conditional reset:
  - Reset (if watchdog activated) when the up-counter value becomes greater than 0x3F
  - Reset (if watchdog activated) if the upcounter is reloaded outside the window
- Early wakeup interrupt (EWI): triggered (if enabled and the watchdog activated) when the upcounter is equal to 0x3F.

25.3 WWDG functional description

25.3.1 Block diagram

Figure 25-1 WWDG block diagram



The application program must write in the WWDG_CR register at regular intervals

during normal operation to prevent a MCU reset. This operation must occur only when the counter value is lower than 0x40 and higher than the window register value.

25.3.2 Controlling the up-counter

When the window watchdog is enabled, it cannot be disabled except by a reset.

If the window watchdog is activated (the WDGEN bit is set in the WWDG_CR register) and when the 7-bit up-counter (CNT[6:0] bits) is incremented from 0x3F to 0x40 (CNT[6] set to 1), it initiates a reset.

The window value is configured through the WWDG configuration register (WWDG_CFG). If the software reloads the counter while the counter is lower than the value stored in the window register, then a reset is generated. Therefore, the up-counter must be reloaded only when the counter value is greater than the window register value and lower than 0x40.

The CNT[6:0] bits store the current count value of the WWDG. The maximum counting period is determined by three factors: the APB1 clock period, the WDGPR[2:0] bit field in the WWDG configuration register (WWDG_CFG), and the CNT[6:0] bit field in the WWDG control register (WWDG_CR).

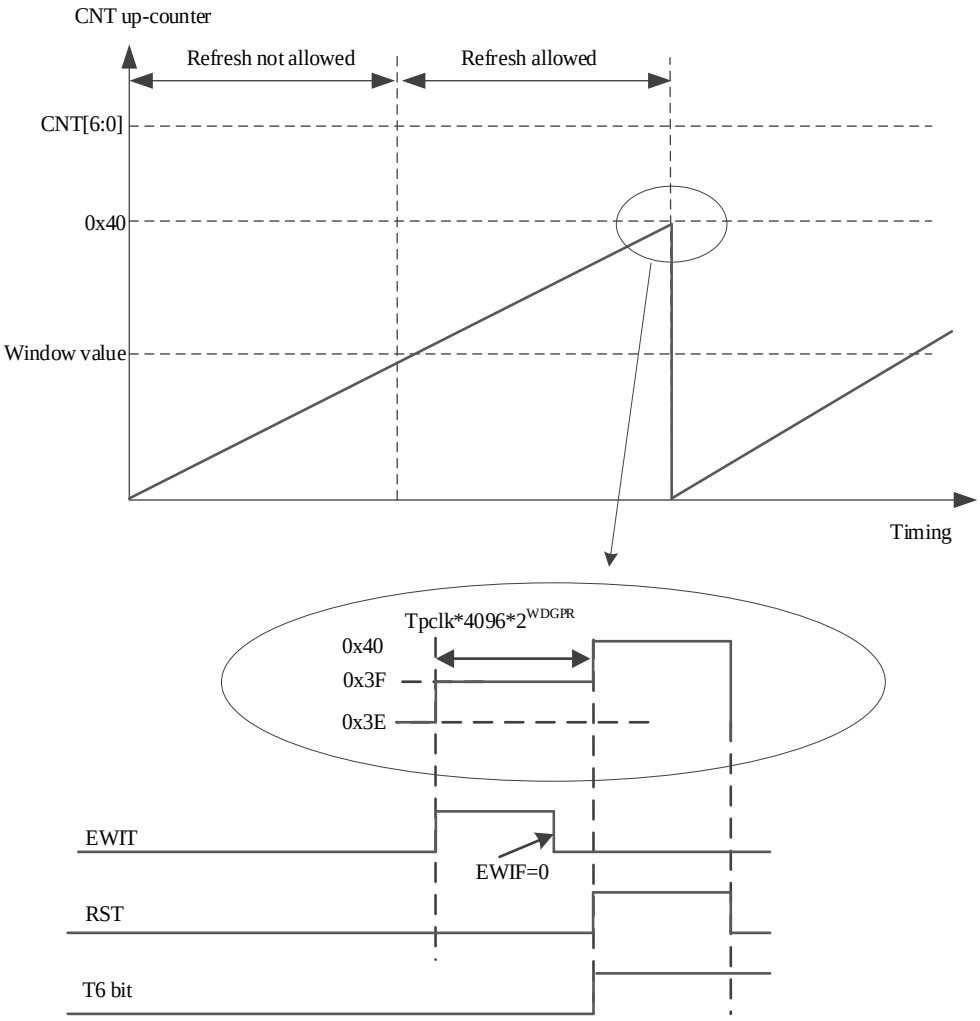
25.3.3 The early wakeup interrupt

The early wakeup interrupt (EWI) can be used if specific safety operations or data logging must be performed before the actual reset is generated. The EWI interrupt is enabled by setting the EWI bit in the WWDG_CFG register. When the up-counter reaches the value 0x3F, an EWI interrupt is generated and the corresponding interrupt service routine (ISR) can be used to trigger specific actions (such as communications or data logging), before resetting the device.

The EWI interrupt is cleared by writing '0' to the EWIF bit in the status register (WWDG_SR).

25.3.4 Timeout setting

Figure 25-2 WWDG timing diagram



The formula to calculate the timeout value is given by:

$t_{WWDG} = t_{PCLK} \times 4096 \times 2^{WDGPR[2:0]} \times (0x40 - CNT[6:0])$ (ms)

where:

t_{WWDG}: WWDG timeout

t_{PCLK}: APB1 clock period measured in ms

4096: value corresponding to internal divider

As an example, let's assume APB frequency is equal to 48 MHz, WDGPR[2:0] is set to 1 and CNT[6:0] is set to 0x2B:

$t_{WWDG} = (1/48000) \times 4096 \times 2^1 \times (0x40 - 0x2B) = 3.58$ (ms)

25.3.5 Debug mode

When the device enters debug mode (processor halted), the WWDG counter either continues to work normally or stops, depending on the configuration of the WWDG_HOLD bit in the *APB1 freeze register (DBG_APB1_FZ)*.

25.4 WWDG registers

The WWDG registers can only be accessed by words (32-bit).

Table 25-1 WWDG base address

Peripheral	Base address
WWDG	0x4000 2C00

25.4.1 WWDG control register (WWDG_CR)

Address offset: 0x00

Reset value: 0x0000 0000

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

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

Bits	Name	Description
31:8	Reserved	Must be kept at reset value
7	WDGEN	Activation bit This bit is set by software and only cleared by hardware after a reset. Writing 0 to this bit by software is invalid. When WDGEN = 1, the watchdog can generate a reset. 0: Watchdog disabled 1: Watchdog enabled
6:0	CNT[6:0]	7-bit counter These bits contain the value of the watchdog counter, incremented every ($4096 \times 2^{WDGPR[2:0]}$) PCLK cycles. A reset is produced when it is incremented from 0x3F to 0x40 (CNT[6] set to 1).

25.4.2 WWDG configuration register (WWDG_CFG)

Address offset: 0x04

Reset value: 0x0000 3800

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		WDGPR[2:0]			Res.	EWI	Res.		WIN[6:0]						
		rw	rw	rw		rs			rw	rw	rw	rw	rw	rw	rw

Bits	Name	Description
31:14	Reserved	Must be kept at reset value
13:11	WDGPR[2:0]	Clock prescaler for counter The pre-divider setting after dividing PCLK by 4096 000: div 1 001: div 2 010: div 4 011: div 8 100: div 16 101: div 32 110: div 64 111: div 128
10	Reserved	Must be kept at reset value
9	EWI	Early wakeup interrupt When set, an interrupt occurs whenever the counter reaches the value 0x3F. This interrupt is only cleared by hardware after a reset.
8:7	Reserved	Must be kept at reset value
6:0	WIN[6:0]	Window value

25.4.3 WWDG status register (WWDG_SR)

Address offset: 0x08

Reset value: 0x 0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.															EWIF
															rc_w0

Bits	Name	Description
31:1	Reserved	Must be kept at reset value
0	EWIF	Early wakeup interrupt flag This bit is set by hardware when the counter has reached the value 0x3F. It must be cleared by software by writing ‘0’. Writing ‘1’ has no effect. This bit is also set if the interrupt is not enabled.

26 SysTick timer (SysTick)

26.1 Introduction

ARM Cortex-M0+ provides a SysTick timer.

26.2 SysTick main features

- 24-bit down counter.
- Exception interrupt handing
- Selectable clock sources: HCLK or HCLK/8

26.3 SysTick functional description

The 24-bit SysTick timer initiates down counting from current value stored in the SysTick_VAL register when enabled by setting the ENABLE bit in the SysTick_CTRL register to 1. Upon reaching zero, the counter automatically reloads the value from the SysTick_LOAD register on the subsequent clock edge, then continues decrementing in a cyclic manner. This hardware-managed reload mechanism guarantees deterministic periodic timing generation.

The COUNTFLAG bit in the SysTick_CTRL register is asserted when the decrementing counter reaches zero. This flag is automatically cleared upon read access to the status register.

Writing any value to the SysTick_VAL register triggers an atomic clear of both the SysTick_VAL register and COUNTFLAG bit, followed by a synchronous reload of the value from the SysTick_LOAD register into the SysTick_VAL register on the subsequent clock edge to restart down counting silently without generating an interrupt. When reading the SysTick_VAL register, the operation returns the instantaneous counter value.

Writing a value of ‘0’ to the SysTick_LOAD register causes the counter to automatically halt after completing the current count cycle. After stopping, the counter value remains at ‘0’.

Before enabling the SysTick timer for the first time, first write the desired value to the SysTick_LOAD register, then write to the SysTick_VAL register. This sequence ensures proper reload value configuration and clears the SysTick_VAL register. When the timer is subsequently enabled, it starts decrementing directly from the preloaded value from the SysTick_LOAD register.

The TENMS[23:0] bit field in the SysTick_CALIB register provides the calibration value for 1ms timing intervals when the external reference clock HCLK/8 is selected as the counter clock source.

When the processor core is halted in Debug mode, the SysTick timer automatically suspends its counting operation.

26.4 SysTick registers

26.4.1 SysTick control and status register (SysTick_CTRL)

Address offset: 0xE000 E010

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.													CLK SOURCE	TICK INT	ENABLE
													rw	rw	rw

Bits	Name	Description
31:17	Reserved	Must be kept at reset value
16	COUNTFLAG	Overflow flag This status bit is set when the counter decrements from 1 to 0. Reading the SysTick_CTRL register automatically clears this flag. Writing any value to the SysTick_VAL register clears the flag. 0: No overflow has occurred 1: Overflow has occurred
15:3	Reserved	Must be kept at reset value
2	CLKSOURCE	SysTick clock source selection 0: External reference clock HCLK/8 1: Core clock HCLK
1	TICKINT	SysTick interrupt enable 0: Disable 1: Enable
0	ENABLE	SysTick counter enable 0: Disable 1: Enable

26.4.2 SysTick reload value register (SysTick_LOAD)

Address offset: 0xE000 E014

Reset value: 0x00XX XXXX

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

Bits	Name	Description
31:24	Reserved	Must be kept at reset value
23:0	RELOAD[23:0]	SysTick reload value When the counter decrements to 0, the reload value is automatically loaded into the SysTick_VAL register.

26.4.3 SysTick current value register (SysTick_VAL)

Address offset: 0xE000 E018

Reset value: 0x00XX XXXX

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

Bits	Name	Description
31:24	Reserved	Must be kept at reset value
23:0	CURRENT[23:0]	Current counter value Reading this register returns the current counter value Writing any value atomically clears both the register and the COUNTFLAG bit in the SysTick_CTRL register

26.4.4 SysTick calibration value register (SysTick_CALIB)

Address offset: 0xE000 E01C

Reset value: 0x4000 176F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NOREF	SKEW	Res.						TENMS[23:16]							
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
TENMS[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits	Name	Description
31	NOREF	Counting clock flag 0: External reference clock HCLK/8
30	SKEW	TENMS accuracy indicator 1: TENMS calibration value represents an approximate 1ms.
29:24	Reserved	Must be kept at reset value
23:0	TENMS[23:0]	1 ms calibration counter value 0x176F The clock source is configured to use the reference clock HCLK/8, which operates at 48 MHz. The SysTick counter value corresponding to a 1 ms interval is calculated based on this configuration.

27 Real-time clock (RTC)

27.1 Introduction

The real-time clock (RTC) provides calendar timekeeping functionality, recording year, month, day, hour, minute, second, and weekday information in BCD format. It includes independently configurable alarm and periodic auto-wakeup interrupts. The RTC supports operation and wake-up functionality across all low-power modes. Additionally, its built-in smooth digital calibration feature enhances the calendar's long-term timing accuracy.

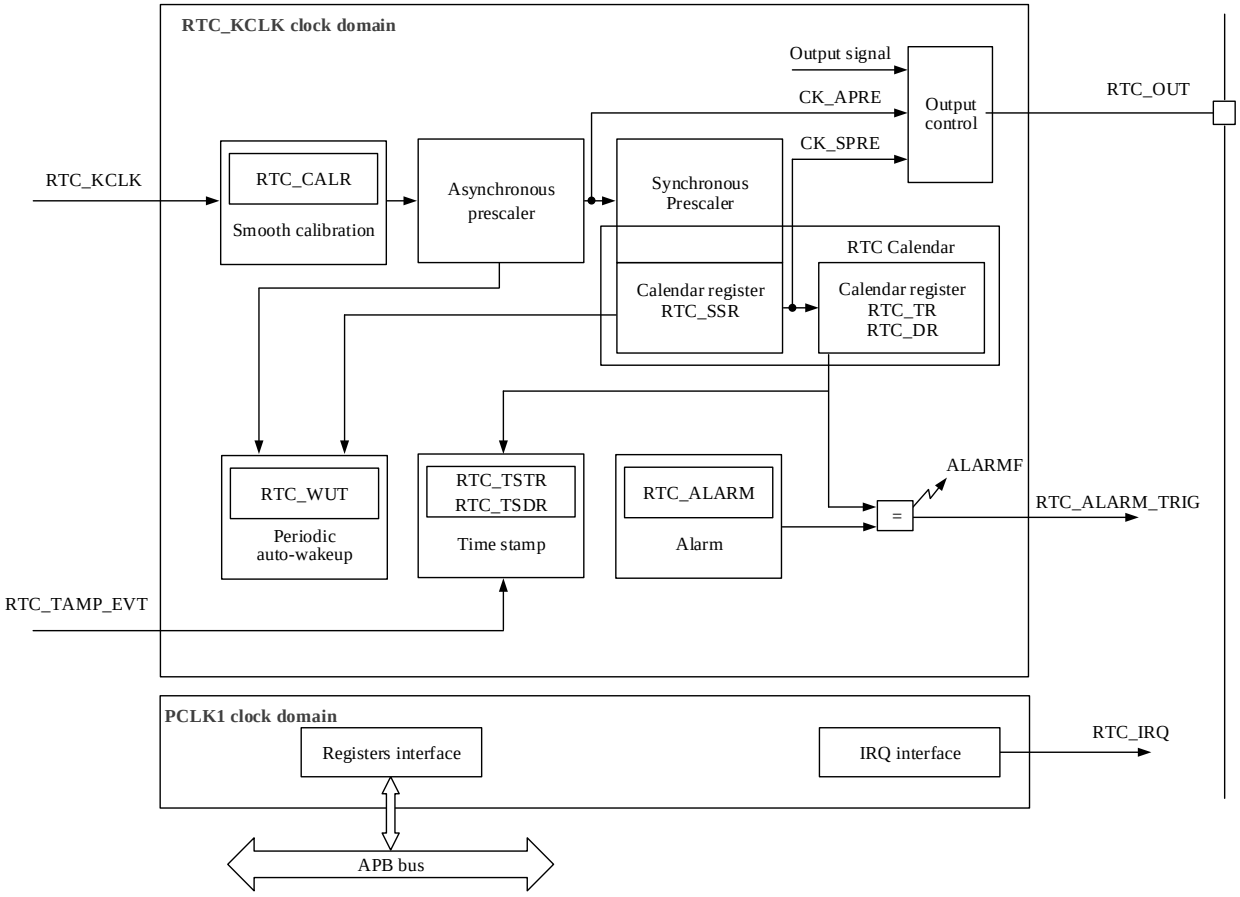
27.2 RTC main features

- Digital calibration circuit with ±0.477 ppm accuracy
- Calendar in BCD (binary-coded decimal) format
- Automatic correction for 28, 29 (leap year), 30, and 31 days of the month
- One programmable alarm
- A configurable set of periodic auto-wakeup interrupts with periods ranging from milliseconds to days
- The internal clock signal is available through a configurable pin (which can be used for temperature-calibrated error measurement)
- Timestamp function triggered by tamper event
- RTC retains functionality through system resets

27.3 RTC functional description

27.3.1 RTC block diagram

Figure 27-1 RTC block diagram



27.3.2 RTC pins and internal signals

Table 27-1 RTC output pins

Pin name	Signal type	Description
RTC_OUT	Output	Output internal clock and flags

The outputs on the RTC_OUT pin (PC13) are selected by configuration of the OUT_SEL[2:0] bits in the RTC_CR register:

- CK_APRE clock: 256 Hz clock output (from the asynchronous prescaler).
- CK_SPRE clock: 1 Hz clock output (from the synchronous prescaler).
- ALARMF flag: Alarm match event flag output.
- Calendar timing carry signals: Day, hour, and minute carry signals output.

- Calibration periodic signal: Digital calibration periodic signal output.

After configuring the output signal, the additional functionality of the RTC_OUT pin can be enabled by setting the OUTEN bit in the RTC_CR register to 1. The pin is fixed as a push-pull output without pull-up or pull-down resistors.

The polarity of the RTC_OUT is determined by the POL control bit in the RTC_CR register so that the opposite of the selected signal is output when POL is set to 1.

RTC_OUT and TAMP_IN are mapped on the same pin (PC13). When both functions are enabled, the hardware prioritizes the RTC_OUT signal.

Table 27-2 RTC internal input/output signals

Internal signal name	Signal type	Description
RTC_KCLK	Input	RTC kernel clock
PCLK1	Input	APB1 clock PCLK1
RTC_TAMP_EVT	Input	Tamper event detected in TAMP peripheral
RTC_ALARM_TRIG	Output	RTC alarm event detection trigger
RTC_IRQ	Output	RTC interrupt

27.3.3 RTC register write protection

The RTC module incorporates a hardware write protection feature to avoid unintended register access. After the V_{CORE_AON} reset, the RTC module enters a write-protected state. Writes to all other RTC registers (excluding the RTC_WPR and RTC_CLR registers) are ignored.

The following steps are required to unlock the write protection on the protected RTC registers:

1) Write 0xCA into the RTC_WPR register.
2) Write 0x53 into the RTC_WPR register.

Writing a wrong key reactivates the write protection.

27.3.4 RTC clock and prescalers

The RTC clock (RTC_KCLK) sources can be:

- A 32.768 KHz external crystal (LXTAL).
- The internal low-speed RC oscillator (RCL with typical frequency of 32 KHz).

RTC clock is selected by configuration of RTCSEL[1:0] bit field in the *V_{CORE_AON}*

domain control register (RCC_AWCR). The RTC_KCLK clock is usually the LXTAL at 32.768 KHz although it is possible to select other clock sources in the RCC. Refer to: *RTC and LCD clock*.

The RTC employs two internal prescalers to divide the input RTC_KCLK clock, generating a 1 Hz clock signal as the counting clock for the calendar function:

- Asynchronous prescaler: Division factor is 128 to obtain an internal clock frequency of 256 Hz (CK_APRE).
- Synchronous prescaler: Division factor is 256 to obtain an internal clock frequency of 1 Hz (CK_SPRE).

The synchronous prescaler consists of an up-counter whose count value represents the calendar subseconds. The current count of the synchronous prescaler can be read via the RTC_SSR register, which has a resolution of 3.91 ms (1/256 seconds).

27.3.5 Reading the calendar

RTC provides a calendar timekeeping function, which records year, month, day, hours, minutes, seconds, and weekday information in BCD (binary-coded decimal) format and can automatically correct for 28, 29 (leap year), 30, and 31 days of the month.

The RTC calendar information can be retrieved through the following calendar registers:

- RTC_SSR for the subseconds
- RTC_TR for the time
- RTC_DR for the date

The RTC_KCLK clock (used for calendar updating) and the PCLK1 clock (used for calendar register accesses) operate in asynchronous domains. The following recommended configuration ensures reliable RTC time reading:

1) Read all relevant calendar registers consecutively.
2) Repeat the consecutive read operations.
3) Compare the two read results. If they match, the data is valid. If not, repeat the steps above.

There is an automatic correction for 28, 29 (leap year), 30, and 31 days of the month between the years 2000 and 2099. Calendar overflow occurs when the RTC count reaches December 31, 2099, at 23:59:59.

27.3.6 RTC calendar configuration

To program the initial time and date calendar values, the following sequence is required:

1) Set MODSEL bit to 1 in the RTC_CSR register to enter initialization mode.
2) Poll MODSTA bit of in the RTC_CSR register. The initialization phase mode is entered when MODSTA is set to 1. In this mode, the calendar counting is halted.
3) Load the initial time and date values in the calendar registers.
4) Exit the initialization mode by clearing the MODSEL bit; this also automatically clears the MODSTA bit. The actual calendar counter value is then automatically loaded and the counting restarts after 1 RTC_KCLK clock cycles.

The current value of the synchronization prescaler can be retrieved by reading the RTC_SSR register. When exiting initialization mode, the counter value in the RTC asynchronous prescaler is cleared to enable subsecond-level calendar time synchronization.

27.3.7 RTC alarm

The programmable alarm function is enabled through the ALARME bit in the RTC_CR register.

Upon enabling the alarm, the ALARMF bit in the RTC_SR register is set to 1 if the calendar seconds, minutes and hours match the values programmed in the alarm register RTC_ALARMR. If the ALARMIE bit in the RTC_CR register is enabled, the ALARMIF interrupt flag in the RTC_ISR register asserts, triggering an interrupt.

When an alarm match event is triggered, the RTC_ALARM_TRIG signal is automatically asserted (set to high level). This signal is automatically de-asserted (set to low level) after the ALARMF flag in the RTC_SR register is cleared. The RTC_ALARM_TRIG signal can be configured to trigger the LPTIM1 counter.

Each calendar field which is not compared can be independently selected through the MSKx bits of the RTC_ALARMR register.

The configuration steps for the RTC alarm are as follows:

1) Clear ALARME in the RTC_CR register to disable alarm.
2) Program the alarm registers (RTC_ALARMR).
3) Set ALARME in the RTC_CR register to enable alarm.

27.3.8 RTC periodic auto-wakeup interrupt

The RTC supports periodic auto-wakeup interrupts at the following frequencies and time intervals: 1024 Hz, 256 Hz, 64 Hz, 16 Hz, 8 Hz, 4 Hz, 2 Hz, and 1 Hz. It also supports 1-second, 1-minute, and 1-hour intervals. Each interrupt source can be individually enabled, or multiple interrupts can be enabled simultaneously.

The corresponding flags in the RTC_SR register are set when the timing cycle is reached. The flags in the RTC_ISR register are set, and interrupts are generated, if the corresponding control bit is set in the RTC_WUT register.

27.3.9 RTC smooth digital calibration

The RTC frequency can be digitally calibrated with a calibration accuracy of about 0.477 ppm with a range from -487.1 ppm to +488.5 ppm.

Temperature data acquired via the ADC temperature sensor can be used with the smooth digital calibration function to implement temperature compensation for RTC, thereby enhancing timing accuracy across varying environmental temperatures.

RTC calibration period

The smooth digital calibration is performed during a cycle of about 2^{20} RTC_KCLK pulses, or 32 seconds when the input frequency is 32768 Hz, with a resolution of about 0.954 ppm ($1/2^{20}$).

This cycle is maintained by a 20-bit counter, CAL_CNT[19:0], clocked by RTC_KCLK. At the end of each calibration cycle, the CALF flag in the RTC_SR register is set to 1. If the CALIE bit in the RTC_CR register is enabled, the CALIF interrupt flag in the RTC_ISR register asserts, generating an interrupt.

RTC calibration parameter

By configuring the CALM[8:0] bit field in the RTC_CALR register, the frequency can be reduced by up to 487.1 ppm ($511/2^{20}$).

By configuring the CALP bit in the RTC_CALR register, the frequency can be increased by up to 488.5 ppm ($512/2^{20}$).

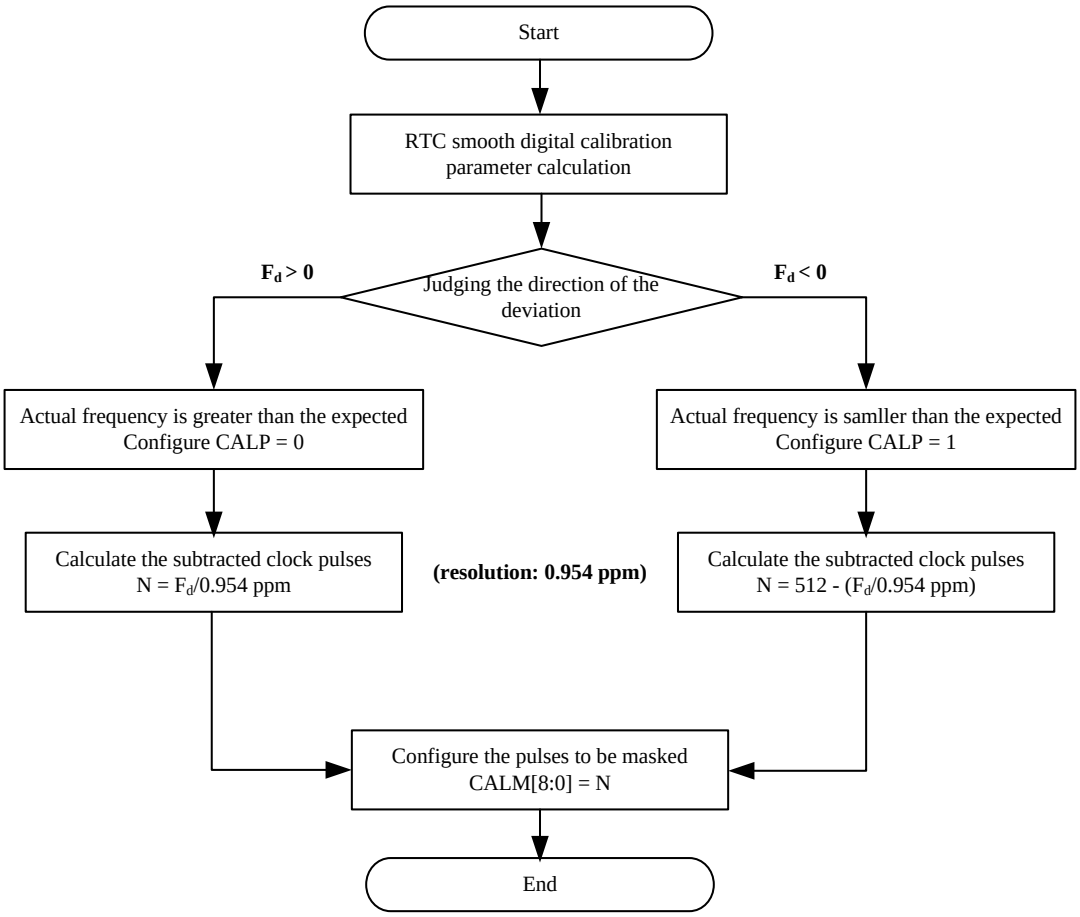
Using CALM together with CALP, an offset ranging from -511 to +512 RTC_KCLK cycles can be added during the 32-second cycle, which translates to a calibration range of -487.1 ppm to +488.5 ppm.

RTC calibration parameter calculation

When calibrating, first measure the frequency deviation of the RTC_KCLK by configuring the calibration parameters in the RTC_CALR register to 0 and routing either the asynchronous (CK_APRE) or synchronous (CK_SPRE) prescaler clock through the RTC_OUT pin.

Measure the frequency of the RTC_OUT clock signal, compute the frequency error F_d of the RTC clock, convert F_d to ppm units, and calculate the calibration parameters using the procedure below:

Figure 27-2 Calibration parameter calculation



The following provides examples of calculating calibration parameters for different RTC clock frequency deviation:

- Example 1: the frequency deviation F_d = 20 ppm:

  1) F_d > 0, the measured frequency exceeds the target frequency
  2) Configure CALP = 0 (No pulses are inserted)
  3) Configure CALM[8:0] = 20/0.954 = 20.964 ≈ 21

- Example 2: the frequency deviation F_d = -20 ppm:

  1) F_d < 0, the measured frequency is less than the target frequency

2) Configure CALP = 1 (512 pulses are inserted)

3) Configure CALM[8:0] = 512 – (20/0.954) ≈ 491

On-the-Fly Recalibration

The calibration register (RTC_CALR) can be updated on-the-fly, and the configured calibration parameters remain effective persistently. The process is as follows:

1) Poll the RTC_CSR/CALSTA (re-calibration pending flag).
2) If it is set to 0, write a new value to RTC_CALR, if necessary. CALSTA is then automatically set to 1.
3) After the write operation to the RTC_CALR register, the new calibration settings take effect when the CALSTA bit is cleared automatically.

27.3.10 Timestamp function

The calendar is saved in the timestamp registers (RTC_TSTR, RTC_TSDR) when a tamper event is detected, and the timestamp flag bit (TSF) in the RTC_SR register is set. Write 1 in the CTSF bit of the RTC_CLR register will clear the TSF flag.

The values in the RTC_TSTR and RTC_TSDR registers are only valid when the TSF flag in the RTC_SR register is set to 1. Clearing the TSF flag to 0 will not erase the values stored in the timestamp registers.

A new tamper event will trigger an update of the timestamp registers (RTC_TSTR, RTC_TSDR) only when the TSF flag is 0. Once the timestamp flag (TSF) is set to 1, subsequent tamper events will not cause further updates to the timestamp register values.

27.3.11 Debug support

When the device enters debug mode (processor halted), to pause the RTC counter, set the RTC_HOLD bit in the *APB1 freeze register (DBG_APB1_FZ) to 1*.

27.4 RTC low-power modes

The RTC operates normally in all power modes. RTC interrupts cause the device to wake up from the Sleep and Stop modes. In V_{BAT} mode, the LXTAL clock source must be selected.

The output function of the RTC_OUT pin remains enabled in all low-power modes.

27.5 RTC interrupts

If a flag is set to '1' in the RTC_ISR register, this indicates that the corresponding interrupt has been generated.

Table 27-3 RTC interrupt requests

Interrupt event	Event flag	Enable control bit	Interrupt clear method
Alarm	ALARMIF	ALARMIE	write 1 in CALARMF
Calibration period	CALIF	CALIE	write 1 in CCALF
Wakeup timer	1024HZIF	1024HZIE	write 1 in C1024HZF
	256HZIF	256HZIE	write 1 in C256HZF
	64HZIF	64HZIE	write 1 in C64HZF
	16HZIF	16HZIE	write 1 in C16HZF
	8HZIF	8HZIE	write 1 in C8HZF
	4HZIF	4HZIE	write 1 in C4HZF
	2HZIF	2HZIE	write 1 in C2HZF
	SECIF	SECIE	write 1 in CSECF
	MINIF	MINIE	write 1 in CMINF
	HOURIF	HOURIE	write 1 in CHOURF
	DAYIF	DAYIE	write 1 in CDAYF

27.6 RTC registers

The RTC registers can only be accessed by words (32-bit).

Table 27-4 RTC base address

Peripheral	Base address
RTC	0x4000 2800

27.6.1 RTC time register (RTC_TR)

Address offset: 0x00

Reset value: 0x0000 0000

Note: This register can only be reset by V_{CORE_AON} domain reset.

This register is write protected. The write access procedure is described in RTC register write protection.

This register must be written in initialization mode only.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.										HT[1:0]		HU[3:0]			
										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

Bits	Name	Description
31:22	Reserved	Must be kept at reset value
21:20	HT[1:0]	Hour tens in BCD format
19:16	HU[3:0]	Hour units in BCD format
15	Reserved	Must be kept at reset value
14:12	MNT[2:0]	Minute tens in BCD format
11:8	MNU[3:0]	Minute units in BCD format
7	Reserved	Must be kept at reset value

6:4	ST[2:0]	Second tens in BCD format
3:0	SU[3:0]	Second units in BCD format

27.6.2 RTC date register (RTC_DR)

Address offset: 0x04

Reset value: 0x0000 2101

Note: This register can only be reset by V_{CORE_AON} domain reset.

This register is write protected. The write access procedure is described in RTC register write protection.

This register must be written in initialization mode only.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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.		DT[1:0]		DU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw

Bits	Name	Description
31:24	Reserved	Must be kept at reset value
23:20	YT[3:0]	Year tens in BCD format
19:16	YU[3:0]	Year units in BCD format
15:13	WDU[2:0]	Week day units 000: Forbidden 001: Monday ⋮ 111: Sunday
12	MT	Month tens in BCD format
11:8	MU[3:0]	Month units in BCD format

7:6	Reserved	Must be kept at reset value
5:4	DT[1:0]	Date tens in BCD format
3:0	DU[3:0]	Date units in BCD format

27.6.3 RTC sub second register (RTC_SSR)

Address offset: 0x08

Reset value: 0x0000 0000

Note: This register can only be reset by V_{CORE_AON} domain reset.

This register is write protected. The write access procedure is described in RTC register write protection.

This register must be written in initialization mode only.

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

Bits	Name	Description
31:8	Reserved	Must be kept at reset value
7:0	SS[7:0]	Sub second value Sub second is the fractional part of second Unit: 3.91 ms (1/256 s)

27.6.4 RTC control and status register (RTC_CSR)

Address offset: 0x0C

Reset value: 0x0000 0000

Note: This register can be reset by V_{CORE_AON} domain reset. A system reset can reset the MODSEL and MODSTA bits.

This register is write protected. The write access procedure is described in RTC

register write protection.

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								MODSEL	MODSTA	Res.					
								rw	r						

Bits	Name	Description
31:17	Reserved	Must be kept at reset value
16	CALSTA	RTC smooth digital calibration status flag This bit is automatically set to 1 when software writes to the RTC_CALR register, indicating that the RTC_CALR register is blocked. When the new calibration settings are taken into account, this bit returns to 0 automatically. 0: Calibration settings are taken into account, the new parameter can be written 1: The calibration parameters have been written but are not yet in effect
15:8	Reserved	Must be kept at reset value
7	MODSEL	Initialization mode 0: Free running mode 1: Initialization mode In initialization mode, the calendar counters are stopped.
6	MODSTA	Initialization flag When this bit is set to 1, the RTC is in initialization state, and the time, date and prescaler registers can be updated. 0: RTC operates normally 1: RTC is in initialization mode
5:0	Reserved	Must be kept at reset value

27.6.5 RTC wakeup timer register (RTC_WUT)

Address offset: 0x14

Reset value: 0x0000 0000

Note: This register can only be reset by V_{CORE_AON} domain reset.

This register is write protected. The write access procedure is described in RTC register write protection.

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					1024HZ IE	256HZ IE	64HZ IE	16HZ IE	8HZ IE	4HZ IE	2HZ IE	SEC IE	MIN IE	HOUR IE	DAY IE
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Name	Description
31:11	Reserved	Must be kept at reset value
10	1024HZIE	1024 Hz periodic auto-wakeup interrupt enable 0: Disable 1: Enable
9	256HZIE	256 Hz periodic auto-wakeup interrupt enable 0: Disable 1: Enable
8	64HZIE	64 Hz periodic auto-wakeup interrupt enable 0: Disable 1: Enable
7	16HZIE	16 Hz periodic auto-wakeup interrupt enable 0: Disable 1: Enable
6	8HZIE	8 Hz periodic auto-wakeup interrupt enable 0: Disable 1: Enable

5	4HZIE	4 Hz periodic auto-wakeup interrupt enable 0: Disable 1: Enable
4	2HZIE	2 Hz periodic auto-wakeup interrupt enable 0: Disable 1: Enable
3	SECIE	Second periodic auto-wakeup interrupt enable 0: Disable 1: Enable
2	MINIE	Minute periodic auto-wakeup interrupt enable 0: Disable 1: Enable
1	HOURIE	Hour periodic auto-wakeup interrupt enable 0: Disable 1: Enable
0	DAYIE	Day periodic auto-wakeup interrupt enable 0: Disable 1: Enable

27.6.6 RTC control register (RTC_CR)

Address offset: 0x18

Reset value: 0x0000 0000

Note: This register can only be reset by V_{CORE_AON} domain reset.

This register is write protected. The write access procedure is described in RTC register write protection.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.								OUT_SEL[2:0]			POL	OUTEN	Res.		
								rw	rw	rw	rw	rw			

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

		rw	rw				rw								

Bits	Name	Description
31:24	Reserved	Must be kept at reset value
23:21	OUT_SEL[2:0]	RTC_OUT output signal selection 000: CK_APRE (256Hz) clock is selected 001: CK_SPRE (1Hz) clock is selected 010: ALARMF flag is selected 011: Reserved (No signal) 100: Minute carry signal is selected 101: Hour carry signal is selected 110: Day carry signal is selected 111: Calibration cycle flag is selected
20	POL	RTC_OUT output polarity 0: Output polarity is the same to the signal 1: Output polarity is opposite of the signal
19	OUTEN	RTC_OUT output enable 0: Output disable 1: Output enable
18:14	Reserved	Must be kept at reset value
13	CALIE	Calibration period interrupt enable 0: Calibration period interrupt disable 1: Calibration period interrupt enable
12	ALARMIE	Alarm interrupt enable 0: Alarm interrupt disable 1: Alarm interrupt enable
11:9	Reserved	Must be kept at reset value
8	ALARME	Alarm enable 0: Alarm disable 1: Alarm enable

7:0	Reserved	Must be kept at reset value

27.6.7 RTC write protection register (RTC_WPR)

Address offset: 0x24

Reset value: 0x0000 0000

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

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

Bits	Name	Description
31:8	Reserved	Must be kept at reset value
7:0	KEY[7:0]	Write protection key Write the following keys consecutively in the correct sequence to unlock the register write protection: Key1: 0xCA Key2: 0x53 Refer to *RTC register write protection* for a description of how to unlock RTC register write protection.

27.6.8 RTC calibration register (RTC_CALR)

Address offset: 0x28

Reset value: 0x0000 0000

Note: This register can only be reset by V_{CORE_AON} domain reset.

This register is write protected. The write access procedure is described in RTC register write protection.

This register can be written only when CALSTA is 0.

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

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

CALP	Res.						CALM[8:0]								
rw							rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Name	Description
31:16	Reserved	Must be kept at reset value
15	CALP	Increase frequency of RTC by 488.5 ppm 0: No RTC_KCLK pulses are added 1: One RTC_KCLK pulse is effectively inserted every 2^{11} pulses Refer to: *RTC smooth digital calibration*
14:9	Reserved	Must be kept at reset value
8:0	CALM[8:0]	Calibration minus Used to configure the number of masked RTC_KCLK pulses within each calibration cycle Refer to: *RTC smooth digital calibration*

27.6.9 RTC timestamp time register (RTC_TSTR)

Address offset: 0x30

Reset value: 0x0000 0000

Note: This register can only be reset by V_{CORE_AON} domain reset. The contents of this register is valid only when the TSF bit in the RTC_SR register is set to 1.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.										HT[1:0]		HU[3:0]			
										r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	MNT[2:0]			MNU[3:0]				Res.	ST[2:0]			SU[3:0]			
	r	r	r	r	r	r	r		r	r	r	r	r	r	r

Bits	Name	Description
31:22	Reserved	Must be kept at reset value
21:20	HT[1:0]	Hour tens in BCD format
19:16	HU[3:0]	Hour units in BCD format

15	Reserved	Must be kept at reset value
14:12	MNT[2:0]	Minute tens in BCD format
11:8	MNU[3:0]	Minute units in BCD format
7	Reserved	Must be kept at reset value
6:4	ST[2:0]	Second tens in BCD format
3:0	SU[3:0]	Second units in BCD format

27.6.10 RTC timestamp date register (RTC_TSDR)

Address offset: 0x34

Reset value: 0x0000 0000

Note: This register can only be reset by V_{CORE_AON} domain reset. The contents of this register is valid only when the TSF bit in the RTC_SR register is set to 1.

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			MT	MU[3:0]				Res.		DT[1:0]		DU[3:0]			
			r	r	r	r	r			r	r	r	r	r	r

Bits	Name	Description
31:13	Reserved	Must be kept at reset value
12	MT	Month tens in BCD format
11:8	MU[3:0]	Month units in BCD format
7:6	Reserved	Must be kept at reset value
5:4	DT[1:0]	Date tens in BCD format

3:0	DU[3:0]	Date units in BCD format

27.6.11 RTC alarm register (RTC_ALARMR)

Address offset: 0x40

Reset value: 0x0000 0000

Note: This register can only be reset by V_{CORE_AON} domain reset.

This register is write protected. The write access procedure is described in RTC register write protection.

This register can be written only when ALARME of RTC_CR is 0.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.								MSK3	Res.	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
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

Bits	Name	Description
31:24	Reserved	Must be kept at reset value
23	MSK3	Alarm hours mask 0: Alarm set if the hours match 1: Hours do not care in alarm comparison
22	Reserved	Must be kept at reset value
21:20	HT[1:0]	Hour tens in BCD format
19:16	HU[3:0]	Hour units in BCD format
15	MSK2	Alarm minutes mask 0: Alarm set if the minutes match 1: Minutes do not care in alarm comparison
14:12	MNT[2:0]	Minute tens in BCD format

11:8	MNU[3:0]	Minute units in BCD format
7	MSK1	Alarm seconds mask 0: Alarm A set if the seconds match 1: Seconds do not care in alarm A comparison
6:4	ST[2:0]	Second tens in BCD format
3:0	SU[3:0]	Second units in BCD format

27.6.12 RTC status register (RTC_SR)

Address offset: 0x50

Reset value: 0x0000 0000

Note: This register can only be reset by V_{CORE_AON} domain reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.					1024HZF	256HZF	64HZF	16HZF	8HZF	4HZF	2HZF	SECF	MINF	HOURF	DAYF
					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
Res.												TSF	Res.	CALF	ALARMF
												r		r	r

Bits	Name	Description
31:27	Reserved	Must be kept at reset value
26	1024HZF	1024 Hz periodic auto-wakeup flag This flag is set at 1024 Hz when RTC clock is enabled.
25	256HZF	256 Hz periodic auto-wakeup flag This flag is set at 256 Hz when RTC clock is enabled.
24	64HZF	64 Hz periodic auto-wakeup flag This flag is set at 64 Hz when RTC clock is enabled.
23	16HZF	16 Hz periodic auto-wakeup flag This flag is set at 16 Hz when RTC clock is enabled.

