CH32L103 Reference Manual

V1.8

https://wch-ic.com

Overview

CH32L103 is an industrial-grade general-purpose microcontroller based on 32-bit RISC-V instruction set and architecture. Adopting QingKe V4C core, it supports hardware interrupt stack and improves interrupt response efficiency. This series of products are loaded with rich peripheral interfaces and functional modules. Its internal organizational structure meets the low-cost and low-power embedded application scenarios.

This manual is aimed at user's application development and provides detailed information about the use of CH32L103 products.

RISC-V core version comparison overview

Features / Core versions	Instruction set	Hardware stack levels	Interrupt nesting levels	Fast interrupt channels	Integer division cycles	Vector table model	Extended instruction	Memory protection
QingKeV4B	IMAC	2	2	4	9	Address or command	Supported	None
QingKeV4C	IMAC	2	2	4	5	Address or command	Supported	Standard
QingKeV4F	IMAFC	3	8	4	5	Address or command	Supported	Standard

Abbreviated description of the bit attribute in the register:

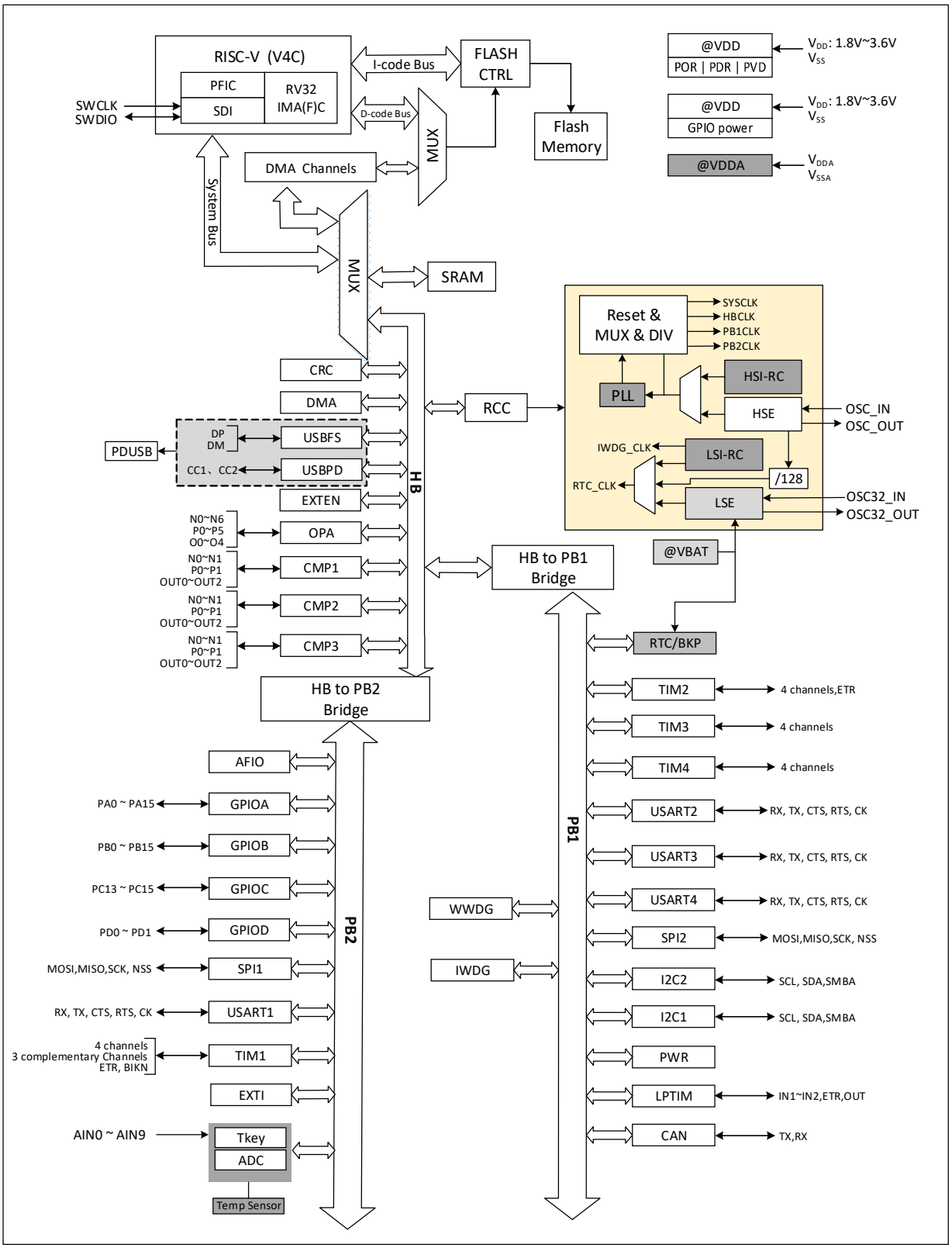
Register bit attributes	Property description
RF	Read-only attribute, read a fixed value.
RO	Read-only attribute, changed by hardware.
RZ	Read-only attribute, auto bit clear 0 after read operation.
WO	Write only attribute (not readable, read value uncertain)
WA	Write-only attribute, writable in Safe mode.
WZ	Write only attribute, auto bit clear 0 after write operation.
RW	Readable and writable.
RWA	Readable, writable in Safe mode.
RW1	Readable, write 1 valid, write 0 invalid.
RW0	Readable, write 0 valid, write 1 invalid.
RW1T	Readable, write 0 invalid, write 1 flipped.
SC	Auto cleared.

Chapter 1 Memory and Bus Architecture

1.1 Bus Architecture

This series of products are general-purpose microcontrollers designed based on the RISC-V instruction set, and the core, arbitration unit, DMA module, SRAM storage and other parts of its architecture interact with each other through multiple sets of buses. Its system block diagram is shown in Figure 1-1.

Figure 1-1 System block diagram



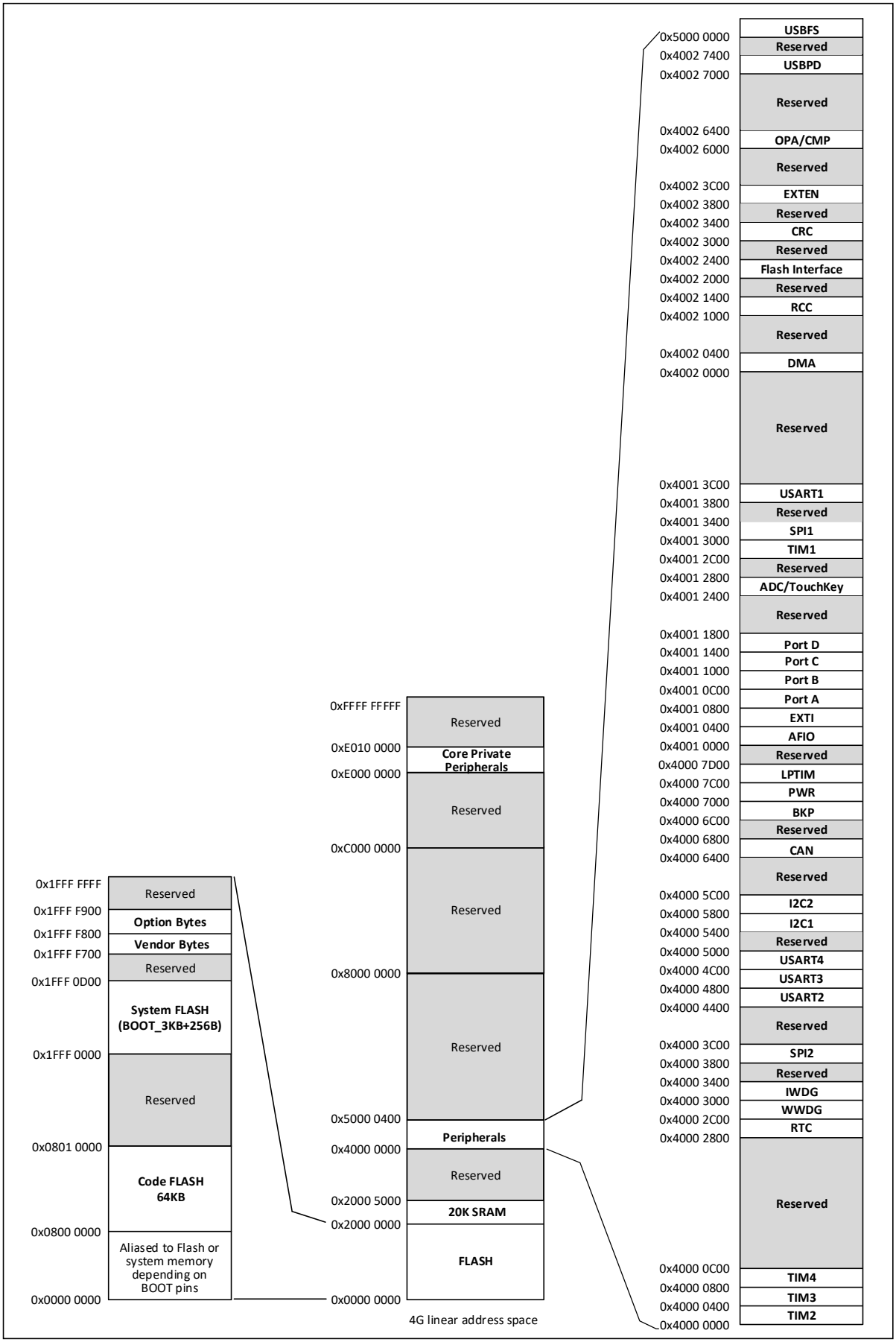
The system is equipped with: General-purpose DMA controller to reduce the CPU burden and improve efficiency; clock tree hierarchy management to reduce the total power consumption of peripherals, as well as data protection mechanisms, clock security system protection mechanisms and other measures to increase system stability.

1.2 Memory Map

The CH32L103 family contains program memory, data memory, core registers, peripheral registers, and more, all addressed in a 4GB linear space.

System storage stores data in small-end format, i.e., low bytes are stored at the low address and high bytes are stored at the high address.

Figure 1-2 Storage image



1.2.1 Memory Allocation

Built-in 20KB SRAM, starting address 0x20000000, supports byte, half-word (2 bytes), and full-word (4 bytes) access.

Built-in up to 64KB program Flash memory (CodeFlash) for storing user applications.

Built-in 3K+256B System memory (Bootloader) for storing the system bootloader (Factory-cured bootloader).

Built-in 256B space for vendor configuration word storage, factory-cured and unmodifiable by users.

Built-in 256B space for user option bytes storage.

1.3 Boot Configuration

The system can select 3 different boot modes via the BOOT0 and BOOT1 pins.

Table 1-1 Boot Mode

BOOT0	BOOT1	Boot Mode
0	X	Boot from Program Flash Memory
1	0	Boot from System Memory
1	1	Boot from internal SRAM

The user selects the boot mode after a reset by setting the status value of the BOOT pin. A system reset or a power reset causes the value of the BOOT pin to be relocked.

The program flash memory, system memory and internal SRAM are accessed differently depending on the boot mode:

- When booted from Program Flash Memory, the program flash memory address is mapped to the 0x00000000 address area and is also able to be accessed in the original address area 0x08000000.
- When booted from System Memory, the system memory address is mapped to the 0x00000000 address area and is also accessible in the original address area 0x1FFF0000.
- Booting from internal SRAM is only accessible from the 0x20000000 address area.

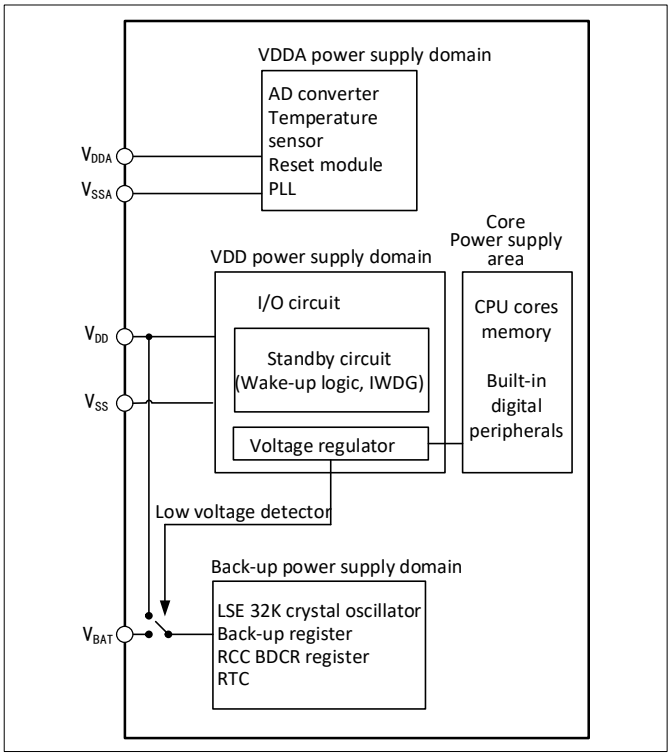
Chapter 2 Power Control (PWR)

2.1 Overview

The system operating voltage V_{DD} ranges from 1.8 to 3.6 V. The built-in voltage regulator provides the low-voltage power supply required by the core. When the main power supply, V_{DD}, is powered down, a backup power source such as a battery can provide power for the real-time clock (RTC) and backup registers through the V_{BAT} pin. If no backup power is required, it is recommended that V_{DD} be connected directly to the V_{BAT} pin.

The V_{DDA} and V_{SSA} pins are dedicated to powering analog-related circuits in the system, including ADC, temperature sensor, etc.

Figure 2-1 Block diagram of power supply structure



After the main power supply V_{DD} is powered down, the analog switch is switched to V_{BAT} and the backup domain is powered by the V_{BAT} pin, at this time, PC13 to 15 cannot be used as GPIOs, and only the following functions can be used:

- PC13 can be used as TAMPER pin, RTC alarm clock or second output.
- PC14 and PC15 can only be used as LSE pins.

When the main power supply VDD is powered on stably, the system automatically switches the backup area to be powered by VDD, and PC13~15 can be used as a GPIO function.

When the PC13~15 pin is output as a GPIO, the speed must be limited to less than 2MHz, the maximum load capacitance must be 30pF, and must not be used in continuous output and suction current situations, such as LED drive.

Note: In the process of restoring the power supply of the main power supply V_{DD}, the internal V_{BAT} power supply is still connected to the external standby power supply through the corresponding V_{BAT} pins. If the V_{DD} is stable within

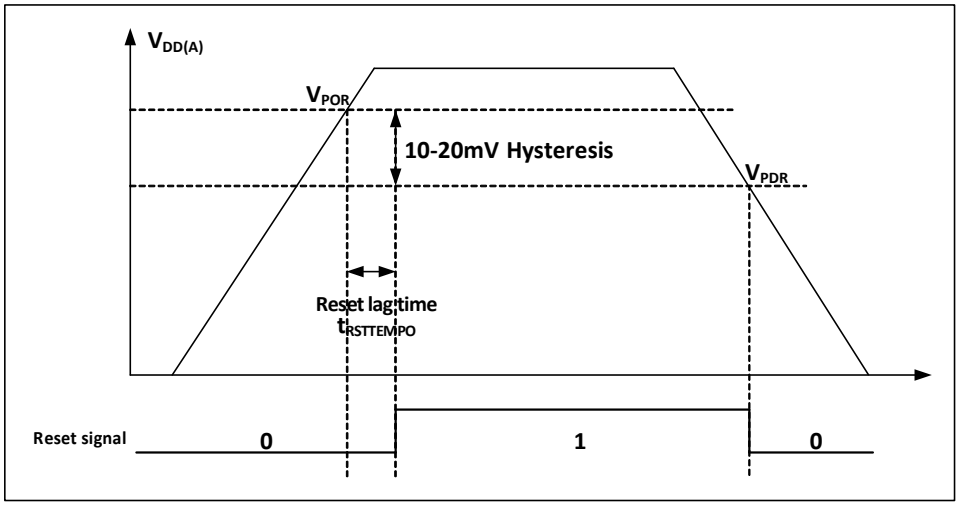
less than the reset lag time $t_{RSTTEMPO}$ and is more than 0.6V higher than the value of V_{BAT}, then there may be a short moment when the current is pumped into the V_{BAT} through the diode between the V_{DD} and the V_{BAT}, and then injected into the battery and other backup power through the V_{BAT} pin. If the backup power supply cannot withstand such instantaneous injection current, it is recommended to add a forward turn-on low voltage drop diode between the backup power supply and the V_{BAT} pin.

2.2 Power Management

2.2.1 Power-on Reset and Power-down Reset

The system integrated a power-on reset POR and a power-down reset PDR circuit. When the chip supply voltage V_{DD} falls below the corresponding threshold voltage, the system is reset by the relevant circuit, and no additional external reset circuit is required. Please refer to the corresponding datasheet for the parameters of the power-on threshold voltage V_{POR} and the power-down threshold voltage V_{PDR}.

Figure 2-2 Schematic diagram of the operation of POR and PDR



2.2.2 Programmable Voltage Detector

The programmable voltage detector, PVD, is mainly used to monitor the change of the main power supply of the system, compared with the threshold voltage set by PLS[2:0] of the power control register PWR_CTLR, and together with the setting of the external interrupt register (EXTI), the relevant interrupt can be generated in order to notify the system to carry out the pre-dropout operation, such as the data saving, in time.

The specific configuration is as follows:

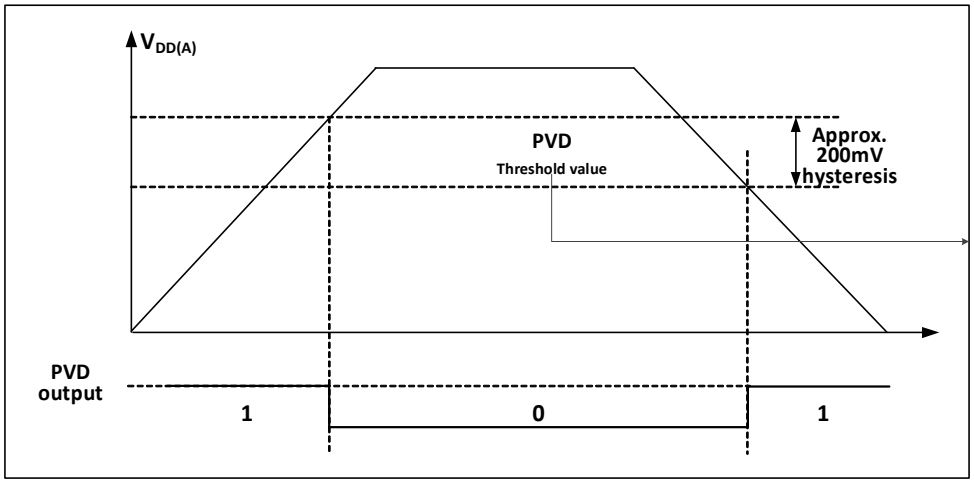
1) Set the PLS[2:0] fields of the PWR_CTLR register to select the voltage threshold to be monitored.

2) Optional interrupt handling. the PVD function internally connects to the rising/falling edge trigger setting on line 16 of the EXTI module, turns on this interrupt (configures EXTI), and generates a PVD interrupt when V_{DD} falls below the PVD threshold or rises above the PVD threshold.

3) Set the PVDE bit of the PWR_CTLR register to enable the PVD function.

4) Read the PVD0 bit of PWR_CSR status register to obtain the relationship between the current system main power supply and the threshold value set by PLS[2:0], and perform the corresponding soft processing. When the V_{DD} voltage is higher than the PLS[2:0] setting threshold, the PVD0 position is 0; when the V_{DD} voltage is lower than the PLS[2:0] setting threshold, the PVD0 position is 1.

Figure 2-3 Schematic diagram of PVD operation



2.3 Low-power Modes

After a system reset, the microcontroller is in a normal operating state (run mode), where system power can be saved by reducing the system main frequency or turning off the unused peripheral clock or reducing the operating peripheral clock. If the system does not need to work, you can set the system to enter low-power mode and let the system jump out of this state by specific events.

Microcontrollers currently offer 3 low-power modes, divided in terms of operating differences between processors, peripherals, voltage regulators, etc.

- Sleep mode: The core stops running and all peripherals (including core private peripherals) are still running.
- Stop mode: Stops all clocks and the system continues to run after waking up. Stop mode is divided into 4 gears, respectively: Stop mode 1, Stop mode 2, Stop mode 3, Stop mode 4, the corresponding configuration method is shown in Table 2-1, stop mode four gears after the configuration of the current reference CH32L103DS0 manual.
- Standby mode: Stop all clocks and reset the microcontroller after wakeup (power reset).

Table 2-1 Low-power mode list

Mode	Entry	Wake-up source	Effect on clock	Voltage regulator
SLEEP	WFI	Any interrupt	Core clock OFF, no effect on other clocks	Normal
	WFE	Wake-up event		
STOP	Set SLEEPDEEP to 1 Clear PDDS to 0 WFI or WFE	Any external interrupt/event, NRST pin reset, WKUP pin rising edge, PVD output, RTC alarm event, USB wake-up signal, USB PD wake-up signal, CMP wake-up signal, LPTIM wake-up signal	Disable HSE, HIS, PLL and peripheral clock	Stop mode1: LPDS=0，PDDS=0
				Stop mode 2 (LDO energy saving mode): LPDS=0, PDDS=0, AUTO_LDO_EC=1 or LPDS=0, PDDS=0, LDO_EC=1
				Stop mode 3: RAMLV=0,

				PDDS=0，LPDS=1 Stop mode 4: RAMLV=1, PDDS=0，LPDS=1
STANDBY	Set SLEEPDEEP to 1 Set PDDS to 1 WFI or WFE	WKUP pin rising edge, RTC alarm event, NRST pin reset, IWDG reset, PVD's output. *Note: Any of the EXTI0~EXTI17 external events can also wake up the system, but the system is reset after waking up.*	Disable HSE, HIS, PLL and peripheral clock	Off

Note: The SLEEPDEEP bit belongs to the core private peripheral control bit, reference PFIC_SCTLR register.

2.3.1 Low-power Configuration Options

- WFI and WFE

WFI: The microcontroller is woken up by an interrupt source with interrupt controller response, and the interrupt service function will be executed first after the system wakes up (except for microcontroller reset).

WFE: The wakeup event triggers the microcontroller to exit low-power mode. Wake-up events include:

1) Configure an external or internal EXTI line to event mode, when no interrupt controller needs to be configured.
2) Or configure an interrupt source, equivalent to a WFI wakeup, where the system prioritizes the execution of the interrupt service function.
3) Or configure the SEVONPEND bit to turn on peripheral interrupt enable, but not interrupt enable in the interrupt controller, and the interrupt pending bit needs to be cleared after the system wakes up.

- SLEEPONEXIT

  Enable: After executing the WFI or WFE instruction, the microcontroller ensures that all pending interrupt services are exited and then enters low-power mode.

  Disable: The microcontroller enters low-power mode immediately after executing the WFI or WFE command.

- SEVONPEND

  Enable: All interrupts or wake-up events can wake up the low-power consumption entered by executing WFE.

  Disable: Only interrupts or wake-up events enabled in the interrupt controller can wake up the low-power consumption entered by executing WFE.

2.3.2 Sleep Mode

In this mode, all IO pins keep their state in Run mode and all peripheral clocks are normal, so try to turn off useless peripheral clocks before entering Sleep mode to reduce low-power consumption. This mode takes the shortest time to wake up.

Enter: Configure core register control bit SLEEPDEEP=0, power control register PDDS=0, execute WFI or WFE, optionally SEVONPEND and SLEEPONEXIT.

Exit: Arbitrary interrupt or wakeup event.

2.3.3 Stop Mode

Stop mode is a combination of peripheral clock control mechanisms based on the core's deep sleep mode

(SLEEPDEEP) and allows the voltage regulator to operate in a much lower power consumption state. In this mode the high frequency clock (HSE/HSI/PLL) domain is switched off, the SRAM and register contents are maintained and the IO pin state is held. The system can continue to run after this mode wakes up and the HSI is called the default system clock.

If flash programming is in progress, the system does not enter stop mode until access to memory is complete; if access to the PB is in progress, the system does not enter stop mode until access to the APB is complete.

Workable modules in stop mode: Independent Watchdog Dog (IWDG), Real Time Clock (RTC), Low Frequency Clock (LSI/LSE), Low Power Timer (LPTIM).

Entry: Configuration of the core register control bit SLEEPDEEP=1, PDDS=0 of the power control register, optional LPDS bit, execution of WFI or WFE, optional SEVONPEND and SLEEPONEXIT. Before entering the STOP voltage regulator low-power mode (PDDS=0, LPDS=1), preset FLASH_LP[1:0] = 10b, i.e. FLASH low power mode 1.

Exit: Any external interrupt/event, external reset on the NRST pin, output of PVD, RTC alarm clock, wake-up signal for USB, wake-up signal for USB PD, wake-up signal for CMP, wake-up signal for LPTIM, and so on.

Different levels of stop mode are selected through different configurations. In stop mode, i.e., PDDS=0 and optional LPDS bit, LPDS=0, the voltage regulator works in stop mode 1; LPDS=0, AUTO_LDO_EC=1, the power system will automatically save energy when the MCU enters stop mode, i.e., it enters stop mode 2, or LPDS=0, LDO_EC=1, also stop mode 2; LPDS=1, the voltage regulator operates in Stop Mode 3. Based on Stop Mode 3, the power consumption is minimized by enabling RAM low voltage mode by configuring RAMLV=1 in PWR_CTLR register, which is called Stop Mode 4.

2.3.4 Standby Mode

The only difference between the Standby mode and the stop mode is that the microcontroller will be reset after exiting under certain specified wake-up conditions and a power reset is performed.

Modules that can work in Standby mode: Independent Watchdog (IWDG), Real Time Clock (RTC), Low Frequency Clock (LSI/LSE).

Entry: Configure the core register control bits SLEEPDEEP = 1, PDDS = 1 for the power control register, perform WFI or WFE, optional SEVONPEND and SLEEPONEXIT, and preset FLASH_LP[1:0] = 10b before entering STANDBY mode, i.e., FLASH low-power mode 1.

Exit:
1) EXTI0~EXTI18 Any external event (exclude interrupt), the microcontroller performs a power reset after this wakeup.
2) Rising edge of WKUP pin, rising edge of RTC alarm event, external reset on NRST pin, IWDG reset, output of PVD, RTC alarm, the microcontroller performs power reset after this wake-up.

In Standby mode, when normal power supply is used, 2K bytes of RAM are not powered down by configuring R2KSTY=1 of PWR_CTLR register, and 18K bytes of RAM are not powered down by configuring R18KSTY=1 of PWR_CTLR register; when VBAT power supply is used, 2K bytes of RAM are not powered down by configuring R2KVBAT=1 of PWR_CTLR register and R18KVBAT=1 to control 18K bytes of RAM from being powered down. On top of that, power consumption can be minimized by configuring RAMLV=1 of PWR_CTLR register to enable RAM low voltage mode.

Note: Putting the microprocessor into Stop or Standby mode in debug mode will lose the debug connection.
R2KSTY=1 controls the address range of 2K byte RAM: 0x20000000-0x20000000+2K
R18KSTY=1 controls the address range of 18K bytes of RAM: 0x20000000+2K-0x20000000+2K+18K

2.3.5 RTC Auto-wakeup

RTC can automatically wake up without external interruption. By programming the time base, it can be awakened periodically from Stop or Standby mode.

The accurate external low frequency 32.768KHz crystal oscillator LSE can be selected as the RTC clock source, or the internal LSI oscillator can be selected as the RTC clock source. The accuracy and power consumption index of LSI is worse than that of LSE.

The RTC alarm clock event can wake the MCU from downtime mode. In order to achieve this function, the external break line 17 needs to be configured, and the RTC needs to be set to generate alarm clock events. To wake up from Standby mode, simply set RTC to generate an alarm clock event.

2.3.6 FLASH Low-power Mode

Before entering the FLASH low-power mode, it is necessary to enable the FLASH to enter the low-power mode by setting the FLASH_LP_REG bit 1 of the power control register (PWR_CTLR), and then set the FLASH_LP bit of the power control register (PWR_CTLR). At this point, the configuration of the MCU's FLASH low-power mode is complete.

2.4 Register Description

Table 2-2 PWR-related registers list

Name	Access address	Description	Reset value
R32_PWR_CTLR	0x40007000	Power control register	0x00000400
R32_PWR_CSR	0x40007004	Power control/status register	0x00000000

2.4.1 Power Control Register (PWR_CTLR)

Offset address: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											RAMLV	R18KVBAT	R2KVBAT	R18KSTY	R2KSTY

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		LDO_EC	AUTO_LDO_EC	FLASH_LP		FLASH_LP_REG	DBP	PLS[2:0]			PVDE	CSBF	CWUF	PDDS	LPDS

Bit	Name	Access	Description	Reset value
[31:21]	Reserved	RO	Reserved	0
20	RAMLV	RW	RAM operates in low voltage mode enable control bit (relatively lower power consumption): 1: On; 0: Off.	0
19	R18KVBAT	RW	18K RAM power switch control bit in Standby mode when	0

			VBAT is powered: 1: Powered; 0: Not powered.	
18	R2KVBAT	RW	2K RAM power switch control bit in Standby mode when VBAT is powered: 1: Powered; 0: Not powered.	0
17	R18KSTY	RW	Standby mode 18K RAM power switch control bit: 1: Powered; 0: Not powered.	0
16	R2KSTY	RW	Standby mode 2K RAM power switch control bit: 1: Powered; 0: Not powered.	0
[15:14]	Reserved	RO	Reserved	0
13	LDO_EC	RW	Power system LDO energy saving mode enable: 1: Energy saving is turned on, approximately 10μA reduction; 0: Does not turn on. *Note: When this bit is turned on, it only supports a maximum total current of no more than 2mA.*	0
12	AUTO_LDO_EC	RW	Preset LDO energy saving mode after entering stop mode: 1: Automatic energy saving when entering stop mode, i.e. stop mode 2; 0: Normal LDO mode.	0
[11:10]	FLASH_LP	RW	Configure the FLASH mode: 00: Idle mode; X1: FLASH low-power mode 0; 10: FLASH low power mode 1. *Note: This bit is preset to FLASH Low Power Mode 1 (FLASH_LP[1:0] = 10b) before entering STANDBY mode or STOP Voltage Regulator Low Power Mode (PDDS=0, LPDS=1).*	01b
9	FLASH_LP_REG	RW	In combination with the FLASH_LP field, the software configures the enable for FLASH to enter low-power mode: 1: FLASH can be enabled to enter low power mode; 0: It is not possible to enable FLASH to enter low-power mode by software.	0
8	DBP	RW	Write enable for the backup domain. This bit must be set to 1 when the RTC clock is the 128th division of the external clock. 1: Write of the RTC and backup registers is allowed; 0: Write of the RTC and backup registers is prohibited.	0
[7:5]	PLS[2:0]	RW	PVD voltage monitoring threshold setting. See the Electrical Characteristics section of the datasheet for a detailed description. 000: Rising edge 1.75V/falling edge 1.70V; 001: Rising edge 1.93V/Falling edge 1.87V; 010: Rising edge 2.14V/Falling edge 2.08V;	000b

			011: Rising edge 2.35V/falling edge 2.28V; 100: Rising edge 2.54V/falling edge 2.46V; 101: Rising edge 2.72V/falling edge 2.63V; 110: Rising edge 2.92V/falling edge 2.83V; 111: Rising edge 3.10V/falling edge 3.01V.	
4	PVDE	RW	Power supply voltage monitoring function enable flag bit: 1: Enable power supply voltage monitoring function; 0: Disable power supply voltage monitoring function.	0
3	CSBF	RW1	Clears the standby status flag bit and reads always 0. 1: Setting 1 clears the SBF standby status flag bit; 0: Clear 0 is invalid.	0
2	CWUF	RW1	Clear the wake-up status flag bit, readout is always 0. 1: Clear the WUF flag bit after 2 system clock cycles after setting 1; 0: Clearing 0 is invalid.	0
1	PDDS	RW	Standby/shutdown mode selection bit for power-down deep sleep scenarios. 1: Enter Standby mode; 0: Enter Shutdown mode, voltage regulator state is controlled by LPDS.	0
0	LPDS	RW	Voltage regulator operating mode selection bit in shutdown mode. PDDS=0, this bit is valid. 1: The voltage regulator operates in low-power mode; 0: The voltage regulator operates in normal mode.	0

Note: Registers BIT16~BIT20 can only be reset by BACKUP, other BITs are reset when waking up from Standby mode.

2.4.2 Power Control/Status Register (PWR_CSR)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							EWUP	Reserved					PVD0	SBF	WUF

Bit	Name	Access	Description	Reset value
[31:9]	Reserved	RO	Reserved	0
8	EWUP	RW	WKUP pin enable bit: 1: WKUP is forced to be configured as an input pull-down state for waking up the MCU from standby; 0: WKUP pin can be used for GPIOs with no wake-up from standby function.	0
[7:3]	Reserved	RO	Reserved	0
2	PVD0	RO	PVD output status flag bit. This bit is valid when PVDE=1	0

			in the PWR_CTLR register. 1: VDD and VDDA are below the PVD threshold set by PLS[2:0]; 0: VDD and VDDA are above the PVD threshold set by PLS[2:0].	
1	SBF	RO	Standby status flag bit, which can be cleared by CSBF position 1. 1: MCU is in Standby mode; 0: MCU is not in Standby mode.	0
0	WUF	RO	Wake-up event status flag bit, which can be cleared by CWUF position 1. 1: A wake-up event or RTC alarm event is detected at the WKUP pin; 0: No wake-up event has occurred.	0

Note: This register remains unchanged after waking up from Standby mode.

Chapter 3 Reset and Clock Control (RCC)

The controller provides different forms of resets and configurable clock tree structures based on the division of power areas and peripheral power management considerations in the application. This section describes the scope of each clock in the system.

3.1 Main Features

- Multiple reset forms
- Multiple clock sources, bus clock management
- Built-in external crystal oscillation monitoring and clock security system
- Independent management of each peripheral clock: Reset, On, Off
- Supports internal clock output

3.2 Reset

The controller provides 3 forms of reset: Power Reset, System Reset, and Backup Domain Reset.

3.2.1 Power Reset

When a power reset occurs, all registers except the backup domain (which is powered by VBAT) will be reset.

The conditions for its production include:

- Power-on/power-off reset (POR/PDR reset)
- Wake-up from Standby mode

3.2.2 System Reset

When a system reset occurs, all registers except the reset flag and the backup domain in the control / status register RCC_RSTSCKR are reset. The source of the reset event is identified by looking at the reset status flag bit in the RCC_RSTSCKR register.

The conditions for its production include:

- Low level signal on NRST pin (External reset)
- Window watchdog count terminated (WWDG reset)
- Independent watchdog count terminated (IWDG reset)
- Software reset (SW reset)
- Low-power management reset
- Core deadlock reset
- OPA reset
- USBPD reset
- ADC reset

Window/independent watchdog reset: Triggered by window/independent watchdog peripheral timer count cycle overflow, see the corresponding chapter for a detailed description.

Software reset: This product resets the system through the SYSRST position 1 of the interrupt configuration register PFIC_CFGR in the programmable interrupt controller PFIC or the RSTSYS position 1 of the configuration register PFIC_SCTLR, refer to the corresponding chapter.

Low-power management reset: By setting the STANDY_RST position 0 in the bytes selected by the user, the Standby mode reset will be enabled. After the process of entering the Standby mode is performed, the execution system is reset instead of entering the Standby mode. The Stop mode reset is enabled by setting the STOP_RST position 0 in the byte selected by the user. At this time, after the process of entering the Stop mode is performed, the system will be reset instead of entering the Stop mode.

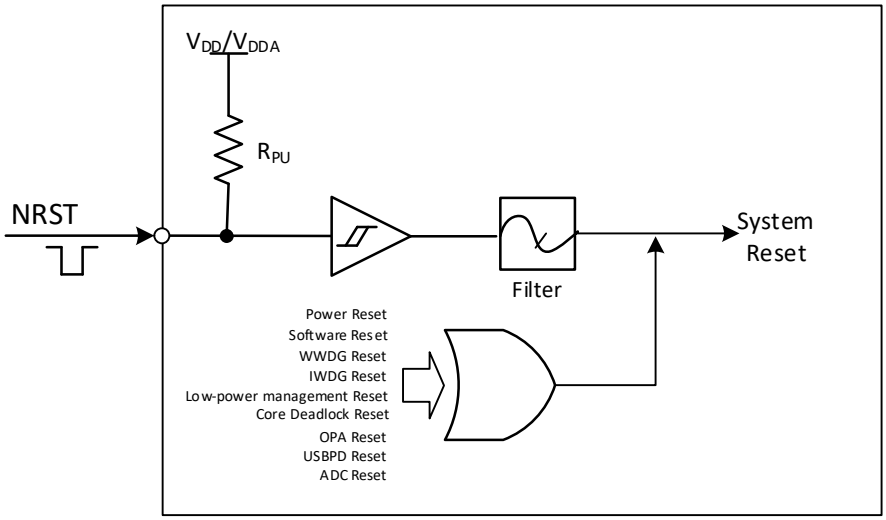
Core Deadlock Reset: when the LOCKUP of PFIC_SCTLR register is 0, the kernel deadlock is enabled, and the kernel will go into deadlock when it executes exceptions and NMI executes instructions. When the LKUPEN bit of the EXTEN_CTR register is enabled, the system will reset in case of a Lock-up condition.

OPA reset: When the OPA reset is enabled, the high level of the OPA output will cause an OPA reset.

USBPD Reset: When PD_RST_EN is 1, the CH32L103 supports the reset generated by the Hard Reset of the USBPD signal frame. If IE_RX_RESET is also 1, the Reset generated by Cable Reset of signal frames is also supported. USBPD does not have a reset flag, but the resulting reset effect is the same as a software reset.

ADC reset: With the ADC Watchdog reset enabled, ADC reset occurs when ADC data is greater than the watchdog high threshold or less than the watchdog low threshold.

Figure 3-1 System reset structure



3.2.3 Backup Domain Reset

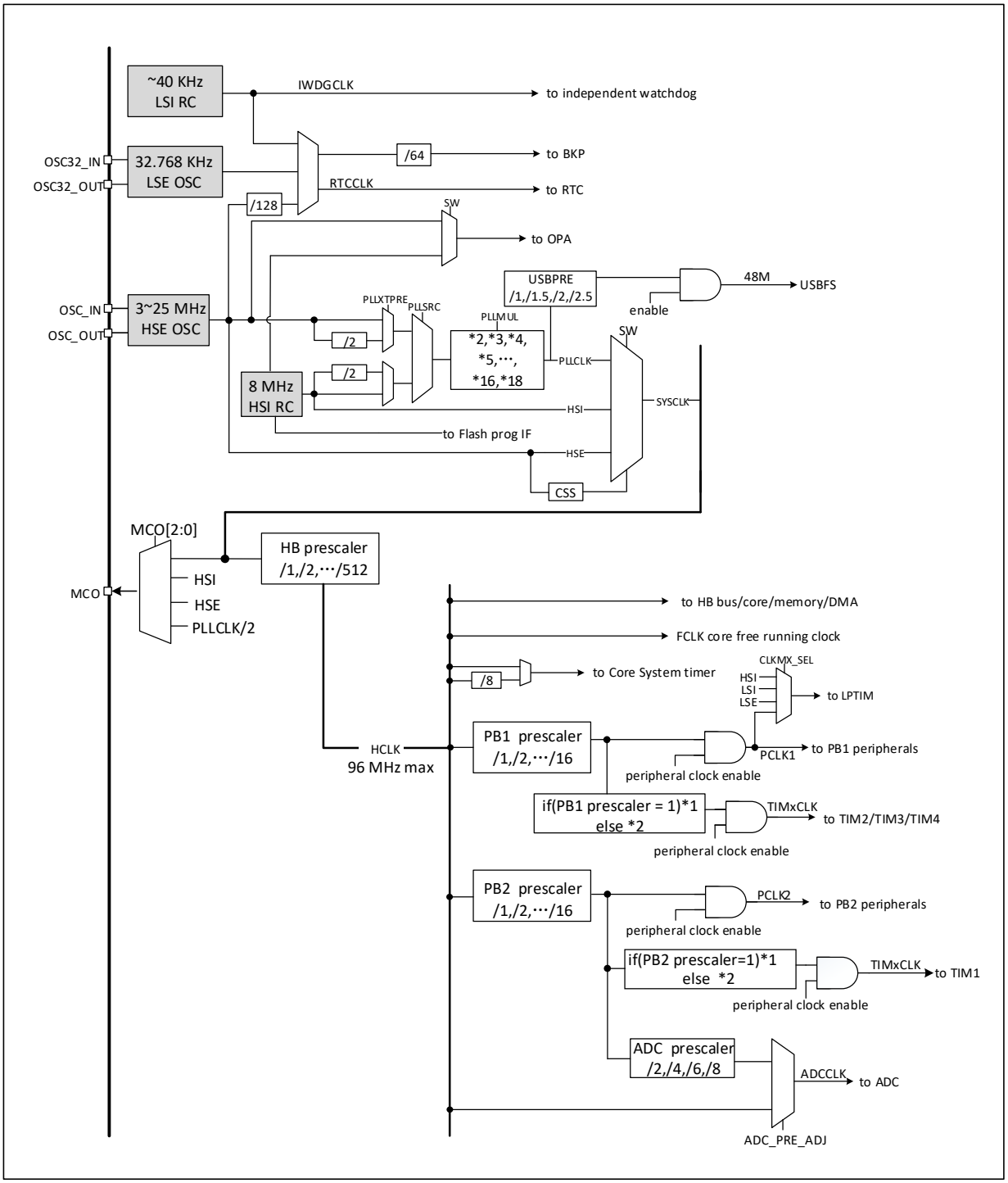
When a backup domain reset occurs, only the backup domain registers are reset, including the backup registers, the RCC_BDCTLR registers (RTC enable and LSE oscillator). The conditions for its generation include:

- Caused by V_{DD} or V_{BAT} power-up with both V_{DD} and V_{BAT} powered down
- Set BDRST to 1 of the RCC_BDCTLR register
- Set BKPRST to 1 of the RCC_PB1PRSTR register

3.3 Clock

3.3.1 System Clock Structure

Figure 3-2 Clock tree block diagram



Note: When using the USB function, the CPU frequency must be 48MHz, 72MHz or 96MHz. when the system wakes up from Stop or Standby mode, the system will automatically switch to HSI as the main frequency.

3.3.2 High-speed Clock (HIS/HSE)

HSI is a high-speed clock signal generated by the RC oscillator of 8MHz in the system. The HIS RC oscillator can

provide the system clock without any external devices. Its start-up time is very short, but the clock frequency accuracy is poor. The HSI is turned on and off by setting the HSION bit in the RCC_CTLR register, and the HSIRDY bit indicates whether the HSIRC oscillator is stable. The system defaults to HSION and HSIRDY setting 1 (it is recommended that you do not turn it off). If the HSIRDYIE bit of the RCC_INTR register is set, the corresponding interrupt will be generated.

- The HSI RC oscillator can enter HSILP position 1 into the internal low-power mode through the RCC_CTRL register.
- Factory calibration: differences in manufacturing processes will lead to different RC oscillation frequencies of each chip, so HSI calibration is performed for each chip before it leaves the factory. After the system is reset, the factory calibration value is loaded into the HSICAL [7:0] in the RCC_CTLR register.
- User adjustment: depending on the voltage or ambient temperature, the application can adjust the HSI frequency through the HSITRIM [4:0] bit in the RCC_CTLR register.

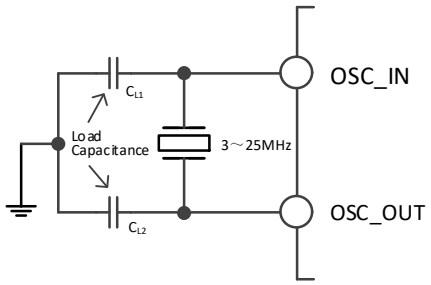
Note: If the HSE crystal oscillator fails, the HSI clock will be used as a backup clock source (clock security system).

HSE is an external high-speed clock signal, including external crystal/ceramic resonator generation or external high-speed clock input.

HSE crystal oscillator can enter HSELP position 1 into low-power mode through the RCC_CTRL register.

- External Crystal/Ceramic Resonator (HSE Crystal): An external 3-25MHz external oscillator provides a more accurate clock source for the system. Further information can be found in the Electrical Characteristics section of the datasheet. The HSE crystal can be turned on and off by setting the HSEON bit in the RCC_CTLR register. The HSERDY bit indicates whether the HSE crystal oscillation is stable or not, and the hardware sends the clock into the system only after the HSERDY bit is set to one. If the HSERDYIE bit in the RCC_INTR register is set, the appropriate interrupt will be generated.

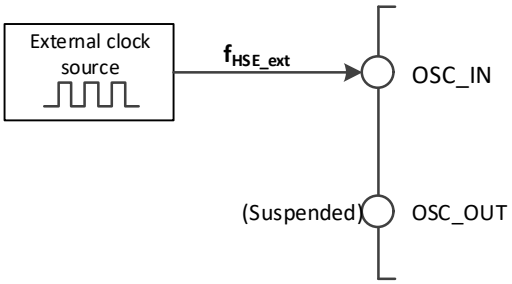
Figure 3-3 High-speed external crystal circuit



Note: The load capacitance needs to be as close as possible to the oscillator pins and the capacitance value selected according to the crystal manufacturer's parameters.

- External High Speed Clock Source (HSE Bypass): this mode feeds the clock source directly from the external source to the OSC_IN pin, with the OSC_OUT pin dangling. A maximum frequency of 25MHz is supported. The application program needs to set the HSEBYP bit to turn on the HSE bypass function with the HSEON bit at 0, and then set the HSEON bit again.

Figure 3-4 High-speed clock source circuit



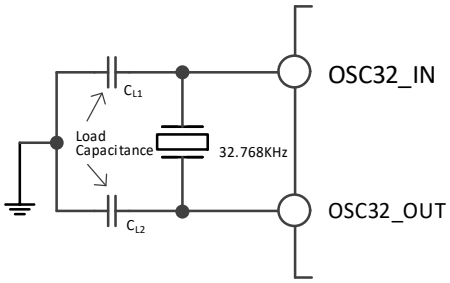
3.3.3 Low-speed Clock (LSI/LSE)

LSI is a low-speed clock signal generated by a RC oscillator of about 40KHz in the system. It can be kept running in both downtime and standby modes, providing clock references for RTC clocks, independent watchdogs, and wake-up units. For further information, please refer to the electrical characteristics section of the data manual. LSI can be turned on and off by setting the LSION bit in the RCC_RSTSCKR register, and then check whether the LSIRC oscillation is stable by querying the LSIRDY bit, and the hardware sends the clock in after LSIRDY position 1. If the LSIRDYIE bit of the RCC_INTR register is set, the corresponding interrupt will be generated.

LSE is an external low-speed clock signal, including external crystal/ceramic resonator generation or external low-speed clock input. It provides a low-power and accurate clock source for RTC clocks or other timing functions.

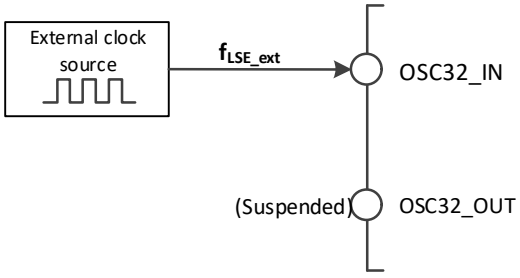
- External crystal/ceramic resonator (LSE crystal): External low-speed oscillator with external 32.768KHz. LSE is turned on and off by setting the LSEON bit in the RCC_BDCTLR register, and the LSERDY bit indicates whether the LSE crystal oscillation is stable or not, and the hardware sends the clock into the system after LSERDY position 1. If the LSERDYIE bit of the RCC_INTR register is set, the corresponding interrupt will be generated.

Figure 3-5 Low-speed external crystal circuit



- External low-speed clock source (LSE bypass): this mode feeds the clock source directly from the outside to the OSC32_IN pin, and the OSC32_OUT pin is suspended. The application needs to set the LSEBYP bit when the LSEON bit is 0, turn on the LSE bypass function, and then set the LSEON bit.

Figure 3-6 Low-speed clock source circuit



3.3.4 PLL Clock

By configuring the RCC_CFGR0 register and the expansion register EXTEN_CTR, the internal PLL clock can choose three clock sources and frequency doubling factors, which must be set before each PLL is turned on, and these parameters cannot be changed once the PLL is started. The PLLON bit in the set RCC_CTLR register is turned on and off, the PLLRDY bit indicates whether the PLL clock is stable, and the hardware feeds the clock into the system after PLLRDY position 1. If the PLLRDYIE bit of the RCC_INTR register is set, the corresponding interrupt will be generated.

PLL clock source:

- HSI clock feed
- HSI clock feed through 2-division frequency
- HSE clock feed
- HSE clock feed through 2-division frequency

3.3.5 Bus/Peripheral Clock

3.3.5.1 System Clock (SYSCLK)

By configuring the RCC_CFGR0 register SW[1:0] bit to configure the system clock source, SWS[1:0] indicates the current system clock source.

- HSI as the system clock.
- HSE as the system clock.
- PLL clock as the system clock.

After the controller is reset, the default HSI clock is selected as the system clock source. The switching between clock sources does not occur until the target clock source is ready.

3.3.5.2 HB/PB1/PB2 Bus Peripheral Clock (HCLK/PCLK1/PCLK2)

HB (High Performance Bus), whose bus peripheral clock is HCLK; PB1 (Peripheral Bus 1), whose bus peripheral clock is PCLK1; and PB2 (Peripheral Bus 2), whose bus peripheral clock is PCLK2.

By configuring the HPRE [3:0], PPRE1 [2:0] and PPRE2 [2:0] bits of the RCC_CFGR0 register, the clocks of the HB, PB1 and PB2 buses can be configured respectively. These bus clocks determine the peripheral interface mounted below them to access the clock reference. The application can adjust different values to reduce the power consumption of some peripherals.

Different peripheral modules can be reset and restored to the initial state through each bit in the RCC_HBRSTR, RCC_PB1PRSTR and RCC_PB2PRSTR registers.

The communication clock interface of different peripheral modules can be turned on or off separately through each bit in the RCC_HBPCENR, RCC_PB1PCENR and RCC_PB2PCENR registers. When using a peripheral, you first need to turn on its clock enable bit before you can access its register.

3.3.5.3 RTC Clock (RTCCLK)

By setting the RTCSEL[1:0] bit of the RCC_BDCTLR register, the RTCCLK clock source can be provided by the HSE/128, LSE, or LSI clock. Before modifying this bit, make sure that the DBP position 1 in the power control register (PWR_CTLR) can be reset only if the backup domain is reset.

- LSE acts as the RTC clock: Since LSE is powered by V_{BAT} in the backup domain, RTC continues to work as long as V_{BAT} maintains power, even though V_{DD} power is cut off.
- LSI as the RTC clock: If the V_{DD} power is cut off, RTC automatic wake up is not guaranteed.
- HSE/128 acts as the RTC clock: If the V_{DD} power supply is cut off or the internal voltage regulator is turned off (the power supply in the 1.8V domain is cut off), the RTC state is uncertain.

3.3.5.4 Independent Watchdog Clock

If the independent watchdog has been started by the hardware configuration or software, the LSI oscillator will be forced to open and cannot be turned off. After the LSI oscillator is stabilized, the clock is supplied to the IWDG.

3.3.5.5 Microcontroller Clock Output (MCO)

The microcontroller allows clock signals to be output to MCO pins. The alternate push-pull output mode is configured in the corresponding GPIO port register. by configuring the RCC_CFGR0 register MCO [2:0] bit, the following 4 clock signals can be selected as MCO clock output:

- System clock (SYSCLK) output.
- HSI clock output.
- HSE clock output.
- PLL clock output through 2 frequency division

3.3.5.6 USB Clock

The USB 48MHz clock source comes from the PLL clock through a configurable frequency divider, where PLL supports three clock configurations, including 48MHz, 72MHz and 96MHz, and outputs the 48MHz clock to USBFS through the USBPRE [1:0] bit of the configuration register RCC_CFGR0.

3.3.5.7 ADC Clock

ADC can select the input clock source and control the duty cycle of the ADC clock through the RCC_CFGR0 register.

The ADC_PRE_ADJ bit is used to select the input clock in clock configuration register 0, and when the position 1, the ADC input clock selects the HCLK clock input. When the position is 0, the ADC input clock selects the clock input of the PLCK2 after ADC frequency division (ADCPRE [1:0]), and the duty cycle of the ADC clock can be changed through the corresponding configuration of the ADC_DUTY_CHG bit.

3.3.6 Clock Security System

The clock security system is a running protection mechanism of the controller, which can switch to the HSI clock in the case of HSE clock transmission failure, and generate interrupt notification, allowing application software to complete the rescue operation.

Activate the clock security system by setting CSSON position 1 of the RCC_CTLR register. At this point, the clock monitor will be enabled after the HSE oscillator startup (HSERDY=1) delay and turned off after the HSE clock is turned off. Once the HSE clock fails during the operation of the system, the HSE oscillator will be turned off, the clock failure event will be sent to the brake input of the advanced timer (TIM1), and the clock security interrupt will be generated, CSSF position 1, and the application will enter the NMI unshielded interrupt. By setting the CSSC bit, the CSSF bit flag can be cleared and the NMI interrupt suspension bit can be revoked.

If the current HSE is the system clock, or the current HSE is the PLL input clock and the PLL is the system clock, the clock security system will automatically switch the system clock to the HSI oscillator and turn off the HSE oscillator and PLL in the event of a HSE failure.