22	8HZF	8 Hz periodic auto-wakeup flag This flag is set at 8 Hz when RTC clock is enabled.
21	4HZF	4 Hz periodic auto-wakeup flag This flag is set at 4 Hz when RTC clock is enabled.
20	2HZF	2 Hz periodic auto-wakeup flag This flag is set at 2 Hz when RTC clock is enabled.
19	SECF	Second periodic auto-wakeup flag This flag is set once per second when RTC clock is enabled.
18	MINF	Minute periodic auto-wakeup flag This flag is set once per minute when RTC clock is enabled.
17	HOURF	Hour periodic auto-wakeup flag This flag is set once per hour when RTC clock is enabled.
16	DAYF	Day periodic auto-wakeup flag This flag is set once per day when RTC clock is enabled.
15:4	Reserved	Must be kept at reset value
3	TSF	Timestamp flag This flag is set by hardware when a timestamp event occurs.
2	Reserved	Must be kept at reset value
1	CALF	Calibration period flag This flag is set by hardware at the end of every calibration period.
0	ALARMF	Alarm match event flag This flag is set by hardware when the time registers (RTC_TR) match the alarm register (RTC_ ALARMR).

27.6.13 RTC interrupt status register (RTC_ISR)

Address offset: 0x54

Reset value: 0x0000 0000

Note: This register can only be reset by V_{CORE_AON} domain reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.					1024HZ IF	256HZ IF	64HZ IF	16HZ IF	8HZ IF	4HZ IF	2HZ IF	SEC IF	MIN IF	HOUR IF	DAY IF
					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
Res.														CALIF	ALARM IF
														r	r

Bits	Name	Description
31:27	Reserved	Must be kept at reset value
26	1024HZIF	1024 Hz periodic auto-wakeup interrupt flag This flag is set by hardware when the 1024 Hz periodic auto-wakeup interrupt occurs.
25	256HZIF	256 Hz periodic auto-wakeup interrupt flag This flag is set by hardware when the 256 Hz periodic auto-wakeup interrupt occurs.
24	64HZIF	64 Hz periodic auto-wakeup interrupt flag This flag is set by hardware when the 64 HZ periodic auto-wakeup interrupt occurs.
23	16HZIF	16 Hz periodic auto-wakeup interrupt flag This flag is set by hardware when the 16 HZ periodic auto-wakeup interrupt occurs.
22	8HZIF	8 Hz periodic auto-wakeup interrupt flag This flag is set by hardware when the 8 HZ periodic auto-wakeup interrupt occurs.
21	4HZIF	4 Hz periodic auto-wakeup interrupt flag This flag is set by hardware when the 4 HZ periodic auto-wakeup interrupt occurs.
20	2HZIF	2 Hz periodic auto-wakeup interrupt flag

		This flag is set by hardware when the 2 HZ periodic auto-wakeup interrupt occurs.
19	SECIF	Second periodic auto-wakeup interrupt flag This flag is set by hardware when the second periodic auto-wakeup interrupt occurs.
18	MINIF	Minute periodic auto-wakeup interrupt flag This flag is set by hardware when the minute periodic auto-wakeup interrupt occurs.
17	HOURIF	Hour periodic auto-wakeup interrupt flag This flag is set by hardware when the hour periodic auto-wakeup interrupt occurs.
16	DAYIF	Day periodic auto-wakeup interrupt flag This flag is set by hardware when the day periodic auto-wakeup interrupt occurs.
15:2	Reserved	Must be kept at reset value
1	CALIF	Calibration period interrupt flag This flag is set by hardware when the calibration period interrupt occurs.
0	ALARMIF	Alarm interrupt flag This flag is set by hardware when the alarm interrupt occurs.

27.6.14 RTC status clear register (RTC_CLR)

Address offset: 0x5C

Reset value: 0x0000 0000

Note: This register can only be reset by V_{CORE_AON} domain reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.					C1024HZF	C256HZF	C64HZF	C16HZF	C8HZF	C4HZF	C2HZF	CSECF	CMINF	CHOURF	CDAYF
					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

Res.												CTSF	Res.	CCALF	CALARMF
												w		w	w

Bits	Name	Description
31:27	Reserved	Must be kept at reset value
26	C1024HZF	Clear 1024 Hz periodic auto-wakeup flag Writing 1 in this bit clears the 1024HZF bit in the RTC_SR register and the 1024HZIF bit in the RTC_ISR register.
25	C256HZF	Clear 256 Hz periodic auto-wakeup flag Writing 1 in this bit clears the 256HZF bit in the RTC_SR register and the 256HZIF bit in the RTC_ISR register.
24	C64HZF	Clear 64 Hz periodic auto-wakeup flag Writing 1 in this bit clears the 64HZF bit in the RTC_SR register and the 64HZIF bit in the RTC_ISR register.
23	C16HZF	Clear 16 Hz periodic auto-wakeup flag Writing 1 in this bit clears the 16HZF bit in the RTC_SR register and the 16HZIF bit in the RTC_ISR register.
22	C8HZF	Clear 8 Hz periodic auto-wakeup flag Writing 1 in this bit clears the 8HZF bit in the RTC_SR register and the 8HZIF bit in the RTC_ISR register.
21	C4HZF	Clear 4 Hz periodic auto-wakeup flag Writing 1 in this bit clears the 4HZF bit in the RTC_SR register and the 4HZIF bit in the RTC_ISR register.
20	C2HZF	Clear 2 Hz periodic auto-wakeup flag Writing 1 in this bit clears the 2HZF bit in the RTC_SR register and the 2HZIF bit in the RTC_ISR register.
19	CSECF	Clear Second periodic auto-wakeup flag Writing 1 in this bit clears the SECF bit in the RTC_SR register and the SECIF bit in the RTC_ISR register.

18	CMINF	Clear Minute periodic auto-wakeup flag Writing 1 in this bit clears the MINF bit in the RTC_SR register and the MINIF bit in the RTC_ISR register.
17	CHOURF	Clear Hour periodic auto-wakeup flag Writing 1 in this bit clears the HOURF bit in the RTC_SR register and the HOURIF bit in the RTC_ISR register.
16	CDAYF	Clear Day periodic auto-wakeup flag Writing 1 in this bit clears the DAYF bit in the RTC_SR register and the DAYIF bit in the RTC_ISR register.
15:4	Reserved	Must be kept at reset value
3	CTSF	Clear timestamp flag Writing 1 in this bit clears the TSF bit in the RTC_SR register.
2	Reserved	Must be kept at reset value
1	CCALF	Clear calibration period flag Writing 1 in this bit clears the CALF bit in the RTC_SR register and the CALIF bit in the RTC_ISR register.
0	CALARMF	Clear alarm flag Writing 1 in this bit clears the ALARMF bit in the RTC_SR register and the ALARMIF bit in the RTC_ISR register.

28 Tamper and backup registers (TAMP)

28.1 Introduction

The tamper detection circuit and backup registers are in the V_{CORE_AON} power domain. Any tamper detection can generate a RTC timestamp event.

The backup registers consist 5 independent 32-bit registers, and the data stored will not be lost in various low-power modes. The backup registers are protected by the tamper detection and can be used to store sensitive information. When a tamper event is detected, the backup registers will be automatically erased.

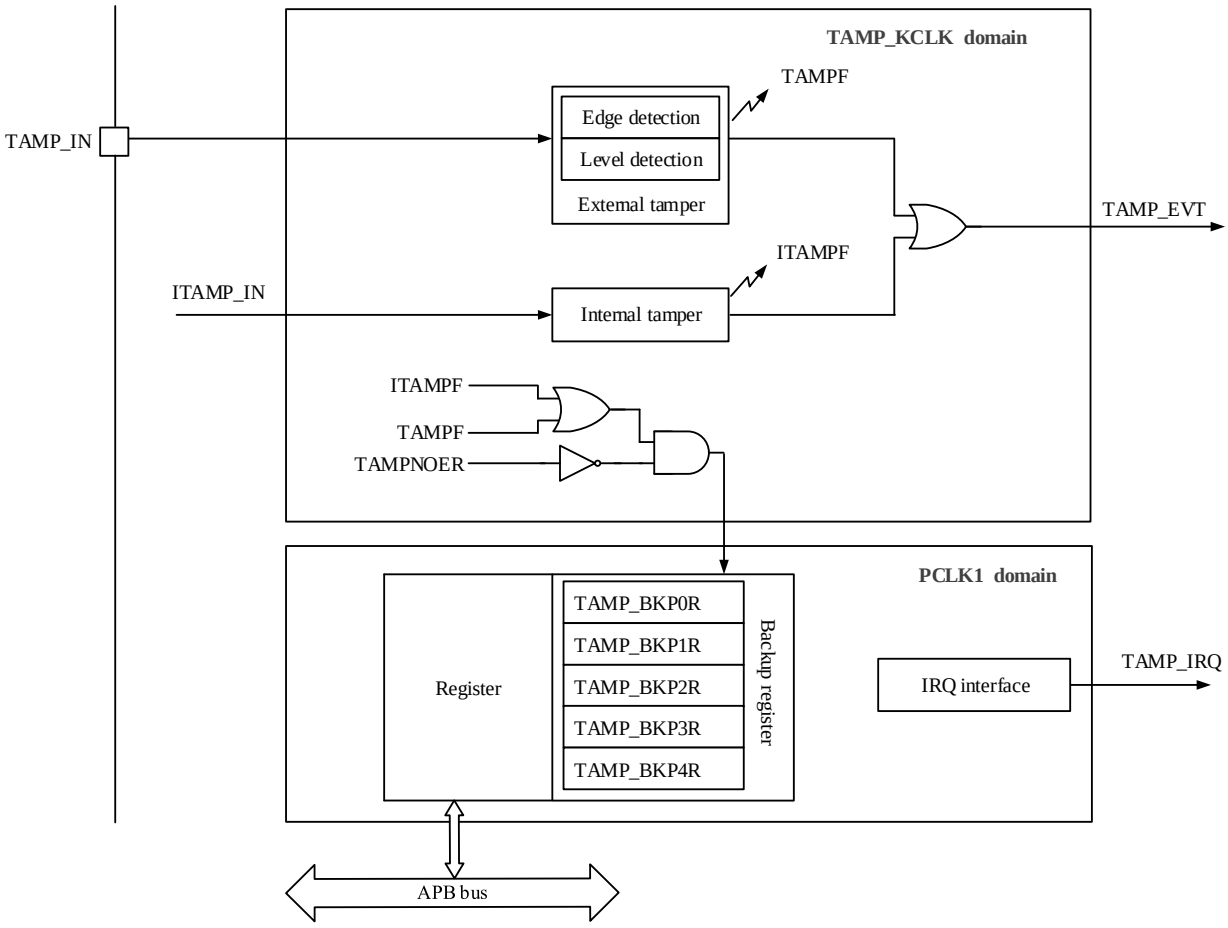
28.2 TAMP main features

- 1 external tamper detection event
- The external tamper pins can be configured for edge detection, or level detection.
- External tamper with configurable filter and internal pull-up.
- 1 internal tamper event source: the LXTAL CSS.
- 5 32-bit backup registers
- Any tamper detection can erase the backup registers
- Any tamper detection can generate a RTC timestamp event.

28.3 TAMP functional description

28.3.1 TAMP block diagram

Figure 28-1 TAMP block diagram



28.3.2 TAMP pins and internal signals

Table 28-1 TAMP input pins

Pin name	Signal type	Description
TAMP_IN	Input	Tamper input pin

By setting the TAMPEN bit in the TAMP enable register (TAMP_ENR) to 1, the TAMP_IN alternate function of the corresponding pin can be enabled, with the pin fixed in input mode. The pull-up resistor enable for the pin can be configured through the TAMPPUEN bit in the TAMP filter control register (TAMP_FLTCR).

RTC_OUT output function and external tamper event detection function share the same pin PC13. If both functions are enabled simultaneously, the pin will be used as the RTC_OUT output pin.

Table 28-2 TAMP internal input/output signals

Signal name	Type	Description
TAMP_KCLK	Input	TAMP kernel clock, connected to RTC_KCLK
PCLK1	Input	APB1 bus clock PCLK1
ITAMP_IN	Input	Internal tamper event sources
TAMP_EVT	Output	The tamp_evt is used to generate a RTC timestamp event
TAMP_IRQ	Output	TAMP interrupt

The TAMP kernel clock and the RTC kernel clock have the same source, and the clock source is selected by configuring the RTCSEL[1:0] bit field in the *V_{CORE_AON} domain control register (RCC_AWCR)*.

28.3.3 TAMP and backup register write protection

TAMP and backup registers are implemented in the VCORE_AON power domain. After reset, the TAMP registers and backup registers will be in a write-protected state, meaning that these registers can only be read but not written.

Before configuring the TAMP registers and backup registers, the VAON_WEN bit in the *Power control register 1 (PMU_CR1)* must be set to 1 to remove the write protection of the registers.

28.3.4 External tamper detection

By setting the TAMPEN bit in the enable register (TAMP_ENR) to 1, the tamper detection function of the corresponding external pin is enabled.

When an external tamper event is detected, the TAMPF flag in the status register (TAMP_SR) is set to 1. If the TAMPIE bit in the interrupt enable register (TAMP_IER) is already set to 1 at this time, the TAMPIF interrupt flag in the interrupt status register (TAMP_ISR) will also be set to 1, generating an interrupt.

By setting the corresponding CTAMPF bit in the status clear register (TAMP_ CLR), both the TAMPF and TAMPIF flags can be cleared simultaneously.

When an external tamper event is detected, the detection of new tamper events will stop until the TAMPF flag is cleared.

Edge detection mode

When the TAMPFLT[1:0] bit field in the filter control register (TAMP_FLTCR) are 00, the external tamper detection will operate in edge detection mode. The trigger edge

(rising or falling) of the pin is configured by the TAMPTRIG bit in the TAMP control register (TAMP_CR).

Level detection mode

When the TAMPFLT[1:0] bit field in the filter control register (TAMP_FLTCR) are not 0, the external tamper detection will operate in level detection mode with digital filtering, controlled by the TAMP control register. The TAMPTRIG bit in TAMP_CR configures the trigger level (high or low level) of the pin.

When the level detection mode is adopted, the internal digital filter is automatically enabled. The sample clock of the digital filter is TAMP_KCLK clock. According to the configuration of the TAMPFLT[1:0] bit field, an external tamper event is triggered only when a valid level is sampled for multiple consecutive TAMP_KCLK clock cycles.

TAMPFLT and TAMPTRIG must be configured when the external tamper detection enable control (TAMPEN bit) is set to 0.

28.3.5 Internal tamper detection

When the LXTAL clock failure is detected, an internal tamper detection event is triggered.

By setting the ITAMPEN bit in the TAMP enable register (TAMP_ENR) to 1, the internal tamper detection function can be enabled. When an internal tamper event is detected, the ITAMPF flag in the status register (TAMP_SR) is set to 1. If the ITAMPIE bit in the interrupt enable register (TAMP_IER) is already set to 1 at this time, the ITAMPIF interrupt flag in the interrupt status register (TAMP_ISR) will also be set to 1, generating an interrupt.

By writing a 1 to the corresponding CITAMPF bit in the status clear register (TAMP_CLR), both the ITAMPF and ITAMPIF flags can be cleared simultaneously.

28.3.6 Tamper triggering RTC timestamp

External or internal tamper detection events can trigger the RTC timestamp. When the TAMPF or ITAMPF flags in the TAMP_SR register are set to 1, the RTC timestamp function is triggered, recording the calendar information at the time the tamper event occurred. At this point, the timestamp flag TSF in the *RTC status register (RTC_SR)* is set to 1.

28.3.7 Backup registers

Backup registers are commonly used to store sensitive information in programs. System reset will not erase the data in the backup registers (TAMP_BKPxR), and the data in the backup registers will not be lost during low-power modes.

When an external or internal tamper event is detected, the data in the backup registers is automatically erased. By setting the TAMPNOER bit in the TAMP control register (TAMP_CR) to 1, this function is disabled.

If there are uncleared tamper detection flags TAMPF or ITAMPF, and the TAMPNOER bit in the TAMP_CR register is 0, then write operations to the backup registers will be ineffective, and the backup registers will retain their reset values.

The backup registers (TAMP_BKPxR) are also erased when the readout protection of the FLASH is changed from level 1 to level 0.

28.4 TAMP low-power modes

TAMP is functional in all power modes. In V_{BAT} mode, the kernel clock must be selected as LXTAL. No clock is needed for the external tamper edge detection to wake up from low-power modes.

28.5 TAMP interrupt

If the flag in the TAMP interrupt status register (TAMP_ISR) is set to 1, it indicates that the corresponding interrupt request has been generated.

Table 28-3 TAMP interrupt request

Interrupt event	Event flag	Enable control bit	Interrupt clear method
External tamper	TAMPIF	TAMPIE	Write 1 in CTAMPF
Internal tamper	ITAMPIF	ITAMPIE	Write 1 in CITAMPF

28.6 TAMP registers

The TAMP registers can only be accessed by words (32-bit).

Table 28-4 TAMP base address

Peripheral	Base address
TAMP	0x4000 B000

28.6.1 TAMP enable register (TAMP_ENR)

Address offset: 0x00

Reset value: 0x0000 0000

Note: Can only be reset by V_{CORE_AON} domain reset.

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

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

Bits	Name	Description
31:19	Reserved	Must be kept at reset value
18	ITAMPEN	Internal tamper enable A tamper is generated when LXTAL CSS. 0: Disable 1: Enable
17:1	Reserved	Must be kept at reset value
0	TAMPEN	Tamper detection on TAMP_IN enable 0: Disable 1: Enable

28.6.2 TAMP control register (TAMP_CR)

Address offset: 0x04

Reset value: 0x0000 0000

Note: Can only be reset by V_{CORE_AON} domain reset.

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

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

Bits	Name	Description
31:25	Reserved	Must be kept at reset value
24	TAMPTRIG	Active level for tamper input If TAMPFLT[1:0] ≠ 00, input level triggers a tamper detection 0: Low level triggers 1: High level triggers If TAMPFLT[1:0] = 00, input edge triggers a tamper detection 0: Rising edge triggers 1: Falling edge triggers
23:1	Reserved	Must be kept at reset value
0	TAMPNOER	Tamper no erase 0: Enable (Tamper event erases the backup registers) 1: Disable (Tamper event does not erases the backup registers)

28.6.3 TAMP filter control register (TAMP_FLTCR)

Address offset: 0x0C

Reset value: 0x0000 0000

Note: Can only be reset by V_{CORE_AON} domain reset.

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

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

Bits	Name	Description
31:8	Reserved	Must be kept at reset value
7	TAMPPUEN	Control the pull-up resistor enable for the external tamper detection pin 0: Disable 1: Enable
6:5	Reserved	Must be kept at reset value
4:3	TAMPFLT[1:0]	Digital filter control for the external intrusion detection pin These bits determine the number of consecutive samples at the specified level needed to activate a tamper event. 00: Edge detection mode (No digital filtering). 01: Level detection mode, Tamper event is activated after 2 consecutive samples at the active level. 10: Level detection mode, Tamper event is activated after 4 consecutive samples at the active level. 11: Level detection mode, Tamper event is activated after 8 consecutive samples at the active level.
2:0	Reserved	Must be kept at reset value

28.6.4 TAMP interrupt enable register (TAMP_IER)

Address offset: 0x2C

Reset value: 0x0000 0000

Note: Can only be reset by V_{CORE_AON} domain reset.

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

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

Bits	Name	Description
31:19	Reserved	Must be kept at reset value

18	ITAMPIE	Internal tamper interrupt enable 0: Disable 1: Enable
17:1	Reserved	Must be kept at reset value
0	TAMPIE	External tamper interrupt enable 0: Disable 1: Enable

28.6.5 TAMP status register (TAMP_SR)

Address offset: 0x30

Reset value: 0x0000 0000

Note: Can only be reset by V_{CORE_AON} domain reset.

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

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

Bits	Name	Description
31:19	Reserved	Must be kept at reset value
18	ITAMPF	Internal tamper flag This flag is set by hardware when LXTAL CSS.
17:1	Reserved	Must be kept at reset value
0	TAMPF	External tamper flag This flag is set by hardware when an external tamper event is detected.

28.6.6 TAMP interrupt status register (TAMP_ISR)

Address offset: 0x34

Reset value: 0x0000 0000

Note: Can only be reset by V_{CORE_AON} domain reset.

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

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

Bits	Name	Description
31:19	Reserved	Must be kept at reset value
18	ITAMPIF	Internal tamper interrupt flag This flag is set by hardware if ITAMPIE is enabled when LXTAL CSS, triggering an interrupt.
17:1	Reserved	Must be kept at reset value
0	TAMPIF	External tamper interrupt flag This flag is set by hardware if TAMPIE is enabled when an external tamper event is detected, triggering an interrupt.

28.6.7 TAMP status clear register (TAMP_CLR)

Address offset: 0x3C

Reset value: 0x0000 0000

Note: Can only be reset by V_{CORE_AON} domain reset.

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

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

Bits	Name	Description
31:19	Reserved	Must be kept at reset value

18	CITAMPF	Clear internal tamper flag Writing 1 in this bit clears the ITAMPF and ITAMPIF.
17:1	Reserved	Must be kept at reset value
0	CTAMPF	Clear external tamper flag Writing 1 in this bit clears the TAMPF and TAMPIF.

28.6.8 TAMP backup x register (TAMP_BKPxR)

Address offset: 0x100 + 0x04 × x (x = 0 to 4)

Reset value: 0x0000 0000

Note: Can only be reset by V_{CORE_AON} domain reset.

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	rw	rw	rw

Bits	Name	Description
31:0	BKP[31:0]	Backup registers The contents remain valid when the device operates in low-power mode. System reset will not erase the data in the backup registers. Refer to: *Backup Registers.*

29 Inter-integrated circuit interface (I2C)

29.1 Introduction

The I2C peripheral interface handles communications between the microcontroller and the serial I2C bus. It supports Standard-mode, Fast-mode and Fast-mode Plus. It provides multimaster capability and controls all I2C bus-specific sequencing, protocol, arbitration and timing.

29.2 I2C main features

- Master mode, slave mode, and multimaster capability
- Standard-mode (up to 100 KHz), Fast-mode (up to 400 KHz), Fast-mode Plus (up to 1 MHz)
- 7-bit and 10-bit addressing mode
- Multiple 7-bit slave addresses (2 address registers, 1 with configurable mask)
- Wakeup from Stop mode on address match
- General call
- Programmable setup and hold times
- Optional clock stretching
- DMA transfer support
- Programmable digital noise filters

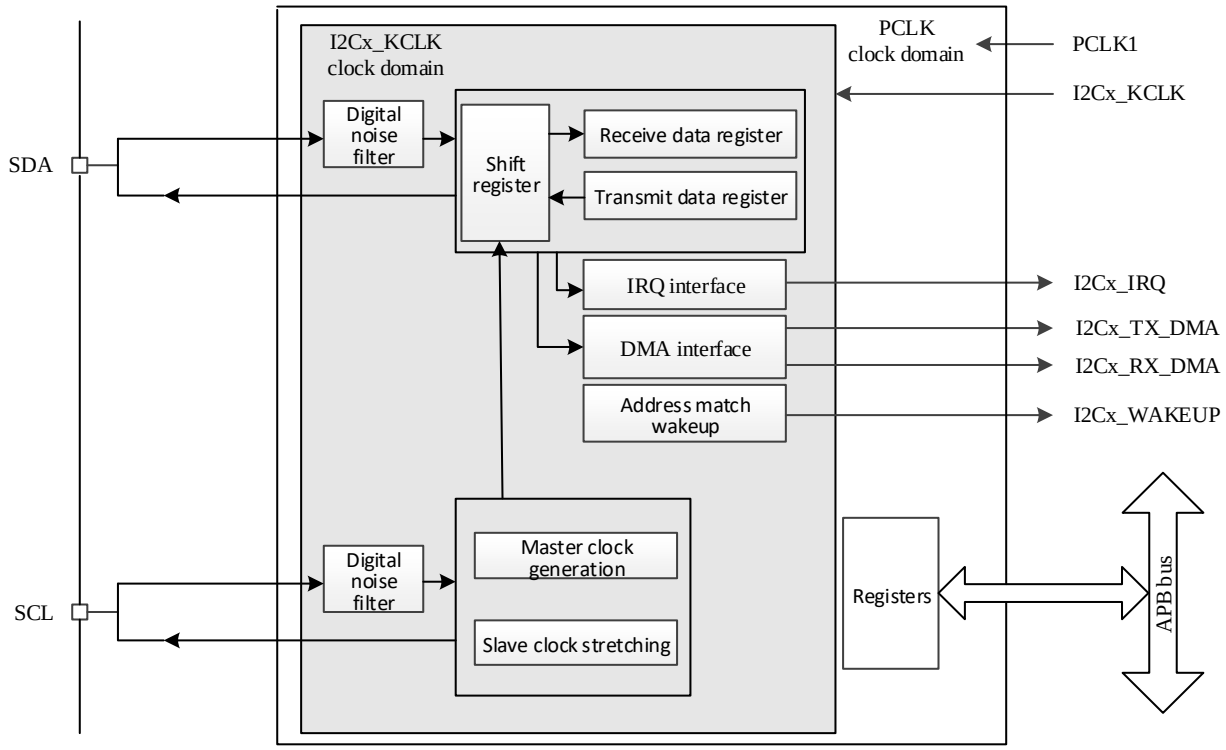
29.3 I2C functional description

The interface is connected to the I2C bus by a data pin (SDA) and by a clock pin (SCL).

29.3.1 I2C block diagram

The block diagram of I2C interface is shown in the following figure.

Figure 29-1 I2Cx block diagram (x = 1)



I2Cx_KCLK clock can be configured in the *Peripherals independent clock configuration register (RCC_CLKSEL)*, using the I2Cx_SEL[1:0] bit field, to select PCLK, SYSCLK, or RCH. The wakeup from Stop mode function is only available when the I2C clock source is RCH. Clock stretching function (NOSTRETCH bit is 0) must be enabled to ensure that an address match can wake up from Stop mode.

29.3.2 I2C pins and internal signals

Table 29-1 I2C input/output pins

Pin name	Signal type	Description
SCL	Bidirectional	I2C clock
SDA	Bidirectional	I2C data

Table 29-2 I2C internal signals

Internal signal name	Signal type	Description
I2Cx_KCLK	Input	I2C kernel clock
PCLK1	Input	I2C APB clock
I2Cx_IRQ	Output	I2C interrupts
I2Cx_TX_DMA	Output	I2C transmit Data DMA request
I2Cx_RX_DMA	Output	I2C receive Data DMA request
I2Cx_WAKEUP	Output	I2C wakeup signal

29.3.3 I2C clock requirements

The I2C kernel is clocked by I2Cx_KCLK, the I2Cx_KCLK period t_{I2Cx_KCLK} must respect the following conditions:

$t_{I2Cx\_KCLK} < (t_{LOW} - t_{FILTERS}) / 8$ and $t_{I2Cx\_KCLK} < t_{HIGH}$

t_{LOW}: SCL low time.

t_{HIGH}: SCL high time.

$t_{FILTERS}$: When enabled, sum of delays brought by the digital filter (digital filter delay is DNF × t_{I2Cx_KCLK}).

29.3.4 Mode selection

The interface can operate in one of the four following modes:

- Slave transmitter
- Slave receiver
- Master transmitter
- Master receiver

By default, it operates in slave mode. The interface automatically switches from slave to master when it generates a START condition, and from master to slave if an arbitration loss or a STOP generation occurs, allowing multimaster capability.

Communication flow

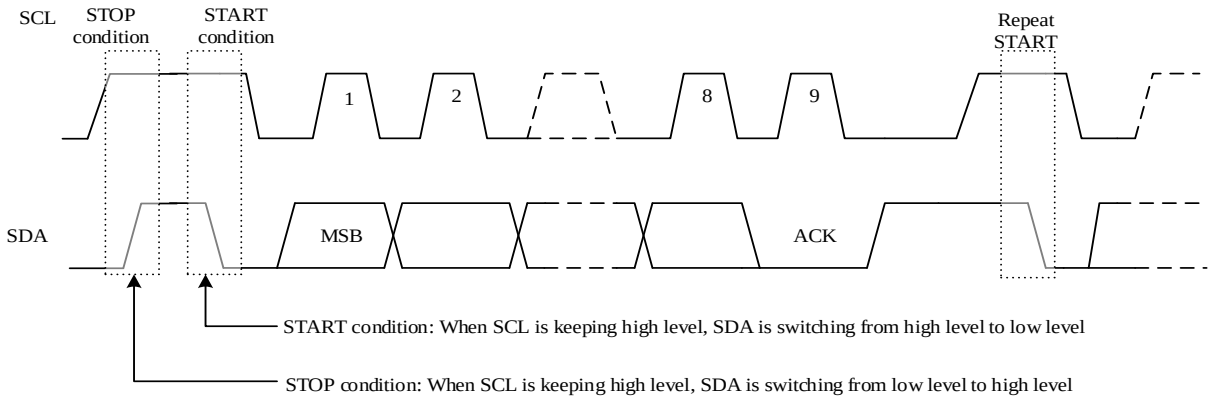
In master mode, the I2C interface initiates a data transfer and generates the clock signal. A serial data transfer always begins with a START condition and ends with a STOP condition.

In slave mode, the interface is capable of recognizing its own address (7 or 10-bit), and the general call address. The General Call address detection can be enabled or disabled by software.

Data and addresses are transferred as 8-bit bytes, MSB first. The first bytes following the START condition contain the address (one in 7-bit mode, two in 10-bit mode). The address is always transmitted in master mode.

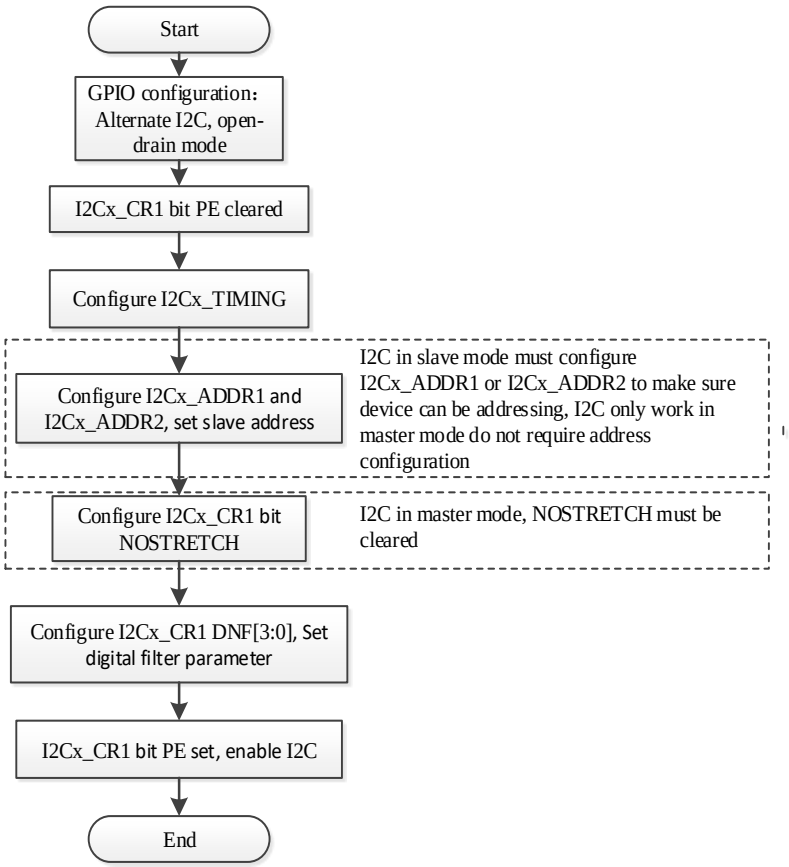
A 9th clock pulse follows the 8 clock cycles of a byte transfer, during which the receiver must send an acknowledge bit to the transmitter.

Figure 29-2 I2C bus protocol



29.3.5 I2C initialization

Figure 29-3 I2C initial flow



Enabling and disabling the peripheral

The I2Cx_KCLK clock must be configured and enabled in the clock controller, then the I2C can be enabled by setting the PE bit in the I2Cx_CR1 register.

When the I2C is disabled (PE = 0), the I2C performs a software reset. Refer to *Software reset* for more details.

Noise filters

I2C provides configurable digital noise filters for input signals filtering.

- Digital filters

  By default, the SDA and SCL digital noise filters are turned off, and the digital filters are enabled by configuring the DNF[3:0] bits in the I2Cx_CR1 register. When the digital filter is enabled, the level of the SCL or the SDA line is internally changed only if it remains stable for more than DNF × I2Cx_KCLK periods. This allows spikes with programmable length of 1 to 15 I2Cx_KCLK periods to be suppressed.

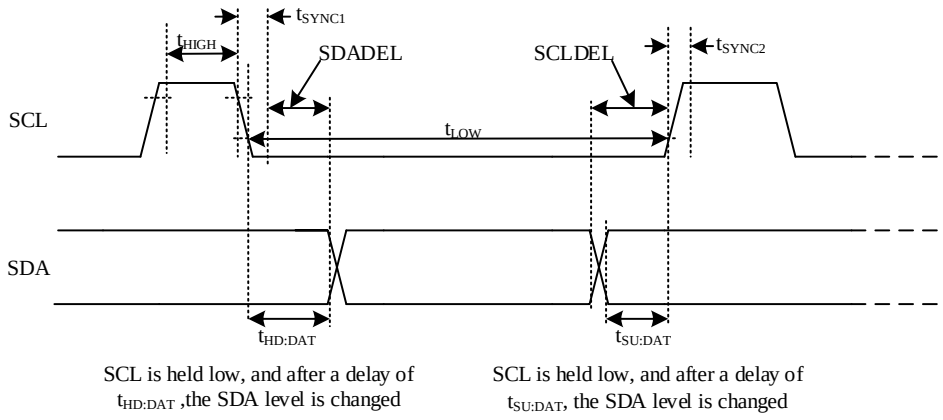
I2C timings

The timings must be configured in order to guarantee a correct data hold and setup time, used in master and slave mode. This is done by programming the PRESC[3:0], SCLDEL[3:0] and SDADEL[3:0] bits in the I2Cx_TIMING register. Additionally, in master mode, the SCL clock high and low levels must be configured by programming the PRESC[3:0], SCLH[7:0] and SCLL[7:0] bits in the I2Cx_TIMING register.

Table 29-3 I2C specific timing requirement

Symbol	Parameter	Standard-mode		Fast-mode		Fast-mode plus		Unit
		Min	Max	Min	Max	Min	Max	
$t_{HD:DAT}$	Data hold time	0	-	0	-	0	-	µs
$t_{VD:DAT}$	Data valid time	-	3.45	-	0.9	-	0.45	µs
$t_{SU:DAT}$	Data setup time	250	-	100	-	50	-	ns
$t_{HD:STA}$	(Repeat) START hold time	4.0	-	0.6	-	0.26	-	µs
$t_{SU:STA}$	START setup time	4.7	-	0.6	-	0.26	-	µs
$t_{SU:STO}$	STOP setup time	4.0	-	0.6	-	0.26	-	µs
t_{BUF}	Idle time between START and STOP	4.7	-	1.3	-	0.5	-	µs
t_{HIGH}	SCL high level time	4.0	-	0.6	-	0.26	-	µs
t_{LOW}	SCL low level time	4.7	-	1.3	-	0.5	-	µs
t_r	Rising edge time	-	1000	-	300	-	120	ns
t_f	Falling edge time	-	300	-	300	-	120	ns

Figure 29-4 I2Cx_TIMING register configuration generates SCL timing



When the SCL falling edge is internally detected, a delay is inserted before sending

SDA output. This delay is t_{SDADEL} = SDADEL × t_{PRESC} + 4 × t_{I2Cx_KCLK} (t_{PRESC} = (PRESC + 1) × t_{I2Cx_KCLK}), t_{SDADEL} impacts the hold time $t_{HD:DAT}$, refer to *Figure: I2Cx_TIMING register configuration generates SCL timing*. The total SDA output delay is: t_{SYNC1} + t_{SDADEL}. t_{SYNC1} and t_{SYNC2} depend on these parameters:

- SCL falling slope
- When enabled, input delay brought by the digital filter t_{DNF} = DNF × t_{I2Cx_KCLK}
- Delay due to SCL synchronization to I2Cx_KCLK

User should follow the following conditions when programming SDADEL:

$\{t_{f(max)} + t_{HD:DAT(min)} - [(DNF + 1) \times t_{I2Cx\_KCLK}]\} / t_{PRESC} \leq$ SDADEL

SDADEL $\leq \{t_{HD:DAT(max)} - [(DNF + 2) \times t_{I2Cx\_KCLK}]\} / t_{PRESC}$

The max value of $t_{HD:DAT}$ in Standard-mode, Fast-mode and Fast-mode Plus are 3.45 µs, 0.9 µs and 0.45 µs, but it must be less than the maximum value of $t_{VD:DAT}$. This maximum must only be met if the device does not stretch the low period of the SCL signal. If the clock stretches the SCL, the data must be valid by the set-up time before it releases the clock. In this case, the previous equation becomes:

SDADEL $\leq \{t_{VD:DAT(max)} - t_{r(max)} - [(DNF + 2) \times t_{I2Cx\_KCLK}]\} / t_{PRESC}$

After t_{SDADEL} delay, or after sending SDA output in case the slave had to stretch the clock because the data was not yet written in the I2Cx_TDR register, SCL line is kept at low level during the setup time. The setup time is t_{SCLDEL} = (SCLDEL + 1) × t_{PRESC} + t_{I2Cx_KCLK} (t_{PRESC} = (PRESC + 1) × t_{I2Cx_KCLK}), t_{SCLDEL} impacts the hold time $t_{SU:DAT}$, refer to *Figure: I2Cx_TIMING register configuration generates SCL timing*. User should follow the following conditions when programming:

$[(t_{r(max)} + t_{SU:DAT(min)}) / t_{PRESC}] - 1 \leq$ SCLDEL

If NOSTRETCH = 1 in slave mode, the SCL is not stretched. Consequently, the SDADEL must be programmed in such way to guarantee also a sufficient setup time.

By configuring SCLH[7:0] bits to set SCL high level time: When the SCL rising edge is internally detected, SCL output will be forced to high level, the time of high level delay is t_{SCLH} = (SCLH + 1) × t_{PRESC} + 3 × t_{I2Cx_KCLK} (t_{PRESC} = (PRESC + 1) × t_{I2Cx_KCLK}), t_{SCLH} impacts the SCL high time t_{HIGH}, refer to *Figure: I2Cx_TIMING register configuration generates SCL timing*.

By configuring SCLL[7:0] bits to set SCL low level time: When the SCL falling edge is internally detected, SCL output will be forced to low level, the time of low level

delay is $t_{SCLL} = (SCLL + 1) \times t_{PRESC} + 4 \times t_{I2Cx\_KCLK}$ ($t_{PRESC} = (PRESC + 1) \times t_{I2Cx\_KCLK}$), t_{SCLL} impacts the SCL low time t_{LOW}, refer to *Figure: I2Cx_TIMING register configuration generates SCL timing*.

I2Cx_TIMING register configuration example

The following table provides the value configured during the initialization of the register I2Cx_TIMING to set I2C different transfer bit rates.

Table 29-4 I2Cx_KCLK = 16 MHz configuration example

Parameter	100 KHz	400 KHz
PRESC[3:0]	0	0
t_{PRESC}	62.5 ns	62.5 ns
SCLDEL[3:0]	0x03	0x01
t_{SCLDEL}	4 × 62.5ns = 250ns	2 × 62.5 ns = 125 ns
SDADEL[3:0]	0x02	0x02
t_{SDADEL}	2 × 62.5ns = 125ns	2 × 62.5 ns = 125 ns
SCLH[7:0]	0x3D	0x10
t_{SCLH}	62 × 62.5 ns = 3875 ns	17 × 62.5 ns = 1062.5 ns
SCLL[7:0]	0x5B	0x12
t_{SCLL}	92 × 62.5 ns = 5750 ns	19 × 62.5 ns = 1187.5 ns

Address mode

I2C supports 7-bit and 10-bit addressing modes, two addressing modes are compatible on the bus.

Figure 29-5 7-bit address communications

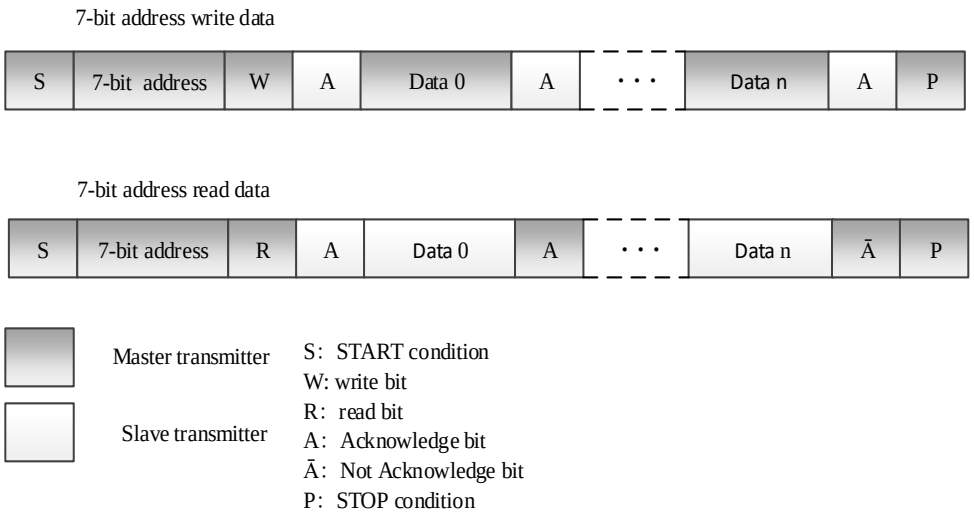
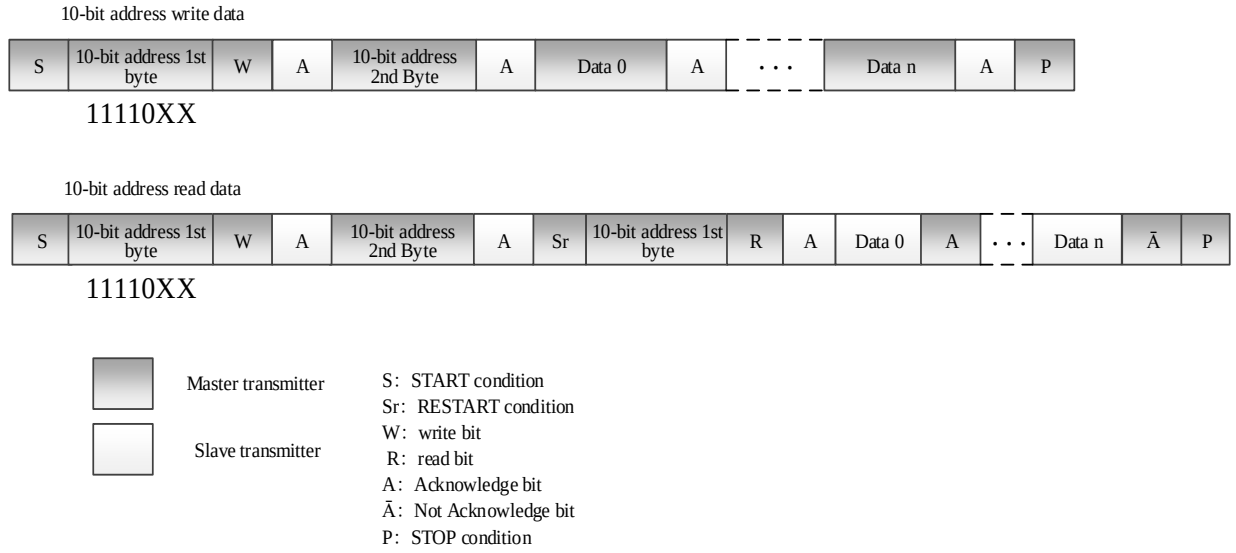


Figure 29-6 10-bit address communications



ADDR1 can be configured either in 7-bit mode or in 10-bit addressing mode. When ADDR1MODE is 1, ADDR1[9:0] is the 10-bit address for slave mode. When ADDR1MODE is 0, ADDR1[7:1] is the 7-bit address for slave mode.

ADDR2 is always a 7-bit address. User configure additional slave 7-bit address through ADDR2[7:1] bits in the I2Cx_ADDR2 register. Up to 7 ADDR LSB can be masked by configuring ADDRMSK[2:0] bits in the I2Cx_ADDR2 register. As soon as ADDRMSK[2:0] bits are not equal to 0, the address comparator for ADDR2 excludes the I2C reserved addresses (0b0000xxx and 0b1111xxx), which are not acknowledged.

The General Call address is enabled by setting the GCEN bit in the I2Cx_CR1 register.

Clock stretching low level

- Slave clock stretching (NOSTRETCH = 0)

  In default mode, the I2C slave stretches the SCL clock in the following situations:

  When the ADDR flag is set: the received address matches with one of the enabled slave addresses. This stretch is released when the ADDR flag is cleared by software setting the ADDRCF bit.

  In transmission, if the previous data transmission is completed and no new data is written in the I2Cx_TDR register, or if the first data byte is not written when the ADDR flag is cleared (TXE = 1). This stretch is released when the data is written to the I2Cx_TDR register.

  In reception when the I2Cx_RDR register is not read yet and a new data

reception is completed. This stretch is released when the I2Cx_RDR register is read.

When TCR = 1 in slave byte control mode, reload mode (SBC = 1 and RELOAD = 1), meaning that the last data byte has been transferred. This stretch is released when then TCR is cleared by writing a non-zero value in the NBYTES[7:0] bits.

- Slave without clock stretching (NOSTRETCH = 1)

  When NOSTRETCH = 1 in the I2Cx_CR1 register, the I2C slave does not stretch SCL signal.

  The SCL clock is not stretched while the ADDR flag is set.

29.3.6 Software reset

A software reset can be performed by clearing the PE bit in the I2Cx_CR1 register. PE must be kept low during at least 3 APB clock cycles in order to perform the software reset. This is ensured by writing the following software sequence: write PE = 0, check PE = 0, write PE = 1. In that case, internal states machines are reset and communication control bits, as well as status bits come back to their reset value. The configuration registers are not impacted. Here is the list of impacted register bits:

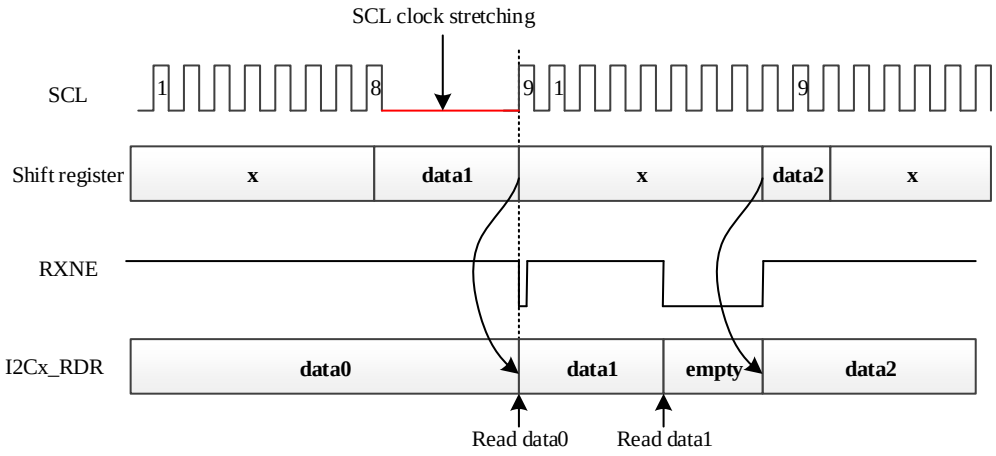
- I2Cx_CR2 register: START, STOP, and NACK.
- I2Cx_ISR register: BUSY, TXE, TXIS, RXNE, ADDR, DIR, NACKF, TCR, TC, STOPF, BERR, ARLO, and OVR.

29.3.7 Data transfer

Reception

The SDA input fills the shift register. After the 8th SCL pulse (when the complete data byte is received), the shift register is copied into the I2Cx_RDR register if it is empty (RXNE = 0). If RXNE = 1, meaning that the previous received data byte has not yet been read, the SCL line is stretched low until I2Cx_RDR is read. The stretch is inserted between the 8th and 9th SCL pulse (before the acknowledge pulse).

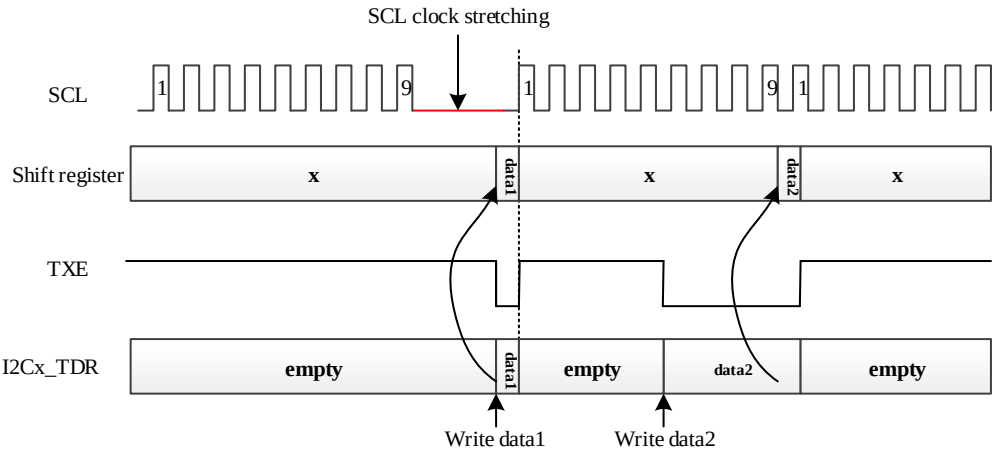
Figure 29-7 Data reception



Transmission

If the I2Cx_TDR register is not empty (TXE = 0), its content is copied into the shift register after the 9th SCL pulse (the acknowledge pulse). Then the shift register content is shifted out on SDA line. If TXE = 1, meaning that no data is written yet in I2Cx_TDR, SCL line is stretched low until I2Cx_TDR is written. The stretch is done after the 9th SCL pulse.

Figure 29-8 Data transmission



Hardware transfer management

The I2C has a byte counter embedded in hardware in order to manage byte transfer and to close the communication in master and slave mode.

- Master mode
  - Send NACK when received the last byte of data
  - Generate STOP condition after receiving or sending data

- After data phase, the SCL clock is being stretched low, and configure START bit to generate RESTART condition.

- Slave mode

  The byte counter is always used in master mode. By default, it is disabled in slave mode, but it can be enabled by software by setting the SBC (Slave Byte Control) bit in the I2Cx_CR2 register.

  - After receiving each byte, stretch the SCL clock to send an ACK or NACK, which is configurable through the NACK bit

The number of bytes to be transferred is programmed in the NBYTES[7:0] bit field in the I2Cx_CR2 register. If the number of bytes to be transferred (NBYTES) is greater than 255, the reload mode must be selected by setting the RELOAD bit in the I2Cx_CR2 register. In this mode, TCR flag is set when the number of bytes programmed in NBYTES has been transferred. SCL is stretched as long as TCR flag is set. TCR is cleared by software when NBYTES is written to a non-zero value.

If slave receiver wants to control the acknowledge value of a received data byte, it is needed to configure reload mode (RELOAD = 1) after the address is matched, and configure NBYTES[7:0] = 1. After clearing the ADDR flag, wait for TCR flag to be set. At this time, set the NACK bit and configure the NBYTES[7:0] = 1 in the I2Cx_CR2 register to release SCL line, and the I2C peripheral will send NACK when received the current byte of data.

Note: The slave byte control mode (SBC = 1) is incompatible with NOSTRETCH = 1. Setting SBC to 1 is not allowed when NOSTRETCH = 1.

When master automatic end mode is enabled (AUTOEND = 1 and RELOAD = 0 in the I2Cx_CR2 register), the master automatically sends a stop condition once the number of bytes programmed in the NBYTES[7:0] bit field has been transferred.

When master software end mode is enabled (AUTOEND = 0 and RELOAD = 0 in the I2Cx_CR2 register), software action is expected once the number of bytes programmed in the NBYTES[7:0] bits have been transferred. The TC flag is set and an interrupt is generated if the TCIE bit is set. The SCL signal is stretched as long as the TC flag is set. The TC flag is cleared by software when the START or STOP bit is set in the I2Cx_CR2 register. This mode must be used when the master wants to send a RESTART condition.

29.3.8 I2C slave mode

Slave transmitter

The TXIS flag is set when the I2Cx_TDR register becomes empty. The TXIS bit is cleared when the I2Cx_TDR register is written with the next data byte to be transmitted.

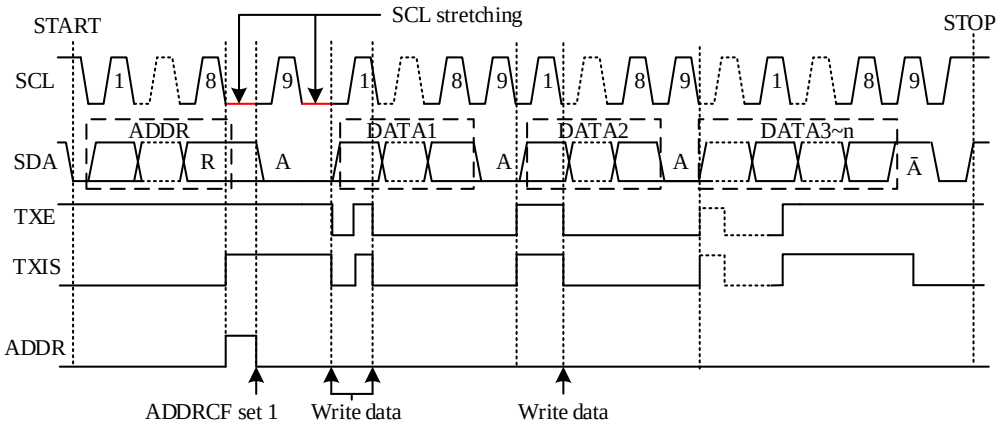
When a NACK is received, the NACKF bit is set in the I2Cx_ISR register and the slave automatically releases the SCL and SDA lines in order to let the master perform a STOP or a RESTART condition. The TXIS bit is not set when a NACK is received.

When a STOP is received and the slave transmission flow will be stopped.

If TXE = 0 when the slave address is received (ADDR = 1), the user can choose either to send the content of the I2Cx_TDR register as the first data byte, or to flush the I2Cx_TDR register by setting the TXE bit in order to program a new data byte.

When clock stretching is enabled (NOSTRETCH = 0), the data transmission sequence diagram is shown below:

Figure 29-9 Slave transmitter and NOSTRETCH = 0

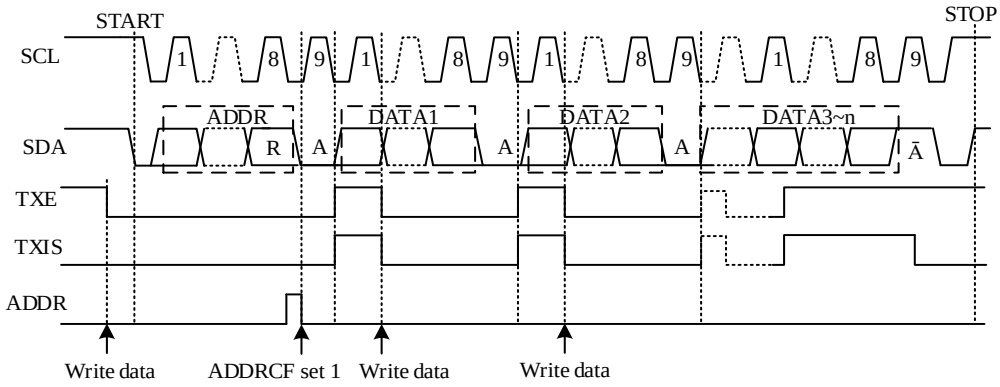


The software operation flow is as follows:

1) Wait for ADDR to be set, check the value of the ADDRCODE[6:0] and the DIR bit in order to confirm the matched address the transmission direction, set ADDRCF to 1 to clear the ADDR flag.
2) Wait for TXIS to be set, write DATA1 to the I2Cx_TDR register.
3) Repeat step 2, sequentially write DATA2~DATAn to the I2Cx_TDR register.
4) Wait for STOPF to be set, clear the STOPF flag, and end the data transmission.

When clock stretching is disabled (NOSTRETCH = 1), the data transmission sequence diagram is shown below:

Figure 29-10 Slave transmitter and NOSTRETCH = 1



The software operation flow is as follows:

1) Wait for TXE to be set, write DATA1 to the I2Cx_TDR register.
2) Wait for ADDR to be set, check the value of the ADDRCODE[6:0] field and the DIR bit in order to confirm the matched address the transmission direction, set ADDRCF to 1 to clear the ADDR flag.
3) Wait for TXIS to be set, write DATA2 to the I2Cx_TDR register.
4) Repeat step 3, sequentially write DATA3~DATAn to the I2Cx_TDR register.
5) Wait for STOPF to be set, clear the STOPF flag, and end the data transmission.

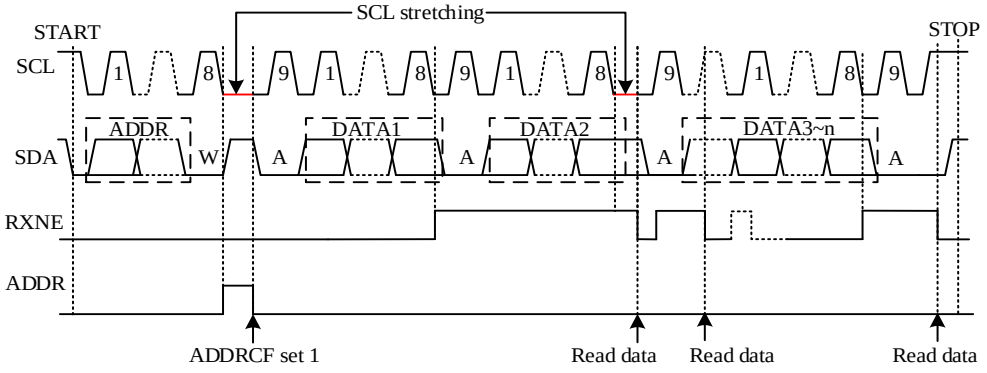
Slave receiver

The RXNE flag is set in the I2Cx_ISR register when the I2Cx_RDR is full, and RXNE is cleared when I2Cx_RDR is read.

When a STOP is received and the slave receiving flow will be stopped.

When clock stretching is enabled (NOSTRETCH = 0), the data receiving sequence diagram is shown below:

Figure 29-11 Slave receiver and NOSTRETCH = 0

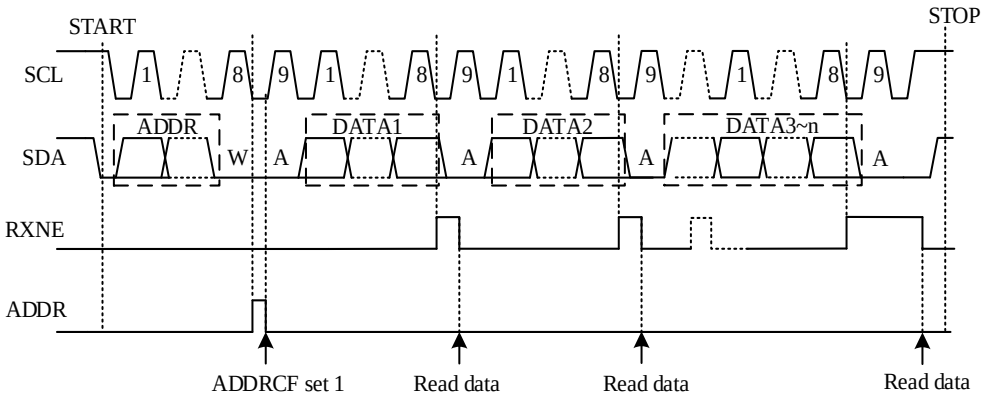


The software operation flow is as follows:

1) Wait for ADDR to be set, check the value of the ADDRCODE[6:0] field and the DIR bit in order to confirm the matched address the transmission direction, set ADDRCF to 1 to clear the ADDR flag.
2) Wait for RXNE to be set, read DATA1 from the I2Cx_RDR register.
3) Repeat step 2, sequentially read DATA2~DATAn from the I2Cx_RDR register.
4) Wait for STOPF to be set, clear the STOPF flag, and end the data receiving sequence.

When clock stretching is disabled (NOSTRETCH = 1), the data receiving sequence diagram is shown below:

Figure 29-12 Slave receiver and NOSTRETCH = 1



The software operation flow is as follows:

1) Wait for ADDR to be set, check the value of the ADDRCODE[6:0] field and the DIR bit in order to confirm the matched address the transmission direction, set ADDRCF to 1 to clear the ADDR flag.
2) Wait for RXNE to be set, read Data1 from the I2Cx_RDR register.
3) Repeat step 2, sequentially read DATA 2~DATAn from the I2Cx_RDR register.
4) Wait for STOPF to be set, clear the STOPF flag, and end the data receiving sequence.

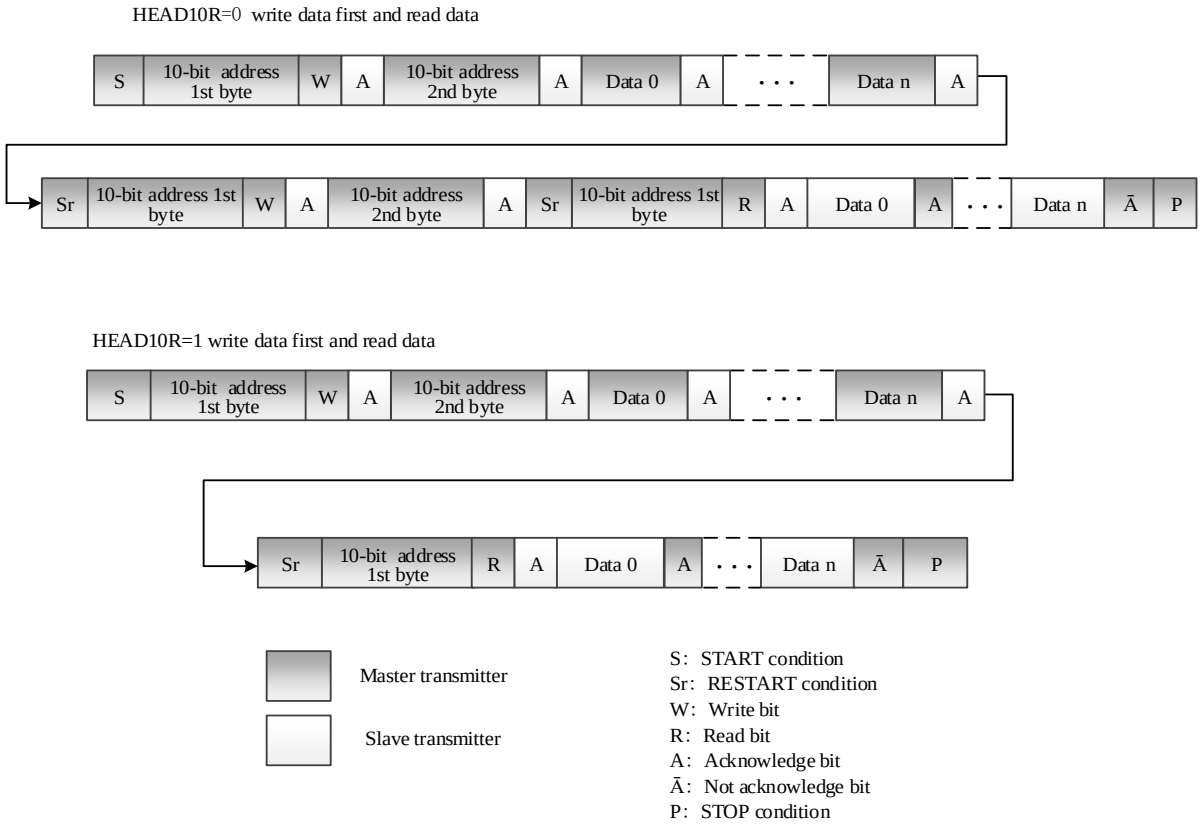
29.3.9 I2C master mode

In order to initiate the communication, the user must program the following parameters for the addressed slave in the I2Cx_CR2 register:

- Addressing mode (7-bit or 10-bit): ADDR10.
- Slave address to be sent: SADDR[9:0].

- Transfer direction: RD_WRN.
- In case of 10-bit address read: HEAD10R bit must be configured to indicate if the complete address sequence must be sent, or only the header in case of a direction change.

Figure 29-13 HEAD10R set or cleared in case of a direction change in 10-bit addressing mode



- The number of bytes to be transferred: NBYTES[7:0].

Master transmitter

In the case of a write transfer, the TXIS flag is set after each byte transmission, after the 9th SCL pulse when an ACK is received. The flag is cleared when the I2Cx_TDR register is written with the next data byte to be transmitted.

- When RELOAD = 0 and NBYTES[7:0] data have been transferred:

  In automatic end mode (AUTOEND = 1), a STOP is automatically sent.

  In software end mode (AUTOEND = 0), the TC flag is set and SCL is stretched low in order to perform software actions:

  - A RESTART condition can be requested by setting the START bit in the I2Cx_CR2 register with the proper slave address configuration, and a

number of bytes to be transferred. Setting the START bit clears the TC flag and the START condition is sent on the bus.

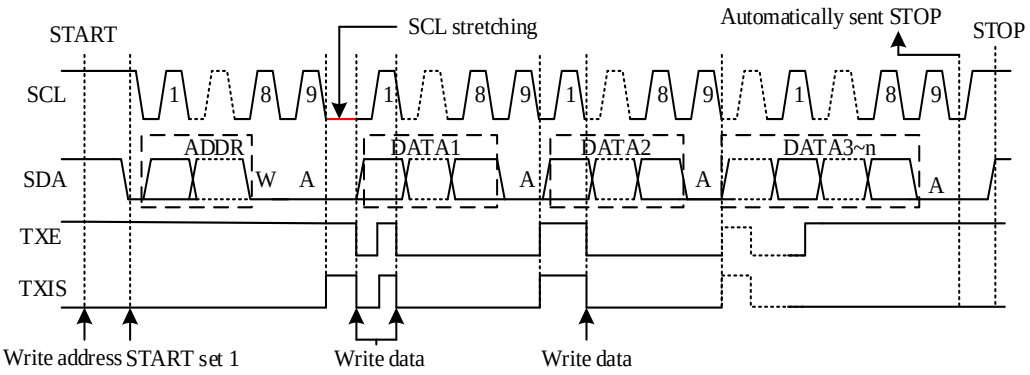
- A STOP condition can be requested by setting the STOP bit in the I2Cx_CR2 register. Setting the STOP bit clears the TC flag and the STOP condition is sent on the bus.

- If a NACK is received

  The TXIS flag is not set, and a STOP condition is automatically sent after the NACK reception. the NACKF flag is set in the I2Cx_ISR register, and an interrupt is generated if the NACKIE bit is set.

When automatic end mode is enabled (AUTOEND = 1), the data transmission sequence diagram is shown below:

Figure 29-14 Master transmitter and AUTOEND = 1

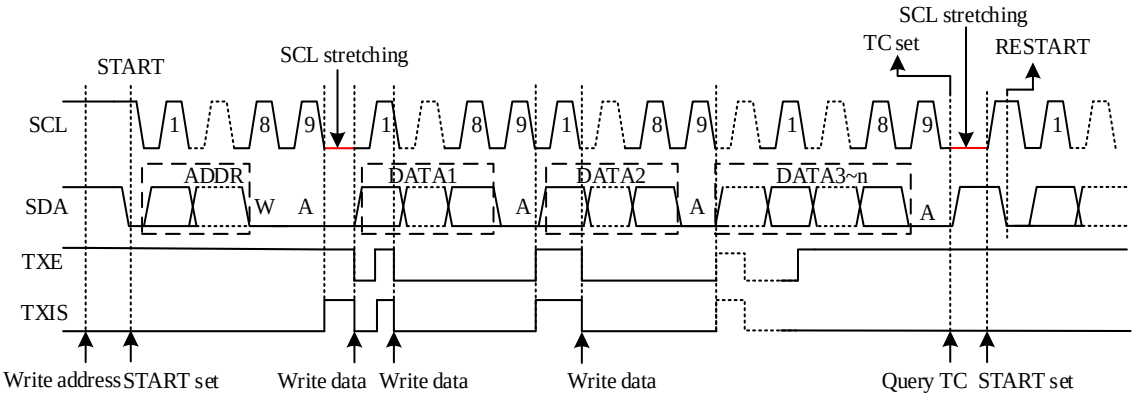


The software operation flow is as follows:

1) Configure the slave address and automatic end mode (AUTOEND = 1), and write the number of bytes to be transferred in NBYTES[7:0].
2) Set the START bit to send START condition.
3) Wait for TXIS to be set, write DATA1 to the I2Cx_TDR register.
4) Repeat step 3, sequentially write DATA2~DATAn to the I2Cx_TDR register.
5) Wait for STOPF to be set, clear the STOPF flag, and end the data transmission.

When software end mode is enabled (AUTOEND = 0), the data transmission sequence diagram is shown below:

Figure 29-15 Master transmitter and AUTOEND = 0



The software operation flow is as follows:

1) Configure the slave address and software end mode (AUTOEND = 0), and write the number of bytes to be transferred in NBYTES[7:0].
2) Set the START bit to send START condition.
3) Wait for TXIS to be set, write DATA1 to the I2Cx_TDR register.
4) Repeat step 3, sequentially write DATA2~DATAn to the I2Cx_TDR register.
5) Wait for TC to be set, set the START bit to send RESTART for next data transfer.

Master receiver

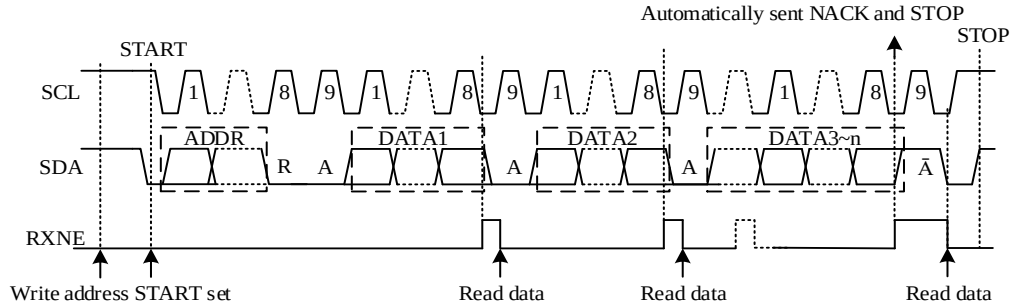
In the case of a read transfer, the RXNE flag is set after each byte reception, after the 8th SCL pulse. The flag is cleared when I2Cx_RDR is read.

When RELOAD = 0 and NBYTES[7:0] data have been transferred:

- In automatic end mode (AUTOEND = 1), a NACK and a STOP are automatically sent after the last received byte.
- In software end mode (AUTOEND = 0), a NACK is automatically sent after the last received byte, the TC flag is set and the SCL line is stretched low in order to allow software actions:
  - A RESTART condition can be requested by setting the START bit in the I2Cx_CR2 register with the proper slave address configuration, and number of bytes to be transferred. Setting the START bit clears the TC flag and the START condition, followed by slave address, are sent on the bus.
  - A STOP condition can be requested by setting the STOP bit in the I2Cx_CR2 register. Setting the STOP bit clears the TC flag and the STOP condition is sent on the bus.

When automatic end mode is enabled (AUTOEND = 1), the data receiving sequence diagram is shown below:

Figure 29-16 Master receiver and AUTOEND = 1

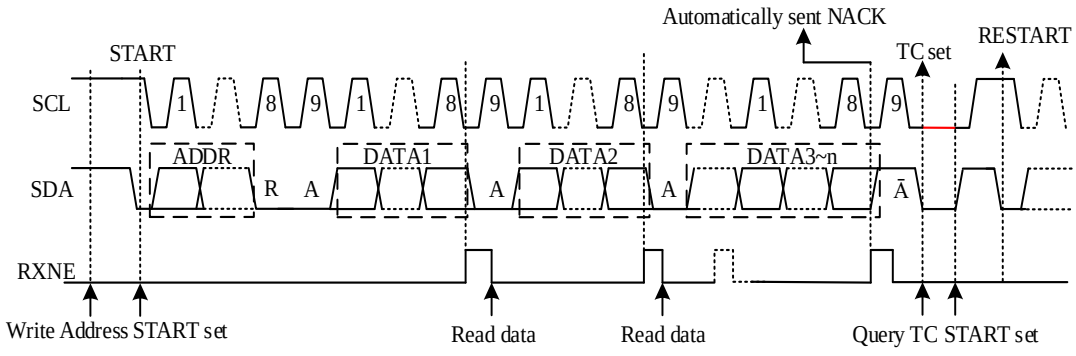


The software operation flow is as follows:

1) Configure the slave address and software end mode (AUTOEND = 1), and write the number of bytes to be received in NBYTES[7:0].
2) Set the START bit to send START condition.
3) Wait for RXNE to be set, read DATA1 from the I2Cx_RDR register.
4) Repeat step 3, sequentially read DATA2~DATAn from the I2Cx_RDR register.
5) Wait for STOPF to be set, clear the STOPF flag, and end the data receiving sequence.

When software end mode is enabled (AUTOEND = 0), the data receiving sequence diagram is shown below:

Figure 29-17 Master receiver and AUTOEND = 0



The software operation flow is as follows:

1) Configure the slave address and software end mode (AUTOEND = 0), and write the number of bytes to be received in NBYTES[7:0].
2) Set the START bit to send START condition.
3) Wait for RXNE to be set, read DATA1 from the I2Cx_RDR register.

4) Repeat step 3, sequential read DATA2~DATA[n] from the I2Cx_RDR register.
5) Wait for TC to be set, set the START bit to send RESTART for next data transfer.

29.3.10 Wakeup from Stop mode on address match

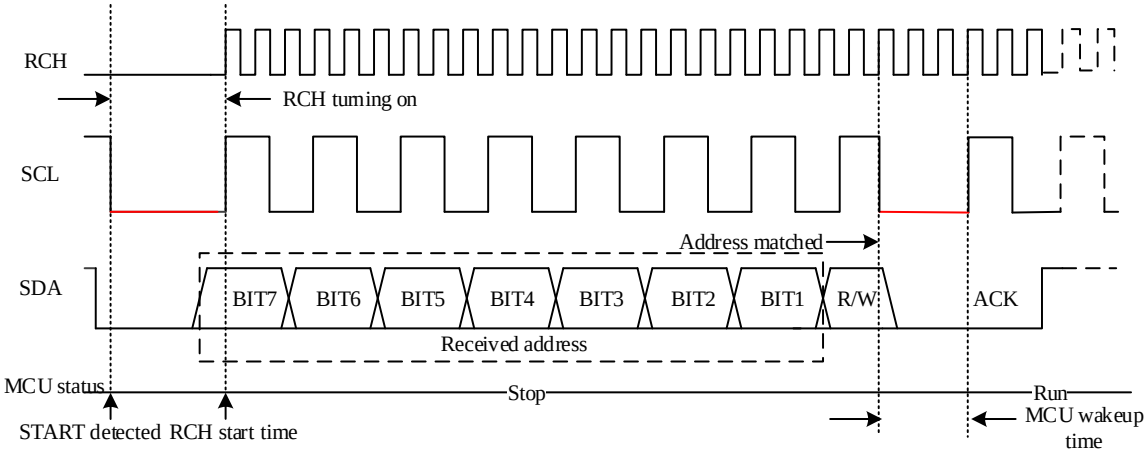
The I2C is able to wake up the MCU from the Stop mode, when it is addressed. The following addressing modes are supported:

- The received address matches I2C own address 1 or I2C own address 2 register.
- Enable general call (I2Cx_CR1 GCEN = 1), and received general call address.

RCH oscillator must be selected as the clock source for I2Cx_KCLK, and kept the NOSTRETCH bit cleared in the I2Cx_CR1 register to enable the SCL stretching function. Additionally, the WUPEN bit is must be set to enable wakeup function. Upon receiving a matched address, the Stop mode can be awakened.

Note: The digital filter is not compatible with wakeup from Stop mode feature. If the DNF bit is not equal to 0, setting the WUPEN bit has no effect.

Figure 29-18 Stop mode wakeup sequence



The wakeup process is as follows:

1) When a START condition is detected, send by the master device, the slave device stretches SCL low and requests to enable the RCH clock.
2) After the RCH clock starts and stabilizes, release SCL to receive the address.
3) If the received address matches, the slave device stretches SCL to wake up the entire system. If the received address does not match, turn off RCH, release SCL, and remain in Stop mode.
4) Once the MCU is fully awake, clear the ADDR flag to release SCL and proceed

with further processing.

29.3.11 DMA transfer

Transmission using DMA

DMA can ben enabled for transmission by setting the TXDMAEN bit in the I2Cx_CR1 register (see: *Direct memory access controller (DMA)*). Data is loaded from source address configured using the DMA peripheral to the I2Cx_TDR register whenever the TXIS bit is set.

Reception using DMA

DMA can be enabled for reception by setting the RXDMAEN bit in the I2Cx_CR1 register (see: *Direct memory access controller (DMA)*). Data is loaded from the I2Cx_RDR register to the destination address configured using the DMA peripheral whenever the RXNE bit is set.

Only the data is transferred with DMA (the transmitted slave address, acknowledge bit, START condition and STOP condition cannot be transferred with DMA):

- In master mode: The slave address, start bit are programmed by software (cannot be achieved through DMA). The DMA must be initialized before setting the START bit. The end of transfer is managed with the NBYTES counter.
- In slave mode:
  - With NOSTRETCH = 0, when all data are transferred using DMA, the DMA must be initialized before the address match event, or in ADDR interrupt subroutine, before clearing ADDR.
  - With NOSTRETCH = 1, the DMA must be initialized before the address match event.

29.3.12 Error conditions

The following errors are the error conditions which cause communication to fail.

- Bus error (BERR)

  During the address and data transmission phases, a bus error is detected when a START or a STOP condition is detected and is not located after a multiple of 9 SCL clock pulses (see: *Communication flow*). When a bus error is detected, the BERR flag is set in the I2Cx_ISR register.

- Arbitration lost (ARLO)

  When an arbitration loss is detected, the ARLO flag is set in the I2Cx_ISR register.

  - In master mode, arbitration loss is detected during the address phase, data phase and data acknowledge phase. In this case, the SDA and SCL lines are released, the START control bit is cleared by hardware and master switches automatically to slave mode.
  - In slave mode, arbitration loss is detected during data phase and data acknowledge phase. In this case, the transfer is stopped, and the SCL and SDA lines are released.

- Overrun/underrun error (OVR)

  An overrun or underrun error is detected in slave mode when NOSTRETCH = 1. When an overrun or underrun error is detected, the OVR flag is set in the I2Cx_ISR register.

  In reception when a new byte is received and the I2Cx_RDR register has not been read yet. The new received byte is lost, and a NACK is automatically sent as a response to the new byte.

  In transmission:

  - When STOPF = 1 and the first data byte should be sent. The content of the I2Cx_TDR register is sent if TXE = 0, 0xFF if not.
  - When a new byte must be sent and the I2Cx_TDR register has not been written yet, 0xFF is sent.

29.4 I2C low-power modes

The status of I2C peripheral in different low-power modes is shown in the table below:

Table 29-5 Low-power mode description

Mode	Description
Sleep	No effect. I2C interrupts cause the device to exit the Sleep mode.
Stop	The I2C registers content is kept. The I2C address match condition causes the device to exit the Stop mode. (see: *Wakeup from Stop mode on address match*)

29.5 I2C interrupts

The table below gives the list of I2C interrupt requests:

Table 29-6 I2C interrupt requests(1)

Interrupt event	Event flag	Enable control bit	Interrupt clear method	Exit the Sleep mode	Exit the Stop mode
Receive buffer not empty	RXNE	RXIE	Read I2Cx_RDR register	√	×
Transmit buffer interrupt status	TXIS	TXIE	Write I2Cx_TDR register	√	×
Stop detection interrupt flag	STOPF	STOPIE	Write STOPCF = 1	√	×
Transfer Complete Reload	TCR	TCIE	Write I2Cx_CR2 with NBYTES[7:0] ≠ 0	√	×
Transfer complete	TC		With START = 1 or STOP = 1	√	×
Address matched	ADDR	ADDRIE	Write ADDRCF = 1	√	√
NACK reception	NACKF	NACKIE	Write NACKCF = 1	√	×
Bus error	BERR	ERRIE	Write BERRCF = 1	√	×
Arbitration loss	ARLO		Write ARLOCF = 1	√	×
Overrun/underrun	OVR		Write OVRCF = 1	√	×

1. “√” means can wake up from low-power mode, “×” means cannot wakeup from low-power mode.

29.6 I2C registers

The I2C registers can only be accessed by words (32-bit).

Table 29-7 I2Cx base address (x = 1)

Peripheral	Base address
I2C1	0x4000 5400

29.6.1 I2C control register 1 (I2Cx_CR1)

Address offset: 0x00

Reset value: 0x0000 0000

Note: No wait states, except if a write access occurs while a write access to this register is ongoing. In this case, wait state are inserted in the second write access until the previous one is complete. The latency of the second write access can be up to 3 × t_{PCLK} + 4 × t_{I2Cx_KCLK}.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.												GCEN	WUPEN	NO STRETCH	SBC
												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.		DNF[3:0]				ERRIE	TCIE	STOP IE	NACK IE	ADDR IE	RXIE	TXIE	PE
rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Name	Description
31:20	Reserved	Must be kept at reset value
19	GCEN	General call enable 0: General call disabled, Address 0b00000000 is NACKed 1: General call enabled, Address 0b00000000 is ACKed
18	WUPEN	Wakeup from Stop mode enable 0: Wakeup from Stop mode disabled 1: Wakeup from Stop mode enabled *Note: WUPEN can be set only when DNF = "0000".*
17	NOSTRETCH	Clock stretching disable This bit is used to disable clock stretching in slave mode. It must be

		kept cleared in master mode. 0: Clock stretching enabled 1: Clock stretching disabled *Note: This bit can only programmed when the I2C is disabled (PE = 0).*
16	SBC	Slave byte control This bit is used to enable hardware byte control in slave mode. 0: Slave byte control disabled 1: Slave byte control enabled
15	RXDMAEN	DMA reception requests enable 0: DMA mode disabled for reception 1: DMA mode enabled for reception
14	TXDMAEN	DMA transmission requests enable 0: DMA mode disabled for transmission 1: DMA mode enabled for transmission
13:12	Reserved	Must be kept at reset value
11:8	DNF[3:0]	Digital noise filter These bits are used to configure the digital noise filter on SCL and SDA input. The digital filter, filters spikes with length of up to DNF[3:0] × I2Cx_KCLK. 0000: Digital filter disabled 0001: Digital filter enabled and filtering capability up to 1 × I2Cx_KCLK ⋮ 1110: Digital filter enabled and filtering capability up to 14 × I2Cx_KCLK 1111: Digital filter enabled and filtering capability up to 15 × I2Cx_KCLK *Note: This filter can only be programmed when I2C is disabled (PE = 0).*
7	ERRIE	Error interrupts enable 0: Error detection interrupts disabled

		1: Error detection interrupts enabled
6	TCIE	Transfer complete interrupt enable 0: Transfer complete interrupt disabled 1: Transfer complete interrupt enabled
5	STOPIE	Stop detection interrupt enable 0: Stop detection interrupt disabled 1: Stop detection interrupt enabled
4	NACKIE	NACK received interrupt enable 0: NACK received interrupt disabled 1: NACK received interrupt enabled
3	ADDRIE	Address match interrupt enable (slave mode) 0: Address match interrupt disabled 1: Address match interrupt enabled
2	RXIE	RX interrupt enable 0: Receive interrupt disabled 1: Receive interrupt enabled
1	TXIE	TX interrupt enable 0: Transmit interrupt disabled 1: Transmit interrupt enabled
0	PE	I2C enable 0: Peripheral disabled 1: Peripheral enabled *Note: When PE = 0, the I2C SCL and SDA lines are released. Internal state machines and status bits are put back to their reset value.*

29.6.2 I2C control register 2 (I2Cx_CR2)

Address offset: 0x04

Reset value: 0x0000 0000

Note: No wait states, except if a write access occurs while a write access to this register is

ongoing. In this case, wait state are inserted in the second write access until the previous one is complete. The latency of the second write access can be up to 3 × t_{PCLK} + 4 × t_{I2Cx_KCLK}.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.						AUTO END	RELOAD	NBYTES[7:0]							
						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
NACK	STOP	START	HEAD10R	ADDR10	RD_WRN	SADDR[9:0]									
rs	rs	rs	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Name	Description
31:26	Reserved	Must be kept at reset value
25	AUTOEND	Automatic end mode 0: Software end mode 1: Automatic end mode *Note: This bit has no effect in slave mode or when the RELOAD bit is set.*
24	RELOAD	Reload mode 0: Disable 1: Enable
23:16	NBYTES[7:0]	Number of bytes The number of bytes to be transmitted/received is programmed there. In slave mode, when SBC is 0, the bit is invalid. *Note: Changing these bits when START bit is set is not allowed.*
15	NACK	NACK generation (slave mode) The bit is set by software, cleared by hardware when the NACK is sent, or when a stop condition or an address matched is received, or when PE = 0. 0: An ACK is sent after current received byte 1: A NACK is sent after current received byte
14	STOP	Stop generation The bit is set by software, cleared by hardware when a stop

		condition is detected, or when PE = 0. 0: No Stop generation 1: Stop generation after current byte transfer
13	START	Start generation This bit is set by software, and cleared by hardware after the start followed by the address sequence is sent, by an arbitration loss, or when PE = 0. It can also be cleared by software by writing '1' to the ADDRCF bit in the I2Cx_ICR register. 0: No start generation 1: Restart/Start generation *Note: This bit has no effect when RELOAD is set. When the bus is busy, set the START will wait for the bus idle before sending the start condition.*
12	HEAD10R	10-bit address header only read direction 0: The master sends the complete 10-bit slave address read sequence: Start + 2 bytes 10-bit address in write direction + Restart + 1st 7-bit of 10-bit address in read direction 1: The master only sends the 1st 7-bit of the 10-bit address, followed by read direction *Note: Changing this bit when the START bit is set is not allowed.*
11	ADDR10	10-bit addressing mode 0: The master operates in 7-bit addressing mode 1: The master operates in 10-bit addressing mode *Note: Changing this bit when the START bit is set is not allowed.*
10	RD_WRN	Transfer direction 0: Master requests a write transfer 1: Master requests a read transfer *Note: Changing this bit when the START bit is set is not allowed.*
9:0	SADDR[9:0]	Slave address In 7-bit addressing mode (ADDR10 = 0): SADDR[7:1] should be written with 7-bit slave address to be sent. In 10-bit addressing mode (ADDR10 = 1): SADDR[9:0] should be written with the 10-bit slave address to be

sent.