3.4 Register Description

Table 3-1 RCC-related registers list

Name	Access address	Description	Reset value
R32_RCC_CTLR	0x40021000	Clock control register	0x0000xx83
R32_RCC_CFGR0	0x40021004	Clock configuration register 0	0x00000000
R32_RCC_INTR	0x40021008	Clock interrupt register	0x00000000
R32_RCC_PB2PRSTR	0x4002100C	PB2 peripheral reset register	0x00000000
R32_RCC_PB1PRSTR	0x40021010	PB1 peripheral reset register	0x00000000
R32_RCC_HBPCENR	0x40021014	HB peripheral clock enable register	0x00000014
R32_RCC_PB2PCENR	0x40021018	PB2 peripheral clock enable register	0x00000000
R32_RCC_PB1PCENR	0x4002101C	PB1 peripheral clock enable register	0x00000000
R32_RCC_BDCTLR	0x40021020	Backup domain control register	0x00000000
R32_RCC_RSTSCKR	0x40021024	Control/status register	0x0C000000
R32_RCC_HBRSTR	0x40021028	HB peripheral reset register	0x00000000

3.4.1 Clock Control Register (RCC_CTLR)

Offset address: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						PLL RDY	PLL ON	Reserved			HSELP	CSSON	HSE BYP	HSE RDY	HSEON

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

Bit	Name	Access	Description	Reset value
[31:26]	Reserved	RO	Reserved	0
25	PLLRDY	RO	PLL clock ready flag (set by hardware): 1: PLL locked; 0: PLL unlocked	0
24	PLLON	RW	PLL enable: 1: Enable PLL clock; 0: Disable PLL clock. *Note: Cleared by hardware when entering Stop or Standby mode.*	0
[23:21]	Reserved	RO	Reserved	0
20	HSELP	RW	HSE low-power mode: 1: On; 0: Off.	1

19	CSSON	RW	Clock security system enable: 1: Enable clock security system. When HSE is ready (HSERDY is set to 1), the hardware turns on the clock monitoring function for HSE and finds that HSE abnormal triggers the CSSF flag and NMI interrupt; when HSE is not ready, the hardware turns off the clock monitoring function for HSE. 0: Disable clock security system.	0
18	HSEBYP	RW	External high-speed clock bypass: 1: Bypass the external high-speed crystal/ceramic resonator (if the HSEON bit is 0, set the HSEBYP bit to turn on the HSE bypass function, and then set the HSEON bit again to take effect); 0: No bypass high-speed external crystal/ceramic resonator.	0
17	HSERDY	RO	External high-speed crystal oscillation stability ready flag (set by hardware): 1: External high-speed crystal oscillation is stable. 0: External high-speed crystal oscillation is not stable. *Note: After the HSEON bit is cleared, it needs 6 HSE cycles to clear 0.*	0
16	HSEON	RW	HSE clock enable: 1: Enable HSE oscillator; 0: Disable HSE oscillator. *Note: Clear by hardware to stop HSE oscillator when entering Stop or Standby mode.*	0
[15:8]	HSICAL[7:0]	RO	Internal high-speed clock calibration value These bits are automatically initialized at system startup.	xxh
[7:3]	HSITRIM[4:0]	RW	Internal high-speed clock adjustment: The user can enter an adjustment value that is superimposed on the HSICAL [7:0] value to adjust the frequency of the internal HSIRC oscillator according to changes in voltage and temperature. The default value is 16, which can adjust the HSI to 8MHz ±1%; the change of HSICAL is adjusted about 20KHz per step when LP=0. The change of HSICAL is adjusted about 2.5KHz per step when LP=1.	10000b
2	HSILP	RW	HIS internal low-power mode: 1: On, the HSI frequency is reduced to 1MHz. 0: Off *Note: The HSI enters the internal low-power mode by loading the value of the 0x1FFFF72A address into HSITRIM[4:0] to realize the correction in the low-power mode*	0

1	HSIRDY	RO	Internal high-speed HSI stable ready flag (set by hardware) 1: Internal high-speed HSI is stable; 0 Internal high-speed HSI is not stable. *Note: After the HSION bit is cleared, it needs 6 HSI cycles to clear 0.*	1
0	HSION	RW	Internal high-speed clock HSI enable control bit: 1: Enable the HSI oscillator; 0: Disable the HSI oscillator. *Note: This bit is set by hardware to 1 to start the internal HSI oscillator when returning from standby and stop mode or when the external oscillator HSE used as the system clock fails.*	1

3.4.2 Clock Configuration Register 0 (RCC_CFGR0)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADC_PRE_ADJ	ADC_DUTY_CHG			Reserved	MCO[2:0]			USBPRE[1:0]		PLLMUL[3:0]				PLLXTPRE	PLLSRC

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCPRE[1:0]		PPRE2[2:0]			PPRE1[2:0]			HPRE[3:0]				SWS[1:0]		SW[1:0]	

Bit	Name	Access	Description	Reset value
31	ADC_PRE_ADJ	RW	ADC input clock selection: 1: Select the HCLK clock input; 0: Select the PLCK2 clock input after ADC division (ADCPRE[1:0]).	0
[30:28]	ADC_DUTY_CHG	RW	ADC clock duty cycle control bit: 000: High level duration is half of ADC clock cycle. 001: High level duration adds 1 HCLK clock cycle. 010: High level duration adds 2 HCLK clock cycles. 011: High level duration adds 3 HCLK clock cycles. 100: High level duration adds 4 HCLK clock cycles. 101: High level duration adds 5 HCLK clock cycles. 110: High level duration adds 6 HCLK clock cycles. 111: High level duration adds 7 HCLK clock cycles	000
27	Reserved	RO	Reserved	0
[26:24]	MCO[2:0]	RW	Microcontroller MCO pin clock output control: 0xx: No clock output. 100: System clock (SYSCLK) output. 101: Internal oscillator clock (HSI) output. 110: External oscillator clock (HSE) output.	000b

			111: PLL clock output after 2 frequency division	
[23:22]	USBPRE[1:0]	RW	USB clock division frequency configuration: 00: 1-division frequency (for PLLCLK=48MHz); 01: 2-division frequency (for PLLCLK=96MHz); 10: 1.5-division frequency (for PLLCLK=72MHz); 11: Reserved.	00b
[21:18]	PLLMUL[3:0]	RW	PLL clock multiplication factor (not writable until PLL is off): 0000: PLL 2x output; 0001: PLL 3x output; 0010: PLL 4x output; 0011: PLL 5x output; 0100: PLL 6x output; 0101: PLL 7x output; 0110: PLL 8x output; 0111: PLL 9x output; 1000: PLL 10x output; 1001: PLL 11x output; 1010: PLL 12x output; 1011: PLL 13x output; 1100: PLL 14x output; 1101: PLL 15x output; 1110: PLL 16x output; 1111: PLL 18x output.	0000b
17	PLLXTPRE	RW	HSE split frequency feed to PLL control (write only when PLL is off): 1: HSE 2-division frequency feed to PLL; 0: HSE no-division feed to PLL.	0
16	PLLSRC	RW	Input clock source to the PLL (not writable until the PLL is turned off): 1: HSE undivided or 2-division fed to the PLL; 0: HSI undivided or 2-division fed to the PLL. *Note: The division of the HSI clock is controlled using the EXTEN_CTR register HSIPRE bit.*	0
[15:14]	ADCPRE[1:0]	RW	ADC clock source pre-divided frequency control: 00: PCLK2 2-division frequency as ADC clock; 01: PCLK2 4-division frequency as ADC clock; 10: PCLK2 6-division frequency as ADC clock; 11: PCLK2 8-division frequency as ADC clock. *Note: The ADC clock should not exceed 48MHz maximum.*	00b
[13:11]	PPRE2[2:0]	RW	PB2 clock source pre-divided control: 0xx: HCLK no frequency division; 100: HCLK 2-division frequency; 101: HCLK 4-division frequency; 110: HCLK 8-division frequency; 111: HCLK 16-division frequency.	000b
[10:8]	PPRE1[2:0]	RW	PB1 clock source pre-divided control: 0xx: HCLK no frequency division; 100: HCLK 2-division frequency; 101: HCLK 4-division frequency; 110: HCLK 8-division frequency; 111: HCLK 16-division frequency.	000b

[7:4]	HPRE[3:0]	RW	HB clock source pre-divided control: 0xxx: SYSCLK no frequency division; 1000: SYSCLK 2-division frequency; 1001: SYSCLK 4-division frequency; 1010: SYSCLK 8-division frequency; 1011: SYSCLK 16-division frequency; 1100: SYSCLK 64-division frequency; 1101: SYSCLK 128-division frequency; 1110: SYSCLK 256-division frequency; 1111: SYSCLK 512-division frequency.	0000b
[3:2]	SWS[1:0]	RO	System clock (SYSCLK) status (set by hardware): 00: System clock source is HSI; 01: System clock source is HSE; 10: System clock source is PLL; 11: Not available.	00b
[1:0]	SW[1:0]	RW	Select the system clock source: 00: HSI as system clock; 01: HSE as system clock; 10: PLL output as system clock; 11: not available. *Note: With Clock Safe System enabled (CSSON=1), the selection of HSI as the system clock is forced by hardware when returning from Standby and Stop mode or when the external oscillator HSE used as the system clock fails.*	00b

3.4.3 Clock Interrupt Register (RCC_INTR)

Offset address: 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CSSC	Reserved		PLL RDYC	HSE RDYC	HSI RDYC	LSE RDYC	LSI RDYC

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			PLL RDYIE	HSE RDYIE	HSI RDYIE	LSE RDYIE	LSI RDYIE	CSSF	Reserved		PLL RDYF	HSE RDYF	HSI RDYF	LSE RDYF	LSI RDYF

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RO	Reserved	0
23	CSSC	WO	Clear clock security system interrupt flag (CSSF): 1: Clear CSSF interrupt flag; 0: No effect.	0
[22:21]	Reserved	RO	Reserved	0
20	PLLRDYC	WO	Clear PLL ready interrupt flag: 1: Clear PLLRDYF interrupt flag;	0

			0: No effect.	
19	HSERDYC	WO	Clear HSE oscillator ready interrupt flag: 1: Clear HSERDYF interrupt flag; 0: No effect.	0
18	HSIRDYC	WO	Clear HSI oscillator ready interrupt flag: 1: Clear HSIRDYF interrupt flag; 0: No effect.	0
17	LSERDYC	WO	Clear LSE oscillator ready interrupt flag: 1: Clear LSERDYF interrupt flag; 0: No effect.	0
16	LSIRDYC	WO	Clear LSI oscillator ready interrupt flag: 1: Clear LSIRDYF interrupt flag; 0: No effect.	0
[15:13]	Reserved	RO	Reserved	0
12	PLLRDYIE	RW	PLL ready interrupt enable: 1: Enable PLL ready interrupt; 0: Disable PLL ready interrupt.	0
11	HSERDYIE	RW	HSE ready interrupt enable: 1: Enable HSE ready interrupt; 0: Disable HSE ready interrupt.	0
10	HSIRDYIE	RW	HSI ready interrupt enable: 1: Enable HSI ready interrupt; 0: Disable HSI ready interrupt.	0
9	LSERDYIE	RW	LSE ready interrupt enable: 1: Enable LSE ready interrupt; 0: Disable LSE ready interrupt.	0
8	LSIRDYIE	RW	LSI ready interrupt enable: 1: Enable LSI ready interrupt; 0: Disable LSI ready interrupt.	0
7	CSSF	RO	Clock security system interrupt flag bit: 1: HSE clock failure, resulting in clock security interrupt CSSI. 0: no clock security system interrupt. Set by hardware, clear when software write CSSC bit 1.	0
[6:5]	Reserved	RO	Reserved	0
4	PLLRDYF	RO	PLL clock ready lock interrupt flag: 1: PLL clock lock generates an interrupt; 0: No PLL clock lock interrupt. Set by hardware, clear when software write PLLRDYC bit 1.	0
3	HSERDYF	RO	HSE clock ready interrupt flag: 1: HSE clock ready generates an interrupt; 0: No HSE clock ready interrupt. Set by hardware, clear when software write HSERDYC	0

			bit 1.	
2	HSIRDYF	RO	HSI clock ready interrupt flag: 1: HSI clock ready generates an interrupt; 0: No HSI clock ready interrupt. Set by hardware, clear when software write HSIRDYC bit 1.	0
1	LSERDYF	RO	LSE clock ready interrupt flag: 1: LSE clock ready generates an interrupt; 0: No LSE clock ready interrupt. Set by hardware, clear when software write LSERDYC bit 1.	0
0	LSIRDYF	RO	LSI clock ready interrupt flag: 1: LSI clock ready generates an interrupt; 0: No LSI clock ready interrupt. Set by hardware, clear when software write LSIRDYC bit 1.	0

3.4.4 PB2 Peripheral Reset Register (RCC_PB2PRSTR)

Offset address: 0x0C

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	USART1 RST	Reserved	SPI1 RST	TIM1 RST	Reserved	ADC RST	Reserved			IOPD RST	IOPC RST	IOPB RST	IOPA RST	Reserved	AFIO RST

Bit	Name	Access	Description	Reset value
[31:15]	Reserved	RO	Reserved	0
14	USART1RST	RW	USART1 interface reset control: 1: Reset module; 0: No effect.	0
13	Reserved	RO	Reserved	0
12	SPI1RST	RW	SPI1 interface reset control: 1: Reset module; 0: No effect.	0
11	TIM1RST	RW	TIM1 module reset control: 1: Reset module; 0: No effect.	0
10	Reserved	RO	Reserved。	0
9	ADCRST	RW	ADC module reset control: 1: Reset module; 0: No effect.	0
[8:6]	Reserved	RO	Reserved	0
5	IOPDRST	RW	IO's PD port module reset control: 1: Reset module; 0: No effect.	0
4	IOPCRST	RW	IO's PC port module reset control: 1: Reset module; 0: No effect.	0

3	IOPBRST	RW	IO's PB port module reset control: 1: Reset module; 0: No effect.	0
2	IOPARST	RW	IO's PA port module reset control: 1: Reset module; 0: No effect.	0
1	Reserved	RO	Reserved	0
0	AFIORST	RW	I/O auxiliary function module reset control: 1: Reset module; 0: No effect.	0

3.4.5 PB1 Peripheral Reset Register (RCC_PB2PRSTR)

Offset address: 0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPTIMRST	Reserved		PWRRST	BKPRST	Reserved	CANRST	Reserved		I2C2RST	I2C1RST	Reserved	USART4RST	USART3RST	USART2RST	Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SPI2RST	Reserved		WWDGRST	Reserved								TIM4RST	TIM3RST	TIM2RST

Bit	Name	Access	Description	Reset value
31	LPTIMRST	RW	LPTIM module reset control: 1: Reset module; 0: No effect.	0
[30:29]	Reserved	RO	Reserved	0
28	PWRRST	RW	Power interface module reset control: 1: Reset module; 0: No effect.	0
27	BKPRST	RW	Backup unit reset control: 1: Reset module; 0: No effect.	0
26	Reserved	RO	Reserved	0
25	CANRST	RW	CAN module reset control: 1: Reset module; 0: No effect.	0
[24:23]	Reserved	RO	Reserved	0
22	I2C2RST	RW	I2C 2 interface reset control: 1: Reset module; 0: No effect.	0
21	I2C1RST	RW	I2C 1 interface reset control: 1: Reset module; 0: No effect.	0
20	Reserved	RO	Reserved	0
19	USART4RST	RW	USART4 interface reset control: 1: Reset module; 0: No effect.	0
18	USART3RST	RW	USART3 interface reset control: 1: Reset module; 0: No effect.	0
17	USART2RST	RW	USART2 interface reset control: 1: Reset module; 0: No effect.	0
[16:15]	Reserved	RO	Reserved	0
14	SPI2RST	RW	SPI2 interface reset control:	0

			1: Reset module; 0: No effect.	
[13:12]	Reserved	RO	Reserved	0
11	WWDGRST	RW	WWDG reset control: 1: Reset module; 0: No effect.	0
[10:3]	Reserved	RO	Reserved	0
2	TIM4RST	RW	Timer 4 module reset control: 1: Reset module; 0: No effect.	0
1	TIM3RST	RW	Timer 3 module reset control: 1: Reset module; 0: No effect.	0
0	TIM2RST	RW	Timer 2 module reset control: 1: Reset module; 0: No effect.	0

3.4.6 HB Peripheral Clock Enable Register (RCC_HBPCENR)

Offset address: 0x14

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			USBFSEN	Reserved					CRCEN	Reserved			SRAMEN	Reserved	DMAEN

Bit	Name	Access	Description	Reset value
[31:18]	Reserved	RO	Reserved	0
17	USBPDEN	RW	USBPD clock enable: 1: USBPD clock on; 0: USBPD clock off.	1
[16:13]	Reserved	RO	Reserved	0
12	USBFSEN	RW	USBFS module clock enable: 1: Module clock on; 0: Module clock off.	0
[11:7]	Reserved	RO	Reserved	0
6	CRCEN	RW	CRC module clock enable: 1: Module clock on; 0: Module clock off.	0
[5:3]	Reserved	RO	Reserved	0
2	SRAMEN	RW	SRAM interface module clock enable: 1: In sleep mode, SRAM interface module clock on; 0: In sleep mode, SRAM interface module clock off.	1
1	Reserved	RO	Reserved	0
0	DMAEN	RW	DMA module clock enable: 1: Module clock on; 0: Module clock off.	0

3.4.7 PB2 Peripheral Clock Enable Register (RCC_PB2PCENR)

Offset address: 0x18

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	USART1EN	Reserved	SPI1EN	TIM1EN	Reserved	ADCEN	Reserved			IOPDEN	IOPCEN	IOPBEN	IOPAEN	Reserved	AFIOEN

Bit	Name	Access	Description	Reset value
[31:15]	Reserved	RO	Reserved	0
14	USART1EN	RW	USART1 interface clock enable: 1: Module clock on; 0: Module clock off.	0
13	Reserved	RO	Reserved	0
12	SPI1EN	RW	SPI1 interface clock enable: 1: Module clock on; 0: Module clock off.	0
11	TIM1EN	RW	TIM1 interface clock enable: 1: Module clock on; 0: Module clock off.	0
10	Reserved	RO	Reserved	0
9	ADCEN	RW	ADC interface clock enable: 1: Module clock on; 0: Module clock off.	0
[8:6]	Reserved	RO	Reserved	0
5	IOPDEN	RW	IO'PD port module clock enable: 1: Module clock on; 0: Module clock off.	0
4	IOPCEN	RW	IO'PC port module clock enable: 1: Module clock on; 0: Module clock off.	0
3	IOPBEN	RW	IO'PB port module clock enable: 1: Module clock on; 0: Module clock off.	0
2	IOPAEN	RW	IO'PA port module clock enable: 1: Module clock on; 0: Module clock off.	0
1	Reserved	RO	Reserved	0
0	AFIOEN	RW	IO auxiliary function module clock enable: 1: Module clock on; 0: Module clock off.	0

3.4.8 PB1 Peripheral Clock Enable Register (RCC_PB1PCENR)

Offset address: 0x1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPTIMEN	Reserved		PWREN	BKPEN	Reserved	CANEN	Reserved		I2C2EN	I2C1EN	Reserved	USART4EN	USART3EN	USART2EN	Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SPI2EN	Reserved		WWDGEN	Reserved								TIM4EN	TIM3EN	TIM2EN

Bit	Name	Access	Description	Reset value
31	LPTIMEN	RW	LPTIM module clock enable: 1: Module clock on; 0: Module clock off.	0
[31:29]	Reserved	RO	Reserved	0
28	PWREN	RW	Power interface module clock enable: 1: Module clock on; 0: Module clock off.	0
27	BKPEN	RW	Backup unit clock enable: 1: Module clock on; 0: Module clock off.	0
26	Reserved	RW	Reserved	0
25	CANEN	RW	CAN module clock enable: 1: Module clock on; 0: Module clock off.	0
[24:23]	Reserved	RO	Reserved	0
22	I2C2EN	RW	I2C 2 interface clock enable: 1: Module clock on; 0: Module clock off.	0
21	I2C1EN	RW	I2C 1 interface clock enable: 1: Module clock on; 0: Module clock off.	0
20	Reserved	RW	Reserved	0
19	USART4EN	RW	USART4 interface clock enable: 1: Module clock on; 0: Module clock off.	0
18	USART3EN	RW	USART3 interface clock enable: 1: Module clock on; 0: Module clock off.	0
17	USART2EN	RW	USART2 interface clock enable: 1: Module clock on; 0: Module clock off.	0
[16:15]	Reserved	RO	Reserved	0
14	SPI2EN	RW	SPI2 interface clock enable: 1: Module clock on; 0: Module clock off.	0
[13:12]	Reserved	RO	Reserved	0
11	WWDGEN	RW	WWDG clock enable: 1: Module clock on; 0: Module clock off.	0
[10:3]	Reserved	RO	Reserved	0
2	TIM4EN	RW	Timer 4 module clock enable: 1: Module clock on; 0: Module clock off.	0
1	TIM3EN	RW	Timer 3 module clock enable: 1: Module clock on; 0: Module clock off.	0
0	TIM2EN	RW	Timer 2 module clock enable: 1: Module clock on; 0: Module clock off.	0

3.4.9 Backup Domain Control Register (RCC_BDCTLR)

Offset address: 0x20

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTCEN	Reserved					RTCSEL[1:0]		Reserved					LSEBYP	LSERDY	LSEON

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	BDRST	RW	Backup domain software reset control: 1: Reset the entire backup domain. 0: Undo the reset.	0
15	RTCEN	RO	RTC clock enable control: 1: Enable the RTC clock; 0: Disable RTC clock. *Note: RTC clock can be enabled only under the condition of RTCSEL!=0, otherwise hardware forces 0.*	0
[14:10]	Reserved	RO	Reserved	0
[9:8]	RTCSEL[1:0]	RO	RTC clock source selection: 00: no clock; 01: LSE oscillator as RTC clock; 10: LSI oscillator as RTC clock; 11: HSE oscillator divided by 128 as RTC clock. *Note: Once the RTC clock source is selected (RTCEN=1), it cannot be changed again until the next time the backup domain is reset. The default can be restored by setting the BDRST bit.*	0
[7:3]	Reserved	RO	Reserved	0
2	LSEBYP	RO	External Low-speed crystal (LSE) bypass control bit: 1: Bypass external low-speed crystal/ceramic resonator (with LSEON bit 0, set the LSEBYP bit to turn on the HSE bypass function, then set the LSEON bit again to take effect); 0: Do not bypass the low-speed external crystal/ceramic resonator.	0
1	LSERDY	RO	External low-speed crystal oscillation stabilization ready flag bit (set by hardware): 1: External low-speed crystal oscillation is stabilized; 0: External low-speed crystal oscillation is not stabilized. *Note: After the LSEON bit is cleared to 0, this bit takes 6 LSE cycles to clear to 0.*	0
0	LSEON	RO	External low-speed crystal oscillation enable control bit:	0

			1: Enable the LSE oscillator; 0: Disable the LSE oscillator.	

3.4.10 Control/Status Register (RCC_RSTSCKR)

Offset address: 0x24

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPWR RSTF	WWDG RSTF	IWDG RSTF	SFT RSTF	POR RSTF	PIN RSTF	Reserved	RMVF	Reserved							

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

Bit	Name	Access	Description	Reset value
31	LPWRRSTF	RO	Low-power reset flag: 1: Occurrence of low-power resets. 0: No low-power reset occurs. Set to 1 by hardware when a low-power management reset occurs; cleared by software writing of the RMVF bit.	0
30	WWDGRSTF	RO	Window watchdog reset flag: 1: Occurrence of a window watchdog reset. 0: No window watchdog reset occurs. Set to 1 by hardware when a window watchdog reset occurs; cleared by software writing of the RMVF bit.	0
29	IWDGRSTF	RO	Independent watchdog reset flag: 1: Occurrence of an independent watchdog reset. 0: No independent watchdog reset occurs. Set to 1 by hardware when an independent watchdog reset occurs; cleared by software writing of the RMVF bit.	0
28	SFTRSTF	RO	Software reset flag: 1: Software reset occurs. 0: No software reset occurs. Set to 1 by hardware when a software reset occurs; software write RMVF bit cleared.	0
27	PORRSTF	RO	Power-up/power-down reset flag: 1: Power-up/power-down reset occurs. 0: No power-up/power-down reset occurs. Set to 1 by hardware when power-up/power-down reset occurs; cleared by software writing of RMVF bit.	1
26	PINRSTF	RO	External manual reset (NST pin) flag: 1: Occurrence of NST pin reset. 0: No NST pin reset occurs. Set to 1 by hardware when NST pin reset occurs; cleared by software writing of RMVF bit.	0

25	Reserved	RO	Reserved	0
24	RMVF	WO	Clear reset flag control: 1: Clear the reset flag. 0: No effect.	0
[23:2]	Reserved	RO	Reserved	0
1	LSIRDY	RO	Internal low-speed clock (LSI) stabilization ready flag bit (set by hardware): 1: The internal low-speed clock (40KHz) is stabilized; 0: The internal low-speed clock (40KHz) is not stabilized. *Note: After the LSION bit is cleared to 0, the bit takes 3 LSI cycles to clear to 0.*	0
0	LSION	RW	Internal low-speed clock (LSI) enable control bits: 1: Enables the LSI (40KHz) oscillator; 0: Disables the LSI (40KHz) oscillator.	0

Note: Except for the reset flag, which can only be cleared by a power-on reset, it is cleared by a system reset.

3.4.11 HB Peripheral Reset Register (RCC_HBRSTR)

Offset address: 0x28

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

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

Bit	Name	Access	Description	Reset value
[31:18]	Reserved	RO	Reserved	0
17	USBPDRST	RW	USBPD reset control: 1: Reset module; 0: No effect.	0
[16:13]	Reserved	RO	Reserved	0
12	USBFSRST	RW	USBFS module reset control: 1: Reset module; 0: No effect.	0
[11:0]	Reserved	RO	Reserved	0

Chapter 4 Backup Register (BKP)

Backup register (BKP) provides 10 16-bit backup data registers that can be used to store 20 bytes of user data. After the main power supply (V_{DD}) is powered off, these data can still be maintained by V_{BAT} power supply, regardless of standby state, system reset or power reset. In addition, the BKP unit also provides tamper detection management, RTC clock calibration and pulse output functions.

4.1 Main Features

- Provide a 20-byte backup data register
- RTC clock calibration function
- Output RTC clock 64-division frequency, alarm clock pulse or second pulse on PC13 pin
- Tamper detection (TAMPER) function

4.2 Function Description

Access to the backup register and RTC is disabled after the microcontroller is reset, and the access to the backup register needs to be enabled by the following operations:

1) Set the PWREN bit and BKPEN bit of the register RCC_PB1PCENR to turn on the operating clock of the power supply and backup interface.
2) Set the DBP bit of the power control register PWR_CTLR to enable access to the backup register and the RTC register.

4.2.1 Backup Data Register

The backup data register can be used as a general data cache. Because of its characteristic of saving data by V_{BAT} power supply under V_{DD} power-off, it can be used to store some important or sensitive data. However, all of this data will be erased after the tamper.

4.2.2 Tamper Detection

Tamper detection means that when a signal (rising edge or falling edge) is provided by the outside world, it indicates that there is an "Tamper event", and the hardware will automatically clear the important information retained in the current system. This way can increase the security of system information.

When a jump edge appears on the tamper detection pin (depending on the TPAL bit), an tamper event will occur, and if the tamper detection is enabled, an tamper detection interruption will occur at the same time. As soon as there is an tamper, the backup data registers will be cleared. In addition, the hardware detection adopts memory mode, and even if the tamper detection function is not turned on (TPE=0), the system will sample whether there is a jump edge, and if the TPAL bit selection is satisfied, the tamper event will be locked in advance, and the tamper event will be triggered under TPE position 1.

For example: when TPAL=0, if the TPE=0 function is not turned on, but the TAMPER pin is already high, once TPE=1, there will be an additional tamper event (the system locks the rising edge in advance). When TPAL=1, if the TPE=0 function is not turned on, but the TAMPER pin is already low, once TPE=1, there will be an additional tamper event (the system locks the falling edge in advance).

Therefore, in order to prevent unnecessary tamper events, resulting in the removal of the backup register, it is recommended that when you want the hardware to detect the tamper pin, you should first clear the tamper event that the hardware may remember by writing the BKP_TPCSR register CTE position 1, and ensure that the current tamper detection pin state is invalid.

Note: when the V_{DD} power is disconnected, the tamper detection function is still valid. To avoid unnecessary reset of the data backup register, the TAMPER pin should be connected to the correct level off the chip.

4.2.3 RTC Calibration

This function must be configured to use the tamper detection pin as a normal IO port. Configure the BKP_TPCTLR register TPE bit to clear 0.

- Pulse Output

Configure the ASOE bit of the BKP_OCTLR register to turn on the RTC pulse output and set the ASOS bit to select whether the seconds pulse output or the alarm pulse output.

- RTC Calibration

After configuring the CCO bit of the BKP_OCTLR register, the internal RTC clock will be output to the trespass detection pin (TAMPER) after a 64-division frequency. The RTC is calibrated by the software in conjunction with modifying the CAL[6:0] bits to adjust the clock through actual testing.

4.2.4 BKP Interface Reset

The BKP region can be independently powered by V_{BAT} with the VDD main power supply down. In the application code control BKP area register reset, the backup data registers BKP_DATAR1-10, the ASOS bit, and the ASOE bit are reset under the BDRST bit of the software-configured RCC_BDCTLR register, and are not affected by the RCC peripheral interface control BKPRST bit.

4.3 Register Description

Table 4-1 BKP-related registers list

Name	Access address	Description	Reset value
R16_BKP_DATAR1	0x40006C04	Backup data register 1	0x0000
R16_BKP_DATAR2	0x40006C08	Backup data register 2	0x0000
R16_BKP_DATAR3	0x40006C0C	Backup data register 3	0x0000
R16_BKP_DATAR4	0x40006C10	Backup data register 4	0x0000
R16_BKP_DATAR5	0x40006C14	Backup data register 5	0x0000
R16_BKP_DATAR6	0x40006C18	Backup data register 6	0x0000
R16_BKP_DATAR7	0x40006C1C	Backup data register 7	0x0000
R16_BKP_DATAR8	0x40006C20	Backup data register 8	0x0000
R16_BKP_DATAR9	0x40006C24	Backup data register 9	0x0000
R16_BKP_DATAR10	0x40006C28	Backup data register 10	0x0000
R16_BKP_OCTLR	0x40006C2C	RTC calibration register	0x0000
R16_BKP_TPCTLR	0x40006C30	Tamper detection control register	0x0000
R16_BKP_TPCSR	0x40006C34	Tamper detection status register	0x0000

4.3.1 Backup Data Register (BKP_DATARx) (x=1-10)

Offset address: 0x04-0x28

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

Bit	Name	Access	Description	Reset value
[15:0]	D[15:0]	RW	*Backup data, can be called by the user program.* *Note: They are reset only by a Backup Domain Reset (BDRST) or (if the Tamper Detection Pin TAMPER function is enabled) by an Tamper Pin Event.*	0

4.3.2 RTC Calibration Register (BKP_OCTLR)

Offset address: 0x2C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						ASOS	ASOE	CCO	CAL[6:0]						

Bit	Name	Access	Description	Reset value
[15:10]	Reserved	RO	Reserved	0
9	ASOS	RW	TAMPER pin alarm/seconds pulse output selection. 1: Output seconds pulse; 0: Output alarm pulse. *Note: This bit will only be reset by the Backup Domain Reset (BDRST).*	0
8	ASOE	RW	TAMPER pin enable pulse output bit 0: Disable the output of alarm pulse or second pulse; 1: Enable the output of alarm pulses or seconds pulses. *Note: This bit will only be reset by the Backup Domain Reset (BDRST).*	0
7	CCO	RW	Calibration clock output select bit 1: The TEMPER pin outputs the RTC clock divided by 64; 0: No calibration clock is output. *Note 1: Turning on this function must disable the tamper detection function.* *Note 2: This bit is cleared when the VDD supply is disconnected.*	0
[6:0]	CAL[6:0]	RW	Calibration Value Register, the value of this register indicates how many of every 220 clock pulses are skipped. This function is used to	0

			calibrate the RTC clock, which can be slowed down from 0 to 121 ppm.	

4.3.3 Tamper Detection Control Register (BKP_TPCTLR)

Offset address: 0x30

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

Bit	Name	Access	Description	Reset value
[15:2]	Reserved	RO	Reserved	0
1	TPAL	RW	Tamper detection pin (TEMPER pin) active level set: 0: High level on the tamper detection pin cause all backup data registers cleared (hardware lock rising edge); 1: Low level on the tamper detection pin causer all backup data registers cleared (hardware lock falling edge);	0
0	TPE	RW	Tamper detection pin enable bit 0: TEMPER pin used as common IO port; 1: TEMPER pin used for the tamper detection.	0

Note: When the TPAL and TPE bits are cleared at the same time, a false tamper event occurs. It is recommended to change the status of the TPAL bit only when TPE is 0.

4.3.4 Tamper Detection Status Register (BKP_TPCSR)

Offset address: 0x34

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						TIF	TEF	Reserved					TPIE	CTI	CTE

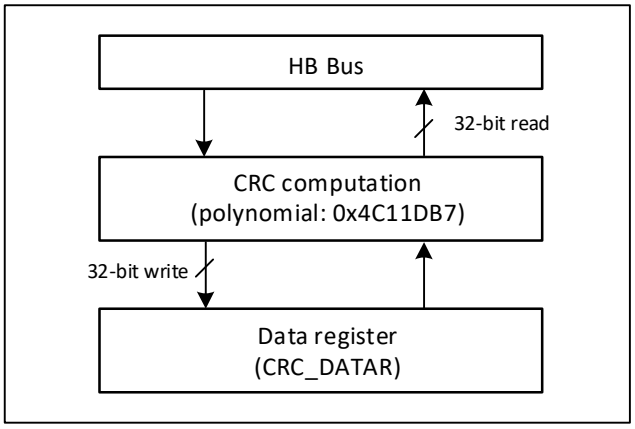
Bit	Name	Access	Description	Reset value
[15:10]	Reserved	RO	Reserved	0
9	TIF	RO	Tamper interrupt flag bit. When a tamper event is detected and the TPIE bit is set to 1, this bit is set. Cleared by writing 1 to the CTI bit. If the TPIE bit is reset, this bit is reset at the same time. *Note: This bit is reset only when the system is reset or woken up from standby mode.*	0
8	TEF	RO	Tamper event flag bit. When a tamper event is detected, this bit is set. Cleared by writing 1 to the CTE bit. *Note: When this bit is 1, all BKP_DATARx registers are cleared, and all write operations to the BKP_DATARx register are invalid before this*	0

			bit is not reset.	
[7:3]	Reserved	RO	Reserved	0
2	TPIE	RW	Tamper interrupt enable bit: 0: Disable tamper detection interrupt; 1: Enable tamper detection interrupt (TPE needs to be set to 1). *Note 1: The tamper interrupt cannot wake up the core from low-power mode.* *Note 2: This bit is reset only when the system is reset or woken up from standby mode.*	0
1	CTI	WO	Tamper detection interrupt clear bit. Write 1 to clear it, and the value read out is invalid.	0
0	CTE	WO	Tamper detection clear bit. Write 1 to clear it, and the value read out is invalid.	0

Chapter 5 Cyclic Redundancy Check (CRC)

The cyclic redundancy check (CRC) computation unit is used to obtain the result of the CRC computation for any 32-bit data based on a fixed generating polynomial. It is generally used in the field of data storage and data communication to verify the correctness of data. The system provides hardware CRC calculation unit which can greatly save CPU and RAM resources to improve efficiency.

Figure 5-1 CRC structure diagram



5.1 Main Features

- CRC32 polynomial (0x4C11DB7): $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^{8}+X^{7}+X^{5}+X^{4}+X^{2}+X+1$;
- Same 32-bit register as input for data and output for CRC32 calculation
- Single conversion time: 4 HB clock cycles (HCLK)

5.2 Function Description

- CRC unit reset

To start a CRC calculation for a new data set, the CRC calculation unit needs to be reset. Writing a 1 to the RST bit of the control register CRC_CTLR will reset the data register by the hardware, restoring the initial value 0xFFFFFFFF.

- CRC calculation

The CRC unit calculates the CRC result of the previous CRC calculation and the CRC result of the newly involved data. The CRC_DATAR data register, for which a write operation will feed new data to the hardware calculation unit; and a read operation will get the value of the latest round of CRC calculation. The hardware calculation interrupts the system write operation, so new values can be written continuously.

Note: The CRC unit calculates the entire 32-bit data, not byte-by-byte.

- Independent data buffer

The CRC unit provides an 8-bit independent data register, CRC_IDATAR, which is used for the application code to temporarily store 1 byte of data independent of the CRC unit reset.

5.3 Register Description

Table 5-1 CRC-related registers list

Name	Access address	Description	Reset value
R32_CRC_DATAR	0x40023000	Data register	0xFFFFFFFF
R8_CRC_IDATAR	0x40023004	Independent data buffer	0x00
R32_CRC_CTLR	0x40023008	Control register	0x00000000

5.3.1 Data Register (CRC_DATAR)

Offset address: 0x00

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

Bit	Name	Access	Description	Reset value
[31:0]	DR[31:0]	RW	Write raw data; read calculations.	0xFFFFFFFF

5.3.2 Independent Data Buffer (CRC_IDATAR)

Offset address: 0x04

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

Bit	Name	Access	Description	Reset value
[7:0]	IDR[7:0]	RW	An 8-bit general-purpose register that can be used as a data cache, this register is not affected by the RST field of the control register.	0

5.3.3 Control Register (CRC_CTLR)

Offset address: 0x08

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

Bit	Name	Access	Description	Reset value
[31:1]	Reserved	RO	Reserved	0
0	RST	WO	CRC calculation unit reset control, write 1	0

			execution, hardware auto clear, after execution, data register is 0xFFFFFFFF.	

Chapter 6 Real Time Clock (RTC)

The Real Time Clock (RTC) is a standalone timer module with a programmable counter up to 32 bits, which can be used with software to realize the real time clock function, and the counter value can be modified to reconfigure the current time and date of the system. The RTC module is in the backup power supply area, and the system reset and wake-up from Standby mode do not have any effect on it.

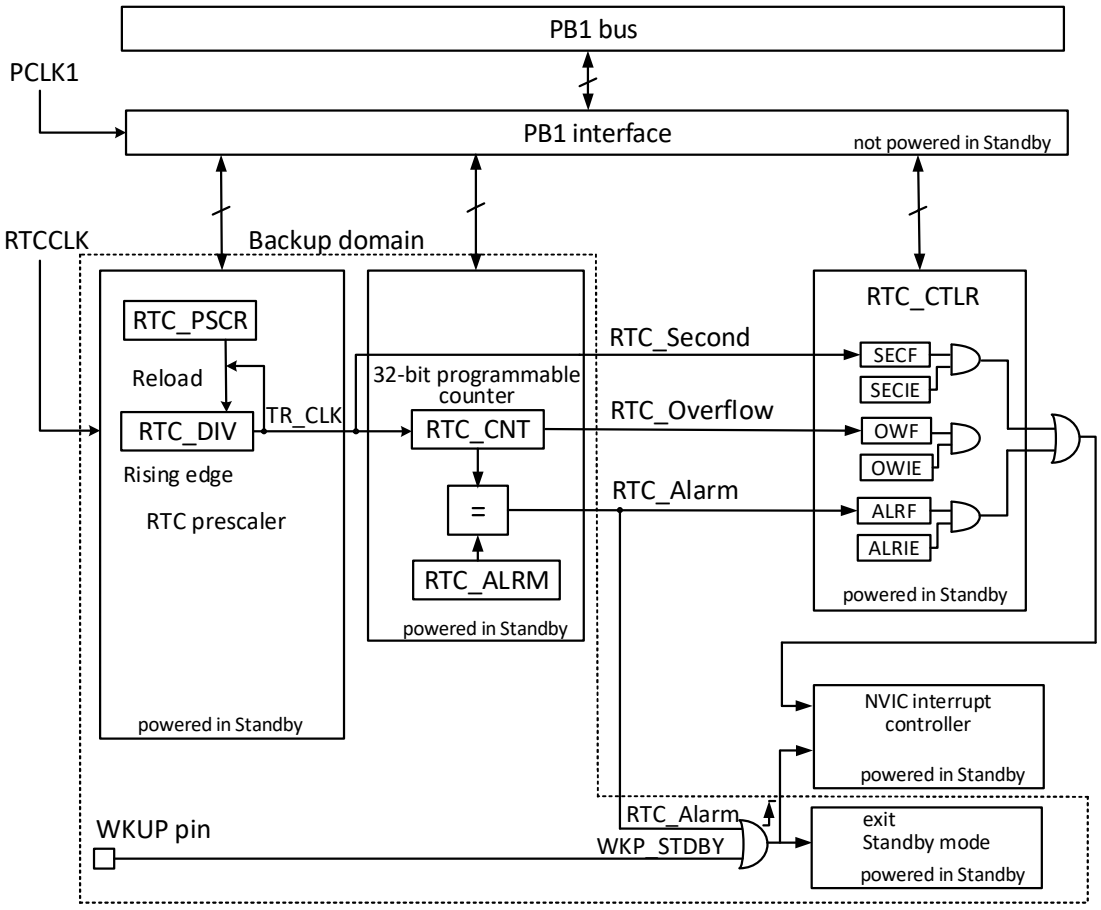
6.1 Main Features

- Prescaler coefficients up to 2^{20}
- 32-bit programmable counter
- Multiple clock sources, interrupts
- Independent reset

6.2 Function Description

6.2.1 Overview

Figure 6-1 RTC structure diagram



As shown in figure 6-1, the RTC module is mainly composed of three parts: PB1 bus interface, frequency divider and counter, control and status register, in which the frequency divider and counter are in the backup area and can

be powered by VBAT. After inputting the frequency divider (RTC_DIV), the RTCCK is divided into TR_CLK. It is worth noting that the inside of the frequency divider (RTC_DIV) is a self-subtractive counter, which will output a TR_CLK from the overrun, then take the default value from the reload value register (RTC_PSCR) and reinstall it into the frequency divider, the read divider actually reads its real-time value (read only), and the write division factor should be written to the reload value register (RTC_PSCR). Generally, when the period of TR_CLK is set to 1 second, TR_CLK will trigger the second event and increase the main counter (RTC_CNT) by 1; when the main counter is increased to the same value as the alarm clock register, the alarm clock event will be triggered; when the master counter increases to overflow, the overflow event will be triggered. All of the above three events can trigger interrupts and correspond to the corresponding interrupts to enable level control.

6.2.2 Reset

Due to the special purpose of the real-time clock, the four sets of registers that are in the backup domain: the prescaler, prescaler reload value, master counter, and alarm clock, can only be reset by the reset signal in the backup domain, refer to the RCC's Backup Domain Reset chapter. The control registers of the real-time clock are controlled by a system reset or a power reset.

6.2.3 Special Read/Write Register Operation

Due to the special use of the real-time clock, RTC and PB1 buses are independent, and the reading of RTC by PB1 is not necessarily real-time. Reading the register of RTC through PB1 must be after PB1 startup and passing through a RTC rising edge. This situation may occur after system reset and power reset, after waking up from Standby or Stop mode. It is convenient to wait for the RSF bit of the control register (CTLR) to be set high. The write operator for RTC must wait for the end of the last write operation and must enter configuration mode. The specific steps are as follows:

1) Query the RTOFF bit until it becomes 1.
2) Set the CNF bit and enter the configuration mode.
3) Write to one or more RTC registers.
4) Clear the CNF bit, exit the configuration mode, and the PB1 interface starts to write the RTC register.
5) Query the RTOFF bit until it becomes 1.

6.3 Register Description

Table 6-1 BKP-related registers list

Name	Access address	Description	Reset value
R16_RTC_CTLRH	0x40002800	RTC control register high	0x0000
R16_RTC_CTLRL	0x40002804	RTC control register low	0x0020
R16_RTC_PSCRH	0x40002808	Prescaler reload register high	0x000X
R16_RTC_PSCRL	0x4000280C	Prescaler reload register low	0xXXXX
R16_RTC_DIVH	0x40002810	Divider register high	0x000X
R16_RTC_DIVL	0x40002814	Divider register low	0xXXXX
R16_RTC_CNTH	0x40002818	RTC counter register high	0xXXXX
R16_RTC_CNTL	0x4000281C	RTC counter register low	0xXXXX
R16_RTC_ALRMH	0x40002820	Alarm clock register high	0xXXXX
R16_RTC_ALRML	0x40002824	Alarm clock register low	0xXXXX

6.3.1 RTC Control Register High (RTC_CTLRH)

Offset address: 0x00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													OWIE	ALRIE	SECIE

Bit	Name	Access	Description	Reset value
[15:3]	Reserved	RO	Reserved	0
2	OWIE	RW	Overflow interrupt enable.	0
1	ALRIE	RW	Alarm interrupt enable.	0
0	SECIE	RW	Second interrupt enable.	0

6.3.2 RTC Control Register Low (RTC_CTLRL)

Offset address: 0x04

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										RTOFF	CNF	RSF	OWF	ALRF	SECF

Bit	Name	Access	Description	Reset value
[15:6]	Reserved	RO	Reserved	0
5	RTOFF	RO	RTC operation status indication bit, indicates the execution status of the last operation to the RTC, the operation to the RTC must wait for this bit to be 1. 1: The last operation on the RTC has been completed; 0: The last operation on the RTC is still in progress.	1
4	CNF	RW	Configuration flag bit, writing this bit by 1 enters the configuration mode, thereby allowing values to be written to the Counter (R16_RTC_CNTx), the Alarm Clock Register (R16_RTC_ALRMx) and the Prescaler Reload Value Register (R16_RTC_PSCRx). The write operation is performed only after the bit is written 1 and re-cleared 0 by the software: 1: Enter configuration mode; 0: Exit configuration mode and start updating the RTC registers.	0
3	RSF	RW0	Register synchronization flag bit, before reading and writing registers such as pre-division frequency (PSCRx), alarm clock (ALRMx) and counter (CNTx) of RTC module, make sure that this bit has been set by hardware to make sure that these registers have been synchronized; when reading and writing these registers, or after PB1 reset or PB1 clock stops, the	0

			first step should reset this bit. 1: The register is synchronized; 0: The register is not synchronized.	
2	OWF	RW0	Counter overflow flag, this bit is set by hardware when the 32-bit counter overflows. An overflow interrupt is also generated if the OWIE bit is set. This bit can only be cleared by software and cannot be set by software.	0
1	ALRF	RW0	Alarm Clock Flag, this bit is set by hardware when the counter value reaches the value of the Alarm Clock Register (ALRMx), and an alarm clock interrupt is also generated if the Alarm Clock Interrupt Enable Bit (ALRIE) is set. This bit can only be cleared by software and cannot be set by software.	0
0	SECF	RW0	The second event flag, when the clock is divided by the prescaler after each falling edge, will cause the counter to increment by one, and at the same time generate a second event, this bit will be set, if the second interrupt is enabled (SECIE is set), and at the same time will also generate a second interrupt. This bit can only be cleared by software and cannot be set by software.	0

6.3.3 Prescaler Reload Register High (RTC_PSCRH)

Offset address: 0x08

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												PRL[19:16]			

Bit	Name	Access	Description	Reset value
[15:4]	Reserved	RO	Reserved	0
[3:0]	PRL[19:16]	WO	Reload value high	x

6.3.4 Prescaler Reload Register Low (RTC_PSCRL)

Offset address: 0x0C

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

Bit	Name	Access	Description	Reset value
[15:0]	PRL[15:0]	WO	The reloading value is low. The actual frequency division factor is (PRLL [19:0] + 1). For example, if the RTC input frequency is 32768Hz, then this value is set to 0x7fff, you can divide the signal with a period of 1 second.	xxxxh

6.3.5 Divider Register High (RTC_DIVH)

Offset address: 0x10

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												DIV[19:16]			

Bit	Name	Access	Description	Reset value
[15:4]	Reserved	RO	Reserved	0
[3:0]	DIV[19:16]	RO	Divider register high.	x

6.3.6 Divider Register Low (RTC_DIVL)

Offset address: 0x14

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

Bit	Name	Access	Description	Reset value
[15:0]	DIV[15:0]	RO	Divider register low. DIV is actually a self-subtractive counter. Every time a clock comes to RTC_CLK, the DIV counter is subtracted by 1. After the overflow, a TR_CLK is output and the value is reloaded from the PSCR. DIV can only read and read the remaining value of the counter of the current frequency divider.	xxxxh

6.3.7 RTC Counter High (RTC_CNTH)

Offset address: 0x18

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[31:16]															

Bit	Name	Access	Description	Reset value
[15:0]	CNT[31:16]	RW	Counter High	xxxxh

6.3.8 RTC Counter Low (RTC_CNTL)

Offset address: 0x1C

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

Bit	Name	Access	Description	Reset value
[15:0]	CNT[15:0]	RW	Counter low, the core device of the RTC timer, the clock is provided by the TRCLK (the period is generally set to 1 second). Calculate the current	xxxxh

			time by reading CNT [31:0]. To write this value, you need to enter configuration mode.	

6.3.9 Alarm Register High (RTC_ALRMH)

Offset address: 0x20

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALR[31:16]															

Bit	Name	Access	Description	Reset value
[15:0]	ALR[31:16]	WO	Alarm register high	xxxxh

6.3.10 Alarm Register Low (RTC_ALRML)

Offset address: 0x24

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

Bit	Name	Access	Description	Reset value
[15:0]	ALR[15:0]	WO	Alarm clock register is low. An alarm clock event occurs when the value of the alarm clock register ALRM [31:0] is the same as that of the counter CNT [31:0]. Changing this value requires you to enter configuration mode.	xxxxh

Chapter 7 Independent Watchdog (IWDG)

The system is equipped with an independent watchdog (IWDG) to detect software failures caused by logic errors and external environment interference. The IWDG clock source comes from LSI and can be run independently of the main program, so it is suitable for situations with low precision requirements.

7.1 Main Features

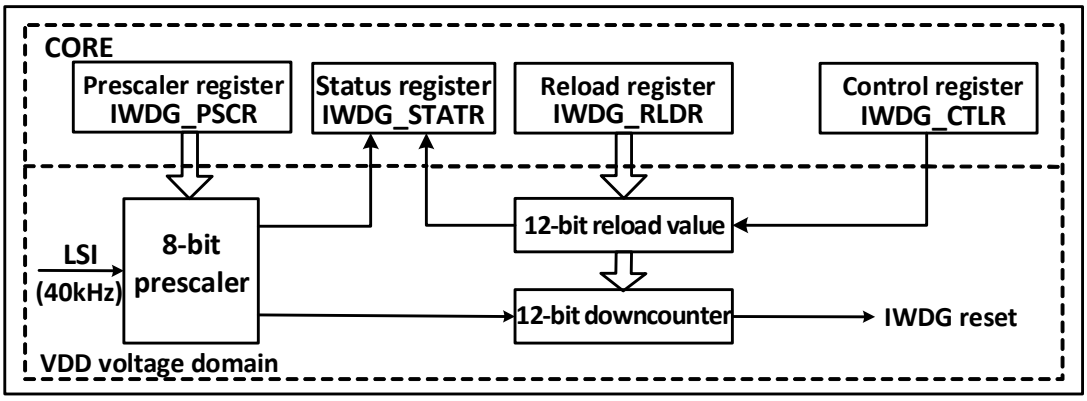
- 12-bit self-subtractive counter.
- Clock source LSI frequency division, can run in low-power mode.
- Reset condition: counter value reduced to 0

7.2 Functional Description

7.2.1 Principle and Application

The stand-alone watchdog is clocked from the LSI clock and functions in shutdown and standby modes. When the watchdog counter decrements itself to 0, a system reset will be generated, so the timeout is (reload value + 1) clock.

Figure 7-1 Structure block diagram of Independent Watchdog



- Enable independent watchdog

After the system reset, the watchdog is OFF. Write 0xCCCC to the IWDG_CTLR register to enable the watchdog, and then it can no longer be disabled unless a reset occurs.

If the hardware independent watchdog enable bit (IWDG_SW) is enabled in User Option Bytes, the IWDG is permanently enabled after the microcontroller reset.

- Watchdog configuration

Inside the watchdog is a 12-bit counter running progressively. When the value of the counter is reduced to 0, a system reset will occur. To enable the IWDG function, you need to perform the following actions:

1) Count time base: IWDG clock source LSI, through the IWDG_PSCR register to set the LSI frequency division value clock as the IWDG count time base. The operation method first writes 0x5555 to the IWDG_CTLR register, and then modifies the frequency division value in the IWDG_PSCR register. The PVU bit in the IWDG_STATR

register indicates the update status of the frequency division value, and the frequency division value can only be modified and read out when the update is completed.

2) Reload value: used to update the current value of the counter in the independent watchdog, and the counter is decremented by this value. The operation method first writes 0x5555 to the IWDG_CTLR register, and then modifies the IWDG_RLDR register to set the target reload value. The RUV bit in the IWDG_STATR register indicates the update status of the reload value, and the IWDG_RLDR register can be modified and read out only after the update is completed.

3) Watchdog enable: write 0xCCCC to the IWDG_CTLR register to turn on the watchdog function.

4) Feeding the dog: that is, before the watchdog counter decreases to 0, refresh the current counter value to prevent system reset. Write 0xAAAA to the IWDG_CTLR register and have the hardware update the IWDG_RLDR register value to the watchdog counter. This action needs to be performed regularly after the watchdog function is turned on, otherwise the watchdog reset action will occur.

7.2.2 Debug Mode

When the system enters the debug mode, the IWDG counter can be configured by the debug module register to continue working or stop.

7.3 Register Description

Table 7-1 IWDG-related registers

Name	Access address	Description	Reset value
R16_IWDG_CTLR	0x40003000	Control register	0x0000
R16_IWDG_PSCR	0x40003004	Prescaler register	0x0000
R16_IWDG_RLDR	0x40003008	Reload register	0x0FFF
R16_IWDG_STATR	0x4000300C	Status register	0x0000

7.3.1 IWDG Control Register (IWDG_CTLR)

Offset address: 0x00

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

Bit	Name	Access	Description	Reset value
[15:0]	KEY[15:0]	WO	Operate the key value lock. 0xAAAA: Feed the dog. Load the IWDG_RLDR register value into the independent watchdog counter. 0x5555: Allow modification of R16_IWDG_PSCR and R16_IWDG_RLDR registers. 0xCCCC: Start the watchdog, which is not subject to this restriction if the hardware watchdog is enabled (user chooses word configuration).	0

7.3.2 Prescaler Register (IWDG_PSCR)

Offset address: 0x04

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

Bit	Name	Access	Description	Reset value
[15:3]	Reserved	RO	Reserved	0
[2:0]	PR[2:0]	RW	IWDG clock division factor, write 0x5555 to KEY before modifying this field. 000: Divided by 4; 001: Divided by 8; 010: Divided by 16; 011: Divided by 32; 100: Divided by 64; 101: Divided by 128; 110: Divided by 256; 111: Reserved. IWDG count time base = LSI/division factor. *Note: Before reading the value of this field, make sure that the PVU bit in the IWDG_STATR register is 0, otherwise the read value is invalid.*	000b

7.3.3 Reload Value Register (IWDG_RLDR)

Offset address: 0x08

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

Bit	Name	Access	Description	Reset value
[15:12]	Reserved	RO	Reserved	0
[11:0]	RL[11:0]	RW	Counter reload value. Write 0x5555 to KEY before modifying this field. When 0xAAAA is written to KEY, the value of this field is loaded into the counter by the hardware, and the counter is then decremented from that value. *Note: Before reading and writing the field value, make sure that the RUV bit in the IWDG_STATR register is 0, otherwise it is invalid to read and write this field.*	FFFh

Note: This register is reset in Standby mode.

7.3.4 Status Register (IWDG_STATR)

Offset address: 0x0C

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

Bit	Name	Access	Description	Reset value
[15:2]	Reserved	RO	Reserved	0
1	RVU	RO	The reinstall value updates the flag bit. Hardware setting or clearing 0. 1: Reload value update is in progress. 0: Reload update ends (up to 5 LSI cycles). *Note: The reload value register IWDG_RLDR can be read and written only after the RVU bit has been cleared.*	0
0	PVU	RO	Clock frequency division coefficient updates flag bits. Hardware setting or clearing 0. 1: Clock division value update is in progress. 0: Clock division value update ends (up to 5 LSI cycles). Note: The frequency division factor register IWDG_PSCR can be read and written only after the PVU bit has been cleared.	0

Note: After the pre-division or reinstallation value is updated, you do not have to wait for RVU or PVU to reset, you can continue to execute the following code. (This write operation continues to be completed even in low power mode.)

Chapter 8 Window Watchdog (WWDG)

The window watchdog is generally used to monitor the software failures of the system, such as external interference, unforeseen logic errors and so on. It needs to refresh the counter (feed the dog) within a specific window time (with upper and lower limits), otherwise the watchdog circuit will produce a system reset earlier or later than this window time.

8.1 Main Features

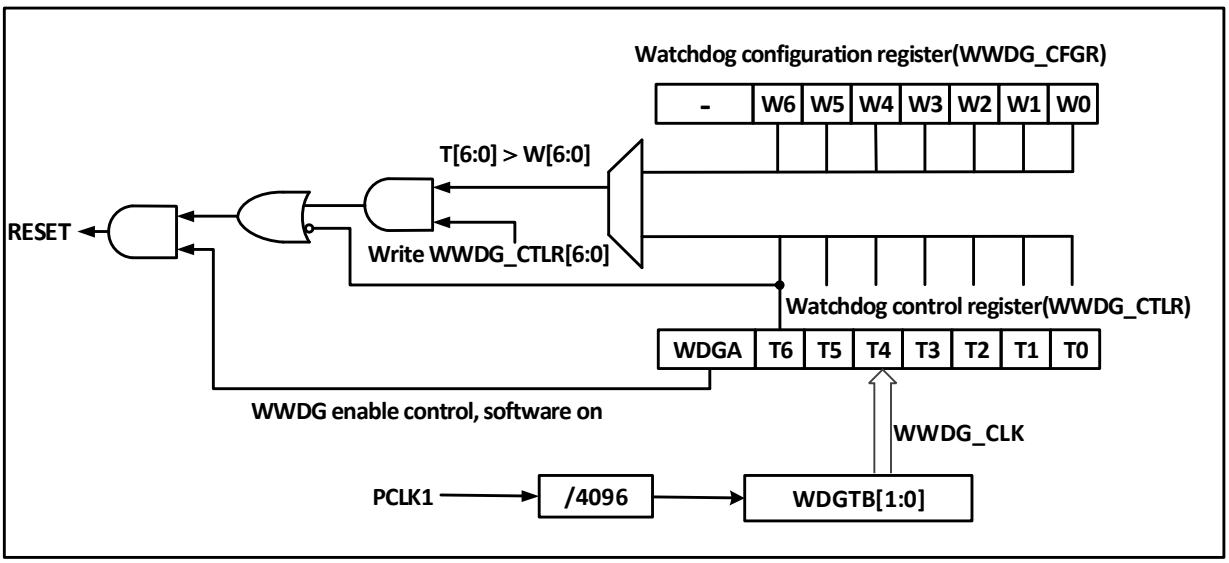
- Programmable 7-bit self-subtractive counter.
- Double conditional reset: the current counter value is less than 0x40, or the counter value is reloaded outside the window time.
- Wake up advance notice function (EWI), which is used to feed the dog in time to prevent system reset

8.2 Function Description

8.2.1 Principle and Application

The window watchdog runs based on a 7-bit decrement counter, which is mounted on the PB1 bus to count the frequency division of the time-based WWDG _ CLK source (PCLK1/4096) clock, and the frequency division factor is set in the WDGTB [1:0] field in the configuration register WWDG_CFGR. The decrement counter is in a state of free operation, regardless of whether the watchdog function is turned on or not, the counter has been cyclically decreasing counting. As shown in figure 8-1, the internal structure block diagram of the window watchdog.

Figure 8-1 Window Watchdog structure diagram



- Enable window watchdog

After the system reset, the watchdog is disabled. Set the WDGA bit in the WWDG_CTLR register to switch on the watchdog, and then it can no longer be disabled unless a reset occurs.

Note: WWDG clock source can be disabled by setting the RCC_PB1PCENR register to suspend WWDG_CLK counting and indirectly stop the watchdog function. Or reset the WWDG module by setting the RCC_PB1PRSTR register, which is equivalent to the function of reset.

- Watchdog configuration

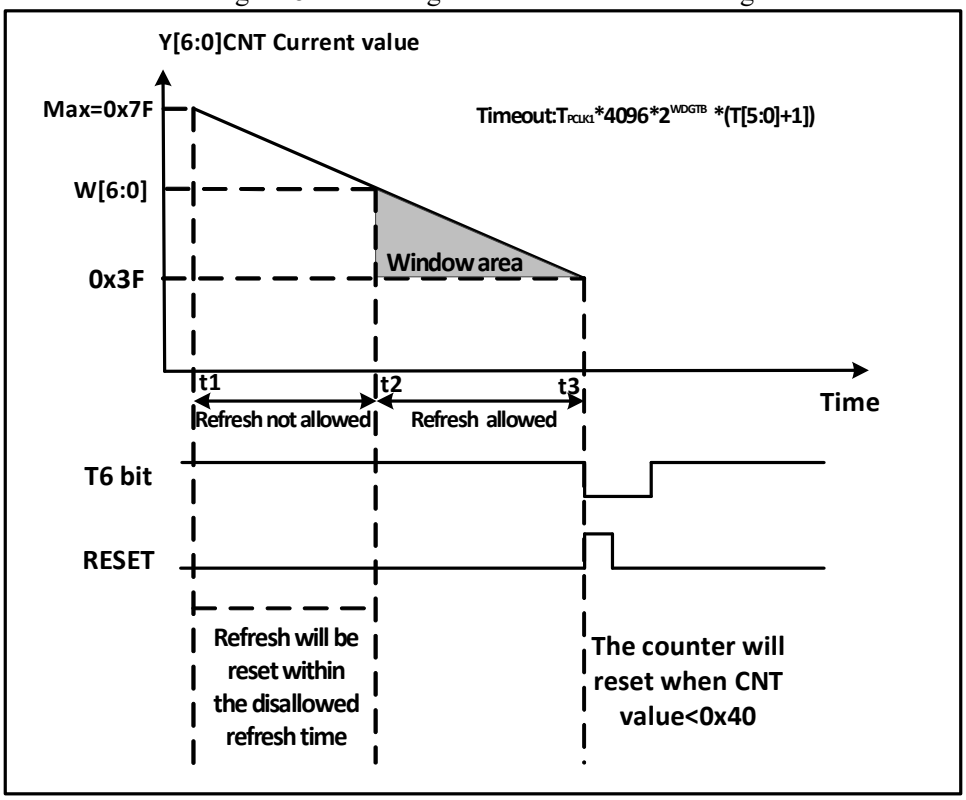
Inside the watchdog is a 7-bit counter running in a continuous cycle, which supports read and write access. To use the watchdog reset function, you need to perform the following actions:

1) Counter time base: The WDGTB[1:0] bits in the WWDG_CFGR register. Note to switch on the WWDG module clock of the RCC unit.
2) Window counter: Set the W[6:0] bits in the WWDG_CFGR register. This counter is used to be compared with the current counter by hardware, the value is configured by the user software and will not change. It serves as the maximum value of window time.
3) Watchdog enable: The WDGA bit in the WWDG_CTLR register is set to 1 by software, and the watchdog function is enabled to reset the system.
4) Feed dog: Refresh the current counter value and configure the T[6:0] bits in the WWDG_CTLR register. This action needs to be executed in the periodic window time after the watchdog function is enabled. Otherwise, the watchdog reset action occurs.

- Feed dog window time

As shown in Figure 8-2, the gray area is the detector window area of the window watchdog. Its maximum timeout (t_2) corresponds to the time point when the current counter value reaches the window value W[6:0]. Its minimum timeout (t_3) corresponds to the time point when the current counter value reaches 0x3F. Within this area time ($t_2<t<t_3$), the feed dog operation can be performed (write T[6:0]) to refresh the current counter value.

Figure 8-2 Counting mode of window watchdog



- Watchdog reset:

1) When the feed dog operation is not performed in time, the value of the T[6:0] counter changes from 0x40 to 0x3F, a "Window Watchdog Reset" occurs, and a system reset occurs. I.e., when T6-bit is detected as 0 by hardware, the system reset occurs.

Note: The application program can write 0 to the T6-bit by software to implement system reset, which is equivalent to software reset function.

2) When the counter refresh action is executed when the feed dog operation is disabled, i.e., when write operation is performed on the T[6:0] bits when $t_1 \leq t \leq t_2$, a "window watchdog reset" occurs, and a system reset occurs.

- Early wake-up

To prevent the system from resetting due to failure to refresh the counter in time, the watchdog module provides early wake-up interrupt (EWI) notification. When the counter is reduced to 0x40, an early wake-up signal is generated, and the WEIF flag is set to 1. If the EWI bit is set, the window watchdog will be triggered to interrupt at the same time. At this point, there is a counter clock cycle (self-reduced to 0x3F) from the hardware reset, during which the application can immediately feed the dog.

8.2.2 Debug Mode

When the system enters debug mode, the WWDG counter can either continues to work normally or stops, depending on the debugging module register.

8.3 Register Description

Table 8-1 WWDG-related registers

Name	Access address	Description	Reset value
R16_WWDG_CTLR	0x40002C00	Control register	0x007F
R16_WWDG_CFGR	0x40002C04	Configuration register	0x007F
R16_WWDG_STATR	0x40002C08	Status register	0x0000

8.3.1 WWDG Control Register (WWDG_CTLR)

Offset address: 0x00

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

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved	0
7	WDGA	RW1	The window watchdog resets the enable position. 1: Enable the watchdog function (can generate reset signal). 0: Disable the watchdog function. Software write 1 is on, but only hardware is allowed to clear 0 after reset.	0
[6:0]	T[6:0]	RW	7-bit self-subtractive counter, minus 1 per $4096*2^{WDGTB}$ PCLK1 cycle. When the counter is reduced from 0x40 to 0x3F, that is, when T6 jumps to 0, a watchdog reset is generated.	7Fh

8.3.2 WWDG Configuration Register (WWDG_CFGR)

Offset address: 0x04

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						EWI	WDGTB[1:0]		W[6:0]						

Bit	Name	Access	Description	Reset value
[15:10]	Reserved	RO	Reserved	0
9	EWI	RW1	Early wakeup interrupt: If it set to 1, interrupt is generated when the counter reaches 0x40. It can only be cleared by hardware after reset.	0
[8:7]	WDGTB[1:0]	RW	Window watchdog time base: 00: Divided by 1, counter time base =PCLK1/4096; 01: Divided by 2, counter time base =PCLK1/4096/2; 10: Divided by 4, counter time base = PCLK1/4096/4; 11: Divided by 8, counter time base = PCLK1/4096/8.	00b
[6:0]	W[6:0]	RW	Window watchdog 7-bit window value. It is used to be compared with the counter value. The feed dog operation can be performed only when the counter value is less than the window value and is greater than 0x3F.	7Fh

8.3.3 WWDG Status Register (WWDG_STATR)

Offset address: 0x08

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

Bit	Name	Access	Description	Reset value
[15:1]	Reserved	WO	Reserved	0
0	EWIF	RW0	Early wakeup interrupt flag. When the counter reaches 0x40, this bit is set by hardware, and it must be cleared by software. User setting is invalid. Even if the EWI is not set, this bit is still set as usual when the event occurs.	0

Chapter 9 Interrupt and Events (PFIC)

The built-in Programmable Fast Interrupt Controller (PFIC) supports up to 255interrupt vectors. The current system manages 52 peripheral interrupt channels and 4 core interrupt channels, and the rest are retained.

9.1 Main Features

9.1.1 PFIC

- 52 peripheral interrupts, each interrupt request has an independent trigger and mask control bit, and a dedicated status bit.
- Programmable multi-level interrupt nesting, maximum nesting depth 2 levels, hardware stack depth 2 levels.
- Dedicated fast interrupt in and out mechanism, hardware automatic stack and recovery, no instruction overhead.
- Vector Table Free (VTF) interrupt response mechanism, 4-channel programmable direct interrupt vector address

9.2 SysTick

The core comes with a 64-bit addition and subtraction counter (SysTick), which supports HCLK or HCLK/8 as a time base, has a higher priority, and can be used as a time benchmark after calibration.

9.3 Vector Table of Interrupt and Exception

Table 9-1 Vector table

No.	Priority	Type	Name	Description	Entrance address
0	-	-	-	-	0x00000000
1	-	-	-	-	0x00000004
2	-5	fixed	NMI	Non-maskable interrupt	0x00000008
3	-4	fixed	HardFault	Abnormal interruption	0x0000000C
4	-	-	-	Reserved	0x00000010
5	-3	fixed	Ecall-M	Machine mode callback interrupt	0x00000014
6-7	-	-	-	Reserved	0x00000018-0x0000001C
8	-2	fixed	Ecall-U	User mode callback interrupt	0x00000020
9	-1	fixed	BreakPoint	Breakpoint callback interrupt	0x00000024
10-11	-	-	-	Reserved	0x00000028-0x0000002C
12	0	programmable	SysTick	SysTick interrupt	0x00000030
13	-	-	-	Reserved	0x00000034
14	1	programmable	SW	Software interrupt	0x00000038
15	-	-	-	Reserved	0x0000003C
16	2	programmable	WWDG	Window watchdog timer interrupt	0x00000040
17	3	programmable	PVD	Supply Voltage Detection Interrupt (EXTI)	0x00000044

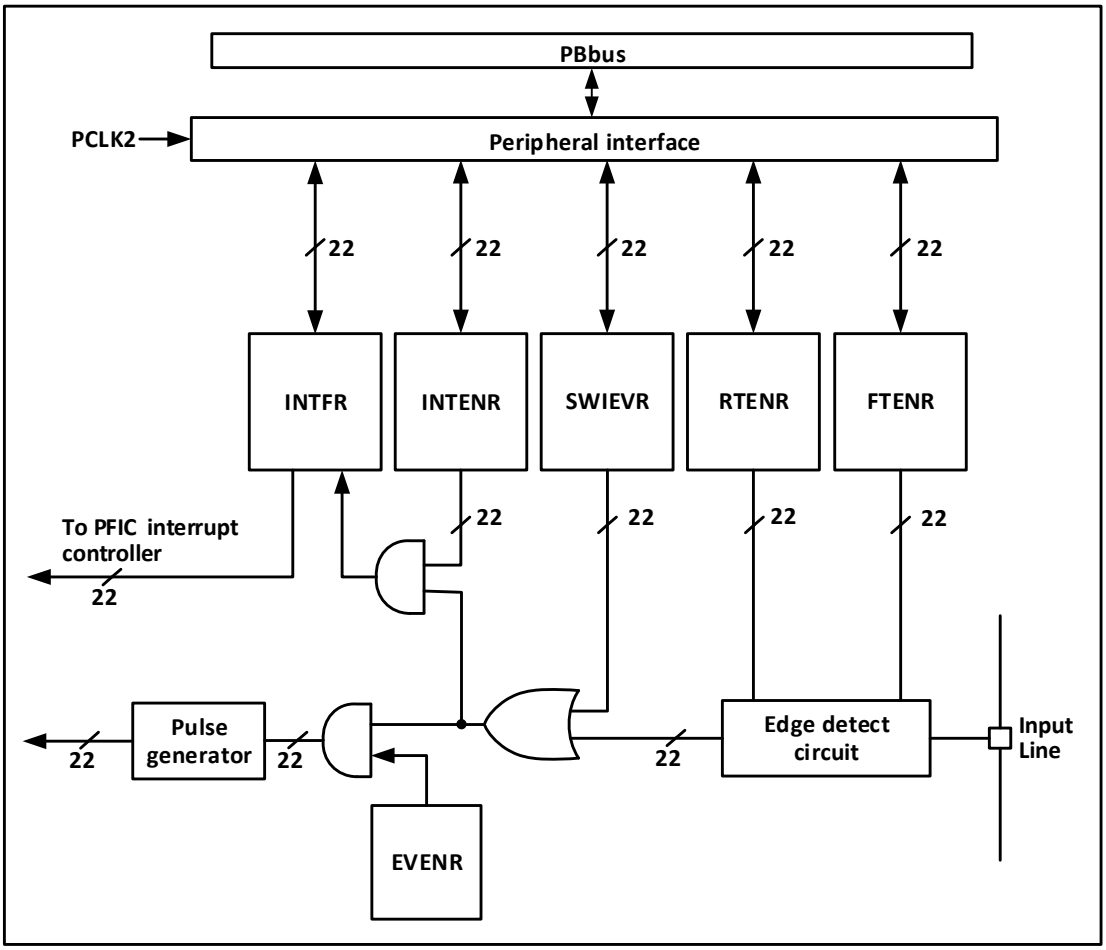
18	4	programmable	TAMPER	Tamper detection interrupt	0x00000048
19	5	programmable	RTC	Real-time clock interrupt	0x0000004C
20	6	programmable	FLASH	Flash global interrupt	0x00000050
21	7	programmable	RCC	Reset and clock control interrupts	0x00000054
22	8	programmable	EXTI0	EXTI line 0 interrupt	0x00000058
23	9	programmable	EXTI1	EXTI line 1 interrupt	0x0000005C
24	10	programmable	EXTI2	EXTI line 2 interrupt	0x00000060
25	11	programmable	EXTI3	EXTI line 3 interrupt	0x00000064
26	12	programmable	EXTI4	EXTI line 4 interrupt	0x00000068
27	13	programmable	DMA_CH1	DMA1 channel 1 global interrupt	0x0000006C
28	14	programmable	DMA_CH2	DMA1 channel 2 global interrupt	0x00000070
29	15	programmable	DMA_CH3	DMA1 channel 3 global interrupt	0x00000074
30	16	programmable	DMA_CH4	DMA1 channel 4 global interrupt	0x00000078
31	17	programmable	DMA_CH5	DMA1 channel 5 global interrupt	0x0000007C
32	18	programmable	DMA_CH6	DMA1 channel 6 global interrupt	0x00000080
33	19	programmable	DMA_CH7	DMA1 channel 7 global interrupt	0x00000084
34	20	programmable	ADC	ADC global interrupt	0x00000088
35	21	programmable	USB_HP or CAN_TX	USB_HP or CAN_TX global interrupt	0x0000008C
36	22	programmable	USB_LP or CAN_RX0	USB_LP or CAN_RX0 global interrupt	0x00000090
37	23	programmable	CAN_RX1	CAN_RX1 global interrupt	0x00000094
38	24	programmable	CAN_SCE	CAN_SCE global interrupt	0x00000098
39	25	programmable	EXTI9_5	EXTI line[9:5] interrupts	0x0000009C
40	26	programmable	TIM1_BRK	TIM1 break interrupt	0x000000A0
41	27	programmable	TIM1_UP	TIM1 update interrupt	0x000000A4
42	28	programmable	TIM1_TRG_COM	TIM1 trigger and communication interrupts	0x000000A8
43	29	programmable	TIM1_CC	TIM1 capture compare interrupt	0x000000AC
44	30	programmable	TIM2	TIM2 global interrupt	0x000000B0
45	31	programmable	TIM3	TIM3 global interrupt	0x000000B4
46	32	programmable	TIM4	TIM4 global interrupt	0x000000B8
47	33	programmable	I2C1_EV	I2C1 event interrupt	0x000000BC
48	34	programmable	I2C1_ER	I2C1 error interrupt	0x000000C0
49	35	programmable	I2C2_EV	I2C2 event interrupt	0x000000C4
50	36	programmable	I2C2_ER	I2C2 error interrupt	0x000000C8
51	37	programmable	SPI1	SPI1 global interrupt	0x000000CC
52	38	programmable	SPI2	SPI2 global interrupt	0x000000D0
53	39	programmable	USART1	USART1 global interrupt	0x000000D4
54	40	programmable	USART2	USART2 global interrupt	0x000000D8
55	41	programmable	USART3	USART3 global interrupt	0x000000DC
56	42	programmable	EXTI15_10	EXTI line[15:10] interrupts	0x000000E0
57	43	programmable	RTCAlarm	RTC alarm clock (EXTI)	0x000000E4

58	44	programmable	LPTIM_WKUP	LPTIM wakeup interrupt	0x000000E8
59	45	programmable	USBFS	USBFS global interrupt	0x000000EC
60	46	programmable	USBFS_WKUP	USBFS wakeup interrupt	0x000000F0
61	47	programmable	USART4	USART4 global interrupt	0x000000F4
62	48	programmable	DMA_CH8	DMA channel 8 global interrupt	0x000000F8
63	49	programmable	LPTIM	LPTIM global interrupt	0x000000FC
64	50	programmable	OPA	OPA global interrupt	0x00000100
65	51	programmable	USBPD	USBPD global interrupt	0x00000104
66	-	-	-	-	0x00000108
67	52	programmable	USBPD_WKUP	USBPD wakeup interrupt	0x0000010C
68	53	programmable	CMP_WKUP	CMP wakeup interrupt	0x00000110

9.4 External Interrupt and Event Controller (EXTI)

9.4.1 Overview

Figure 9-1 External interrupt (EXTI) interface block diagram



As can be seen from figure 9-1, the trigger source of the external interrupt can be either a software interrupt (SWIEVR) or an actual external interrupt channel, and the signal of the external interrupt channel is first screened

by the edge detection circuit. As long as one of the software interrupts or external interrupt signals is generated, it will be output to both event enabling and interrupt enabling circuits through the OR gate circuit in the diagram. As long as an interrupt is enabled or an event is enabled, an interrupt or event will occur. The six registers of EXTI are accessed by the processor through the PB2 interface.

9.4.2 Wakeup Event

The system can wake up sleep patterns caused by WFE instructions through wake-up events. Wake-up events are generated through the following two configurations:

- Enable an interrupt in the register of the peripheral, but not in the PFIC of the core, as well as the SEVONPEND bit in the core. Reflected in EXTI, it enables EXTI interrupts, but does not enable EXTI interrupts in PFIC, while enabling SEVONPEND bits. When CPU wakes up from WFE, the interrupt flag bit and PFIC hang bit of EXTI need to be cleared.
- Enable an EXTI channel to be an event channel, and after CPU wakes up from WFE, there is no need to clear the operation of the interrupt flag bit and PFIC hang bit.

9.4.3 Description

The use of external interrupt needs to configure the corresponding external interrupt channel, that is, select the appropriate trigger edge to enable the corresponding interrupt. When a set trigger edge appears on the external interrupt channel, an interrupt request will be generated and the corresponding interrupt flag bit will be set. Write 1 to the flag bit to clear it.

Use external hardware interrupt steps:
1) Configure GPIO operation.
2) Configure the interrupt enable bit (EXTI_INTENR) of the corresponding external interrupt channel.
3) Configure trigger edge (EXTI_RTENR or EXTI_FTENR), select rising edge trigger, falling edge trigger or double edge trigger.
4) Configure EXTI interrupts in the PFIC of the core to ensure that they respond correctly.

To use external hardware events:
1) Configure GPIO operation.
2) Configure the event enable bit (EXTI_EVENR) of the corresponding external interrupt channel.
3) Configure trigger edge (EXTI_RTENR or EXTI_FTENR), select rising edge trigger, falling edge trigger or double edge trigger.

Use software interrupt / event steps:
1) Enable external interrupt (EXTI_INTENR) or external event (EXTI_EVENR).
2) If you use the interrupt service function, you need to set the EXTI interrupt in the PFIC of the core.
3) Set the software interrupt trigger (EXTI_SWIEVR), that is, an interrupt will occur.

9.4.4 External Event Map

Table 9-2 EXTI interrupt map

External interrupt/event line	Mapping event description
EXTI0~EXTI15	Px0~Px15 (x=A/B/C/D), any of the IO ports can be enabled for external interrupt/event functionality,

	configured by the AFIO_EXTICRx register.
EXTI16	PVD event: Exceed the voltage detector threshold
EXTI17	RTC alarm event
EXTI19	USBPD wakeup event
EXTI20	USBFS wakeup event
EXTI21	LPTIM wakeup event
EXTI22	COMP wakeup event

9.5 Register Description

9.5.1 EXTI Registers

Table 9-3 EXTI-related registers

Name	Access address	Description	Reset value
R32_EXTI_INTENR	0x40010400	Interrupt enable register	0x00000000
R32_EXTI_EVENR	0x40010404	Event enable register	0x00000000
R32_EXTI_RTENR	0x40010408	Rising edge trigger enable register	0x00000000
R32_EXTI_FTENR	0x4001040C	Falling edge trigger enable register	0x00000000
R32_EXTI_SWIEVR	0x40010410	Software interrupt event register	0x00000000
R32_EXTI_INTFR	0x40010414	Interrupt flag register	0x0000XXXX

9.5.1.1 Interrupt Enable Register (EXTI_INTENR)

Offset address: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									MR22	MR21	MR20	MR19	Reserved	MR17	MR16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR15	MR14	MR13	MR12	MR11	MR10	MR9	MR8	MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0

Bit	Name	Access	Description	Reset value
[31:23]	Reserved	RO	Reserved	0
[22:0]	MRx	RW	Enable the interrupt request signal for external interrupt channel x: 1: Enable interrupt for this channel; 0: Mask interrupt for this channel.	0

9.5.1.2 Event Enable Register (EXTI_EVENR)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									MR22	MR21	MR20	MR19	Reserved	MR17	MR16

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

MR15	MR14	MR13	MR12	MR11	MR10	MR9	MR8	MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0

Bit	Name	Access	Description	Reset value
[31:23]	Reserved	RO	Reserved	0
[22:0]	MRx	RW	Enable the event request signal for external interrupt channel x: 1: Enable event for this channel; 0: Mask event for this channel.	0

9.5.1.3 Rising Edge Trigger Enable Register (EXTI_RTENR)

Offset address: 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									TR22	TR21	TR20	TR19	Reserved	TR17	TR16

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

Bit	Name	Access	Description	Reset value
[31:23]	Reserved	RO	Reserved	0
[22:0]	TRx	RW	Enable rising edge triggering of external interrupt channel x: 1: Enable rising edge triggering for this channel; 0: Disable rising edge triggering for this channel.	0

9.5.1.4 Falling Edge Trigger Enable Register (EXTI_FTENR)

Offset address: 0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									TR22	TR21	TR20	TR19	Reserved	TR17	TR16

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

Bit	Name	Access	Description	Reset value
[31:23]	Reserved	RO	Reserved	0
[22:0]	TRx	RW	Enable falling edge triggering of external interrupt channel x: 1: Enable falling edge triggering for this channel; 0: Disable falling edge triggering for this channel.	0

9.5.1.5 Software Interrupt Event Register (EXTI_SWIEVR)

Offset address: 0x10

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIER15	SWIER 14	SWIER13	SWIER12	SWIER11	SWIER 10	SWIER 9	SWIER 8	SWIER 7	SWIER 6	SWIER 5	SWIER 4	SWIER 3	SWIER 2	SWIER 1	SWIER 0

Bit	Name	Access	Description	Reset value
[31:23]	Reserved	RO	Reserved	0
[22:0]	SWIERx	RW	A software interrupt is set on the corresponding externally triggered interrupt channel. Setting it here causes the interrupt flag bit (EXTI_INTFR) to be set to the corresponding position bit, and an interrupt or event is generated if the interrupt enable (EXTI_INTENR) or event enable (EXTI_EVENR) is turned on.	0

9.5.1.6 Interrupt Flag Register (EXTI_INTFR)

Offset address: 0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									IF22	IF21	IF20	IF19	Reserved	IF17	IF16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IF15	IF14	IF13	IF12	IF11	IF10	IF9	IF8	IF7	IF6	IF5	IF4	IF3	IF2	IF1	IF0

Bit	Name	Access	Description	Reset value
[31:23]	Reserved	RO	Reserved	0
[22:0]	IFx	RW1	Interrupt Flag Bit, this location bit flag indicates that the corresponding external interrupt has occurred. Writing 1 clears this bit.	0

9.5.2 PFIC Registers

Table 9-4 PFIC-related registers

Name	Access address	Description	Reset value
R32_PFIC_ISR1	0xE000E000	PFIC interrupt enable status register 1	0x0000000C
R32_PFIC_ISR2	0xE000E004	PFIC interrupt enable status register 2	0x00000000
R32_PFIC_ISR3	0xE000E008	PFIC interrupt enable status register 3	0x00000000
R32_PFIC_IPR1	0xE000E020	PFIC interrupt pending status register 1	0x00000000
R32_PFIC_IPR2	0xE000E024	PFIC interrupt pending status register 2	0x00000000
R32_PFIC_IPR3	0xE000E028	PFIC interrupt pending status register 3	0x00000000

R32_PFIC_ITHRESDR	0xE000E040	PFIC interrupt priority threshold configuration register	0x00000000
R32_PFIC_CFGR	0xE000E048	PFIC interrupt configuration register	0x00000000
R32_PFIC_GISR	0xE000E04C	PFIC interrupt global status register	0x00000000
R32_PFIC_VTFIDR	0xE000E050	PFIC VTF interrupt ID configuration register	0x00000000
R32_PFIC_VTFADDRR0	0xE000E060	PFIC VTF interrupt 0 address register	0x00000000
R32_PFIC_VTFADDRR1	0xE000E064	PFIC VTF interrupt 1 address register	0x00000000
R32_PFIC_VTFADDRR2	0xE000E068	PFIC VTF interrupt 2 address register	0x00000000
R32_PFIC_VTFADDRR3	0xE000E06C	PFIC VTF interrupt 3 address register	0x00000000
R32_PFIC_IENR1	0xE000E100	PFIC interrupt enable set register 1	0x00000000
R32_PFIC_IENR2	0xE000E104	PFIC interrupt enable set register 2	0x00000000
R32_PFIC_IENR3	0xE000E108	PFIC interrupt enable set register 3	0x00000000
R32_PFIC_IRER1	0xE000E180	PFIC interrupt enable clear register 1	0x00000000
R32_PFIC_IRER2	0xE000E184	PFIC interrupt enable clear register 2	0x00000000
R32_PFIC_IRER3	0xE000E188	PFIC interrupt enable clear register 3	0x00000000
R32_PFIC_IPSR1	0xE000E200	PFIC interrupt pending set register 1	0x00000000
R32_PFIC_IPSR2	0xE000E204	PFIC interrupt pending set register 2	0x00000000
R32_PFIC_IPSR3	0xE000E208	PFIC interrupt pending set register 3	0x00000000
R32_PFIC_IPRR1	0xE000E280	PFIC interrupt pending clear register 1	0x00000000
R32_PFIC_IPRR2	0xE000E284	PFIC interrupt pending clear register 2	0x00000000
R32_PFIC_IPRR3	0xE000E288	PFIC interrupt pending clear register 3	0x00000000
R32_PFIC_IACTR1	0xE000E300	PFIC interrupt activation register 1	0x00000000
R32_PFIC_IACTR2	0xE000E304	PFIC interrupt activation register 2	0x00000000
R32_PFIC_IACTR3	0xE000E308	PFIC interrupt activation register 3	0x00000000
R32_PFIC_IPRIORx	0xE000E400	PFIC interrupt priority configuration register	0x00000000
R32_PFIC_SCTLR	0xE000ED10	PFIC system control register	0x00000000

Note: 1. NMI, EXC, ECALL-M, ECALL-U, BREAKPOINT interrupts are always enabled by default.

2. ECALL-M, ECALL-U, and BREAKPOINT are all EXC cases, and the status is indicated by EXC status bit 3.

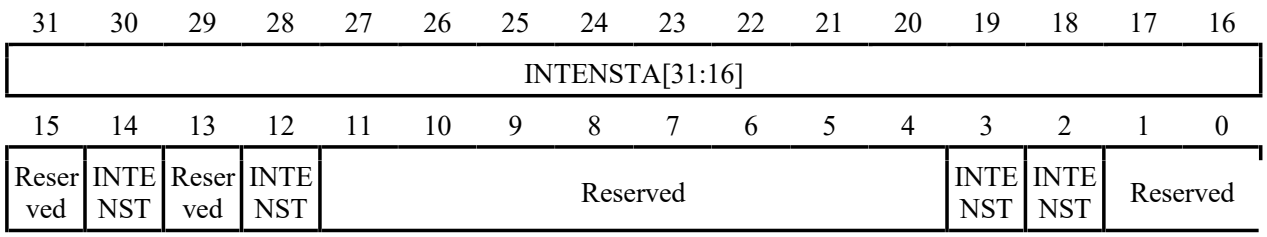
3. NMI and EXC support interrupt suspend clear and set operations, but not interrupt enable clear and set operations.

4. ECALL-M, ECALL-U, and BREAKPOINT do not support interrupt hang clear and set, interrupt enable clear and set operations.

Note: When masking arbitrary interrupts using the PFIC_IENRx register or global interrupts using the CSR register, add a "fence.i" instruction for synchronization between the core control state and the interrupt enable state.

9.5.2.1 PFIC Interrupt Enable Status Register 1 (PFIC_ISR1)

Offset address: 0x00



	A14		A12		A3	A2	

Bit	Name	Access	Description	Reset value
[31:16]	INTENSTA	RO	16#-31#Interrupt current enable status. 1: Current numbered interrupt is enabled; 0: Current numbered interrupt not enabled.	0
15	Reserved	RO	Reserved	0
14	INTENSTA	RO	14# Interrupt current enable status. 1: Current numbered interrupt is enabled; 0: Current numbered interrupt not enabled.	0
13	Reserved	RO	Reserved	0
12	INTENSTA	RO	12# Interrupt current enable status. 1: Current numbered interrupt is enabled; 0: Current numbered interrupt not enabled.	0
[11:4]	Reserved	RO	Reserved	0
[3:2]	INTENSTA	RO	2#-3# Interrupt current enable status. 1: Current numbered interrupt is enabled; 0: Current numbered interrupt not enabled.	0
[1:0]	Reserved	RO	Reserved	0

9.5.2.2 PFIC Interrupt Enable Status Register 2 (PFIC_ISR2)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INTENSTA[63:48]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTENSTA[47:32]															

Bit	Name	Access	Description	Reset value
[31:0]	INTENSTA	RO	32#-63# Interrupt current enable status. 1: Current numbered interrupt is enabled; 0: Current numbered interrupt not enabled.	0

9.5.2.3 PFIC Interrupt Enable Status Register 3 (PFIC_ISR3)

Offset address: 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											INTENSTA[68:64]				

Bit	Name	Access	Description	Reset value
[31:5]	Reserved	RO	Reserved	0

[4:0]	INTENSTA	RO	64#-68# Interrupt current enable status. 1: Current numbered interrupt is enabled; 0: Current numbered interrupt not enabled.	0

9.5.2.4 PFIC Interrupt Pending Status Register 1 (PFIC_IPR1)

Offset address: 0x20

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	PENDSTA14	Reserved	PENDSTA12	Reserved								PENDSTA3	PENDSTA2	Reserved	

Bit	Name	Access	Description	Reset value
[31:12]	PENDSTA	RO	16#-31#interrupts currently pending status. 1: Current numbered interrupt is pending; 0: Current numbered interrupt is not pending.	0
15	Reserved	RO	Reserved	0
14	PENDSTA	RO	14# interrupts currently pending status. 1: Current numbered interrupt is pending; 0: Current numbered interrupt is not pending.	0
13	Reserved	RO	Reserved	0
12	PENDSTA	RO	12# interrupts currently pending status. 1: Current numbered interrupt is pending; 0: Current numbered interrupt is not pending.	0
[11:4]	Reserved	RO	Reserved	0
[3:2]	PENDSTA	RO	2#-3# interrupts currently pending status. 1: Current numbered interrupt is pending; 0: Current numbered interrupt is not pending.	0
[1:0]	Reserved	RO	Reserved	0

9.5.2.5 PFIC Interrupt Pending Status Register 2 (PFIC_IPR2)

Offset address: 0x24

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PENDSTA[63:48]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PENDSTA[47:32]															

Bit	Name	Access	Description	Reset value
[31:0]	PENDSTA	RO	32#-63# interrupts currently pending status. 1: Current numbered interrupt is pending; 0: Current numbered interrupt is not pending.	0

9.5.2.6 PFIC Interrupt Pending Status Register 3 (PFIC_IPR3)

Offset address: 0x28

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											PENDSTA[68:64]				

Bit	Name	Access	Description	Reset value
[31:5]	Reserved	RO	Reserved	0
[4:0]	PENDSTA	RO	64#-68# interrupts currently pending status. 1: Current numbered interrupt is pending; 0: Current numbered interrupt is not pending.	0

9.5.2.7 PFIC Interrupt Priority Threshold Configuration Register (PFIC_ITHRESDR)

Offset address: 0x40

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
Reserved																								THRESHOLD[7:0]							

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved	0
[7:0]	THRESHOLD	RW	Interrupt priority threshold setting value. Interrupt priority values lower than the current set value do not perform interrupt service when hung; a 0 in this register indicates that the threshold register function is invalid. [7:5]: Priority threshold value. [4:0]: Reserved, fixed to 0, write invalid.	0

9.5.2.8 PFIC Interrupt Configuration Register (PFIC_CFGR)

Offset address: 0x48

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

Bit	Name	Access	Description	Reset value
[31:16]	KEYCODE	WO	Corresponding to different target control bits, the corresponding security access identification data needs to be written synchronously in order to be	0

			modified, and the readout data is fixed to 0. KEY1 = 0xFA05; KEY2 = 0xBCAF; KEY3 = 0xBEEF.	
[15:8]	Reserved	RO	Reserved	0
7	RSTSYS	WO	System reset (synchronized write to KEY3). Auto clear 0. Write 1 is valid, write 0 is invalid. *Note: Same function as PFIC_SCTLR register SYSRST bit.*	0
[6:0]	Reserved	RO	Reserved	0

9.5.2.9 PFIC Interrupt Global Status Register (PFIC_GISR)

Offset address: 0x4C

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						GPEND STA	GACT STA	NESTSTA[7:0]							

Bit	Name	Access	Description	Reset value
[31:10]	Reserved	RO	Reserved	0
9	GPENDSTA	RO	Whether an interrupt is currently pending: 1: Yes; 0: No.	0
8	GACTSTA	RO	Whether the interrupt is executed currently: 1: Yes; 0: No.	0
[7:0]	NESTSTA	RO	Current interrupt nesting status, currently supports a maximum of 2 levels of nesting, with a maximum hardware stack depth of 2 levels. 0x03: Level 2 interrupt in progress; 0x01: Level 1 interrupt in progress; 0x00: No interrupt occurred; Other: Impossible situation.	0x00

9.5.2.10 PFIC VTF Interrupt ID Configuration Register (PFIC_VTFIDR)

Offset address: 0x50

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VTFID3								VTFID2							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VTFID1								VTFID0							

Bit	Name	Access	Description	Reset value
[31:24]	VTFID3	RW	Configure VTF interrupt 3 ID	0
[23:16]	VTFID2	RW	Configure VTF interrupt 2 ID	0
[15:8]	VTFID1	RW	Configure VTF interrupt 1 ID	0
[7:0]	VTFID0	RW	Configure VTF interrupt 0 ID	0

9.5.2.11 PFIC VTF Interrupt 0 Address Register (PFIC_VTFADDRR0)

Offset address: 0x60

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR0[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR0[15:1]															VTF0EN

Bit	Name	Access	Description	Reset value
[31:1]	ADDR0	RW	VTF interrupt 0 service program address bit[31:1], bit0 is 0.	0
0	VTF0EN	RW	VTF interrupt 0 enable bit: 1: Enabled VTF interrupt 0 channel; 0: Disabled.	0

9.5.2.12 PFIC VTF Interrupt 1 Address Register (PFIC_VTFADDRR1)

Offset address: 0x64

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR1[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR1[15:1]															VTF1EN

Bit	Name	Access	Description	Reset value
[31:1]	ADDR1	RW	VTF interrupt 1 service program address bit[31:1], bit0 is 0.	0
0	VTF1EN	RW	VTF interrupt 1 enable bit: 1: Enabled VTF interrupt 1 channel; 0: Disabled.	0

9.5.2.13 PFIC VTF Interrupt 2 Address Register (PFIC_VTFADDRR2)

Offset address: 0x68

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR2[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR2[15:1]															VTF2EN

Bit	Name	Access	Description	Reset value
[31:1]	ADDR2	RW	VTF interrupt 2 service program address bit[31:1], bit0 is 0.	0
0	VTF2EN	RW	VTF interrupt 2 enable bit: 1: Enabled VTF interrupt 2 channel; 0: Disabled.	0

9.5.2.14 PFIC VTF Interrupt 3 Address Register (PFIC_VTFADDRR3)

Offset address: 0x6C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR3[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR3[15:1]															VTF3EN

Bit	Name	Access	Description	Reset value
[31:1]	ADDR3	RW	VTF interrupt 3 service program address bit[31:1], bit0 is 0.	0
0	VTF3EN	RW	VTF interrupt 3 enable bit: 1: Enabled VTF interrupt 3 channel; 0: Disabled.	0

9.5.2.15 PFIC Interrupt Enable Set Register 1 (PFIC_IENR1)

Offset address: 0x100

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	INTEN14	Reserved	INTEN12	Reserved											

Bit	Name	Access	Description	Reset value
[31:16]	INTEN	WO	16#-31# interrupt enable control. 1: Current numbered interrupt enable; 0: No effect.	0
15	Reserved	RO	Reserved	0
14	INTEN	WO	14# interrupt enable control. 1: Current numbered interrupt enable; 0: No effect.	0

13	Reserved	RO	Reserved	0
12	INTEN	WO	12# interrupt enable control. 1: Current numbered interrupt enable; 0: No effect.	0
[11:0]	Reserved	RO	Reserved	0

9.5.2.16 PFIC Interrupt Enable Set Register 2 (PFIC_IENR2)

Offset address: 0x104

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INTEN[63:48]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTEN[47:32]															

Bit	Name	Access	Description	Reset value
[31:0]	INTEN	WO	32#-63# interrupt enable control. 1: Current numbered interrupt enable; 0: No effect.	0

9.5.2.17 PFIC Interrupt Enable Set Register 3 (PFIC_IENR3)

Offset address: 0x108

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											INTEN[68:64]				

Bit	Name	Access	Description	Reset value
[31:5]	Reserved	RO	Reserved	0
[4:0]	INTEN	WO	64#-68# interrupt enable control. 1: Current numbered interrupt enable; 0: No effect.	0

9.5.2.18 PFIC Interrupt Enable Clear Register 1 (PFIC_IRER1)

Offset address: 0x180

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INTRST[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	INTRSET14	Reserved	INTRSET12	Reserved											

Bit	Name	Access	Description	Reset value
[31:16]	INTRSET	WO	16#-31# interrupt shutdown control.	0

			1: Current numbered interrupt shutdown; 0: No effect.	
15	Reserved	RO	Reserved	0
14	INTRSET	WO	14# interrupt shutdown control. 1: Current numbered interrupt shutdown; 0: No effect.	0
13	Reserved	RO	Reserved	0
12	INTRSET	WO	12# interrupt shutdown control. 1: Current numbered interrupt shutdown; 0: No effect.	0
[11:0]	Reserved	RO	Reserved	0

9.5.2.19 PFIC Interrupt Enable Clear Register 2 (PFIC_IRER2)

Offset address: 0x184

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INTRSET[63:48]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTRSET[47:32]															

Bit	Name	Access	Description	Reset value
[31:0]	INTRSET	WO	32#-63# interrupt shutdown control. 1: Current numbered interrupt shutdown; 0: No effect.	0

9.5.2.20 PFIC Interrupt Enable Clear Register 3 (PFIC_IRER3)

Offset address: 0x188

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											INTRSET[68:64]				

Bit	Name	Access	Description	Reset value
[31:5]	Reserved	RO	Reserved	0
[4:0]	INTRSET	WO	64#-68# interrupt shutdown control. 1: Current numbered interrupt shutdown; 0: No effect.	0

9.5.2.21 PFIC Interrupt Pending Set Register 1 (PFIC_IPSR1)

Offset address: 0x200

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	PEND SET14	Reserved	PENDSET12	Reserved								PEND SET3	PEND SET2	Reserved	

Bit	Name	Access	Description	Reset value
[31:12]	PENDSET	WO	16#-31# interrupt pending settings. 1: Current numbered interrupt pending; 0: No effect.	0
15	Reserved	RO	Reserved	0
14	PENDSET	WO	14# interrupt pending settings, 13# and 15# reserved. 1: Current numbered interrupt pending; 0: No effect.	0
13	Reserved	RO	Reserved	0
12	PENDSET	WO	12# interrupt pending settings, 13# and 15# reserved. 1: Current numbered interrupt pending; 0: No effect.	0
[11:4]	Reserved	RO	Reserved	0
[3:2]	PENDSET	WO	2#-3# interrupt pending settings. 1: Current numbered interrupt pending; 0: No effect.	0
[1:0]	Reserved	RO	Reserved	0

9.5.2.22 PFIC Interrupt Pending Set Register 2 (PFIC_IPSR2)

Offset address: 0x204

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PENDSET[63:48]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PENDSET[47:32]															

Bit	Name	Access	Description	Reset value
[31:0]	PENDSET	WO	32#-63# interrupt pending settings. 1: Current numbered interrupt pending; 0: No effect.	0

9.5.2.23 PFIC Interrupt Pending Set Register 3 (PFIC_IPSR3)

Offset address: 0x208

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											PENDSET[68:64]				

Bit	Name	Access	Description	Reset value
[31:5]	Reserved	RO	Reserved	0
[4:0]	PENDSET	WO	64#-68# interrupt pending settings. 1: Current numbered interrupt pending; 0: No effect.	0

9.5.2.24 PFIC Interrupt Pending Clear Register 1 (PFIC_IPRR1)

Offset address: 0x280

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	PENDRST14	Reserved	PENDRST12	Reserved								PENDRST3	PENDRST2	Reserved	

Bit	Name	Access	Description	Reset value
[31:16]	PENDRST	WO	16#-31# interrupt pending clear. 1: Current numbered interrupt clears the pending state; 0: No effect.	0
15	Reserved	RO	Reserved	0
14	PENDRST	WO	14# interrupt pending clear. 1: Current numbered interrupt clears the pending state; 0: No effect.	0
13	Reserved	RO	Reserved	0
12	PENDRST	WO	12# interrupt pending clear. 1: Current numbered interrupt clears the pending state; 0: No effect.	0
[11:4]	Reserved	RO	Reserved。	0
[3:2]	PENDRST	WO	2#-3# interrupt pending clear. 1: Current numbered interrupt clears the pending state; 0: No effect.	0
[1:0]	Reserved	RO	Reserved	0

9.5.2.25 PFIC Interrupt Pending Clear Register 2 (PFIC_IPRR2)

Offset address: 0x284

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

PENDRST[63:48]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PENDRST[47:32]															

Bit	Name	Access	Description	Reset value
[31:0]	PENDRST	WO	32#-63# interrupt pending clear. 1: Current numbered interrupt clears the pending state; 0: No effect.	0

9.5.2.26 PFIC Interrupt Pending Clear Register 3 (PFIC_IPRR3)

Offset address: 0x288

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											PENDRST[68:64]				

Bit	Name	Access	Description	Reset value
[31:5]	Reserved	RO	Reserved	0
[4:0]	PENDRST	WO	64#-68# interrupt pending clear. 1: Current numbered interrupt clears the pending state; 0: No effect.	0

9.5.2.27 PFIC Interrupt Activation Register 1 (PFIC_IACTR1)

Offset address: 0x300

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IACTS[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	IACTS14	Reserved	IACTS12	Reserved								IACTS3	IACTS2	Reserved	

Bit	Name	Access	Description	Reset value
[31:16]	IACTS	RO	16#-31# interrupt execution status. 1: Current numbered interrupt is executing; 0: Current numbered interrupt not executed.	0
15	Reserved	RO	Reserved	0
14	IACTS	RO	14# interrupt execution status. 1: Current numbered interrupt is executing; 0: Current numbered interrupt not executed.	0
13	Reserved	RO	Reserved	0

12	IACTS	RO	12# interrupt execution status. 1: Current numbered interrupt is executing; 0: Current numbered interrupt not executed.	0
[11:4]	Reserved	RO	Reserved	0
[3:2]	IACTS	RO	2#-3# interrupt execution status. 1: Current numbered interrupt is executing; 0: Current numbered interrupt not executed.	0
[1:0]	Reserved	RO	Reserved	0

9.5.2.28 PFIC Interrupt Activation Register 2 (PFIC_IACTR2)

Offset address: 0x304

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IACTS[63:48]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IACTS[47:32]															

Bit	Name	Access	Description	Reset value
[31:0]	IACTS	RO	32#-63# interrupt execution status. 1: Current numbered interrupt is executing; 0: Current numbered interrupt not executed.	0

9.5.2.29 PFIC Interrupt Activation Register 3 (PFIC_IACTR3)

Offset address: 0x308

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											IACTS[68:64]				

Bit	Name	Access	Description	Reset value
[31:5]	Reserved	RO	Reserved	0
[4:0]	IACTS	RO	64#-68# interrupt execution status. 1: Current numbered interrupt is executing; 0: Current numbered interrupt not executed.	0

9.5.2.30 PFIC Interrupt Priority Configuration Register (PFIC_IPRIORx) (x=0-17)

Offset address: 0x400 - 0x4FF

The controller supports 256 interrupts (0-255), and the priority of each interrupt is controlled by 8bits.

	31 - 24	23 - 16	15 - 8	7 - 0
IPRIOR17	Reserved	Reserved	Reserved	PRIO_68
IPRIOR16	PRIO_67	PRIO_66	PRIO_65	PRIO_64

…	…	…	…	…
IPRIOR0	PRIO_3	PRIO_2	PRIO_1	PRIO_0

Bit	Name	Access	Description	Reset value
[551:544]	IP_68	RW	Same as IP_0 description.	0
…	…	…	…	…
[31:24]	IP_3	RW	Same as IP_0 description.	0
[23:16]	IP_2	RW	Same as IP_0 description.	0
[15:8]	IP_1	RW	Same as IP_0 description.	0
[7:0]	IP_0	RW	No. 0 Interrupt Priority Configuration: [7:5]: Priority control bits. If no nesting is configured, no preemption bit; If 2 levels of nesting are configured, bit7 is a preemption bit; The smaller the priority value is, the higher the priority is. If interrupts of the same preemption priority level hang at the same time, the interrupt with higher priority is prioritized for execution. [4:0]: Reserved, fixed to 0, write invalid.	0

9.5.2.31 PFIC System Control Register (PFIC_SCTLR)

Offset address: 0xD10

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									LOCKUP	SETEVENT	SEVONPEND	WFITOWFE	SLEEPDEEP	SLEEPONEXIT	Reserved

Bit	Name	Access	Description	Reset value
31	SYSRST	WO	System reset, auto clear 0. Write 1 is valid, write 0 is invalid, same effect as PFIC_CFGR register.	0
[30:5]	Reserved	RO	Reserved	0
6	LOCKUP	RW	Core deadlock enable (core will enter deadlock when executing exceptions and NMI executing instructions) 1: Enable; 0: Disable.	0

5	SETEVENT	WO	Set up events that can wake up the WFE situation.	0
4	SEVONPEND	RW	When an event occurs or interrupts the pending state, the system can be woken up from the WFE instruction, and if the WFE instruction is not executed, the system will be awakened immediately after the next execution of the instruction. 1: Enabled events and all interrupts (including unopened interrupts) can wake up the system. 0: Only enabled events and enabled interrupts can wake up the system.	0
3	WFITOWFE	RW	Execute the WFI instruction as if it were a WFE. 1: Treats subsequent WFI instructions as WFE instructions; 0: No effect.	0
2	SLEEPDEEP	RW	Control the low-power mode of the system: 1：deepsleep 0：sleep	0
1	SLEEPONEXIT	RW	Controls the state of the system after leaving the interrupt service program: 1: The system enters low-power mode; 0: The system enters the main program.	0
0	Reserved	RO	Reserved	0

9.5.3 Dedicated CSR Registers

Some control and status registers (CSR) are defined in the RISC-V architecture to configure or identify or record the running status. CSR registers are internal registers within the core and use a dedicated 12-bit address space. In addition to the standard registers defined in the RISC-V privileged architecture document, the chip also adds some custom registers that need to be accessed by csr instructions.

Note: Such registers marked with the "MRW, MRO, MRW1" attribute need to be accessed by the system in machine mode.

9.5.3.1 Interrupt System Control Register (INTSYSCR)

CSR address: 0x804

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

Bit	Name	Access	Description	Reset value
[31:6]	Reserved	MRO	Reserved	0
5	GIHWSTKNEN	MRW1	Global interrupt and hardware stack off enable. *Note: This bit is often used in real-time operating systems. When the interrupt switches the context, set this bit to turn off global interrupts and hardware stacking out of the stack, and when the context switching is completed and the interrupt returns from execution, the hardware automatically clears this bit.*	0
[4:2]	Reserved	MRO	Reserved	0
1	INESTEN	MRW	Interrupt nesting enable: 0: Interrupt nesting function disabled; 1: Interrupt nesting function enabled.	0
0	HWSTKEN	MRW	Hardware stack enable: 0: Hardware stack function disable; 1: hardware stack function enable.	0

9.5.3.2 Machine Trap-vector Base Address Register (MTVEC)

CSR address: 0x305

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BASEADDR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASEADDR[15:2]														MODE1	MODE0

Bit	Name	Access	Description	Reset value
[31:2]	BASEADDR[31:2]	MRW	Interrupt vector table base address.	0
1	MODE1	MRW	Interrupt vector table recognition mode: 0: Recognized by jump instruction, limited range, supports non-jump instructions; 1: Recognized by absolute address, full range supported, but must jump.	0
0	MODE0	MRW	Interrupt or exception entry address mode selection: 0: Use uniform entry address; 1: Address offset based on interrupt number *4.	0

9.5.4 Physical Memory Protection (PMP)

In order to improve system security, the architecture of RISC-V defines a set of physical address access restrictions, which can set its read, write and execution properties for the physical memory in the area, and protect the area length

at least 4 bytes. The PMP unit always works in user mode and optionally in machine mode. If the current memory limit is violated, it will cause a system EXC.

The PMP unit contains 4 sets of 8-bit configuration registers (32bit) and four sets of address registers, which need to be accessed using csr instructions and are performed in machine mode.

9.5.4.1 PMP Configuration Register (PMPCFG0)

CSR address: 0x3A0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
pmp3cfg								pmp2cfg							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
pmp1cfg								pmp0cfg							

Bit	Name	Access	Description	Reset value
[31:24]	pmp3cfg	MRW	See pmp0cfg.	0
[23:16]	pmp2cfg	MRW	See pmp0cfg.	0
[15:8]	pmp1cfg	MRW	See pmp0cfg.	0
[7:0]	pmp0cfg	MRW	Bit / Name / Description 7 / L / Lock enable, unlockable in machine mode 0: Not locked; 1: Lock the related registers. [6:5] / - / Reserved [4:3] / A / Address alignment and protected area range selection. 2 / X / Executable attributes. 1 / W / Writable attributes. 0 / R / Readable attributes.	0

For address aligned and protection region range selection, it performs memory protection for the region between A_ADDR and B_ADDR (A_ADDR and B_ADDR are both required 4-byte aligned):

1. If B_ADDR – A_ADDR == 2^2, it is based on NA4;
2. If B_ADDR – A_ADDR == $2^{(G+2)}$ and G≥1, and if A_ADDR= $2^{(G+2)}$, it is aligned based on NAPOT;
3. Otherwise it is based on TOR.

A value	Name	Description
00b	OFF	No region to protect.
01b	TOR	Top aligned region protection: Under pmp0cfg, 0 ≤ region < pmpaddr0; Under pmp1cfg, pmpaddr0 ≤ region < pmpaddr1; Under pmp2cfg, pmpaddr1 ≤ region <mpaddr2; Under pmp3cfg, pmpaddr2 ≤ region < pmpaddr3. $pmpaddr_{i-1}$ = A_ADDR >> 2; $pmpaddr_i$ = B_ADDR >> 2.

10b	NA4	Fixed 4-byte region protection. pmp0cfg~mp3cfg corresponds to pmpaddr0~mpaddr3 as the start address. $pmpaddr_i$ = A_ADDR >> 2.
11b	NAPOT	$2^{(G+2)}$ region protection, G≥1, in this case, A_ADDR is aligned on $2^{(G+2)}$. $pmpaddr_i$ = ((A_ADDR |($2^{(G+2)}$-1))& ~(1<<G+1))>> 2.

9.5.4.2 PMP Address 0 Register (PMPADDR0)

CSR address: 0x3B0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR0[33:18]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR0[17:2]															

Bit	Name	Access	Description	Reset value
[31:0]	ADDR0	MRW	PMP sets bit[33:2] of address 0. The actual high 2 bits are not used.	0

9.5.4.3 PMP Address 1 Register (PMPADDR1)

CSR address: 0x3B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR1[33:18]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR1[17:2]															

Bit	Name	Access	Description	Reset value
[31:0]	ADDR1	MRW	PMP set bit[33:2] of address 1, actual high 2 bits unused.	0

9.5.4.4 PMP Address 2 Register (PMPADDR2)

CSR address: 0x3B2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR2[33:18]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR2[17:2]															

Bit	Name	Access	Description	Reset value
[31:0]	ADDR2	MRW	PMP set bit[33:2] of address 2, actual high 2 bits unused.	0

9.5.4.5 PMP Address 3 Register (PMPADDR3)

CSR address: 0x3B3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR3[33:18]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR3[17:2]															

Bit	Name	Access	Description	Reset value
[31:0]	ADDR3	MRW	PMP set bit[33:2] of address 3, actual high 2 bits unused.	0

9.5.5 STK Registers

Table 9-6 STK-related registers

Name	Access address	Description	Reset value
R32_STK_CTLR	0xE000F000	System count control register	0x00000000
R32_STK_SR	0xE000F004	System count status register	0x00000000
R32_STK_CNTL	0xE000F008	System counter low register	0x00000000
R32_STK_CNTH	0xE000F00C	System counter high register	0x00000000
R32_STK_CMPLR	0xE000F010	Count/compare low register	0x00000000
R32_STK_CMPHR	0xE000F014	Count/compare high register	0x00000000

Note: Applied for general-purpose MCUs designed based on 32-bit RISC-V instruction set and architecture.

9.5.5.1 System Count Control Register (STK_CTLR)

Offset address: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWIE	Reserved														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										INIT	MODE	STRE	STCLK	STIE	STE

Bit	Name	Access	Description	Reset value
31	SWIE	RW	Software interrupt trigger enable (SWI): 1: Trigger software interrupt; 0: Turn off the trigger. After entering the software interrupt, software clearing of 0 is required, otherwise the trigger is continuously triggered.	0
[30:6]	Reserved	RO	Reserved	0
5	INIT	W1	Counter initial value update: 1: Updated to 0 for upward counting and to the comparison value for downward counting; 0: Invalid.	0

4	MODE	RW	Counting modes: 1: Counting down; 0: Counting up.	0
3	STRE	RW	Auto reload count enable bit: 1: Count up to the comparison value and then restart counting from 0. After counting down to 0, count down again from the comparison value; 0: Continue counting upward after counting upward to the comparison value, after counting downward to 0, restart counting downward from the maximum value.	0
2	STCLK	RW	Counter clock source select bit: 1: HCLK for time base; 0: HCLK/8 for time base.	0
1	STIE	RW	Counter interrupt enable control bit: 1: Enable counter interrupt; 0: Disable counter interrupt.	0
0	STE	RW	System counter enable control bits: 1: Starts the system counter STK; 0: Turn off the system counter STK and the counter stops counting.	0

9.5.5.2 System Count Status Register (STK_SR)

Offset address: 0x04

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

Bit	Name	Access	Description	Reset value
[31:1]	Reserved	RO	Reserved	0
0	CNTIF	RW0	Count value comparison flag, write 0 to clear, write 1 to invalidate: 1: Count up to reach the comparison value, count down to 0; 0: Comparison value not reached.	0

9.5.5.3 System Counter Low Register (STK_CNTL)

Offset address: 0x08

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

CNT[15:0]

Bit	Name	Access	Description	Reset value
[31:0]	CNT[31:0]	RW	Current counter count low 32-bit.	0

Note: Register STK_CNTL and register STK_CNTH together form the 64-bit system counter.

9.5.5.4 System Counter High Register (STK_CNTH)

Offset address: 0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT[63:48]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[47:32]															

Bit	Name	Access	Description	Reset value
[31:0]	CNT[63:32]	RW	Current counter count high 32-bit.	0

Note: Register STK_CNTL and register STK_CNTH together form the 64-bit system counter.

9.5.5.5 Count/Compare Low Register (STK_CMPLR)

Offset address: 0x10

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

Bit	Name	Access	Description	Reset value
[31:0]	CMP[31:0]	RW	Set the low 32 bits of the compare counter value.	0

Note: Register STK_CMPLR and register STK_CMPHR together form the 64-bit counter comparison value.

9.5.5.6 Count/Compare High Register (STK_CMPHR)

Offset address: 0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMP[63:48]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP[47:32]															

Bit	Name	Access	Description	Reset value
[31:0]	CMP[63:32]	RW	Set the high 32 bits of the compare counter value.	0

Note: Register STK_CMPLR and register STK_CMPHR together form the 64-bit counter comparison value.

Chapter 10 GPIO and Alternate Functions (GPIO/AFIO)

The GPIO ports can be configured for multiple input or output modes, have built-in shutdown pull-up or pull-down resistors, and can be configured for push-pull or open-drain functions. The GPIO ports can also be multiplexed for other functions.

The PA11/PA12 can be multiplexed as USB I/O pins with two sets of mutually exclusive pull-up and pull-down resistors. When RB_UC_RST_SIE=1 as a normal GPIO, the pull-up and pull-down resistors are controlled in the same way and with the same characteristics as other GPIOs. When RB_UC_RST_SIE=0 as USB dedicated pin UDM/UDP, the USB pull-up resistor is about 1.5K, refer to Table 20-2 to realize the control, the USB pull-down resistor is about 15K, which is controlled by RB_UH_PD_DIS in R8_USB_CTRL, neither of them is controlled by GPIO.

PB6/PB7 can be multiplexed as USB PD I/O pins with two sets of mutually independent pull-up and pull-down resistors. Among them, the pull-up and pull-down resistors with the same characteristics as other GPIOs are controlled in the same way as other GPIOs. Another set of pull-up current and pull-down resistors Rd (not built-in for some package forms of the chip) are used for Type-C pins and are controlled by R16_PORT_CC1/R16_PORT_CC2. In addition, the USBPD_IN_HVT bit in AFIO_CR selects the high threshold input mode for PB6/PB7.

10.1 Main Features

Each pin of the port can be configured into one of the following modes:

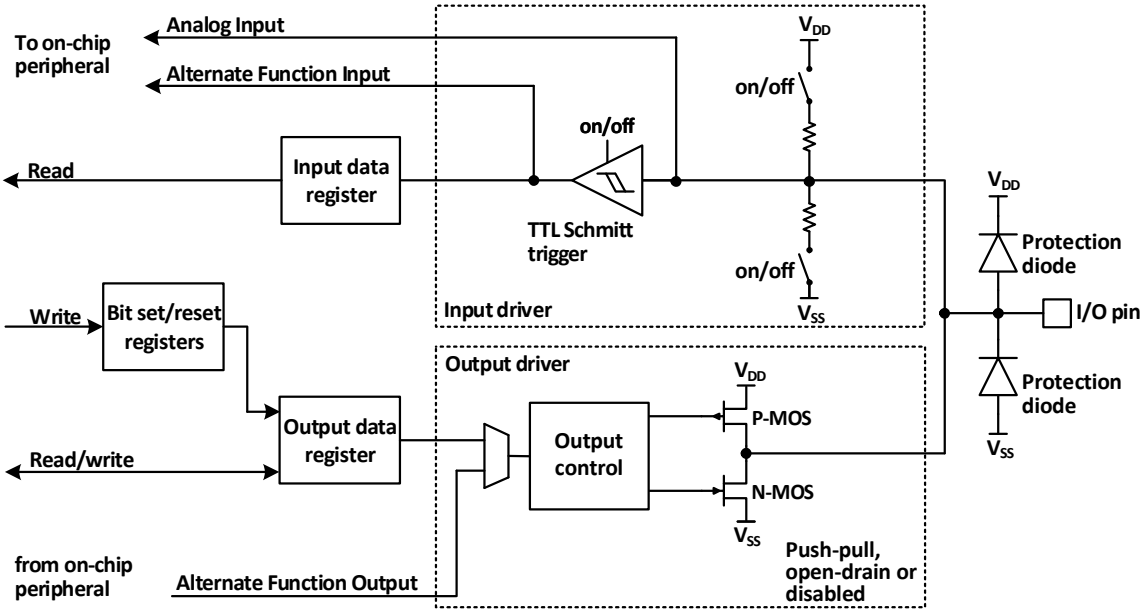
- Floating input
- Pull-up input
- Pull-down input
- Analog input
- Open-drain output
- Push-pull output
- Input and output of alternate function

Many pins have alternate functions, and many other peripherals map their own output and input channels to these pins. The specific application of these alternate pins needs to be with reference to each peripheral, and this chapter shall specify whether these pins are alternate and remapped.