Note: Changing this bit when the START bit is set is not allowed.

29.6.3 I2C address register 1 (I2Cx_ADDR1)

Address offset: 0x08

Reset value: 0x0000 0000

Note: No wait states, except if a write access occurs while a write access to this register is ongoing. In this case, wait state are inserted in the second write access until the previous one is complete. The latency of the second write access can be up to 3 × t_{PCLK} + 4 × t_{I2Cx_KCLK}.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR1EN	Res.				ADDR1 MODE	ADDR1[9:0]									
rw					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Name	Description
31:16	Reserved	Must be kept at reset value
15	ADDR1EN	Own address 1 enable 0: Own address 1 disabled. The received slave address ADDR1 is NACKed 1: Own address 1 enabled. The received slave address ADDR1 is ACKed
14:11	Reserved	Must be kept at reset value
10	ADDR1MODE	Own address 1 10-bit mode 0: Own address ADDR1 is a 7-bit address 1: Own address ADDR1 is a 10-bit address *Note: This bit can be written only when ADDR1EN = 0*
9:0	ADDR1[9:0]	Interface own slave address 1 7-bit addressing mode (ADDR1MODE = 0): ADDR1[7:1] contains the 7-bit own slave address.

10-bit addressing mode (ADDR1MODE = 1):

ADDR1[9:0] contains the 10-bit own slave address

Note: This bit can be written only when ADDR1EN = 0

29.6.4 I2C address register 2 (I2Cx_ADDR2)

Address offset: 0x0C

Reset value: 0x0000 0000

Note: No wait states, except if a write access occurs while a write access to this register is ongoing. In this case, wait state are inserted in the second write access until the previous one is complete. The latency of the second write access can be up to 3 × t_{PCLK} + 4 × t_{I2Cx_KCLK}.

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR2EN	Res.				ADDR2MSK[2:0]			ADDR2[7:1]							Res.
rw					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	

Bits	Name	Description
31:16	Reserved	Must be kept at reset value
15	ADDR2EN	Own address 2 enable 0: Own address 2 disabled. The received slave address ADDR2 is NACKed 1: Own address 2 enabled. The received slave address ADDR2 is ACKed
14:11	Reserved	Must be kept at reset value
10:8	ADDR2MSK[2:0]	Own address 2 masks 000: ADDR2[7:1] are compared 001: ADDR2[1] is masked. Only ADDR2[7:2] are compared 010: ADDR2[2:1] are masked. Only ADDR2[7:3] are compared 011: ADDR2[3:1] are masked. Only ADDR2[7:4] are compared 100: ADDR2[4:1] are masked. Only ADDR2[7:5] are compared 101: ADDR2[5:1] are masked. Only ADDR2[7:6] are compared

		110: ADDR2[6:1] are masked. Only ADDR2[7] is compared 111: ADDR2[7:1] are masked. No comparison is done, and all (except reserved) 7-bit addresses are acknowledged. *Note: This bit can be written only when ADDR2EN = 0*
7:1	ADDR2[7:1]	Interface own slave address 2 7-bit addressing mode *Note: This bit can be written only when ADDR2EN = 0.*
0	Reserved	Must be kept at reset value

29.6.5 I2C timing register (I2Cx_TIMING)

Address offset: 0x10

Reset value: 0x0030 3D5B

Note: This register can be written only when PE = 0.

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

Bits	Name	Description
31:28	PRESC[3:0]	Timing prescaler This field is used to prescale I2Cx_KCLK in order to generate the clock period SCLDEL used for data setup and hold counters and for SCL high and low level counters: t_{PRESC} = (PRESC + 1) × t_{I2Cx_KCLK}
27:24	Reserved	Must be kept at reset value
23:20	SCLDEL[3:0]	Data setup time This field is used to generate a delay t_{SCLDEL} between SDA edge and SCL rising edge. In master mode and in slave mode with NOSTRETCH = 0, the SCL line is stretched low during t_{SCLDEL}. t_{SCLDEL} = (SCLDEL + 1) × t_{PRESC}

Refer to *Figure: I2Cx_TIMING register configuration generates SCL timing*.

Note: t_{SCLDEL} is used to generate $t_{SU:DAT}$ timing.

19:16 SDADEL[3:0]

Data hold time

This field is used to generate the delay t_{SDADEL} between SCL falling edge and SDA edge. In master mode and in slave mode with NOSTRETCH = 0, the SCL line is stretched low during t_{SDADEL}.

t_{SDADEL} = SDADEL × t_{PRESC}

Refer to *Figure: I2Cx_TIMING register configuration generates SCL timing*.

Note: t_{SDADEL} is used to generate $t_{HD:DAT}$ timing.

15:8 SCLH[7:0]

SCL high period

This field is used to generate the SCL high period in master mode

t_{SCLH} = (SCLH + 1) × t_{PRESC}

Refer to *Figure: I2Cx_TIMING register configuration generates SCL timing*.

Note: t_{SCLH} is used to generate $t_{SU:STO}$ and $t_{HD:STA}$.

7:0 SCLL[7:0]

SCL low period

This field is used to generate SCL low period in master mode.

t_{SCLL} = (SCLL + 1) × t_{PRESC}

Refer to *Figure: I2Cx_TIMING register configuration generates SCL timing*.

Note: t_{SCLL} is used to generate t_{BUF} and $t_{SU:STA}$.

29.6.6 I2C interrupt and status register (I2Cx_ISR)

Address offset: 0x18

Reset value: 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.								ADDRCODE[6:0]							DIR
								r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUSY	Res.				OVR	ARLO	BERR	TCR	TC	STOPF	NACKF	ADDR	RXNE	TXIS	TXE
r					r	r	r	r	r	r	r	r	r	rs	rs

Bits	Name	Description
31:24	Reserved	Must be kept at reset value
23:17	ADDRCODE[6:0]	Address matched code These bits are updated with the received address when an address match event occurs (ADDR = 1). In the case of a 10-bit address, ADDCODE provides the 10-bit header followed by the MSBs of the address.
16	DIR	Transfer direction This flag is updated when an address match event occurs 0: Write transfer, slave enter receiver mode 1: Read transfer, slave enter transmitter mode
15	BUSY	Bus busy This flag indicates that a communication is in progress on the bus. it is set by hardware when START condition is detected. It is cleared by hardware when a STOP condition is detected, or when PE = 0
14:11	Reserved	Must be kept at reset value
10	OVR	Overrun/Underrun This flag is set by hardware in slave mode with NOSTRETCH = 1, when an overrun/underrun error occurs. It is cleared by software by setting the OVRCF bit *Note: This bit is cleared by hardware when PE = 0.*
9	ARLO	Arbitration lost This flag is set by hardware in case of arbitration loss. It is cleared by software by setting ARLOCF bit *Note: This bit is cleared by hardware when PE = 0.*
8	BERR	Bus error This flag is set by hardware when a misplaced START or STOP condition is detected whereas the peripheral is involved in the transfer. The flag is not set during the address phase in slave mode. It is cleared by software by setting BERRCF bit1. *Note: This bit is cleared by hardware when PE = 0.*

7	TCR	Transfer complete reload This flag is set by hardware when RELOAD = 1 and NBYTES data have been transferred. It is cleared by software when NBYTES is written to a non-zero value. *Note: This bit is cleared by hardware when PE = 0. This bit is only for master mode, or for slave mode when the SBC bit is set.*
6	TC	Transfer complete This flag is set by hardware when RELOAD = 0, AUTOEND = 0 and NBYTES data have been transferred. It is cleared by software when START bit or STOP bit is set. *Note: This bit is cleared by hardware when PE = 0.*
5	STOPF	Stop detection flag This flag is set by hardware when a STOP condition is detected on the bus and the peripheral is involved in this transfer: either as a master, provided that the STOP condition is generated by the peripheral, or as a slave, provided that the peripheral has been addressed previously during this transfer. It is cleared by software by setting the STOPCF bit. *Note: This bit is cleared by hardware when PE = 0.*
4	NACKF	Not Acknowledge received flag This flag is set by hardware when a NACK is received after a byte transmission. It is cleared by software by setting the NACKCF bit. *Note: This bit is cleared by hardware when PE = 0.*
3	ADDR	Address matched This bit is set by hardware as soon as the received slave address matched with one of the enabled salve addresses. It is cleared by software by setting ADDRCF bit. *Note: This bit is cleared by hardware when PE = 0.*
2	RXNE	Receive data register not empty This bit is set by hardware when the received data is copied into the I2Cx_RDR register, and is ready to be read. It is cleared when

I2Cx_RDR is read.

Note: This bit is cleared by hardware when PE = 0.

1	TXIS	Transmit interrupt status This bit is set by hardware when the I2Cx_TDR register is empty and the data to be transmitted must be written in the I2Cx_TDR register. It is cleared when the next data to be sent is written in the I2Cx_TDR register. This bit can be written to '1' by software when NOSTRETCH = 1 only, in order to generate a TXIS event (interrupt is TXIE = 1 or DMA request if TXDMAEN = 1). *Note: This bit is cleared by hardware when PE = 0.*
0	TXE	Transmit data register empty This bit is set by hardware when the I2Cx_TDR register is empty. It is cleared when the next data to be sent is written in the I2Cx_TDR register. This bit can be written to '1' by software in order to flush the transmit data register I2Cx_TDR. *Note: This bit is set by hardware when PE = 0.*

29.6.7 I2C interrupt clear register (I2Cx_ICR)

Address offset: 0x1C

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					OVRCF	ARLOCF	BERRCF	Res.		STOPCF	NACKCF	ADDRCF	Res.		
					w	w	w			w	w	w			

Bits	Name	Description
31:11	Reserved	Must be kept at reset value
10	OVRCF	Overrun/Underrun flag clear Writing 1 to this bit clears the OVR flag in the I2Cx_ISR register.

9	ARLOCF	Arbitration lost flag clear Writing 1 to this bit clears the ARLO flag in the I2Cx_ISR register.
8	BERRCF	Bus error flag clear Writing 1 to this bit clears the BERR flag in the I2Cx_ISR register.
7:6	Reserved	Must be kept at reset value
5	STOPCF	STOP detection flag clear Writing 1 to this bit clears the STOPF flag in the I2Cx_ISR register.
4	NACKCF	NACK flag clear Writing 1 to this bit clears the NACKF flag in the I2Cx_ISR register.
3	ADDRCF	Address matched flag clear Writing 1 to this bit clears the ADDR flag in the I2Cx_ISR register. Writing 1 to this bit also clears the START bit in the I2Cx_CR2 register.
2:0	Reserved	Must be kept at reset value

29.6.8 I2C receive data register (I2Cx_RDR)

Address offset: 0x24

Reset value: 0x0000 0000

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

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

Bits	Name	Description
31:8	Reserved	Must be kept at reset value
7:0	RXDATA[7:0]	Data byte received from the I2C bus.

29.6.9 I2C transmit data register (I2Cx_TDR)

Address offset: 0x28

Reset value: 0x0000 0000

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

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

Bits	Name	Description
31:8	Reserved	Must be kept at reset value
7:0	TXDATA[7:0]	Data byte to be transmitted to the I2C bus *Note: These bits can be written only when TXE = 1.*

30 Universal synchronous asynchronous receiver transmitter (USART)

30.1 Introduction

The USART offers a flexible means to perform full-duplex or half-duplex data exchange with external equipment, requiring a synchronous or asynchronous communication data format. It features an integrated fractional baud rate generator, with a very wide range of baud rates.

The USART supports single-wire half-duplex communication, Smartcard protocol, and IrDA (infrared data association) SIR ENDEC specifications. It also supports DMA (direct memory access) communication, Multiprocessor communication, and Modem operations (CTS/RTS).

The system provides 1 USART (USART1) and 3 UARTs (UART2/3/4).

30.2 USART main features

- Baud rate generator systems
  - Up to 6 Mbps with system clock 48 MHz, 8 oversampling
  - Oversampling method by 16 or 8
  - Fractional baud rate generator
- Character format
  - Data word length: 8 or 9 bits
  - Stop bits: 0.5, 1, 1.5 or 2 bits
  - Odd parity, even parity, no parity
  - Data order with MSB-first or LSB-first shifting
- SPI Master Mode, up to 6 Mbps
- Single-wire half-duplex communications
- Hardware flow control for modem and RS-485 transceiver
- Multiprocessor communications: wakeup from mute mode by idle line detection or address mark detection

- IrDA SIR encoder decoder supporting normal mode and low-power mode
- Smartcard mode:
  - Supports the T = 0 and T = 1 asynchronous protocols for smartcards as defined in the ISO/IEC 7816-3 standard
  - 0.5 and 1.5 stop bits for Smartcard operation
- Modbus communication with timeout feature and CR/LF character recognition
- LIN mode with break generation and break detection
- DMA communications
- Swappable Tx/Rx pin configuration
- Separate signal polarity control for transmission and reception

30.3 USART feature list

Table 30-1 USART and LPUART feature list(1)

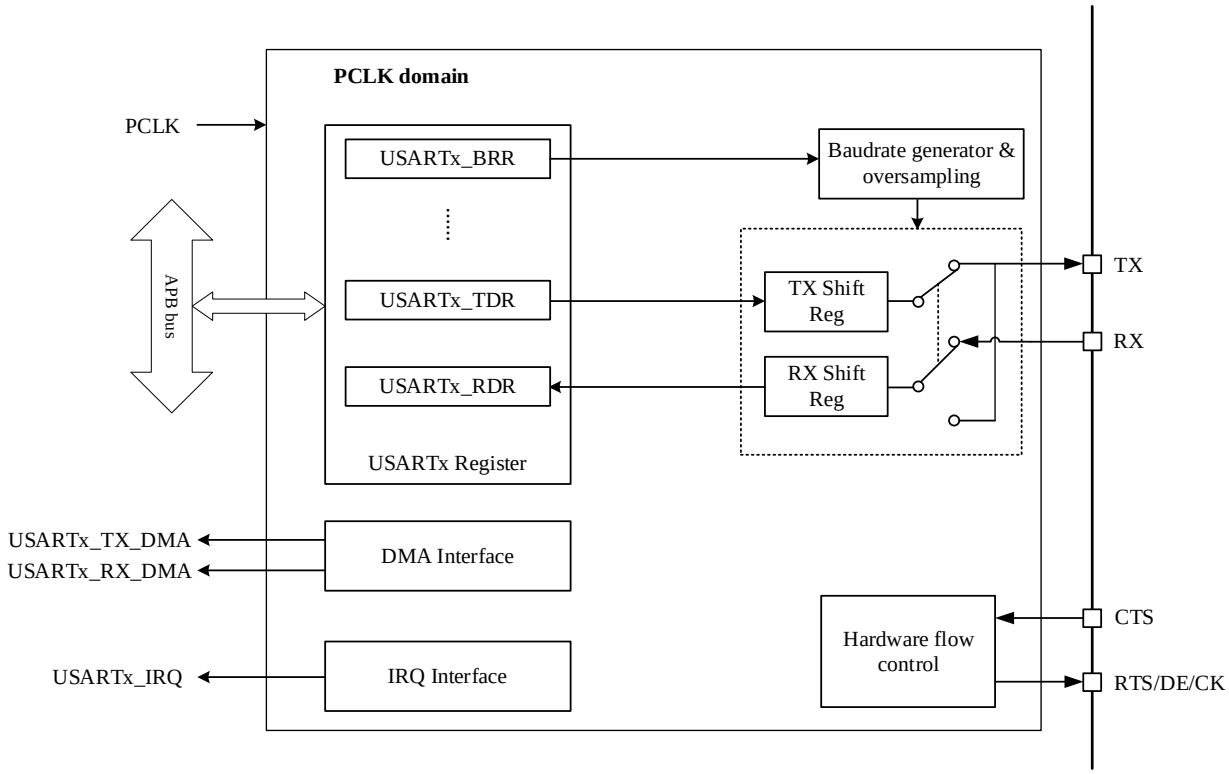
USART / LPUART function	USART1	UART2/3/4	LPUART1/2
Synchronous master mode	√	×	×
Single-wire half-duplex communication	√	√	√
Hardware flow control for modem	√	√	√
RS485	√	×	×
Multiprocessor communication	√	×	×
Smartcard master mode	√	×	×
IrDA SIR ENDEC block	√	×	×
Modbus communication	√	×	×
LIN mode	√	×	×
Receive timeout	√	×	×
Dual clock domain and wakeup from low-power mode	×	×	√
DMA communication	√	√	√
Data length	8,9 bits		
Break character	√	×	×
Idle character	√	×	×

1. “√” indicates support for this feature, “×” indicates no support for this feature.

30.4 USART functional description

30.4.1 USART block diagram

Figure 30-1 USARTx block diagram (x = 1, 2, 3, 4)



30.4.2 USART pins and signals

Table 30-2 USART1, UART2/3/4 internal signals

Signal	Type	Description
USARTx_TX_DMA	Output	USART DMA request signal for transmission
USARTx_RX_DMA	Output	USART DMA request signal for reception
USARTx_IRQ	Output	USART interrupt request signal

USART pins

USART bidirectional communications require a minimum of two pins: Receive Data In (RX) and Transmit Data Out (TX). The TX and RX pin swap functionality is supported, configured in the SWAP bit of the USARTx_CR2 register (bit 15).

The following pins are required in RS232 Hardware flow control mode:

- CTS (Clear To Send)

  The pin is input, when driven high, this signal blocks the data transmission at the

end of the current transfer.

- RTS (Request To Send)

  The pin is output, when it is low, this signal indicates that the USART is ready to receive data.

The following pin is required in RS485 Hardware control mode:

- DE (Driver Enable)

  The output signal of this pin activates the transmission mode of the external transceiver.

The following pin is required in synchronous master mode and Smartcard mode:

- CK
  - This pin acts as clock output in Synchronous master mode.
  - In Smartcard mode, CK output provides the clock to the smartcard.

30.4.3 USART character description

The word length can be programmed through the WL bit in the USARTx_CR1 register (bit 12). See Figure: *Character format diagram*.

- 8 bit word length: WL = 0
- 9 bit word length: WL = 1

Idle character

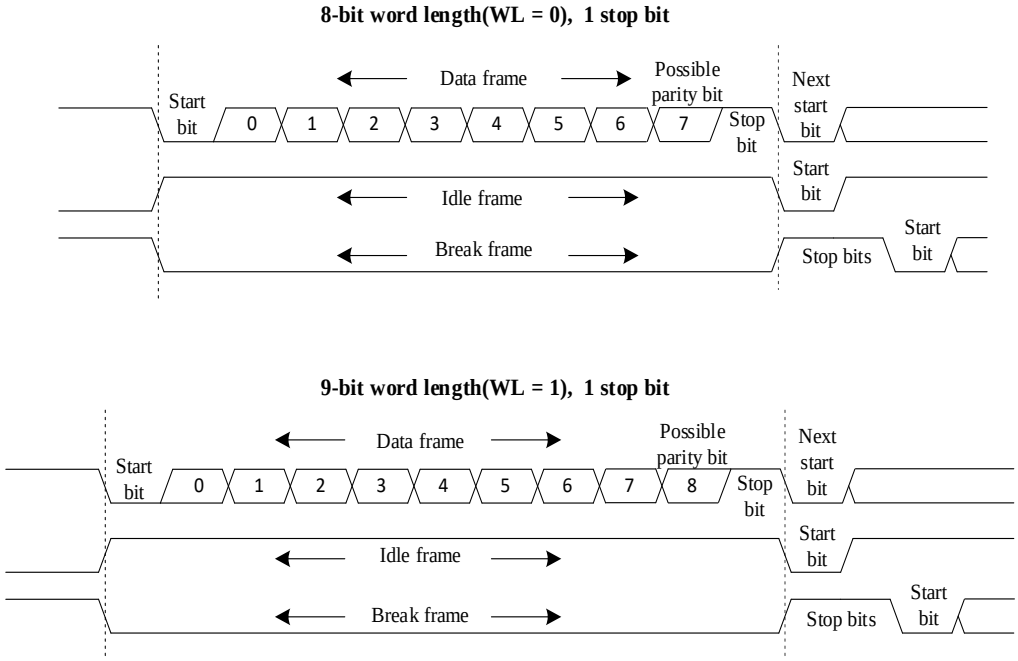
An Idle character is interpreted as an entire frame of 1.

Break character

A Break character is interpreted as an entire 0 for a frame period. At the end of the break frame, 2 stop bits are inserted.

By default, the TX and RX pins are low during the start bit and high during the stop bit. The polarity of the transmit and receive signals can be independently configured in the TXIVC and RXIVC bits of the USARTx_CR2 register (bit 16 and 17). See the following diagram:

Figure 30-2 Character format diagram



30.4.4 USART parity control

Table 30-3 USART parity control

Word length WL	Parity control PEN	USART character format
0	0	| start bit | 8 bit word length | stop bit |
	1	| start bit | 7 bit word length | parity bit | stop bit |
1	0	| start bit | 9 bit word length | stop bit |
	1	| start bit | 8 bit word length | parity bit | stop bit |

Even parity

When even parity (PTS bit is 0) is enabled, the total number of 1 in a character (including the parity bit) is even.

Odd parity

When odd parity (PTS bit is 1) is enabled, the total number of 1 in a character (including the parity bit) is odd.

Receiving parity check

The parity check can be enabled by setting the PEN bit in the USARTx_CR1 register (bit 10). If a parity error occurs, the Parity Error flag (PE) in the USARTx_ISR register (bit 0) is set to 1. If the PEIE bit in the USARTx_CR1 register (bit 8) is set to 1, an

interrupt is generated. Writing 1 to the PECF bit in the USARTx_ICR register (bit 0) clears the PE flag.

30.4.5 USART transmitter

The transmitter can send data words of either 8 or 9 bits, depending on the WL bit status. The transmitter can be enabled by setting the Transmit Enable (TE) bit to 1. The data in the transmit shift register is output on the TX pin while the corresponding clock pulses are output on the CK pin.

Character transmission

The transmitter can be enabled by setting the TE bit in the USARTx_CR1 register (bit 3) to 1.

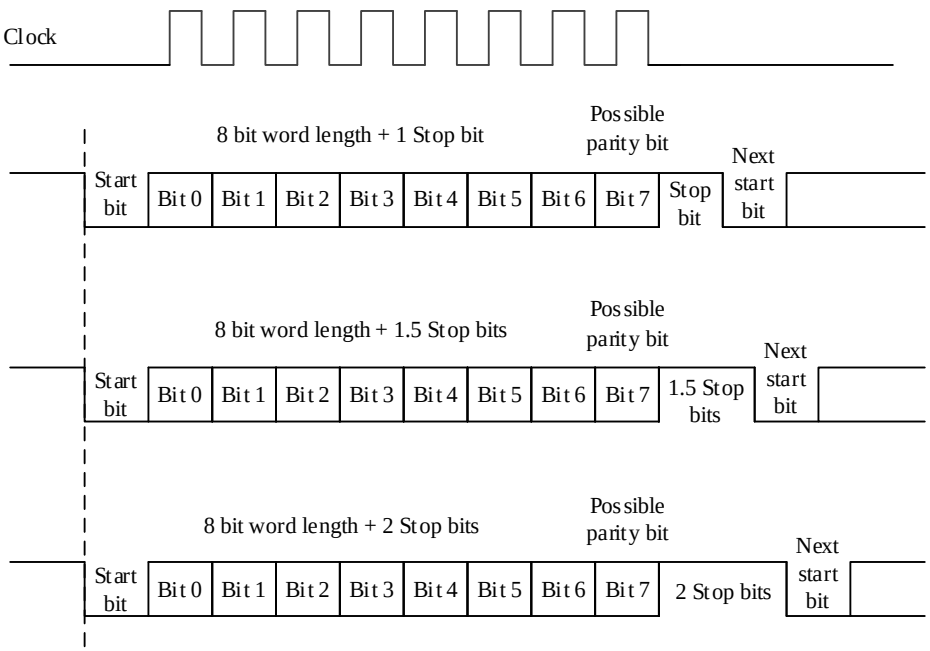
The transmission begins when the data to be sent is written to the USARTx_TDR register. During a transmission, data shifts out the least significant bit first (default configuration) on the TX pin. Each character is transmitted starting with the start bit and ending with the stop bit.

Stop bit

The number of stop bits to be transmitted with every character can be programmed by STOPBIT[1:0] in the USARTx_CR2 register (bit 13 and 12), as 0.5 bit, 1 bit, 1.5 bits, or 2 bits.

The following are examples of different stop bit configurations:

Figure 30-3 Configurable stop bits



Character transmission procedure

1) Select the desired baud rate using the USARTx_BRR register.
2) Program the WL bits in the USARTx_CR1 register to define the word length.
3) Program the STOPBIT[1:0] bits in the USARTx_CR2 register to define the number of stop bits.
4) Enable the USART by writing the UE bit in the USARTx_CR1 register to 1.
5) Select DMA enable (DMAT) in the USARTx_CR3 register if DMA communication must take place. Refer to the instructions in *USART communication using DMA* to configure the DMA registers.
6) Set the TE bit in the USARTx_CR1 register to enable the transmitter. Wait for the TEACK bit to be set, indicating that the transmitter is ready to send.
7) Write the data to be sent into the USARTx_TDR register. Writing data to the USARTx_TDR register will clear the TXE flag, indicating that the USARTx_TDR register is full.
8) When the last data to be sent is written to the USARTx_TDR register, wait until the TC flag in the USARTx_ISR register is set. The TC flag being set indicates that the last character has been successfully sent.

Character transmission process

When writing 1 character to the USARTx_TDR register, the TXE flag is cleared to 0.

When the transmit data register USARTx_TDR is empty, the TXE flag is automatically set to 1, indicating that data can be written. If the TXE interrupt enable TXEIE bit is set to 1, an interrupt will be generated.

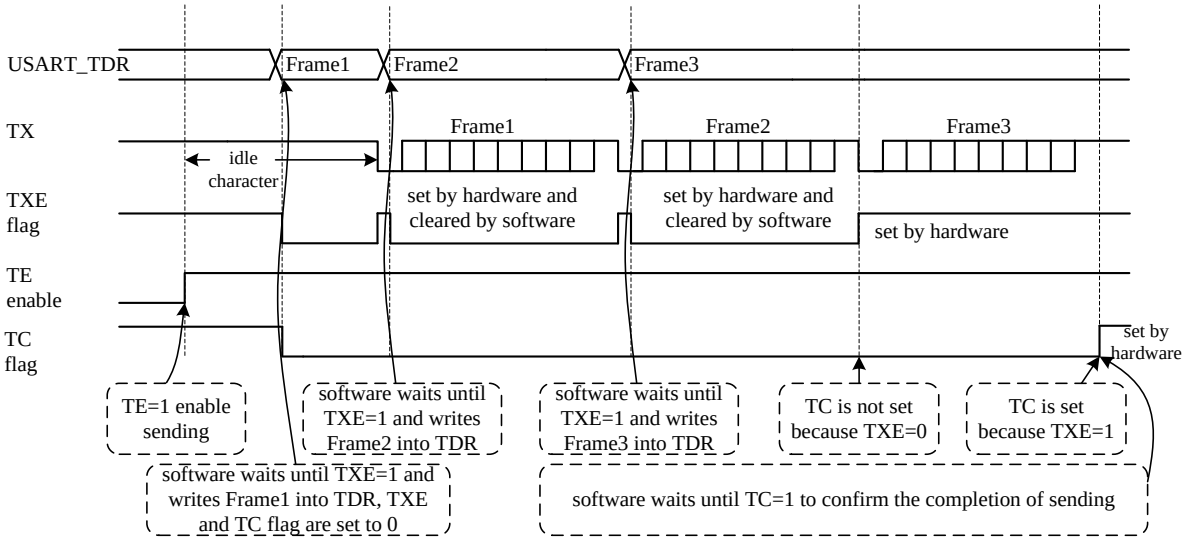
When the transmitter is not enabled (TE = 0), writing to the USARTx_TDR register clears the TXE bit to 0. Once data transmission starts, the TXE bit is immediately set to 1, indicating that the next data can be written.

When using DMA communication, each time the TXE bit is set, data is loaded from the memory area configured by the DMA to the USARTx_TDR register.

When all data has been written to the USARTx_TDR register and all data on the transmit line has been completely sent, the TC flag in the USARTx_ISR register is set to 1.

See the figure below for actions of TC and TXE during transmission:

Figure 30-4 TC/TXE behavior when transmitting



Break character

By setting the BKSENDQ bit in the USARTx_RQR register to 1, a break frame will be sent after the current transmission ends. BKSEND bit is automatically cleared to 0 during the stop bits of the break character. The length of the break character depends on the WL bit; see *Figure: Character format diagram*.

The USART inserts a logic 1 signal (stop) for the duration of 2 bits at the end of the break frame to guarantee the recognition of the start bit of the next frame.

Idle character

Setting the TE bit drives the USART1 to send an idle character before the first data frame while UART2/3/4 do not.

30.4.6 USART receiver

The USART can receive data words of either 8 or 9 bits depending on the WL bit in the USARTx_CR1 register.

Start bit detection

In the USART, the start bit is detected when a specific sequence of samples is recognized.

When oversampling by 8, the sequence is: 111 0 X X 0 0 0 X X

When oversampling by 16, the sequence is: 111 0 X 0 X 0 X 0 0 0 0 X X X X X X

Figure 30-5 Start bit detection when oversampling by 8

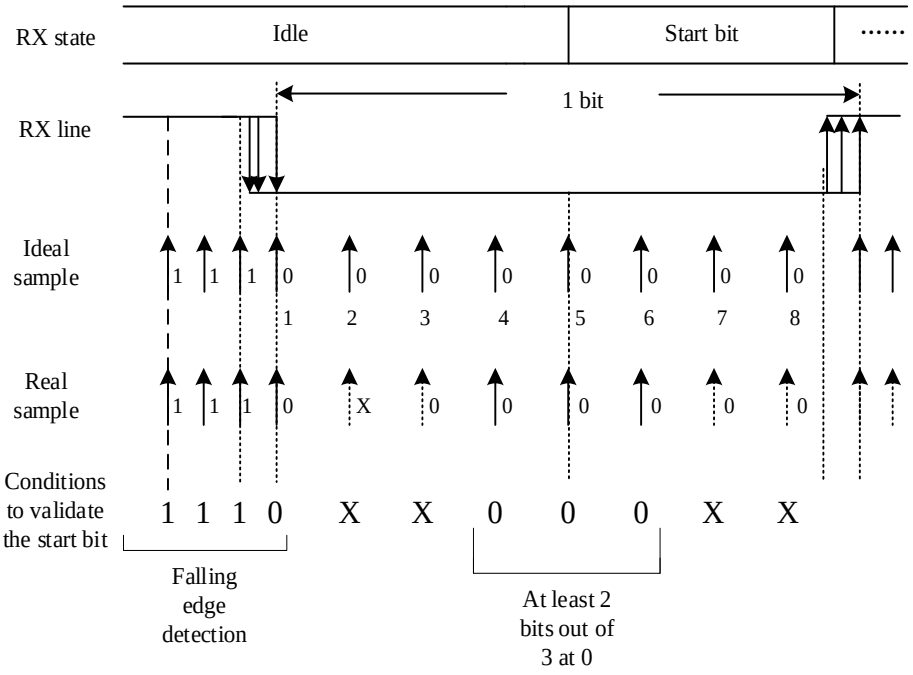
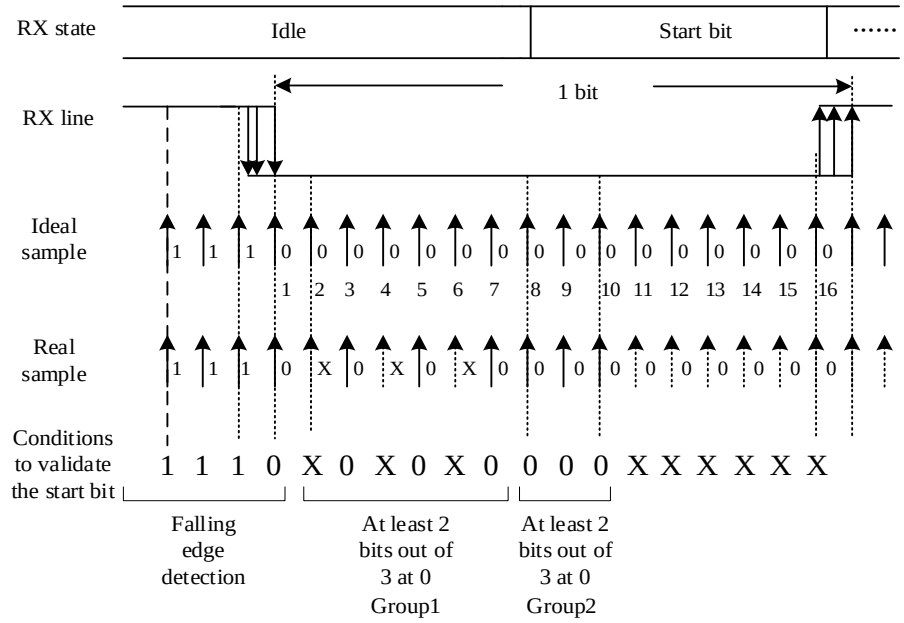


Figure 30-6 Start bit detection when oversampling by 16



When sampling by 8, the start bit is confirmed if the 3 sampled bits are at 0 (sampling on the 4th, 5th and 6th bits finds the 3 bits at 0).

When sampling by 16, the start bit is confirmed if the 3 sampled bits are at 0 (first sampling on the 3rd, 5th and 7th bits finds the 3 bits at 0 and second sampling on the 8th, 9th and 10th bits also finds the 3 bits at 0).

The start bit is validated but the NOISE flag is set if:

- When sampling by 8, 2 out of the 3 sampled bits are at 0 (sampling on the 4th, 5th and 6th bits), for example, the sampling result is 001
- When sampling by 16, 2 out of the 3 sampled bits are at 0 (sampling on the 3rd, 5th and 7th bits and sampling on the 8th, 9th and 10th bits), for example, the sampling result is 010 for the 3rd, 5th and 7th bits, the sampling result is 001 for the 8th, 9th and 10th bits
- When sampling by 16, for one of the samplings (sampling on the 3rd, 5th and 7th bits or sampling on the 8th, 9th and 10th bits), 2 out of the 3 bits are found at 0. For example, the sampling result is 000 for the 3rd, 5th and 7th bits, the sampling result is 001 for the 8th, 9th and 10th bits

The above description includes detecting the start bit without noise and detecting the start bit with noise. If neither of the above conditions are met, the start detection aborts and the receiver returns to the idle state, no flag is set.

Character reception

Character reception procedure is as below:

1) Select the desired baud rate using the USARTx_BRR register.
2) Program the WL bits in the USARTx_CR1 register to define the word length.
3) Program the STOPBIT[1:0] bits in the USARTx_CR2 register to define the number of stop bits.
4) Enable the USART by writing the UE bit in the USARTx_CR1 register to 1.
5) Select DMA enable (DMAR) in the USARTx_CR3 register if DMA communication must take place. Refer to the instructions in *USART communication using DMA* to configure the DMA registers.
6) Set the RE bit in the USARTx_CR1 register to enable the receiver.
7) Wait for the REACK bit to be set, indicating that the receiver is ready to receive.

When a character is received:

- When not using DMA communication:

  The RXNE flag is set to indicate that the content of the shift register is transferred to the USARTx_RDR register.

  Clearing the RXNE flag is done by performing a software read from the USARTx_RDR register. The RXNE flag can also be cleared by programming RXFLUQ bit to 1 in the USARTx_RQR register.

- When using DMA communication:

  Since the USARTx_RDR register has only 1 byte of space, the RXNE flag is set every time when a character is received. The RXNE flag is cleared when the DMA reads the USARTx_RDR register.

- The error flags can be set if a frame error, noise, an overrun or parity error was detected during reception.

Break character

When a break character is received, the USART handles it as a framing error.

Idle character

When an idle frame is detected, the IDLE flag is set and an interrupt is generated if the IDLEIE bit is set.

Overrun error

The overrun error detection is enabled by default. When an overrun error is detected, the ORE flag is set. An interrupt is generated if either the RXNEIE or the EIE bit in the USARTx_CR1 register is set.

The ORE bit is reset by setting the ORECF bit in the USARTx_ICR register.

Overrun error detection can be disabled by setting the ORED bit in the USARTx_CR3 register to 1. When the detection is disabled, the ORE flag remains at 0.

The RXNE flag is set after every byte reception. An overrun error occurs if a character is received and RXNE has not been reset, or the previous DMA request has not been serviced. Data can not be transferred from the shift register to the RDR register until the RXNE bit is cleared.

When an overrun error occurs:

- The ORE flag is set.
- The shift register is overwritten. After that, any data received during overrun is lost.
- The RDR content is not lost until the ORE flag is cleared. The previous data is available by reading the USARTx_RDR register.
- An interrupt is generated if either the RXNEIE or the EIE bit in the USARTx_CR1 register is set.

The ORE bit, when set, indicates that at least 1 data has been lost.

Selecting the oversampling method

The receiver implements 16x or 8x oversampling techniques (except in synchronous mode).

The oversampling method can be selected by programming the OVS8 bit in the USARTx_CR1 register (bit 15). There are two options, see *Figure: Data sampling - oversampling by 8* and *Figure: Data sampling - oversampling by 16*.

- Select 8x oversampling (OVS8 = 1), which samples each bit 8 times, supporting a higher speed (up to USARTx_PCLK/8) compared to 16x oversampling. In this case the maximum receiver tolerance is reduced, making it suitable for systems with low noise.
- Select 16x oversampling (OVS8 = 0), which samples each bit 16 times, increasing the receiver tolerance. In this case, the maximum speed is limited to USARTx_PCLK/16.

Programming the OBS bit in the USARTx_CR3 register selects the method used to evaluate the logic level. Two options are available:

- The majority vote of the three samples in the center of the received bit. In this case, when the 3 samples used for the majority vote are not equal, the NOISE bit is set.
- A single sample in the center of the received bit, with no NOISE bit set.

Note: Oversampling by 8 is not available in the Smartcard, and LIN modes. In those modes, the OVS8 bit is forced to 0 by hardware.

Depending on your application:

- Select the three sample majority vote method (OBS = 0) when operating in a noisy environment and reject the data when a noise is detected because this indicates that a glitch occurred during the sampling.
- Select the single sample method (OBS = 1) when the line is noise-free to increase the receiver tolerance (see *Tolerance of the USART receiver*).

When noise is detected in a frame:

- The NOISE bit is set at the rising edge of the RXNE flag.
- The invalid data is transferred from the shift register to the USARTx_RDR

register.

- When the NOISE bit is set, an interrupt is generated if the EIE bit in the USARTx_CR3 register is set. When the NOISE bit is set, the RXNE bit is also set, an interrupt is generated if the RXNEIE bit is set.

The NOISE bit is reset by setting NOISECF bit in the USARTx_ICR register.

Figure 30-7 Data sampling - oversampling by 8 (OBS = 0)

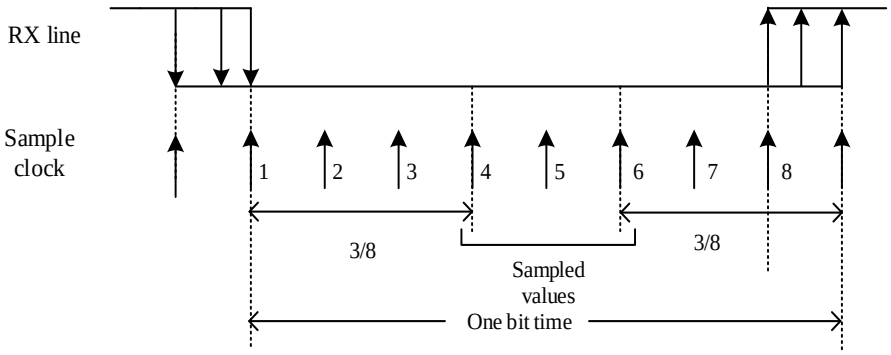
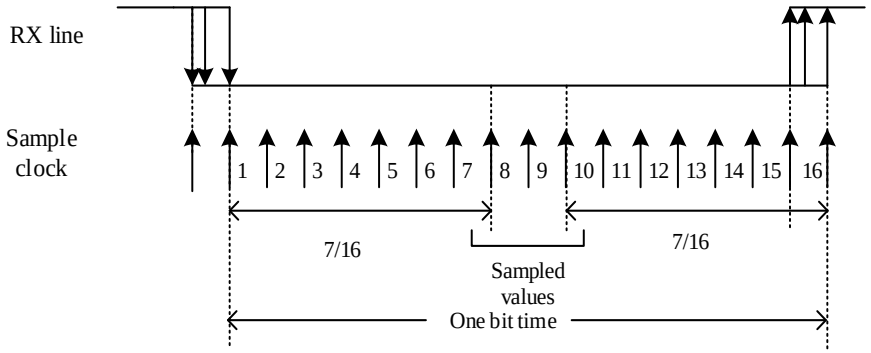


Figure 30-8 Data sampling - oversampling by 16 (OBS = 0)



Framing error

A framing error is detected when the stop bit is not recognized on reception at the expected time. When the framing error is detected:

- The FE bit is set by hardware.
- The invalid data is transferred from the shift register to the USARTx_RDR register.
- When the FE bit is set, an interrupt is generated if the EIE bit in the USARTx_CR3 register is also set. When the FE bit is set, the RXNE bit is also set, an interrupt is generated if the RXNEIE bit is set.

The FE bit is reset by setting FECF bit in the USARTx_ICR register.

Configurable stop bits during reception

The number of stop bits to be received can be configured through the STOPBIT[1:0] bits in the USARTx_CR2 register.

When the sampling method is chosen to be 3 samples (OBS = 0), there are several options:

- 0.5 stop bit: when it is selected, frame errors and break frames are not detected.
- 1 stop bit: sampling for 1 stop bit is done on the middle position of the stop bit for 3 times.
- 1.5 stop bits: sampling for 1.5 stop bits is done at the end of the first stop bit for 3 times.
- 2 stop bits: sampling for 2 stop bits is done on the middle position of the first stop bit for 3 times.

When the sampling method is chosen to be 1 samples (OBS = 1), there are several options:

- 0.5 stop bit: when it is selected, frame errors and break frames are not detected.
- 1 stop bit: sampling for 1 stop bit is done on the middle position of the stop bit for 1 time.
- 1.5 stop bits: sampling for 1.5 stop bits is done at the end of the first stop bit for 1 time.
- 2 stop bits: sampling for 2 stop bits is done on the middle position of the first stop bit for 1 time.

30.4.7 USART baud rate generation

The baud rate for the receiver and transmitter are both set to the value programmed in the USARTx_BRR register.

The baud rate is given by the following formula:

In case of oversampling by 16 (OVS8 = 0): $\mathbf{Tx/Rx\ baud} = \frac{USART\_PCLK}{BRR}$

In case of oversampling by 8 (OVS8 = 1): $\mathbf{Tx/Rx\ baud} = \frac{2\times USART\_PCLK}{BRR}$

BRR is an unsigned decimal number stored in the USARTx_BRR register.

Note: In case of oversampling by 16 and 8, BRR must be greater than or equal to 16.

Derive BRR based on the clock frequency and baud rate

Example1: To obtain 9600 baud with USARTX_PCLK = 16 MHz

- In case of oversampling by 16:

  BRR = 16 000 000/9600

  BRR = 0d1667 = 0x683

- In case of oversampling by 8:

  BRR = 2 × 16 000 000/9600

  BRR = 0d3333 = 0xD05

Example2: To obtain 921.6 K baud with USARTX_PCLK = 48 MHz

- In case of oversampling by 16:

  BRR = 48 000 000/921 600

  BRR = 0d52 = 0x34

- In case of oversampling by 8:

  BRR = 2 × 48 000 000/921 600

  BRR = 0d104 = 0x68

30.4.8 Tolerance of the USART receiver to clock deviation

The USART asynchronous receiver operates correctly only if the total clock system deviation is less than the tolerance of the USART receiver. The causes which contribute to the total deviation are:

- DTR_ERR: deviation due to the transmitter error, which also includes the deviation of the transmitter's local oscillator.
- DQU_ERR: error due to the baud rate quantization of the receiver.
- DRX_OSERR: deviation of the receiver local oscillator.
- DL_ERR: deviation due to the transmission line (generally due to the transceivers which can introduce an asymmetry between the low-to-high transition timing and the high-to-low transition timing).

  The receive tolerance must meet:

DTR_ERR + DQU_ERR + DRX_OSERR + DL_ERR < USART receiver tolerance.

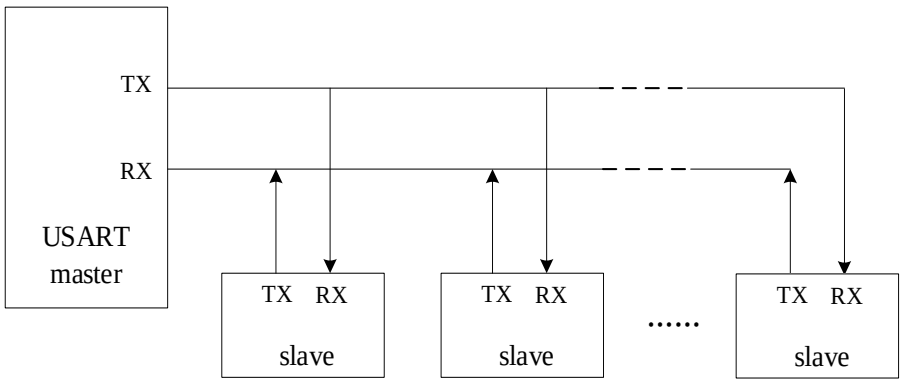
The USART receive tolerance depends on the following settings:

- 8- or 9-bit character length defined by the WL bits in the USARTx_CR1 register, the receive tolerance is greater when using 8-bit character length.
- Oversampling by 8 or 16 defined by the OVS8 bit in the USARTx_CR1 register, the receive tolerance is greater when using oversampling by 16.
- Bits BRR[3:0] of the USARTx_BRR register are equal to or different from 0000, since it is necessary to support oversampling clock, BRR is constrained to be no less than 16, which is reflected in the USARTx_BRR register as '0x10h', when BRR[3:0] equals 0000, BRR is an integer multiple of 16, when BRR[3:0] does not equal 0000, a fractional part is introduced. Therefore, the receive tolerance is greater when BRR[3:0] equals 0000.
- Use of 1 bit or 3 bits to sample the data, depending on the value of the OBS bit in the USARTx_CR3 register, the receive tolerance is greater when using 1 bit to sample the data.

30.4.9 USART multiprocessor communication

Several USARTs are connected in a network for USART multiprocessor communications. See the figure below:

Figure 30-9 Multiprocessor communication connection



In multiprocessor configurations, the slave device enables mute mode to avoid redundant USART service overhead caused by unaddressed devices participating in the communication. Mute mode is enabled by setting the RXMME bit in the USARTx_CR1 register.

The USART does not enter mute mode immediately when mute mode is enabled, and

the RWU bit (mute mode status flag) in the USARTx_ISR register remains 0. The USART can enter mute mode by software (writing 1 to the MUTEQ bit in the USARTx_RQR register) or by hardware automatically. For more details, see *Figure: Mute mode using address mark detection*.

After entering mute mode, the RWU flag is set to 1, and the RXNE flag will not be set to 1.

The USART can exit from mute mode using one of two methods, depending on the RXWKUP bit in the USARTx_CR1 register:

- Idle line detection if the RXWKUP bit is reset.
- Address mark detection if the RXWKUP bit is set.

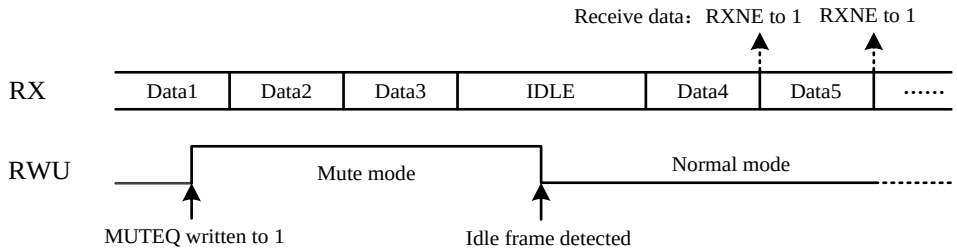
Idle line detection (RXWKUP = 0)

The USART enters mute mode when the MUTEQ bit in the USARTx_RQR register is written to '1' and the RWU bit is automatically set (indicates the USART has entered mute mode).

The USART wakes up when an idle frame is detected. The RWU bit is then cleared by hardware and the IDLE bit in the USARTx_ISR register is set.

An example of mute mode behavior using idle line detection is given in Figure: Mute mode using idle line detection:

Figure 30-10 Mute mode using idle line detection



Note: If the MUTEQ is set while the IDLE character has already elapsed, Mute mode is not entered (RWU is not set). It is necessary to write 1 to the MUTEQ bit after the start bit of the next data is received and before the next idle frame arrives, in order to re-enter the mute mode.

Address mark detection (RXWKUP = 1)

In this mode, bytes are recognized as addresses if their MSB is a '1', otherwise they are considered as data.

In an address byte, the address length can be detected as 4 bits or 7, 8 bits. The local address of the slave device is set in the ADDR[7:0] bit field of the USARTx_CR2 register, and the setting method is as follows:

- ADDRM is 0: the address length is 4 bits, stored in ADDR[3:0].
- ADDRM is 1:
  - When the character length is 8 bits, the address length is 7 bits, stored in ADDR[6:0].
  - When the character length is 9 bits, the address length is 8 bits, stored in ADDR[7:0].

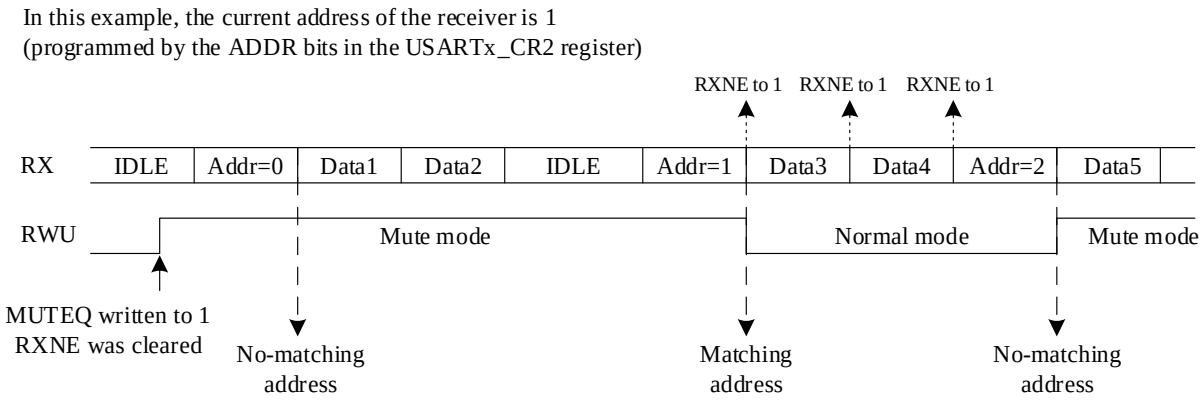
In this mode, the USART enters mute mode when the MUTEQ bit is written to 1, or an address character is received which does not match its programmed address. The RXNE flag is not set for this address byte and no interrupt or DMA request is issued.

The USART exits from mute mode when an address character is received which matches the programmed address. Then the RWU bit is cleared and subsequent bytes are received normally. The RXNE flag is set for the address character since the RWU bit has been cleared.

The USART enters mute mode when an address character is received which does not match its programmed address again.

An example of mute mode behavior using address mark detection is given as below:

Figure 30-11 Mute mode using address mark detection



30.4.10 USART Modbus communication

The USART offers basic support for the end of the block detection in the Modbus/RTU and Modbus/ASCII protocols, without software overhead or other resources.

The control part of the protocol must be implemented in software, including address recognition, block integrity control and command interpretation.

Modbus/RTU

The receiving USART implements block end detection through the timeout function. The value corresponding to a timeout is configured in the RTO[0:23] bits of the USARTx_RTO register, with a unit of 1 data bit. For configuration details, refer to: *USART receiver timeout*.

Modbus/ASCII

In this mode, the end of a block is recognized by a specific (CR/LF) character sequence. The USART manages this mechanism using the character match function.

By programming the CR/LF ASCII code in the ADDR[7:0] field and by activating the character match interrupt (CMIE = 1), an interrupt is generated when a CR/LF has been received.

30.4.11 USART LIN (local interconnection network) mode

The LIN mode is selected by setting the LINEN bit in the USARTx_CR2 register. In LIN mode, the following bits must be kept cleared:

- CLKEN in the USARTx_CR2 register
- STOP[1:0], SCEN, HDEN and IREN in the USARTx_CR3 register

LIN transmission

Data transmission in LIN mode is the same as in UART mode, only supporting 8-bit character length, which means the WL bit in the USARTx_CR1 register must be cleared to 0.

In LIN mode, setting the BKSENDQ bit sends 13 bit 0 as a break character.

LIN reception

When LIN mode is enabled, the break detection circuit is activated. The detection is totally independent from the normal UART receiver, with no effect on the normal receiving process.

Break detection can be configured to detect a low level of 10 or 11 bits, which is configured through the LBDL bit in the USARTx_CR2 register.

Similar to data reception, after a start bit has been detected, the circuit samples the next

bits exactly like for the data. If it samples a 0 for each consecutive bits and also samples a 0 at the 10th or 11th bit (configured by the LBDL bit), and are followed by a delimiter character, it indicates that a break frame has been detected. The LBD bit in the USARTx_ISR register is set to 1. If the LBDIE bit is set to 1, an interrupt is generated. Before validating the break, the delimiter is checked for as it signifies that the RX line has returned to a high level.

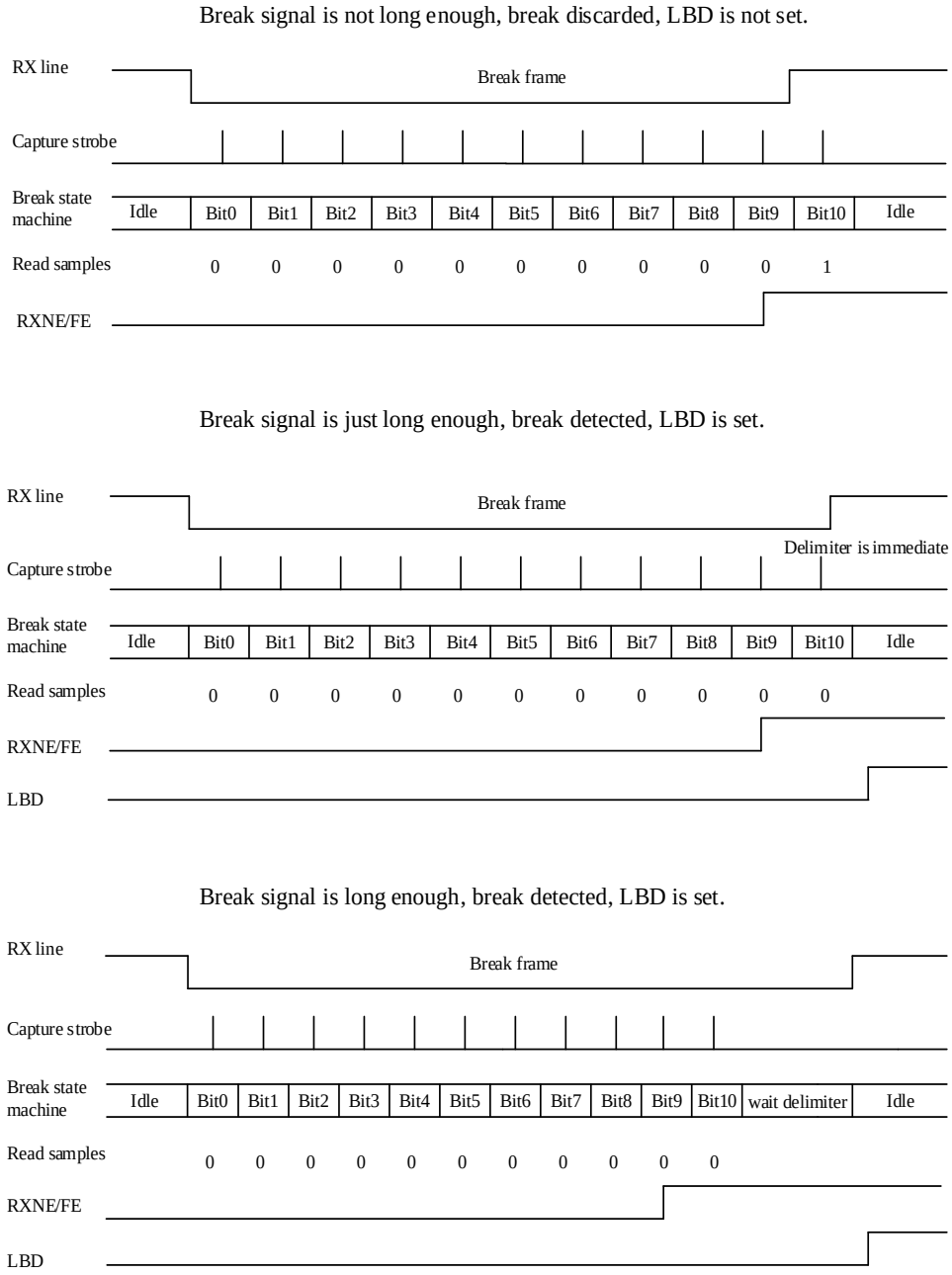
If a 1 is sampled before the bit10 or bit11 have occurred, the break detection circuit cancels the current detection and searches for a start bit again.

If the LIN mode is disabled (LINEN = 0), the receiver does not take into account the break detection.

If the LIN mode is enabled (LINEN = 1), as soon as a framing error occurs (stop bit detected at 0, which is the case for any break frame), the receiver stops until the break detection terminates, if the break word was not complete, or receives a delimiter if a break has been detected.

The behavior of the break detector, the break flag and the frame error is shown as below:

Figure 30-12 Break detection in LIN mode (11-bit break length, LBDL bit is set)



30.4.12 USART synchronous master mode (SPI)

The synchronous master mode is selected by programming the CLKEN bit in the USARTx_CR2 register to 1. In synchronous master mode, the following bits must be kept cleared:

- LINEN bit in the USARTx_CR2 register
- SCEN, HDEN and IREN bits in the USARTx_CR3 register

In synchronous master mode, the configuration is as follows:

- The CK pin is the output of the USART synchronous master mode transmitter clock. No clock pulses are sent to the CK pin during start bit and stop bit.
- The CPOL bit in the USARTx_CR2 register is used to select the clock polarity.
- The CPHA bit in the USARTx_CR2 register is used to select the phase of the clock.

(For details, see *Figure: Data clock timing diagram in synchronous master mode (WL = 0)* and *Figure: Data clock timing diagram in synchronous master mode (WL = 1)*)

During the idle state, idle frame and send break, the external CK clock is not activated.

In synchronous master mode, if RE bit is set to 1, the data are sampled on CK (rising or falling edge, depending on CPOL and CPHA), without any oversampling, and the parity function is not supported.

The master device can use a GPIO as the slave device's chip select control.

Figure 30-13 Example of synchronous master transmission

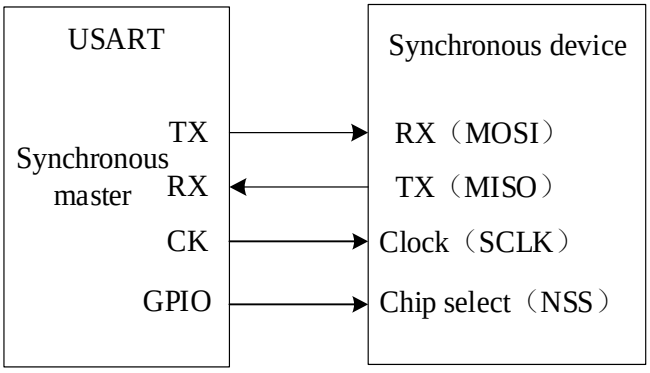


Figure 30-14 Data clock timing diagram in synchronous master mode (WL = 0)

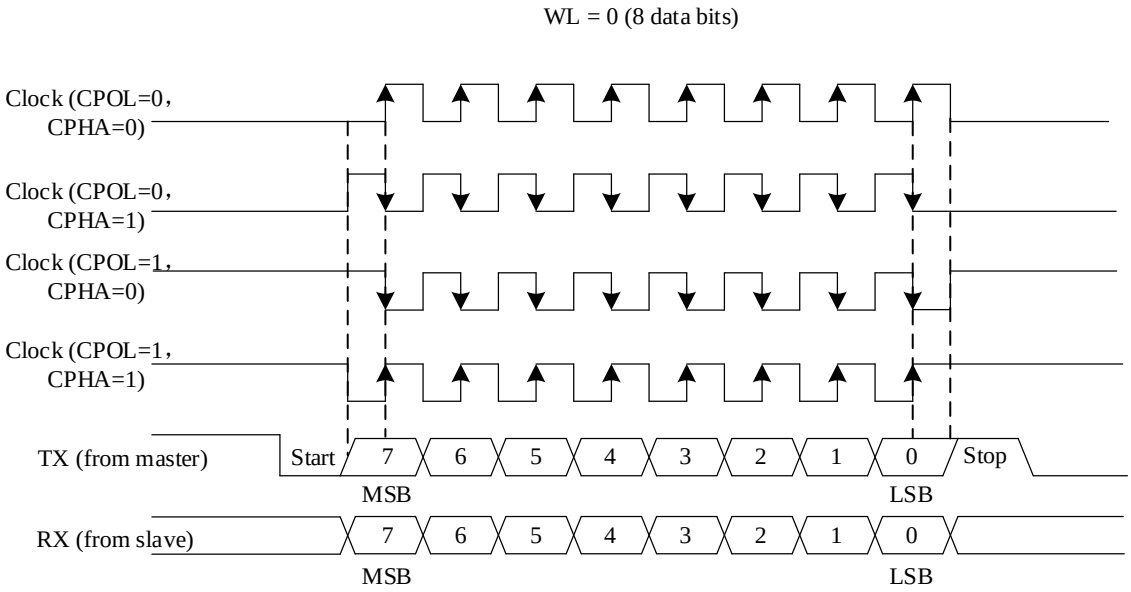
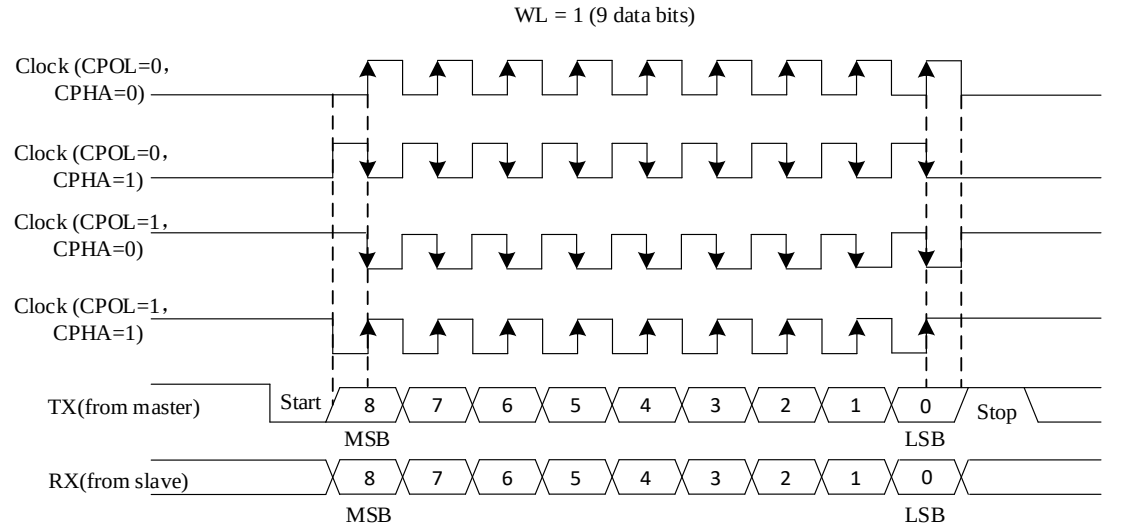


Figure 30-15 Data clock timing diagram in synchronous master mode (WL = 1)



30.4.13 USART single-wire half-duplex communication

Single-wire half-duplex mode is supported by USART and is selected by setting the HDEN bit in the USARTx_CR3 register. In this mode, the following bits must be kept cleared:

- LINEN and CLKEN bits in the USARTx_CR2 register
- SCEN and IREN bits in the USARTx_CR3 register

The USART can be configured to follow a single-wire half-duplex protocol where the TX and RX lines are internally connected.

Writing 1 to the HDEN bit enables the single-wire half-duplex mode, which results in the following actions:

- The TX and RX lines are internally connected
- By default, the TX pin is used for data transmission and reception, while the RX pin is not used and can be used as a GPIO.
- When the SWAP bit in USARTx_CR2 is set to 1, the TX/RX pin swap is enabled, and the RX pin is used for data transmission and reception.
- The TX pin is always released when no data is transmitted. The I/O must be configured so that TX is configured as alternate function open-drain with an internal or external pull-up.

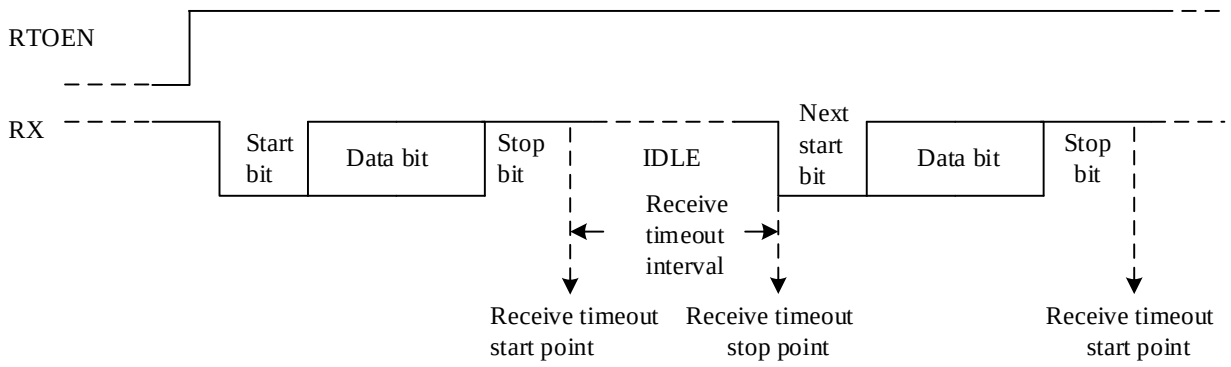
30.4.14 USART receiver timeout

The receiver timeout feature is enabled by setting the RTOEN bit in the USARTx_CR2 register. The timeout duration is programmed using the RTOV[15:0] bit fields in the USARTx_RTOR register and the unit is 1 data bit.

The starting point for counting is the end of the stop bit of the previous received data, depending on the configuration of the STOPBIT[1:0] bits in the USARTx_CR2 register. The receive timeout counting stops when the start bit of the next frame of data is detected. For receive timeout in Smartcard mode, see *Smartcard Mode (ISO7816)*.

When the timeout duration has elapsed, the RTOF flag in the USARTx_ISR register is set. A timeout interrupt is generated if RTOIE bit in the USARTx_CR1 register is set.

Figure 30-16 Receive timeout timing diagram



30.4.15 USART Smartcard mode (ISO7816)

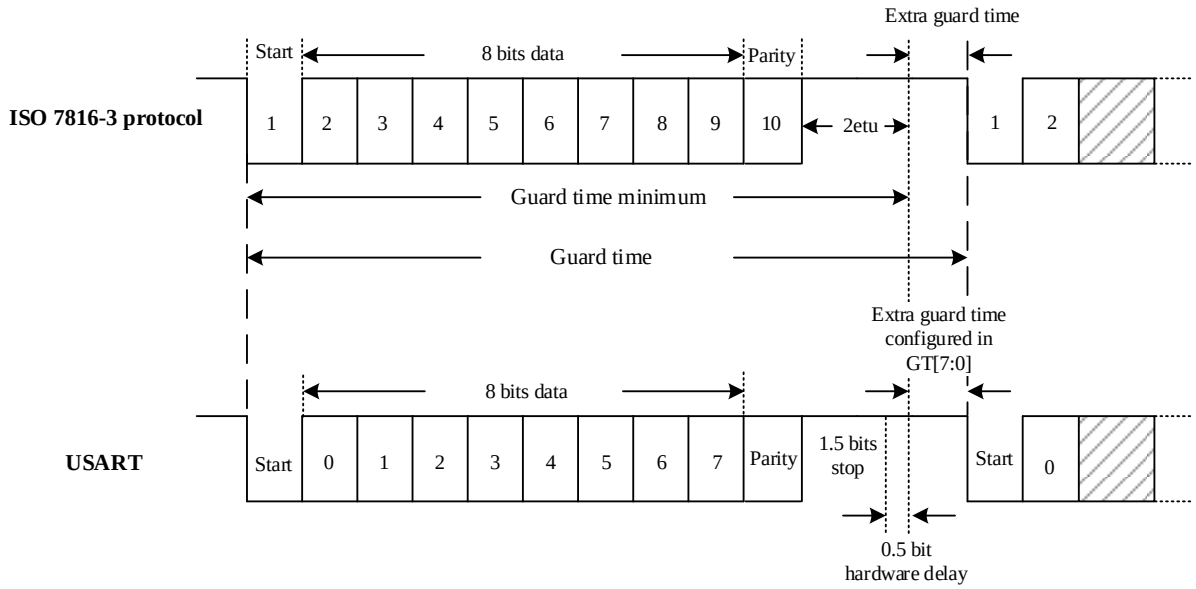
Smartcard mode complies with the ISO7816 standard and supports both T = 0 and T = 1 protocols.

Smartcard mode is selected by setting the SCEN bit in the USARTx_CR3 register. In Smartcard mode, the following bits must be kept cleared:

- LINEN bit in the USARTx_CR2 register
- HDSEL and IREN bits in the USARTx_CR3 register

In Smartcard mode, the USART should be configured as 9 bits data length (WL = 1), with even parity enabled (PEN = 1, PTS = 0). Fixed to use three samples (OBS = 0), and 16x oversampling (OV8 = 0). The extra guard time can be configured through the GT[7:0] bits. Additionally, by configuring the data endianness (MSBFIRST) and data polarity (DATAIVC), the USART can adapt to both direct and inverse conventions. The character frame format is shown in the following figure:

Figure 30-17 ISO 7816-3 asynchronous protocol



USART outputs the clock through the CK pin, and the output clock frequency can be configured in the PSV[4:0] bits.

Smartcard mode supports 0.5 or 1.5 stop bits. When configured for 1.5 stop bits, NACK functionality is supported, which can be enabled by setting the SCNAK bit. After enabled, NACK is detected during transmission, and NACK is returned upon receiving a parity error. The error retry mechanism is supported, and the number of

retries can be configured in the SCRETRY[2:0] bits.

The receive timeout feature starts when RTOEN is set to 1. Each time a complete frame of data is received, the timeout counter is cleared and restarted. This function is used for timeout sequence control, such as detecting WT in T = 0 and BWT, CWT in T = 1. The RTOV[7:0] bit field is used to configure the timeout duration, with a unit of 1etu. The receive timeout flag RTOF is set to 1 if no data is received by the end of the timeout interval.

The block length counter is used for block end detection and is configured in the BLKN[7:0] bit field of the USARTx_RTO register to specify the expected number of data bytes to be received.

T = 0

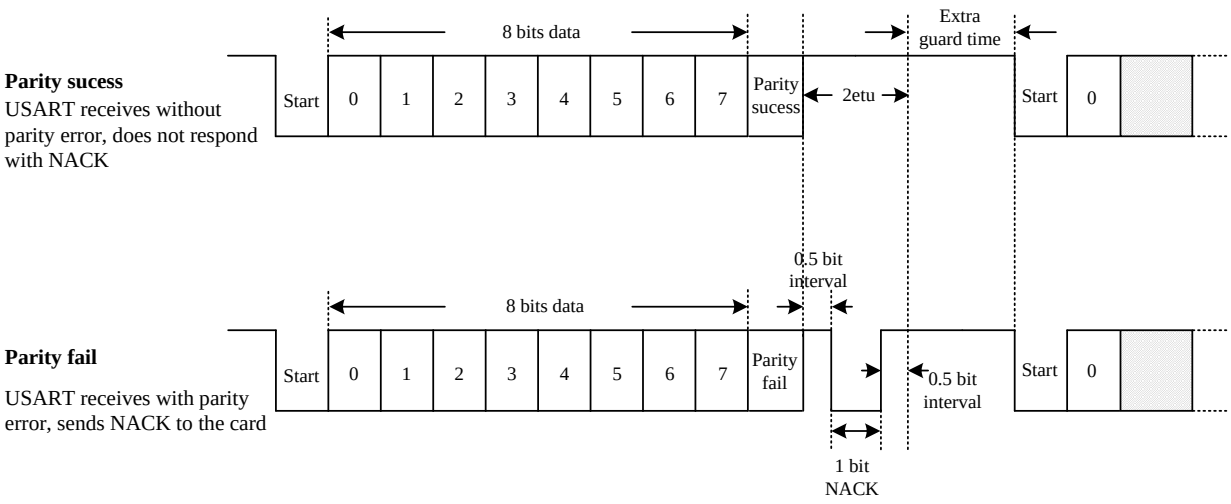
Configuration for 1.5 stop bits: STOPBIT[1:0] = 11. Enable NACK function: SCNAK = 1. The number of retries is configured in the SCRETRY[2:0] bit field. The clock frequency provided to the smartcard and the extra guard time can be configured in the USARTx_GTPR register.

If a parity error is detected during reception, a NACK is returned after completing the reception of one frame. If parity errors are still detected after the number of retries configured in the SCRETRY[2:0] bit field, the receiver stops retrying, and the PE flag is set to 1.

When an error occurs during reception, the RXNE flag is not set to 1, and a DMA receive request is not generated.

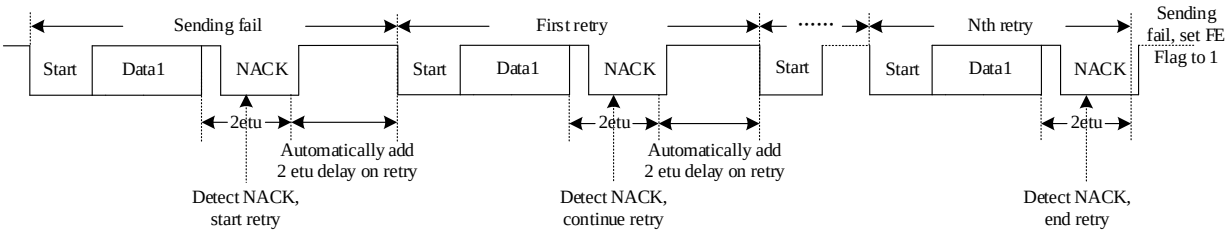
Timing diagram is as below:

Figure 30-18 USART NACK timing diagram during receiving



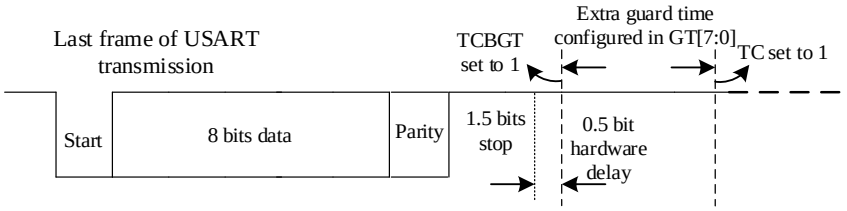
If a NACK is detected from the card when transmitting, retry the transmission. If NACK is still detected after the configured number of retries, set the FE flag to 1. When retrying, a 2-bit delay is automatically inserted between retry data. Timing diagram is shown below:

Figure 30-19 USART automatic retry timing diagram during sending



After completing the transmission of the last frame of data, the transmission TCBGT flag is immediately set to 1 before the guard time. The TC flag is set to 1 after the extra guard time. See the following figure:

Figure 30-20 TCBGT and TC flag setting timing (1.5 stop bits)



T = 1

When T = 1 compared to T = 0, the NACK function is not required (SCNAK bit is cleared to 0), and the stop bit can be configured to 0.5 bit (STOPBIT[1:0] = 01). The detection of BWT and CWT is implemented through the receive timeout function, which is configured by the RTOV[7:0] bit field to set the timeout duration.

The block length counter is used for end-of-block detection. When the BLKN[7:0] bit field is configured to 0x00, the USART will detect the end of the block after receiving the 4th character, setting the EBF flag to 1. If the EBIE bit is set to 1, an interrupt is generated.

The block length counter is reset when data to be transmitted is written (TXE = 0) or when the receiver is disabled (RE = 0).

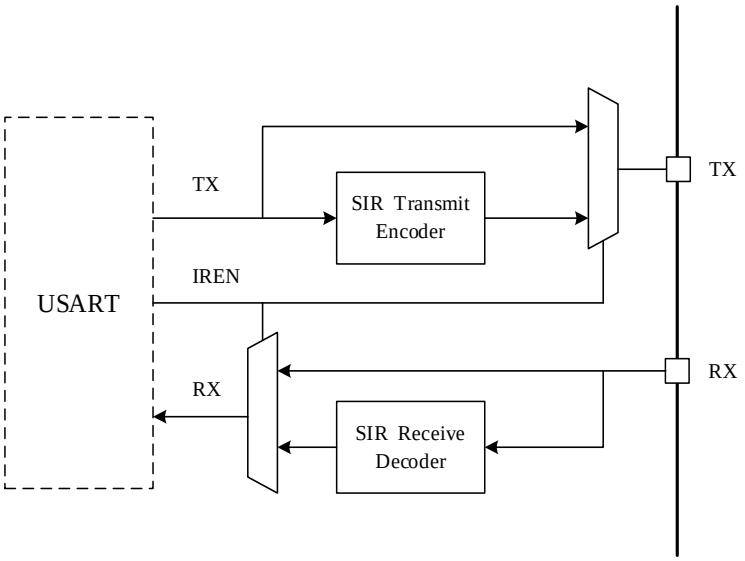
30.4.16 USART IrDA SIR ENDEC block

The IrDA SIR module supports communication with external low-power infrared

transceivers. The block diagram is shown below.

The SIR encoder modulates the data stream from the USART and outputs it to the external infrared transmitter. The SIR decoder demodulates the output signal from the infrared receiver into a serial data stream and transmits it to the USART.

Figure 30-21 USART IrDA SIR block diagram

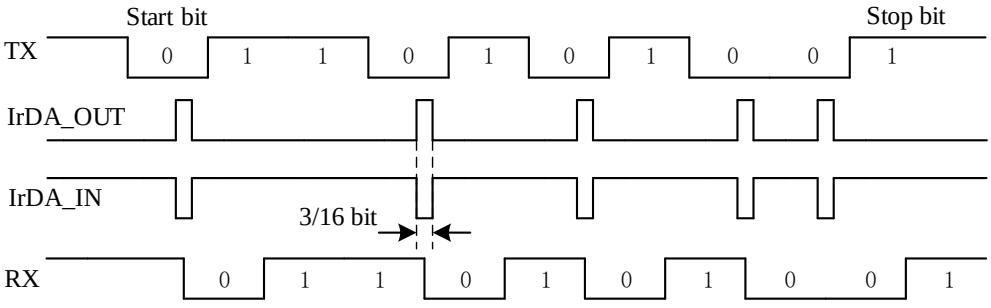


IrDA mode is selected by setting the IREN bit in the USARTx_CR3 register. In IrDA mode, the following bits must be kept cleared:

- STOPBIT, LINEN and CLKEN bits in the USARTx_CR2 register
- SCEN and HDEN bits in the USARTx_CR3 register

The SIR receive decoder interprets high levels as logic 1 and low-level pulses as logic 0. The polarity of the SIR transmit encoder output is opposite to that of the receiver decoder input, sending 0 as a high-level pulse and 1 as a low-level pulse. See the following figure:

Figure 30-22 IrDA SIR data modulation in normal mode



The communication baud rate can be configured through the USARTx_BRR register.

According to the IrDA SIR specification, USART supports only bit rates up to 115.2 Kbps for the SIR ENDEC.

- During receive decoding, the recognizable pulse width is related to the PSV[7:0] bit field of the USARTx_GTPR register.
  - The IrDA specification requires the acceptance of pulses greater than 1.41 µs. Pulses of width less than 1 PSV period are always rejected, but those of width greater than one and less than two periods may be accepted or rejected, those greater than two periods are accepted as a pulse.

    When the communication baud rate is 115200 bps, the standard requires the pulse width to be in the range of 1.41 to 2.23 µs, with a typical value of 1.63 µs. This pulse width is three times the period of the clock divided by the value configured in the PSV[7:0] bit field of the USARTx_GTPR register. (1.41 µs corresponds to three times the period of a 2.12 MHz clock, 2.23 µs corresponds to three times the period of a 1.42 MHz clock, and the typical value of 1.63 µs corresponds to three times the period of a 1.84 MHz clock.)
  - The IrDA SIR encoder/decoder does not work when PSV[7:0] = 0.
- In IrDA mode, only 16x oversampling is supported.

IrDA normal mode

- Transmitter
  - During transmission encoding, a high-level pulse with a width of 3/16 of a data bit represents 0, and a low level represents 1.
- Receiver
  - During receive decoding, a low-level pulse with a width of 3/16 of a data bit represents 0, and a high level represents 1.
  - During receive decoding, the recognizable pulse width is configured through the PSV[7:0] bit field of the USARTx_GTPR register. The recognizable pulse width is three times the period of the clock after divided by the value configured in PSV[7:0].

IrDA low-power mode

- Transmitter

  In low-power mode, the pulse width is not maintained at 3/16 of the bit period. Instead, the width of the pulse is 3 times the period of the clock after

USARTX_PCLK is divided by the value in the PSV[7:0] bit field.

According to the IrDA specification, to meet the minimum pulse width requirement, the frequency of USARTx_PCLK after division should be in the range of 1.42 MHz to 2.12 MHz.

- Receiver

  Receiving in low-power mode is similar to receiving in normal mode. For glitch detection the USART should discard pulses of duration shorter than 1 PSV. Pulses with a width of 1 to 2 PSV may be either accepted or filtered out. A valid low is accepted only if its duration is greater than 2 PSV.

30.4.17 USART communication using DMA

The USART is capable of performing continuous communications using the DMA. The DMA requests for Rx and Tx are generated independently.

Transmission using DMA

DMA mode can be enabled for transmission by setting DMAT bit in the USARTx_CR3 register. Data are loaded from a storage area configured using the DMA peripheral to the USARTx_TDR register whenever the TXE flag is set. To map a DMA channel for USART transmission, use the following procedure (x denotes the channel number):

1) Configure the relevant information for the channel in the DMA register DMA_CCx. See the *DMA channel configuration*.
2) Configure the total number of bytes to be sent in the DMA register DMA_CNDTRx.
3) Configure the memory area address as the source address in the DMA register DMA_CSARx. Write the address of the USARTx_TDR register to the DMA register DMA_CDARx. Each time the TXE bit is set to 1, data is loaded from the memory area to the USARTx_TDR register.
4) Configure the interrupt conditions according to the specific application.
5) Write 1 to the TCCF bit in the USARTx_ICR register to clear the TC flag in the USARTx_ISR register.
6) Set the EN bit in the DMA_CCx register to 1 to enable the channel.

An interrupt is generated on the corresponding DMA channel when the number of data transfers set in the DMA controller is reached.

In transmission mode, when the DMA has completed writing all the data to be sent, the TFx flag in the DMA_ISR register is set to 1. Additionally, it is able to confirm

whether the USART communication is completed by checking the TC flag in the USARTx_ISR register. The TC flag remains 0 during data transmission and is automatically set to 1 after the last character is transmitted.

Reception using DMA

DMA mode can be enabled for reception by setting the DMAR bit in the USARTx_CR3 register.

Data are loaded from the USARTx_RDR register to a storage area configured using the DMA peripheral whenever a data byte is received. To map a DMA channel for USART reception, use the following procedure (x denotes the channel number):

1) Configure the relevant information for the channel in the DMA register DMA_CCx. See the *DMA channel configuration.*

2) Configure the total number of bytes to be received in the DMA register DMA_CNDTRx.

3) Configure the address of the USARTx_RDR register in the DMA register DMA_CSARx. Write the memory area address as the source address to the DMA register DMA_CDARx. Each time the RXNE bit is set to 1, data is loaded from the USARTx_RDR register to the memory area.

4) Configure the interrupt conditions according to the specific application.

5) Set the EN bit in the DMA_CCx register to 1 to enable the channel.

An interrupt is generated on the corresponding DMA channel when the number of data transfers set in the DMA controller is reached.

Error flags and interrupts in DMA communication

In the USARTx_CR3 register, the DDRE bit can be configured to disable DMA in case of a receive error, including frame errors, parity errors, or noise.

By default, DDRE is 0, so DMA is not disabled in case of a receive error. The corresponding error flags are set to 1, but RXNE remains 0, and no DMA request is generated. The data with error will not be received into the USARTx_RDR register, but the next correct data can be received.

When DDRE is 1, the corresponding error flags and RXNE are both set to 1 in case of a receive error, and the data with error will be received into the USARTx_RDR register. However, DMA requests are masked. In this case, when a receive error occurs, the software must disable the DMA request (DMAR = 0) and clear RXNE to 0, and then

clear the error flags. After that DMA reception can continue.