10.2 Function Description

10.2.1 Overview

Figure 10-1 Basic structure of GPIO module



As shown in figure 10-1, the IO port structure, each pin has two protection diodes inside the chip, and the IO port can be divided into input and output drive modules. The input driver has a weak pull-up resistor, which can be connected to AD and other analog input peripherals; if you input to a digital peripheral, you need to go through a TTL Schmitt trigger, and then connect to the GPIO input register or other multiplexed peripherals. The output driver has a pair of MOS tubes, and the IO port can be configured to open leakage or push-pull output by configuring whether the upper and lower MOS tubes are enabled or not; the output driver can also be configured to be controlled by GPIO or by other peripherals that are reused.

10.2.2 GPIO Initialization

Just after the reset, the GPIO port is running in the initial state, at this time, most IO ports are running in the floating input state, but there are also HSE and other peripheral-related pins that run on the peripheral reuse function. For specific initialization functions, please refer to the relevant sections of the pin description.

10.2.3 External Interrupt

All GPIO ports can be configured with external interrupt input channels, but an external interrupt input channel can only be mapped to one GPIO pin, and the sequence number of the external interrupt channel must be consistent with the tag of the GPIO port, for example, PA1 (or PB1, PC1, PD1) can only be mapped to EXTI1, and EXTI1 can only accept the mapping of one of PA1, PB1, PC1 or PD1.

10.2.4 Alternate Function

When using the alternate function, you must pay attention to:

- Using the alternate function of the input direction, the port must be configured in the alternate input mode, and the up and down settings can be set according to the actual needs.

- Using the alternate function in the output direction, the port must be configured in the alternate output mode, and push-pull or open-drain can be set according to the actual situation.
- For two-way alternate, the port must be configured in multiplexed output mode, when the driver is configured in floating input mode.

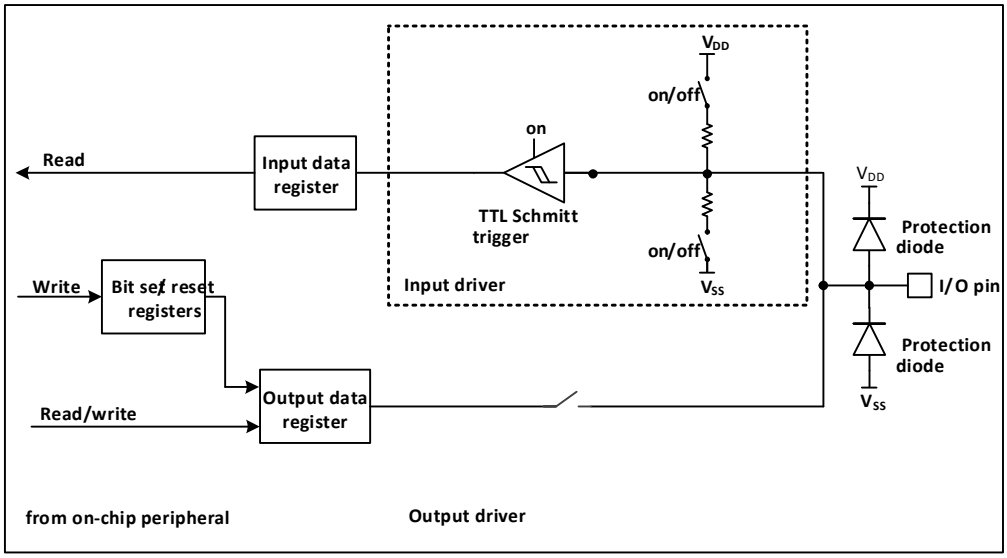
Multiple peripherals may be reused to this pin in the same IO port, so in order to maximize the exertion space of each peripheral, the multiplex pin of the peripheral can be remapped and remapped to other pins to avoid the occupied pins in addition to the default alternate pin.

10.2.5 Locking Mechanism

The locking mechanism can lock the configuration of the IO port. After a specific write sequence, the selected IO pin configuration is locked and cannot be changed until the next reset.

10.2.6 Input Configuration

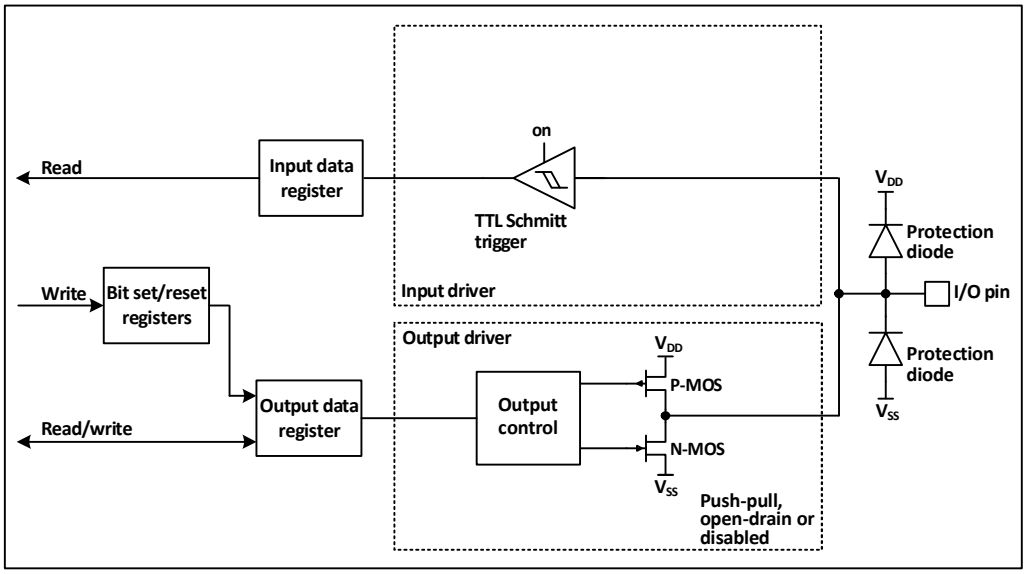
Figure 10-2 Input configuration structure block diagram of GPIO module



When the IO port is configured in input mode, the output driver is disconnected, the input up and down is optional, and the alternate function and analog input are not connected. The data on each IO port is sampled to the input data register at each PB2 clock, and the level state of the corresponding pin is obtained by reading the corresponding bit of the input data register.

10.2.7 Output Configuration

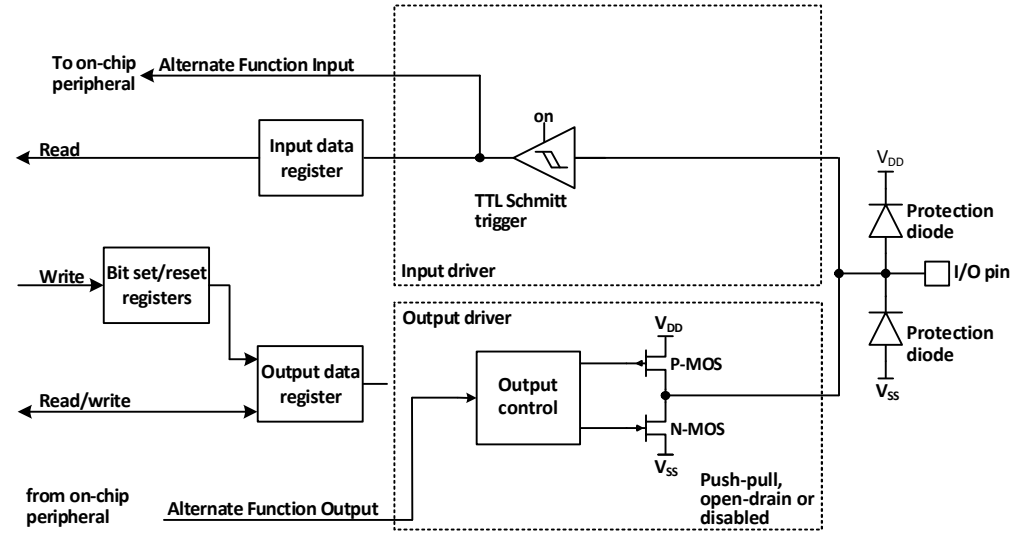
Figure 10-3 Output configuration structure block diagram of GPIO module



When the IO port is configured in output mode, a pair of MOS in the output driver can be configured in push-pull or open-drain mode as needed, without using the alternate function. The input-driven pull-up resistor is disabled, the TTL Schmitt trigger is activated, and the level that appears on the IO pin will be sampled to the input data register at each PB2 clock, so reading the input data register will get the IO state, and in push-pull output mode, access to the output data register will get the last written value.

10.2.8 Alternate Function Configuration

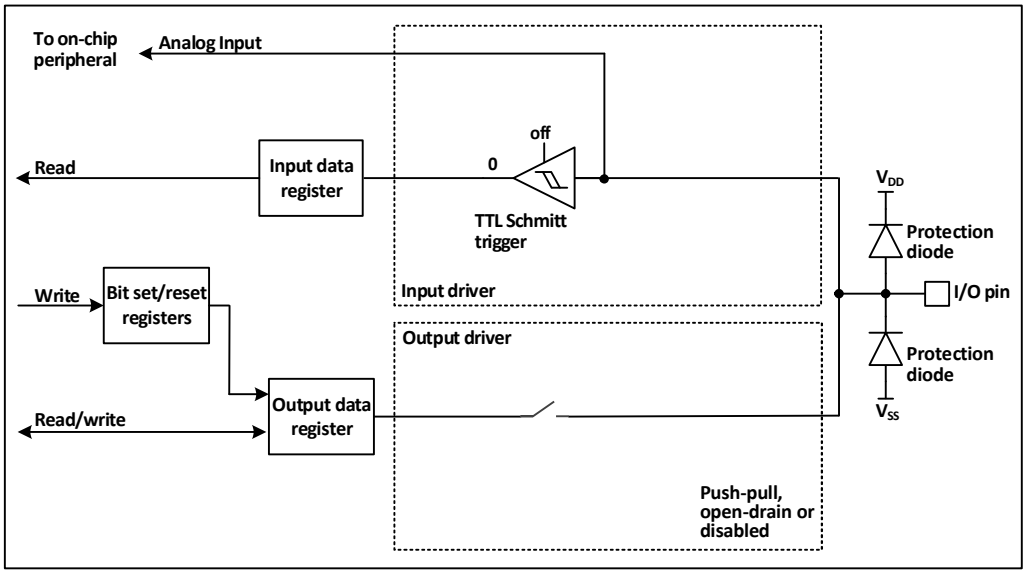
Figure 10-4 Structure block diagram when GPIO module is multiplexed by other peripherals



When the alternate function is enabled, the output driver is enabled and can be configured in open-drain or push-pull mode as needed, the Schmitt trigger is also turned on, the input and output lines of the alternate function are connected, but the output data register is disconnected, and the level that appears on the IO pin will be sampled to the input data register at each PB2 clock. In open-drain mode, reading the input data register will get the current state of the IO port. In push-pull mode, reading the output data register will get the last written value.

10.2.9 Analog Input Configuration

Figure 10-5 Configuration structure block diagram of GPIO module as analog input



When the analog input is enabled, the output buffer is disconnected, the input of the Schmitt trigger in the input driver is disabled to prevent consumption on the IO port, the pull-up resistor is prohibited, and the read input data register will always be 0.

10.2.10 Peripheral GPIO Setting

The following table recommends the corresponding GPIO port configuration of each peripheral pin.

Table 10-1 Advanced-control timer (TIM1)

TIM1	Configuration	GPIO configuration
TIM1_CHx	Input capture channel x	Floating input
	Output compare channel x	Push-pull alternate output
TIM1_CHxN	Complementary output channel x	Push-pull alternate output
TIM1_BKIN	Break input	Floating input
TIM1_ETR	External trigger clock input	Floating input

Table 10-2 General-purpose timer (TIM2/3/4)

TIM2/3/4 pin	Configuration	GPIO configuration
TIM2/3/4_CHx	Input capture channel x	Floating input
	Output compare channel x	Push-pull alternate output
TIM2/3/4_ETR	External trigger clock input	Floating input

Table 10-3 Universal Synchronous/Asynchronous Receiver Transmitter (USART)

USART pin	Configuration	GPIO configuration
USARTx_TX	Full-duplex mode	Push-pull alternate output
	Half-duplex synchronous mode	Open-drain alternate output
USARTx_RX	Full-duplex mode	Floating input or pull-up input
	Half-duplex synchronous mode	Not used
USARTx_CK	Synchronous mode	Push-pull alternate output

USARTx_RTS	Hardware flow control	Push-pull alternate output
USARTx_CTS	Hardware flow control	Floating input or pull-up input

Table 10-4 Serial Peripheral Interface (SPI) module

SPI pin	Configuration	GPIO configuration
SPIx_SCK	Master mode	Push-pull alternate output
	Slave mode	Floating input
SPIx_MOSI	Full-duplex master mode	Push-pull alternate output
	Full-duplex slave mode	Floating input or pull-up input
	Simplex bidirectional data line/master mode	Push-pull alternate output
	Simplex bidirectional data line/slave mode	Not used
SPIx_MISO	Full duplex master mode	Floating input or pull-up input
	Full duplex slave mode	Push-pull alternate output
	Simplex bidirectional data line/master mode	Not used
	Simplex bidirectional data line/slave mode	Push-pull alternate output
SPIx_NSS	Hardware master or slave mode	Floating input or pull-up or pull-down input
	Hardware master mode/NSS output enable	Push-pull alternate output
	Software mode	Not used

Table 10-5 Inter IC Sound (I2C) module

I2C pin	Configuration	GPIO configuration
I2C _SCL	I2C clock	Open-drain alternate output
I2C _SDA	I2C data	Open-drain alternate output

Table 10-6 Controller LAN (CAN) module

CAN pin	GPIO configuration
CANx_TX	Push-pull alternate output
CANx_RX	Floating input or pull-up input

Table 10-7 USB Host Device (USBFS) controller

USBFS pin	GPIO configuration
USBFS_DM/USBFS_DP	After the USB is enabled, alternate IO port connects to the internal USBFS transceiver automatically

Table 10-8 Analog-to-Digital Converter (ADC)

ADC pinout	GPIO configuration
ADC	Analog input

Table 10-9 Other IO function settings

Pinout	Configuration function	GPIO configuration
TAMPER_RTC	RTC output	Hardware automatic setting
	Tamper event input	
MCO	Clock output	Push-pull alternate output
EXTI	External interrupt input	Floating input or pull-up or pull-down input

Table 10-10 USB PD/Type-C controller

USBPD pin	GPIO configuration
CC1/CC2	After the USB is enabled, alternate IO port connects to the internal PD transceiver automatically

10.2.11 Alternate Function Remapping GPIO Settings

10.2.11.1 OSC32_IN/OSC32_OUT as GPIO Port PC14/PC15

When LSEON=0, the LSE oscillator pin OSC32_IN/OSC32_OUT can be used as PC14/PC15 of GPIO respectively.

When LSEON=1, it is used as LSE pin.

Note: Power-on defaults to normal IO function

10.2.11.2 OSC_IN/OSC_OUT as GPIO port PD0/PD1

OSC_IN/OSC_OUT can be used as GPIO PD0/PD1, which is realized by setting Remap Register 1 (AFIO_PCFR1).

10.2.11.3 Timer Alternate Function Remapping

Table 10-11 TIM1 alternate function remapping

Alternate function	000	001	010	011	100	101
TIM1_ETR	PA12	PA12	PB6	PA3	PB6	PA13
TIM1_CH1	PA8	PA8	PA1	PA1	PB7	PB1
TIM1_CH2	PA9	PA9	PA7	PA7	PB8	PB8
TIM1_CH3	PA10	PA10	PA14	PA14	PB12	PB6
TIM1_CH4	PA11	PA11	PB1	PB1	PA2	PA3
TIM1_BKIN	PB12	PA6	PA13	PA13	PA12	PA12
TIM1_CH1N	PB13	PA7	PB11	PB11	PA14	PA14
TIM1_CH2N	PB14	PB0	PB0	PB0	PB1	PA1
TIM1_CH3N	PB15	PB1	PB9	PB9	PB9	PB7

Table 10-12 TIM2 alternate function remapping

Alternate function	000	001	010	011	100	101	111
TIM2_ETR	PA0	PA15	PA0	PA15	PA3	PA12	PA12
TIM2_CH1	PA0	PA15	PA0	PA15	PA3	PA12	PA12
TIM2_CH2	PA1	PB3	PA1	PB3	PA2	PA2	PB8

TIM2_CH3	PA2	PA2	PB10	PB10	PB12	PB12	PA5
TIM2_CH4	PA3	PA3	PB11	PB11	PA6	PA6	PA4

Table 10-13 TIM3 alternate function remapping

Alternate function	0	1
TIM3_CH1	PA6	PB4
TIM3_CH2	PA7	PB5
TIM3_CH3	PB0	PB0
TIM3_CH4	PB1	PB1

Table 10-14 TIM4 alternate function remapping

Alternate function	0	1
TIM4_CH1	PB6	PB10
TIM4_CH2	PB7	PB11
TIM4_CH3	PB8	PB8
TIM4_CH4	PB9	PB9

Table 10-15 LPTIM alternate function remapping

Alternate function	0	1
LPT_IN1	PB12	PB5
LPT_IN2	PB13	PB7
LPT_ETR	PB14	PB6
LPT_OUT	PB15	PB2

10.2.11.4 USART Alternate Function Remapping

Table 10-16 USART1 alternate function remapping

Alternate function	000	001	010	011	100	101
USART1_TX	PA9	PB6	PA4	PA5	PB11	PB12
USART1_RX	PA10	PB7	PA5	PA4	PB9	PA12
USART1_CK	PA8	PA8	PA3	PA6	PA6	PB6
USART1_CTS	PA11	PA11	PA2	PB7	PA14	PB7
USART1_RTS	PA12	PA12	PA13	PB8	PA13	PB8

Table 10-17 USART2 alternate function remapping

Alternate function	00	10	11
USART2_TX	PA2	PA11	PA12
USART2_RX	PA3	PA12	PA11
USART2_CK	PA4	PA4	PA4
USART2_CTS	PA0	PA0	PA0
USART2_RTS	PA1	PA1	PA1

Table 10-18 USART3 alternate function remapping

Alternate function	00	10	11
USART3_TX	PB10	PD1	PD0
USART3_RX	PB11	PD0	PD1
USART3_CK	PB12	PB12	PB12
USART3_CTS	PB13	PB13	PB13
USART3_RTS	PB14	PB14	PB14

Table 10-19 USART4 alternate function remapping

Alternate function	0	1
USART4_TX	PB0	PA5
USART4_RX	PB1	PB5
USART4_CK	PB2	PA6
USART4_CTS	PB3	PA7
USART4_RTS	PB4	PA15

10.2.11.5 SPI Alternate Function Remapping

Table 10-20 SPI1 alternate function remapping

Alternate function	00	01	10	11
SPI1_NSS	PA4	PA15	PA12	PB12
SPI1_SCK	PA5	PB3	PB6	PB6
SPI1_MISO	PA6	PB4	PB8	PB8
SPI1_MOSI	PA7	PB5	PB7	PB7

10.2.11.6 I2C Alternate Function Remapping

Table 10-21 I2C1 alternate function remapping

Alternate function	00	10	11
I2C1_SCL	PB6	PA13	PB9
I2C1_SDA	PB7	PA12	PB11

10.2.11.7 CAN Alternate Function Remapping

Table 10-22 CAN alternate function remapping

Alternate function	00	10	11
TX	PA12	PB9	PD1
RX	PA11	PB8	PD0

10.2.12 BC Function Configuration

The AFIO_CR register includes the control bits for the UDM/UDP pins of the BC interface, the comparator and the

voltage output.

Setting RB_UC_RST_SIE=0 turns on USB, setting RB_UH_PD_DIS=0 in R8_UHOST_CTRL enables the internal 15K pull-down, and setting BC_VSRC=1 for UDM/UDP enables the BC protocol source voltage VBC_SRC output.

Setting UDM/UDP's BC_CMPE=1 enables the BC protocol comparator; a read operation of UDM/UDP's BC_CMPO bit acquires the state of the pin voltage compared with the BC protocol reference VBC_REF.

The AFIO_CR register also includes the USBPD_IN_HVT bit, which sets the PD pin PB6/PB7 high threshold input mode. This bit is a high threshold input at position 1, with a typical value of about 2.2V, which reduces I/O power consumption during PD communication; this bit is a normal GPIO threshold input at position 0.

10.3 Register Description

10.3.1 GPIO Registers

Unless otherwise specified, the GPIO registers must be operated in words (operate these registers in 32 bits).

Table 10-23 GPIO-related registers

Name	Access address	Description	Reset value
R32_GPIOA_CFGLR	0x40010800	PA port configuration register low	0x44444444
R32_GPIOB_CFGLR	0x40010C00	PB port configuration register low	0x44444444
R32_GPIOC_CFGLR	0x40011000	PC port configuration register low	0x44444444
R32_GPIOD_CFGLR	0x40011400	PD port configuration register low	0x44444444
R32_GPIOA_CFGHR	0x40010804	PA port configuration register high	0x44444444
R32_GPIOB_CFGHR	0x40010C04	PB port configuration register high	0x44444444
R32_GPIOC_CFGHR	0x40011004	PC port configuration register high	0x44444444
R32_GPIOD_CFGHR	0x40011404	PD port configuration register high	0x44444444
R32_GPIOA_INDR	0x40010808	PA port input data register	0x0000XXXX
R32_GPIOB_INDR	0x40010C08	PB port input data register	0x0000XXXX
R32_GPIOC_INDR	0x40011008	PC port input data register	0x0000XXXX
R32_GPIOD_INDR	0x40011408	PD port input data register	0x0000XXXX
R32_GPIOA_OUTDR	0x4001080C	PA port output data register	0x00000000
R32_GPIOB_OUTDR	0x40010C0C	PB port output data register	0x00000000
R32_GPIOC_OUTDR	0x4001100C	PC port output data register	0x00000000
R32_GPIOD_OUTDR	0x4001140C	PD port output data register	0x00000000
R32_GPIOA_BSHR	0x40010810	PA port set/reset register	0x00000000
R32_GPIOB_BSHR	0x40010C10	PB port set/reset register	0x00000000
R32_GPIOC_BSHR	0x40011010	PC port set/reset register	0x00000000
R32_GPIOD_BSHR	0x40011410	PD port set/reset register	0x00000000
R32_GPIOA_BCR	0x40010814	PA port reset register	0x00000000
R32_GPIOB_BCR	0x40010C14	PB port reset register	0x00000000
R32_GPIOC_BCR	0x40011014	PC port reset register	0x00000000
R32_GPIOD_BCR	0x40011414	PD port reset register	0x00000000
R32_GPIOA_LCKR	0x40010818	PA port lock configuration register	0x00000000

R32_GPIOB_LCKR	0x40010C18	PB port lock configuration register	0x00000000
R32_GPIOC_LCKR	0x40011018	PC port lock configuration register	0x00000000
R32_GPIOD_LCKR	0x40011418	PD port lock configuration register	0x00000000

10.3.1.1 GPIO Configuration Register Low (GPIOx_CFGLR) (x=A/B/C/D)

Offset address: 0x00

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
CNF7[1:0]	MODE7[1:0]	CNF6[1:0]	MODE6[1:0]	CNF5[1:0]	MODE5[1:0]	CNF4[1:0]	MODE4[1:0]
15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
CNF3[1:0]	MODE3[1:0]	CNF2[1:0]	MODE2[1:0]	CNF1[1:0]	MODE1[1:0]	CNF0[1:0]	MODE0[1:0]

Bit	Name	Access	Description	Reset value
[31:30] [27:26] [23:22] [19:18] [15:14] [11:10] [7:6] [3:2]	CNFy[1:0]	RW	(y=0-7), the configuration bits of port x, through which the corresponding ports are configured. When entering a mode (MODE=00b): 00: Analog input mode. 01: Floating input mode. 10: With pull-up mode. 11: Keep. In output mode (MODE > 00b): 00: Universal push-pull output mode. 01: Universal open-leak output mode. 10: Alternate function push-pull output mode. 11: Alternate function open-leak output mode.	01b
[29:28] [25:24] [21:20] [17:16] [13:12] [9:8] [5:4] [1:0]	MODEy[1:0]	RW	(y=0-7), port x mode selection, configure the corresponding port through these bits. 00: Input mode. 01: Output mode, maximum speed 10MHz. 10: Output mode, maximum speed 2MHz. 11: Output mode, maximum speed 50MHz.	00b

10.3.1.2 GPIO Configuration Register High (GPIOx_CFGHR) (x=A/B/C/D)

Offset address: 0x04

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
CNF15[1:0]	MODE15[1:0]	CNF14[1:0]	MODE14[1:0]	CNF13[1:0]	MODE13[1:0]	CNF12[1:0]	MODE12[1:0]
15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
CNF11[1:0]	MODE11[1:0]	CNF10[1:0]	MODE10[1:0]	CNF9[1:0]	MODE9[1:0]	CNF8[1:0]	MODE8[1:0]

Bit	Name	Access	Description	Reset value
[31:30] [27:26] [23:22] [19:18] [15:14] [11:10] [7:6] [3:2]	CNFy[1:0]	RW	(y=8-15), the configuration bits for port x, by which the corresponding port is configured. When in input mode (MODE=00b): 00: Analog input mode; 01: Float input mode; 10: With pull-up and pull-down modes. 11: Reserved. In output mode (MODE>00b): 00: General-purpose push-pull output mode; 01: General-purpose open-drain output mode; 10: Alternate function push-pull output mode; 11: Alternate function open-drain output mode.	01b
[29:28] [25:24] [21:20] [17:16] [13:12] [9:8] [5:4] [1:0]	MODEy[1:0]	RW	(y=8-15), the mode bits of port x, by which the corresponding port is configured. 00: Input mode; 01: Output mode, max. speed 10MHz. 10: Output mode, maximum speed 2MHz; 11: Output mode, maximum speed 50MHz.	00b

10.3.1.3 Port Input Register (GPIOx_INDR) (x=A/B/C/D)

Offset address: 0x08

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

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:0]	IDRy	RO	(y=0-15), port input data. These bits are read-only and can only be read out in 16-bit form. The value read out is the high and low state of the corresponding bit.	X

10.3.1.4 Port Output Register (GPIOx_OUTDR) (x=A/B/C/D)

Offset address: 0x0C

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

ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:0]	ODRy	RW	For output mode: (y=0-15). the port outputs data. This data can only be manipulated in 16-bit form. the IO port externally outputs the values of these registers. For input modes with pull-down: 0: Pull-down input; 1: Pull-up input.	0

10.3.1.5 Port Set/Reset Register (GPIOx_BSHR) (x=A/B/C/D)

Offset address: 0x10

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
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

Bit	Name	Access	Description	Reset value
[31:16]	BRy	WO	(y=0-15), the corresponding OUTDR bits are cleared for these location bits; writing 0 has no effect. These bits can only be accessed in 16-bit format. If both BR and BS bits are set, the BS bit acts.	0
[15:0]	BSy	WO	(y=0-15), for which positional bits cause the corresponding OUTDR positional bits, and writing 0 has no effect. These bits can only be accessed in 16-bit format. If both BR and BS bits are set, the BS bit acts.	0

10.3.1.6 Port Reset Register (GPIOx_BCR) (x=A/B/C/D)

Offset address: 0x14

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

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:0]	BRy	WO	(y=0-15), the corresponding OUTDR bits are cleared for these location bits; writing 0 has no	0

			effect. These bits can only be accessed in 16-bit form.	

10.3.1.7 Configuration Lock Register (GPIOx_LCKR) (x=A/B/C/D)

Offset address: 0x18

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

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	LCKK	RW	Lock key, which can be locked by writing a specific sequence, but it can be read out at any time. When it reads 0, it means that the lock is not in effect, and read 1 means that the lock is in effect. The write sequence of the lock key is: write 1-write 0-write 1-read 0-read 1, the last step is not necessary, but can be used to confirm that the lock key has been activated. Any error when writing a sequence does not activate a lock, and the value of LCK [15:0] cannot be changed when writing a sequence. After the lock is in effect, the configuration of the port can be changed only after the next reset.	0
[15:0]	LCKy	RW	(y=0-15), where 1 indicates that the configuration of the corresponding port is locked. These bits can only be changed before the LCKK is locked. Locked configurations refer to the configuration registers GPIOx_CFGLR and GPIOx_CFGHR.	0

Note: After the LOCK sequence is performed on the corresponding port bit, the configuration of the port bit cannot be changed until the next system reset.

10.3.2 AFIO Registers

Unless otherwise specified, the AFIO registers must be operated in words (operate these registers in 32 bits).

Table 10-24 AFIO-related registers

Name	Access address	Description	Reset value
R32_AFIO_ECR	0x40010000	Event control register	0x00000000
R32_AFIO_PCFR1	0x40010004	Remap register1	0x00000000
R32_AFIO_EXTICR1	0x40010008	External interrupt configuration register 1	0x00000000
R32_AFIO_EXTICR2	0x4001000C	External interrupt configuration register 2	0x00000000
R32_AFIO_EXTICR3	0x40010010	External interrupt configuration register 3	0x00000000

R32_AFIO_EXTICR4	0x40010014	External interrupt configuration register 4	0x00000000
R32_AFIO_CR	0x40010018	Control register	0x00000000
R32_AFIO_PCFR2	0x4001001C	Remap register 2	0x00000000

10.3.2.1 Event Control Register (AFIO_ECR)

Offset address: 0x00

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								EVOE	PORT[2:0]			PIN[3:0]			

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved	0
7	EVOE	RW	Allow event output bit, for which the location bit causes the core's EVENTOUT to connect to the IO port selected by PORT and PIN.	0
[6:4]	PORT[2:0]	RW	Used to select the port from which the core outputs EVENTOUT: 000: Selects the PA port; 001: Selects the PB port; 010: Selects the PC port; 011: Selects the PD port; Other: Reserved.	000b
[3:0]	PIN[3:0]	RW	The value of this bit is used to determine the specific pin number that selects the core output EVENTOUT to the port, with values 0-15 corresponding to pins 0-15 of the selected Px in the PORT, respectively.	0

10.3.2.2 Remap Register 1 (AFIO_PCFR1)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					SW_CFG[2:0]			Reserved							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PD0PD1_RM	CAN_RM[1:0]		TIM4_RM	Reserved	TIM3_RM	TIM2_RM][1:0]		TIM1_RM][1:0]		USART3_RM[1:0]		USART2_RM	USART1_RM	I2C1_RM	SPI1_RM

Bit	Name	Access	Description	Reset value
[31:27]	Reserved	RO	Reserved	0
[26:24]	SW_CFG[2:0]	WO	These bits are used to configure the IO ports for the SW function and the trace function. SDI is the debugging interface for accessing the core. It is always used as the SDI port after a system reset.	000b

			0xx: Enable SDI; 100: Disable SDI as a GPIO function; Other: Invalid.	
[23:16]	Reserved	RO	Reserved	0
15	PD0PD1_RM	RW	Pin PD0&PD1 remap bit, which can be read or written by the user. It controls whether the GPIO functions of PD0 & PD1 are remapped, i.e. PD0 & PD1 are mapped to OSC_IN & OSC_OUT. 0: Pin is used as crystal pin; 1: Pin is used as GPIO port.	0
[14:13]	CAN_RM[1:0]	RW	CAN alternate function remapping bits which can be read or written by the user. Controls the remapping of CAN_RX and CAN_TX: 00: CAN_RX is mapped to PA11 and CAN_TX is mapped to PA12; 10: CAN_RX mapped to PB8, CAN_TX mapped to PB9; 01: Reserved; 11: CAN_RX mapped to PD0, CAN_TX mapped to PD1.	00b
12	TIM4_RM	RW	The remap bit for Timer 4, which can be read or written by the user. It controls the remapping of channels 1 through 4 of Timer 4 on the GPIO port: 0: Default mapping (CH1/PB6, CH2/PB7, CH3/PB8, CH4/PB9); 1: Remapping (CH1/PB10, CH2/PB11, CH3/PB8, CH4/PB9).	0
11	Reserved	RO	Reserved	0
10	TIM3_RM	RW	The remap bit for Timer 3, which can be read or written by the user. It controls the remapping of channels 1 through 4 of Timer 3 on the GPIO port: 0: Default mapping (CH1/PA6, CH2/PA7, CH3/PB0, CH4/PB1); 1: Remapping (CH1/PB4，CH2/PB5，CH3/PB0，CH4/PB1).	0
[9:8]	TIM2_RM[1:0]	RW	The remap bits for Timer 2 are used in conjunction with the TIM2_RM_H field of Remap Register 2, {TIM2_RM_H, TIM2_RM}. These bits can be read and written by the user. It controls the mapping of channels 1 through 4 of Timer2 and external triggering (ETR) on the GPIO port: 000: Default mapping (CH1/ETR/PA0, CH2/PA1, CH3/PA2, CH4/PA3); 001: Partial mapping (CH1/ETR/PA15, CH2/PB3,	00b

			CH3/PA2, CH4/PA3); 010: Partial mapping (CH1/ETR/PA0, CH2/PA1, CH3/PB10, CH4/PB11); 011: Full mapping (CH1/ETR/PA15, CH2/PB3, CH3/PB10, CH4/PB11); 100: Complete mapping (CH1/ETR/PA3, CH2/PA2, CH3/PB12, CH4/PA6); 101: Complete mapping (CH1/ETR/PA12, CH2/PA2, CH3/PB12, CH4/PA6); 111: Complete mapping (CH1/ETR/PA12, CH2/PB8, CH3/PA5, CH4/PA4).	
[7:6]	TIM1_RM[1:0]	RW	The remapped bits of Timer 1 are used in conjunction with the TIM1_RM_H field of Remap Register 2, {TIM1_RM_H, TIM1_RM}. These bits can be read and written by the user. It controls the mapping of channels 1 through 4, 1N through 3N, External Trigger (ETR), and Brake Input (BKIN) of Timer 1 on the GPIO port: 000: Default mapping (ETR/PA12, CH1/PA8, CH2/PA9, CH3/PA10, CH4/PA11, BKIN/PB12, CH1N/PB13, CH2N/PB14, CH3N/PB15) 001: Partial mapping (ETR/PA12, CH1/PA8, CH2/PA9, CH3/PA10, CH4/PA11, BKIN/PA6, CH1N/PA7, CH2N/PB0, CH3N/PB1) 010: Full mapping (ETR/PB6, CH1/PA1, CH2/PA7, CH3/PA14, CH4/PB1, BKIN/PA13, CH1N/PB11, CH2N/PB0, CH3N/PB9) 011: Full mapped (ETR/PA3, CH1/PA1, CH2/PA7, CH3/PA14, CH4/PB1, BKIN/PA13, CH1N/PB11, CH2N/PB0, CH3N/PB9) 100: Full mapped (ETR/PB6, CH1/PB7, CH2/PB8, CH3/PB12, CH4/PA2, BKIN/PA12, CH1N/PA14, CH2N/PB1, CH3N/PB9) 101: Full mapped (ETR/PA13, CH1/PB1, CH2/PB8, CH3/PB6, CH4/PA3, BKIN/PA12, CH1N/PA14, CH2N/PA1, CH3N/PB7) 111: Controls LSI inputs only for LSI calibration	00b
[5:4]	USART3_RM[1:0]	RW	Remapping bits for USART3, these bits can be read and written by the user. It controls the mapping of the TX, RX, CK, CTS, and RTS alternate functions of the USART3 to the GPIO ports: 00: Default mapping (TX/PB10, RX/PB11, CK/PB12, CTS/PB13, RTS/PB14)	00b

			10: Partial remapping (TX/PD1, RX/PD0, CK/PB12, CTS/PB13, RTS/PB14) 11: Partial remapping (TX/PD0, RX/PD1, CK/PB12, CTS/PB13, RTS/PB14)	
3	USART2_RM	RW	The remap bit of USART2, combined with the USART2_RM_H field of remap register 2, {USART2_RM_H,USART2_RM}. This bit can be read and written by the user. It controls the mapping of the TX, RX, CK, CTS, and RTS multiplexing functions of USART2 to the GPIO ports: 00: Default mapping (TX/PA2, RX/PA3, CK/PA4, CTS/PA0, RTS/PA1) 01: Reserved 10: Partial mapping (TX/PA11, RX/PA12, CK/PA4, CTS/PA0, RTS/PA1) 11: Partial mapping (TX/PA12, RX/PA11, CK/PA4, CTS/PA0, RTS/PA1)	0
2	USART1_RM	RW	The remap bit of USART1, combined with the USART1_RM_H field of Remap Register 2, {USART1_RM_H,USART1_RM}. This bit can be read and written by the user. It controls the mapping of the TX, RX, CK, CTS and RTS multiplexing functions of USART1 to the GPIO ports: 000: Default mapping (TX/PA9, RX/PA10, CK/PA8, CTS/PA11, RTS/PA12) 001: Partial mapping (TX/PB6, RX/PB7, CK/PA8, CTS/PA11, RTS/PA12) 010: Full mapping (TX/PA4, RX/PA5, CK/PA3, CTS/PA2, RTS/PA13) 011: Full mapped (TX/PA5, RX/PA4, CK/PA6, CTS/PB7, RTS/PB8) 100: Full mapped (TX/PB11, RX/PB9, CK/PA6, CTS/PA14, RTS/PA13) 101: Full mapped (TX/PB12, RX/PA12, CK/PB6, CTS/PB7, RTS/PB8)	0
1	I2C1_RM	RW	The remapping of I2C1 is combined with the I2C1_RM_H field of Remap Register 2, {I2C1_RM_H, I2C1_RM}. This bit can be read or written by the user. It controls the mapping of I2C1's SCL and SDA multiplexing functions on the GPIO port: 00: Default mapping (SCL/PB6, SDA/PB7)	0

			10: Full mapping (SCL/PA13, SDA/PA12) 11: Full mapping (SCL/PB9, SDA/PB11)	
0	SPI1_RM	RW	The remap of SPI1 is combined with the SPI1_RM_H field of remap register 2, {SPI1_RM_H, SPI1_RM}. This bit can be read or written by the user. It controls the mapping of SPI1's NSS, SCK, MISO, and MOSI multiplexing functions on the GPIO port: 00: Default mapping (NSS/PA4, SCK/PA5, MISO/PA6, MOSI/PA7) 01: Full mapping (NSS/PA15, SCK/PB3, MISO/PB4, MOSI/PB5) 10: Full mapped (NSS/PA12, SCK/PB6, MISO/PB8, MOSI/PB7) 11: Full mapped (NSS/PB12, SCK/PB6, MISO/PB8, MOSI/PB7)	0

10.3.2.3 External Interrupt Configuration Register 1 (AFIO_EXTICR1)

Offset address: 0x08

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

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:12] [11:8] [7:4] [3:0]	EXTIx[3:0]	RW	(x=0-3), external interrupt input pin configuration bit. Used to determine which pin the external interrupt pin is mapped to: 0000: xth pin of the PA pin; 0001: xth pin of the PB pin; 0010: xth pin of the PC pin; 0011: xth pin of the PD pin; Other: Reserved.	0000b

10.3.2.4 External Interrupt Configuration Register 2 (AFIO_EXTICR2)

Offset address: 0x0C

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

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:12] [11:8] [7:4] [3:0]	EXTIx[3:0]	RW	(x=4-7), external interrupt input pin configuration bit. Used to determine which port pin the external interrupt pin is mapped to: 0000: xth pin of the PA pin; 0001: xth pin of the PB pin; 0010: xth pin of the PC pin; 0011: xth pin of the PD pin; Other: Reserved.	0000b

10.3.2.5 External Interrupt Configuration Register 3 (AFIO_EXTICR3)

Offset address: 0x10

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

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:12] [11:8] [7:4] [3:0]	EXTIx[3:0]	RW	(x=8-11), external interrupt input pin configuration bit. Used to determine which port pin the external interrupt pin is mapped to: 0000: xth pin of the PA pin; 0001: xth pin of the PB pin; 0010: xth pin of the PC pin; 0011: xth pin of the PD pin; Other: Reserved.	0000b

10.3.2.6 External Interrupt Configuration Register 4 (AFIO_EXTICR4)

Offset address: 0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI15[3:0]				EXTI14[3:0]				EXTI13[3:0]				EXTI12[3:0]			

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:12] [11:8] [7:4] [3:0]	EXTIx[3:0]	RW	(x=12-15), external interrupt input pin configuration bit. Used to determine which port pin the external interrupt pin is mapped to: 0000: xth pin of the PA pin;	0000b

			0001: xth pin of the PB pin; 0010: xth pin of the PC pin; 0011: xth pin of the PD pin; Other: Reserved.	

10.3.2.7 Control Register (AFIO_CR)

Offset address: 0x18

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										UDM_BC_CMPO	UDP_BC_CMPO	UDM_BC_CMPE	UDP_BC_CMPE	UDM_BC_VSRC	UDP_BC_VSRC

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

Bit	Name	Access	Description	Reset value
[31:22]	Reserved	RO	Reserved	0
21	UDM_BC_CMPO	RO	PA11/UDM pin BC protocol comparator status: 1: PA11 voltage is above the BC protocol reference value VBC_REF; 0: PA11 voltage is below the BC protocol reference value VBC_REF.	0
20	UDP_BC_CMPO	RO	PA12/UDP pin BC protocol comparator status: 1: PA12 voltage is above the BC protocol reference value VBC_REF; 0: PA12 voltage is below the BC protocol reference value VBC_REF.	0
19	UDM_BC_CMPE	RW	PA11/UDM pin BC protocol comparator enable: 1: Enable; 0: Disable.	0
18	UDP_BC_CMPE	RW	PA12/UDP pin BC protocol comparator enable: 1: Enable; 0: Disable.	0
17	UDM_BC_VSRC	RW	PA11/UDM pin BC protocol source voltage enable: 1: PA11 outputs BC protocol source voltage V_{BC_SRC}; 0: Output disabled.	0
16	UDP_BC_VSRC	RW	PA12/UDP pin BC protocol source voltage enable: 1: PA12 outputs BC protocol source voltage V_{BC_SRC}; 0: Output disabled.	0
[15:10]	Reserved	RO	Reserved	0
9	USBPD_IN_HVT	RW	PD pin PB6/PB7 high threshold input mode: 1: High threshold input, typical value about 2.2V,	0

			reduces I/O power consumption during PD communication; 0: Normal GPIO threshold input.	
[8:0]	Reserved	RW	Reserved	0

10.3.2.8 Remap Register 2 (AFIO_PCFR2)

Offset address: 0x1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						LPTIM_RM	SPI1_RM_H	I2C1_RM_H	TIM1_RM_H	TIM2_RM_H	USART1_RM_H		USART2_RM_H	Reserved	USART4_RM
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bit	Name	Access	Description	Reset value
[31:26]	Reserved	RO	Reserved	0
25	LPTIM_RM	RW	LPTIM remapping 0: Default mapping (CH1/PB12, CH2/PB13, ETR/PB14, OC/PB15) 1: Remapping (CH1/PB5, CH2/PB7, ETR/PB6, OC/PB2)	0
24	SPI1_RM_H	RW	SPI1 remap, combined with the SPI1_RM field of remap register 1 {SPI1_RM_H, SPI1_RM} 00: Default mapping (NSS/PA4, SCK/PA5, MISO/PA6, MOSI/PA7) 01: Full mapping (NSS/PA15, SCK/PB3, MISO/PB4, MOSI/PB5) 10: Fully mapped (NSS/PA12, SCK/PB6, MISO/PB8, MOSI/PB7) 11: Fully mapped (NSS/PB12, SCK/PB6, MISO/PB8, MOSI/PB7)	0
23	I2C1_RM_H	RW	The remapping of I2C1 is combined with the I2C1_RM field of Remap Register 1, {I2C1_RM_H, I2C1_RM}. This bit can be read or written by the user. It controls the mapping of I2C1's SCL and SDA multiplexing functions on the GPIO port: 00: Default mapping (SCL/PB6, SDA/PB7, SMBA/PB5) 10: Fully mapping (SCL/PA13, SDA/PA12, SMBA/PB5) 11: Fully mapped (SCL/PB9, SDA/PB11, SMBA/PB5)	0
22	TIM1_RM_H	RW	The remap bits for Timer 1 are used in conjunction with the TIM1_RM field of Remap Register 1, {TIM1_RM_H, TIM1_RM}. These bits can be read and written by the user. It controls the mapping of channels 1 through 4, 1N through 3N, External Trigger (ETR), and Brake Input	0

			(BKIN) of Timer 1 on the GPIO port: 000: Default mapping (ETR/PA12, CH1/PA8, CH2/PA9, CH3/PA10, CH4/PA11, BKIN/PB12, CH1N/PB13, CH2N/PB14, CH3N/PB15) 001: Fully mapped (ETR/PA12, CH1/PA8, CH2/PA9, CH3/PA10, CH4/PA11, BKIN/PA6, CH1N/PA7, CH2N/PB0, CH3N/PB1) 010: Fully mapping (ETR/PB6, CH1/PA1, CH2/PA7, CH3/PA14, CH4/PB1, BKIN/PA13, CH1N/PB11, CH2N/PB0, CH3N/PB9) 011: Fully mapped (ETR/PA3, CH1/PA1, CH2/PA7, CH3/PA14, CH4/PB1, BKIN/PA13, CH1N/PB11, CH2N/PB0, CH3N/PB9) 100: Fully mapped (ETR/PB6, CH1/PB7, CH2/PB8, CH3/PB12, CH4/PA2, BKIN/PA12, CH1N/PA14, CH2N/PB1, CH3N/PB9) 101: Fully mapped (ETR/PA13, CH1/PB1, CH2/PB8, CH3/PB6, CH4/PA3, BKIN/PA12, CH1N/PA14, CH2N/PA1, CH3N/PB7) 111: Controls LSI inputs only for LSI calibration	
21	TIM2_RM_H	RW	The remap bits for Timer 2 are used in conjunction with the TIM2_RM field of Remap Register 1, {TIM2_RM_H, TIM2_RM}. These bits can be read and written by the user. It controls the mapping of channels 1 through 4 of Timer 2 and external triggering (ETR) on the GPIO port: 000: Default mapping (CH1/ETR/PA0, CH2/PA1, CH3/PA2, CH4/PA3); 001: Partial mapping (CH1/ETR/PA15, CH2/PB3, CH3/PA2, CH4/PA3); 010: Partial mapping (CH1/ETR/PA0, CH2/PA1, CH3/PB10, CH4/PB11); 011: Fully mapping (CH1/ETR/PA15, CH2/PB3, CH3/PB10, CH4/PB11); 100: Fully mapping (CH1/ETR/PA3, CH2/PA2, CH3/PB12, CH4/PA6); 101: Fully mapping (CH1/ETR/PA12, CH2/PA2, CH3/PB12, CH4/PA6); 111: Fully mapping (CH1/ETR/PA12, CH2/PB8, CH3/PA5, CH4/PA4).	0
[20:19]	USART1_RM_H	RW	The remap bit of USART1, combined with the USART1_RM field of Remap Register 1, {USART1_RM_H, USART1_RM}. This bit can be read or written by the user. It controls the mapping of USART1's CTS, RTS, CK, TX, and RX multiplexing functions on the	00b

			GPIO port: 000: Default mapping (TX/PA9, RX/PA10, CK/PA8, CTS/PA11, RTS/PA12) 001: Fully mapping (TX/PB6, RX/PB7, CK/PA8, CTS/PA11, RTS/PA12) 010: Fully mapped (TX/PA4, RX/PA5, CK/PA3, CTS/PA2, RTS/PA13) 011: Fully mapped (TX/PA5, RX/PA4, CK/PA6, CTS/PB7, RTS/PB8) 100: Fully mapped (TX/PB11, RX/PB9, CK/PA6, CTS/PA14, RTS/PA13) 101: Fully mapped (TX/PB12, RX/PA12, CK/PB6, CTS/PB7, RTS/PB8)	
18	USART2_RM_H	RW	The remap bit for USART2, combined with the USART2_RM field of Remap Register 1, {USART2_RM_H, USART2_RM}. This bit can be read or written by the user. It controls the mapping of USART2's CTS, RTS, CK, TX and RX multiplexing functions on the GPIO port: 00: Default mapping (TX/PA2, RX/PA3, CK/PA4, CTS/PA0, RTS/PA1) 01: Reserved 10: Partial mapping (TX/PA11, RX/PA12, CK/PA4, CTS/PA0, RTS/PA1) 11: Partial mapping (TX/PA12, RX/PA11, CK/PA4, CTS/PA0, RTS/PA1)	0
17	Reserved	RW	Reserved	0
16	USART4_RM	RW	Remap bit for USART4. This bit can be read or written by the user. It controls the mapping of the TX, RX, CK, CTS, and RTS multiplexing functions of USART2 to the GPIO ports: 0: Default mapping (TX/PB0, RX/PB1, CK/PB2, CTS/PB3, RTS/PB4) 1: Fully mapping (TX/PA5, RX/PB5, CK/PA6, CTS/PA7, RTS/PA15)	0
[15:0]	Reserved	RO	Reserved	0

Chapter 11 Direct Memory Access Control (DMA)

Direct Memory Access (DMA) controllers provide a high-speed means of transferring data between a peripheral and memory or between memory and memory without CPU intervention, and data can be moved quickly through the DMA to conserve CPU resources for other operations.

Each channel of the DMA controller is dedicated to managing requests for memory access from one or more peripherals. There is also an arbiter to coordinate priorities between channels.

11.1 Main Features

- Multiple independent configurable channels
- Each channel is directly connected to dedicated hardware DMA request, and supports software trigger
- Support cyclic buffer management
- The priority of requests between multiple channels can be set by software programming (very high, high, medium and low level). When the priority settings are equal, it is determined by the channel number (the lower the channel number, the higher the priority)
- Support transmission from peripheral to memory, memory to peripheral, and memory to memory
- Flash memory, SRAM, peripheral SRAM, PB1, PB2 and HB peripherals can all be used as the sources and targets of access
- Number of programmable data transfer byte: 65535 at most

11.2 Functional Description

11.2.1 DMA Channel Processing

1) Arbitration priority

DMA requests generated by multiple independent channels are logically or structurally input to the DMA controller, and currently only one channel request is answered. The arbitrator within the module selects the access of the peripheral / memory to be initiated according to the priority of the channel request.

In software management, applications can independently configure priority levels for each channel, including the highest, high, medium and low levels, by setting the PL [1:0] bit of the DMA_CFGRx register. When the software settings between the channels are the same, the modules will be selected according to the fixed hardware priority, and the lower channel number will have higher priority than the higher one.

2) DMA configuration

When the DMA controller receives a request signal, it accesses the requesting peripheral or memory to establish the peripheral or data transfer between the memory and the memory. It mainly includes the following three steps:

1) data is taken from the memory address indicated by the peripheral data register or the current peripheral / memory address register, and the starting address of the first transmission is the peripheral base address or memory address specified by the DMA_PADDRx or DMA_MADDRx register.
2) the data is stored in the peripheral data register or the memory address indicated by the current peripheral / memory address register, and the starting address of the first transmission is the peripheral base address or memory

address specified in the DMA_PADDRx or DMA_MADDRx register.

3) perform a decrement operation of the values in the DMA_CNTRx register, which indicates the current number of outstanding operations.

Each channel has 3 DMA data transfer modes:

- Peripheral to memory (MEM2MEM=0, DIR=0)
- Memory to peripheral (MEM2MEM=0, DIR=1)
- Memory to memory (MEM2MEM=1)

Note: Memory-to-memory mode does not require a peripheral request signal, after configuring this mode (MEM2MEM=1), the channel is turned on (EN=1) to start data transfer. This method does not support cyclic mode.

The configuration process is as follows:

1) Set the first address of the peripheral register or the memory data address in the memory-to-memory mode (MEM2MEM=1) in the DMA_PADDRx register. When an DMA request occurs, this address will be the source or destination address of the data transfer.

2) Set the memory data address in the DMA_MADDRx register. When a DMA request occurs, the transmitted data will be read from or written to this address.

3) Set the amount of data to be transferred in the DMA_CNTRx register. After each data transmission, this value decreases.

4) Set the channel priority in the PL [1:0] bit of the DMA_CFGRx register.

5) Set the data transfer direction, loop mode, incremental mode of peripherals and memory, data width of peripherals and memory, transmission halfway, transmission completion, transmission error interrupt enable level in DMA_CFGRx register.

6) Set the EN bit of the DMA_CFGRx register and start the channel x.

Note: The control bits in the DMA_PADDRx/DMA_MADDRx/DMA_CNTRx register and the data transfer direction (DIR), loop mode (location), peripheral and memory incremental mode (MINC/PINC) in the DMA_CFGRx register can be configured to write only when the DMA channel is turned off.

3) Cycle mode

Set the CIRC location 1 of the DMA_CFGRx register to enable the circular mode function of channel data transfer. In circular mode, when the number of data transfers becomes 0, the contents of the DMA_CNTRx register are automatically reloaded to their initial values, the internal peripherals and memory address registers are also reloaded to the initial address values set by the DMA_PADDRx and DMA_MADDRx registers, and the DMA operation continues until the channel is closed or DMA mode is turned off.

4) DMA processing status

- Transfer more than half: corresponding to the HTIFx bit hardware setting in the DMA_INTFR register. When the number of DMA transmissions is reduced to less than half of the initial setting value, more than half of the DMA transfer flag will be generated, and if HTIE is set in the DMA_CFGRx register, an interrupt will occur. The hardware uses this flag to remind the application that it can prepare for a new round of data transfer.
- Transfer completed: corresponding to the TCIFx bit hardware setting in the DMA_INTFR register. When the number of DMA transmissions is reduced to 0, the DMA transfer completion flag will be generated, and if TCIE is set in the DMA_CFGRx register, an interrupt will occur.
- Transfer error: corresponding to the TEIFx bit hardware setting in the DMA_INTFR register. Reading and writing a reserved address area will result in a DMA transmission error. At the same time, the module hardware will automatically clear the EN bits of the DMA_CFGRx register corresponding to the channel where the error

occurred, and the channel is closed. If TEIE is set in the DMA_CFGRx register, an interrupt will occur.

When querying the status of the DMA channel, the application can first access the GIFx bit of the DMA_INTFR register, determine which channel has the DMA event, and then deal with the specific DAM event content of the channel.

11.2.2 Programmable Data Transmission Total Size/Data Bit Width/Alignment

The total amount of data transmitted by DMA in one round of each channel is programmable, up to 65535 times. The DMA_CNTRx register indicates the number of transfers to be transmitted. In EN=0, the setting value is written, and after the EN=1 opens the DMA transmission channel, the register becomes read-only, and the value decreases after each transfer.

The transmission data values of peripherals and memory support the function of automatic increment of address pointer, and the pointer increment is programmable. The first transmitted data address they access is stored in the DMA_PADDRx and DMA_MADDRx registers. By setting the PINC bit or MINC location 1 of the DMA_CFGRx register, you can turn on the peripheral address self-increment mode or the memory address self-increment mode, respectively. PSIZE [1:0] sets the peripheral address to take data size and address self-increase, and MSIZE [1:0] sets the memory address to take data size and address self-increase. There are 3 options: 8-bit, 16-bit, 32-bit. The specific data transfer methods are as follows:

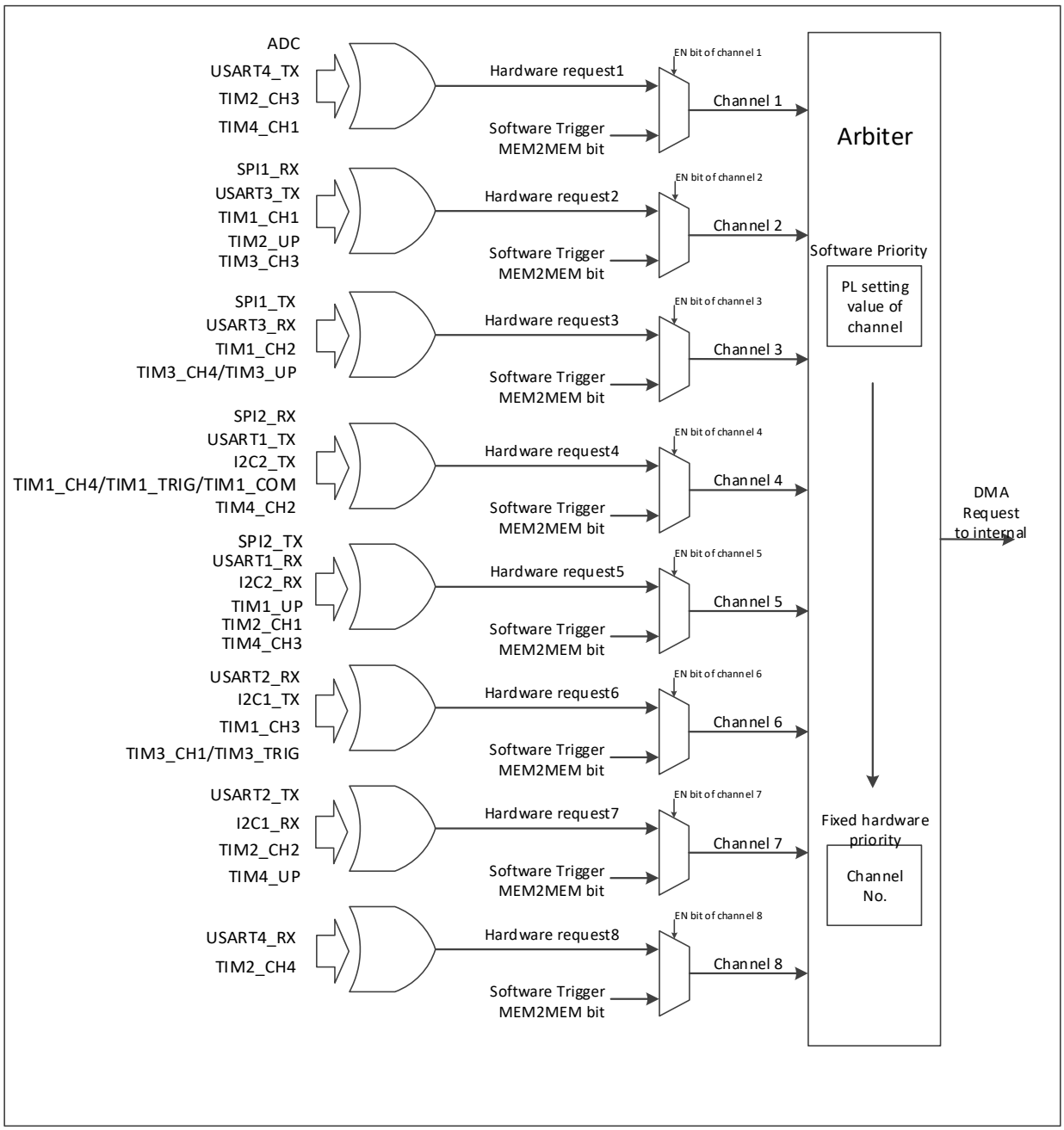
Table 11-1 DMA transfer under different data bit width (PINC=MINC=1)

Source bit width	Target bit width	Transfer data Number	Source: address/data	Target: address/data	Transfer operation
8	8	4	0x00/B0 0x01/B1 0x02/B2 0x03/B3	0x00/B0 0x01/B1 0x02/B2 0x03/B3	● The source address increment is aligned with the bit width of the data set by the source, and the size of the value is equal to the bit width of the source data. ● The target address increment is aligned with the bit width of the target set data, and the size of the value is equal to the bit width of the target data. ● DMA transfer to the target side of the data based on the principle: data size is not enough to make up the high bit 0, data size overflow high bit removed ● Data storage mode: small end mode, low address stores low bytes, high address stores high bytes.
8	16	4	0x00/B0 0x01/B1 0x02/B2 0x03/B3	0x00/00B0 0x02/00B1 0x04/00B2 0x06/00B3	
8	32	4	0x00/B0 0x01/B1 0x02/B2 0x03/B3	0x00/000000B0 0x04/000000B1 0x08/000000B2 0x0C/000000B3	
16	8	4	0x00/B1B0 0x02/B3B2 0x04/B5B4 0x06/B7B6	0x00/B0 0x01/B2 0x02/B4 0x03/B6	
16	16	4	0x00/B1B0 0x02/B3B2 0x04/B5B4 0x06/B7B6	0x00/B1B0 0x02/B3B2 0x04/B5B4 0x06/B7B6	
16	32	4	0x00/B1B0 0x02/B3B2 0x04/B5B4	0x00/0000B1B0 0x04/0000B3B2 0x08/0000B5B4	

			0x06/B7B6	0x0C/0000B7B6	
32	8	4	0x00/B3B2B1B0 0x04/B7B6B5B4 0x08/BBBAB9B8 0x0C/BFBEBDBC	0x00/B0 0x01/B4 0x02/B8 0x03/BC	
32	16	4	0x00/B3B2B1B0 0x04/B7B6B5B4 0x08/BBBAB9B8 0x0C/BFBEBDBC	0x00/B1B0 0x02/B5B4 0x04/B9B8 0x06/BDBC	
32	32	4	0x00/B3B2B1B0 0x04/B7B6B5B4 0x08/BBBAB9B8 0x0C/BFBEBDBC	0x00/B3B2B1B0 0x04/B7B6B5B4 0x08/BBBAB9B8 0x0C/BFBEBDBC	

11.2.3 DMA Request Mapping

Figure 11-1 DMA request mapping



The DMA controller provides 8 channels, and each channel corresponds to multiple peripheral requests. By setting the corresponding DMA control bit in the corresponding peripheral register, the DMA function of each peripheral can be turned on or off independently. The specific correspondence is as follows.

Peripheral	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7	Channel 8
ADC	ADC							
SPI1		SPI1_RX	SPI1_TX					
SPI2				SPI2_RX	SPI2_TX			
USART1				USART1_TX	USART1_RX			

USART2						USART2_RX	USART2_TX	
USART3		USART3_TX	USART3_RX					
USART4	USART4_TX							USART4_RX
I2C1						I2C1_TX	I2C1_RX	
I2C2				I2C2_TX	I2C2_RX			
TIM1		TIM1_CH1	TIM1_CH2	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_UP	TIM1_CH3		
TIM2	TIM2_CH3	TIM2_UP			TIM2_CH1		TIM2_CH2	TIM2_CH4
TIM3		TIM3_CH3	TIM3_CH4 TIM3_UP			TIM3_CH1 TIM3_TRIG		
TIM4	TIM4_CH1			TIM4_CH2	TIM4_CH3		TIM4_UP	

11.3 Register Description

Table 11-2 DMA registers

Name	Access address	Description	Reset value
R32_DMA_INTFR	0x40020000	DMA interrupt status register	0x00000000
R32_DMA_INTFCR	0x40020004	DMA interrupt flag clear register	0x00000000
R32_DMA_CFGR1	0x40020008	DMA channel 1 configuration register	0x00000000
R32_DMA_CNTR1	0x4002000C	DMA channel 1 transfer data number register	0x00000000
R32_DMA_PADDR1	0x40020010	DMA channel 1 peripheral address register	0x00000000
R32_DMA_MADDR1	0x40020014	DMA channel 1 memory address register	0x00000000
R32_DMA_CFGR2	0x4002001C	DMA channel 2 configuration register	0x00000000
R32_DMA_CNTR2	0x40020020	DMA channel 2 transfer data number register	0x00000000
R32_DMA_PADDR2	0x40020024	DMA channel 2 peripheral address register	0x00000000
R32_DMA_MADDR2	0x40020028	DMA channel 2 memory address register	0x00000000
R32_DMA_CFGR3	0x40020030	DMA channel 3 configuration register	0x00000000
R32_DMA_CNTR3	0x40020034	DMA channel 3 transfer data number register	0x00000000
R32_DMA_PADDR3	0x40020038	DMA channel 3 peripheral address register	0x00000000
R32_DMA_MADDR3	0x4002003C	DMA channel 3 memory address register	0x00000000
R32_DMA_CFGR4	0x40020044	DMA channel 4 configuration register	0x00000000
R32_DMA_CNTR4	0x40020048	DMA channel 4 transfer data number register	0x00000000
R32_DMA_PADDR4	0x4002004C	DMA channel 4 peripheral address register	0x00000000
R32_DMA_MADDR4	0x40020050	DMA channel 4 memory address register	0x00000000
R32_DMA_CFGR5	0x40020058	DMA channel 5 configuration register	0x00000000
R32_DMA_CNTR5	0x4002005C	DMA channel 5 transfer data number register	0x00000000

R32_DMA_PADDR5	0x40020060	DMA channel 5 peripheral address register	0x00000000
R32_DMA_MADDR5	0x40020064	DMA channel 5 memory address register	0x00000000
R32_DMA_CFGR6	0x4002006C	DMA channel 6 configuration register	0x00000000
R32_DMA_CNTR6	0x40020070	DMA channel 6 transfer data number register	0x00000000
R32_DMA_PADDR6	0x40020074	DMA channel 6 peripheral address register	0x00000000
R32_DMA_MADDR6	0x40020078	DMA channel 6 memory address register	0x00000000
R32_DMA_CFGR7	0x40020080	DMA channel 7 configuration register	0x00000000
R32_DMA_CNTR7	0x40020084	DMA channel 7 transfer data number register	0x00000000
R32_DMA_PADDR7	0x40020088	DMA channel 7 peripheral address register	0x00000000
R32_DMA_MADDR7	0x4002008C	DMA channel 7 memory address register	0x00000000
R32_DMA_CFGR8	0x40020094	DMA channel 8 configuration register	0x00000000
R32_DMA_CNTR8	0x40020098	DMA channel 8 transfer data number register	0x00000000
R32_DMA_PADDR8	0x4002009C	DMA channel 8 peripheral address register	0x00000000
R32_DMA_MADDR8	0x400200A0	DMA channel 8 memory address register	0x00000000

11.3.1 DMA Interrupt Flag Register (DMA_INTFR)

Offset address: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TEIF8	HTIF8	TCIF8	GIF8	TEIF7	HTIF7	TCIF7	GIF7	TEIF6	HTIF6	TCIF6	GIF6	TEIF5	HTIF5	TCIF5	GIF5
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEIF4	HTIF4	TCIF4	GIF4	TEIF3	HTIF3	TCIF3	GIF3	TEIF2	HTIF2	TCIF2	GIF2	TEIF1	HTIF1	TCIF1	GIF1

Bit	Name	Access	Description	Reset value
31/27/23/19/15/11/7/3	TEIFx	RO	Transmission error flag for channel x (x= 1/2/3/4/5/6/7/8): 1: A transmission error occurred on channel x. 0: There is no transmission error on channel x. The hardware is set, and the software writes the CTEIFx bit to clear this flag.	0
30/26/22/18/14/10/6/2	HTIFx	RO	Transmission more than half flag of channel x (x, x, 1, 2, 2, 3, 4, 5, 6, 6, 7, 8): 1: More than half of the transmission events occurred on channel x. 0: No more than half of the transmission is on channel x. The hardware is set, and the software writes the CHTIFx bit to clear this flag.	0
29/25/21/17/13/9/5/1	TCIFx	RO	Transmission completion flags for channel x (x=1/2/3/4/5/6/7/8): 1: A transmission completion event was generated on channel x; 0: No transmission completion event on channel x. Hardware sets and software writes the CTCIFx bit to clear	0

			this flag.	
28/24/20/16/12/8/4/0	GIFx	RO	Global interrupt flag for channel x (x=1/2/3/4/5/6/7/8): 1: TEIFx or HTIFx or TCIFx was generated on channel x; 0: No TEIFx or HTIFx or TCIFx has been generated on channel x. Hardware sets and software writes the CGIFx bit to clear this flag.	0

11.3.2 DMA Interrupt Flag Clear Register (DMA_INTFCR)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CTEIF8	CHTIF8	CTCIF8	CGIF8	CTEIF7	CHTIF7	CTCIF7	CGIF7	CTEIF6	CHTIF6	CTCIF6	CGIF6	CTEIF5	CHTIF5	CTCIF5	CGIF5
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTEIF4	CHTIF4	CTCIF4	CGIF4	CTEIF3	CHTIF3	CTCIF3	CGIF3	CTEIF2	CHTIF2	CTCIF2	CGIF2	CTEIF1	CHTIF1	CTCIF1	CGIF1

Bit	Name	Access	Description	Reset value
31/27/23/19/15/11/7/3	CTEIFx	WO	Clear the transmission error flag for channel x (x=1/2/3/4/5/6/8): 1: Clear the TEIFx flag in the DMA_INTFR register; 0: No effect.	0
30/26/22/18/14/10/6/2	CHTIFx	WO	Clear the transmit half flag for channel x (x=1/2/3/4/5/6/8): 1: Clear the HTIFx flag in the DMA_INTFR register; 0: No effect.	0
29/25/21/17/13/9/5/1	CTCIFx	WO	Clear the transmission completion flag for channel x (x=1/2/3/4/5/6/7/8): 1: Clear the TCIFx flag in the DMA_INTFR register; 0: No effect.	0
28/24/20/16/12/8/4/0	CGIFx	WO	Clear the global interrupt flag for channel x (x=1/2/3/4/5/6/7/8): 1: Clears the TEIFx/HTIFx/TCIFx/ GIFx flags in the DMA_INTFR register; 0: No effect.	0

11.3.3 DMA Channel x Configuration Register (DMA_CFGRx) (x=1/2/3/4/5/6/7/8)

Offset address: 0x08 + (x-1)*20

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN

Bit	Name	Access	Description	Reset value
[31:15]	Reserved	RO	Reserved	0
14	MEM2MEM	RW	Memory to memory mode enable: 1: Enable memory to memory mode; 0: Disable memory to memory mode.	0
[13:12]	PL[1:0]	RW	Channel priority level setting: 00: Low; 01: Medium; 10: High; 11: Very high.	00b
[11:10]	MSIZE[1:0]	RW	Memory address data width setting: 00: 8-bit; 01: 16-bit; 10: 32 bits; 11: Reserved.	00b
[9:8]	PSIZE[1:0]	RW	Peripheral address data width setting: 00: 8-bit; 01: 16-bit; 10: 32 bits; 11: Reserved.	00b
7	MINC	RW	Memory address increment mode enables: 1: Enable memory address increment operation; 0: Memory address remains unchanged operation.	0
6	PINC	RW	Peripheral address increment mode enables: 1: Enable memory address increment operation; 0: Peripheral address remains unchanged operation.	0
5	CIRC	RW	DMA channel cyclic mode enable: 1: Enable cyclic operation; 0: Perform a single operation.	0
4	DIR	RW	Data transmission direction: 1: Read from memory. 0: Read from the peripheral.	0
3	TEIE	RW	Transmission error interrupt enable control: 1: Enable transmission error interrupt. 0: Disable transmission error interrupt.	0
2	HTIE	RW	More than half of the transmission interruption enables control: 1: Enable transmission for more than half an interruption. 0: Disable transmission for more than half an interruption.	0
1	TCIE	RW	Transfer complete interrupt enable control: 1: Enable transmission completion interrupt; 0: Disable transmission completion interrupt.	0
0	EN	RW	Channel enable control: 1: Channel enabled; 0: Channel disabled. When a DMA transmission error occurs, it will be cleared to 0 automatically by hardware, and channel is disabled.	0

11.3.4 DMA Channel x Transfer Data Number Register (DMA_CNTRx) (x=1/2/3/4/5/6/7/8)

Offset address: 0x0C + (x-1)*20

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

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:0]	NDT[15:0]	RW	Number of data transfers, range 0-65535. Indicating the remaining number of transfers (the register contents are decremented after each DMA transfer). When the channel is in loop mode, the contents of the register are automatically reloaded to the previously configured value.	0

Note:Indicating the current number of transfers. When the register content is 0, no data transfer occurs regardless of whether the channel is open or not.

11.3.5 DMA Channel x Peripheral Address Register (DMA_PADDRx) (x=1/2/3/4/5/6/7/8)

Offset address: 0x10 + (x-1)*20

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

Bit	Name	Access	Description	Reset value
[31:0]	PA[31:0]	RW	The base address of the peripheral, which is used as the source or destination address of the peripheral data transmission. When PSIZE [1:0] = '01' (16 bits), the module automatically ignores bit0, and the operation address automatically aligns with 2 bytes; when PSIZE [1:0] ='10'(32 bits), the module automatically ignores bit [1:0], and the operation address automatically aligns with 4 bytes.	0

Note: This register can only be changed when EN=0, and cannot be written when EN=1.

11.3.6 DMA Channel x Memory Address Register (DMA_MADDRx) (x=1/2/3/4/5/6/7/8)

Offset address: 0x14 + (x-1)*20

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

Bit	Name	Access	Description	Reset value
[31:0]	MA[31:0]	RW	Memory data address, as the source or destination address for data transmission. When MSIZE [1:0] = '01' (16 bits), the module automatically ignores bit0, and the operation address	0

			automatically aligns with 2 bytes; when MSIZE [1:0] ='10'(32 bits), the module automatically ignores bit [1:0], and the operation address automatically aligns with 4 bytes.	

Note: This register can only be changed when EN=0, and cannot be written when EN=1.

Chapter 12 Analog-to-digital Converter (ADC)

The ADC module contains a 12-bit successive approximation analog-to-digital converter with the highest 48MHz input clock. Support 10 external channels and 3 internal signal source sampling sources. It can complete the functions of single conversion, continuous conversion, automatic scanning mode, discontinuous mode, external trigger mode and so on. You can monitor whether the channel voltage is within the threshold range by simulating the watchdog function.

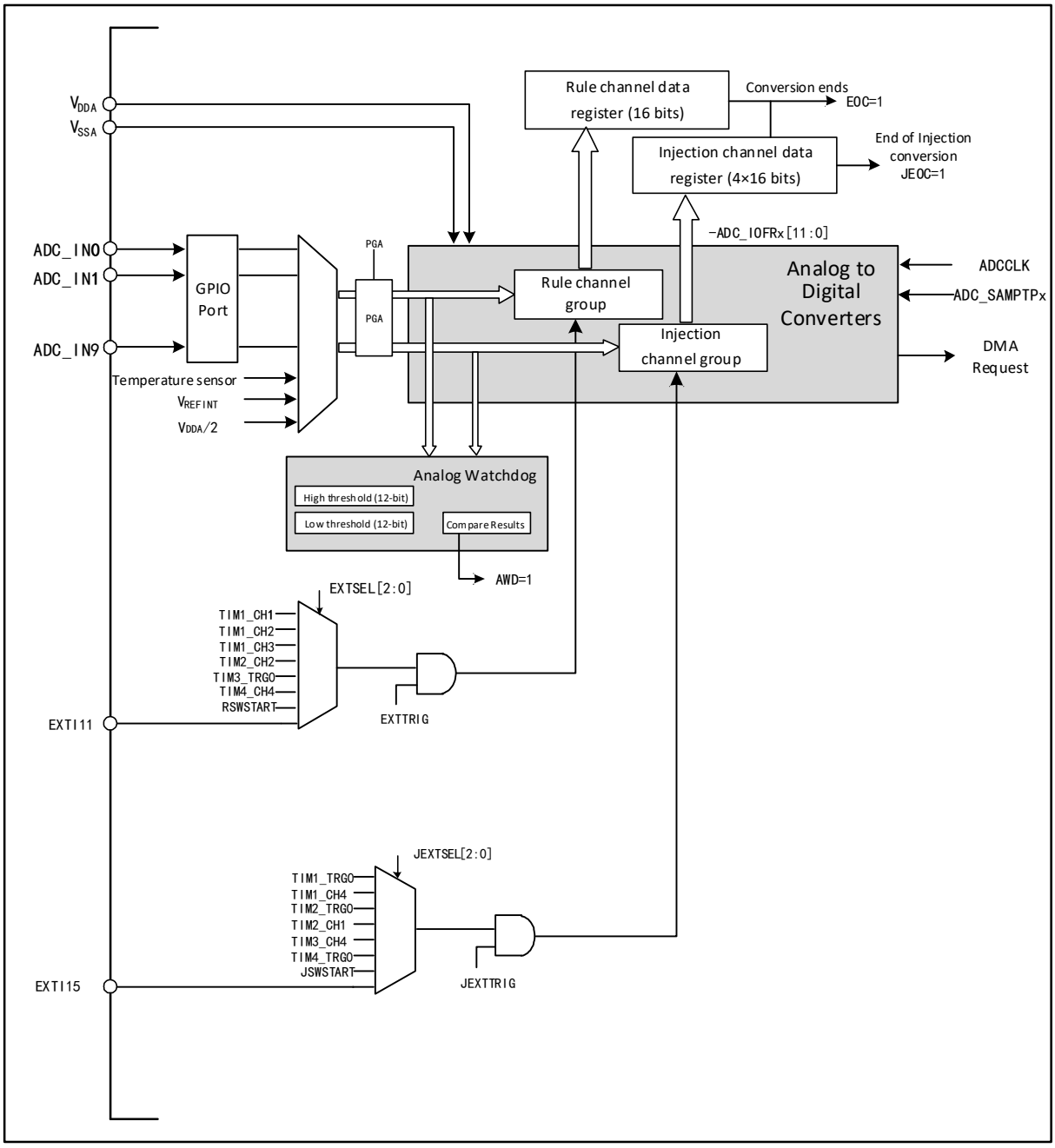
12.1 Main Features

- 12-bit resolution
- 10 external channels and 3 internal signal sources for sample
- Multiple sampling conversion modes for multiple channels: single, continuous, scan, trigger, discontinuous, etc.
- Data alignment mode: Left alignment, right alignment
- Sampling time can be programmed separately per channel
- Both regular conversion and injected conversion support external trigger
- Analog watchdog monitors the channel voltage, and has self-calibration function
- ADC channel input range: $0 \leq V_{IN} \leq V_{DDA}$
- Adjustable input gain, to implement small signal amplification and sampling

12.2 Functional Description

12.2.1 Module Structure

Figure 12-1 ADC module block diagram



12.2.2 ADC Configuration

1) Module power on

An ADON bit of 1 in the ADC_CTLR2 register indicates that the ADC module is powered on. When the ADC module enters the power-on state (ADON=1) from the power-off mode (ADON=0), it needs to be delayed for a period of time for t_{STAB} to stabilize the module. After that, the ADON bit is written again as 1, which is used as the startup signal for the software to start the ADC conversion. By clearing the ADON bit to 0, you can terminate the current conversion and put the ADC module in power-off mode, where ADC consumes almost no power.

2) Sample clock

The register operation of the module is based on the PCLK2 (PB2 bus) clock. The clock reference ADCCLK of the conversion unit is synchronized with the PCLK2. The frequency division is configured by the ADCPRE [1:0] domain of the RCC_CFGR0 register, and the maximum cannot exceed 48MHz.

3) Channel configuration

The ADC module provides 13 channel sampling sources, including 10 external channels and 3 internal channels. They can be configured into 2 translation groups: rule groups and injection groups. To realize the group conversion consisting of a series of transformations in any order on any number of channels.

Conversion group:

- Rule group: consists of up to 16 transformations. Regular channels and their conversion order are set in the ADC_RSQRx register. The total number of translations in the rule group should be written to L [3:0] in the ADC_RSQR1 register.
- Injection group: composed of up to 4 transformations. The injection channels and their conversion order are set in the ADC_ISQR register. The total number of translations in the injection group should be written to JL [1:0] in the ADC_ISQR register.

Note: If the ADC_RSQRx or ADC_ISQR register is changed during the conversion, the current conversion is terminated and a new startup signal will be sent to ADC to convert the newly selected group.

3 internal channels:

- Temperature sensor: Connect the ADC_IN16 channel to measure the internal temperature of the chip.
- V_{REFINT} internal reference voltage: Connect the ADC_IN17 channel.
- $V_{DDA}/2$: Connect the ADC_IN18 ADC_IN18.

4) Calibration

ADC has a built-in self-calibration mode. After calibration, the accuracy error caused by the change of internal capacitor bank can be greatly reduced. During calibration, an error correction code is calculated on each capacitor to eliminate errors on each capacitor in subsequent conversion.

FIFO_EN bit 1 of the ADC_CFG register, initialize the calibration register by writing RSTCAL bit 1 of the ADC_CTLR2 register, and wait for the RSTCAL hardware to clear 0 to indicate that the initialization is complete. Set the CAL bit to start the calibration function, once the calibration is finished, the hardware will clear the CAL bit automatically. Convert the internal channel 18 intermediate voltage value, the 2048 value is compared with the converted value for offset. After that the normal conversion function can start, adding the offset value after each ADC turn. It is recommended that an ADC calibration be performed when the ADC module is powered up.

Note: Before starting the calibration, you must ensure that the ADC module is powered on (ADON=1) for at least two ADC clock cycles.

5) Programmable sample time

ADC uses several ADCCLK cycles to sample the input voltage, and the number of sampling cycles of the channels can be changed by the SMPx [2:0] bit in the ADC_SAMPTR1 and ADC_SAMPTR2 registers. Each channel can be sampled at different times.

The total conversion time is calculated as follows:

T_{CONV} = Sampling time + 12.5T_{ADCCLK}

The sampling time can be determined according to SMPx [2:0]

SMPx[2:0]	Sample time (ADC_LP=0)	Sample time (ADC_LP=1)
000	1.5 cycles	7.5 cycles
001	7.5 cycles	11.5 cycles
010	13.5 cycles	17.5 cycles
011	28.5 cycles	27.5 cycles
100	41.5 cycles	47.5 cycles
101	55.5 cycles	
110	71.5 cycles	
111	239.5 cycles	

The regular channel transformation of ADC supports the DMA feature. The value of regular channel conversion is stored in a single data register ADC_RDATAR. In order to prevent the continuous conversion of multiple regular channels from not taking the data from the ADC_RDATAR register in time, the DMA function of ADC can be turned on. The hardware generates an DMA request at the end of the conversion of the regular channel (EOC setting) and transfers the translated data from the ADC_RDATAR register to the destination address specified by the user. After the channel configuration of the DMA controller module is completed, write the DMA location 1 of the ADC_CTLR2 register and turn on the DMA function of ADC.

Note: the injection group transformation does not support the DMA feature.

6) Data alignment

The ALIGN bit in the ADC_CTLR2 register selects the data storage alignment after ADC conversion. 12-bit data supports left and right alignment modes.

The data register ADC_RDATAR of the regular group channel stores the actual converted 12-bit digital value, while the data register ADC_IDATARx of the injected group channel is the value written after the actual converted data minus the offset of the ADC_IOFRx register, so there are positive and negative SIGNB.

Figure 12-2 Data left alignment

Rule group data register

D11	D10	D9	D8	D7	D6	D5	D4	D4	D2	D1	D0	0	0	0	0

Injected group data register

SIGNB	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0

Figure 12-3 Data right alignment

Rule group data register

0	0	0	0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

Injected group data register

SIGNB	SIGNB	SIGNB	SIGNB	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

7) ADC configuration enable

The ADC module provides an ADC_CFG register that enables a module function by setting the corresponding control bit.

The ADC_DUTY_EN bit is the ADC clock duty cycle control bit. When this position is 1, the input clock is delayed

4ns to get an ADC clock with a duty cycle of about 75%. When set to 0, the input clock is not processed.

The FIFO_EN bit enables ADCFIFO at high levels.

ADC_LP bit controls ADC low power mode, vcmbuffer and comparator have high power consumption when set 1, which is suitable for sampling rate of 1m and above, and enter low power mode when setting 0. vcmbuffer and comparator have lower power consumption, which is suitable for sampling rate of 1m and below.

The [3:0] position of ADC_BUFTRIM is the ADCBUFFER misalignment calibration control position, in which the highest position selects the positive and negative polarity of misalignment calibration, and the [2:0] controls the calibration gear.

12.2.3 External Trigger Source

The start event of an ADC transformation can be triggered by an external event. If the EXTTRIG or JEXTTRIG bit of the ADC_CTLR2 register is set, the conversion of the rule group or injection group channel can be triggered by external events, respectively. At this point, the configuration of the EXTSEL[2:0] and JEXTSEL[2:0] bits determines the external event sources of the rule group and the injection group.

Table 12-1 External trigger sources of regular group channel

EXTSEL[2:0]	Trigger source	Type
000	CC1 event of timer 1	Internal signal from on-chip timers
001	CC2 event of timer 1	
010	CC3 event of timer 1	
011	CC2 event of timer 2	
100	TRGO event of timer 3	
101	CC4 event of timer 4	
110	EXTI line 11	From external pin
111	RSWSTART position 1 software trigger	Software control bit

Table 12-2 External trigger sources of injected group channel

JEXTSEL[2:0]	Trigger source	Type
000	TRGO event of timer1	Internal signal from on-chip timers
001	CC4 event of timer1	
010	TRGO event of timer2	
011	CC1 event of timer2	
100	CC4 event of timer3	
101	TRGO event of timer4	
110	EXTI line 15	From external pin
111	JSWSTART position 1 software trigger	Software control bit

12.2.4 Conversion Mode

Table 12-3 Conversion mode combination

ADC_CTLR1 and ADC_CTLR2 register control bits					ADC conversion mode
CONT	SCAN	DISCEN/JDISCEN	IAUTO	Start event	

0	0	0	0	ADON bit set to 1	Single channel mode: A regular channel performs a single conversion.
				External trigger mode	Single channel mode: A regular channel or a channel of an injection channel performs a single conversion.
	1	0	0	ADON bit set to 1 or external trigger mode	Single scan mode: performs a single conversion on all selected regular group channels (ADC_RSQRx) or all injection group channels (ADC_ISQR) one by one in sequence. Trigger injection method: during the rule group channel conversion, all the transformations of the injection group channel can be inserted, and then the rule group channel conversion can be continued; however, the rule group channel transformation will not be inserted when the conversion is injected into the group channel.
			1	ADON bit set to 1 or external trigger mode	Single scan mode: performs a single conversion on all selected regular group channels (ADC_RSQRx) or all injection group channels (ADC_ISQR) one by one in sequence. Automatic injection mode: after the rule group channel is converted, the injection group channel is automatically converted. *Note: the external trigger signal of the injection channel is not allowed during the conversion process.*
	0	1 (DISCEN and JDISCEN cannot be 1 at the same time)	0	External trigger mode	Single break mode: each time an event is started, a short sequence of channel number conversions (the number defined by DISCNUM [2:0]) is performed until all selected channel conversions are completed. *Note: Rule group and injection group select this mode control bit as DISCEN and JDISCEN respectively. Discontinuous mode cannot be configured for both rule group and injection group. Discontinuous mode can only be used for one set of transformations.*
			1	-	Disable such mode.
	1	1	X	-	No such mode.
1	0	0	0	ADON bit set	Continuous single channel / scan mode:

	1	0	0	to 1 or external trigger mode	repeat a new round of conversion at the end of each round until the CONT is cleared 0.
			1		

Note: The external trigger events of the rule group and the injection group are different, and the 'ADON' bit can only start the rule group channel transformation, so the start events of the rule group and injection group channel transformation are independent.

1) Single channel conversion mode

In this mode, only one conversion is performed for the current 1 channel. This mode performs conversion on the channels sorted first in the rule group or injection group, where it can be started by setting ADON position 1 of the ADC_CTLR2 register (for regular channels only) or by external triggering (for regular channels or injection channels). Once the conversion of the selected channel is completed:

If the rule group channel is converted, the conversion data is stored in the 16-bit ADC_RDATAR register, the EOC flag is set, and if the EOCIE bit is set, the ADC interrupt will be triggered.

If the injection group channel is converted, the conversion data is stored in the 16-bit ADC_IDATAR1 register, the EOC and JEOC flags are set, and if the JEOCIE or EOCIE bit is set, the ADC interrupt will be triggered.

2) Single scan mode conversion

Enter ADC scan mode by setting the scan bit of the ADC_CTLR1 register to 1. This mode is used to scan a set of analog channels and perform a single conversion one by one for all channels selected by the ADC_RSQRx register (for regular channels) or ADC_ISQR (for injection channels). When the current channel conversion ends, the next channel of the same group is automatically converted.

In the scanning mode, according to the state of IAUTO bits, it can be divided into trigger injection mode and automatic injection mode.

- Trigger injection

The IAUTO bit is 0, when the trigger event of the injection group channel conversion occurs in the process of scanning the rule group channel, the current conversion is reset, and the sequence of the injection channel is carried out in a single scan mode. After all the selected injection group channel scan conversion is completed, the last interrupted rule group channel conversion is restored.

If the start event of the regular channel occurs when scanning the channel sequence of the injection group, the conversion of the injection group will not be interrupted, but the conversion of the rule sequence will be performed after the conversion of the injection sequence is completed.

Note: when using triggered injection transformations, you must ensure that the interval between triggered events is longer than the injection sequence. For example, if it takes 28 ADCCLK to complete the conversion of the injection sequence, the minimum time between events to trigger the injection channel is 29 ADCCLK.

- Automatic injection

The IAUTO bit is 1, and after scanning all the channel translations selected by the rule group, the channel selected by the injection group is converted automatically. This method can be used to convert up to 20 conversion sequences in ADC_RSQRx and ADC_ISQR registers.

In this mode, the external trigger (JEXTTRIG=0) of the injection channel must be disabled.

Note: When the ADC clock pre-division factor (ADCPRE [1:0]) is 4 to 8, 1 ADCCLK interval is automatically inserted when switching from regular conversion to injection sequence or from injection conversion to regular sequence; when the ADC clock pre-division factor is 2, there is a delay of 2 ADCCLK intervals.

3) Single discontinuous mode conversion

Enter the break mode of the rule group or injection group by setting the DISCEN or JDISCEN bit of the ADC_CTLR1 register to 1. This mode distinguishes scanning a complete set of channels in the scan mode, but divides a group of channels into multiple short sequences, and each external trigger event will perform a scan conversion of a short sequence.

The length of the short sequence n (n < = 8) is defined in the DISCNUM [2:0] of the ADC_CTLR1 register. When DISCEN is 1, it is the discontinuous mode of the rule group, and the total length to be converted is defined in the L[3:0] of the ADC_RSQR1 register. When JDISCEN is 1, it is the discontinuous mode of the injection group, and the total length to be converted is defined in the JL [1:0] of the ADC_ISQR register. Both rule group and injection group cannot be set to discontinuous mode at the same time.

Example of rule group discontinuity mode:
DISCEN=1, DISCNUM [2:0] = 3, L[3:0] = 8, to be switched channel = 1, 3, 2, 5, 8, 4, 7, 6.
The first external trigger: the conversion sequence is: 1, 3, 2.
The second external trigger: the conversion sequence is: 5, 8, 4.
The third external trigger: the conversion sequence is: 7, 6, and EOC events are generated at the same time.
The 4th external trigger: the conversion sequence is: 1, 3, 2.

Example of injection group discontinuity mode:
JDISCEN=1, DISCNUM [2:0] = 1, JL[1:0] = 3, Channel to be changed = 1, 3, 2.
The first external trigger: the conversion sequence is: 1.
The second external trigger: conversion sequence is: 3.
The third external trigger: the conversion sequence is: 2, and both EOC and JEOC events are generated.
The 4th external trigger: conversion sequence is: 1.

Note:
1. When a rule group or injection group is converted in discontinuous mode, the conversion sequence does not automatically start from scratch at the end of the conversion sequence. When all subgroups are converted, the next trigger event initiates the conversion of the first subgroup.
2. You cannot use both automatic injection (IAUTO=1) and discontinuous mode.
3. Discontinuous mode cannot be set for both rule groups and injection groups, and can only be used for one set of transformations.

4) Continuous conversion

In continuous conversion mode, another conversion is started as soon as the current ADC conversion is completed, and the conversion does not stop on the last channel of the selection group, but continues again from the first channel of the selection group. Boot events in this mode include external trigger events and ADON position 1. After setting boot, CONT position 1 needs to be set.

If a regular channel is converted, the converted data is stored in the ADC_RDATAR register, the end of the conversion indicates that the EOC is set, and if EOCIE is set, an interrupt is generated.

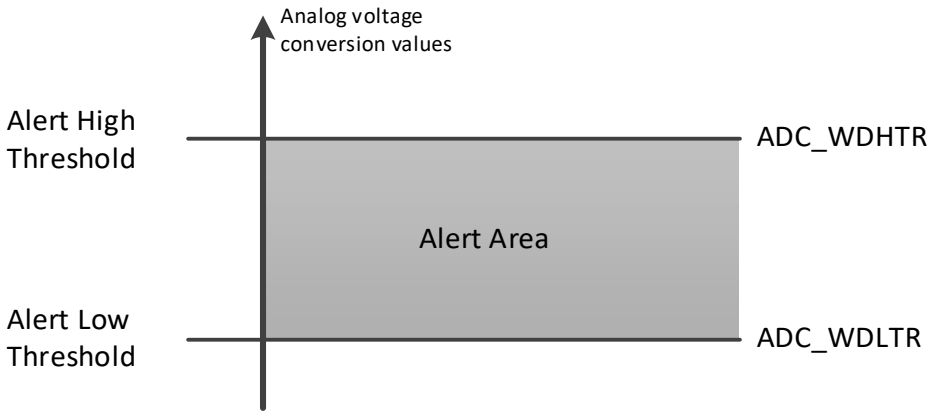
If an injection channel is converted, the converted data is stored in the ADC_IDATARx register, the end of injection

conversion flag JEOC is set, and an interrupt is generated if JEOCIE is set.

12.2.5 Analog Watchdog

If the analog voltage being converted by the ADC is below the low threshold or above the high threshold, the AWD analog watchdog status bit is set. The threshold setting is located in the lowest 12 valid bits of the ADC_WDHTR and ADC_WDLTR registers. The corresponding interrupt is allowed to be generated by setting the AWDIE bit in the ADC_CTLR1 register. Analog watchdog reset can be enabled by setting the AWDRST_EN bit of the ADC_CFG register.

Figure 12-4 Analog watchdog threshold area



Configure the AWDSGL, AWDEN, JAWDEN, and AWDCH[4:0] bits of the ADC_CTLR1 register to select the channel for analog watchdog alert, see the following table for the relationship:

Table 12-4 Analog Watchdog channel selection

Analog Watchdog alert channel	ADC_CTLR1 register control bit			
	AWDSGL	AWDEN	JAWDEN	AWDCH[4:0]
No alert	Ignore	0	0	Ignore
All injection channels	0	0	1	Ignore
All rule channels	0	1	0	Ignore
All injection and rule channels	0	1	1	Ignore
Single injected channel	1	0	1	Determine channel No.
Single regular channel	1	1	0	Determine channel No.
Single injected and regular channel	1	1	1	Determine channel No.

12.2.6 Temperature Sensor

Chip built-in temperature sensor, connected to the ADC_INT16 channel, through ADC to convert the output voltage of the sensor into digital value to feedback the internal temperature of the chip, it is recommended to set the sampling time is 17.1μs. The output voltage of the temperature sensor varies linearly with the temperature. Due to the manufacturing discreteness, the slope and offset of the linear curve are different, so the internal temperature sensor

is more suitable for detecting the change of temperature rather than measuring the absolute temperature. If you need to measure the temperature accurately, you should use an external temperature sensor.

By setting the TSVREFE position 1 of the ADC_CTLR2 register, awakening the ADC internal sampling channel, software startup or external trigger starts the temperature sensor channel conversion of the ADC, and reads the data result (mV). Among them, the conversion formula of numerical value and temperature (℃) is as follows:

Temperature (℃) = ((V_{SENSE}-V_{25})/Avg_Slope)+25

V25: The voltage value of the temperature sensor at 25℃

Avg_Slope: Average slope of temperature and V_{SENSE} curve (mV/℃)

Refer to the actual values of V_{25} and Avg_Slope in the electrical characteristics section of the datasheet.

Note: it takes a setup time for the internal temperature sensor to power on (the TSVREFE bit is changed from 0 to 1), and the ADC module also needs a setup time (the ADON bit is changed from 0 to 1), so in order to shorten the waiting time, both the ADON and TSVREFE bits can be set.

12.3 Register Description

Table 12-5 ADC-related registers

Name	Access address	Description	Reset value
R32_ADC_STATR	0x40012400	ADC status register	0x00000000
R32_ADC_CTLR1	0x40012404	ADC control register 1	0x00000000
R32_ADC_CTLR2	0x40012408	ADC control register 2	0x00000000
R32_ADC_SAMPTR1	0x4001240C	ADC sample time configuration register 1	0x00000000
R32_ADC_SAMPTR2	0x40012410	ADC sample time configuration register 2	0x00000000
R32_ADC_IOFR1	0x40012414	ADC injected channel data offset register 1	0x00000000
R32_ADC_IOFR2	0x40012418	ADC injected channel data offset register 2	0x00000000
R32_ADC_IOFR3	0x4001241C	ADC injected channel data offset register 3	0x00000000
R32_ADC_IOFR4	0x40012420	ADC injected channel data offset register 4	0x00000000
R32_ADC_WDHTR	0x40012424	ADC watchdog high threshold register	0x00000FFF
R32_ADC_WDLTR	0x40012428	ADC watchdog high threshold register	0x00000000
R32_ADC_RSQR1	0x4001242C	ADC regular channel sequence register 1	0x00000000
R32_ADC_RSQR2	0x40012430	ADC regular channel sequence register 2	0x00000000
R32_ADC_RSQR3	0x40012434	ADC regular channel sequence register 3	0x00000000
R32_ADC_ISQR	0x40012438	ADC injected channel sequence register	0x00000000
R32_ADC_IDATAR1	0x4001243C	ADC injected data register 1	0x00000000
R32_ADC_IDATAR2	0x40012440	ADC injected data register 2	0x00000000
R32_ADC_IDATAR3	0x40012444	ADC injected data register 3	0x00000000
R32_ADC_IDATAR4	0x40012448	ADC injected data register 4	0x00000000
R32_ADC_RDATAR	0x4001244C	ADC regular data register	0x00000000
R32_ADC_CFG	0x40012450	ADC configuration register	0x00000000

12.3.1 ADC Status Register (ADC_STATR)

Offset address: 0x00

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

Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											STRT	JSTRT	JEOC	EOC	AWD

Bit	Name	Access	Description	Reset value
[31:5]	Reserved	RO	Reserved	0
4	STRT	RW0	Regular channel conversion start status: 1: The regular channel conversion has started; 0: The regular channel conversion has not started. This bit is set by hardware and cleared by software (invalid if writing 1).	0
3	JSTRT	RW0	Injected channel conversion start status: 1: The injected channel conversion has started; 0: The injected channel conversion has not started; This bit is set by hardware and cleared by software (invalid if writing 1).	0
2	JEOC	RW0	Injected channel group conversion completion status: 1: The conversion has completed; 0: The conversion has not completed. This bit is set to 1 by hardware (the conversion of all injected channels is completed), and cleared by software (invalid if writing 1).	0
1	EOC	RW0	Conversion completion status: 1: The conversion has completed; 0: The conversion has not completed. This bit is set to 1 by hardware (the regular or injected channel group conversion ends), and is cleared by software (invalid if writing 1) or clearing when ADC_RDATAR is read.	0
0	AWD	RW0	Analog watchdog flag bit: 1: The analog watchdog event occurs; 0: No analog watchdog event occurs. This bit is set to 1 by hardware (the conversion value is out of the ADC_WDHTR and ADC_WDLTR register range), and is cleared by software (invalid if writing 1).	0

12.3.2 ADC Control Register 1 (ADC_CTLR1)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			PGA[1:0]		BUFEN	TKITUNE	TKENABLE	RAWDEN	JAWDEN	Reserved					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

DISCNUM[2:0]	JDISCEN	DISCEN	JAUTO	AWDSGL	SCAN	JEOCIE	AWDIE	EOCIE	AWDCH[4:0]

Bit	Name	Access	Description	Reset value
[31:29]	Reserved	RO	Reserved	0
[28:27]	PGA[1:0]	RW	ADC channel gain configuration: 00: x1 01: x4 10: x16 11: x64 *Note: The input gain can be configured, for amplifying small signals and sampling. ADC_Buffer needs to be enabled to use this function.*	00b
26	BUFEN	RW	ADC BUFFER enable. 0: Disable input Buffer. 1: Enable input Buffer.	0
25	TKITUNE	RW	TKEY module charging current configuration: 0: Charging current is 35μA; 1: Charging current is reduced half.	0
24	TKENABLE	RW	TKEY module enable control, including TKEY_F and TKEY_V units: 1: Enable TKEY module; 0: Disable TKEY module.	0
23	AWDEN	RW	Analog watchdog enable bit on regular channels: 1: Enable analog watchdog on regular channels; 0: Disable analog watchdog on regular channels.	0
22	JAWDEN	RW	Analog watchdog enable bit on injected channels: 1: Enable analog watchdog on injected channels; 0: Disable analog watchdog on injected channels.	0
[21:16]	Reserved	RO	Reserved	0000b
[15:13]	DISCNUM[2:0]	RW	In discontinuous mode, the number of regular channels to be converted after external trigger: 000: 1 channel; … 111: 8 channels.	000b
12	JDISCEN	RW	Discontinuous mode enable bit on injected channel: 1: Enable discontinuous mode on the injected channel; 0: Disable discontinuous mode on the injected channel.	0
11	DISCEN	RW	Discontinuous mode enable level on the regular channel: 1: Enable discontinuous mode on regular channel. 0: Disable discontinuous mode on regular channel.	0
10	JAUTO	RW	After opening the regular channel, automatically convert the enable bit of the injected channel group:	0

			1: Enable automatic injection channel group conversion. 0: Disable automatic injection channel group conversion. *Note: This mode needs to disable the external trigger function of the injection channel.*	
9	AWDSGL	RW	In scan mode, analog watchdog enable bit on a single channel: 1: Enable analog watchdog on single channel (AWDCH[4:0] selection); 0: Disable analog watchdog on all channels.	0
8	SCAN	RW	Scan mode enable bit: 1: Enable scan mode (continuous conversion of all channels selected by ADC_IOFRx and ADC_RSQRx); 0: Disable scan mode.	0
7	JEOCIE	RW	Injected channel group conversion completion interrupt enable bit: 1: Enable injected channel group transfer completion interrupt (IEOC flag); 0: Disable injected channel group transfer completion interrupt.	0
6	AWDIE	RW	Analog watchdog interrupt enable bit: 1: Enable analog watchdog interrupt; 0: Disable analog watchdog interrupt. *Note: In scan mode, if this interrupt occurs, the scan will be aborted.*	0
5	EOCIE	RW	Conversion completion (regular or injected channel group) interrupt enable bit; 1: Enable the transfer completion bit (EOC flag): 0: Disable the transfer completion interrupt.	0
[4:0]	AWDCH[4:0]	RW	Analog watchdog channel selection bit: 00000: Analog input channel 0; 00001: Analog input channel 1; … 01001: Analog input channel 9; 01010: Reserved; 01011: Reserved; 01100: Reserved; 01101: Reserved; 01110: Reserved; 01111: Reserved; 10000: Analog input channel 16; 10001: Analog input channel 17. 10010: Analog input channel 18. Other: Reserved.	00000b

12.3.3 ADC Control Register 2 (ADC_CTLR2)

Offset address: 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								TSVREFE	SWSTART	JSWSTART	EXTTRIG	EXTSEL[2:0]			Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JEXTTRIG	JEXTSEL[2:0]			ALIGN	Reserved		DMA	Reserved				RSTCAL	CAL	CONT	ADON

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RO	Reserved	0
23	TSVREFE	RW	Temperature sensor and internal voltage (V_{REFINT}) channel enable: 1: Enable the temperature sensor and V_{REFINT} channel; 0: Disable the temperature sensor and V_{REFINT} channel.	0
22	RSWSTART	RW	To start a regular channel conversion, you need to set the software trigger: 1: Start regular channel conversion. 0: Reset state. This bit is set by the software, and the hardware is cleared 0 after the conversion starts.	0
21	JSWSTART	RW	To start an injection channel conversion, you need to set the software trigger: 1: Start the injection channel conversion. 0: Reset state. This bit is set by the software, and the hardware or software clears 0 after the conversion starts.	0
20	EXTTRIG	RW	External trigger conversion mode enable for regular channels: 1: Enable conversion on external event; 0: Disable conversion on external event.	0
[19:17]	EXTSEL[2:0]	RW	External trigger event selection for regular channel: 000: CC1 event of timer 1; 001: CC2 event of timer 1; 010: CC3 event of timer 1; 011: CC2 event of timer 2; 100: TRGO event of timer 3; 101: CC4 event of timer 4; 110: EXTI line11; 111: RSWSTART software trigger.	000b
16	Reserved	RO	Reserved	0
15	JEXTTRIG	RW	External trigger conversion mode enable for injected channels:	0

			1: Enable conversion on external event; 0: Disable conversion on external event.	
[14:12]	JEXTSEL[2:0]	RW	External trigger event select for injected channels: 000: TRGO event of timer 1; 001: CC4 event of timer 1; 010: TRGO event of timer 2; 011: CC1 event of timer 2; 100: CC4 event of timer 3; 101: TRGO event of timer 4; 110: EXTI line 15; 111: JSWSTART software trigger.	000b
11	ALIGN	RW	Data alignment: 1: Left alignment; 0: Right alignment.	0
[10:9]	Reserved	RO	Reserved	0
8	DMA	RW	Direct memory access (DMA) mode enable: 1: Enable DMA mode; 0: Disable DMA mode.	0
[7:4]	Reserved	RO	Reserved	0
3	RSTCAL	RW	Reset calibration, this bit is set by software, and cleared by hardware after reset: 1: Initialize calibration register; 0: The calibration register initialized. *Note: If RSTCAL is set while the conversion is in progress, it takes extra cycles to clear the calibration register.*	0
2	CAL	RW	A/D calibration, set by software and cleared by hardware when the calibration is completed. 1: Enable the calibration: 0: Calibration completed.	0
1	CONT	RW	Continuous conversion enable: 1: Continuous conversion mode; 0: Single conversion mode. If this bit is set, the conversion will continue until the bit is cleared.	0
0	ADON	RW	A/D converter ON/OFF When this bit is 0, writing 1 will wake up the ADC from power-down mode; when this bit is 1, writing 1 will start the conversion. 1: Enable ADC and to start conversion; 0: Disable ADC conversion/calibration, and go to power down mode. *Note: When only ADON changes in the register, a conversion will be started. If any other bits are sent to change, a new conversion will not be started.*	0

12.3.4 ADC Sample Time Configuration Register 1 (ADC_SAMPTR1)

Offset address: 0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					SMP18[2:0]			SMP17[2:0]			SMP16[2:0]			Reserved	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bit	Name	Access	Description	Reset value
[31:27]	Reserved	RO	Reserved	0
[26:18]	SMPx[2:0]	RW	SMPx[2:0]: Sample time configuration of channel x: 000: 1.5 cycles; 001: 7.5 cycles; 010: 13.5 cycles; 011: 28.5 cycles; 100: 41.5 cycles; 101: 55.5 cycles; 110: 71.5 cycles; 111: 239.5 cycles; SMPx[2:0]: Sample time configuration of channel x (ADC_LP=1): 000: 7.5 cycles; 001: 11.5 cycles; 010: 17.5 cycles; 011: 27.5 cycles; 100: 47.5 cycles; 101: 55.5 cycles; 110: 71.5 cycles; 111: 239.5 cycles; These bits are used to independently select the sample time of each channel, and the channel configuration value must remain unchanged during the sampling period.	000b
[17:0]	Reserved	RO	Reserved	0

12.3.5 ADC Sample Time Configuration Register 2 (ADC_SAMPTR2)

Offset address: 0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		SMP9[2:0]			SMP8[2:0]			SMP7[2:0]			SMP6[2:0]			SMP5[2:1]	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMP5[0]		SMP4[2:0]			SMP3[2:0]			SMP2[2:0]			SMP1[2:0]			SMP0[2:0]	

Bit	Name	Access	Description	Reset value
[31:30]	Reserved	RO	Reserved	0
[29:0]	SMPx[2:0]	RW	SMPx [2:0]: sampling time configuration for channel x (ADC_LP=0): 000: 1.5 cycles; 001: 7.5 cycles; 010: 13.5 cycles; 011: 28.5 cycles; 100: 41.5 cycles; 101: 55.5 cycles; 110: 71.5 cycles; 111: 239.5 cycles;	000b

			SMPx [2:0]: sampling time configuration for channel x (ADC_LP=1): 000: 1.5 cycles; 001: 7.5 cycles; 010: 13.5 cycles; 011: 28.5 cycles; 100: 41.5 cycles; 101: 55.5 cycles; 110: 71.5 cycles; 111: 239.5 cycles; These bits are used to select the sampling time of each channel independently, and the channel configuration value must remain unchanged during the sampling period.	

12.3.6 ADC Injected Channel Data Offset Register (ADC_IOFRx) (x=1/2/3/4)

Offset address: 0x14 + (x-1)*4

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

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

Bit	Name	Access	Description	Reset value
[31:12]	Reserved	RO	Reserved	0
[11:0]	JOFFSETx[11:0]	RW	Data offset for injected channel x. When the injected channel is converted, these bits define the value to be subtracted from the original conversion data. The result of the conversion can be read in the ADC_IDATARx register.	0

12.3.7 ADC Watchdog High Threshold Register (ADC_WDHTR)

Offset address: 0x24

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

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

Bit	Name	Access	Description	Reset value
[31:12]	Reserved	RO	Reserved	0
[11:0]	HT[11:0]	RW	Analog watchdog high threshold set bits.	0xFFF

Note: The values of WDHTR and LTR can be changed during the conversion, but they will take effect in the next conversion.

12.3.8 ADC Watchdog Low Threshold Register (ADC_WDLTR)

Offset address: 0x28

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

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

Bit	Name	Access	Description	Reset value
[31:12]	Reserved	RO	Reserved	0
[11:0]	LT[11:0]	RW	Analog watchdog low threshold set bits.	0

Note: The values of WDHTR and LTR can be changed during the conversion, but they will take effect in the next conversion.

12.3.9 ADC Regular Channel Sequence Register1 (ADC_RSQR1)

Offset address: 0x2C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								L[3:0]				RSQ16[4:1]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ16[0]	SQ15[4:0]					SQ14[4:0]					SQ13[4:0]				

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RO	Reserved	0
[23:20]	L[3:0]	RW	The number of channels to be converted in a regular channel conversion sequence: 0000-1111: 1-16 conversions.	0
[19:15]	SQ16[4:0]	RW	Number of the 16th conversion channel in the regular sequence (0-9, 16-18).	0
[14:10]	SQ15[4:0]	RW	Number of the 15th conversion channel in the regular sequence (0-9, 16-18).	0
[9:5]	SQ14[4:0]	RW	Numbering of the 14th conversion channel in the regular sequence (0-9, 16-18).	0
[4:0]	SQ13[4:0]	RW	Numbering of the 13th conversion channel in the regular sequence (0-9, 16-18).	0

12.3.10 ADC Regular Channel Sequence Register 2 (ADC_RSQR2)

Offset address: 0x30

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		SQ12[4:0]					SQ11[4:0]					SQ10[4:1]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ10[0]	SQ9[4:0]					SQ8[4:0]					SQ7[4:0]				

Bit	Name	Access	Description	Reset value
[31:30]	Reserved	RO	Reserved	0
[29:25]	SQ12[4:0]	RW	Number of the 12th conversion channel in the regular sequence (0-9, 16-18).	0

[24:20]	SQ11[4:0]	RW	Number of the 11th conversion channel in the regular sequence (0-9, 16-18).	0
[19:15]	SQ10[4:0]	RW	Number of the 10th conversion channel in the regular sequence (0-9, 16-18).	0
[14:10]	SQ9[4:0]	RW	Number of the 9th conversion channel in the regular sequence (0-9, 16-18).	0
[9:5]	SQ8[4:0]	RW	Number of the 8th conversion channel in the regular sequence (0-9, 16-18).	0
[4:0]	SQ7[4:0]	RW	Number of the 7th conversion channel in the regular sequence (0-9, 16-18).	0

12.3.11 ADC Regular Channel Sequence Register 3 (ADCx_RSQR3)

Offset address: 0x34

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		SQ6[4:0]					SQ5[4:0]					SQ4[4:1]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ4[0]		SQ3[4:0]					SQ2[4:0]					SQ1[4:0]			

Bit	Name	Access	Description	Reset value
[31:30]	Reserved	RO	Reserved	0
[29:25]	SQ6[4:0]	RW	Number of the 6th conversion channel in the regular sequence (0-9, 16-18).	0
[24:20]	SQ5[4:0]	RW	Number of the 5th conversion channel in the regular sequence (0-9, 16-18).	0
[19:15]	SQ4[4:0]	RW	Number of the 4th conversion channel in the regular sequence (0-9, 16-18).	0
[14:10]	SQ3[4:0]	RW	Number of the 3th conversion channel in the regular sequence (0-9, 16-18).	0
[9:5]	SQ2[4:0]	RW	Number of the 2nh conversion channel in the regular sequence (0-9, 16-18).	0
[4:0]	SQ1[4:0]	RW	Number of the 1st conversion channel in the regular sequence (0-9, 16-18).	0

12.3.12 ADC Injected Channel Sequence Register (ADC_ISQR)

Offset address: 0x38

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										JL[1:0]		JSQ4[4:1]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JSQ4[0]		JSQ3[4:0]					JSQ2[4:0]					JSQ1[4:0]			

Bit	Name	Access	Description	Reset value
[31:22]	Reserved	RO	Reserved	0

[21:20]	JL[1:0]	RW	Injects the number of channels to be converted in the channel conversion sequence: 00-11: 1-4 conversions.	0
[19:15]	JSQ4[4:0]	RW	Inject the number of the 4th conversion channel in the sequence (0-9, 16-18). *Note: The software writes and assigns the channel number (0-9, 16-17) as the 4th in the sequence to be converted.*	0
[14:10]	JSQ3[4:0]	RW	Inject the number of the 3rd conversion channel in the sequence (0-9, 16-18).	0
[9:5]	JSQ2[4:0]	RW	Inject the number of the 2nd conversion channel in the sequence (0-9, 16-18).	0
[4:0]	JSQ1[4:0]	RW	Inject the number of the 1st conversion channel in the sequence (0-9, 16-18).	0

Note: Unlike regular conversion sequences, if the length of JL[1:0] is less than 4, the sequence order of conversions starts with (4-JL).

For example, when JL[1:0]=3 (4 injected transitions in the sequencer), the ADC will convert channels in the following order: JSQ1[4:0], JSQ2[4:0], JSQ3[4:0], and JSQ4[4:0];

When JL[1:0]=2 (3 injected transitions in the sequencer), the ADC will convert the channels in the following order: JSQ2[4:0], JSQ3[4:0] and JSQ4[4:0];

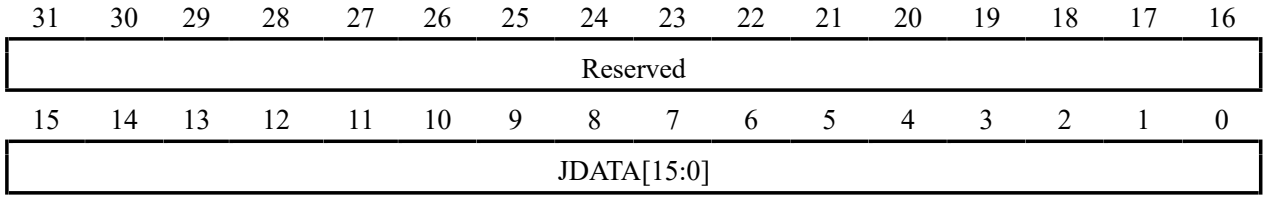
When JL[1:0]=1 (2 injected conversions in the sequencer), the ADC converts the channels in the following order: first JSQ3[4:0], then JSQ4[4:0];

When JL[1:0] = 0 (1 injection conversion in the sequencer), the ADC will convert only the JSQ4[4:0] channels.

If ADCx_ISQR[21:0]=10 00111 00011 00111 00010, the ADC will convert channels in the following order: JSQ2[4:0], JSQ3[4:0], and JSQ4[4:0], indicating that the scan conversions are performed in the following channel order: 7, 3, 7.