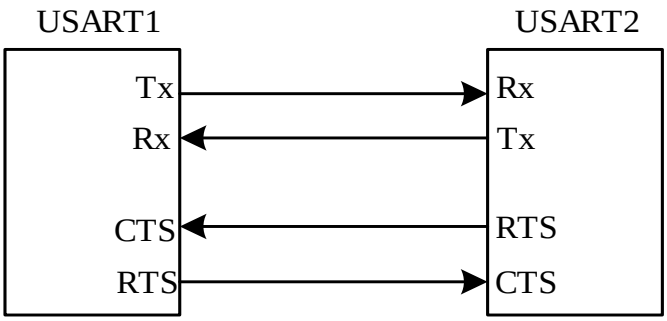
When an overrun error occurs, the ORE flag is set to 1. During this time, received data will only refresh the shift register and will not overwrite the last data in the RDR. Therefore, regardless of whether the DDRE bit is set to 1, RXNE must be cleared to 0, and the error flag must be cleared. After that data reception can continue.

30.4.18 RS232 hardware flow control and RS485 control

RS232 hardware flow control RTS/CTS is used for communication traffic control. RTS is an output pin, and CTS is an input pin. The connection relationship between the RTS and CTS pins is shown in the following figure.

CTS and RTS hardware flow control can be independently configured through the RTSE bit and CTSE bit in the USARTx_CR3 register. Setting these bits to 1 enables the respective flow control, while clearing them to 0 disables it.

Figure 30-23 Hardware flow control between 2 USARTs

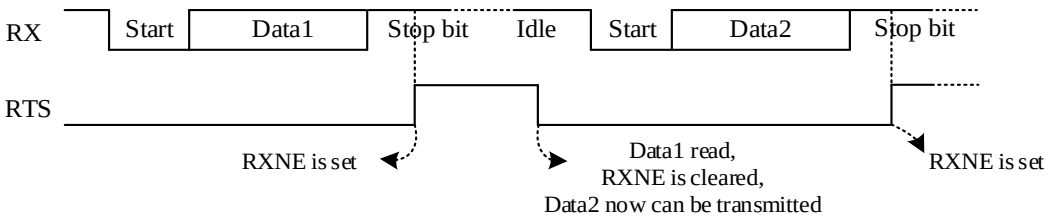


RS232 RTS flow control

If the RTS flow control is enabled (RTSE = 1), then RTS is asserted (tied low) as long as the USART receiver is ready to receive a new data (RXNE = 0). When the receive register is full (RXNE = 1), RTS is deasserted, indicating that the transmission is expected to stop at the end of the current frame.

An example of communication with RTS flow control enabled is as below:

Figure 30-24 RS232 RTS flow control



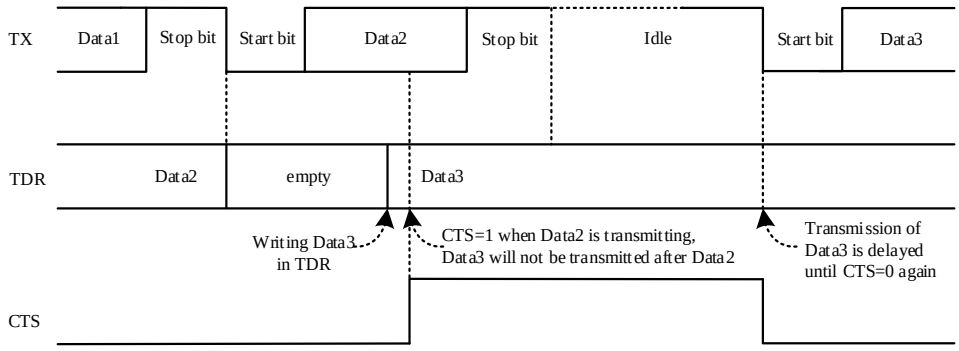
RS232 CTS flow control

If the CTS flow control is enabled (CTSE = 1), then the transmitter checks the CTS input before transmitting the next frame. The CTSIF status bit is automatically set by hardware as soon as the CTS input toggles. An interrupt is generated if the CTSIE bit in the USARTx_CR3 register is set.

- If CTS is asserted (tied low), then the next data can be transmitted.
- When CTS is deasserted during a transmission, the current transmission is completed before the transmitter stops. Data being written to the transmit data register is not sent out. When the CTS is asserted again, transmission resumes.

An example of communication with CTS flow control enabled is as below:

Figure 30-25 RS232 CTS flow control



Note: For correct behavior, CTS must be deasserted at least 5 USARTx_PCLK clock source periods before the end of the current character.

RS485 driver enable

The driver enable feature is enabled by setting bit DEM in the USARTx_CR3 register. This enables the user to activate the external transceiver control, through the DE (Driver Enable) signal.

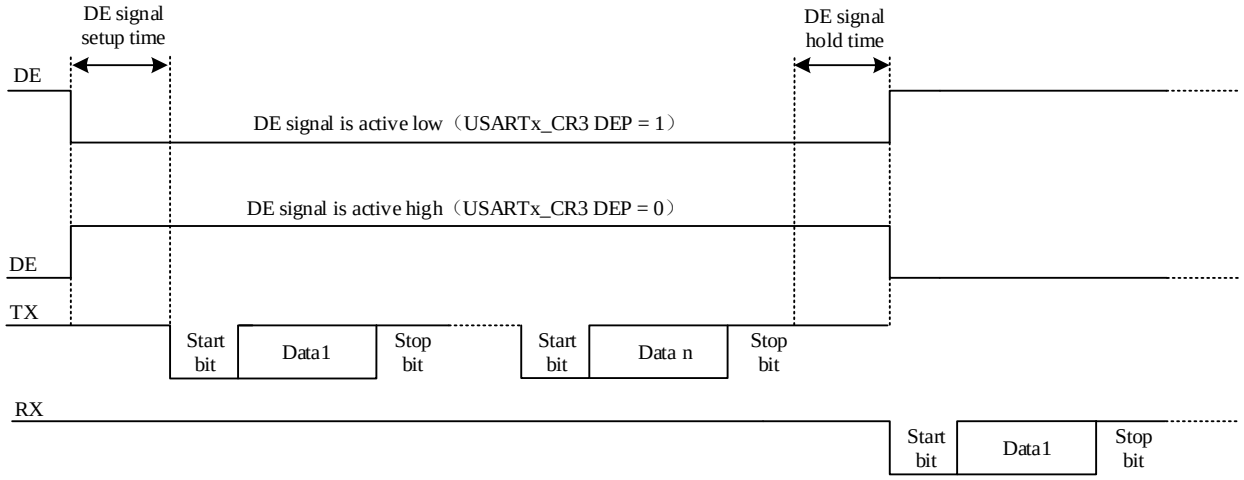
The setup time is the time between the activation of the DE signal and the beginning of the start bit. It is programmed using the DEST[4:0] bit field in the USARTx_CR1 register.

The hold time is the time between the end of the last stop bit, in a transmitted message, and the de-activation of the DE signal. It is programmed using the DEHT[4:0] bit field in the USARTx_CR1 register.

The polarity of the DE signal can be configured using the DEP bit in the USARTx_CR3 register. The DEST and DEHT are expressed in sample time units (1/8

or 1/16 bit time, depending on the oversampling rate).

Figure 30-26 RS485 DE signal transmission/reception control timing



30.5 USART interrupts

USART_IRQ is the USART interrupt request signal. The sources of the interrupt request signals are detailed in the following table:

Table 30-4 USART interrupt requests

Interrupt event	Event flag	Enable control bit	Interrupt clear method
Transmit data register empty	TXE	TXEIE	TXE cleared when a data is written in TDR
Receive data register not empty	RXNE	RXNEIE	RXNE cleared by reading RDR or by setting RXFLUQ bit
Transmission complete	TC	TCIE	TC cleared when a data is written in TDR or by setting TCCF bit
Transmission complete Before Guard Time	TCBGT	TCBGTIE	TCBGT cleared when a data is written in TDR or by setting TCBGTCF bit
CTS interrupt	CTSIF	CTSIE	CTSIF cleared by software by setting CTSCF bit
Idle line detected	IDLE	IDLEIE	IDLE cleared by setting IDLECF bit
Parity error	PE	PEIE	PE cleared by setting PECF bit
Overrun error detected	ORE	RXNEIE/EIE	ORE cleared by setting ORECF bit
Noise error	NOISE	EIE	NOISE cleared by setting NOISECF bit
Framing error	FE		FE flag cleared by setting FECF bit
Character match	CMF	CMIE	CMF cleared by setting CMCF bit

Interrupt event	Event flag	Enable control bit	Interrupt clear method
Receiver timeout	RTOF	RTOIE	RTOF cleared by setting RTOCF bit
LIN break	LBD	LBDIE	LBD cleared by setting LBDCF bit
End of Block	EBF	EBIE	EBF is cleared by setting EBCF bit

30.6 USART registers

The USART registers can only be accessed by words (32-bit).

Table 30-5 USARTx base address (x = 1, 2, 3, 4)

Peripheral	Base address
USART1	0x4001 3800
UART2	0x4000 4400
UART3	0x4000 4800
UART4	0x4000 4C00

30.6.1 USART control register 1 (USARTx_CR1)

Address offset: 0x00

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.				EBIE	RTOIE	DEST[4:0]					DEHT[4:0]				
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVS8	CMIE	RXMME	WL	RXW KUP	PEN	PTS	PEIE	TXEIE	TCIE	RXNE IE	IDLE IE	TE	RE	Res.	UE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

Bits	Name	Description
31:28	Reserved	Must be kept at reset value
27	EBIE	End of Block interrupt enable 0: Disable 1: Enable *Note: This function is not supported in UART2~4, this bit is reserved and must be kept at reset value.*
26	RTOIE	Receiver timeout interrupt enable 0: Disable 1: Enable *Note: This function is not supported in UART2~4, this bit is reserved and must be kept at reset value.*
25:21	DEST[4:0]	Driver enable setup time Used to configure the interval time between the RS485 transceiver

enable signal DE becoming active (high) and the start bit of transmission, measured in sampling clock cycles. Depending on the oversampling setting, this can be 1/8 bit or 1/16 bit time.
The polarity of the DE signal can be configured in the DEP bit of the USARTx_CR3 register. Example: For a communication baud rate of 9600 bps, the time per bit is 0.1042 ms. When configured for 16x oversampling, setting DEST[4:0] = 0x8 achieves a setup time of 0.5 bit time (0.0521 ms).
Note: These bits can only be written when the USART is disabled (UE = 0). This function is not supported in UART2~4, these bits are reserved and must be kept at reset value.

20:16	DEHT[4:0]	Driver enable hold time The time between the last stop bit of the last frame and the DE signal becoming inactive (low). The unit is in sampling clock cycles, which can be 1/8 or 1/16 bit time depending on the oversampling setting. The polarity of the DE signal can be configured in the DEP bit of the USARTx_CR3 register. If new data is written to the USARTx_TDR register during the DEHT period, the new data will only be transmitted after the DEHT and DEST periods have elapsed. Example: For a communication baud rate of 9600 bps, the time per bit is 0.1042 ms. When configured for 16x oversampling, setting DEHT[4:0] = 0x8 achieves a hold time of 0.5 bit time (0.0521 ms). *Note: These bits can only be written when the USART is disabled (UE = 0). This function is not supported in UART2~4, these bits are reserved and must be kept at reset value.*
15	OVS8	Oversampling mode 0: Oversampling by 16 1: Oversampling by 8 *Note: This bit can only be written when the USART is disabled (UE = 0). In IrDA and Smartcard modes, this bit must be kept at reset value.*
14	CMIE	Character match interrupt enable 0: Disable 1: Enable

Note: This function is not supported in UART2~4, this bit must be kept at reset value.

13	RXMME	RX mute mode enable Used to enter mute mode. The conditions for entering and exiting mute mode under different configurations are described in the *multiprocessor communication* chapter. 0: The receiver operates only in normal mode 1: The receiver is allowed to switch between mute mode and normal mode *Note: This function is not supported in UART2~4, this bit must be kept at reset value.*
12	WL	Word length This bit is used to determine the word length. 0: 8 data bits 1: 9 data bits *Note: This bit can only be written when the USART is disabled (UE = 0).*
11	RXWKUP	Receiver wakeup method 0: Idle line 1: Address mark *Note: This bit can only be written when the USART is disabled (UE = 0). This function is not supported in UART2~4, this bit must be kept at reset value.*
10	PEN	Parity control enable 0: Disable 1: Enable *Note: Parity checking occupies 1 data bit.* *In synchronous (SPI) mode, writing 1 to this bit does not enable the parity checking function.* *This bit can only be written when the USART is disabled (UE = 0).*
9	PTS	Parity selection This bit selects the odd or even parity when the parity

		generation/detection is enabled (PEN bit set). 0: Even parity 1: Odd parity *Note: This bit can only be written when the USART is disabled (UE = 0).* *In synchronous (SPI) mode, writing 1 to this bit does not enable the function.*
8	PEIE	PE interrupt enable 0: Disable 1: Enable
7	TXEIE	TXE interrupt enable 0: Disable 1: Enable
6	TCIE	Transmission complete interrupt enable 0: Disable 1: Enable
5	RXNEIE	RXNE interrupt enable 0: Disable 1: Enable
4	IDLEIE	IDLE interrupt enable 0: Disable 1: Enable *Note: This function is not supported in UART2~4, this bit must be kept at reset value.*
3	TE	Transmitter enable 0: Disable 1: Enable
2	RE	Receiver enable 0: Disable 1: Enable

1	Reserved	Must be kept at reset value
0	UE	USART enable When this bit is cleared, the USART configuration is kept, but the USARTx_TDR and USARTx_RDR are cleared, all USARTx_ISR status flags are reset. 0: Disable 1: Enable *Note: When UE = 0, DMA requests are also reset, so the DMA channel must be disabled before clearing UE to 0.*

30.6.2 USART control register 2 (USARTx_CR2)

Address offset: 0x04

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR[7:0]								RTOEN	Res.			MSB FIRST	DATA IVC	TXIVC	RXIVC
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	STOPBIT[1:0]		CLKEN	CPOL	CPHA	Res.		LBDIE	LBDL	ADDRM	Res.			
rw	rw	rw	rw	rw	rw	rw			rw	rw	rw				

Bits	Name	Description
31:24	ADDR[7:0]	Address of the USART node These bits give the address of the USART node or a character code to be recognized. They are used to wake up the MCU with address mark detection in multiprocessor communication during mute mode. The MSB of the character sent by the transmitter should be equal to 1. The ADDR[7:0] bits, in conjunction with the ADDRM bit in USARTx_CR2, implement the address matching mode. For more details, see the *ADDRM bit*. They can also be used for character detection during normal reception. In this case, the whole received character (8-bit) is compared to the ADDR[7:0] value and CMF flag is set on match. *Note: This bit can only be written when the receiver or USART is disabled (RE = 0 or UE = 0). This function is not supported in UART2~4, these bits must be kept at reset value.*

23	RTOEN	Receiver timeout enable 0: Disable 1: Enable After enabling the receive timeout, if no data is received within the time programmed in RTOV[23:0], the RTOF flag in the USARTx_ISR register is set to 1. *Note: This function is not supported in UART2~4, this bit must be kept at reset value.*
22:20	Reserved	Must be kept at reset value
19	MSBFIRST	Most significant bit first 0: Data is transmitted/received with data bit 0 first, following the start bit. 1: Data is transmitted/received with the MSB (bit 7/8) first, following the start bit *Note: This bit can only be written when the USART is disabled (UE = 0).*
18	DATAIVC	Binary data inversion 0: Logical data from the data register are send/received in positive/direct logic. (1 = H, 0 = L) 1: Logical data from the data register are send/received in negative/inverse logic. (1 = L, 0 = H) *Note: This bit can only be written when the USART is disabled (UE = 0).*
17	TXIVC	TX pin active level inversion 0: TX pin signal works using the standard logic levels (V_{DD} = 1, GND = 0) 1: TX pin signal values are inverted (V_{DD} = 0, GND = 1) *Note: This bit can only be written when the USART is disabled (UE = 0).*
16	RXIVC	TX pin active level inversion 0: RX pin signal works using the standard logic levels (V_{DD} =1, GND = 0)

1: RX pin signal values are inverted (V_{DD} = 0, GND = 1)

Note: This bit can only be written when the USART is disabled (UE = 0).

15	SWAP	Swap TX/RX pins 0: TX/RX pins are used as defined in standard pinout 1: The TX and RX pins functions are swapped This enables to work in the case of a cross-wired connection to another USART. *Note: This bit can only be written when the USART is disabled (UE = 0).*
14	LINEN	LIN mode enable 0: Disable 1: Enable *Note: This bit can only be written when the USART is disabled (UE = 0). This function is not supported in UART2~4, this bit must be kept at reset value.*
13:12	STOPBIT[1:0]	Stop bits These bits are used for programming the stop bits. 00: 1 stop bit 01: 0.5 stop bit 10: 2 stop bits 11: 1.5 stop bits *Note: This bit can only be written when the USART is disabled (UE = 0).*
11	CLKEN	Clock enable This bit enables the user to enable the CK pin 0: Disable 1: Enable *Note: This bit can only be written when the USART is disabled (UE = 0). This function is not supported in UART2~4, this bit must be kept at reset value.*
10	CPOL	Clock polarity 0: Steady low value on CK pin outside transmission window

		1: Steady high value on CK pin outside transmission window *Note: This bit can only be written when the USART is disabled (UE = 0). This function is not supported in UART2~4, this bit must be kept at reset value.*
9	CPHA	Clock phase 0: The first clock transition is the first data capture edge 1: The second clock transition is the first data capture edge *Note: This bit can only be written when the USART is disabled (UE = 0). This function is not supported in UART2~4, this bit must be kept at reset value.*
8:7	Reserved	Must be kept at reset value
6	LBDIE	Break detection interrupt enable 0: Disable 1: Enable *Note: This function is not supported in UART2~4, this bit must be kept at reset value.*
5	LBDL	Break detection length 0: 10 bits 1: 11 bits *Note: This bit can only be written when the USART is disabled (UE = 0). This function is not supported in UART2~4, this bit must be kept at reset value.*
4	ADDRM	Address mode select 0: 4-bit address detection, stored in ADDR[3:0] 1: For 8 bits data length, address is 7-bit, stored in ADDR[6:0] for 9 bits data length, address is 8-bit, stored in ADDR[7:0] *Note: This bit can only be written when the USART is disabled (UE = 0). This function is not supported in UART2~4, this bit must be kept at reset value.*
3:0	Reserved	Must be kept at reset value

30.6.3 USART control register 3 (USARTx_CR3)

Address offset: 0x08

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.							TCBGTIE	Res.				SCRETRY[2:0]			Res.
							rw					rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEP	DEM	DDRE	ORED	OBS	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	SCNAK	HDEN	IRLP	IREN	EIE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Name	Description
31:25	Reserved	Must be kept at reset value
24	TCBGTIE	Transmission complete before guard time, interrupt enable 0: Disable 1: Enable *Note: This function is not supported in UART2~4, this bit must be kept at reset value.*
23:20	Reserved	Must be kept at reset value
19:17	SCRETRY[2:0]	Smartcard auto-retry count This configuration specifies the number of automatic transmit retries before the FE bit is set during transmission, and the number of automatic receive retries before the PE bit is set during reception. When the USART is enabled (UE = 1), this bit field can only be written with 0x0 to disable both transmit and receive retries. 0x0: Disable retries: during transmission, automatic retransmission will not occur. During reception, if an error occurs, retries will not be enabled, and a parity error (PE =1) will be output directly. 0x1 to 0x7: Number of automatic retransmission attempts *Note: This bit can only be written when the USART is disabled (UE = 0).*
16	Reserved	Must be kept at reset value

15	DEP	RS485 driver enable polarity selection 0: DE signal is active high 1: DE signal is active low *Note: This bit can only be written when the USART is disabled (UE = 0). This function is not supported in UART2~4, this bit must be kept at reset value.*
14	DEM	RS485 driver enable mode 0: DE function is disabled 1: DE function is enabled. The DE signal is output on the RTS pin *Note: This bit can only be written when the USART is disabled (UE = 0). This function is not supported in UART2~4, this bit must be kept at reset value.*
13	DDRE	DMA disable on reception error 0: DMA is not disabled when a receive error occurs The corresponding error flag is set to 1, but RXNE remains 0, so no DMA request is generated. This also avoids an overrun, and the erroneous data is not received. However, the next correct data can still be received. 1: DMA is disabled when a receive error occurs The corresponding error flag and RXNE are both set to 1. No DMA request is generated until the error flag is cleared. This means that the software must first disable the DMA request (set DMAR to 0) and clear RXNE to 0, and then clear the error flag before reception can continue. *Note: This bit can only be written when the USART is disabled (UE = 0). The reception errors are: parity error, framing error or noise error.*
12	ORED	Overrun disable 0: Enable overrun detection If the received data has not been read (RXNE = 1) and new data is received, the overrun error flag ORE is set to 1. 1: Disable overrun detection, ORE flag will not be set to 1. *Note: This bit can only be written when the USART is disabled (UE = 0).*

11	OBS	One sample bit method enable 0: Each bit is sampled three times, and the sampled value is confirmed using a majority voting method 1: Each bit is sampled once, with no noise detection capability *Note: This bit can only be written when the USART is disabled (UE = 0). In Smartcard mode, this bit is reserved and must be kept at reset value.*
10	CTSIE	CTS interrupt enable 0: Disable 1: Enable
9	CTSE	CTS enable 0: Disable 1: Enable *Note: This bit can only be written when the USART is disabled (UE = 0).*
8	RTSE	RTS enable 0: Disable 1: Enable *Note: This bit can only be written when the USART is disabled (UE = 0).*
7	DMAT	DMA enable transmitter 0: Disable 1: Enable
6	DMAR	DMA enable receiver 0: Disable 1: Enable
5	SCEN	Smartcard mode enable 0: Disable 1: Enable *Note: This bit can only be written when the USART is disabled (UE = 0). Smartcard mode is not supported in UART2~4, this bit must be kept at reset value.*

4	SCNAK	Smartcard NACK enable 0: When a parity error occurs during reception, no NACK is sent. During transmission, NACK is not detected 1: When a parity error occurs during reception, a NACK is sent. During transmission, NACK is detected *Note: This bit can only be written when the USART is disabled (UE = 0). Smartcard mode is not supported in UART2~4, this bit must be kept at reset value.*
3	HDEN	Half-duplex selection 0: Disable 1: Enable *Note: This bit can only be written when the USART is disabled (UE = 0).*
2	IRLP	IrDA low-power 0: Normal mode 1: Low-power mode *Note: This bit can only be written when the USART is disabled (UE = 0). IrDA mode is not supported in UART2~4, this bit must be kept at reset value.*
1	IREN	IrDA mode enable 0: Disable 1: Enable *Note: This bit can only be written when the USART is disabled (UE = 0). IrDA mode is not supported in UART2~4, this bit must be kept at reset value.*
0	EIE	Error interrupt enable If a frame error, overrun error, or noise error occurs (FE = 1, ORE = 1, or NOISE = 1 in the USARTx_ISR register), this bit is set to 1, which generates an interrupt. 0: Disable 1: Enable

30.6.4 USART baud rate register (USARTx_BRR)

Address offset: 0x0C

Reset value: 0x0000 0000

Note: This register can only be written when the USART is disabled (UE = 0).

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

Bits	Name	Description
31:16	Reserved	Must be kept at reset value
15:0	BRR[15:0]	USART baud rate See *USART baud rate generation.*

30.6.5 USART guard time and prescaler register (USARTx_GTPR)

Address offset: 0x10

Reset value: 0x0000 0000

Note: This register can only be written when the USART is disabled (UE = 0).

UART2 to UART4 do not support the receive timeout feature. This register is reserved and is forced by hardware to "0x00000000".

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

Bits	Name	Description
31:16	Reserved	Must be kept at reset value
15:8	GT[7:0]	Guard time value

This bit field is used to set the additional guard time (in units of 1 data bit time) in Smartcard mode.
After the transmission is complete, the TC flag is not immediately set to 1. Instead, the TCBGT flag is first set to 1, and after the additional guard time has elapsed, the TC flag is set to 1.
Note: This bit can only be written when the USART is disabled (UE = 0).

7:0	PSV[7:0]	Prescaler value In IrDA mode, the value of PSV[7:0] is used to divide the USARTx_PCLK clock, which then serves as the operating clock for both IrDA normal mode and IrDA low-power mode. 00000000: Reserved - do not program this value 00000001: Divides the source clock by 1 00000010: Divides the source clock by 2 00000011: Divides the source clock by 3 ⋮ 11111111: Divides the source clock by 255 In Smartcard mode, PSV[4:0] is the value of the Smartcard clock prescaler, used to divide the USARTx_PCLK clock, which then serves as the clock output to the Smartcard. The value set in the register (5 valid bits) is multiplied by 2 to obtain the clock source prescaler factor for the Smartcard. 00000: Reserved - do not program this value 00001: Divides the source clock by 2 00010: Divides the source clock by 4 00011: Divides the source clock by 6 ⋮ 11111: Divides the source clock by 62 *Note: This bit can only be written when the USART is disabled (UE = 0).* *In Smartcard mode, the PSV[7:5] bit fields are automatically cleared to 0 by hardware.*

30.6.6 USART receiver timeout register (USARTx_RTO)

Address offset: 0x14

Reset value: 0x0000 0000

Note: UART2 to UART4 do not support the receive timeout feature. This register is reserved and is forced by hardware to "0x00000000".

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BLKN[7:0]								RTOV[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
RTOV[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Name	Description
31:24	BLKN[7:0]	Block Length This bit field is used to configure the expected block length for the Smartcard T =1 protocol. When the BLKN bit field is set to the minimum value "0x00," the USART detects the end of the block after receiving the 4th character, and the EBF flag is set to 1. In Smartcard mode, the block length counter is reset when the transmit register is empty (TXE = 0). *Note: When Smartcard mode is enabled, the block length counting function is automatically enabled. In modes other than Smartcard mode, programming this bit field has no effect on the block length counting function.* *This function is not supported in UART2~4, this bit must be kept at reset value.*
23:0	RTOV[23:0]	Receiver timeout value This bit field provides the receiver's timeout period (in units of 1 data bit). If no data is received within the time configured in RTOV, the RTOF flag is set to 1. If the interrupt is enabled (RTOIE =1), an interrupt is generated. In Smartcard mode, this bit field is used to configure CWT and BWT (see: *USART Smartcard mode*). *Note: The RTOV can be written in real-time. If the new value is less than or equal to the current counter value, the RTOF flag is set to 1.* *This function is not supported in UART2~4, this bit must be*

kept at reset value.

30.6.7 USART request register (USARTx_RQR)

Address offset: 0x18

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.											TX FLUQ	RX FLUQ	MUTEQ	BK SENDQ	Res.
											w	w	w	w	

Bits	Name	Description
31:5	Reserved	Must be kept at reset value
4	TXFLUQ	Transmit data flush request Writing 1 to this bit sets the TXE flag. This enables to discard the transmit data.
3	RXFLUQ	Receive data flush request Writing 1 to this bit empties the entire receive data and clears the bit RXNE. This enables to discard the received data without reading them, and avoid an overrun condition.
2	MUTEQ	Mute mode request Writing 1 to this bit puts the USART in mute mode and sets the RWU flag. *Note: This function is not supported in UART2~4, this bit must be kept at reset value.*
1	BKSENDQ	Send break request Writing 1 to this bit sets the BKSEND flag and request to send a BREAK on the line. *Note: In synchronous mode (SPI) and Smartcard mode, writing 1 to this bit does not send a break frame.* *This function is not supported in UART2~4, this bit must be kept at reset value.*

0	Reserved	Must be kept at reset value

30.6.8 USART interrupt and status register (USARTx_ISR)

Address offset: 0x1C

Reset value: 0x0000 00C0

Note: In Smartcard mode, when toggling the UE bit (UE bit goes from 0 to 1), the TCBGT bit is set.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.						TCBGT	Res.		REACK	TEACK	Res.	RWU	BKSEND	CMF	BUSY
						r			r	r		r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			EBF	RTOF	CTS	CTSIF	LBD	TXE	TC	RXNE	IDLE	ORE	NOISE	FE	PE
			r	r	r	r	r	r	r	r	r	r	r	r	r

Bits	Name	Description
31:26	Reserved	Must be kept at reset value
25	TCBGT	Transmission complete before guard time flag This function used in Smartcard mode: TCBGT bit is set when the last data written in the USARTx_TDR has been transmitted correctly (the smartcard did not send back any NACK) out of the shift register. This bit is cleared by software, by writing 1 to the TCBGTCF in the USARTx_ICR register or by a write to the USARTx_TDR register. 0: Transmission is not complete or transmission is complete unsuccessfully (NACK is received from the smartcard) 1: Transmission is complete successfully (before Guard time completion and there is no NACK from the smartcard) *Note: This function is not supported in UART2~4, this bit must be kept at reset value.*
24:23	Reserved	Must be kept at reset value
22	REACK	Receive enable acknowledge flag 0: The receiver is not ready for reception

		1: The receiver is ready for reception
21	TEACK	Transmit enable acknowledge flag When TE bit is cleared, the transmitter is disabled, this bit is cleared by hardware. 0: The transmitter is not ready for sending 1: The transmitter is ready for sending
20	Reserved	Must be kept at reset value
19	RWU	Receiver wakeup from mute mode This bit indicates if the USART is in mute mode, it is cleared/set by hardware. When wakeup from mute mode this bit is cleared, when in mute mode, this bit is set. 0: Receiver in active mode 1: Receiver in mute mode *Note: This function is not supported in Smartcard mode and UART2~4, this bit must be kept at reset value.*
18	BKSEND	Send break flag Writing 1 to the BKSENDQ bit in the USARTx_RQR register generates a request to send a break frame, setting this bit to 1. This bit is automatically cleared to 0 during the stop bit of the break frame. 0: The break frame has been sent 1: There is a request to send a break frame *Note: This function is not supported in UART2~4, this bit must be kept at reset value.*
17	CMF	Character match flag This bit is set to 1 by hardware when a character defined by ADDR[7:0] is received. It is cleared to 0 by writing 1 to the CMCF bit in the USARTx_ICR register. 0: No character match detected 1: Character match detected *Note: This function is not supported in UART2~4, this bit must be kept at reset value.*

16	BUSY	Busy flag When there is data being received on the RX line, this bit is set to 1 from the time the start bit is detected. This bit is cleared to 0 at the end of reception, regardless of whether the reception was successful. 0: Idle state 1: Receiving data
15:13	Reserved	Must be kept at reset value
12	EBF	End of block flag For smartcard T = 1, this bit is set to 1 after a complete block (Block) is received. When the total number of bytes received (including the start field) is greater than or equal to BLKN + 4, it is considered that the end of the block has been detected, and this bit is set to 1. This bit is cleared to 0 by writing 1 to the EBCF bit in the USARTx_ICR register. 0: Not at the end of the block 1: At the end of the block *Note: This function is not supported in UART2~4, this bit must be kept at reset value.*
11	RTOF	Receiver timeout flag When the receive timeout is enabled (RTOEN = 1). When the timeout value, programmed in the RTOV, register has elapsed, without any communication, this bit is set to 1. It is cleared by software by writing 1 to the RTOCF bit in the USARTx_ICR register. In Smartcard mode, the timeout period should be configured according to the CWT or BWT time (see: *Smartcard chapter*). 0: Timeout value not reached 1: Timeout value reached without any data reception *Note: This function is not supported in UART2~4, this bit must be kept at reset value.* *When RE = 0, the receive timeout function is immediately disabled.*
10	CTS	CTS flag This bit state is the inverse of the CTS input pin state.

		0: CTS pin is high 1: CTS pin is low
9	CTSIF	CTS interrupt flag When CTS is enabled (CTSE bit is set to 1), this bit is set to 1 when the CTS input experiences a level change. This bit is cleared to 0 by writing 1 to the CTSCF bit in the USARTx_ICR register. 0: CTS pin level not changed 1: CTS pin level changed
8	LBD	Break detection flag This bit is set by hardware when the LIN break is detected. 0: LIN Break not detected 1: LIN Break detected *Note: This function is not supported in UART2~4, this bit must be kept at reset value.*
7	TXE	Transmit data register empty When the contents of the USARTx_TDR register have been transmitted, the TXE bit is set to 1. Writing to the USARTx_TDR register clears this bit. Additionally, writing 1 to the TXFLUQ bit in the USARTx_RQR register will clear the transmit data and set the TXE flag to 1. 0: USARTx_TDR is not empty 1: USARTx_TDR is empty
6	TC	Transmission complete This bit indicates that the last data written in the USARTx_TDR has been transmitted out of the shift register. It is set by hardware when the transmission of a frame containing data is complete and when TXE is set. TC bit is cleared by software, by writing 1 to the TCCF in the USARTx_ICR register or by writing to the USARTx_TDR register. 0: Transmission is not complete 1: Transmission is complete
5	RXNE	Read data register not empty When data is received, the RXNE bit is set to 1.

Reading the USARTx_RDR register clears this bit. Additionally, writing 1 to the RXFLUQ bit in the USARTx_RQR register will clear the received data and reset the RXNE flag to 0.

0: Data is not received

1: Received data is ready to be read

4 IDLE

IDLE line detected

Writing 1 to the IDLECF bit in the USARTx_ICR register clears this bit.

0: No Idle line is detected

1: Idle line is detected

Note: This function is not supported *in UART2~4, this bit must be kept at reset value.*

After this flag is cleared, the IDLE bit will be set to 1 again only after a valid start bit is received.

3 ORE

Overrun error

This bit is set by hardware when the data currently being received in the shift register is ready to be transferred into the USARTx_RDR register while RXNE = 1 or the previous DMA read request has not been processed.

It is cleared by a software, writing 1 to the ORECF, in the USARTx_ICR register.

0: No overrun error

1: Overrun error is detected

Note: When the ORED bit in the USARTx_CR3 register is set to 1, overrun error detection is disabled, and this bit is cleared (no overrun detection).

2 NOISE

Noise detection flag

This bit is set by hardware when noise is detected on a received frame. It is cleared by software, writing 1 to the NOISECF bit in the USARTx_ICR register.

0: No noise is detected

1: Noise is detected

Note: When the line is noise-free, single-sample mode (OBS = 1) can be used to improve the USART's receive tolerance. In this mode, there is no noise detection, and the NOISE flag is

not output.

1	FE	Framing error This bit is set by hardware when a de-synchronization, excessive noise or a break character is detected. It is cleared by software, writing 1 to the FECF bit in the USARTx_ICR register. 0: No Framing error is detected 1: Framing error or break character is detected
0	PE	Parity error This bit is set by hardware when a parity error occurs in receiver mode. It is cleared by software, writing 1 to the PECF in the USARTx_ICR register. 0: No parity error 1: Parity error

30.6.9 USART interrupt flag clear register (USARTx_ICR)

Address offset: 0x20

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.														CMCF	Res.
														w	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			EBCF	RTOCF	Res.	CTSCF	LBDCF	TCBGTCF	TCCF	Res.	IDLECF	ORECF	NOISECF	FECF	PECF
			w	w		w	w	w	w		w	w	w	w	w

Bits	Name	Description
31:18	Reserved	Must be kept at reset value
17	CMCF	Character match clear flag Writing 1 to this bit clears the CMF flag in the USARTx_ISR register. *Note: This function is not supported in UART2~4, this bit must be kept at reset value.*

16:13	Reserved	Must be kept at reset value
12	EBCF	End of block clear flag Writing 1 to this bit clears the EBF flag in the USARTx_ISR register. *Note: This function is not supported in UART2~4, this bit must be kept at reset value.*
11	RTOCF	Receiver timeout clear flag Writing 1 to this bit clears the RTOF flag in the USARTx_ISR register. *Note: This function is not supported in UART2~4, this bit must be kept at reset value.*
10	Reserved	Must be kept at reset value
9	CTSCF	CTS clear flag Writing 1 to this bit clears the CTSIF flag in the USARTx_ISR register.
8	LBDCF	Break detection clear flag Writing 1 to this bit clears the LBD flag in the USARTx_ISR register. *Note: This function is not supported in UART2~4, this bit must be kept at reset value.*
7	TCBGTCF	Transmission complete before Guard time clear flag Writing 1 to this bit clears the TCBGT flag in the USARTx_ISR register. *Note: This function is not supported in UART2~4, this bit must be kept at reset value.*
6	TCCF	Transmission complete clear flag Writing 1 to this bit clears the TC flag in the USARTx_ISR register.
5	Reserved	Must be kept at reset value
4	IDLECF	Idle line detected clear flag

		Writing 1 to this bit clears the IDLE flag in the USARTx_ISR register. *Note: This function is not supported in UART2~4, this bit must be kept at reset value.*
3	ORECF	Overrun error clear flag Writing 1 to this bit clears the ORE flag in the USARTx_ISR register.
2	NOISECF	Noise detected clear flag Writing 1 to this bit clears the NOISE flag in the USARTx_ISR register.
1	FECF	Framing error clear flag Writing 1 to this bit clears the FE flag in the USARTx_ISR register.
0	PECF	Parity error clear flag Writing 1 to this bit clears the PE flag in the USARTx_ISR register.

30.6.10 USART receive data register (USARTx_RDR)

Address offset: 0x24

Reset value: 0x0000 0000

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

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

Bits	Name	Description
31:9	Reserved	Must be kept at reset value
8:0	RDR[8:0]	Receive data value When receiving with the parity enabled, the value read in the MSB bit is the received parity bit.

30.6.11 USART transmit data register (USARTx_TDR)

Address offset: 0x28

Reset value: 0x0000 0000

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

Bits	Name	Description
31:9	Reserved	Must be kept at reset value
8:0	TDR[8:0]	Transmit data value When transmitting with the parity enabled, the value written in the MSB has no effect because it is replaced by the parity.

31 Low-power universal asynchronous receiver transmitter (LPUART)

31.1 Introduction

The LPUART supports both full-duplex and half-duplex communication modes. It integrates a fractional baud rate generator, allowing the baud rate to be configured over a wide range.

The LPUART can achieve a communication baud rate of 9600 bps using a low-speed 32.768 KHz LXTAL clock source. When using a higher frequency clock source, it can support higher communication baud rates.

The LPUART can operate in low-power modes and supports wake up from Stop mode. It also supports DMA communication, single-wire half-duplex communication, and hardware flow control.

The system provides 2 LPUARTs (LPUART1/2).

31.2 LPUART main features

- Programmable baud rate
  - From 300 bps to 9600 bps using a 32.768 KHz clock source
  - Higher baud rates can be achieved by using a higher frequency clock source, up to16 Mbps
  - Fractional baud rate generator
- Dual clock domain with dedicated kernel clock for peripherals: PCLK, LXTAL, RCH, and SYSCLK
- Supporting wake up from Stop mode, and the method is configurable
- Character format
  - Data word length: 8 or 9 bits
  - Stop bits: 1 or 2 bits
  - Odd parity, even parity, no parity
  - Data order with MSB-first or LSB-first shifting
- Single-wire half-duplex communications

- Hardware flow control for modem
- Swappable Tx/Rx pin configuration
- Separate signal polarity control for transmission and reception
- DMA communications

31.3 LPUART functional description

31.3.1 LPUART block diagram

Figure 31-1 LPUARTx block diagram (x = 1, 2)

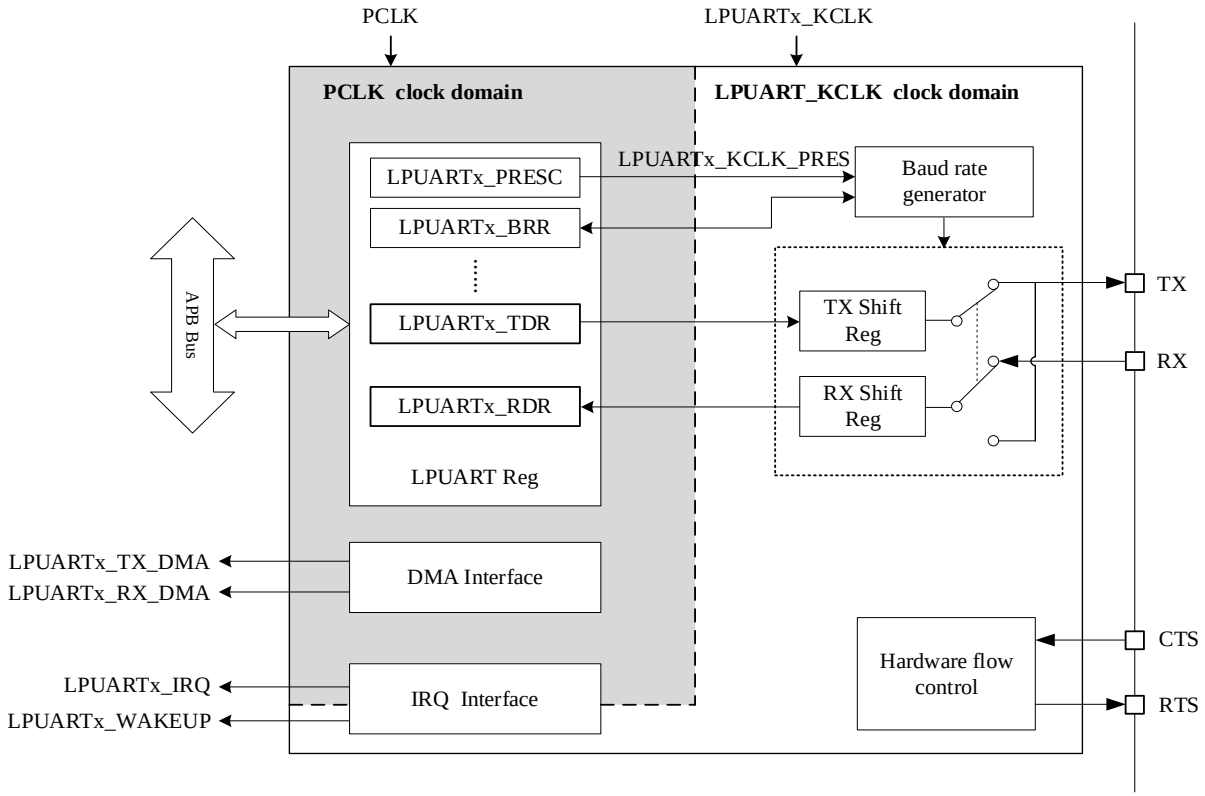


Table 31-1 LPUART internal signals

Signal	Type	Description
PCLK	Input	Clock source for the PCLK domain
LPUARTx_KCLK	Input	Clock source for the LPUARTx_KCLK domain
LPUARTx_TX_DMA	Output	LPUART DMA request signal of transmission
LPUARTx_RX_DMA	Output	LPUART DMA request signal of reception
LPUARTx_IRQ	Output	LPUART interrupt request signal
LPUARTx_WAKEUP	Output	LPUART wakeup request signal

As shown in the table, the LPUART operates in two clock domains:

- The clock source for the PCLK clock domain is the same as the APB bus clock.
- The LPUARTx_KCLK clock source is provided by the RCC and can be configured as PCLK, LXTAL, RCH, or SYSCLK. When the clock source is configured as RCH or LXTAL, it supports waking up from Stop mode.