12.3.13 ADC Injected Data Register (ADC_IDATARx) (x=1/2/3/4)

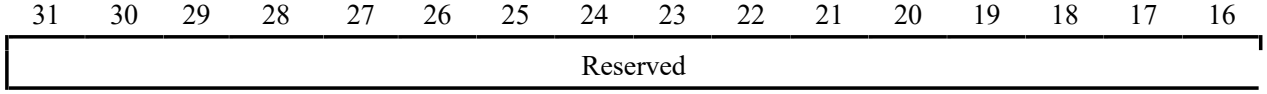
Offset address: 0x3C + (x-1)*4



Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved.	0
[15:0]	JDATA[15:0]	RO	Injected channel converted data (data left alignment or right alignment).	0

12.3.14 ADC Regular Data Register (ADC_RDATAR)

Offset address: 0x4C



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

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:0]	DATA	RO	Regular channel converted data (data left alignment or right alignment).	0

12.3.15 ADC Configuration Register (ADC_CFG)

Offset address: 0x50

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TKEY_DRV_OUTEN							TKEY_DRV_EN	ADC_DUTY_EN	FIFO_EN	ADC_LP	AWDRST_EN	ADC_BUFTRIM			

Bit	Name	Access	Description	Reset value
[31:19]	Reserved	RO	Reserved	0
[18:9]	TKEY_DRV_OUTEN	RW	TOUCHKEY Multi-mask each channel enable, active high.	0
8	TKEY_DRV_EN	RW	TOUCHKEY Multi-mask enable, active high.	0
7	ADC_DUTY_EN	RW	ADC clock duty cycle control bit 0: Input clock is not processed; 1: Extend the high level of the ADC clock by 4ns with no change in period.	0
6	FIFO_EN	RW	ADC FIFO enable, active high	0
5	ADC_LP	RW	ADC low-power mode control bit. 0: Low-power consumption mode, low power consumption of vcmbuffer and comparator, suitable for sampling rates of 1m and below. 1: vcmbuffer and comparator have high power consumption and are suitable for sampling rates of 1m and above.	0
4	AWDRST_EN	RW	Analog watchdog reset enable bit: 0: Disable analog watchdog reset; 1: Enable analog watchdog reset.	0
[3:0]	ADC_BUFTRIM	RW	ADCBUFFER misalignment calibration control position, of which the highest position selects the misalignment calibration polarity positive or negative, [2:0] controls the calibration gear.	0

Chapter 13 Touch Key Detection (TKEY)

The touch detection control (TKEY) unit, with the help of the voltage conversion function of the ADC module, realizes the touch key detection function by converting the capacitance to the voltage for sampling. The detection channel reuses 10 external channels of the ADC, and the touch key detection is realized through the single conversion mode of the ADC module.

13.1 Functional Description

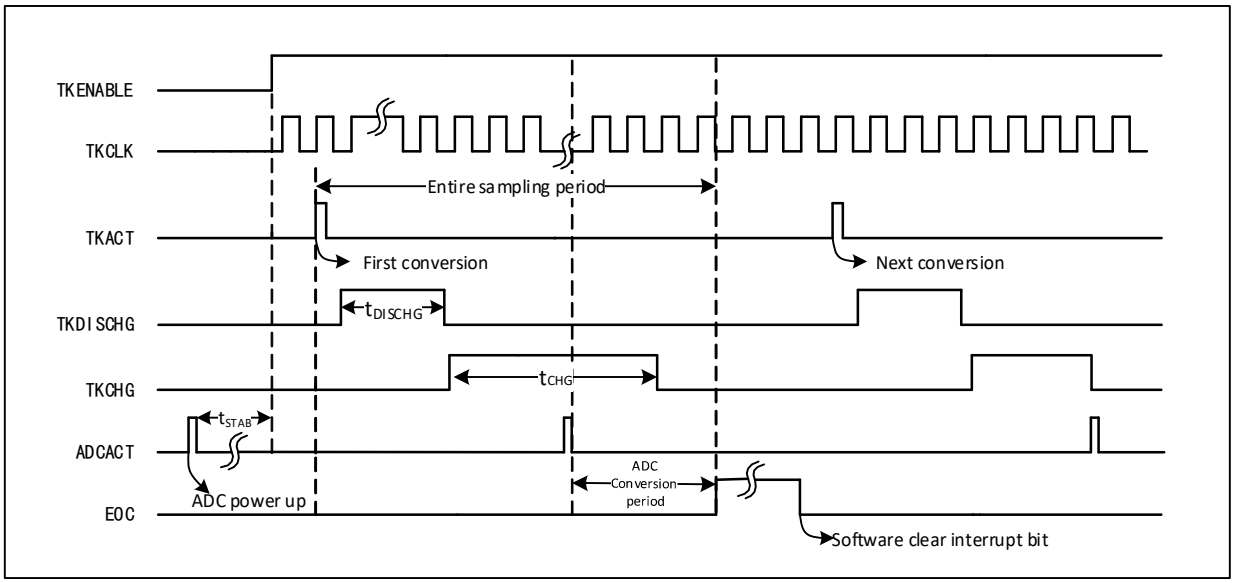
- Enable TKEY

For TKEY detection, the ADC module is needed. To enable TKEY, ensure that ADC is powered on (ADON=1), then set the TKENABLE bit in the ADC_CTLR1 register to 1. The charge current of the TKEY module can be adjusted by the TKITUNE bit.

TKEY only supports single 1-channel conversion mode, which configures the channel to be converted as the first channel in the regular sequence of ADC. And software starts conversion (write to TKEY_ACT_DCG).

Note: When the TKEY conversion is disabled, ADC channel configuration function can still be retained.

Figure 13-1 TKEY working timing diagram



- Programmable sampling time

The TKEY unit conversion requires discharging the channel using a number of ADCCLK clock cycles (tDISCHG) before charging the channel for voltage sampling over a number of ADCCLK cycles (tCHG), the number of charging cycles being the configured value of SMPx[2:0] in the ADC_SAMPTR2 register plus the TKEY_CHGOFFSET offset. Each channel can be individually adjusted with a different charging cycle for the sampling voltage.

- TKEY multi-mask

TKEY_DRV_EN bit is valid at high level, when set, TOUCHKEY multi-masking enable, for the control of the channel master switch; TKEY_DRV_OUTEN bit is valid at high level, when enabled, individually control each channel enable.

13.2 TKEY Operations Steps

TKEY detection belongs to the expansion function of ADC module, and its working principle is to change the capacitance perceived by hardware channel through "touch" and "non-touch" mode, and then convert the change of capacitance into voltage change through the number of charge and discharge cycles that can be set, and finally convert it into digital value through ADC module.

When sampling, you need to configure ADC to work in a single channel mode, and the "write operation" of the TKEY_ACT register initiates a conversion. The specific process is as follows:
1) Initialize the ADC function, configure the ADC module as a single conversion module, set the ADON bit to 1, and wake the ADC module. Open the TKEY unit at TKENABLE location 1 of the ADC_CTLR1 register.
2) Set the channel to be converted, write the channel number to the first conversion position in the ADC rule group sequence (ADC_RSQR3 [4:0]), and set L [3:0] to 1.
3) Set the charging sampling time of the channel, write the ADC_SMPPTR2 register, and configure a different charging time for each channel.
4) Write the TKEY_CHGOFFSET register to set the channel's charge time offset (low 10 bits valid) to adjust the charge time.
5) Write the TKEY_ACT_DCG register, set the discharge time (valid for low 10 bits), and start a sampling and conversion of TKEY.
6) Wait for the EOC conversion end flag position 1 of the ADC status register, and read the ADC_DR register to get the conversion value.
7) Repeat steps 2-6 if you need to make the next conversion. If you do not need to modify the channel charging sampling time, you can omit step 3 or 4.

13.3 TKEY Register Description

Table 13-1 TKEY-related registers

Name	Access address	Description	Reset value
R32_TKEY_CHGOFFSET	0x4001243C	TKEY charge time offset register	0x00000000
R32_TKEY_ACT_DCG	0x4001244C	TKEY start-up and discharge time register	0x00000000
R32_TKEY_DR	0x4001244C	TKEY data register	0x00000000

13.3.1 TKEY Charge Time Offset Register (TKEY_CHGOFFSET)

Offset address: 0x3C

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

Bit	Name	Access	Description	Reset value
[31:10]	Reserved	RO	Reserved	0
[9:0]	TKCGOFFSET[9:0]	WO	TKEY Charge Time Offset Configuration Value.	0

			Total charge time TCHG = TKCGOFFSET + SMPx	

Note: This register maps the injected data register1 (ADC_IDATAR1) of the ADC module. So when write operation is performed on this address register, it serves as TKEY charge time offset (TKEY_ CHGOFFSET). When read operation is performed, it serves as the injected data register1 (ADC_IDATAR1) of the ADC module.

13.3.2 TKEY Activate and Discharge Time Register (TKEY_ACT_DCG)

Offset address: 0x4C

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

Bit	Name	Access	Description	Reset value
[31:10]	Reserved	RO	Reserved	0
[9:0]	TKACT_DCG[9:0]	WO	Write the discharge time and activate a TKEY channel detection. Unit: System clock	0

Note: This register maps the regular data register (ADC_RDATAR) of the ADC module.

13.3.3 TKEY Data Register (TKEY_DR)

Offset address: 0x4C

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

位	名称	访问	描述	复位值
[31:16]	Reserved	RO	Reserved	0
[15:0]	DATA[15:0]	RO	Converted data.	0

Note: This register maps the regular data register (ADC_RDATAR) of the ADC module.

Chapter 14 Advanced-control Timer (ADTM)

The advanced-control timer module contains a powerful 16-bit automatic reset timer (TIM1), which can be used to measure pulse width or generate pulses, PWM waves, etc. Used in motor control, power supply and other fields.

14.1 Main Features

Advanced-control timer (TIM1) features include:

- 16-bit auto-reload counter, supports up count, down count and up/down count
- 16-bit prescaler; the frequency division factor is dynamically adjustable from 1 to 65536
- 4 independent compare/capture channels
- Each compare/capture channel supports multiple working modes, such as: input capture, output compare, PWM generation and single pulse output
- Complementary outputs with programmable deadband time
- External signal to control timer
- Repetition counter to update the timer after the determination of the cycle
- Break signal input to put the timer's output signals in reset status or in a known status
- DMA generation in multiple modes
- Incremental encoder
- Cascade connection and synchronization between timers

14.2 Principle and Structure

This section describes the internal structure of the advanced-control timer.

14.2.1 Overview

As shown in Figure 14-1, the structure of the advanced-control timer can be roughly divided into 3 parts: Input clock part, core counterpart and compare/capture channel part.

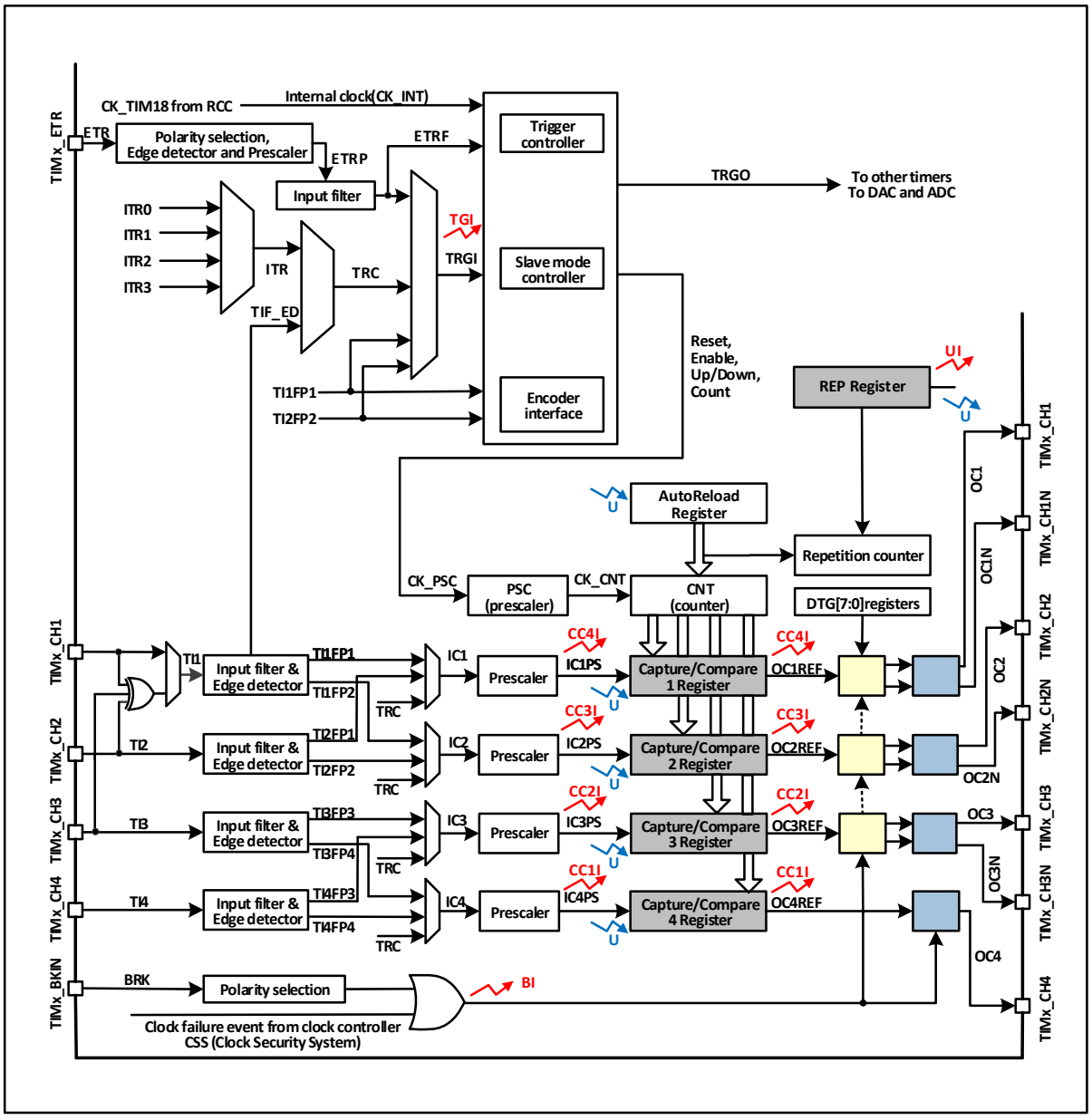
The advanced-control timer clock can come from PB bus clock (CK_INT), external clock input pin (TIMx_ETR), other timers with clock output function (ITRx), or the input end of compare capture channel (TIMx_CHx). These input clock signals will become CK_PSC clocks after various set filtering and frequency division operations, and will output to the core counterpart. In addition, these complex clock sources can also be output as TRGO to other timers, ADC and other peripherals.

The core of the advanced-control timer is a 16-bit counter (CNT). After CK_PSC is divided by the prescaler (PSC), it becomes CK_CNT and output to CNT. CNT supports up-counting mode, down-counting mode and up/down counting mode, and there is an automatic reload value register (ATRLR). After each counting cycle is completed, CNT will be reloaded with the initial value. In addition, there is an auxiliary counter that counts the number of times that ATRLR reloads the initial value for CNT. When the number of times reaches the number set in the repeat count register (RPTCR), a specific event can be generated.

The advanced-control timer has 4 groups of compare/capture channels. On each group of compare/capture channel, pulses can be inputted from its dedicated pins or output waveforms to the pins, i.e., the compare/capture channels support input and output modes. The input of each channel of the compare/capture register supports operations such as filtering, frequency division and edge detection, and supports mutual triggering between channels, and can also

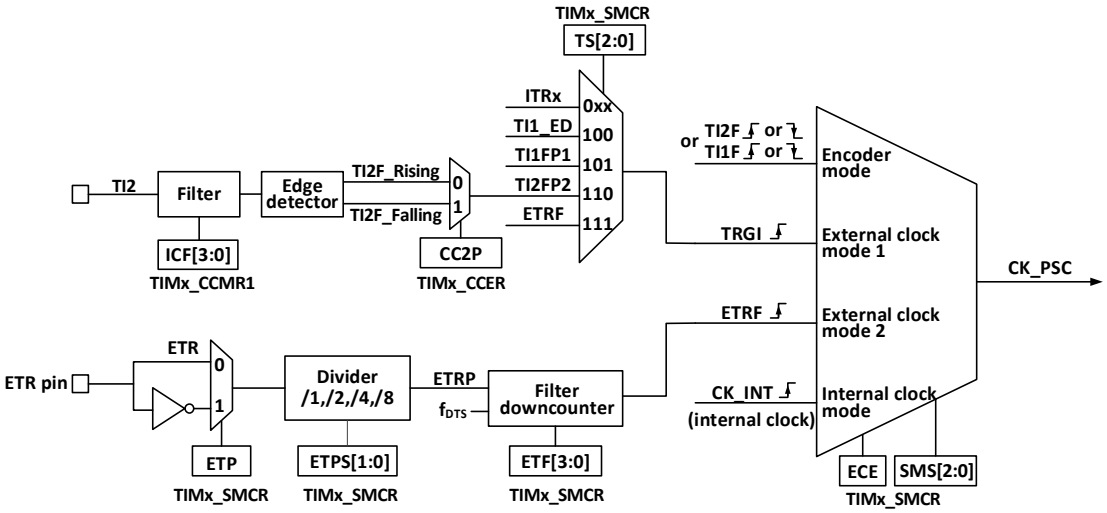
provide a clock for the core counter CNT. Each compare/capture channel has a set of compare/capture register (CHxCVR), which supports comparison with the main counter (CNT) so as to output pulse.

Figure 14-1 Structure block diagram of advanced-control timer



14.2.2 Clock Input

Figure 14-2 CK_PSC source block diagram of advanced-control timer



There are many clock sources for advanced-control timer CK_PSC, which can be divided into 4 categories:

1) The route of external clock pin (ETR) input clock: ETR→ETRP→ETRF;
2) Internal PB clock input route: CK_INT;
3) The route from the compare/capture channel pin (TIMx_CHx): TIMx_CHx→TIx→TIxFPx; this route is also used in encoder mode;
4) Input from other internal timers: ITRx;

The actual operation can be divided into 4 categories by determining the input pulse selection of the SMS from the CK_PSC source:

1) Select the internal clock source (CK_INT);
2) External clock source mode 1;
3) External clock source mode 2;
4) Encoder mode;

The 4 clock sources mentioned above can be selected by these 4 operations.

14.2.2.1 Internal Clock Source (CK_INT)

If the advanced-control timer is started when the SMS field is kept at 000b, then the internal clock source (CK_INT) is selected as the clock. At this moment, CK_INT is CK_PSC.

14.2.2.2 External Clock Source Mode 1

If SMS is set to 111b, the external clock source mode1 is enabled. When external clock source 1 is enabled, TRGI is selected as the source of CK_PSC. It is worth noting that you need to configure TS to select the source of TRGI. For TS, the following pulses can be used as the clock sources:

1) Internal Trigger (ITRx, x is 0,1,2,3);
2) Signal of compare/capture 1 after passing through the edge detector (TI1F_ED);
3) Signals TI1FP1 and TI2FP2 of compare/capture channel;
4) Signal ETRF from external clock pin.

14.2.2.3 External Clock Source Mode 2

Use external trigger mode 2 to count on every rising or falling edge of the external clock pin input. When the ECE bit is set, the external clock source mode 2 is used. When the external clock source mode2 is used, ETRF is selected as CK_PSC. The ETR pin passes through the optional inverter (ETP) and frequency divider (ETPS) to become ETRP, and then passes through the filter (ETF) to become ETRF.

When ECE bit is set and the SMS is set to 111b, it means that the TS selects ETRF as the input.

14.2.2.4 Encoder Mode

Set SMS as 001b, 010b and 011b to enable the encoder mode. After enabling the encoder mode, you may choose to use another transition edge as a signal for signal output at a certain level in TI1FP1 and TI2FP2. This mode is used when the external encoder is used. Refer to Section 14.3.10 for specific functions.

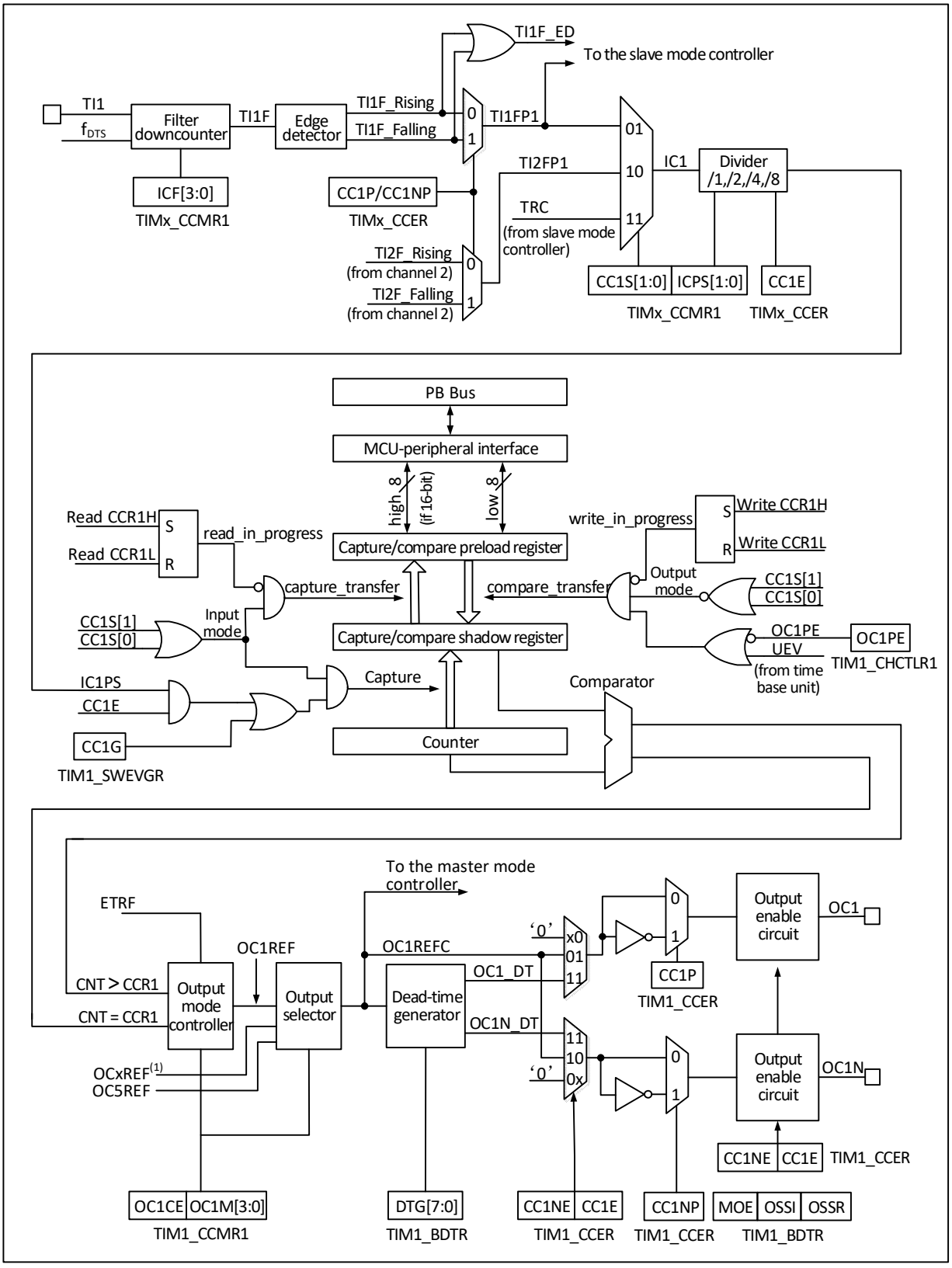
14.2.3 Counter and Periphery

CK_PSC inputs to the prescaler (PSC) for frequency division. PSC is 16 bits, and the actual frequency division factor is equivalent to the value of R16_TIMx_PSC+1. CK_PSC will become CK_INT after PSC. The changed value of R16_TIM1_PSC will not take effect in real time, but will be updated to the PSC after the update event. Update events include clearing and resetting the UG bit. The core of the timer is a 16-bit counter (CNT). CK_CNT will eventually be inputted to CNT. CNT supports up-counting mode, down-counting mode and up/down counting mode, and there is an automatic reload value register (ATRLR) which re-loads the initial value for CNT after each counting cycle is completed. In addition, there is an auxiliary counter that records the number of times that ATRLR reloads the initial value for CNT. When the number of times reaches the number set in the repeat count register (RPTCR), a specific event can be generated.

14.2.4 Compare/Capture Channel and Periphery

The compare/capture channel is the main component of the timer to achieve complex functions. Its core is the compare/capture register, supplemented by the digital filtering of the peripheral input part, frequency division and channel multiplexing, the output part comparator and output control.

Figure 14-3 Structure block diagram of compare/capture channel



The block diagram of the compare/capture channel is as shown in Figure 14-3. After the signal is inputted from the channel x pin, it can be selected as TIx (the source of TI1 may be more than CH1. See the timer structure block diagram 14-1). TI1 passes through the filter (ICF[3:0]) to generate TI1F, and then is divided into TI1F_Rising and TI1F_Falling after passing through the edge detector. These 2 signals are selected (CC1P) to generate TI1FP1,

TI1FP1 and TI2FP1 from channel 2 are sent to CC1S together to be selected as IC1, and then sent to the compare/capture register after going through the ICPS frequency division.

The compare/capture register is composed of a preload register and a shadow register, and only the preload register is operated during reading and writing. In the capture mode, the capture occurs on the shadow register, and then copied to the preload register; in the comparison mode, the content of the preload register is copied to the shadow register, and then the content of the shadow register is compared with the core counter (CNT).

14.3 Function and Implementation

The advanced-control timer complex functions are implemented by the operation of comparison &capture channel, clock input circuit, counter and peripheral parts of the timer. The timer's clock input can come from multiple clock sources including the input of the compare/capture channel. The operation of compare/capture channel and the clock source selection directly determines its function. The compare/capture channel is bidirectional and can work in input and output modes.

14.3.1 Counter Mode

Incremental counting mode

In incremental counting mode, the counter counts from 0 to the automatic reload value (The contents of the R16_TIMx_ATRLR register), then counts again from 0 and generates an overflow event on the counter.

If a repeat counter is used, an update event (UEV) is generated when the number of iterations of the increment count reaches the number of times programmed in the repeat counter register (R16_TIMx_RPTCR+1). Otherwise, an update event is generated each time the counter is overflowed.

An update event is also generated when you set the UG location 1 of the TIMx_SWEVGR register (through the software or using the slave mode controller).

The UEV event can be disabled by setting UDIS location 1 in the R16_TIMx_CTLR1 register by the software. This avoids updating the shadow register when a new value is written to the preloaded register. No update event occurs until the UDIS bit is written to 0. However, both the counter and the prescaler counter re-count from 0 (while the prescaler ratio remains the same). In addition, if the URS bit (update request selection) in the R16_TIMx_CTLR1 register is set to 1, setting the UG location 1 generates an update event UEV, but does not set the UIF flag to 1 (Therefore, no interrupts or DMA requests are sent). In this way, if the counter is cleared to zero when a capture event occurs, there will be no update interrupt and capture interrupt at the same time.

When an update event occurs, all registers are updated and the update flag (UIF bit in the R16_TIMx_INTFR register) is set to 1 (Depending on the URS bit):
1) The contents of the R16_TIMx_RPTCR register will be reloaded in the repeat counter.
2) The automatic overload shadow register will be updated with the preloaded value (R16_TIMx_ATRLR).
3) The preload value (Contents of the R16_TIMx_PSC register) will be reloaded in the buffer of the prescaler

Decreasing counting mode

In decrement counting mode, the counter decrements the count from the automatic overload value (The contents of the R16_TIMx_ATRLR register) to 0, then re-starts the count from the automatic overload value and generates a counter underflow event.

If a repeat counter is used, an update event (UEV) is generated when the number of repetitions of the decrement count reaches the number of times programmed in the repeat counter register (R16_TIMx_RPTCR+1). Otherwise, an update event is generated each time the counter is overflowed.

An update event is also generated when you set the UG location 1 of the R16_TIMx_EGR register (through the software or using the slave mode controller).

The UEV update event can be disabled by setting UDIS location 1 in the R16_TIMx_CTLR1 register by the software. This avoids updating the shadow register when a new value is written to the preloaded register. No update event occurs until the UDIS bit is written to 0. However, the counter starts counting again from the current automatic reload value, while the prescaler counter starts counting again at 0 (But the pre-division ratio remains the same).

In addition, if the URS bit (Update request selection) in the R16_TIMx_CTLR1 register is set to 1, setting the UG location 1 generates an update event UEV, but does not set the UIF flag to 1 (Therefore, no interrupts or DMA requests are sent). In this way, if the counter is cleared to zero when a capture event occurs, there will be no update interrupt and capture interrupt at the same time.

When an update event occurs, all registers are updated and the update flag (UIF bit in the R16_TIMx_INTFR register) is set to 1 (Depending on the URS bit):
1) the contents of the R16_TIMx_RPTCR register will be reloaded in the repeat counter.
2) the preload value (Contents of the R16_TIMx_PSC register) will be reloaded in the buffer of the prescaler.
3) the automatic reload activity register will be updated with the preloaded value (The contents of the R16_TIMx_ATRLR register).
Note: The automatic overload register is updated before the counter is overloaded, so the next count cycle is the desired new cycle length.

Center alignment mode (Increment / decrement count)
In center alignment mode, the counter counts from 0 to the automatic overload value (The contents of the R16_TIMx_ATRLR register)-1, generating a counter overflow event; then counting down to 1 from the automatic overload value and generating a counter underflow event. Then recount starts at 0.

The center alignment mode is valid when the CMS bit in the R16_TIMx_CTLR1 register is not "00". When a channel is configured in output mode, its output comparison interrupt flag is set to 1 in the following mode, that is, counter decrement count (Center alignment mode 1 minute CMS = "01"), counter increment count (Center alignment mode 2 # CMS = "10"), and counter increment / decrement count (Center alignment mode 3M CMS = "11").

In this mode, the DIR direction bit of the R16_TIMx_CTLR1 register is not writable, but is updated by the hardware and indicates the current counter direction.

An update event is generated each time a counter overflow and underflow occurs, or an update event can be generated by using UG location 1 in the R16_TIMx_SWEVGR register (Through the software or using the slave mode controller). In this case, the counter and the prescaler counter will start counting again from 0.

The UEV update event can be disabled by setting UDIS location 1 in the R16_TIMx_CTLR1 register by the software. This avoids updating the shadow register when a new value is written to the preloaded register. No update event occurs until the UDIS bit is written to 0. However, the counter still increments and decrements the count based on the current automatic overload value.

In addition, if the URS bit (Update request selection) in the R16_TIMx_CTLR1 register is set to 1, setting the UG location 1 generates a UEV update event, but the UIF flag is not set to 1 (Therefore, no interrupts or DMA requests are sent). In this way, if the counter is cleared to zero when a capture event occurs, there will be no update interrupt and capture interrupt at the same time.

When an update event occurs, all registers are updated and the update flag (UIF bit in the R16_TIMx_INTFR register) is set to 1 (Depending on the URS bit):
1) The contents of the R16_TIMx_RPTCR register will be reloaded in the repeat counter.
2) The preload value (Contents of the R16_TIMx_PSC register) will be reloaded in the buffer of the prescaler.
3) The automatic reload activity register will be updated with the preloaded value (The contents of the R16_TIMx_ATRLR register). Note that if the update operation is triggered by an overflow on the counter, the automatic overload register is updated before the counter is overloaded, so the next count cycle is the desired new cycle length (The counter is overloaded with new values).

14.3.2 Input Capture Mode

The input capture mode is one of basic functions of timer. The principle of the input capture mode is that when a certain edge on the ICxPS signal is detected, a capture event will occur, and the current value of the counter will be latched into the compare/capture register (R16_TIMx_CHCTLRx). When a capture event occurs, CCxIF (In R16_TIMx_INTFR) bit will be set. If an interrupt or DMA is enabled, a corresponding interrupt or DMA will be generated. If CCxIF is already set when a capture event occurs, then the CCxOF bit will be set. CCxIF can be cleared by software or by hardware through reading the compare/capture register. CCxOF is cleared by the software.

Take an example of channel 1 to illustrate the steps to use the input capture mode, as follows:
1) Configure CCxS and select the source of ICx signal. For example, it is set to 10b, and TI1FP1 is selected as the source of IC1, and the default setting cannot be used. CCxS defaults to use the compare capture module as the output channel;
2) Configure ICxF and set the digital filter of the TI signal. The digital filter will output a jump based on the determined frequency and determined sampling times. The sampling frequency and times are determined by ICxF;
3) Configure CCxP bit and set the polarity of TIxFPx. For example, maintain CC1P bit to be low and select the jump of rising edge;
4) Configure ICxPS and set ICx signal as the frequency division factor between ICxPS. For example, maintain the ICxPS as 00b without frequency division;
5) Configure the CCxE bit to allow to capture the core counter (CNT) value to the compare/capture register. Set the CC1E bit;
6) Configure the CCxIE and CCxDE bits as needed to decide whether to enable interrupt or DMA.

After these operations, the compare & capture channel configuration is completed.

When TI1 inputs a captured pulse, the value of the core counter (CNT) will be recorded in the compare/capture register, and CC1IF will be set. When CC1IF has been set before, the CCIOF bit will also be set. If CC1IE is set, then an interrupt will be generated; if CC1DE is set, a DMA request will be generated. An input capture event can be generated by software through writing the event generation register (TIMx_SWEVGR).

14.3.3 Compare Output Mode

The compare output mode is one of basic functions of timer. The principle of the compare output mode is to output a specific change or waveform when the value of the core counter (CNT) is consistent with the value of the

compare/capture register. OCxM (in R16_TIMx_CHCTLRx) and the CCxP bit (in R16_TIMx_CCER) determine whether the output is determined high or low level or level inversion. When a comparison consistent event is generated, the CCxIF bit will be also set. If the CCxIE bit is preset, an interrupt will be generated; if the CCxDE bit is preset, a DMA request will be generated.

The procedure of compare output mode configuration is as follows:
1) Configure the clock source and auto-reload value of the core counter (CNT);
2) Set the count value to be compared to the compare/capture register (R16_TIMx_CHxCVR);
3) If an interrupt needs to be generated, set the CCxIE bit;
4) Keep OCxPE as 0 and disable the preload register of the compare register;
5) Set the output mode, and set OCxM and CCxP bit;
6) Enable the output and set the CCxE bit;
7) Set the CEN bit to start the timer.

14.3.4 Forced Output Mode

The output mode of the compare/capture channel of the timer can be forced to output a certain level by software, instead of relying on the shadow register and the core counter of the compare/capture register.

The specific means is to set OCxM to 100b, which means to force OCxREF to be low; or to set OCxM to 101b, which means setting OCxREF to a high value by force.

It should be noted that if OCxM is set to 100b or 101b by force, the comparison process between the internal core counter and the compare/capture register will be still in progress, the corresponding flag bit will be still set, and interrupts and DMA request will still be generated.

14.3.5 PWM Input Mode

The PWM input mode is used to measure the duty cycle and frequency of the PWM, which is a special case of the input capture mode. The operation is the same as the input capture mode except for the following differences: PWM occupies 2 compare/capture channels, and the input polarity of the 2 channels is set to opposite. One of the signals is set to trigger input, and SMS is set to reset mode.

For example, to measure the cycle and frequency of the PWM wave input from TI1, the following operations are required:
1) Set TI1 (TI1FP1) as the input of IC1 signal. Set CC1S to 01b;
2) Set TI1FP1 as the rising edge valid. Keep CC1P to 0;
3) Set TI1 (TI1FP2) as the input of IC2 signal. Set CC2S to 10b;
2) Set TI1FP2 as the falling edge valid. Set CC2P to 1;
5) The source of the clock source is TI1FP1. Set TS to 101b;
6) Set SMS to reset mode, i.e., 100b;
7) Enable the input capture. Set CC1E and CC2E bits;
In this way, the value of the compare/capture register 1 is the cycle of PWM, and the value of the compare/capture register 2 is its duty cycle.

Note: Since only TI1FP1 and TI2FP2 are connected to the slave mode controller, only TIM1_CH1/TIM1_CH2 can be used for PWM input mode.

14.3.6 PWM Output Mode

The PWM output mode is one of basic functions of timer. The most common method of PWM output mode is to use the reload value to determine the PWM frequency, and to use the capture comparison register to determine the duty cycle. Set 110b or 111b in OCxM to use PWM mode 1 or mode 2, set the OCxPE bit to enable the preload register, and finally set the ARPE bit. Since the value of the preload register can be sent to the shadow register when an update event occurs, it is necessary to set the UG bit to initialize all registers before the core counter starts counting. In the PWM mode, the core counter and the compare/capture register are always being compared. According to the CMS bit, the timer can output edge-aligned or center-aligned PWM signals.

- Edge alignment

When the edge alignment is used, the core counter counts up or down. In the scenario of PWM mode 1, when the value of the core counter is greater than that of the compare/capture register, OCxREF will be high; when the value of the core counter is less than the compare capture register (such as When the core counter increases to the value of R16_TIMx_ATRLR and returns to all 0s), OCxREF drops to low.

- Central alignment

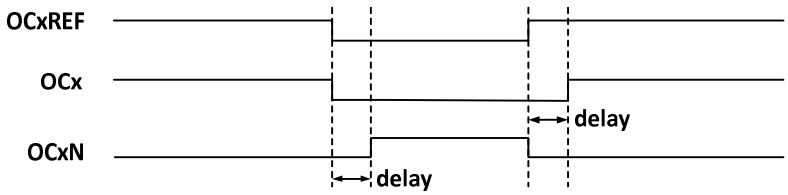
When the center-aligned mode is used, the core counter will run in a mode where up counting and down counting are performed alternately, and OCxREF performs rising and falling jumps when the values of the core counter and the compare/capture register are consistent. However, in 3 types of central alignment mode of comparison flag, the bit setting timing is different somewhat. When the center-alignment mode is used, it is the best to generate a software update flag (setting the UG bit) before starting the core counter.

14.3.7 Complementary Output and Deadband

The compare/capture channel generally has 2 output pins (compare/capture channel 4 has only one output pin), to output 2 complementary signals (OCx and OCxN). OCx and OCxN can be independently set by the CCxP and CCxNP bits. The output enable is set independently through CCxE and CCxNE, and the deadband and other controls are performed through the MOE, OIS, OISN, OSSI and OSSR bits. Meanwhile, OCx and OCxN outputs are enabled to insert into the deadband, each channel has a 10-bit deadband generator. If there is a break circuit, set the MOE bit. OCx and OCxN are generated by OCxREF in association. If OCx and OCxN are both high and effective, then OCx will be the same as OCxREF, but the rising edge of OCx is equivalent to OCxREF with a delay. OCxN is opposite to OCxREF, and its rising edge has a delay relative to the falling edge of the reference signal, and if the delay is greater than the effective output width, the corresponding pulse will not be generated.

Figure 14-4 shows the relationship between OCx, OCxN and OCxREF, and shows the deadband.

Figure 14-4 Complementary output and deadband



14.3.8 Break Signal

When the break signal is generated, the output enable signal and the invalid level will be modified according to the MOE, OIS, OISN, OSSI and OSSR bits. But OCx and OCxN will not be at the effective level at any time. The break event source can come from the break input pin, or it can be a clock failure event, and the clock failure event will

be generated by CSS (Clock Security System).

After the system reset, the break function will be disabled by default (MOE bit is low). Setting the BKE bit can enable the break function. The polarity of the input break signal can be set by setting BKP. The BKE and BKP signals can be written at the same time. There will be an APB clock delay before the actual write, so you need to wait for an PB cycle to read the written value correctly.

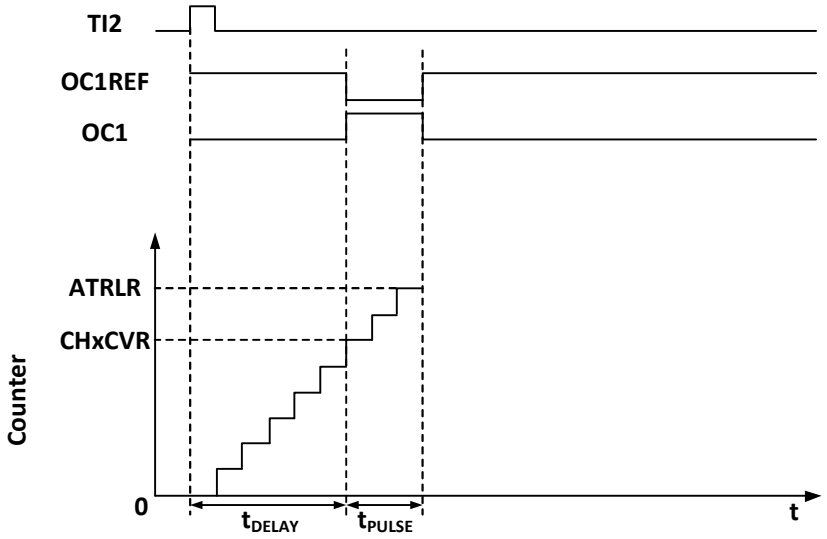
When the selected level appears on the break pin, the system will generate the following actions:
1) The MOE bit is asynchronously cleared, and the output is set to the invalid status, idle status or reset status according to the setting of the SOOI bit;
2) After MOE is cleared, each output channel will output the level determined by OISx;
3) During the supplementary output: the output will be in an invalid status, depending on the polarity;
4) If BIE is set, an interrupt will be generated when BIF is set; if the BDE bit is set, a DMA request will be generated;
5) If AOE is set, the MOE bit will be automatically set during the next update of event UEV.

14.3.9 Single Pulse Mode

The single pulse mode can be used to allow the microcontroller to respond to a specific event to generate a pulse after a delay. The delay and pulse width are programmable. Setting the OPM bit can make the core counter stop when the next update event UEV is generated (the counter turns over to 0).

As shown in Figure 14-5, it is necessary to detect the beginning of a rising edge on the TI2 input pin. After delaying Tdelay, a positive pulse of length Tpulse will be generated on OC1:

Figure 14-5 Single pulse generation



1) Set TI2 as trigger. Set CC2S to 01b and map TI2FP2 to TI2; set CC2P bit to 0b and set TI2FP2 to rising edge detection; set TS to 110b and set TI2FP2 as the trigger source; set SMS to 110b, and TI2FP2 is used to start the counter;
2) Tdelay is determined by the value of the compare/capture register, and Tpulse is determined by the value of the auto-reload value register and the value of the compare/capture register.

14.3.10 Encoder Mode

The encoder mode is a typical application of the timer. It can be used to access the dual-phase output of the encoder. The counting direction of the core counter is synchronized with the rotating shaft of the encoder. Each pulse

outputted by the encoder will increase the core counter by adding one or subtracting one. The steps to use the encoder are: set the SMS field to 001b (counting only on TI2 edge), 010b (counting only on TI1 edge) or 011b (counting on both TI1 and TI2 edges), and connect the encoder to compare/capture channel 1, 2 input terminals, set a value for the reload value register and this value can be set to be greater. In the encoder mode, the internal compare/capture register of timer, prescaler, repeat count register, etc. all work normally. The following table shows the relationship between the counting direction and the encoder signal.

Table 14-1 Relationship between counting direction of timer encoder mode and encoder signal

Counting active edge	Relative signal level	TI1FP1 signal edge		TI2FP2 signal	
		Rising edge	Falling edge	Rising edge	Falling edge
Only count at TI1 edge	High	Downcount	Upcount	Not count	
	Low	Upcount	Downcount		
Only count at TI2 edge	High	Not count		Upcount	Downcount
	Low			Downcount	Upcount
Count on both edges of TI1 and TI2	High	Downcount	Upcount	Upcount	Downcount
	Low	Upcount	Downcount	Downcount	Upcount

14.3.11 Synchronization of TIMx Timers and External Triggers

The timer can be synchronized with an external trigger in reset mode, gated mode and trigger mode.

Slave mode: reset mode

The counter and its prescaler can be reinitialized in response to a trigger input event; if the URS bit of the R16_TIMx_CTLR1 register is low, an update event UEV is generated; all preload registers (R16_TIMx_ATRLR, R16_TIMx_CHxCVR) are then updated.

In the following example, the up counter is cleared to zero when a rising edge occurs on the TI1 input:

1) Configure channel 1 to detect the rising edge of TI1. Configure the input filter bandwidth (this example does not require any filter, so keep IC1F = 0000). There is no need to configure the capture divider as it is not used for trigger operation. the CC1S bit selects only the input capture source, i.e. CC1S=01 (in R16_TIMx_CCMR1). Write CC1P=0 and CC1NP='0' to the R16_TIMx_CCER register to verify polarity (rising edge detection only).

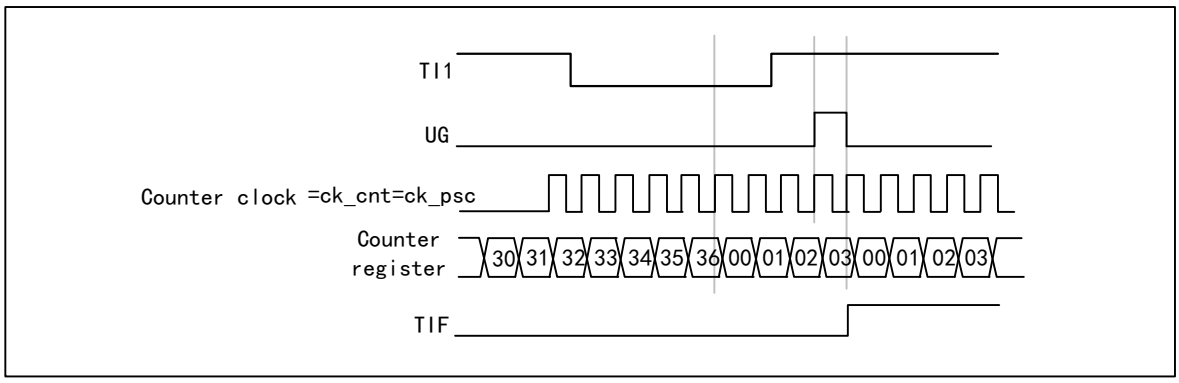
2) Write SMS=100 to R16_TIMx_SMCFGR to configure the timer to reset mode; write TS=101 to R16_TIMx_SMCFGR to select TI1 as input source.

3) Write CEN=1 to R16_TIMx_CTLR1 to start the counter.

The counter counts using the internal clock and then runs normally, when there is a TI1 rising edge the counter clears and starts counting again from 0. At the same time, the trigger flag TIF position 1, after enabling interrupt or DMA, allows an interrupt or DMA request to be sent. (Depends on the TIE (Interrupt Enable) bit and TDE (DMA Enable) bit in the R16_TIMx_DMAINTENR register).

The diagram below shows the action when the auto-reload register R16_TIMx_ARR = 0x36. The delay between the rising edge of TI1 and the actual counter reset is caused by the resynchronization circuitry at the TI1 input.

Figure 14-6 Control circuit in reset mode



Slave mode: Gated mode

The level of the input signal enables the counter. In the following example, the counter counts up only when TI1 is low:

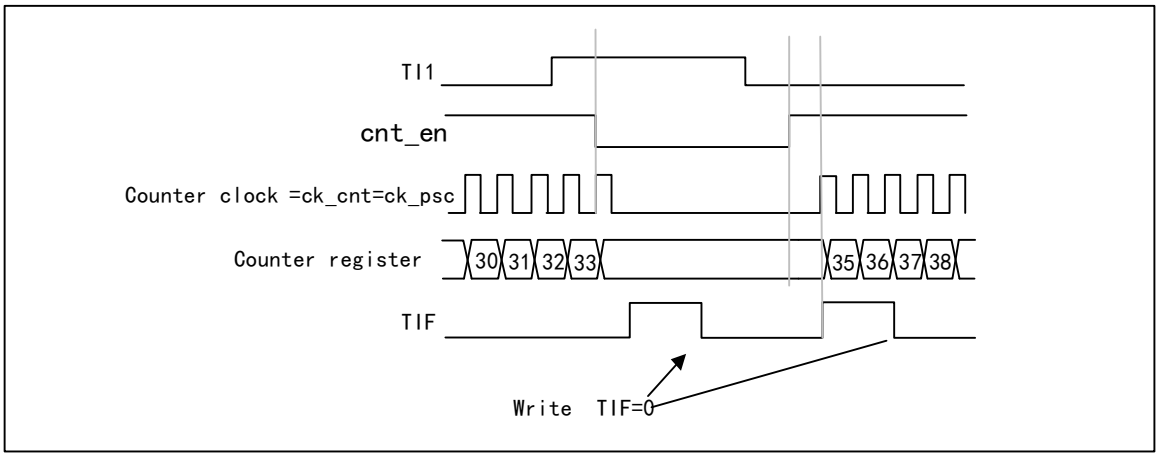
1) Configure channel 1 to detect a low level on TI1. Configure the input filter bandwidth (this example does not require any filter, so keep IC1F = 0000). There is no need to configure the capture divider as it is not used for trigger operation. the CC1S bit selects only the input capture source, i.e. CC1S=01 (in R16_TIMx_CCMR1). Write CC1P=1 and CC1NP='0' to the R16_TIMx_CCER register to verify polarity (detect low level only).

2) Write SMS=101 to R16_TIMx_SMCFGR to configure the timer for gated mode; write TS=101 to R16_TIMx_SMCFGR to select TI1 as input source.

3) Write CEN=1 to R16_TIMx_CTLR1 to start the counter. In gated mode, if CEN=0, the counter will not start regardless of the trigger input level.

As long as TI1 is low, the counter starts counting based on the internal clock and stops counting when TI1 goes high. When the counter starts or stops it sets TIF position 1 in R16_TIMx_INTFR. The delay between the rising edge of TI1 and the actual counter reset is caused by the resynchronization circuitry at the TI1 input.

Figure 14-7 Control circuit in gated mode



Slave mode: Trigger mode

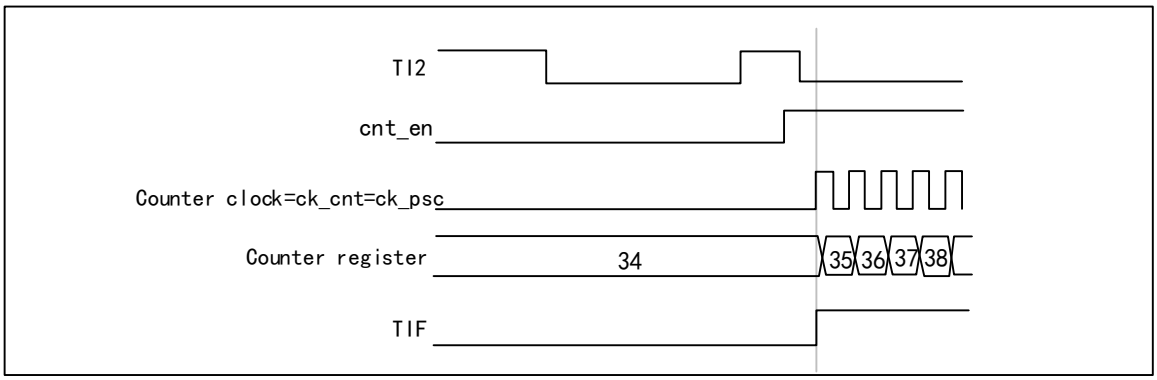
An event on the selected input will enable the counter. In the following example, the up counter is activated when a rising edge occurs on the TI2 input:

1) Configure channel 2 to detect the rising edge of TI2. Configure the input filter bandwidth (this example does not

require any filter, so keep IC2F = 0000). There is no need to configure the capture divider as it is not used for trigger operation. the CC2S bit selects only the input capture source by setting CC2S=01 (in R16_TIMx_CCMR1). Write CC2P=1 and CC2NP='0' to the R16_TIMx_CCER register to verify polarity (detect low level only)
2) Write SMS=110 to R16_TIMx_SMCFGR register to configure the timer to trigger mode; write TS=110 to R16_TIMx_SMCFGR register to select TI2 as input source.

When there is a rising edge of TI2, the counter starts counting driven by the internal clock, while TIF is set to 1. The delay between the rising edge of TI2 and the actual counter start is caused by the resynchronization circuitry at the TI2 input.

Figure 14-8 Control circuit in trigger mode

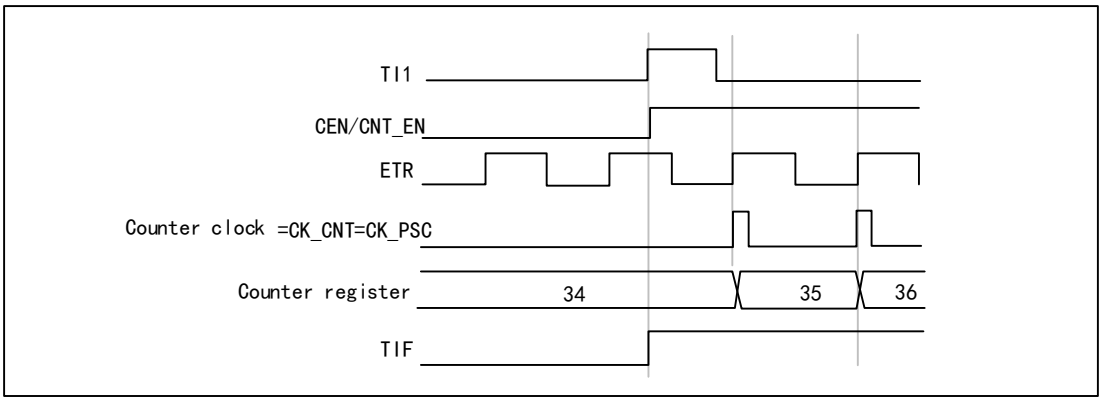


Slave mode: external clock mode 2 + trigger mode
External clock mode 2 can be used in conjunction with another slave mode in addition to external clock mode 1 and encoder mode. In this case, the ETR signal is used as an input to the external clock and the other input can be used as a trigger input in reset mode, gated mode or trigger mode. It is not recommended to select ETR as TRGI via the TS bit of the R16_TIMx_SMCFGR register. in the following example, the up counter is incremented on each rising edge of ETR as soon as a rising edge occurs on TI1:

1) Configuring the R16_TIMx_SMCFGR register to configure the external trigger input circuit:
-ETF=0000: no filtering;
—ETPS=00: no prescaler used;
—ETP=0: detect the rising edge of ETR, set ECE=1 to enable external clock mode 2.
2) Configure channel 1 to detect the rising edge of TI:
—IC1F=0000: no filtering;
—no need to configure the capture divider as it is not used for trigger operation;
-setting CC1S=01 in the R16_TIMx_CHCTLR1 register to select the input capture source;
- set CC1P=0 in the R16_TIMx_CCER register to determine the polarity (only rising edges are detected).
3) Write SMS=110 to R16_TIMx_SMCFGR register to configure the timer to trigger mode. Write TS=101 to the R16_TIMx_SMCFGR register to select TI1 as the input source.

When a rising edge occurs on TI1, the counter is enabled, TIF is set to 1 and the counter starts counting on the rising edge of ETR. the delay between the rising edge of the ETR signal and the actual counter reset is caused by the resynchronization circuit at the ETRP input.

Figure 14-9 Control circuit in external clock mode 2 + trigger mode

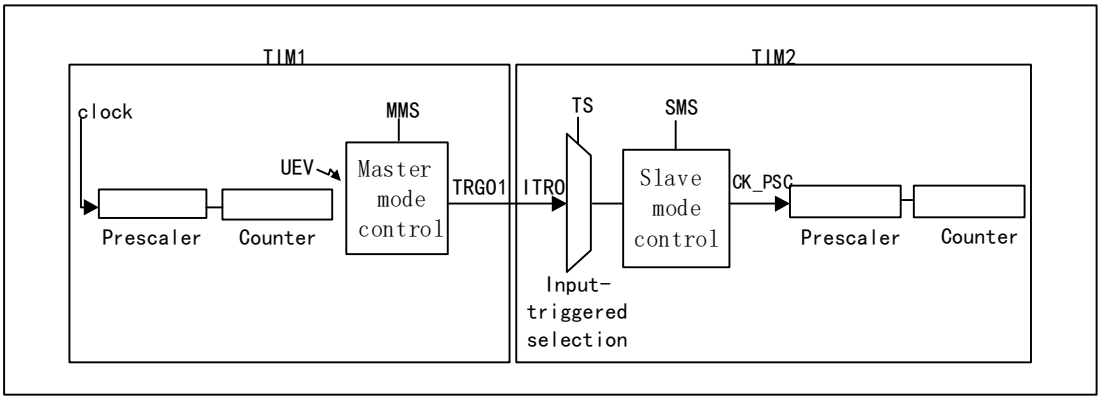


14.3.12 Timer Synchronization Mode

The TIMx timers are connected together from within to synchronize or cascade the timers. When a timer is configured in master mode, the counter of another timer configured in slave mode can be reset, started, stopped or clocked.

Using one timer as a prescaler for another timer

Figure 14-10 Master/slave timer example



For example, timer 1 can be configured as a prescaler for timer 2. To do this:

1) Configure Timer 1 in Master mode to output a periodic trigger signal every time an update event UEV occurs. If MMS=010 is written to R16_TIM1_CTLR2, TRGO1 will output a rising edge whenever an update event is generated.

2) To connect the TRGO1 output of Timer 1 to Timer 2, Timer 2 must be configured in slave mode, using ITR0 as the internal trigger. This can be selected via the TS bit in the R16_TIM2_SMCFGR register (write TS=000).

3) The slave mode controller is then set to external clock mode 1 (write SMS=111 in the R16_TIM2_SMCFGR register). In this way the clock for timer 2 will be supplied by the rising edge of the periodic trigger signal of timer 1 (corresponding to the counter overflow of timer 1).

4) Finally both timers must be enabled simultaneously by setting the corresponding CEN bits (R16_TIMx_CTLR1 register) of both timers to 1.