31.3.2 LPUART pins

LPUART bidirectional communications require a minimum of two pins: Receive Data In (RX) and Transmit Data Out (TX). The TX and RX pin swap functionality is supported, configured in the SWAP bit of the LPUARTx_CR2 register.

The following pins are required in RS232 Hardware flow control mode:

- CTS (Clear To Send)

  The pin is input, when driven high, this signal blocks the data transmission at the end of the current transfer.

- RTS (Request To Send)

  The pin is output, when it is low, this signal indicates that the LPUART is ready to receive data.

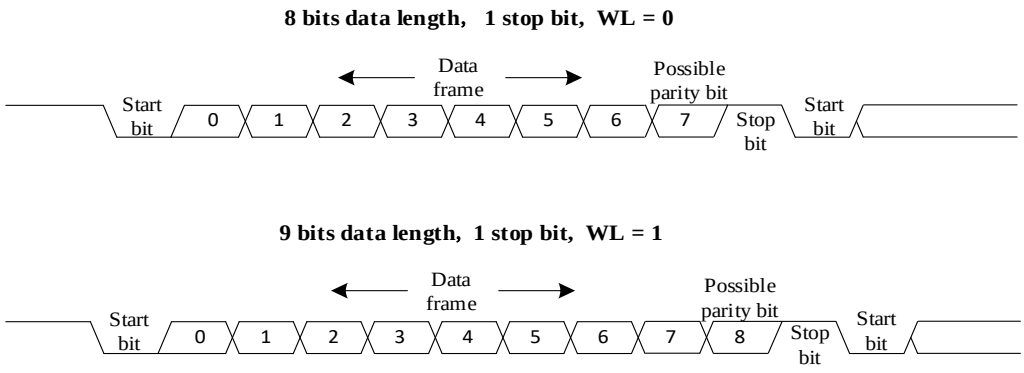
31.3.3 LPUART character description

The word length can be programmed through the WL bit in the LPUARTx_CR1 register. See the Figure: *Character format diagram*.

- 8 bit word length: WL = 0
- 9 bit word length: WL = 1

By default, the TX and RX pins are low during the start bit and high during the stop bit. The polarity of the transmission and reception signals can be independently configured in the TXIVC and RXIVC bits of the LPUARTx_CR2 register. See the following diagram:

Figure 31-2 Character format diagram



31.3.4 LPUART parity control

Table 31-2 LPUART parity control

Word length WL	Parity control PEN	LPUART character format
0	0	| start bit | 8 bit word length | stop bit |
	1	| start bit | 7 bit word length | parity bit | stop bit |
1	0	| start bit | 9 bit word length | stop bit |
	1	| start bit | 8 bit word length | parity bit | stop bit |

Even parity

When even parity (PTS bit is 0) is enabled, the total number of "1"s in a character (including the parity bit) is even.

Odd parity

When odd parity (PTS bit is 1) is enabled, the total number of "1"s in a character (including the parity bit) is odd.

Receiving parity check

The parity check can be enabled by setting the PEN bit in the LPUARTx_CR1 register. If a parity error occurs, the PE flag in the LPUARTx_ISR register is set to 1. If the PEIE bit in the LPUARTx_CR1 register is set to 1, an interrupt is generated. Writing 1 to the PECF bit in the LPUARTx_ICR register clears the PE flag.

31.3.5 LPUART transmitter

The transmitter can send data words of either 8 or 9 bits, depending on the WL bit status. The transmitter can be enabled by setting the Transmit Enable (TE) bit to 1. The

data in the transmit shift register is output on the TX pin.

Character transmission

The transmitter can be enabled by setting the TE bit in the LPUARTx_CR1 register to 1.

Writing the data to be transmitted into the LPUARTx_TDR register initiates the transmission. During a transmission, data shifts out the least significant bit first (default configuration) on the TX pin.

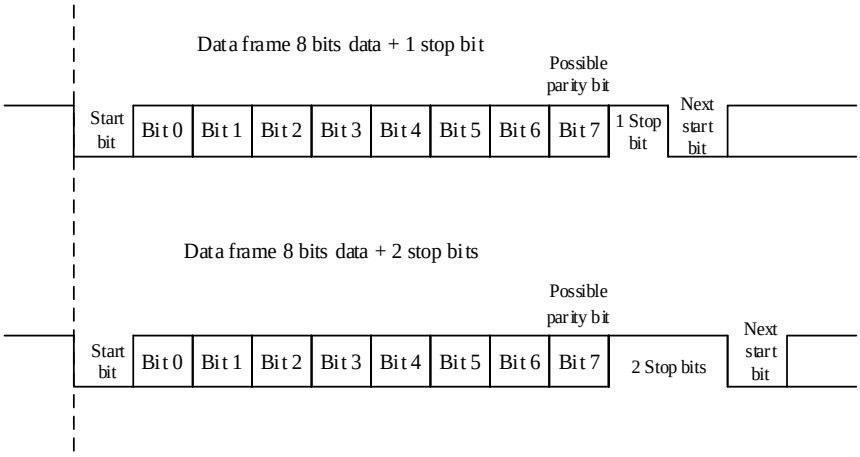
Each character is transmitted starting with the start bit and ending with the stop bit.

Stop bit

The number of stop bits to be transmitted with every character can be programmed by STOPBIT[1:0] in the LPUARTx_CR2 register, as 1 bit or 2 bits.

The following are examples of different stop bit configurations:

Figure 31-3 Configurable stop bits



Character transmission procedure

1) Select the desired baud rate using the LPUARTx_BRR register.
2) Program the WL bits in the LPUARTx_CR1 register to define the word length.
3) Program the STOPBIT bit in the LPUARTx_CR2 register to define the number of stop bits.
4) Enable the LPUART by writing the UE bit in the LPUARTx_CR1 register to 1.
5) Select DMA enable (DMAT) in the LPUARTx_CR3 register if DMA communication must take place. Refer to the instructions in *LPUART communication using DMA* to configure the DMA registers.
6) Set the TE bit in the LPUARTx_CR1 register to enable the transmitter. Wait for

the TEACK bit to be set, indicating that the transmitter is ready to send.

7) Write the data to be sent into the LPUARTx_TDR register. Writing data to LPUARTx_TDR will clear the TXE flag, indicating that the LPUARTx_TDR register is full.

8) When the last data to be sent is written to the LPUARTx_TDR register, wait until the TC flag in the LPUARTx_ISR register is set. The TC flag being set indicates that the last character has been successfully sent.

Character transmission process

When writing 1 character to the LPUARTx_TDR register, the TXE flag is cleared to 0.

When the transmit data register LPUARTx_TDR is empty, the TXE flag is automatically set to 1, indicating that:

- The data has been moved from the LPUARTx_TDR register to the shift register, and the transmission of data begins.
- Data can be written to the LPUARTx_TDR register without overwriting the previous data.

If the TXE interrupt enable TXEIE bit is set to 1, an interrupt will be generated.

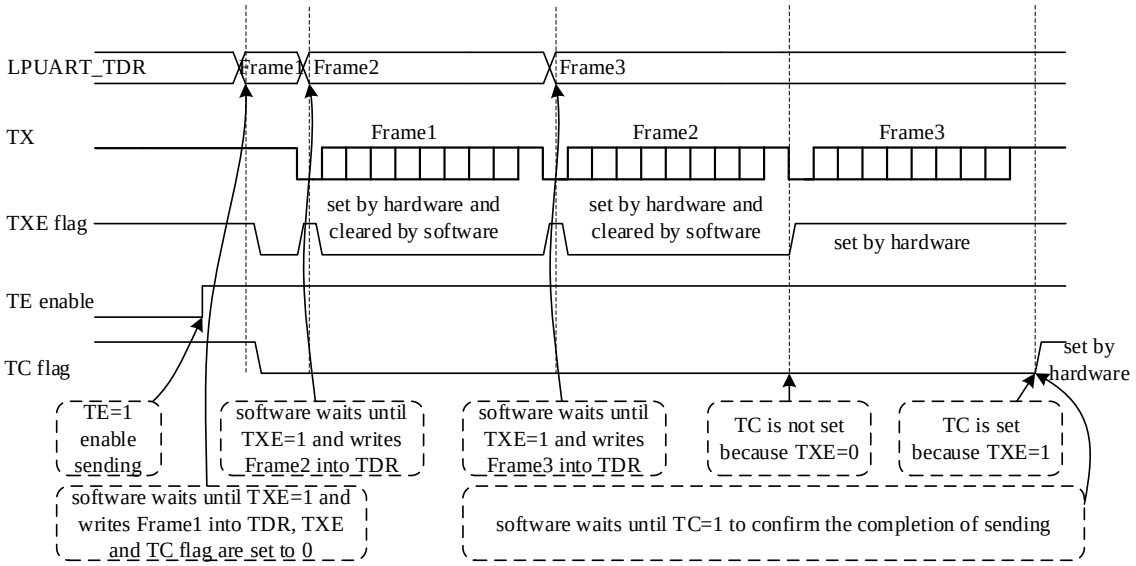
When the transmitter is not enabled (TE = 0), writing to the LPUARTx_TDR register clears the TXE bit to 0. Once data transmission starts, the TXE bit is immediately set to 1, indicating that the next data can be written.

When all data has been written to the LPUARTx_TDR register and all data on the transmit line has been completely sent, the TC flag in the LPUARTx_ISR register is set to 1.

An interrupt is generated if the TCIE bit in the LPUARTx_CR1 register is set to 1.

The actions of TC and TXE during transmission are shown in the following figure:

Figure 31-4 TC/TXE behavior when transmitting



31.3.6 LPUART receiver

The LPUART can receive data words of either 8 or 9 bits depending on the WL bit in the LPUARTx_CR1 register.

Start bit detection

In the LPUART, the start bit is detected when a falling edge occurs on the Rx line, followed by a sample taken in the middle of the start bit to confirm that it is still ‘0’. If the start sample is at ‘0’, the receiver continues to sample all incoming bits normally. Else, the start bit is discarded and the receiver waits for a new start bit.

Character reception

To receive a character, follow the sequence below:

1) Select the desired baud rate using the LPUARTx_BRR register.
2) Program the WL bits in the LPUARTx_CR1 register to define the word length.
3) Program the STOPBIT bit in the LPUARTx_CR2 register to define the number of stop bits.
4) Enable the LPUART by writing the UE bit in the LPUARTx_CR1 register to 1.
5) Select DMA enable (DMAR) in the LPUARTx_CR3 register if DMA communication must take place. Refer to the instructions in *LPUART communication using DMA* to configure the DMA registers.
6) Set the RE bit in the LPUARTx_CR1 register to enable the receiver.
7) Wait for the REACK bit to be set, indicating that the receiver is ready to receive.

When a character is received:

- When not using DMA communication:

  The RXNE flag is set to indicate that the content of the shift register is transferred to the LPUARTx_RDR register.

  Clearing the RXNE flag is done by performing a software read from the LPUART_RDR register. The RXNE flag can also be cleared by programming RXFLUQ bit to '1' in the LPUARTx_RQR register.

- When using DMA communication:

  Since the LPUARTx_RDR register has only 1 byte of space, the RXNE flag is set every time when a character is received. The RXNE flag is cleared when the DMA reads the LPUARTx_RDR register.

- The error flags can be set if a frame error, an overrun or parity error was detected during reception.

Overrun error

The overrun error detection is enabled by default. When an overrun error is detected, the ORE flag is set. An interrupt is generated if either the RXNEIE or the EIE bit in the LPUARTx_CR1 register is set.

The ORE bit is reset by setting the ORECF bit in the LPUARTx_ICR register.

The RXNE flag is set after every byte reception. An overrun error occurs if a character is received and RXNE has not been reset, or the previous DMA request has not been serviced. Data can not be transferred from the shift register to the LPUARTx_RDR register until the RXNE bit is cleared.

When an overrun error occurs:

- The ORE flag is set.
- The shift register is overwritten. The RDR content is not lost until the ORE flag is cleared. After that, any data received during overrun is lost.
- The previous data is available by reading the LPUARTx_RDR register before the RXNE flag is cleared.
- An interrupt is generated if either the RXNEIE or the EIE bit in the LPUARTx_CR1 register is set.

The ORE bit, when set, indicates that at least 1 data has been lost.

Framing error

A framing error is detected when the stop bit is not recognized on reception at the expected time. When the framing error is detected:

- The FE bit is set by hardware.
- The invalid data is transferred from the shift register to the LPUARTx_RDR register.
- When the FE bit is set, an interrupt is generated if the EIE bit in the LPUARTx_CR3 register is also set.

The FE bit is reset by setting FECF bit in the LPUARTx_ICR register.

Configurable stop bits during reception

The number of stop bits to be received can be configured through the STOPBIT bit in the LPUARTx_CR2 register.

There are several options:

- 1 stop bit: sampling for 1 stop bit is done on the middle position of the stop bit.
- 2 stop bits: sampling for 2 stop bits is done on the middle position of the second stop bit.

If a stop bit is not detected during sampling, a frame error is generated, and the FE flag and the RXNE flag are set to 1.

31.3.7 LPUART baud rate generation

The baud rate for the receiver and transmitter (Rx and Tx) are both set to the value programmed in the LPUARTx_BRR register:

$$\text{Tx/Rx baud rate} = \frac{256 * LPUARTx\_KCLK\_PRES}{BRR}$$

The LPUARTx_KCLK_PRES is obtained by prescaling the LPUARTx_KCLK clock with the value defined in the LPUARTx_PRESC register.

The LPUARTx_KCLK clock source is provided by the RCC and can be configured as PCLK, LXTAL, RCH, or SYSCLK. Refer to the *Peripheral independent clock configuration register (RCC_CLKSEL)*.

The BRR is stored in the LPUARTx_BRR register, and its value range is:

$$0x100000 > BRR \geq 0x300$$

Table 31-3 Error calculation for programmed baud rates at LPUARTx_KCLK_PRES = 32.768 KHz

Baud rate desired	Baud rate actual	BRR value	Error(%)
300 bps	300 bps	0x6D3A	0
600 bps	600 bps	0x369D	0
1200 bps	1200.0872 bps	0x1B4E	0.007
2400 bps	2400.17 bps	0xDA7	0.007
4800 bps	4801.7 bps	0x6D3	0.035
9600 bps	9608.94 bps	0x369	0.093

31.3.8 Tolerance of the LPUART receiver to clock deviation

The LPUART operates correctly only if the total clock system deviation is less than the tolerance of the LPUART receiver. The causes which contribute to the total deviation are:

- DTR_ERR: deviation due to the transmitter error, which also includes the deviation of the transmitter's local oscillator.
- DQU_ERR: error due to the baud rate quantization of the receiver.
- DRX_OSERR: deviation of the receiver local oscillator.
- DL_ERR: deviation due to the transmission line (generally due to the transceivers which can introduce an asymmetry between the low-to-high transition timing and the high-to-low transition timing).

The receive tolerance must meet:

DTR_ERR + DQU_ERR + DRX_OSERR + DL_ERR + DWKUP < LPUART receiver tolerance.

DWKUP is the error due to sampling point deviation when the wakeup from low-power mode is used.

The LPUART receive tolerance depends on the following settings:

- 8- or 9-bit character length defined by the WL bits in the LPUARTx_CR1 register, the receive tolerance is greater when using 8-bit character length.
- Number of stop bits defined through STOPBIT in the LPUARTx_CR2 register, the receive tolerance is greater when using 1-bit stop bit.
- The value of the LPUARTx_BRR register. The receive tolerance is greater when the value is larger.

31.3.9 LPUART single-wire half-duplex communication

Single-wire half-duplex mode is supported by LPUART and is selected by setting the HDEN bit in the LPUARTx_CR3 register. The TX and RX lines are internally connected.

Writing 1 to the HDEN bit enables the single-wire half-duplex mode, which results in the following actions:

- The TX and RX lines are internally connected
- By default, the TX pin is used for data transmission and reception, while the RX pin is not used and can be used as a GPIO.
- When the SWAP bit in LPUARTx_CR2 is set to 1, the TX/RX pin swap is enabled, and the RX pin is used for data transmission and reception.
- The TX pin is always released when no data is transmitted. The I/O must be configured so that TX is configured as alternate function open-drain with an internal or external pull-up.

31.3.10 LPUART communication using DMA

The LPUART is capable of performing continuous communications using the DMA. The DMA requests for Rx and Tx are generated independently.

Transmission using DMA

DMA mode can be enabled for transmission by setting DMAT bit in the LPUARTx_CR3 register. Data are loaded from a storage area configured using the DMA peripheral to the LPUARTx_TDR register whenever the TXE flag is set. To map a DMA channel for LPUART transmission, use the following procedure (x denotes the channel number):

1) Configure the relevant information for the channel in the DMA register DMA_CCx. See the *DMA channel configuration*.
2) Configure the total number of bytes to be sent in the DMA register DMA_CNDTRx.
3) Configure the memory area address as the source address in the DMA register DMA_CSARx. Write the address of the LPUARTx_TDR register to the DMA register DMA_CDARx. Each time the TXE bit is set to 1, data is loaded from the memory area to the LPUARTx_TDR register.
4) Write 1 to the TCCF bit in the LPUARTx_ICR register to clear the TC flag in

the LPUARTx_ISR register.

5) Set the EN bit in the DMA_CCx register to 1 to enable the channel.

An interrupt is generated on the corresponding DMA channel when the number of data transfers set in the DMA controller is reached.

In transmit mode, when the DMA has completed writing all the data to be sent, the TFx flag in the DMA_ISR register is set to 1. Additionally, it is able to confirm whether the LPUART communication is completed by checking the TC flag in the LPUARTx_ISR register. The TC flag remains 0 during data transmission and is automatically set to 1 after the last character is transmitted.

Reception using DMA

DMA mode can be enabled for reception by setting the DMAR bit in the LPUARTx_CR3 register.

Data are loaded from the LPUARTx_RDR register to a storage area configured using the DMA peripheral whenever a data byte is received. To map a DMA channel for LPUART reception, use the following procedure (x denotes the channel number):

1) Configure the relevant information for the channel in the DMA register DMA_CCx. See the *DMA channel configuration*.

2) Configure the total number of bytes to be received in the DMA register DMA_CNDTRx.

3) Configure the address of the LPUARTx_RDR register in the DMA register DMA_CSARx. Write the memory area address as the source address to the DMA register DMA_CDARx. Each time the RXNE bit is set to 1, data is loaded from the LPUARTx_RDR register to the memory area.

4) Set the EN bit in the DMA_CCx register to 1 to enable the channel.

DMA action on reception error

In the LPUARTx_CR3 register, the DDRE bit can be configured to disable DMA in case of a receive error, including frame errors, parity errors.

By default, DDRE is 0, so DMA is not disabled in case of a receive error. The corresponding error flags are set to 1, but RXNE remains 0, and no DMA request is generated. The data with error will not be received into the LPUARTx_RDR register, but the next correct data can be received.

When DDRE is 1, the corresponding error flags and RXNE are both set to 1 in case of

a receive error, and the data with error will be received into the LPUARTx_RDR register. However, DMA requests are masked. In this case, when a receive error occurs, the software must disable the DMA request (DMAR = 0) and clear RXNE to 0, and then clear the error flags before DMA reception can continue.

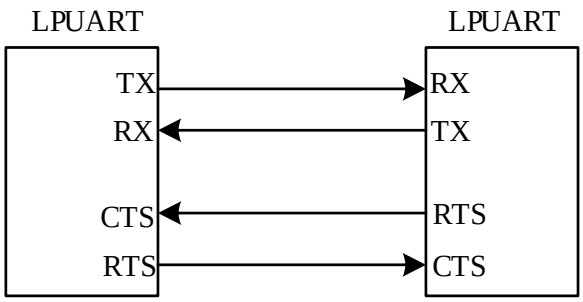
When an overrun error occurs, the ORE flag is set to 1. During this time, received data will only refresh the shift register and will not overwrite the last data in the RDR. Therefore, regardless of whether the DDRE bit is set to 1, RXNE must be cleared to 0, and the error flag must be cleared before data reception can continue.

31.3.11 RS232 hardware flow control

RS232 hardware flow control RTS/CTS is used for communication traffic control. RTS is the output pin, and CTS is the input pin. The connection relationship between the RTS and CTS pins is shown in the following diagram.

RS232 RTS and CTS flow control can be enabled independently by writing the RTSE and CTSE bits respectively to 1 in the LPUARTx_CR3 register. Set to enable, clear to disable.

Figure 31-5 Hardware flow control between 2 LPUARTs

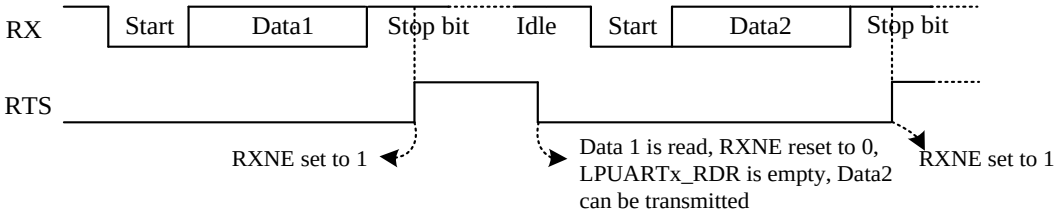


RS232 RTS flow control

If the RTS flow control is enabled (RTSE = 1), then RTS is asserted (tied low) as long as the LPUART receiver is ready to receive a new data. When the receive register is full, RTS is deasserted, indicating that the transmission is expected to stop at the end of the current frame.

An example of communication with RTS flow control enabled is as below:

Figure 31-6 RTS flow control



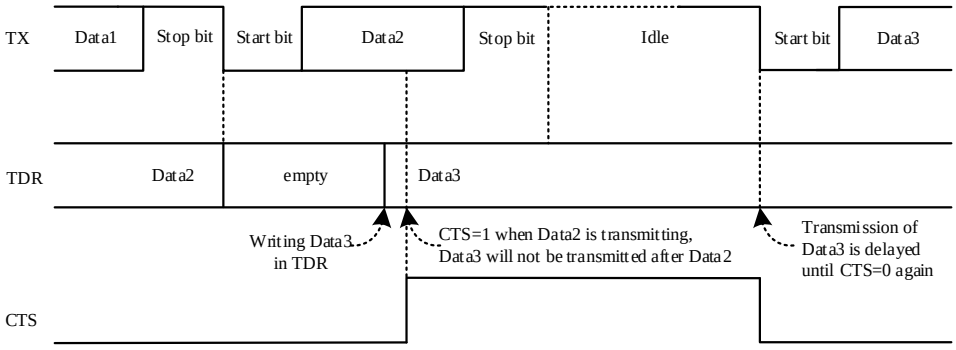
RS232 CTS flow control

If the CTS flow control is enabled (CTSE = 1), then the transmitter checks the CTS input before transmitting the next frame. The CTSIF status bit is automatically set by hardware as soon as the CTS input toggles. An interrupt is generated if the CTSIE bit in the LPUARTx_CR3 register is set.

- If CTS is asserted (tied low), then the next data can be transmitted.
- When CTS is deasserted during a transmission, the current transmission is completed before the transmitter stops. Data being written to the transmit data register is not sent out. When the CTS is asserted again, transmission resumes.

An example of communication with CTS flow control enabled is as below:

Figure 31-7 CTS flow control



Note: For correct behavior, CTS must be deasserted at least 5 LPUARTx_KCLK clock source periods before the end of the current character.

31.4 LPUART low-power management

31.4.1 LPUART supports for low-power mode

LPUART supports operation in low-power modes. When the UEWK bit in the LPUARTx_CR1 register is set to 1, it allows the system to be woken up from Stop mode.

By selecting a clock source that operates in Stop mode (RCH or LXTAL) for the LPUART, data can be transmitted normally even when the PCLK clock is disabled, and the system can be woken up from Stop mode.

The LPUART_WAKEUP signal is a wakeup signal issued by the LPUART, which supports waking up the system from Stop mode. The trigger event sources for the LPUART_WAKEUP signal can be configured by enabling the relevant interrupts. The following wakeup events can be used as trigger events for the LPUART_WAKEUP signal:

- Wake up when a frame of data is received, enabled by setting RXNEIE = 1.
- Wake up when a character matching the CMFD[7:0] bit field in the LPUARTx_CR2 register is received, enabled by setting CMIE = 1.
- The LPWK bit (Low Power Wakeup flag) being set to 1 can also serve as a trigger event for the LPUART_WAKEUP signal, enabled by setting LPWKIE = 1. The conditions for setting LPWK to 1 are configured in the LPWKS bit and have the following two options:
  - Detecting a start bit
  - Receiving a frame of data

Note: After initialization and enabling reception (RE = 1), ensure that the REACK bit is set to 1 before entering low-power mode to confirm that LPUART reception is enabled.

31.4.2 Wakeup from low-power mode when LPUART_KCLK is OFF in low-power mode

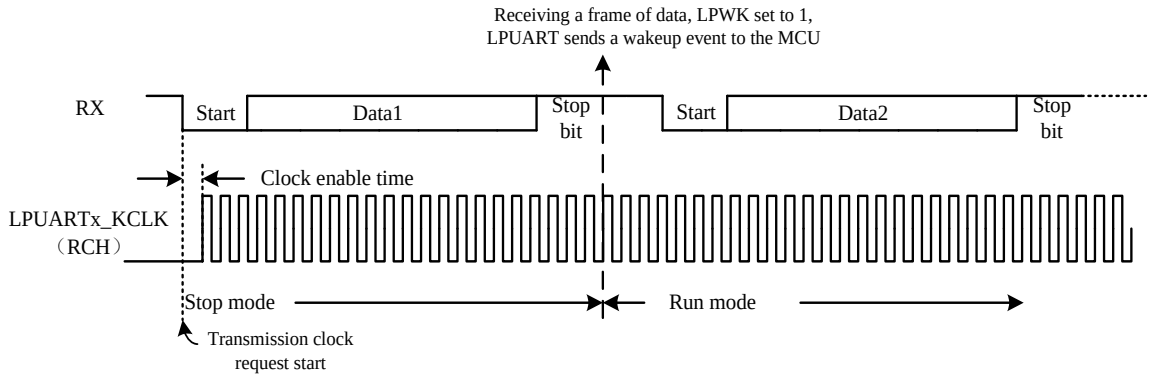
If the LPUARTx_KCLK clock stops during Stop mode, the LPUARTx_KCLK clock is re-enabled when a falling edge is detected on the LPUART receive line.

After the LPUARTx_KCLK clock is re-enabled, if a valid wakeup event is detected, the system wakes up from Stop mode and resumes normal data reception.

If the wakeup event is invalid, the LPUARTx_KCLK clock is disabled again, and the system remains in Stop mode.

The following figure show an example where the wakeup event is the reception of a frame of data:

Figure 31-8 Wakeup event verified



Note: When the wakeup event is start bit detection, the LPUART sends a wakeup event upon successful detection of the start bit.

The status of the LPUART peripheral in different low-power modes is shown in the following table:

Table 31-4 Low-power mode description

Mode	Description
Sleep	The LPUART communication is unaffected, and the wake-up request signal or interrupt request signal output by the LPUART can cause the system to exit Sleep mode.
Stop	When the clock source is configured as RCH or LXTAL, the wake-up request signal or interrupt request signal output by the LPUART can wake up the system in this mode. The UEWK bit needs to be set to 1.

31.5 LPUART interrupts

LPUARTx_IRQ is the LPUART interrupt request signal and LPUARTx_WAKEUP is the LPUART wakeup request signal. The sources of these signals are detailed in the following table:

Table 31-5 LPUART interrupt requests(1)

Interrupt event	Event flag	Enable Control bit	Interrupt clear method	Out signal	
				LPUART_ IRQ	LPUART_ WAKEUP
Transmit data register empty	TXE	TXEIE	TXE cleared when a data is written in TDR	√	×

Interrupt event	Event flag	Enable Control bit	Interrupt clear method	Out signal	
				LPUART_ IRQ	LPUART_ WAKEUP
Receive data register not empty	RXNE	RXNEIE	RXNE cleared by reading RDR or by setting RXFLUQ bit	√	√
Overrun error detected	ORE	RXNEIE	ORE cleared by setting ORECF bit	√	×
Transmission Complete	TC	TCIE	TC cleared when a data is written in TDR or by setting TCCF bit	√	×
CTS interrupt	CTSIF	CTSIE	CTSIF cleared by software by setting CTSCF bit	√	×
Parity error	PE	PEIE	PE cleared by setting PECF bit	√	×
Framing Error	FE	EIE	FE flag cleared by setting FECF bit	√	×
Overrun error	ORE		ORE cleared by setting ORECF bit	√	×
Character match	CMF	CMIE	CMF cleared by setting CMCF bit	√	√
Wakeup from low-power mode	LPWK	LPWKIE	LPWK cleared by setting LPWKCF bit	√	√

1. "√" indicates support for the feature, while "×" indicates no support for the feature.

31.6 LPUART registers

The LPUART registers can only be accessed by words (32-bit).

Table 31-6 LPUARTx base address (x = 1, 2)

Peripheral	Base address
LPUART1	0x4000 8000
LPUART2	0x4000 8400

31.6.1 LPUART control register 1 (LPUARTx_CR1)

Address offset: 0x00

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	CMIE	Res.	WL	Res.	PEN	PTS	PEIE	TXEIE	TCIE	RXNEIE	Res.	TE	RE	UEWK	UE
	rw		rw		rw	rw	rw	rw	rw	rw		rw	rw	rw	rw

Bits	Name	Description
31:15	Reserved	Must be kept at reset value
14	CMIE	Character match interrupt enable 0: Disable 1: Enable
13	Reserved	Must be kept at reset value
12	WL	Word length This bit is used to determine the word length. 0: 8 bit 1: 9 bit *Note: This bit can only be written when the LPUART is disabled (UE = 0).*
11	Reserved	Must be kept at reset value

10	PEN	Parity control enable 0: Disable 1: Enable *Note: This bit can only be written when the LPUART is disabled (UE = 0).*
9	PTS	Parity selection 0: Even parity 1: Odd parity *Note: This bit can only be written when the LPUART is disabled (UE = 0).*
8	PEIE	PE interrupt enable 0: Disable 1: Enable
7	TXEIE	TDR empty interrupt enable 0: Disable 1: Enable
6	TCIE	Transmission complete interrupt enable 0: Disable 1: Enable
5	RXNEIE	RDR not empty interrupt enable 0: Disable 1: Enable
4	Reserved	Must be kept at reset value
3	TE	Transmitter enable 0: Disable 1: Enable *Note: After clearing TE, it is necessary to wait for the TEACK flag to clear before proceeding with subsequent operations.*
2	RE	Receiver enable 0: Disable

1: Enable

Note: After clearing RE, it is necessary to wait for the REACK flag to clear before proceeding with subsequent operations.

1	UEWK	LPUART wake up function enable in low-power mode 0: Disable 1: Enable
0	UE	LPUART enable When this bit is cleared, the LPUART prescaler immediately stops, and the current operation is discarded. The configuration of the LPUART is kept, but the TDR and RDR registers are cleared, all the status flags, in the LPUARTx_ISR register are reset. 0: Disable 1: Enable

31.6.2 LPUART control register 2 (LPUARTx_CR2)

Address offset: 0x04

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMFD[7:0]								Res.				MSB FIRST	DATA IVC	TX IVC	RX IVC
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 BIT	Res.												
rw		rw													

Bits	Name	Description
31:24	CMFD[7:0]	Address of the LPUART node This bit field is used for character match detection. The whole received character (8-bit) is compared to the ADD[7:0] value and CMF flag is set on match. *Note: This bit can only be written when the receiver is disabled (RE = 0) or the LPUART is disabled (UE = 0).*
23:20	Reserved	Must be kept at reset value

19	MSBFIRST	Most significant bit first 0: Data is transmitted/received with data bit 0 first, following the start bit 1: Data is transmitted/received with the MSB (bit 7/8) first, following the start bit *Note: This bit can only be written when the LPUART is disabled (UE = 0).*
18	DATAIVC	Binary data inversion 0: Logical data from the data register are send/received in positive/direct logic (1 = H, 0 = L) 1: Logical data from the data register (including the parity bit) are send/received in negative/inverse logic (1 = L, 0 = H) *Note: This bit can only be written when the LPUART is disabled (UE = 0).*
17	TXIVC	TX pin active level inversion 0: TX pin signal works using the standard logic levels (V_{DD} = 1, GND = 0) 1: TX pin signal values are inverted (V_{DD} = 0, GND = 1) *Note: This bit can only be written when the LPUART is disabled (UE = 0).*
16	RXIVC	RX pin active level inversion 0: RX pin signal works using the standard logic levels (V_{DD} = 1, GND = 0) 1: RX pin signal values are inverted (V_{DD} = 0, GND = 1) *Note: This bit can only be written when the LPUART is disabled (UE = 0).*
15	SWAP	Swap TX/RX pins 0: TX/RX pins are used as defined in standard pinout 1: The TX and RX pins functions are swapped. This enables to work in the case of a cross-wired connection to another UART. *Note: This bit can only be written when the LPUART is disabled (UE = 0).*

14	Reserved	Must be kept at reset value
13	STOPBIT	STOP bit This bit is used for programming the stop bit 0: 1 stop bit 1: 2 stop bits *Note: This bit can only be written when the LPUART is disabled (UE = 0).*
12:0	Reserved	Must be kept at reset value

31.6.3 LPUART control register 3 (LPUARTx_CR3)

Address offset: 0x08

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		DDRE	Res.		CTSIE	CTSE	RTSE	DMAT	DMAR	Res.		HDEN	Res.		EIE
		rw			rw	rw	rw	rw	rw			rw			rw

Bits	Name	Description
31:23	Reserved	Must be kept at reset value
22	LPWKIE	Wakeup from low-power mode interrupt enable 0: Disable 1: Enable
21	Reserved	Must be kept at reset value
20	LPWKS	Wakeup from low-power mode interrupt flag selection This bit specifies the event which activates the LPWK (wakeup from low-power mode flag) 0: LPWK active on start bit detection 1: LPWK active on receiving a frame *Note: This bit can only be written when the LPUART is disabled*

(UE = 0).

19:14	Reserved	Must be kept at reset value
13	DDRE	DMA disable on reception error 0: DMA is not disabled in case of reception error. The corresponding error flag is set but RXNE is kept 0, the DMA request is not asserted. 1: DMA is disabled following a reception error. The corresponding error flag is set, as well as RXNE. The DMA request is masked until the error flag is cleared. *Note: The reception errors are: parity error and framing error. This bit can only be written when the LPUART is disabled (UE = 0).*
12:11	Reserved	Must be kept at reset value
10	CTSIE	CTS interrupt enable 0: Disable 1: Enable
9	CTSE	CTS enable 0: Disable 1: Enable *Note: This bit can only be written when the LPUART is disabled (UE = 0).*
8	RTSE	RTS enable 0: Disable 1: Enable *Note: This bit can only be written when the LPUART is disabled (UE = 0).*
7	DMAT	DMA enable transmitter 0: Disable 1: Enable
6	DMAR	DMA enable receiver

		0: Disable 1: Enable
5:4	Reserved	Must be kept at reset value
3	HDEN	Half-duplex enable 0: Disable 1: Enable *Note: This bit can only be written when the LPUART is disabled (UE = 0).*
2:1	Reserved	Must be kept at reset value
0	EIE	Error interrupt enable This bit is required to enable interrupt generation in case of a framing error, overrun error (FE = 1 or ORE = 1 in the LPUARTx_ISR register). 0: Disable 1: Enable

31.6.4 LPUART baud rate register (LPUARTx_BRR)

Address offset: 0x0C

Reset value: 0x0000 0000

Note: This register can only be written when the LPUART is disabled (UE = 0).

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

Bits	Name	Description
31:20	Reserved	Must be kept at reset value
19:0	BRR[19:0]	LPUART baud rate BRR value range is:

0x100000 > BRR ≥ 0x300

See *LPUART baud rate generation*.

31.6.5 LPUART request register (LPUARTx_RQR)

Address offset: 0x18

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.											TX FLUQ	RX FLUQ	Res.		
											w	w			

Bits	Name	Description
31:5	Reserved	Must be kept at reset value
4	TXFLUQ	Transmit data flush request This bit is used to flush the TDR and sets the flag TXE. *Note: After writing 1 to this bit, it is necessary to wait at least one LPUARTx_KCLK_PRES cycle before writing to the TDR.*
3	RXFLUQ	Receive data flush request Writing 1 to this bit clears the RDR and the RXNE flag. Unread data will be discarded, which can prevent an overrun error.
2:0	Reserved	Must be kept at reset value

31.6.6 LPUART interrupt and status register (LPUARTx_ISR)

Address offset: 0x1C

Reset value: 0x0000 00C0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.									REACK.	TEACK	LPWK	Res.		CMF	BUSY
									r	r	r			r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					CTS	CTSIF	Res	TXE	TC	RXNE	Res.	ORE	Res.	FE	PE
					r	r		r	r	r		r		r	r

Bits	Name	Description
31:23	Reserved	Must be kept at reset value
22	REACK	Receive enable acknowledge flag 0: The receiver is not ready 1: The receiver is ready
21	TEACK	Transmit enable acknowledge flag When TE is cleared, this bit is cleared after the transmitter has completed disabling. When the transmitter is ready to send, this bit is set to 1. 0: The transmitter is not ready 1: The transmitter is ready
20	LPWK	Wakeup from low-power mode flag This bit is set by hardware, when a wakeup event selected by LPWKS is detected. It is cleared by software, writing 1 to the LPWKCF in the LPUARTx_ICR register. When UEWK is cleared, LPWK flag is also cleared.
19:18	Reserved	Must be kept at reset value
17	CMF	Character match flag This bit is set by hardware, when the character defined by CMFD[7:0] is received. It is cleared by software, writing 1 to the CMCF in the LPUARTx_ICR register. 0: No character match detected 1: Character match detected
16	BUSY	Busy flag It is active when a communication is ongoing on the RX line (successful start bit detected). It is reset at the end of the reception (successful or not). 0: LPUART is idle (no reception)

		1: Reception on going
15:11	Reserved	Must be kept at reset value
10	CTS	CTS flag It is an inverted copy of the status of the CTS input pin. 0: CTS line is high 1: CTS line is low
9	CTSIF	CTS interrupt flag This bit is set by hardware when the CTS input toggles, if the CTSE bit is set. It is cleared by software, by writing 1 to the CTSCF bit in the LPUARTx_ICR register. 0: No change occurred on the CTS status line 1: A change occurred on the CTS status line
8	Reserved	Must be kept at reset value
7	TXE	TDR empty When the data in the LPUARTx_TDR register has been transmitted, this bit is set to 1. Writing 1 to the TXFLUQ bit in the LPUARTx_RQR register also sets this bit to 1. Writing to the LPUARTx_TDR register clears this bit. 0: LPUARTx_TDR is not empty 1: LPUARTx_TDR is empty
6	TC	Transmission complete This bit is set by hardware if the transmission of a frame containing data is complete and if TXE is set. It is cleared by software, writing 1 to the TCCF in the LPUARTx_ICR register or by writing to the LPUART_TDR register. 0: Transmission is not complete 1: Transmission is complete
5	RXNE	RDR not empty When data is received, the RXNE bit is set to 1.

		Reading the LPUARTx_RDR register clears this bit. Writing 1 to the RXFLUQ bit in the LPUARTx_RQR register also clears the RXNE flag. 0: Data is not received 1: Received data is ready to be read
4	Reserved	Must be kept at reset value
3	ORE	Overrun error This bit is set by hardware when the data currently being received in the shift register is ready to be transferred into the LPUARTx_RDR register while RXNE = 1 or the previous DMA read request has not been processed. It is cleared by software, writing 1 to the ORECF, in the LPUARTx_ICR register. 0: No overrun error 1: Overrun error is detected *Note: When this bit is set, the LPUARTx_RDR register content is not lost but the shift register is overwritten.*
2	Reserved	Must be kept at reset value
1	FE	Framing error This bit is set by hardware when a de-synchronization is detected. It is cleared by software, writing 1 to the FECF bit in the LPUARTx_ICR register. 0: No framing error is detected 1: Framing error is detected
0	PE	Parity error This bit is set by hardware when a parity error occurs in receiver mode. It is cleared by software, writing 1 to the PECF in the LPUARTx_ICR register. 0: No parity error 1: Parity error

31.6.7 LPUART interrupt flag clear register (LPUARTx_ICR)

Address offset: 0x20

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.						CTSCF	Res.		TCCF	Res.		ORECF	Res.	FECF	PECF
						w			w			w		w	w

Bits	Name	Description
31:21	Reserved	Must be kept at reset value
20	LPWKCF	Wakeup from low-power mode clear flag Writing 1 to this bit clears the LPWK flag in the LPUARTx_ISR register.
19:18	Reserved	Must be kept at reset value
17	CMCF	Character match clear flag Writing 1 to this bit clears the CMF flag in the LPUARTx_ISR register.
16:10	Reserved	Must be kept at reset value
9	CTSCF	CTS clear flag Writing 1 to this bit clears the CTSIF flag in the LPUARTx_ISR register.
8:7	Reserved	Must be kept at reset value
6	TCCF	Transmission complete clear flag Writing 1 to this bit clears the TC flag in the LPUARTx_ISR register.
5:4	Reserved	Must be kept at reset value

3	ORECF	Overrun error clear flag Writing 1 to this bit clears the ORE flag in the LPUARTx_ISR register.
2	Reserved	Must be kept at reset value
1	FECF	Framing error clear flag Writing 1 to this bit clears the FE flag in the LPUARTx_ISR register.
0	PECF	Parity error clear flag Writing 1 to this bit clears the PE flag in the LPUARTx_ISR register.

31.6.8 LPUART receive data register (LPUARTx_RDR)

Address offset: 0x24

Reset value: 0x0000 0000

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

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

Bits	Name	Description
31:9	Reserved	Must be kept at reset value
8:0	RDR[8:0]	Receive data value When receiving with the parity enabled, the value read in the MSB bit is the received parity bit.

31.6.9 LPUART transmit data register (LPUARTx_TDR)

Address offset: 0x28

Reset value: 0x0000 0000

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

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

Bits	Name	Description
31:9	Reserved	Must be kept at reset value
8:0	TDR[8:0]	Transmit data value When transmitting with the parity enabled, the value written in the MSB has no effect because it is replaced by the parity.

31.6.10 LPUART prescaler register (LPUARTx_PRESC)

Address offset: 0x2C

Reset value: 0x0000 0000

Note: This register can only be written when the LPUART is disabled (UE = 0).

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

Bits	Name	Description
31:4	Reserved	Must be kept at reset value
3:0	PRESCALER[3:0]	Clock prescaler The LPUARTx_KCLK clock is divided by the prescaler factor in this register to obtain LPUARTx_KCLK_PRES. 0000: Input clock not divided 0001: Input clock divided by 2 0010: Input clock divided by 4 0011: Input clock divided by 6 0100: Input clock divided by 8 0101: Input clock divided by 10

0110: Input clock divided by 12

0111: Input clock divided by 16

1000: Input clock divided by 32

1001: Input clock divided by 64

1010: Input clock divided by 128

1011: Input clock divided by 256

Remaining combinations: Reserved (input clock divided by 256)

32 Serial peripheral interface (SPI)

32.1 Introduction

The serial peripheral interface (SPI) protocol supports synchronous full-duplex serial communication with external devices.