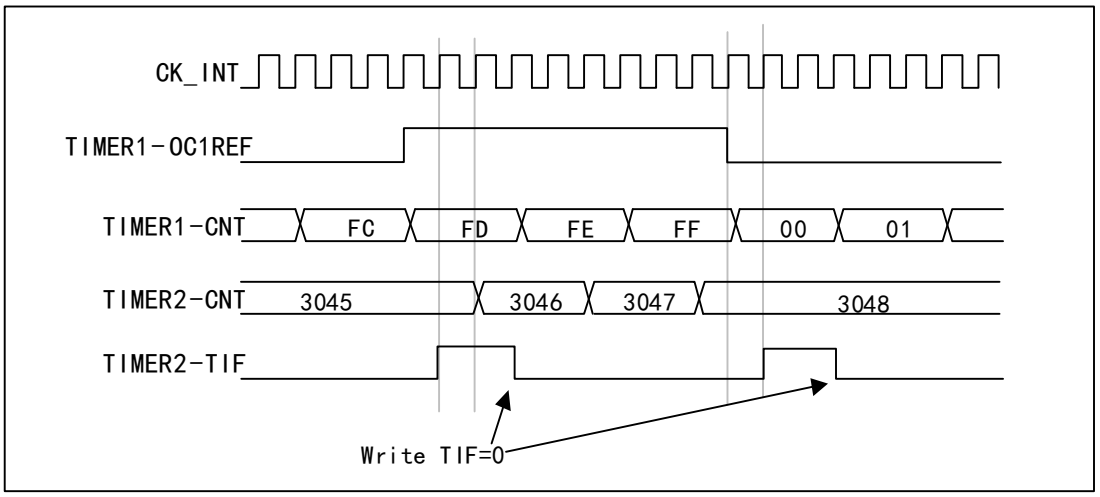
Note: If the OCx signal of timer 1 is selected as trigger output (MMS=1xx), the rising edge of this signal will be used to drive the counter of timer 2.

Using one timer to enable another timer

In this example Timer 2 is enabled by comparing the output of Timer 1 with 1. Timer 2 counts according to the divided internal clock only when OC1REF of Timer 1 is high. The clock frequency of both counters is based on CK_INT by prescaler performing a 3-way frequency (fCK_CNT=fCK_INT/3).

1) Configure timer 1 in main mode and send its output compare 1 reference signal (OC1REF) as trigger output (MMS=100 in R16_TIM1_CR2 register).
2) Configure Timer 1 for OC1REF waveform (R16_TIM1_CCMR1 register).
3) Configure Timer 2 to receive an input trigger from Timer 1 (TS=000 in R16_TIM2_SMCFGR register).
4) Configure Timer 2 to gated mode (SMS=101 in the R16_TIM2_SMCFGR register).
5) Enable Timer 2 by writing a "1" to the CEN bit (R16_TIM2_CTLR1 register).
6) Enable Timer 1 by writing "1" to the CEN bit (R16_TIM1_CTLR1 register).
Note: The clock of Counter 2 is not synchronized with Counter 1, this mode only affects the counter enable signal of Timer 2.

Figure 14-11 Gating Timer 2 using OC1REF of Timer 1

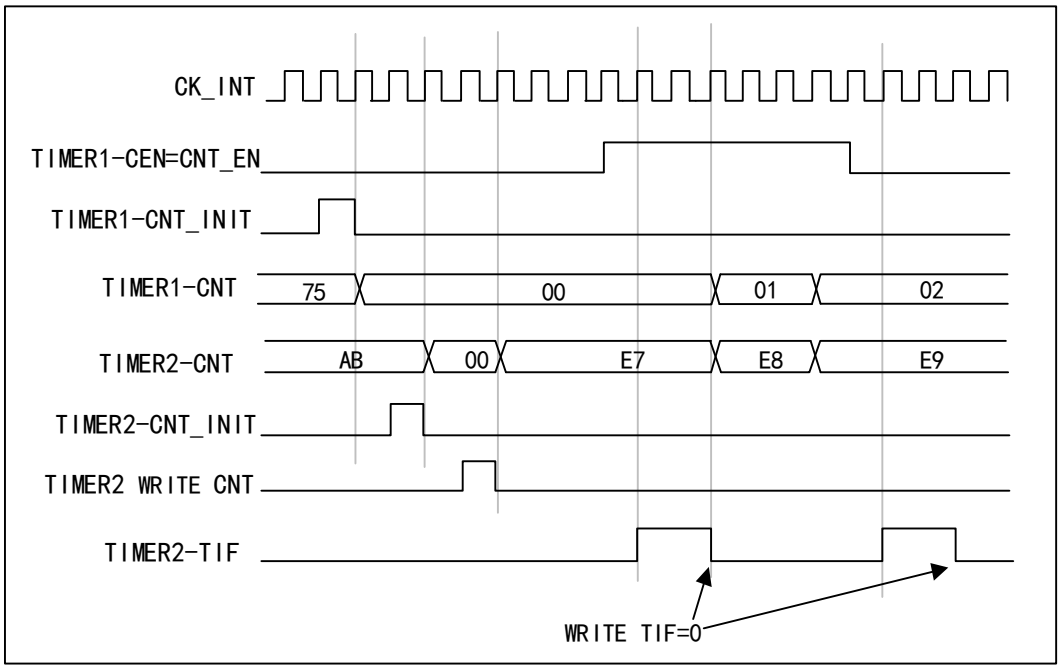


The counter and prescaler of timer 2 are not initialized before start-up. Therefore counting starts from the respective current value. Before starting Timer 1, both timers can be counted from the specified value by resetting them. This makes it possible to write any desired value into the timer counter. Both timers can be easily reset by software using the UG bit in the R16_TIMx_SWEVGR register.

In the next example, timer 1 is synchronized with timer 2. Timer 1 is in master mode and counts from 0. Timer 2 is in slave mode and counts from 0xE7. Both timers have the same prescale ratio. When timer 1 is disabled by writing "0" to the CEN bit in the R16_TIM1_CTLR1 register, timer 2 will stop:
1) Configure Timer 1 in main mode and send its output compare 1 reference signal (OC1REF) as the trigger output (MMS=100 in the R16_TIM1_CTLR2 register).
2) Configure Timer 1 for the OC1REF waveform (R16_TIM1_CHCTLR1 register).
3) Configure Timer 2 to receive an input trigger from Timer 1 (TS=000 in R16_TIM2_SMCFGR register).
4) Configure Timer 2 to gated mode (SMS=101 in the R16_TIM2_SMCFGR register).
5) Reset Timer 1 by writing "1" to the UG bit (R16_TIM1_SWEVGR register).
6) Reset Timer 2 by writing "1" to the UG bit (R16_TIM2_SWEVGR register).
7) Initialize Timer 2 to 0xE7 by writing "0xE7" to Timer 2's counter (R16_TIM2_CNTL).
8) Enable Timer 2 by writing "1" to the CEN bit (R16_TIM2_CTLR1 register).
9) Enable Timer 1 by writing "1" to the CEN bit (R16_TIM1_CTLR1 register).

10) Stop Timer 1 by writing "0" to the CEN bit (R16_TIM1_CTLR1 register).

Figure 14-12 Gating Timer 2 using Timer 1's enable signal

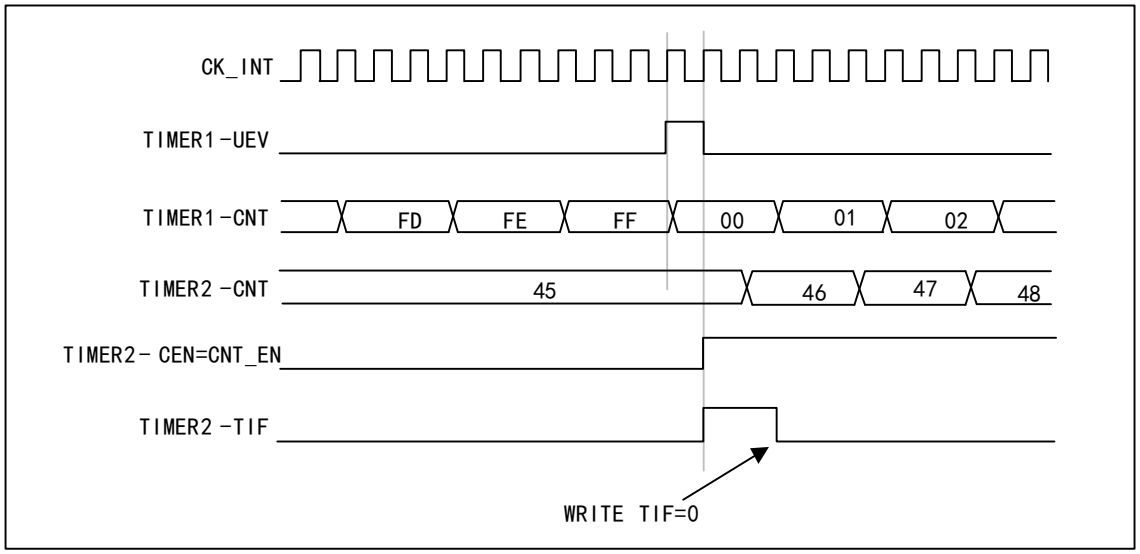


Using one timer to start another timer

This example uses the update event of Timer 1 to enable Timer 2. As soon as Timer 1 generates an update event, Timer 2 starts counting from the current value (which may not be 0) according to the internal clock after dividing the frequency. When Timer 2 receives a trigger signal, its CEN bit is automatically set to 1 and the counter starts counting until a "0" is written to the CEN bit of the R16_TIM2_CTLR1 register and the counter stops counting. The clock frequency of both counters is based on CK_INT and is divided by 3 through a prescaler (fCK_CNT=fCK_INT/3).

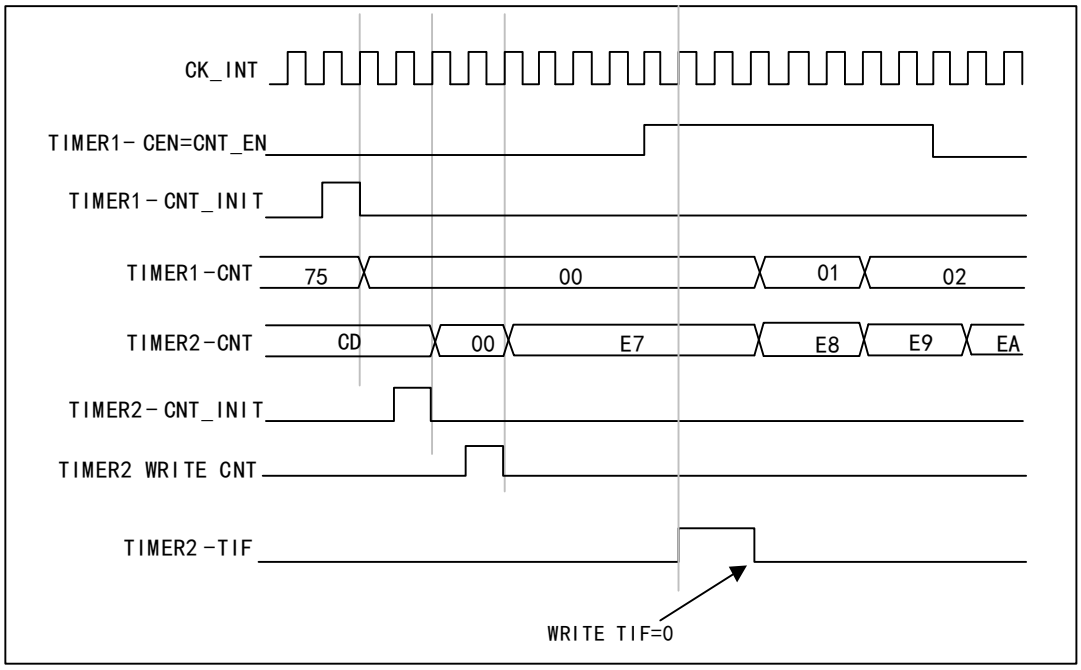
1) Configure timer 1 in main mode and send its update event (UEV) as trigger output (MMS=010 in R16_TIM1_CTLR2 register).
(MMS=010 in R16_TIM1_CTLR2 register).
2) Configure the period of timer 1 (R16_TIM1_ATRLR register).
3) Configure Timer 2 to receive an input trigger from Timer 1 (TS=000 in the R16_TIM2_SMCFGR register).
4) Configure Timer 2 to trigger mode (SMS=110 in the R16_TIM2_SMCFGR register).
5) Start Timer 1 by writing "1" to the CEN bit (R16_TIM1_CTLR1 register).

Figure 14-13 Triggering Timer 2 using Timer 1 update event



As shown in the example above, the user can initialize both counters before starting to count. Figure 14-14 shows the counting behaviour with the same configuration as Figure 14-13, except in trigger mode (SMS=110 in the R16_TIM2_SMCFGR register) rather than gated mode.

Figure 14-14 Triggering Timer 2 using Timer 1's enable signal



Using one timer as a prescaler for another timer

For example, timer 1 can be configured as a prescaler for timer 2. To do this:

1) Configure Timer 1 in main mode, sending its update event (UEV) as a trigger output (MMS=010 in the R16_TIM1_CTLR2 register). This will then output a periodic signal each time the counter overflows.

2) Configure the period of timer 1 (R16_TIM1_ATRLR register).

3) Configure Timer 2 to receive an input trigger from Timer 1 (TS=000 in the R16_TIM2_SMCFGR register).

4) Configure Timer 2 for external clock mode (SMS=111 in the R16_TIM2_SMCFGR register).

5) Start Timer 2 by writing a "1" to the CEN bit (R16_TIM2_CTLR1 register).

6) Timer 1 is started by writing "1" to the CEN bit (R16_TIM1_CTLR1 register).

Synchronized start of 2 timers using an external trigger

In this example, Timer 1 is enabled when there is a rising edge on the TI1 input of Timer 1, and Timer 2 is enabled at the same time as Timer 1. To ensure that the two counters are aligned, Timer 1 must be configured in master/slave mode (corresponding to TI1 as slave and Timer 2 as master):

1) Configure Timer 1 in Master mode, sending its enable signal as a trigger output (MMS=001 in the R16_TIM1_CTLR2 register).

2) Configure Timer 1 as Slave mode to receive the input trigger from TI1 (TS=100 in R16_TIM1_SMCFGR register).

3) Configure Timer 1 to trigger mode (SMS=110 in the R16_TIM1_SMCFGR register).

4) Configure Timer 1 to Master/Slave mode by writing MSM=1 (R16_TIMx_SMCR register).

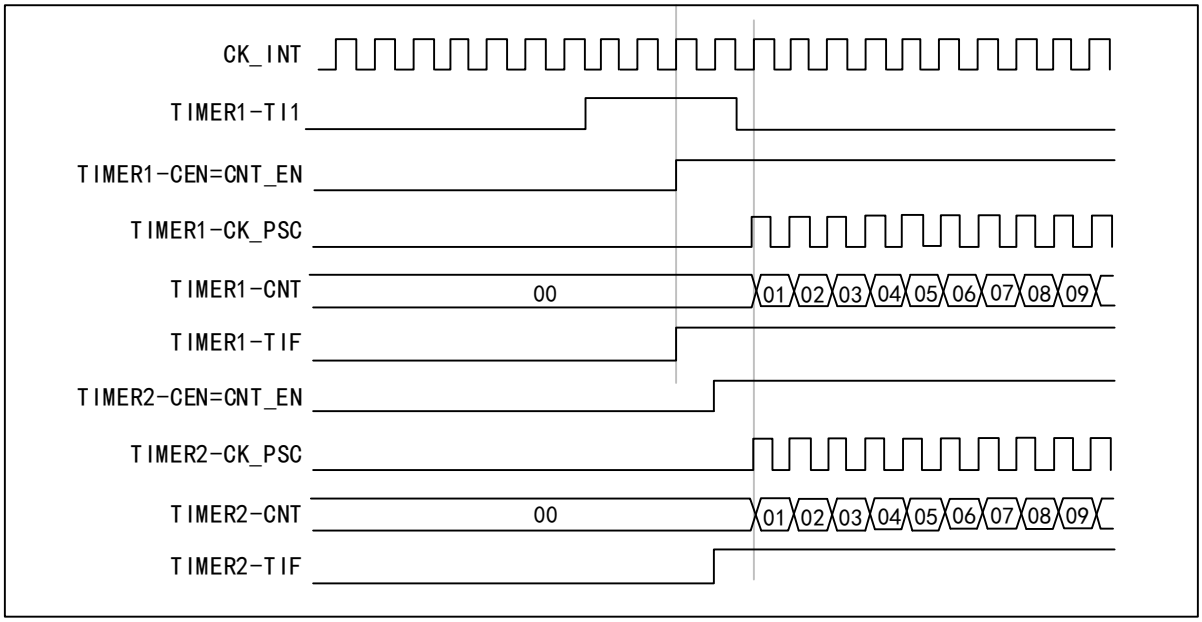
5) Configure Timer 2 to receive an input trigger from Timer 1 (TS=000 in R16_TIM2_SMCFGR register).

6) Configure Timer 2 to trigger mode (SMS=110 in the R16_TIM2_SMCFGR register).

When a rising edge occurs on TI1 (Timer 1), both counters start counting synchronously according to the internal clock and both TIF flags are set to 1.

Note: In this example, both timers are initialized (by setting their respective UG positions to 1) prior to start-up. Both counters count from 0, but an offset can easily be inserted between the two by writing to either counter register (R16_TIMx_CNT). It may be noted that the master/slave mode creates a delay between CNT_EN and CK_PSC for timer 1.

Figure 14-15 Triggering Timer 1 and Timer 2 using Timer 1's TI1 input



Timers are capable of outputting clock pulses (TRGO) and also receiving inputs from other timers (ITRx). The source of ITRx (TRGO from other timers) is different for different timers. The timer internal trigger connections are shown in Table 14-2.

Table 14-2 TIMx internal trigger connections

Slave timer	ITR0(TS=000)	ITR1(TS=001)	ITR2(TS=010)	ITR3(TS=011)
TIM1	-	TIM2	TIM3	TIM4

14.3.13 Debug Mode

When the system enters debug mode, the timer continues to run or stops according to the settings of the DBG module.

14.4 Register Description

Table 14-3 TIM1-related registers list

Name	Access address	Description	Reset value
R16_TIM1_CTLR1	0x40012C00	Control register 1	0x0000
R16_TIM1_CTLR2	0x40012C04	Control register 2	0x0000
R16_TIM1_SMCFGR	0x40012C08	Slave mode control register	0x0000
R16_TIM1_DMAINTENR	0x40012C0C	DMA/interrupt enable register	0x0000
R16_TIM1_INTFR	0x40012C10	Interrupt status register	0x0000
R16_TIM1_SWEVGR	0x40012C14	Event generation register	0x0000
R16_TIM1_CHCTLR1	0x40012C18	Compare/capture control register 1	0x0000
R16_TIM1_CHCTLR2	0x40012C1C	Compare/capture control register 2	0x0000
R16_TIM1_CCER	0x40012C20	Compare/capture enable register	0x0000
R16_TIM1_CNT	0x40012C24	Counters	0x0000
R16_TIM1_PSC	0x40012C28	Counting clock prescaler	0x0000
R16_TIM1_ATRLR	0x40012C2C	Auto-reload value register	0xFFFF
R16_TIM1_RPTCR	0x40012C30	Repeat Count Register	0x0000
R32_TIM1_CH1CVR	0x40012C34	Compare/capture register 1	0x00000000
R32_TIM1_CH2CVR	0x40012C38	Compare/capture register 2	0x00000000
R32_TIM1_CH3CVR	0x40012C3C	Compare/capture register 3	0x00000000
R32_TIM1_CH4CVR	0x40012C40	Compare/capture register 4	0x00000000
R16_TIM1_BDTR	0x40012C44	Brake and deadband registers	0x0000
R16_TIM1_DMACFGR	0x40012C48	DMA control register	0x0000
R32_TIM1_DMAADR	0x40012C4C	DMA address register for continuous mode	0x00000000

14.4.1 Control Register 1 (TIM1_CTLR1)

Offset address: 0x00

15	14	13	12	11 10	9 8	7	6 5	4	3	2	1	0
TMR_CAP_LVL_EN	TMR_CAP_OV_EN	Reserved	BKSEL	Reserved	CKD[1:0]	ARPE	CMS[1:0]	DIR	OPM	URS	UDIS	CEN

Bit	Name	Access	Description	Reset value
15	TMR_CAP_LVL_EN	RW	In double edge capture mode, capture level indication enables: 0: Disable the indication function. 1: Enable indicates function. *Note: after enabling, [16] of CHxCVR indicates the level corresponding to the capture value.*	0
14	TMR_CAP_OV_EN	RW	Capture value mode configuration.	0

			0: The capture value is the value of the actual counter 1: The CHxCVR value is 0xFFFF when a counter overflow is generated before capture.	
13	Reserved	RO	Reserved	0
12	BKSEL	RW	TIM1 Brake input source selection: 0: Brake comes from IO or OPA; 1: Brake signal comes from the comparator (if CMP3 is enabled, the brake comes from the output of CMP3; if CMP3 is not enabled and CMP2 is enabled, the brake comes from the output of CMP2; if CMP3 and CMP2 are not enabled and CMP1 is enabled, the brake comes from the output of CMP1)	0
[11:10]	Reserved	RO	Reserved	0
[9:8]	CKD[1:0]	RW	These 2 bits define the division ratio between the timer clock (CK_INT) frequency, the dead time and the sampling clock used by the dead time generator and the digital filter (ETR,TIx). 00: Tdts=Tck_int 01: Tdts = 2 × Tck_int 10: Tdts = 4 × Tck_int 11: Reserved.	00b
7	ARPE	RW	Auto-reload preload enable bit. 0: Disable Auto-reload Value Register (ATRLR). 1: Enable Auto-reload Value Register (ATRLR).	0
[6:5]	CMS[1:0]	RW	Central alignment mode selection. 00: Edge-aligned mode. The counter counts up or down based on the direction bit (DIR). 01: Central alignment mode 1. The counter counts up and down alternately. The output compare interrupt flag bit of the channel configured as output (CCxS=00 in the CHCTLRx register) is set only when the counter counts down. 10: Central alignment mode 2. The counter counts up and down alternately. The output compare interrupt flag bit of the channel configured as output (CCxS=00 in the CHCTLRx register) is set only when the counter counts up. 11: Central alignment mode 3. The counter counts up and down alternately. The output compare interrupt flag bit of the channel configured as output (CCxS=00 in the CHCTLRx register) is set when the counter counts both up and down. *Note: When the counter is enabled (CEN=1), the transition from edge-aligned mode to center-aligned*	00b

			mode is not allowed.	
4	DIR	RW	Counter direction: 0: The counting mode of the counter is increment. 1: The counting mode of the counter is subtractive. *Note: This bit is not valid when the counter is configured in central alignment mode or encoder mode.*	0
3	OPM	RW	Single pulse mode. 0: The counter does not stop when the next update event occurs. 1: The counter stops (clear the CEN bit) when the next update event occurs.	0
2	URS	RW	Update request source, by which the software selects the source of the UEV event. 0: If an update interrupt or DMA request is enabled, an update interrupt or DMA request is generated by any of the following events. -Counter overflow/underflow -Setting the UG position -Updates generated by the slave mode controller 1: If an update interrupt or DMA request is enabled, only an update interrupt or DMA request is generated if the counter overflows/underflows.	0
1	UDIS	RW	Updates are prohibited, and the software allows / disables the generation of UEV events through this bit. 0: UEV is allowed. The UEV event is generated by any of the following events: -Counter overflow / underflow. -Set the UG bit. Updates generated from the mode controller with cached registers are loaded into their preload values. 1: UEV is prohibited. No update events are generated, and the registers (ARR, PSC, CCRx) hold their values. If the UG bit is set or a hardware reset is issued from the mode controller, the counter and prescaler are reinitialized.	0
0	CEN	RW	Enables the counter. 0: Disable the counter. 1: Enable the counter. *Note: The external clock, gated mode and encoder mode will not work until the CEN bit is set in software. Trigger mode can automatically set the CEN bit in hardware.*	0

14.4.2 Control Register 2 (TIM1_CTLR2)

Offset address: 0x04

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS[2:0]			CCDS	CCUS	Reserved	CCPC

Bit	Name	Access	Description	Reset value
15	Reserved	RO	Reserved	0
14	OIS4	RW	Output idle state 4. 1: When MOE=0, if OC4N is implemented, OC1=1 after deadband; 0: When MOE=0, if OC4N is implemented, OC1=0 after deadband. *Note: This bit cannot be modified after LOCK (TIMx_BDTR register) level 1, 2 or 3 has been set.*	0
13	OIS3N	RW	Output idle state 3. 1: OC1N = 1 after the dead zone when MOE = 0. 0: When MOE=0, OC1N=0 after dead zone. *Note: This bit cannot be modified after the LOCK (TIMx_BDTR register) level 1, 2 or 3 has been set.*	0
12	OIS3	RW	Output idle state 3, see OIS4.	0
11	OIS2N	RW	Output idle state 2, see OIS3N.	0
10	OIS2	RW	Output idle state 2, see OIS4.	0
9	OIS1N	RW	Output idle state 1, see OIS3N.	0
8	OIS1	RW	Output idle state 1, see OIS4.	0
7	TI1S	RW	TI1 selection. 1: TIMx_CH1, TIMx_CH2 and TIMx_CH3 pins connected to TI1 input after heterodyning. 0: TIMx_CH1 pin is connected directly to TI1 input.	0
[6:4]	MMS[2:0]	RW	Master mode selection: These 3 bits are used to select the synchronization information (TRGO) sent to the slave timer in master mode. The possible combinations are as follows. 000: The UG bit of the Reset-TIMx_EGR register is used as the trigger output (TRGO). In the case of a reset generated by a trigger input (from a mode controller in reset mode), there is a delay in the signal on TRGO relative to the actual reset. 001: Enable - The counter enable signal CNT_EN is used as a trigger output (TRGO). Sometimes it is necessary to start multiple timers at the same time or to control the enable from timers over a period of time. The counter enable signal is generated by the logical or of the trigger input signal in CEN control bit and gated mode. When the counter enable signal is controlled by	000b

			a trigger input, there is a delay on TRGO unless master/slave mode is selected (see the description of the MSM bit in the TIMx_SMCR register). 010: Update - The update event is selected as a trigger input (TRGO). For example, the clock of a master timer may be used as a prescaler for a slave timer. 011: Comparison pulse - on the occurrence of a capture or a successful comparison, when the CC1IF flag is to be set (even if it is already high), the trigger output sends a positive pulse (TRGO). 100: The comparison-OC1REF signal is used as a trigger output (TRGO). 101: The comparison-OC2REF signal is used as a trigger output (TRGO). 110: The comparison-OC3REF signal is used as a trigger output (TRGO). 111: The comparison -OC4REF signal is used as the trigger output (TRGO).	
3	CCDS	RW	Capture the DMA selection for comparison. 1: Send a DMA request for CHxCVR when an update event occurs. 0: Generate a DMA request for CHxCVR when CHxCVR occurs.	0
2	CCUS	RW	Compare capture control update selection bits. 1: If CCPC is set, they can be updated by setting the COM bit or a rising edge on TRGI. 0: If the CCPC is set, they can only be updated by setting the COM bit. *Note: This bit only works for channels with complementary outputs.*	0
1	Reserved	RO	Reserved	0
0	CCPC	RW	Compare capture preload control bits. 1: CCxE, CCxNE and OCxM bits are preloaded and when this bit is set they are only updated when the COM bit is set. 0: CCxE, CCxNE and OCxM bits are not preloaded. *Note: This bit only works for channels with complementary outputs.*	0

14.4.3 Slave Mode Control Register (TIM1_SMCFGR)

Offset address: 0x08

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			Reserved	SMS[2:0]		

Bit	Name	Access	Description	Reset value
15	ETP	RO	ETR trigger polarity selection, this bit selects whether to input ETR directly or to input the inverse of ETR. 1: Invert ETR, low or falling edge active; 0: ETR, active high or rising edge.	0
14	ECE	RW	External clock mode 2 enable selection. 1: Enables external clock mode 2. 0: Disable external clock mode 2. *Note 1: Slave mode can be used simultaneously with external clock mode 2: reset mode, gated mode and trigger mode; however, TRGI cannot be connected to ETRF in this case (TS bit cannot be '111').* *Note 2: When both external clock mode 1 and external clock mode 2 are enabled, the external clock input is ETRF.*	0
[13:12]	ETPS[1:0]	RW	The external trigger signal (ETRP) divides the frequency of this signal, which cannot exceed a maximum of 1/4 of the TIMxCLK frequency, and can be downconverted through this domain. 00: Prescaler off. 01: ETRP frequency divided by 2. 10: ETRP frequency divided by 4. 11: ETRP frequency divided by 8.	00b
[11:8]	ETF[3:0]	RW	Externally triggered filtering, in fact, the digital filter is an event counter, which uses a certain sampling frequency to record up to N events and then produces a jump in the output. 0001: Sampling frequency Fsampling=Fck_int, N=2. 0010: Sampling frequency Fsampling=Fck_int, N=4. 0011: Sampling frequency Fsampling=Fck_int, N=8. 0100: Sampling frequency Fsampling = Fdts/2, N = 6. 0101: Sampling frequency Fsampling = Fdts/2, N = 8. 0110: Sampling frequency Fsampling = Fdts/4, N = 6. 0111: Sampling frequency Fsampling = Fdts/4, N = 8. 1000: Sampling frequency Fsampling = Fdts/8, N = 6. 1001: Sampling frequency Fsampling = Fdts/8, N = 8. 1010: Sampling frequency Fsampling = Fdts/16, N = 5. 1011: Sampling frequency Fsampling = Fdts/16, N = 6. 1100: Sampling frequency Fsampling = Fdts/16, N = 8. 1101: Sampling frequency Fsampling = Fdts/32, N = 5. 1110: Sampling frequency Fsampling = Fdts/32, N = 6. 1111: Sampling frequency Fsampling=Fdts/32, N=8.	0000b
7	MSM	RW	Master/slave mode selection. 1: The event on the trigger input (TRGI) is delayed to	0

			allow perfect synchronization between the current timer (via TRGO) and its slave timer. This is useful when the synchronization of several timers to a single external event is required. 0: Does not function.	
[6:4]	TS[2:0]	RW	Trigger selection field, these 3 bits select the trigger input source used to synchronize the counter. 000: Internal trigger 0 (ITR0). 001: Internal trigger 1 (ITR1). 010: Internal trigger 2 (ITR2). 011: Internal trigger 3 (ITR3). 100: Edge detector of TI1 (TI1F_ED). 101: Filtered timer input 1 (TI1FP1). 110: Filtered timer input 2 (TI2FP2). 111: External trigger input (ETRF). The above only changes when SMS is 0. *Note: See Table 14-2 for details.*	000b
3	Reserved	RO	Reserved.	0
[2:0]	SMS[2:0]	RW	Input mode selection field. Selects the clock and trigger mode of the core counter. 000: driven by the internal clock CK_INT. 001: Encoder mode 1, where the core counter increments or decrements the count at the edge of TI2FP2 depending on the level of TI1FP1. 010: Encoder mode 2, where the core counter increments or decrements the count at the edge of TI1FP1, depending on the level of TI2FP2. 011: Encoder mode 3, where the core counter increments and decrements the count on the edges of TI1FP1 and TI2FP2 depending on the input level of another signal; 100: reset mode, where the rising edge of the trigger input (TRGI) will initialize the counter and generate a signal to update the registers. 101: Gated mode, when the trigger input (TRGI) is high, the counter clock is turned on; at the trigger input becomes low, the counter is stopped, and the counter starts and stops are controlled. 110: Trigger mode, where the counter is started on the rising edge of the trigger input TRGI and only the start of the counter is controlled. 111: External clock mode 1, rising edge of the selected trigger input (TRGI) drives the counter.	000b

14.4.4 DMA/Interrupt Enable Register (TIM1_DMAINTENR)

Offset address: 0x0C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE

Bit	Name	Access	Description	Reset value
15	Reserved	RO	Reserved	0
14	TDE	RW	Trigger the DMA request enable bit. 1: Allow DMA requests to be triggered. 0: Triggering of DMA requests is disabled.	0
13	COMDE	RW	DMA request enable bit of COM. 1: Allow DMA requests for COM. 0: DMA request for COM is disabled.	0
12	CC4DE	RW	Compare the DMA request enable bit of capture channel 4. 1: Allow comparison of DMA requests for capture channel 4. 0: Disable comparison of DMA requests for capture channel 4.	0
11	CC3DE	RW	Compare the DMA request enable bit of capture channel 3. 1: Allow comparison of DMA requests for capture channel 3. 0: Disable comparison of DMA requests for capture channel 3.	0
10	CC2DE	RW	Compare the DMA request enable bit of capture channel 2. 1: Allow comparison of DMA requests for capture channel 2. 0: Disable comparison of DMA requests for capture channel 2.	0
9	CC1DE	RW	Compare the DMA request enable bit of capture channel 1. 1: Allow comparison of DMA requests for capture channel 1. 0: Disable comparison of DMA requests for capture channel 1.	0
8	UDE	RW	Updated DMA request enable bit. 1: DMA requests that allow updates. 0: DMA requests for updates are disabled.	0
7	BIE	RW	Brake interrupt enable bit. 1: Allow brakes to be interrupted. 0: Brake interruption is prohibited.	0

6	TIE	RW	Trigger the interrupt enable bit. 1: Enable triggering of interrupts. 0: Trigger interrupt is disabled.	0
5	COMIE	RW	COM interrupt allow bit. 1: Allow COM interrupts. 0: COM interrupt is disabled.	0
4	CC4IE	RW	Compare capture channel 4 interrupt enable bit. 1: Allow comparison of capture channel 4 interrupts. 0: Disable compare capture channel 4 interrupt.	0
3	CC3IE	RW	Compare capture channel 3 interrupt enable bit. 1: Allow comparison of capture channel 3 interrupts. 0: Disable compare capture channel 3 interrupt.	0
2	CC2IE	RW	Compare capture channel 2 interrupt enable bit. 1: Allow comparison of capture channel 2 interrupts. 0: Disable compare capture channel 2 interrupt.	0
1	CC1IE	RW	Compare capture channel 1 interrupt enable bit. 1: Allow comparison of capture channel 1 interrupts. 0: Disable compare capture channel 1 interrupt.	0
0	UIE	RW	Update the interrupt enable bit. 1: Allow updates to be interrupted. 0: Disable update interruption.	0

14.4.5 Interrupt Status Register (TIM1_INTFR)

Offset address: 0x10

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CC4OF	CC3OF	CC2OF	CC1OF	Reserved	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF

Bit	Name	Access	Description	Reset value
[15:13]	Reserved	RO	Reserved	0
12	CC4OF	RW0	Compare capture channel 4 repeat capture flag bit.	0
11	CC3OF	RW0	Compare capture channel 3 repeat capture flag bit.	0
10	CC2OF	RW0	Compare capture channel 2 repeat capture flag bit.	0
9	CC1OF	RW0	Compare capture channel 1 repeat capture flag bit is used only when the compare capture channel is configured for input capture mode. This flag is set by hardware and a software write of 0 clears this bit. 1: The value of the counter is captured into the capture comparison register when the status of CC1IF has been set. 0: No duplicate captures are generated.	0
8	Reserved	RO	Reserved	0
7	BIF	RW0	The brake interrupt flag bit, once the brake input is	0

			valid, by hardware for this position bit, can be cleared by software. 1: A set valid level is detected on the brake pin input. 0: No braking event is generated.	
6	TIF	RW0	Trigger interrupt flag bit, when a trigger event occurs by hardware to this location bit, by software to clear. Trigger events include the detection of a valid edge at the TRGI input from a mode other than gated, or any edge in gated mode. 1: Trigger event generation. 0: No trigger event is generated.	0
5	COMIF	RW0	COM interrupt flag bit, this bit is set by hardware and cleared by software once a COM event is generated. com events including CCxE, CCxNE, OCxM are updated. 1: COM event generation. 0: No COM event is generated.	0
4	CC4IF	RW0	Compare capture channel 4 interrupt flag bit.	0
3	CC3IF	RW0	Compare capture channel 3 interrupt flag bit.	0
2	CC2IF	RW0	Compare capture channel 2 interrupt flag bit.	0
1	CC1IF	RW0	Compare capture channel 1 interrupt flag bit. If the compare capture channel is configured in output mode. This bit is set by hardware when the counter value matches the comparison value, except in centrosymmetric mode. This bit is cleared by software. 1: The value of the core counter matches the value of compare capture register 1; 0: No match occurs. If compare capture channel 1 is configured as input mode. This bit is set by hardware when a capture event occurs, and it is cleared by software or by reading the compare capture register. 1: The counter value has been captured compare capture register 1. 0: No input capture is generated.	0
0	UIF	RW0	Update interrupt flag bit, this bit is set by hardware when an update event is generated and cleared by software. 1: Update interrupt generation. 0: No update event is generated. The following scenarios generate update events. If UDIS = 0, when the repeat counter value overflows or underflows.	0

			If URS = 0, UDIS = 0, when the UG bit is set, or when the counter core counter is reinitialized by software. If URS = 0, UDIS = 0, when the counter CNT is reinitialized by a trigger event.	

14.4.6 Event Generation Register (TIM1_SWEVGR)

Offset address: 0x14

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved.	0
7	BG	WO	The brake event generation bit, which is set and cleared by software, is used to generate a brake event. 1: Generate a brake event. At this point, MOE=0, BIF=1, if the corresponding interrupt and DMA are enabled, the corresponding interrupt and DMA are generated. 0: No action.	0
6	TG	WO	The trigger event generation bit, which is set by software and cleared by hardware, is used to generate a trigger event. 1: Generate a trigger event, TIF is set, and the corresponding interrupts and DMAs are generated if enabled. 0: No action.	0
5	COMG	WO	Compare capture control update generation bit. Generates a compare capture control update event. This bit is set by software and automatically cleared by hardware. 1: when CCPC = 1, allow updating of CCxE, CCxNE, OCxM bits. 0: No action. *Note: This bit is only valid for channels with complementary outputs (channels 1, 2, 3).*	0
4	CC4G	WO	Compare capture event generation bit 4. generates compare capture event 4.	0
3	CC3G	WO	Compare capture event generation bit 3. generates compare capture event 3.	0
2	CC2G	WO	Compare capture event generation bit 2. generates compare capture event 2.	0
1	CC1G	WO	Compare capture event generation bit 1. generates compare capture event 1.	0

			This bit is set by software and cleared by hardware. It is used to generate a compare capture event. 1: Generate a compare capture event on compare capture channel 1. If compare capture channel 1 is configured as output. Set the CC1IF bit. Generate the corresponding interrupts and DMAs if they are enabled. If compare capture channel 1 is configured as input. The current core counter value is captured to compare capture register 1; set the CC1IF bit to generate the corresponding interrupts and DMAs if they are enabled; if CC1IF is already set, set the CC1OF bit. 0: No action.	
0	UG	WO	Update event generation bit to generate an update event. This bit is set by software and is automatically cleared by hardware. 1: Initialize the counter and generate an update event. 0: No action. *Note: The prescaler counter is also cleared to zero, but the prescaler factor remains unchanged. The core counter is cleared if in centrosymmetric mode or incremental counting mode; if in decremental counting mode, the core counter takes the value of the reload value register.*	0

14.4.7 Compare/Capture Control Register 1 (TIM1_CHCTLR1)

Offset address: 0x18

The channel can be used in input (capture mode) or output (compare mode), and the direction of the channel is defined by the corresponding CCxS bit. The other bits of this register have different roles in input and output modes. OCxx describes the function of the channel in output mode and ICxx describes the function of the channel in input mode.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]				IC1PSC[1:0]			

Compare mode (pin direction is output).

Bit	Name	Access	Description	Reset value
15	OC2CE	RW	Compare capture channel 2 clear enable bit. 1: Clear OC2REF bit zero once ETRF input is detected high; 0: OC2REF is not affected by ETRF input.	0
[14:12]	OC2M[2:0]	RW	Compare Capture Channel 2 mode setting field.	0

			The 3 bits define the action of the output reference signal OC2REF, which determines the values of OC2, OC2N. OC2REF is active high, while the active levels of OC2 and OC2N depend on the CC2P, CC2NP bits. 000: Freeze. Comparison of the value of the capture register with the value of the comparison between the core counters does not work for OC2REF. 001: force to set to valid level. Forcing OC2REF high when the core counter has the same value as the comparison capture register 2. 010: Force to set to invalid level. Forcing OC2REF low when the value of the core counter is the same as the comparison capture register 2. 011: Flip. Flips the level of OC2REF when the core counter is the same as the value of compare capture register 2. 100: Forced to invalid level. Forces OC2REF to low. 101: Forced to valid level. Force OC2REF to high. 110: PWM Mode 1: In up count, once the core counter is smaller than the value of the compare capture register, channel 2 is valid, otherwise it is invalid; in downward counting, once the core counter is larger than the value of the compare capture register, channel 2 is invalid (OC2REF=0), otherwise it is valid (OC2REF=1) 111: PWM mode 2: In up count, once the core counter is smaller than the value of the compare capture register, channel 2 is invalid level, otherwise it is valid level; in down count, once the core counter is larger than the value of the compare capture register, channel 2 is valid level (OC2REF=1), otherwise it is invalid level (OC2REF=0). *Note: This bit cannot be modified once the LOCK level is set to 3 and CC1S=00b. In PWM mode 1 or PWM mode 2, the OC2REF level is changed only when the comparison result is changed or when switching from freeze mode to PWM mode in the output comparison mode.*	
11	OC2PE	RW	Compare Capture Register 2 preload enable bit. 1: Enable the preload function of compare capture register 2, read and write operations only operate on the preload registers, the preload value of compare capture register 2 is loaded into the current shadow register when the update event comes; 0: Disable the preload function of compare capture	0

			register 2, compare capture register 2 can be written at any time, and the newly written value takes effect immediately. *Note: Once the LOCK level is set to 3 and CC2S=00, this bit cannot be modified; PWM mode can be used only in single pulse mode (OPM=1) without confirming the pre-load register, otherwise its action is not determined.*	
10	OC2FE	RW	Compare Capture Channel 2 fast enable bit, this bit is used to speed up the response of the compare capture channel output to a trigger input event. 1: The active edge of the input to the flipflop acts as if a comparison match has occurred. Therefore, the OC is set to the comparison level independent of the comparison result. The delay between the valid edge of the sample trigger and the output of the compare capture channel 2 is reduced to 3 clock cycles. 0: Based on the value of the counter and compare capture register 1, compare capture channel 2 operates normally, even if the flip-flop is open. The minimum delay to activate the compare capture channel 2 output is 5 clock cycles when the input of the flipflop has a valid edge. OC2FE only works when the channel is configured to PWM1 or PWM2 mode.	0
[9:8]	CC2S[1:0]	RW	Compare capture channel 2 input selection fields. 00: comparison capture channel 2 is configured as an output. 01: comparison capture channel 2 is configured as an input and IC2 is mapped on TI2. 10: comparison capture channel 2 is configured as an input and IC2 is mapped on TI1. 11: Compare Capture Channel 2 is configured as an input and IC2 is mapped on TRC. This mode works only when the internal trigger input is selected (by the TS bit). *Note: Compare Capture Channel 2 is writable only when the channel is off (when CC2E is zero).*	0
7	OC1CE	RW	Compare capture channel 1 clear enable bit.	0
[6:4]	OC1M[2:0]	RW	Compare capture channel 1 mode setting field.	0
3	OC1PE	RW	Compare capture register 1 preload enable bit.	0
2	OC1FE	RW	Compare capture channel 1 fast enable bit.	0
[1:0]	CC1S[2:0]	RW	Compare capture channel 1 input selection fields.	0

Capture mode (pin direction is input).

Bit	Name	Access	Description	Reset value
[15:12]	IC2F[3:0]	RW	The input capture filter 2 configuration field, these bits set the sampling frequency of the TI1 input and the digital filter length. The digital filter consists of an event counter, which records N events and then generates a jump in the output. 0000: no filter, sampled at fDTS. 1000: sampling frequency Fsampling = Fdts/8, N = 6. 0001: sampling frequency Fsampling=Fck_int, N=2. 1001: sampling frequency Fsampling = Fdts/8, N = 8. 0010: sampling frequency Fsampling=Fck_int, N=4. 1010: sampling frequency Fsampling = Fdts/16, N = 5. 0011: sampling frequency Fsampling=f=Fck_int, N=8. 1011: sampling frequency Fsampling = Fdts/16, N = 6. 0100: sampling frequency Fsampling = Fdts/2, N = 6. 1100: sampling frequency Fsampling = Fdts/16, N = 8. 0101: sampling frequency Fsampling = Fdts/2, N = 8. 1101: sampling frequency Fsampling = Fdts/32, N = 5. 0110: sampling frequency Fsampling = Fdts/4, N = 6. 1110: sampling frequency Fsampling = Fdts/32, N = 6. 0111: sampling frequency Fsampling = Fdts/4, N = 8. 1111: Sampling frequency Fsampling=Fdts/32, N=8.	0
[11:10]	IC2PSC[1:0]	RW	Compare capture channel 2 prescaler configuration field, these 2 bits define the prescaler coefficient for compare capture channel 2. Once CC1E = 0, the prescaler is reset. 00: Without prescaler, one capture is triggered for each edge detected on the capture input. 01: Capture triggered every 2 events. 10: Capture triggered every 4 events. 11: Capture is triggered every 8 events.	0
[9:8]	CC2S[1:0]	RW	Compare the capture channel 2 input selection field, these 2 bits define the direction of the channel (input/output), and the selection of the input pin. 00: Compare capture channel 1 channel is configured as an output. 01: Compare capture channel 1 channel is configured as an input and IC1 is mapped on TI1. 10: Compare capture channel 1 channel is configured as an input and IC1 is mapped on TI2. 11: Compare capture channel 1 channel is configured as an input and IC1 is mapped on TRC. This mode works only when the internal trigger input is selected (by the TS bit).	0

			Note: CC1S is writable only when the channel is off (CC1E is 0).	
[7:4]	IC1F[3:0]	RW	Input capture filter 1 configuration field.	0
[3:2]	IC1PSC[1:0]	RW	Compare the capture channel 1 prescaler configuration field.	0
[1:0]	CC1S[1:0]	RW	Compare capture channel 1 input selection fields.	0

14.4.8 Compare/Capture Control Register 2 (TIM1_CHCTLR2)

Offset address: 0x1C

The channel can be used in input (capture mode) or output (compare mode), and the direction of the channel is defined by the corresponding CCxS bit. The other bits of this register serve different purposes in input and output modes. OCxx describes the function of the channel in output mode and ICxx describes the function of the channel in input mode.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M[2:0]			OC4PE	OC4FE	CC4S[1:0]		OC3CE	OC3M[2:0]			OC3PE	OC3FE	CC3S[1:0]	
IC4F[3:0]				IC4PSC[1:0]				IC3F[3:0]				IC3PSC[1:0]			

Compare mode (pin direction is output):

Bit	Name	Access	Description	Reset value
15	OC4CE	RW	Compare capture channel 4 clear enable bit.	0
[14:12]	OC4M[2:0]	RW	Compare capture channel 4 mode setting field.	0
11	OC4PE	RW	Compare capture register 4 preload enable bit.	0
10	OC4FE	RW	Compare capture channel 4 fast enable bit.	0
[9:8]	CC4S[1:0]	RW	Compare capture channel 4 input selection fields.	0
7	OC3CE	RW	Compare capture channel 3 clear enable bit.	0
[6:4]	OC3M[2:0]	RW	Compare capture channel 3 mode setting field.	0
3	OC3PE	RW	Compare capture register 3 preload enable bit.	0
2	OC3FE	RW	Compare capture channel 3 fast enable bit.	0
[1:0]	CC3S[1:0]	RW	Compare capture channel 3 input selection fields.	0

Capture mode (pin direction is input).

Bit	Name	Access	Description	Reset value
[15:12]	IC4F[3:0]	RW	Input capture filter 4 configuration field.	0
[11:10]	IC4PSC[1:0]	RW	Compare capture channel 4 prescaler configuration field.	0
[9:8]	CC4S[1:0]	RW	Compare capture channel 4 input selection fields.	0
[7:4]	IC3F[3:0]	RW	Input capture filter 3 configuration field.	0
[3:2]	IC3PSC[1:0]	RW	Compare capture channel 3 prescaler configuration fields.	0
[1:0]	CC3S[1:0]	RW	Compare capture channel 3 input selection fields.	0

14.4.9 Compare/Capture Enable Register (TIM1_CCER)

Offset address: 0x20

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E

Bit	Name	Access	Description	Reset value
[15:14]	Reserved	RO	Reserved	0
13	CC4P	RW	Compare the capture channel 4 output polarity setting bit.	0
12	CC4E	RW	Compare capture channel 4 output enable bit.	0
11	CC3NP	RW	Compare capture channel 3 complementary output polarity setting bit.	0
10	CC3NE	RW	Compare capture channel 3 complementary output enable bits.	0
9	CC3P	RW	Compare capture channel 3 output polarity setting bit.	0
8	CC3E	RW	Compare capture channel 3 output enable bit.	0
7	CC2NP	RW	Compare capture channel 2 complementary output polarity setting bit.	0
6	CC2NE	RW	Compare capture channel 2 complementary output enable bits.	0
5	CC2P	RW	Compare capture channel 2 output polarity setting bit.	0
4	CC2E	RW	Compare capture channel 2 output enable bit.	0
3	CC1NP	RW	Compare capture channel 1 complementary output polarity setting bit.	0
2	CC1NE	RW	Compare capture channel 1 complementary output enable bit.	0
1	CC1P	RW	Compare the output polarity setting bits of capture channel 1. The CC1 channel is configured to output: 1: OC1 low level effective. 0: OC1 high level is effective. The CC1 channel is configured to enter: The bit selects whether the inverse signal of IC1 or IC1 is used as the trigger or capture signal. 1: inversion: capture occurs at the falling edge of the IC1; when used as an external trigger, the IC1 is inverted. 0: no inversion: capture occurs on the rising edge of the IC1; when used as an external trigger, the IC1 is not inverted. *Note: Once the LOCK level (the LOCK bit in the TIMx_BDTR register) is set to 3 or 2, the bit cannot be modified.*	0
0	CC1E	RW	Compare the output enable bits of capture channel 1. The CC1 channel is configured to output: 1: On. The OC1 signal is output to the corresponding output pin, and its output level depends on the values of MOE, OSSI, OSSR, OIS1, OIS1N and CC1NE bits. 0: Off. OC1 forbids output, so the output level of OC1 depends on the values of MOE, OSSI, OSSR, OIS1, OIS1N, and CC1NE bits.	0

			The CC1 channel is configured to enter: This bit determines whether the value of the counter can be captured into the TIMx_CCR1 register. 1: Capture enable. 0: Capture is prohibited.	

14.4.10 Counter of Advanced-control Timer (TIM1_CNT)

Offset address: 0x24

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

Bit	Name	Access	Description	Reset value
[15:0]	CNT[15:0]	RW	The real-time value of the timer's counter.	0

14.4.11 Counting Clock Prescaler (TIM1_PSC)

Offset address: 0x28

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

Bit	Name	Access	Description	Reset value
[15:0]	PSC[15:0]	RW	The dividing factor of the prescaler of the timer; the clock frequency of the counter is equal to the input frequency of the divider/(PSC+1).	0

14.4.12 Auto-reload Value Register (TIM1_ATRLR)

Offset address: 0x2C

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

Bit	Name	Access	Description	Reset value
[15:0]	ARR[15:0]	RW	The value of this field will be loaded into the counter, see section 14.2.3 for when the ATRLR acts and updates; the counter stops when the ATRLR is empty.	0xFFFF

14.4.13 Repeat Count Value Register (TIM1_RPTCR)

Offset address: 0x30

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

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved	0
[7:0]	REP[7:0]	RW	The value of the repeat counter.	0

14.4.14 Compare/Capture Register 1 (TIM1_CH1CVR)

Offset address: 0x34

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15：0]															

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	LEVEL1	RO	The level indicator bit corresponding to the capture value	0
[15:0]	CCR1[15:0]	RW	Compare/capture register channel 1.	0

14.4.15 Compare/Capture Register 2 (TIM1_CH2CVR)

Offset address: 0x38

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL2
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15：0]															

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	LEVEL2	RO	The level indicator bit corresponding to the capture value	0
[15:0]	CCR1[15:0]	RW	Compare/capture register channel 2.	0

14.4.16 Compare/Capture Register 3 (TIM1_CH3CVR)

Offset address: 0x3C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL3
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15：0]															

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	LEVEL3	RO	The level indicator bit corresponding to the capture value	0
[15:0]	CCR1[15:0]	RW	Compare/capture register channel 3.	0

14.4.17 Compare/Capture Register 4 (TIM1_CH4CVR)

Offset address: 0x40

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL4
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15：0]															

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	LEVEL4	RO	The level indicator bit corresponding to the capture value	0
[15:0]	CCR1[15:0]	RW	Compare/capture register channel 4.	0

14.4.18 Brake and Deadband Register (TIM1_BDTR)

Offset address: 0x44

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							

Bit	Name	Access	Description	Reset value
15	MOE	RW	Main output enable bit. Once the brake signal is active, it will be cleared asynchronously. 1: Allow OCx and OCxN to be set as outputs. 0: Disable the output of OCx and OCxN or force to idle state.	0
14	AOE	RW	Auto output enable. 1: MOE can be set by software or set in the next update event. 0: MOE can only be set by software.	0
13	BKP	RW	The brake input polarity setting bit. 1: Brake input active high. 0: Brake input is active low. *Note: When LOCK level 1 is set, this bit cannot be modified. A write to this bit requires an PB clock before it can take effect.*	0
12	BKE	RW	Brake function enable bit. 1: Turn on the brake input. 0: Brake input is disabled. *Note: When LOCK level 1 is set, this bit cannot be modified. A write to this bit requires an PB clock before it can take effect.*	0
11	OSSR	RW	1: When the timer is not working, once CCxE=1 or CCxNE=1, first turn on OC/OCN and outputinvalid	0

			level, then set OCx, OCxN enable output signal=1. 0: When the timer is not operating, OC/OCN output is disabled. *Note: When LOCK level 1 is set, this bit cannot be modified.*	
10	OSSI	RW	1: when the timer is not operating, once CCxE = 1 or CCxNE = 1, OC/OCN first outputs its idle level, then OCx, OCxN enable output signal = 1. 0: When the timer is not operating, OC/OCN output is disabled. *Note: When LOCK level 1 is set, this bit cannot be modified.*	0
[9:8]	LOCK[1:0]	RW	Lock the function setting field. 00: Disable the locking function. 01: Lock level 1, no DTG, BKE, BKP, AOE, OISx and OISxN bits can be written. 10: Lock level 2, where the bits in lock level 1 cannot be written, nor the CC polarity bits, nor the OSSR and OSSI bits. 11: Lock level 3, cannot write to the bits in lock level 2, and cannot write to the CC control bits. *Note: After system reset, the LOCK bit can only be written once and cannot be modified again until reset.*	0
[7:0]	DTG[7:0]	RW	Deadband setting bits that define the duration of the deadband between complementary outputs. Assume that DT denotes its duration. DTG[7:5]=0xx=>DT=DTG[7:0]*Tdtg, Tdtg=TDTS； DTG[7:5]=10x=>DT=(64+DTG[5:0])*Tdtg, Tdtg=2*TDTS; DTG[7:5]=110=>DT=(32+DTG[4:0])*Tdtg, Tdtg =8×TDTS; DTG[7:5]=111=>DT=(32+DTG[4:0])*Tdtg, Tdtg =16*TDTS. *Note: Once the LOCK level (the LOCK[1:0] bit in the TIM1_BDTR register) is set to 1, 2, or 3, these bits cannot be modified.*	0

14.4.19 DMA Control Register (TIM1_DMACFGR)

Offset address: 0x48

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

Bit	Name	Access	Description	Reset value
[15:13]	Reserved	RO	Reserved	0

[12:8]	DBL[4:0]	RW	The length of the DMA continuous transmission, the actual value of which is the value of this field + 1.	0
[7:5]	Reserved	RO	Reserved	0
[4:0]	DBA[4:0]	RW	These bits define the offset of the DMA in continuous mode from the address where control register 1 is located.	0

14.4.20 DMA Address Register in Continuous Mode (TIM1_DMAADR)

Offset address: 0x4C

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

Bit	Name	Access	Description	Reset value
[15:0]	DMAB [31:0]	RW	The address of the DMA in continuous mode.	0

Chapter 15 General-Purpose Timer (GPTM)

The general timer module consists of two 16-bit automatic reassembling timers (TIM2, TIM3) and a 32-bit automatic reinstalling timer (TIM4), which are used to measure pulse width or generate pulses and PWM waves with specific frequencies. It can be used in automatic control, power supply and other fields.