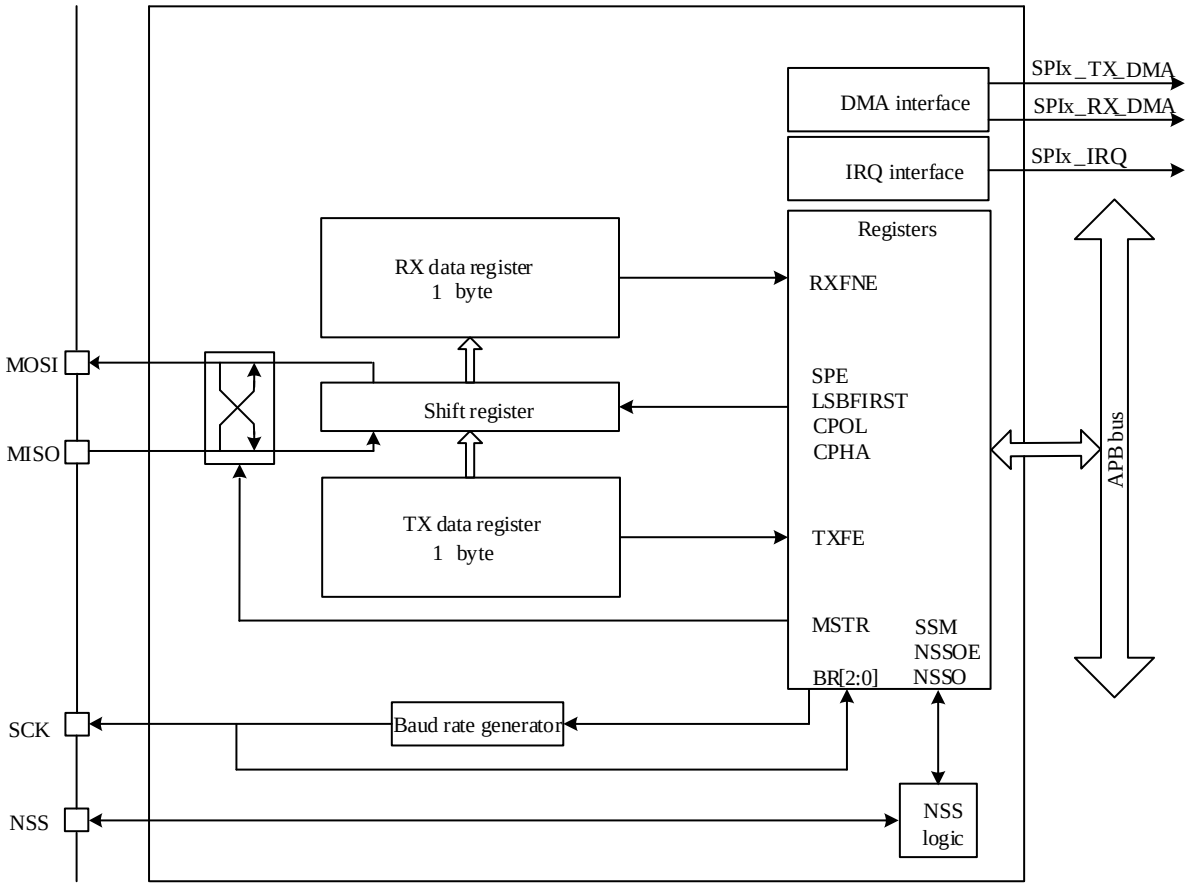
32.2 SPI main features

- Supports synchronous full-duplex communication with the standard four lines: NSS, SCK, MISO, and MOSI
- Master or slave mode
- Multimaster mode capability
- Slave mode with software NSS
- 7 master mode baud rate prescalers up to PCLK/2
- Programmable clock polarity and phase
- Fixed 8-bit data transmission size
- Programmable data order with MSB-first or LSB-first shifting
- SPI Motorola support
- Overrun and master mode fault with interrupt capability
- DMA transfer support

32.3 SPI functional description

32.3.1 SPI block diagram

Figure 32-1 SPIx block diagram (x = 1, 2)



32.3.2 SPI pins and internal signals

Table 32-1 SPI input/output pins

Pin name	Signal type	Description
MISO	Bidirectional	Master input/slave output data
MOSI	Bidirectional	Master output/slave input data
SCK	Bidirectional	Serial clock output pin for SPI masters and input pin for SPI slaves
NSS	Bidirectional	Slave select pin

Table 32-2 SPI internal signals

Internal signal name	Signal type	Description
SPIx_IRQ	Output	SPI interrupts
SPIx_TX_DMA	Output	DMA request of SPI transmitting data

Internal signal name	Signal type	Description
SPIx_RX_DMA	Output	DMA request of SPI receiving data

32.3.3 Single master communication

The SPI supports full-duplex synchronous serial communication with extern devices. The master transmits the data to be sent to the slave via the MOSI line and receives data from the slave via the MISO line. The pin connection diagrams of single master communicate with single slave or multi-slave are shown below:

Figure 32-2 Single-master / single-slave application

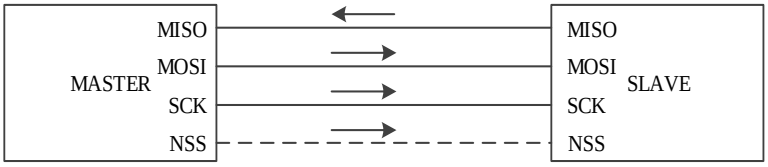
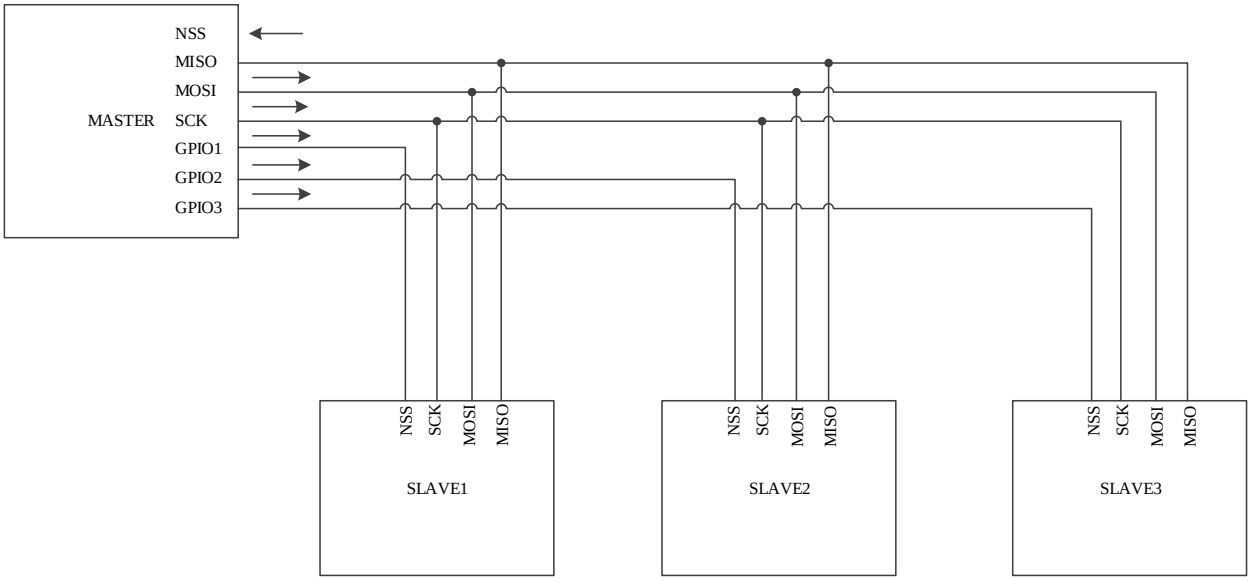


Figure 32-3 Single-master / multi-slave application



The SPI operates in master mode by setting MSTR bit in the SPIx_CR1 register.

The SPI clock signal SCK is provided by master, and the bit rate is controlled by the BR[2:0] field in the SPIx_CR1 register, with a configurable range from $f_{PCLK}/2$ to $f_{PCLK}/128$.

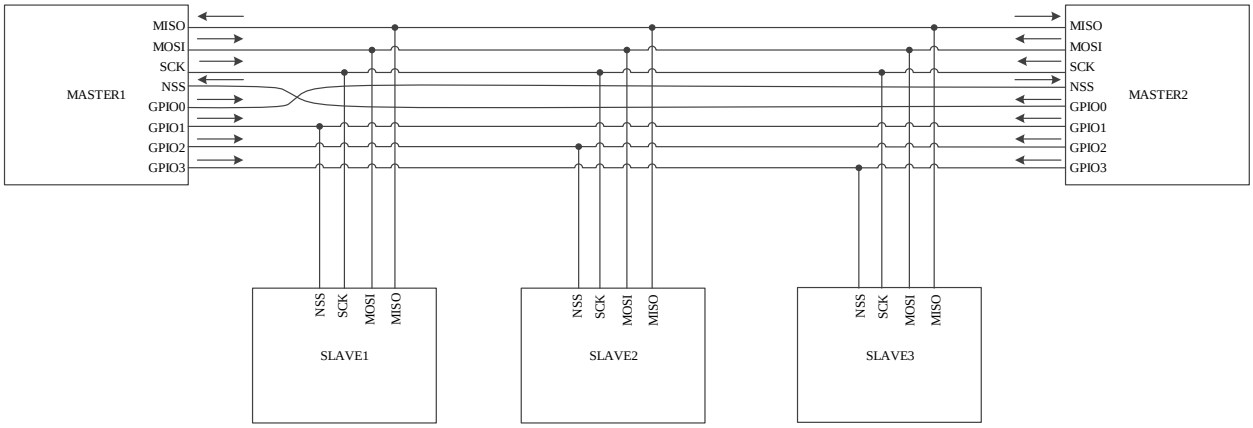
The control method of NSS refers to: *Slave select (NSS) pin management*.

32.3.4 Multi-master communication

When SPI bus contains multiple masters, the user can use master mode fault feature

which detects a potential conflict between two nodes trying to master the bus at the same time. The pin connection diagram of multi-master application is shown below:

Figure 32-4 Multi-master / multi-slave application



When there is no data transmission from the master devices on the bus, the master devices need to keep in a disabled state (SPE = 0), and GPIO1~3 are configured in input mode. When the master and slave need to communicate, follow the steps below:

1) When MASTER1 controls the bus, it switches itself into master mode. If the NSS inputs is low-level, it means the bus has been occupied by MASTER2, and MASTER1 will generate master mode fault.
2) MASTER1 drivers low-level to the NSS pin of MASTER2 through GPIO0, indicating the bus is occupied.
3) MASTER1 outputs low-level through GPIO1 to select SLAVE1 for communication.
4) After the communication is completed, MASTER1 switches from the master mode to the disabled state.
5) MASTER1 drivers high-level to the NSS pin of MASTER2 through GPIO0, indicating the bus is idle.
6) The GPIO1 of MASTER1 is configured in input mode.

If potentially both nodes raised their mastering request at the same time, a bus conflict event appears (refer to: *Master mode fault*). Then the user can apply some simple arbitration process. e.g. to postpone next attempt by predefined different time-outs applied at both nodes.

Except for the difference in NSS configuration, the multi-master configuration is the same as the single-master configuration. Please refer to *Slave select (NSS) pin management*.

32.3.5 Slave communication

The SPI interface operates in slave mode, when the MSTR bit in SPIx_CR1 register is cleared. In slave mode, the sequence of SPI data is controlled by master clock. The slave should write the transmitted data to the SPIx_DR register before the first clock edge arrives to ensure the correctness of the transmitted data. The slave pin connection is referred to *Figure: Single-master / single-slave application*.

32.3.6 Slave select (NSS) pin management

In slave mode, the NSS can be configured as either hardware chip select or software chip select:

- Hardware slave selection (SSM = 0): The slave can communicate to the SPI master when the NSS is pulled down to low level.
- Software slave selection (SSM = 1): The input of NSS pin is ignored, SPI is always selected.

In master mode, the NSS pin can be used as an input or output:

- When used as output, it is applied in Single-master application and drive the chip select signal of a single SPI slave through NSS pin, or drive the chip select signal of multiple slaves through several GPIO.
  - If GPIO is used as the chip select pin, NSSOE should remain 1. According to the SPI communication protocol, operate the GPIO for chip selection, then clear the NSSO bit before sending data.
  - If NSS is used as the chip select pin, NSSOE should remain 1. According to the SPI communication protocol, operate the NSSO bit for chip selection.
- When used as input, it is applied in Multi-master application to detect that if the SPI bus is occupied or not.
  - If the NSS pin is held high and NSSOE is kept cleared in master mode. According to the SPI communication protocol, operate the GPIO for chip selection, then clear the NSSO bit before sending data.
  - If the NSS pin is held low and NSSOE is kept cleared, the SPI will enter the master mode fault error.

The configuration application of NSS is shown in the following table:

Table 32-3 NSS configuration

Mode	NSS type	NSSOE	NSSO	SSM	Description
Slave mode	Input	0	1	0	The NSS of slave is used as input to determine whether it has been selected by the master.
				1	The slave is always in a selected state, the input of NSS pin will always be ignored.
Master mode	Input	0	0/1	0	When the NSS pin is held high, after performing chip selection with GPIO, the NSSO bit should also be operated.
	Output	1	0/1	0	NSS is the output signal, and the output level of NSS is controlled by NSSO bit.

32.3.7 Communication formats

During SPI communication, receive and transmit operations are performed simultaneously. The serial clock (SCK) synchronizes the shifting and sampling of the information on the data lines. The communication format depends on the clock phase and the clock polarity. Refer to the following table.

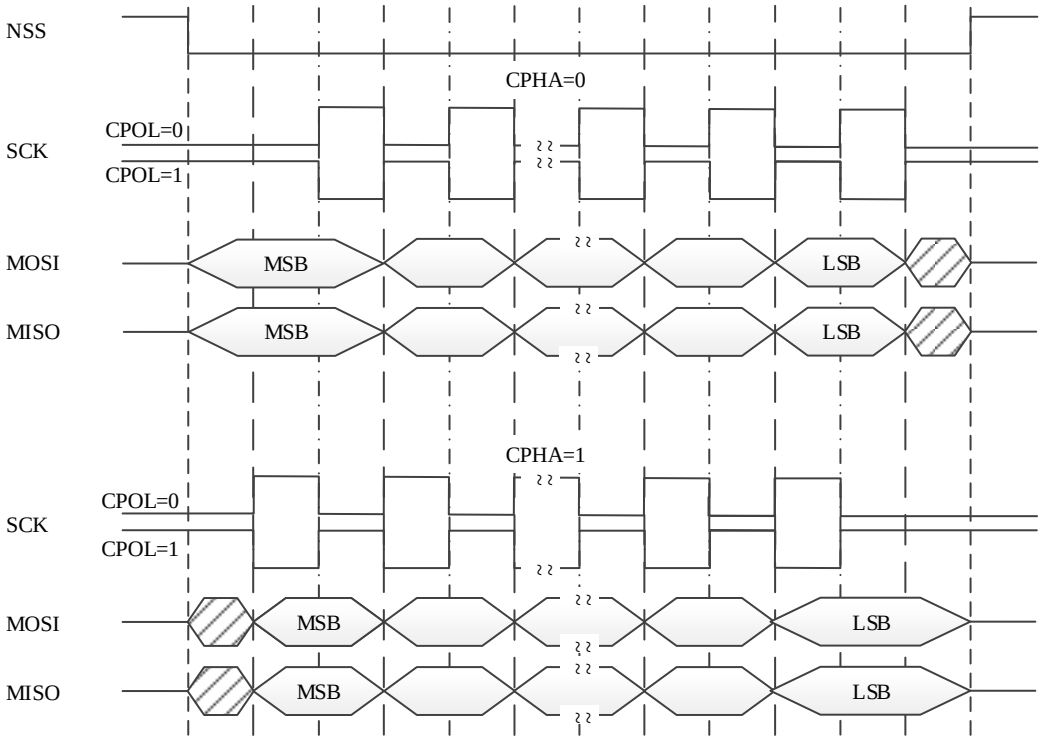
Table 32-4 Communication modes configuration

SPI mode	CPOL	CPHA	Description
Mode 0	0	0	The SCK pin has a low-level idle state. The first edge on the SCK pin captures the first data bit transacted. Data is sampled on the rising edge of clock and updated on the falling edge of clock.
Mode 1	0	1	The SCK pin has a low-level idle state. The second edge on the SCK pin captures the first data bit transacted. Data is sample on the falling edge of clock and updated on the rising edge of clock.

SPI mode	CPOL	CPHA	Description
Mode 2	1	0	The SCK pin has a high-level idle state. The first edge on the SCK pin captures the first data bit transacted. Data is sample on the falling edge of clock and updated on the rising edge of clock.
Mode 3	1	1	The SCK pin has a high-level idle state. The second edge on the SCK pin captures the first data bit transacted. Data is sampled on the rising edge of clock and updated on the falling edge of clock.

The combination of the CPOL bit and the CPHA bit in the SPIx_CR1 register selects one of four communication modes.

Figure 32-5 Data clock timing diagram



The SPI shift register can be set up to shift out MSB-first or LSB-first, depending on the LSBFIRST bit in the SPIx_CR1 register: LSBFIRST = 0 selects MSB-first transmission, and LSBFIRST = 1 selects LSB-first transmission.

32.3.8 SPI status flags

The SPI provides transmission data register empty status, reception data register not empty status, and busy status. These statuses enable the management of data transmission and reception by monitoring their conditions.

- Tx data register empty flag (TXFE)

  The TXFE flag is set when transmission data register has no data to send. The flag is cleared by hardware when the transmission data register is not empty.

- Rx data register not empty flag (RXFNE)

  The RXFNE flag is set when reception data register is not empty. The flag is cleared by hardware only when the data is read from the reception data register.

- Busy flag (BUSY)

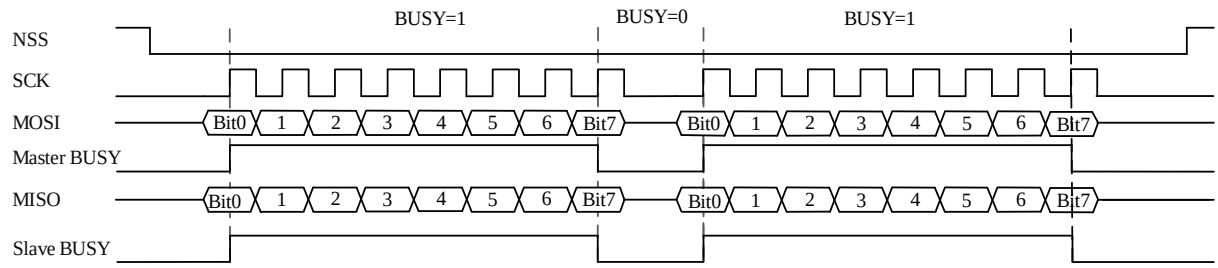
  The BUSY flag is set and cleared by hardware.

  When BUSY is set, it indicates that a data transfer is in progress on the SPI (the SPI bus is busy). The BUSY flag can be used to detect the end of a transfer so that the software can properly disable the SPI and avoid data corruption.

  The BUSY flag is cleared under any one of the following conditions:

  - In master or slave mode, the time interval between the last sampling clock edge of previous data transfer and the first sampling clock edge of next data transfer.
  - In master mode, when the transmission data register is empty and the SPI finishes the last data transmission.
  - In multi-master application, when a master mode fault (MMF) is detected.

Figure 32-6 BUSY flag status



32.3.9 SPI error flags

A SPI interrupt is generated if one of following error flags is set and interrupt is enabled by setting ERRIE bit.

- Overrun flag

  An overrun condition occurs when data is received by the shift register and the

reception data register has no space to store this received data. This can set the OVR flag to 1. When an overrun condition occurs, the newly received value does not overwrite the previous one in the reception data register, and the newly received value is discarded. Clear the OVR bit by setting the OVRCF bit in the SPIx_ICR register.

- Master mode fault

  When the SPI operates in master mode with the NSS pin configured as an input, pulling the NSS signal low will set the MMF bit, indicating a master mode fault. This condition is typically encountered in multi-master system.

  Master mode fault affects the SPI interface by hardware in the following ways:

  - The SPE bit is cleared, and the SPI interface is disabled.
  - The MSTR bit is cleared, thus forcing the SPI into slave mode.

  Clear the MMF bit by setting the MMFCF bit in the SPIx_ICR register.

  To avoid any multiple master conflict in a system comprising several MCUs, the NSS pin must be pulled high during the MMF bit clearing duration. The SPE and MSTR bits can be re-enabled by setting them to 1 after clearing the MMF bit. In a slave device the MMF bit cannot be set except as the result of a previous multi-master conflict.

32.3.10 SPI initialization

The configuration procedure is almost the same for master and slave, except for BR[2:0] and NSSOE field. For specific mode, follow the dedicated sections. The SPI initialization steps are as follows:

1) Write proper GPIO register: Configure NSS, SCK, MOSI, and MISO pins. Refer to: *General-purpose I/Os (GPIO)*. In master mode, the NSS pin may be used as a GPIO when hardware-managed NSS is disabled.
2) Clear the SPE bit in the SPIx_CR1 register.
3) Write to the SPIx_CR1 register:
   - Configure the serial clock baud rate using the BR[2:0] bits. The step is not required in slave mode.
   - Configure the CPOL and CPHA bits combination to define one of the four relationships between the data transfer and the serial clock.

- Configure the LSBFIRST bit to select the data transfer order.
- Configure the MSTR bit to select master or slave mode.
- Configure the NSSOE bit to select the NSS state. The step is not required in slave mode.

4) Set the SPE bit in the SPIx_CR1 register.

32.3.11 Data transmission and reception procedures

Polling and interrupt procedure

The write access to the SPIx_DR register can be managed by the TXFE status. A write operation may be performed when the TXFE flag is set to 1.

The read access to the SPIx_DR register can be managed by the RXFNE status. A read operation may be performed when the RXFNE flag is set to 1.

TXFE and RXFNE events can be judged and handled through either polling or interrupts methods. When using interrupt method, set the corresponding interrupt bits in the SPx_CR1 register, and an interrupt will be generated when an event occurs. Refer to: *SPI interrupts*.

To ensure the integrity of the last data communication, the correct procedure is:

1) Wait until TXFE = 1 (no more data to transmit).
2) Wait until BUSY = 0 (the last data transfer is processed).
3) Read the received data until RXFNE = 0 or clear the reception data register.

DMA procedure

When DMA transmission is enabled via setting the DMAT bit in the SPIx_CR1 register, a DMA request is issued each time the TXFE bit is set to 1.

When DMA reception is enabled via setting the DMAR bit in the SPIx_CR1 register, a DMA request is issued each time the RXFNE bit is set to 1.

When the SPI is only used to transmit data, it is possible to enable only the SPI transmission DMA channel. In this case, the OVR flag is set because the data received is not read.

In transmission mode, when the DMA has written all the data to be transmitted, the TXFE and BUSY flags can be monitored to ensure that the SPI communication is complete. This is required to avoid corrupting the last transmission before disabling the SPI.

When starting communication using DMA, these steps must be followed in order:

1) Configure the data transmission and reception DMA channels, refer to: *Direct memory access controller (DMA)*.
2) Enable DMA channel for SPI reception.
3) Enable DMA Rx mode by setting the DMAR bit in the SPI_CR1 register.
4) Enable DMA channel for SPI transmission.
5) Enable DMA Tx mode by setting the DMAT bit in the SPI_CR1 register.

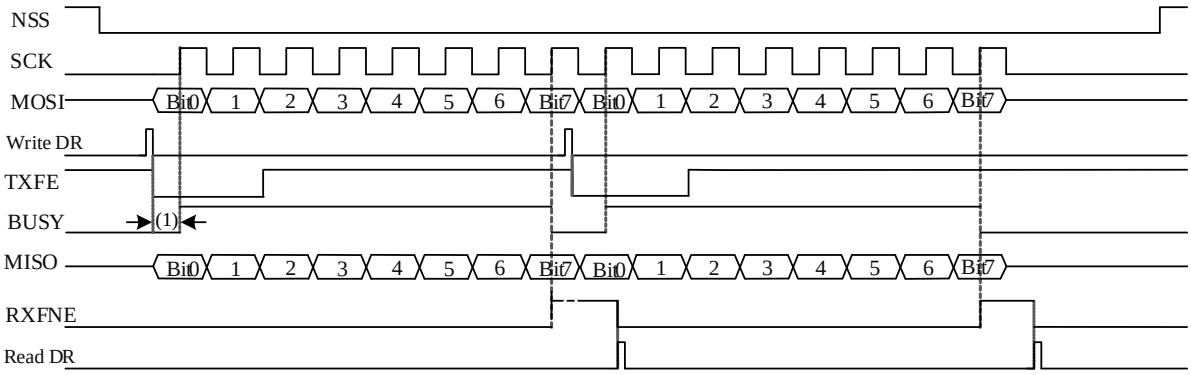
To close communication it is mandatory to follow these steps in order:

1) Wait for DMA transmission and reception channels transfer complete.
2) Wait for Tx data register is empty and the TXFE flag is set.
3) Wait for BUSY flag is cleared, indicating the last data frame is processed.
4) Wait for Rx data register is empty and the RXFNE flag is cleared.
5) Clear the DMAR and DMAT bits in the SPI_CR1 register.
6) Disable the DMA transmission and reception channel, refer to: *Direct memory access controller (DMA)*.

The clock signal is provided by the master device until the transmit data register and shift register are empty, after which the clock stops until the master initiates another transmission. The communication process is illustrated in the following figure.

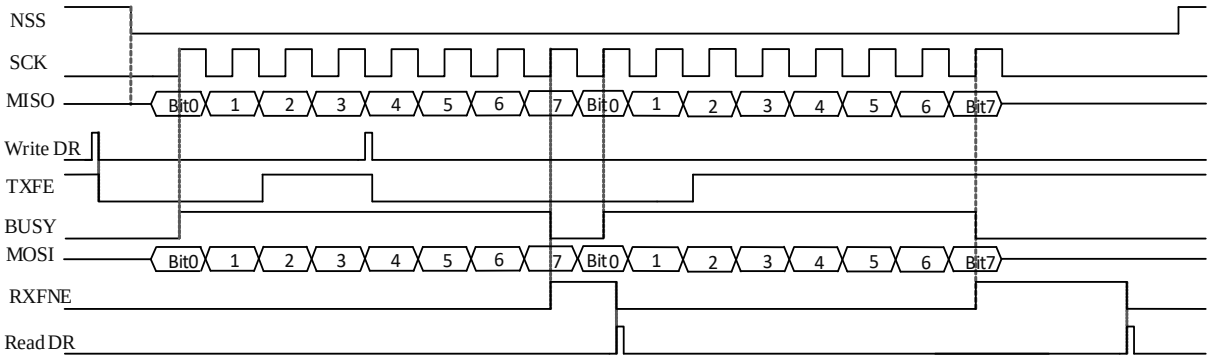
The following diagram shows the communication timing for 2 bytes data transfer in SPI master mode. The SCK pin has a low-level idle state, and the first edge on the SCK pin captures the first data bit transacted.

Figure 32-7 Master communication timing



The following diagram shows the communication timing for 2 bytes data transfer in SPI slave mode. The SCK pin has a low-level idle state, the first edge on the SCK pin captures the first data bit transacted.

Figure 32-8 Slave communication timing



32.3.12 SPI disable

When disabling the SPI, ensure that the last communication data transfer is complete. refer to section: *Data transmission and reception procedures*, otherwise it may corrupt ongoing interactions. The SPI will be disable after SPE bit in the SPIx_CR1 register is cleared.

32.4 SPI interrupts

The table below gives the list of SPI interrupt requests.

Table 32-5 SPI interrupt requests

Interrupt event	Event flag	Enable control bit	Interrupt clear method
Transmission data register empty flag	TXFE	TXFEIE	Write the SPIx_DR register making Tx data register not empty
Reception data register not empty flag	RXFNE	RXFNEIE	Read the SPIx_DR register making Rx data register empty
Overrun error	OVR	ERRIE	Set the OVRCF bit in the SPIx_ICR register
Master mode fault	MMF		Set the MMFCF bit in the SPIx_ICR register

32.5 SPI registers

The SPI registers can only be accessed by words (32-bit).

Table 32-6 SPI base address

Peripheral	Base address
SPI1	0x4001 3000
SPI2	0x4000 3800

32.5.1 SPI control register 1 (SPIx_CR1)

Address offset: 0x00

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.						TXFEIE	Res.				RXFNEIE	Res.	ERRIE	DMAT	DMAR
						rw					rw		rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				SSM	NSSOE	Res.	LSB FIRST	Res.	MSTR	CPOL	CPHA	BR[2:0]			SPE
				rw	rw		rw		rw	rw	rw	rw	rw	rw	rw

Bits	Name	Description
31:26	Reserved	Must be kept at reset value
25	TXFEIE	Tx data register empty interrupt enable 0: Disable 1: Enable
24:21	Reserved	Must be kept at reset value
20	RXFNEIE	Rx data register not empty interrupt enable 0: Disable 1: Enable
19	Reserved	Must be kept at reset value
18	ERRIE	Error interrupt enable This bit controls the generation of an interrupt when an error occurs

		(MMF or OVR) 0: Disable 1: Enable
17	DMAT	DMA Tx enable 0: Disable 1: Enable
16	DMAR	DMA Rx enable 0: Disable 1: Enable
15:12	Reserved	Must be kept at reset value
11	SSM	Software slave management 0: The slave SPI is selected by the NSS pin input 1: The slave SPI is always selected, and ignores the NSS pin input *Note: This bit is valid only in master mode, and is forced to 0 in slave mode by hardware.*
10	NSSOE	NSS output enable 0: NSS output is disabled in master mode 1: NSS output is enabled in master mode *Note: This bit is valid only in master mode, and is forced to 0 in slave mode by hardware.*
9	Reserved	Must be kept at reset value
8	LSBFIRST	Frame format 0: Data is transmitted / received with the MSB first 1: Data is transmitted / received with the LSB first *Note: This bit can only be written when SPI is disabled (SPE = 0).*
7	Reserved	Must be kept at reset value

6	MSTR	Master selection 0: Slave configuration 1: Master configuration *Note: This bit can only be written when SPI is disabled (SPE = 0), and should be written after clock polarity configuration (CPOL bit).*
5	CPOL	Clock polarity 0: SCK to 0 when idle 1: SCK to 1 when idle *Note: This bit can be written only when SPI is disabled (SPE = 0).*
4	CPHA	Clock phase 0: The first clock transition is the first data capture edge 1: The second clock transition is the first data capture edge *Note: This bit can only be written when SPI is disabled (SPE = 0).*
3:1	BR[2:0]	Baud rate control 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: Reserved (keep the previously valid configuration) *Note: These bits can only be written when SPI is disabled (SPE = 0).*
0	SPE	SPI enable 0: Disable 1: Enable

32.5.2 SPI control register 2 (SPIx_CR2)

Address offset: 0x04

Reset value: 0x0000 0001

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

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

Bits	Name	Description
31:1	Reserved	Must be kept at reset value
0	NSSO	NSS pin output This bit is used in conjunction with NSSOE. The NSS output is valid when NSSOE is 1. 0: NSS output 0 1: NSS output 1 *Note: This bit is valid only in master mode, and is forced to 0 in slave mode by hardware.*

32.5.3 SPI data clear register (SPIx_DATACLR)

Address offset: 0x08

Reset value: 0x0000 0000

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

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

Bits	Name	Description
31:2	Reserved	Must be kept at reset value
1	TXCLR	Tx data clear When this bit is written to 1, the Tx data register will be cleared and TXFE flag will be set to 1.

0	RXCLR	Rx data clear When this bit is written to 1, the Rx data register will be cleared and RXFNE flag will be set to 0.

32.5.4 SPI interrupt and status register (SPIx_ISR)

Address offset: 0x0C

Reset value: 0x0000 0100

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.							TXFE	Res.	RXFNE	Res.	BUSY	Res.	OVR	Res.	MMF
							r		r		r		r		r

Bits	Name	Description
31:9	Reserved	Must be kept at reset value
8	TXFE	Tx data register empty This bit is set to 1 when the Tx data register is empty, otherwise cleared to 0 by hardware. 0: Tx data register not empty 1: Tx data register empty
7	Reserved	Must be kept at reset value
6	RXFNE	Rx data register not empty This bit is set to 1 when the Rx data register is not empty, otherwise cleared to 0 by hardware. 0: Rx data register empty 1: Rx data register not empty
5	Reserved	Must be kept at reset value
4	BUSY	Busy flag

		This bit is set and cleared by hardware. 0: SPI not busy 1: SPI is busy in communication
3	Reserved	Must be kept at reset value
2	OVR	Overrun flag This bit is set to 1 when the Rx data register is full and the shift register receives a new data. Writing 1 to OVRCF bit in the SPIx_ICR register will clear this bit. 0: No overrun occurred 1: Overrun occurred
1	Reserved	Must be kept at reset value
0	MMF	Master mode fault flag When the SPI operates in multi-master mode, this bit will be set if the NSS pin detects low-level input. Writing 1 to MMFCF bit in the SPIx_ICR register will clear this bit. 0: No master mode fault occurred 1: Master mode fault occurred *Note: When master mode fault occurred, the MSTR and SPE will be cleared by hardware. The MSTR and SPE can not be set when MMF flag is 1.*

32.5.5 SPI interrupt flag clear register (SPIx_ICR)

Address offset: 0x10

Reset value: 0x0000 0000

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

Bits	Name	Description
31:3	Reserved	Must be kept at reset value
2	OVRCF	Overrun flag clear Writing 1 to this bit clears the OVR flag in the SPIx_ISR register.
1	Reserved	Must be kept at reset value
0	MMFCF	Master mode fault flag clear Writing 1 to this bit clears the MMF flag in the SPIx_ISR register.

32.5.6 SPI data register (SPIx_DR)

Address offset: 0x14

Reset value: 0x0000 0000

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

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

Bits	Name	Description
31:8	Reserved	Must be kept at reset value
7:0	DR[7:0]	Data register Data received or to be transmitted. When the data register is read, the Rx data register is accessed while the writing to data register accesses Tx data register.

33 Debug support (DBG)

33.1 Introduction

The device is built around a Cortex-M0+ core which integrates hardware debugging capabilities, enabling complex debugging operations. The SWD (Serial Wire Debug) port allow the core to be stopped either on a given instruction fetch (breakpoint) or data access (watchpoint). When stopped, the core's internal state and the system's external state may be examined by the debug interface. Once the MCU is connected to the debugger host, debugging operations can be performed through the debug interface.

33.2 DBG main features

The debug features embedded in the Cortex-M0+ core are a subset of the Arm CoreSight Design Kit.

The Arm Cortex-M0+ core provides integrated on-chip debug support. It is comprised of: SW-DP (Serial wire debug port), DWT (Data watchpoint trigger), and BPU (Break point unit).

33.3 SWD port

33.3.1 SWD port pins

The GPIO pins PA13 and PA14 can be multiplexed as the SWDIO and SWCLK signals of the SWD port. These pins are available on all packages.

Table 33-1 SW debug port pins

SW-DP pin name	Debug assignment	Pin assignment
SWDIO	Serial wire data input/output	PA13
SWCLK	Serial wire clock	PA14

33.3.2 SW-DP pin assignment

After system reset, PA13 is multiplexed as SWDIO with a pull-up resistor and PA14 is multiplexed as SWCLK with a pull-down resistor. If the SWD port is not used, PA13 and PA14 can be configured as I/O by setting the *GPIO port pull-up/pull-down register (GPIOx_MODE)*. For more details, please refer to *I/O pin alternate function multiplexer and mapping*.

33.4 BPU

The Cortex-M0+ BPU implementation provides four breakpoint registers.

33.5 DWT

The Cortex-M0+ DWT implementation provides one watchpoint register set.

33.6 MCU debug component (DBG)

The MCU debug component helps the debugger provide support for:

- Low-power modes
- Clock control for timers and watchdog during a breakpoint

33.6.1 Debug support for low-power modes

Debugging functionality supporting Sleep Mode and Stop Mode.

The core does not allow HCLK to be turned off during a debug session. As these are required for the debugger connection, during a debug, they must remain active.

For this, the debugger host must first set some debug configuration registers to enable debugging during low-power mode:

- In Sleep mode, the HCLK domain remains active. Consequently, this mode does not impose any restrictions on the standard debug features.
- In Stop mode, setting the DBG_STOP bit in the DBG_CR register to 1 switches the system clock to RCHSYS, which provides the clock signal to the HCLK domain.

33.6.2 Debug support for timers and watchdog

When the MCU's core is halted in debug mode, it is necessary to choose whether the counters of timers or the watchdog should continue counting:

- The TIMx counters either continue to work normally or stop, depending on the configuration of the TIMx_HOLD (x=3, 4, 5 or 8) bit in the *APB1 freeze register (DBG_APB1_FZ)*.
- The LPTIMx counters either continue to work normally or stop, depending on the configuration of the LPTIMx_HOLD (x=1 or 2) bit in the *APB1 freeze register (DBG_APB1_FZ)*.

- The RTC counters either continue to work normally or stop, depending on the configuration of the RTC_HOLD bit in the *APB1 freeze register (DBG_APB1_FZ)*.
- The IWDG counters either continue to work normally or stop, depending on the configuration of the IWDG_HOLD bit in the *APB1 freeze register (DBG_APB1_FZ)*.
- The WWDG counters either continue to work normally or stop, depending on the configuration of the WWDG_HOLD bit in the *APB1 freeze register (DBG_APB1_FZ)*.

33.7 DBG registers

The DBG registers can only be accessed by words (32-bit).

Table 33-2 DBG base address

Peripheral	Base address
DBG	0x4001 5800

33.7.1 DBG configuration register (DBG_CR)

Address offset: 0x00

Reset value: 0x0000 0000

Note: This register can only be reset by POR reset.

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

Bits	Name	Description
31:1	Reserved	Must be kept at reset value
0	DBG_STOP	Debug Stop mode 0: Disable 1: Enable

33.7.2 APB1 freeze register (DBG_APB1_FZ)

Address offset: 0x04

Reset value: 0x0000 0000

Note: This register can only be reset by POR reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	LPTIM2_HOLD	LPTIM1_HOLD	Res.										IWDG_HOLD	WWDG_HOLD	RTC_HOLD
	rw	rw											rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.									TIM8_HOLD	Res.		TIM5_HOLD	TIM4_HOLD	TIM3_HOLD	Res.
									rw			rw	rw	rw	

Bits	Name	Description
31	Reserved	Must be kept at reset value
30	LPTIM2_HOLD	This bit enables/disables the clock to the counter of LPTIM2 when the core is halted 0: Enable 1: Disable
29	LPTIM1_HOLD	This bit enables/disables the clock to the counter of LPTIM1 when the core is halted 0: Enable 1: Disable
28:19	Reserved	Must be kept at reset value
18	IWDG_HOLD	This bit enables/disables the clock to the counter of IWDG when the core is halted 0: Enable 1: Disable
17	WWDG_HOLD	This bit enables/disables the clock to the counter of WWDG when the core is halted 0: Enable 1: Disable
16	RTC_HOLD	This bit enables/disables the clock to the counter of RTC when the core is halted 0: Enable 1: Disable
15:7	Reserved	Must be kept at reset value
6	TIM8_HOLD	This bit enables/disables the clock to the counter of TIM8 when the core is halted 0: Enable 1: Disable

5:4	Reserved	Must be kept at reset value
3	TIM5_HOLD	This bit enables/disables the clock to the counter of TIM5 when the core is halted 0: Enable 1: Disable
2	TIM4_HOLD	This bit enables/disables the clock to the counter of TIM4 when the core is halted 0: Enable 1: Disable
1	TIM3_HOLD	This bit enables/disables the clock to the counter of TIM3 when the core is halted 0: Enable 1: Disable
0	Reserved	Must be kept at reset value

34 Device electronic signature

The device electronic signature can be read using the debug interface or by the CPU. It contains factory-programmed identification data, which allow user firmware or other external devices to automatically match to the characteristics of MCU.

34.1 Unique device ID register (96 bits)

The 96-bit unique device identifier provides a reference number which is unique for any device and in any context. These bits cannot be altered by the user, and its primary application scenarios include:

- For use as serial numbers
- For use as part of the security keys in order to increase the security of code in Flash memory while using and combining this unique ID with software cryptographic primitives and protocols before programming the internal Flash memory
- To activate secure boot processes, etc.

Address: 0x1FFF 0340, For further details, refer to *Table: Device information*.

34.2 Device type register

Address: 0x1FFF 03A4, For further details, refer to *Table: Device information*.

34.3 Flash memory size data register

Address: 0x1FFF 03A8, For further details, refer to *Table: Device information*.

34.4 SRAM size data register

Address: 0x1FFF 03AC, For further details, refer to *Table: Device information*.

35 Revision history

Table 35-1 Revision history

Date	Revision	Changes
14-Nov-2023	V1.0	Initial release.
26-Dec-2023	V1.1	1. Enhance Functional Descriptions and Standardize Paragraph Formatting. 2. Add SSOP24 Package Information.
27-Feb-2024	V1.2	1. RCC: Enhance the Description of LXTAL Drive Modes. 2. ADC: Enhance the Description of ADC Settling Time. 3. COMP: Enhance the Description of 6-bit DAC Settling Time.
10-Apr-2024	V1.3	1. ADC: Correct the terminology for TS Sampling Time and Low-Frequency Trigger Time. 2. LCD: Update the circuit diagram.
28-Jun-2024	V1.4	1. RCC: To improve the compatibility of the LXTAL crystal, recommendations for register control bit configurations have been added. 2. LCD: Enhance the description of the LCD_CR register modifications to explicitly specify the correct sequence for enabling and configuring the LCD. 3. SPI: Update the SPI initialization process to explicitly specify the required configuration sequence for clock polarity and master/slave mode.
10-Sep-2024	V1.5	1. I2C: Correct the behavior description of the TXIS flag in Figures 29-9 and 29-10. Enhance the details of the action triggered when the TXE control bit in the I2Cx_ISR register is set to 1.
05-Nov-2024	V1.6	1. PMU: Elaborate the descriptions to clarify the correlations between the V_{BAT} pin power connection configurations and their corresponding V_{BAT} mode settings.
03-Dec-2024	V1.7	1. LCD: Add explanations for the charge pump clock divider configuration and provide suggested division values. 2. ADC: Elaborate on the internal switching operations between the V_{BAT} and V_{DDA} channels. 3. DMA: Add descriptions explaining the procedures for stopping and restarting DMA operations.
30-Dec-2024	V1.8	1. Flash: Supplement option byte factory default value 2. LPTIM: LPTIM enable/disable timing description 3. USART/LPUART: Recovering DMA transfer after communication errors in DMA communication mode
15-Jan-2025	V1.9	1. PMU: Corrected BGR stabilization description 2. VREFBUF: Corrected VREFBUF stabilization description

Date	Revision	Changes
		3. SYSCFG: Corrected 6bit DAC stabilization description

36 Contact information

Website: www.hed.com.cn

Address: Building C, CEC Network Security and Information Technology Base, South Region of Future Science and Technology Park, Beiqijia County, Changping District, Beijing, China

Post code: 102209

Feel free to contact us for any comment or suggestion during purchase and use.