15.1 Main Features

The main features of a 16-bit general-purpose timer include:

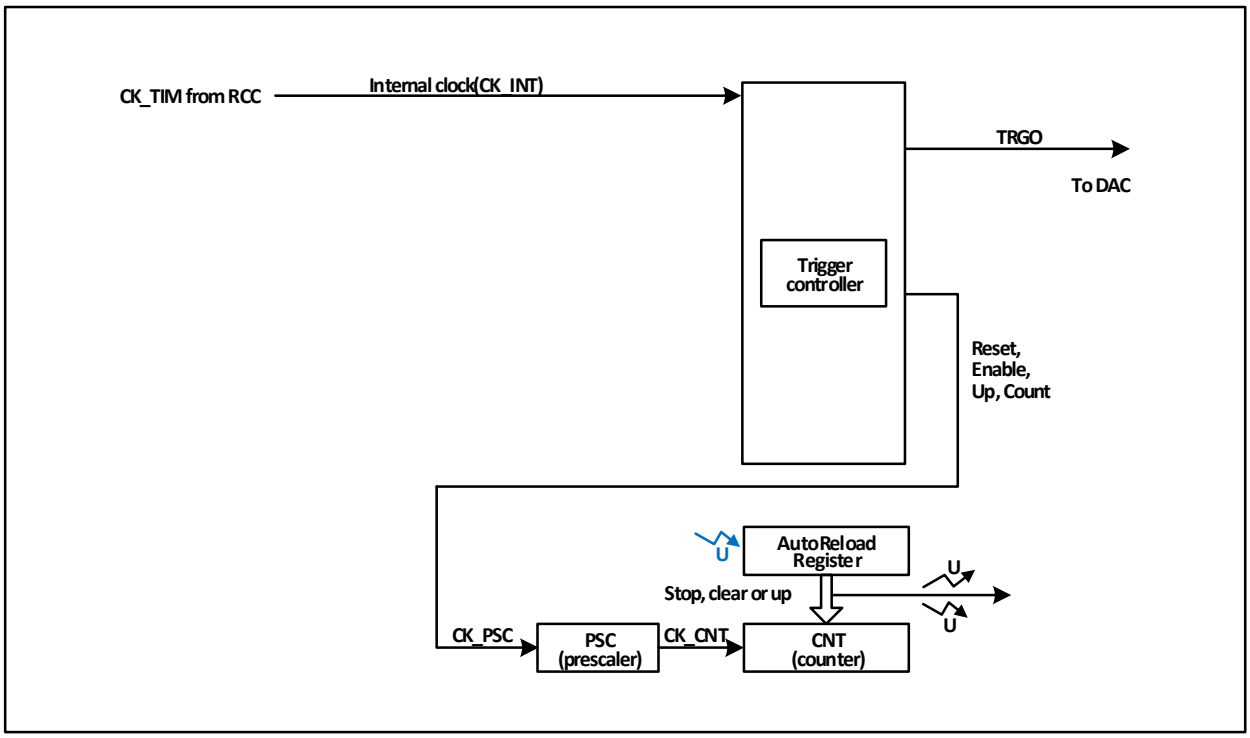
- 16-bit automatic reinstall counter, supporting increasing counting mode, decreasing counting mode and increasing and decreasing counting mode.
- 16-bit prescaler, the frequency division coefficient is dynamically adjustable from 1 to 65536.
- Support four independent comparison capture channels.
- Each comparison capture channel supports multiple operating modes, such as input capture, output comparison, PWM generation and single pulse output.
- Support external signal control timer.
- Support using DMA in multiple modes.
- Support incremental coding, cascading and synchronization between timers.

The main features of a 32-bit general-purpose timer include:

- 32-bit automatic reinstall counter, supporting increasing counting mode, decreasing counting mode and increasing and decreasing counting mode.
- 16-bit prescaler, the frequency division coefficient is dynamically adjustable from 1 to 65536.
- Support four independent comparison capture channels.
- Each comparison capture channel supports multiple operating modes, such as input capture, output comparison, PWM generation and single pulse output.
- Support external signal control timer.
- Support using DMA in multiple modes.
- Support incremental coding, cascading and synchronization between timers

15.2 Principle and Structure

Figure 15-1 Block diagram of the structure of the general-purpose timer



15.2.1 Overview

As shown in Figure 15-1, the structure of the general-purpose timer can be roughly divided into three parts, namely the input clock part, the core counter part and the compare capture channel part.

The clock for the general-purpose timer can come from the HB bus clock (CK_INT), from the external clock input pin (TIMx_ETR), from other timers with clock output (ITRx), and from the input of the compare capture channel (TIMx_CHx). These input clock signals become CK_PSC clocks after various set filtering and dividing operations, etc., and are output to the core counter section. In addition, these complex clock sources can also be output as TRGO to other peripherals such as timers and ADCs.

The core of the general-purpose timer is a 16-bit counter (CNT). CK_PSC is divided by a prescaler (PSC) to become CK_CNT and then finally fed to the CNT, which supports incremental counting mode, decremental counting mode, and incremental and decremental counting mode, and has an auto-reload register (ATRLR) to reload the initialization value for the CNT at the end of each counting cycle.

The general-purpose timer has 4 sets of compare capture channels, each of which can input pulses from exclusive pins or output waveforms to pins, i.e., the compare capture channels support both input and output modes. The input of each channel of the compare capture register supports filtering, dividing, edge detection, and other operations, and supports mutual triggering between channels, and can also provide clock for the core counter CNT. Each compare capture channel has a set of compare capture registers (CHxCVR) that support comparison with the main counter (CNT) to output pulses.

15.2.2 Difference between General-purpose Timer and Advanced-control Timer

Compared with the advanced-control timer, the general-purpose timer is lack of the following functions:

1) The general-purpose timer lacks a repeated counting register that counts the count cycle of core counter.

2) The compare/capture of general-purpose timer lacks deadband generation and has no complementary output.

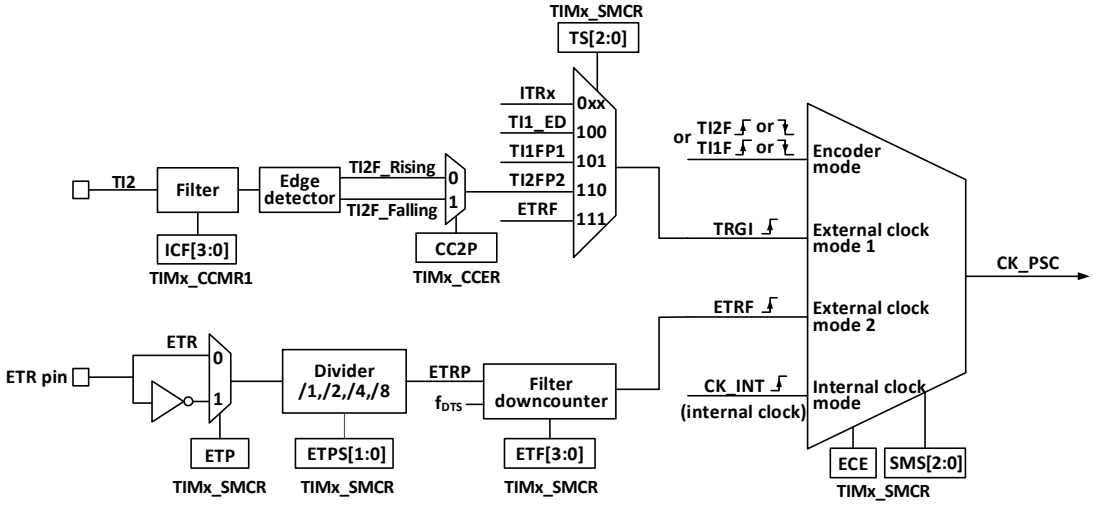
3) The general-purpose timer has no break signal mechanism.

4) The default clock CK_INT of the general-purpose timer comes from PB1, while the CK_INT of the advanced-control timer comes from PB2.

15.2.3 Clock Input

This section describes the source of CK_PSC. The clock source part of the general structure block diagram of the general-purpose timer is abstracted here.

Figure 15-2 Block diagram of general-purpose timer source



The available input clocks can be divided into 4 categories:

1) External clock pin (ETR) input: ETR→ETRP→ETRF;

2) Internal PB clock input: CK_INT;

3) From the compare/capture pin (TIMx_CHx): TIMx_CHx→TIx→TIxFPx; it is also used in encoder mode;

4) Input from other internal timers: ITRx.

The actual operation can be divided into 3 categories by determining the input pulse selection of the SMS from the CK_PSC source:

1) Select the internal clock source (CK_INT);

2) External clock source mode 1;

3) External clock source mode 2;

4) Encoder code.

The 4 clock sources mentioned above can be selected by these 4 operations.

15.2.3.1 Internal Clock Source (CK_INT)

If the general-purpose timer is started when the SMS domain is kept at 000b, then the internal clock source (CK_INT) is selected as the clock. At this moment, CK_INT is CK_PSC.

15.2.3.2 External Clock Source Mode 1

If SMS is set to 111b, the external clock source mode1 is enabled. When external clock source mode1 is enabled, TRGI is selected as the source of CK_PSC. It is worth noting that the user needs to configure TS to select the source of TRGI. For TS, the following pulses can be used as the clock source:

1) Internal Trigger (ITRx, x is 0, 1, 2, 3);
2) Signal of compare/capture1 after passing through the edge detector (TI1F_ED);
3) Signals TI1FP1 and TI2FP2 of compare/capture;
4) Signal ETRF from external clock pin.

15.2.3.3 External Clock Source Mode 2

Use external trigger mode2 to count on every rising or falling edge of the external clock pin input. When the ECE bit is set, the external clock source mode2 is enabled. When the external clock source mode2 is enabled, ETRF is selected as CK_PSC. The ETR pin passes through the optional inverter (ETP) and frequency divider (ETPS) to become ETRP, and then passes through the filter (ETF) to become ETRF.

When the ECE bit is set and the SMS is set to 111b, it means that the TS selects ETRF as the input.

15.2.3.4 Encoder Mode

Set SMS to 001b, 010b and 011b to enable the encoder mode. After enable the encoder mode, you may choose to use another transition edge as a signal for signal output at a certain level in TI1FP1 and TI2FP2. This mode is used when the external encoder is used. Refer to Section15.3.7 for the specific functions.

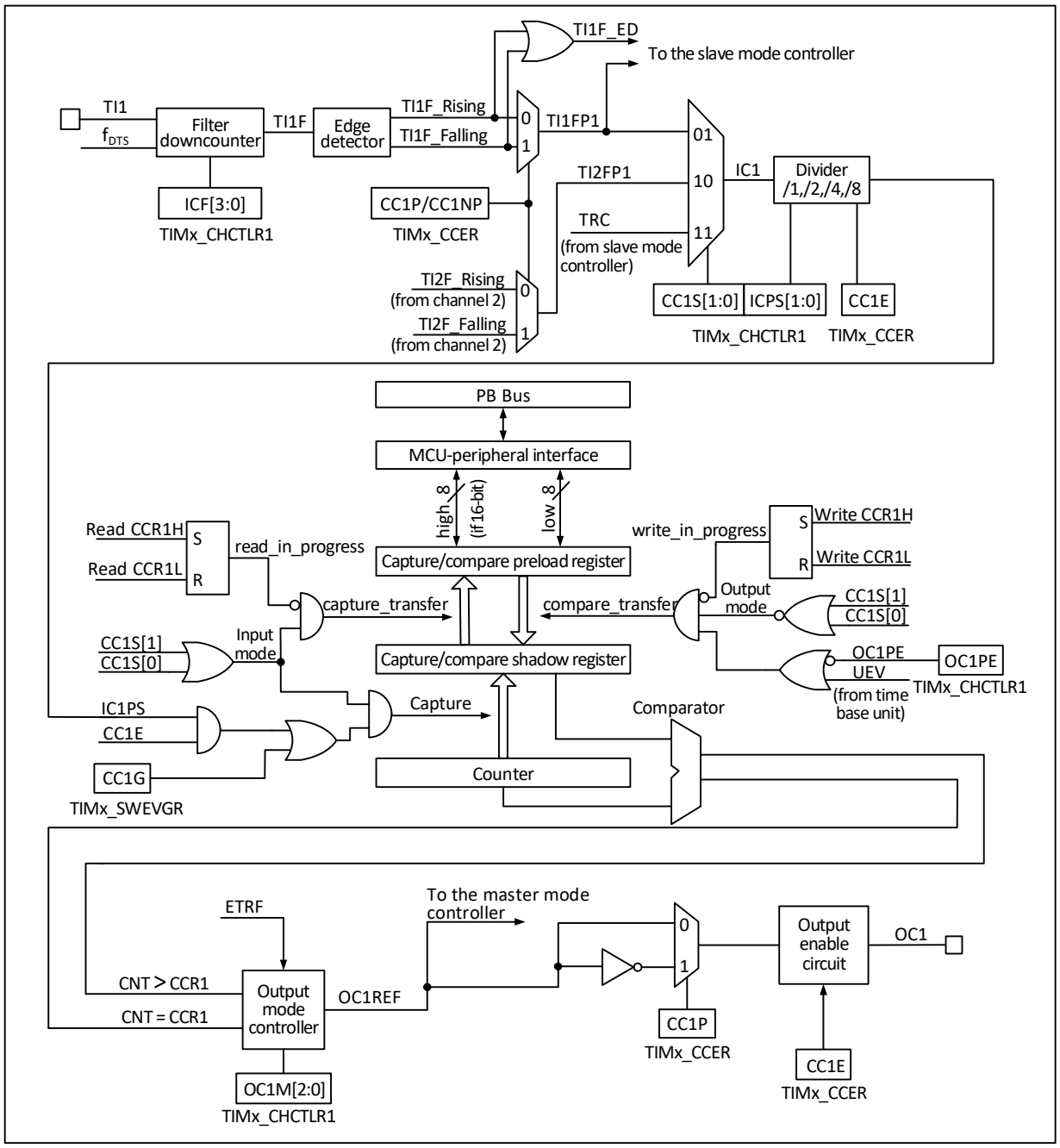
15.2.4 Counter and Periphery

CK_PSC inputs to the prescaler (PSC) for frequency division. PSC has 16 bits, and the actual frequency division factor is equivalent to the value of R16_TIMx_PSC+1. CK_PSC becomes CK_INT through PSC. The changed value of R16_TIM1_PSC does not take effect in real time, but can be updated to the PSC after the update event. Update events include clearing and resetting the UG bit.

15.2.5 Compare/capture Channel

The compare/capture is the core of the timer to achieve complex functions. Its core is the compare/capture register, supplemented by the digital filtering of the peripheral input part, frequency division and channel multiplexing, the output part comparator and output control. The block diagram of the compare/capture is as shown in Figure 15-3.

Figure 15-3 Structure block diagram of compare/capture channel



After the signal is input from the channel x pin, it can be selected as TIx (the source of TI1 may be more than CH1. See Figure 15-1 timer block diagram). TI1 passes through the filter (ICF[3:0]) to generate TI1F, and then is divided into TI1F_Rising and TI1F_Falling after passing through the edge detector. These 2 signals are selected (CC1P) to generate TI1FP1, and TI1FP1 and TI2FP1 from channel 2 are sent to CC1S together to be selected as IC1, and then sent to the compare/capture register after going through the ICPS frequency division.

The compare/capture register is composed of preload register and shadow register, and only the preload register is operated during reading and writing. In the capture mode, the capture occurs on the shadow register, and then copied to the preload register; in the comparison mode, the content of the preload register is copied to the shadow register, and then the content of the shadow register is compared with the core counter (CNT).

15.3 Function and Implementation

The general-purpose timer complex functions are implemented by the operation of compare/capture channel, clock input circuit, counter and peripheral parts of the timer. The timer's clock input can come from multiple clock sources including the input of the compare/capture. The operation of compare/capture register channel and the clock source selection directly determines its function. The compare/capture is bidirectional and can work in input and output modes.

15.3.1 Input Capture Mode

Input capture mode is one of the basic functions of timer. The principle of the input capture mode is that when a determined edge on the ICxPS signal is detected, a capture event is generated and the current value of the counter is locked in the comparison capture register (R16_TIMx_CHCTLRx). When a capture event occurs, the CCxIF (in R16_TIMx_INTFR) is set, and if the interrupt or DMA is enabled, the corresponding interrupt or DMA is generated. If the CCxIF is already set when the capture event occurs, the CCxOF bit will be set. CCxIF can be cleared by software or by hardware by reading comparison capture registers. The CCxOF is cleared by the software.

Give an example of channel 1 to illustrate the steps to use input capture mode, as follows:
1) Configure the ICx domain and select the source of the CCXS signal. For example, if you set it to 10b and select TI1FP1 as the source of the IC1, you cannot use the default setting. The CCXS domain defaults to using the comparison capture module as the output channel.
2) Configure the ICxF domain and set the digital filter of the TI signal. The digital filter will output a jump at a certain frequency and a certain number of samples. The sampling frequency and number are determined by ICxF.
3) Configure the CCxP bit and set the polarity of TIxFPx. For example, keep the CC1P bit low and choose the rising edge jump.
4) Configure the ICxPS domain and set the ICx signal to become the frequency division coefficient between ICxPS. For example, keep the ICxPS at 00b without frequency division.
5) Configure the CCxE bit to allow the value of the core counter (CNT) to be captured into the comparison capture register. Set CC1E bit.
6) Configure the CCxIE and CCxDE bits as needed to decide whether to allow enable interrupts or DMA.

At this point, the comparison capture channel configuration has been completed.

When the TI1 inputs a captured pulse, the value of the core counter (CNT) is recorded in the comparison capture register, the CC1IF is set, and the CCIOFbit is set when the CC1IF has been previously set. If the CC1IE bit, an interrupt is generated; if the CC1DE is set, an DMA request is generated. An input capture event can be generated by the software by writing events to generate registers (R16_TIMx_SWEVGR).

15.3.2 Compare Output Mode

The compare output mode is one of basic functions of timer. The principle of the compare output mode is to output a specific change or waveform when the value of the core counter (CNT) is consistent with the value of the compare/capture register. OCxM (in R16_TIMx_CHCTLRx) and the CCxP bit (in R16_TIMx_CCER) determine whether the output is determined high or low level or level inversion. When a compare consistent event is generated, the CCxIF bit will be also set. If the CCxIE bit is preset, an interrupt will be generated; if the CCxDE bit is preset, a DMA request will be generated.

The procedure of compare output mode configuration is as follows:
1) Configure the clock source and auto-reload value of the core counter (CNT);

2) Set the count value to be compared to the compare/capture register (R16_TIMx_CHxCVR);
3) If an interrupt needs to be generated, set the CCxIE bit;
4) Keep OCxPE as 0 and disable the preload register of the compare/capture register;
5) Set the output mode, and set OCxM and CCxP bit;
6) Enable the output and set the CCxE bit;
7) Set the CEN bit and start the timer;

15.3.3 Forced Output Mode

The output mode of the compare/capture of the timer can be forced to output a certain level by software, instead of relying on the shadow register and the core counter of the compare/capture register.

The specific means is to set OCxM to 100b, which means to force OCxREF to be low; or to set OCxM to 101b, which means setting OCxREF to a high value by force.

It shall be noted that if OCxM is set to 100b or 101b by force, the compare process between the internal main counter and the compare/capture register will be still in progress, the corresponding flag bit will be still set, and interrupts and DMA request will still be generated.

15.3.4 PWM Input Mode

The PWM input mode is used to measure the duty cycle and frequency of the PWM, which is a special case of the input capture mode. The operation is the same as the input capture mode except for the following differences: PWM occupies 2 compare/captures, and the input polarity of the 2 channels is set to opposite. One of the signals is set to trigger input, and SMS is set to reset mode.

For example, to measure the cycle and frequency of the PWM wave input from TI1, the following operations are required:
1) Set TI1 (TI1FP1) as the input of IC1 signal. Set CC1S to 01b;
2) Set TI1FP1 as the rising edge valid. Keep CC1P as 0;
3) Set TI1 (TI1FP2) as the input of IC2 signal. Set CC2S to 10b;
2) Set TI1FP2 as the falling edge valid. Set CC2P to 1;
5) The source of the clock source is TI1FP1. Set TS to 101b;
6) Set SMS to reset mode, i.e., 100b;
7) Enable the input capture. Set CC1E and CC2E bits.

Note: Since only TI1FP1 and TI2FP2 are connected to the slave mode controller, only TIMx_CH1/TIMx_CH2 can be used for PWM input mode.

15.3.5 PWM Output Mode

The PWM output mode is one of basic functions of timer. The most common method of PWM output mode is to use the reload value to determine the PWM frequency, and to use the capture comparison register to determine the duty cycle. Set 110b or 111b in OCxM to use PWM mode 1 or mode 2, set the OCxPE bit to enable the preload register, and finally set the ARPE bit. Since the value of the preload register can be sent to the shadow register when an update event occurs, it is necessary to set the UG bit to initialize all registers before the core counter starts counting. In the PWM mode, the core counter and the compare/capture register are always being compared. According to the CMS bit, the timer can output edge-aligned or center-aligned PWM signals.

- Edge alignment

When the edge alignment is used, the core counter counts up or down. In the scenario of PWM mode 1, when the value of the core counter is greater than that of the compare/capture register, OCxREF will rise to be high; when the value of the core counter is less than the compare/capture register (such as When the core counter increases to the value of R16_TIMx_ATRLR and returns to all 0s), OCxREF drops to low.

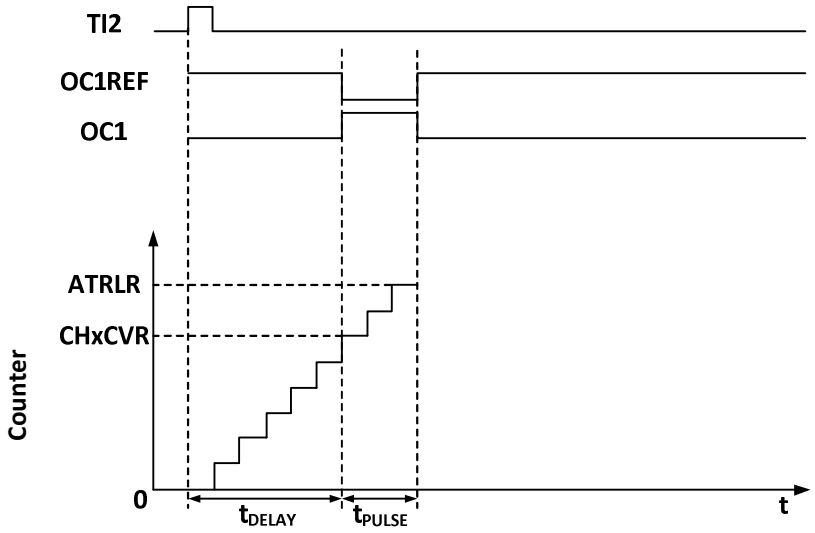
- Central alignment

When the center-aligned mode is used, the core counter will run in a mode where up counting and down counting are performed alternately, and OCxREF performs rising and falling jumps when the values of the core counter and the compare/capture register are consistent. However, in 3 types of central alignment mode, the bit setting timing of comparison flag is different somewhat. When the center-alignment mode is used, it is the best to generate a software update flag (set the UG bit) before starting the core counter.

15.3.6 Single Pulse Mode

The single pulse mode can be used to respond to a specific event to generate a pulse after a delay. The delay and pulse width are programmable. Setting the OPM bit can make the core counter stop when the next update event UEV is generated (the counter turns over to 0).

Figure 15-4 Event generation and pulse response



As shown in Figure 15-4, it is necessary to detect the beginning of a rising edge on the TI2 input pin. After delaying Tdelay, a positive pulse of length Tpulse will be generated on OC1:

1) Set TI2 as trigger. Set the CC2S field to 01b and map TI2FP2 to TI2; set the CC2P bit to 0b and set TI2FP2 to rising edge detection; set the TS field to 110b and set TI2FP2 as the trigger source; set the SMS field to 110b, and TI2FP2 is used to start the counter;

2) Tdelay is defined by the value of the compare/capture register, and Tpulse is determined by the value of the auto-reload value register and the value of the compare/capture register.

15.3.7 Encoder Mode

The encoder mode is a typical application of the timer. It can be used to access the dual-phase output of the encoder. The count direction of the core counter is synchronized with the rotating shaft of the encoder. Each pulse output by the encoder will increase the core counter by adding one or subtracting one. The steps to use the encoder are: set the SMS field to 001b (count only on TI2 edge), 010b (count only on TI1 edge) or 011b (count on both TI1 and TI2

edges), and connect the encoder to compare/capture 1, 2 inputs, set a value for the reload value register and this value can be set to be greater. In the encoder mode, the internal compare/capture register of timer, prescaler, repeat count register and other registers all work normally. The following table shows the relationship between the counting direction and the encoder signal.

Table 15-1 Relationship between count direction of timer in encoder mode and encoder signal

Count active edge	Relative signal level	TI1FP1 signal edge		TI2FP2 signal edge	
		Rising edge	Falling edge	Rising edge	Falling edge
Only count at TI1 edge	High	Downcount	Upcount	Not count	
	Low	Upcount	Downcount		
Only count at TI2 edge	High	Not count		Upcount	Downcount
	Low			Downcount	Upcount
Count on both edges of TI1 and TI2	High	Downcount	Upcount	Upcount	Downcount
	Low	Upcount	Downcount	Downcount	Upcount

15.3.8 Timer Synchronous Mode

The timer can output clock pulses (TRGO) and can also receive input from other timers (ITRx). The sources of ITRx of different timers (TRGO of other timers) are different. Table 15-2 shows the internal trigger connection of timers.

Figure 15-2 GTPM internal trigger connection

Slave timer	ITR0(TS=000)	ITR1(TS=001)	ITR2(TS=010)	ITR3(TS=011)
TIM2	TIM1	USB	TIM3	TIM4
TIM3	TIM1	TIM2	-	TIM4
TIM4	TIM1	TIM2	TIM3	-

15.3.9 Debug Mode

When the system enters debug mode, the timer continues to run or stops according to the setting of the DBG module.

15.4 Register Description

Table 15-3 TIM2 registers

Name	Offset address	Description	Reset value
R16_TIM2_CTLR1	0x40000000	TIM2 control register1	0x0000
R16_TIM2_CTLR2	0x40000004	TIM2 control register2	0x0000
R16_TIM2_SMCFGR	0x40000008	TIM2 slave mode configuration register	0x0000
R16_TIM2_DMAINTENR	0x4000000C	TIM2 DMA/interrupt enable register	0x0000
R16_TIM2_INTFR	0x40000010	TIM2 interrupt flag register	0x0000
R16_TIM2_SWEVGR	0x40000014	TIM2 event generation register	0x0000
R16_TIM2_CHCTLR1	0x40000018	TIM2 compare/capture control register 1	0x0000

R16_TIM2_CHCTLR2	0x4000001C	TIM2 compare/capture control register 2	0x0000
R16_TIM2_CCER	0x40000020	TIM2 compare/capture enable register	0x0000
R16_TIM2_CNT	0x40000024	TIM2 counter	0x0000
R16_TIM2_PSC	0x40000028	TIM2 prescaler	0x0000
R16_TIM2_ATRLR	0x4000002C	TIM2 auto-reload register	0xFFFF
R32_TIM2_CH1CVR	0x40000034	TIM2 compare/capture register 1	0x00000000
R32_TIM2_CH2CVR	0x40000038	TIM2 compare/capture register 2	0x00000000
R32_TIM2_CH3CVR	0x4000003C	TIM2 compare/capture register 3	0x00000000
R32_TIM2_CH4CVR	0x40000040	TIM2 compare/capture register 4	0x00000000
R16_TIM2_DMACFGR	0x40000048	TIM2 DMA configuration register	0x0000
R16_TIM2_DMAADR	0x4000004C	TIM2 DMA address register in continuous mode	0x00000000

Table 15-4 TIM3 registers

Name	Offset address	Description	Reset value
R16_TIM3_CTLR1	0x40000400	TIM3 control register 1	0x0000
R16_TIM3_CTLR2	0x40000404	TIM3 control register 2	0x0000
R16_TIM3_SMCFGR	0x40000408	TIM3 slave mode configuration register	0x0000
R16_TIM3_DMAINTENR	0x4000040C	TIM3 DMA/interrupt enable register	0x0000
R16_TIM3_INTFR	0x40000410	TIM3 interrupt flag register	0x0000
R16_TIM3_SWEVGR	0x40000414	TIM3 event generation register	0x0000
R16_TIM3_CHCTLR1	0x40000418	TIM3 compare/capture control register 1	0x0000
R16_TIM3_CHCTLR2	0x4000041C	TIM3 compare/capture control register 2	0x0000
R16_TIM3_CCER	0x40000420	TIM3 compare/capture enable register	0x0000
R16_TIM3_CNT	0x40000424	TIM3 counter	0x0000
R16_TIM3_PSC	0x40000428	TIM3 prescaler	0x0000
R16_TIM3_ATRLR	0x4000042C	TIM3 auto reload register	0xFFFF
R32_TIM3_CH1CVR	0x40000434	TIM3 compare/capture register 1	0x000000000
R32_TIM3_CH2CVR	0x40000438	TIM3 compare/capture register 2	0x00000000
R32_TIM3_CH3CVR	0x4000043C	TIM3 compare/capture register 3	0x00000000
R32_TIM3_CH4CVR	0x40000440	TIM3 compare/capture register 4	0x00000000
R16_TIM3_DMACFGR	0x40000448	TIM3 DMA configuration register	0x0000
R16_TIM3_DMAADR	0x4000044C	TIM3 DMA address register in continuous mode	0x00000000

Table 15-5 TIM4 registers

Name	Offset address	Description	Reset value
R16_TIM4_CTLR1	0x40000800	TIM4 control register1	0x0000
R16_TIM4_CTLR2	0x40000804	TIM4 control register2	0x0000
R16_TIM4_SMCFGR	0x40000808	TIM4 slave mode configuration register	0x0000
R16_TIM4_DMAINTENR	0x4000080C	TIM4 DMA/interrupt enable register	0x0000
R16_TIM4_INTFR	0x40000810	TIM4 interrupt flag register	0x0000

R16_TIM4_SWEVGR	0x40000814	TIM4 event generation register	0x0000
R16_TIM4_CHCTLR1	0x40000818	TIM4 compare/capture control register1	0x0000
R16_TIM4_CHCTLR2	0x4000081C	TIM4 compare/capture control register2	0x0000
R16_TIM4_CCER	0x40000820	TIM4 compare/capture enable register	0x0000
R16_TIM4_CNT	0x40000824	TIM4 counter	0x00000000
R16_TIM4_PSC	0x40000828	TIM4 prescaler	0x0000
R16_TIM4_ATRLR	0x4000082C	TIM4 auto-reload register	0xFFFFFFFF
R16_TIM4_CH1CVR	0x40000834	TIM4 compare/capture register1	0x00000000
R16_TIM4_CH2CVR	0x40000838	TIM4 compare/capture register2	0x00000000
R16_TIM4_CH3CVR	0x4000083C	TIM4 compare/capture register3	0x00000000
R16_TIM4_CH4CVR	0x40000840	TIM4 compare/capture register4	0x00000000
R16_TIM4_DMACFGR	0x40000848	TIM4 DMA configuration register	0x0000
R16_TIM4_DMAADR	0x4000084C	TIM4 DMA address register in continuous mode	0x00000000

15.4.1 Control Register 1 (TIMx_CTLR1) (x=2/3/4)

Offset address: 0x00

15	14	13 12 11 10	9 8	7	6 5	4	3	2	1	0
TMR_CAP_LVL_EN	TMR_CAP_OV_EN	Reserved	CKD[1:0]	ARPE	CMS[1:0]	DIR	OPM	URS	UDIS	CEN

Bit	Name	Access	Description	Reset value
15	TMR_CAP_LVL_EN	RW	In double edge capture mode, capture level indication enables: 0: Disable indication function. 1: Enable indicates function. *Note: After enabling, [17] of CHxCVR indicates the level corresponding to the capture value.*	0
14	TMR_CAP_OV_EN	RW	Capture value mode configuration: 0: Capture value is the value of the actual counter. 1: When a counter overflow occurs before capture, the CHxCVR value is 0xFFFF.	0
[13:10]	Reserved	RO	Reserved	0
[9:8]	CKD[1:0]	RW	These 2 bits define the frequency division ratio of timer clock (CK_INT) frequency and sampling clock used for the digital filter: 00: Tdts=Tck_int; 01: Tdts= 2xTck_int; 10: Tdts= 4xTck_int; 11: Reserved.	00b
7	ARPE	RW	Auto-reload and preload enable: 1: Auto-reload value register (ATRLR) enabled; 0: Auto-reload value register (ATRLR) disabled.	0
[6:5]	CMS[1:0]	RW	Central alignment mode selection:	00b

			00: Edge alignment mode. The counter counts up or down according to the direction bit (DIR). 01: Center alignment mode 1. The counter counts up and down alternately. The output comparison interrupt flag bit of the channel configured as an output (CCxS=00 in the CHCTLRx register) is only set when the counter counts down. 10: Center alignment mode 2. The counter counts up and down alternately. The output comparison interrupt flag bit of the channel configured as an output (CCxS=00 in the CHCTLRx register) is only set when the counter counts up. 11: Center alignment mode 3. The counter counts up and down alternately. The output comparison interrupt flag bit of the channel configured as an output (CCxS=00 in the CHCTLRx register) is only set when the counter counts up and down. *Note: When the counter is enabled (CEN=1), it is not allowed to switch from edge alignment mode to center alignment mode.*	
4	DIR	RW	Counter direction: 0: Upcount; 1: Downcount. *Note: When the counter is configured in the center alignment mode or encoder mode, this bit will be invalid.*	0
3	OPM	RW	Single pulse mode. 1: The counter stops when the next update event (clearing the CEN bit) occurs; 0: The counter does not stop when the next update event occurs.	0
2	URS	RW	Updates are prohibited, and the software allows / disables the generation of UEV events through this bit. 1: UEV is prohibited. No update events are generated, and the registers (ATRLR, PSC, CHCTLRx) hold their values. If the UG bit is set or a hardware reset is issued from the mode controller, the counter and prescaler are reinitialized. 0: UEV is allowed. The UEV event is generated by any of the following events: -Counter overflow / underflow. -Set the UG bit. -Updates generated from the mode controller. Registers with caches are loaded into their preloaded	0

			values.	
1	UDIS	RW	Updates are prohibited, and the software allows / disables the generation of UEV events through this bit. 1: UEV is prohibited. No update events are generated, and the registers (ATRLR, PSC, CHCTLRx) hold their values. If the UG bit is set or a hardware reset is issued from the mode controller, the counter and prescaler are reinitialized. 0: UEV is allowed. The UEV event is generated by any of the following events: -Counter overflow / underflow. -Set the UG bit. -Updates generated from the mode controller. Registers with caches are loaded into their preloaded values.	0
0	CEN	RW	Enable counter (Counter enable). 1: Enable counter. 0: Disables counters. *Note: the external clock, gating mode and encoder mode will not work until the CEN bit is set in the software. The trigger mode automatically sets the CEN bit through the hardware.*	0

15.4.2 Control Register 2 (TIMx_CTLR2) (x=2/3/4)

Offset address: 0x04

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TI1S	MMS[2:0]			CCDS	Reserved		

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved	0
7	TI1S	RW	TI1 selection: 1: TIMx_CH1, TIMx_CH2 and TIMx_CH3 pins are connected to TI1 input through XOR; 0: TIMx_CH1 pin is directly connected to TI1 input.	0
[6:4]	MMS[2:0]	RW	Master mode selection: These 3 bits are used to select the synchronization information (TRGO) sent to the slave timer in the master mode. The possible combination is as follows: 000: Reset – The UG bit is used as a trigger output (TRGO). If it is a reset generated by a trigger input (the slave mode controller is in reset mode), the signal on TRGO will have a delay relative to the actual reset;	000b

			001: Enable-the counter enables signal CNT_EN to be used as a trigger output (TRGO). Sometimes, it is necessary to start multiple timers at the same time or control to enable slave timers within a period of time. The counter enable signal is generated by the logical OR of the CEN control bit and the trigger input signal in the gating mode. When the counter enable signal is controlled by the trigger input, there will be a delay on TRGO, unless the master/slave mode is selected (see the description of the MSM bit in the TIMx_SMCFGR register); 010: An update event is selected as the trigger input (TRGO). For example, the clock of a master timer can be used as a prescaler for a slave timer; 011: Comparison pulse, when a capture occurs or a comparison is successful, and the CC1IF flag is to be set (even if it is already high), the trigger output will send a positive pulse (TRGO); 100: OC1REF signal is used as trigger output (TRGO); 101: OC2REF signal is used as trigger output (TRGO); 110: OC3REF signal is used as trigger output (TRGO); 111: OC4REF signal is used as trigger output (TRGO).	
3	CCDS	RW	1: When an update event occurs, send a DMA request of CHxCVR; 0: When CHxCVR occurs, a DMA request of CHxCVR will be generated.	0
[2:0]	Reserved	RO	Reserved.	0

15.4.3 Slave Mode Control Register (TIMx_SMCFGR) (x=2/3/4)

Offset address: 0x08

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			Reserved	SMS[2:0]		

Bit	Name	Access	Description	Reset value
15	ETP	RO	ETR trigger polarity selection; this bit selects whether to directly input ETR or input inverted ETR. 1: ETR inverted, active at low level or falling edge; 0: ETR, active at high level or rising edge.	0
14	ECE	RW	External clock mode 2 enable. 1: External clock mode 2 enabled; 0: External clock mode 2 disabled. *Note 1: Slave mode can be used simultaneously with external clock mode 2: reset mode, gating mode and*	0

			trigger mode; however, TRGI cannot be connected to ETRF at this time (TS bit cannot be 111b). *Note 2: When both external clock mode 1 and external clock mode 2 are enabled at the same time, the input of the external clock will be ETRF.*	
[13:12]	ETPS[1:0]	RW	External trigger prescaler (ETRP); the frequency must be at most 1/4 of TIMxCLK frequency, and the frequency can be reduced through this domain. 00: Prescale OFF; 01: ETRP frequency divided by 2; 10: ETRP frequency divided by 4; 11: ETRP frequency divided by 8.	00b
[11:8]	ETF[3:0]	RW	External trigger filter. In fact, the digital filter is an event counter. N events are needed to validate a transition on the output. 0000: No filter, sampling is done at Fdts; 0001: Fsampling=Fck_int, N=2; 0010: Fsampling=Fck_int，N=4; 0011: Fsampling=Fck_int, N=8; 0100: Fsampling=Fdts/2, N=6; 0101: Fsampling=Fdts/2, N=8; 0110: Fsampling=Fdts/4, N=6; 0111: Fsampling=Fdts/4, N=8; 1000: Fsampling=Fdts/8, N=6; 1001: Fsampling=Fdts/8, N=8; 1010: Fsampling=Fdts/16, N=5; 1011: Fsampling=Fdts/16, N=6; 1100: Fsampling=Fdts/16, N=8; 1101: Fsampling=Fdts/32, N=5; 1110: Fsampling=Fdts/32, N=6; 1111: Fsampling=Fdts/32, N=8.	0000b
7	MSM	RW	Master/Slave mode selection: 1: The event on the trigger input (TRGI) is delayed to allow perfect synchronization between the current timer (via TRGO) and its slave timer. This is very useful when it is required to synchronize several timers to a single external event; 0: Not action.	0
[6:4]	TS[2:0]	RW	Trigger selection; these 3 bits select the trigger input source used to synchronize the counter. 000: Internal trigger 0 (ITR0); 001: Internal trigger 1 (ITR1); 010: Internal trigger 2 (ITR2); 011: Internal trigger 3 (ITR3);	000b

			100: Edge detector of TI1 (TI1F_ED); 101: Timer input 1 (TI1FP1) after filtering; 110: Timer input 2 (TI12FP2) after filtering; 111: External trigger input (ETRF); The values can be changed only when SMS is 0.	
3	Reserved	RO	Reserved	0
[2:0]	SMS[2:0]	RW	Input mode selection. Select the clock and trigger mode of the core counter. 000: Driven by the internal clock CK_INT; 001: Encoder mode1; depending on TI1FP1 level, the core counter counts up or down on edge of TI2FP2; 010: Encoder mode2; depending on TI2FP2 level, the core counter counts up or down on edge of TI1FP1; 011: Encoder mode3; depending on the input level of another signal, the core counter counts up and down on the edge of TI1FP1 and TI2FP2; 100: Reset mode; the rising edge of the trigger input (TRGI) will initialize the counter and generate a signal for updating the register; 101: Gating mode; when the trigger input (TRGI) is high, the clock of the counter will be turned on; when the trigger input becomes low, the counter will stop, and the start and stop of the counter will be controlled; 110: Trigger mode; the counter starts on the rising edge of the trigger input TRGI, and only the start of the counter is controlled; 111: External clock mode1; the rising edge of the selected trigger input (TRGI) drives the counter.	000b

15.4.4 DMA/Interrupt Enable Register (TIMx_DMAINTENR) (x=2/3/4)

Offset address: 0x0C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TDE	Reserved	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Reserved	TIE	Reserved	CC4IE	CC3IE	CC2IE	CC1IE	UIE

Bit	Name	Access	Description	Reset value
15	Reserved	RO	Reserved	0
14	TDE	RW	Trigger DMA request enable. 1: Trigger DMA request enabled; 0: Trigger DMA request disabled.	0
13	Reserved	RW	Reserved	0
12	CC4DE	RW	DMA request enable of compare/capture4. 1: DMA request of compare/capture4 enabled; 0: DMA request of compare/capture4 disabled.	0

11	CC3DE	RW	DMA request enable of compare/capture3. 1: DMA request of compare/capture3 enabled; 0: DMA request of compare/capture3 disabled.	0
10	CC2DE	RW	DMA request enable of compare/capture2. 1: DMA request of compare/capture2 enabled; 0: DMA request of compare/capture2 disabled.	0
9	CC1DE	RW	DMA request enable of compare/capture1. 1: DMA request of compare/capture1 enabled; 0: DMA request of compare/capture1 disabled.	0
8	UDE	RW	Update DMA request enable. 1: Update DMA request enabled; 0: Update DMA request disabled.	0
7	Reserved	RO	Reserved	0
6	TIE	RW	Trigger interrupt enable. 1: Trigger interrupt enabled; 0: Trigger interrupt disabled.	0
5	Reserved	RO	Reserved	0
4	CC4IE	RW	Interrupt enable of compare/capture4. 1: Interrupt of compare/capture4 enabled; 0: Interrupt of compare/capture4 disabled.	0
3	CC3IE	RW	Interrupt enable of compare/capture3. 1: Interrupt of compare/capture3 enabled; 0: Interrupt of compare/capture3 disabled.	0
2	CC2IE	RW	Interrupt enable of compare/capture2. 1: Interrupt of compare/capture2 enabled; 0: Interrupt of compare/capture2 disabled.	0
1	CC1IE	RW	Interrupt enable of compare/capture1. 1: Interrupt of compare/capture1 enabled; 0: Interrupt of compare/capture1 disabled.	0
0	UIE	RW	Update interrupt enable. 1: Update interrupt enabled; 0: Update interrupt disabled.	0

15.4.5 Interrupt Flag Register (R16_TIMx_INTFR) (x=2/3/4)

Offset address: 0x10

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CC4OF	CC3OF	CC2OF	CC1OF	Reserved		TIF	Reserved	CC4IF	CC3IF	CC2IF	CC1IF	UIF

Bit	Name	Access	Description	Reset value
[15:13]	Reserved	RO	Reserved	0
12	CC4OF	RW0	Compare/capture4 recapture flag.	0
11	CC3OF	RW0	Compare/capture3 recapture flag.	0
10	CC2OF	RW0	Compare/capture2 recapture flag.	0

9	CC1OF	RW0	Compare/capture1 recapture flag is only used when the compare/capture is configured in the input capture mode. This flag bit is set by the hardware, write 0 by software to clear the bit. 1: When the value of the counter is captured into the capture comparison register, the status of CC1IF has been set; 0: No recapture is generated.	0
[8:7]	Reserved	RO	Reserved	0
6	TIF	RW0	Trigger interrupt flag. When a trigger event occurs, set by hardware and cleared by software. Trigger events include the detection of a valid edge at the TRGI input terminal from modes other than gating mode, or any edge in gating mode. 1: Trigger event occurs; 0: No trigger event occurs.	0
5	Reserved	RO	Reserved.	0
4	CC4IF	RW0	Compare/capture 4 interrupt flag.	0
3	CC3IF	RW0	Compare/capture 3 interrupt flag.	0
2	CC2IF	RW0	Compare/capture 2 interrupt flag.	0
1	CC1IF	RW0	Compare/capture 1 interrupt flag. If the compare/capture is configured as the output mode, this bit is set by hardware when the counter value matches the compare value, except in center-aligned mode. This bit is cleared by software. 1: The value of core counter matches the value of compare/capture register 1; 0: No. If the compare/capture is configured as the output mode, this bit is set by hardware when a capture event occurs, and it is cleared by software or cleared by reading the compare/capture register. 1: The counter value has been captured by the compare/capture register 1; 0: No input capture is generated.	0
0	UIF	RW0	Update interrupt flag. When an update event occurs, it is set by hardware and cleared by software. 1: Update interrupt generated; 0: No update interrupt generated. The update event generates in case of the following circumstances: For UDIS=0, when the repeated counter value overflows or underflows; For URS=0, UDIS=0, when the UG bit is set, or when the	0

			counter core is reinitialized by software; For URS=0, UDIS=0, when the counter CNT is reinitialized by a trigger event.	

15.4.6 Event Generation Register (TIMx_SWEVGR) (x=2/3/4)

Offset address: 0x14

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									TG	Reserved	CC4G	CC3G	CC2G	CC1G	UG

Bit	Name	Access	Description	Reset value
[15:7]	Reserved	RO	Reserved.	0
6	TG	WO	Trigger event generation. Set by software, and cleared by hardware to generate a trigger event. 1: A trigger event generated; if TIF is set and the corresponding interrupt and DMA are enabled, the corresponding interrupt and DMA will be generated; 0: No effect.	0
5	Reserved	RO	Reserved.	0
4	CC4G	WO	Compare/capture 4 generation.	0
3	CC3G	WO	Compare/capture 3 generation.	0
2	CC2G	WO	Compare/capture 2 generation.	0
1	CC1G	WO	Compare/capture1 generation. This bit is set by software and cleared by hardware. It is used to generate a compare/capture event. 1: Generate compare/capture event on channel 1: If compare/capture 1 is configured as output: Set the CC1IF bit. If the corresponding interrupt and DMA are enabled, the corresponding interrupt and DMA will be generated; If compare/capture 1 is configured as input, the current core counter value is captured to compare/capture register 1; set the CC1IF bit, if the corresponding interrupt and DMA are enabled, the corresponding interrupt and DMA will be generated. If the CC1IF bit has been set, set the CC1OF bit. 0: No effect.	0
0	UG	WO	Update event generation. This bit is set by software and cleared automatically by hardware. 1: Initialize the counter and generate an update event; 0: No effect. *Note: The counter of the prescaler is also cleared, but the prescaler factor remains unchanged. In Centro symmetric mode or up-counting mode, the core counter*	0

			will be cleared; in the down-counting mode, the core counter will take the value of the reload value register.	

15.4.7 Compare/Capture Control Register 1 (TIMx_CHCTLR1) (x=2/3/4)

Offset address: 0x18

The channel can be used for input (capture mode) or output (compare mode), and the direction of the channel is defined by the corresponding CCxS bit. The functions of other bits of this register are different in input and output modes. OCxx describes the function of the channel in output mode, and ICxx describes the function of the channel in input mode.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]				IC1PSC[1:0]			

Compare mode (pin direction is output):

Bit	Name	Access	Description	Reset value
15	OC2CE	RW	Compare/capture 2 clear enable. 1: Once the ETRF input high level is detected, clear the OC2REF bit to zero; 0: OC2REF is not affected by the ETRF input.	0
[14:12]	OC2M[2:0]	RW	Compare/capture2 mode setting. These 3 bits define the action of the output reference signal OC2REF, and OC2REF determines the value of OC2 and OC2N. OC2REF is active at high level, while the active level of OC2 and OC2N depends on the CC2P and CC2NP bits. 000: Frozen. The comparison value between the value of the compare/capture register and the core counter has no effect on OC2REF; 001: Active by force. When the core counter and compare/capture register1 have the same value, force OC2REF to be high; 010: Set as inactive level by force. When the value of the core counter is the same as compare/capture register 1, force OC2REF to be low; 011: Overturn. When the core counter and compare/capture register1 have the same value, overturn the level of OC2REF; 100: Inactive by force. Force OC2REF to be low. 101: Force to be active level. Force OC2REF to be high. 110: PWM Mode 1: In up count, once the core counter is less than the value of the compare capture register, channel 2 is valid level, otherwise it is invalid level; in	000b

			downward counting, once the core counter is greater than the value of the compare capture register, channel 2 is invalid level (OC2REF=0), otherwise it is valid level (OC2REF=1). 111: PWM mode 2: In up count, once the core counter is smaller than the value of the compare capture register, channel 2 is invalid level, otherwise it is valid level; in down count, once the core counter is larger than the value of the compare capture register, channel 2 is valid level (OC2REF=1), otherwise it is invalid level (OC2REF=0). *Note: Once the LOCK level is set to 3 and CC2S=00b, this bit cannot be modified. In PWM mode1 or PWM mode2, the OC2REF level changes only when the comparison result changes or when switching from freezing mode to PWM mode in output compare mode.*	
11	OC2PE	RW	Compare/capture register 2 preload enable. 1: Enable the preload function of the compare/capture register. Read and write operations are only made on the preload register. The preload value of the compare/capture register 2 is loaded into the current shadow register when the update event arrives; 0: Disable the pre-loading function of compare/capture register 2. Compare/capture register 2 can be written at any time, and the newly written value takes effect immediately. *Note: Once the LOCK level is set to 3 and CC2S=00b, this bit cannot be modified; only in single pulse mode (OPM=1) you can use PWM mode without confirming the preload register; otherwise its action is uncertain.*	0
10	OC2FE	RW	Compare/capture 2 fast enable. It is used to speed up the response of the compare/capture output to the trigger input event. 1: The effect of the inactive edge inputted to the trigger is like a comparison match. Therefore, OC is set to the comparison level regardless of the comparison result. The delay between the valid edge of the sampling trigger and the output of compare/capture 2 is shortened to 3 clock cycles; 0: According to the value of counter and compare/capture register 2, compare/capture 2 operates normally, even if the trigger is turned on. When the input of the trigger has a valid edge, the minimum delay for activating the output of the	0

			compare/capture 2 is 5 clock cycles. OC2FE only works when the channel is configured in PWM1 or PWM2 mode.	
[9:8]	CC2S[1:0]	RW	Compare/capture 2 input selection. 00: The compare/capture 2 is configured as output; 01: Compare/capture 2 is configured as input, and IC2 is mapped on TI2; 10: Compare/capture 2 is configured as input, and IC2 is mapped on TI1; 11: Compare/capture 2 is configured as an input, and IC2 is mapped on TRC. This mode only works when the internal trigger input is selected (selected by the TS bit). *Note: Compare/capture 2 is only writable when the channel is switched off (CC2E is zero).*	00b
7	OC1CE	RW	Compare/capture 1 clear enable.	0
[6:4]	OC1M[2:0]	RW	Compare/capture 1 mode setting.	0
3	OC1PE	RW	Compare/capture register 1 preload enable.	0
2	OC1FE	RW	Compare/capture 1 fast enable.	0
[1:0]	CC1S[1:0]	RW	Compare/capture 1 input selection.	0

Capture mode (pin direction is input):

Bit	Name	Access	Description	Reset value
[15:12]	IC2F[3:0]	RW	Input capture2 filter configuration. These bits set the sampling frequency and digital filter length of TI1 input. The digital filter is composed of an event counter, in which N events are needed to validate a transition on the output. 0000: No filter, sampling is done at Fdts; 1000: Fsampling=Fdts/8, N=6; 0001: Fsampling=Fck_int, N=2; 1001: Fsampling=Fdts/8, N=8; 0010: Fsampling=Fck_int， N=4; 1010: Fsampling=Fdts/16, N=5; 0011: Fsampling=f=Fck_int, N=8; 1011: Fsampling=Fdts/16, N=6; 0100: Fsampling=Fdts/2, N=6; 1100: Fsampling=Fdts/16, N=8; 0101: Fsampling=Fdts/2, N=8; 1101: Fsampling=Fdts/32, N=5; 0110: Fsampling=Fdts/4, N=6; 1110: Fsampling=Fdts/32, N=6; 0111: Fsampling=Fdts/4, N=8; 1111: Fsampling=Fdts/32, N=8.	0000b

[11:10]	IC2PSC[1:0]	RW	Compare/capture 2 prescaler configuration. These 2 bits define the prescaler factor of compare/capture 2. Once CC1E=0, the prescaler will be reset. 00: Prescaler OFF, each edge detected on the capture input port triggers a capture; 01: Trigger a capture every 2 events; 10: Trigger a capture every 4 events; 11: Trigger a capture every 8 events.	00b
[9:8]	CC2S[1:0]	RW	Compare/capture2 input selection. These 2 bits define the direction of the channel (input/output) and selection of input pins. 00: Compare/capture1 is configured as output; 01: Compare/capture1 is configured as input, and IC1 is mapped on TI1; 10: Compare/capture1 is configured as input, and IC1 is mapped on TI2; 11: Compare/capture1 is configured as an input, and IC1 is mapped on TRC. This mode only works when the internal trigger input is selected (selected by the TS bit). *Note: CC1S can be written only when the channel is closed (CC1E is 0).*	00b
[7:4]	IC1F[3:0]	RW	Input capture1 filter configuration.	0
[3:2]	IC1PSC[1:0]	RW	Compare/capture1 prescaler configuration.	0
[1:0]	CC1S[1:0]	RW	Compare/capture1 input selection.	0

15.4.8 Compare/Capture Control Register 2 (TIMx_CHCTLR2) (x=2/3/4)

Offset address: 0x1C

The channel can be used for input (capture mode) or output (comparison mode), and the direction of the channel is defined by the corresponding CCxS bit. The functions of other bits of this register are different in input and output modes. OCxx describes the function of the channel in output mode, and ICxx describes the function of the channel in input mode.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M[2:0]			OC4PE	OC4FE	CC4S[1:0]		OC3CE	OC3M[2:0]			OC3PE	OC3FE	CC3S[1:0]	
IC4F[3:0]				IC4PSC[1:0]				IC3F[3:0]				IC3PSC[1:0]			

Compare mode (pin direction is output):

Bit	Name	Access	Description	Reset value
15	OC4CE	RW	Compare/capture 4 clear enable	0
[14:12]	OC4M[2:0]	RW	Compare/capture 4 mode setting	0
11	OC4PE	RW	Compare/capture 4 preload enable	0
10	OC4FE	RW	Compare/capture 4 fast enable	0

[9:8]	CC4S[1:0]	RW	Compare/capture 4 input selection	0
7	OC3CE	RW	Compare/capture 3 clear enable	0
[6:4]	OC3M[2:0]	RW	Compare/capture 3 mode setting	0
3	OC3PE	RW	Compare/capture 3 preload enable	0
2	OC3FE	RW	Compare/capture 3 fast enable	0
[1:0]	CC3S[1:0]	RW	Compare/capture 3 input selection	0

Capture mode (pin direction is input):

Bit	Name	Access	Description	Reset value
[15:12]	IC4F[3:0]	RW	Input capture 4 filter configuration	0
[11:10]	IC4PSC[1:0]	RW	Compare/capture 4 prescale configuration	0
[9:8]	CC4S[1:0]	RW	Compare/capture 4 input selection	0
[7:4]	IC3F[3:0]	RW	Input capture 3 filter configuration	0
[3:2]	IC3PSC[1:0]	RW	Compare/capture 3 prescale configuration	0
[1:0]	CC3S[1:0]	RW	Compare/capture 3 input selection	0

15.4.9 Compare/Capture Enable Register (TIMx_CCER) (x=2/3/4)

Offset address: 0x20

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		CC4P	CC4E	Reserved		CC3P	CC3E	Reserved		CC2P	CC2E	Reserved		CC1P	CC1E

Bit	Name	Access	Description	Reset value
[15:14]	Reserved	RO	Reserved	0
13	CC4P	RW	Compare/capture 4 output polarity	0
12	CC4E	RW	Compare/capture 4 output enable	0
[11:10]	Reserved	RO	Reserved	0
9	CC3P	RW	Compare/capture 3 output polarity	0
8	CC3E	RW	Compare/capture 3 output enable	0
[7:6]	Reserved	RO	Reserved	0
5	CC2P	RW	Compare/capture 2 output polarity	0
4	CC2E	RW	Compare/capture 2 output enable	0
[3:2]	Reserved	RO	Reserved	0
1	CC1P	RW	Compare/capture 1 output polarity	0
0	CC1E	RW	Compare/capture 1 output enable	0

15.4.10 Counter of General-purpose Timer (TIMx_CNT) (x=2/3)

Offset address: 0x24

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

Bit	Name	Access	Description	Reset value

[15:0]	CNT[15:0]	RW	Real-time value of timer counter.	0

15.4.11 Counter of General-purpose Timer (TIMx_CNT) (x=4)

Offset address: 0x24

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

Bit	Name	Access	Description	Reset value
[31:0]	CNT[31:0]	RW	Real-time value of timer counter.	0

15.4.12 Prescaler (TIMx_PSC) (x=2/3/4)

Offset address: 0x28

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

Bit	Name	Access	Description	Reset value
[15:0]	PSC[15:0]	RW	The frequency division factor of the timer's prescaler; the clock frequency of the counter is equal to the input frequency of the frequency divider/(PSC+1).	0

15.4.13 Auto-reload Register (TIMx_ATRLR) (x=2/3)

Offset address: 0x2C

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

Bit	Name	Access	Description	Reset value
[15:0]	ARR[15:0]	RW	The value of ATRLR[15:0] is loaded into the counter. Please refer to Section 15.2.4 for ATRLR acting and update time. When ATRLR is empty, the counter stops.	0xFFFF

15.4.14 Auto-reload Register (TIMx_ATRLR) (x=4)

Offset address: 0x2C

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

Bit	Name	Access	Description	Reset value
[31:0]	ARR[31:0]	RW	The value of ATRLR[31:0] is loaded into the counter. Please refer to Section 15.2.4 for ATRLR acting and update time. When ATRLR is empty, the counter stops.	0xFFFFFFFF

15.4.15 Compare/Capture Register 1 (TIMx_CH1CVR) (x=2/3)

Offset address: 0x34

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15：0]															

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	LEVEL1	RO	The level indicator bit corresponding to the capture value	0
[15:0]	CCR1[15:0]	RW	Compare/capture register channel 1.	0

15.4.16 Compare/Capture Register 1 (TIMx_CH1CVR) (x=4)

Offset address: 0x34

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR1[31：16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15：0]															

Bit	Name	Access	Description	Reset value
[31:0]	CCR1[31:0]	RW	Compare/capture register channel 1.	0

15.4.17 Compare/Capture Register 2 (TIMx_CH2CVR) (x=2/3)

Offset address: 0x38

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL2
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15：0]															

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	LEVEL2	RO	The level indicator bit corresponding to the capture	0

			value	
[15:0]	CCR2[15:0]	RW	Compare/capture register channel 2.	0

15.4.18 Compare/Capture Register2 (TIMx_CH2CVR) (x=4)

Offset address: 0x38

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR2[31：16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15：0]															

Bit	Name	Access	Description	Reset value
[31:0]	CCR2[31:0]	RW	Compare/capture register channel 2.	0

15.4.19 Compare/Capture Register3 (TIMx_CH3CVR) (x=2/3)

Offset address: 0x3C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL3
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15：0]															

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	LEVEL3	RO	The level indicator bit corresponding to the capture value	0
[15:0]	CCR3[15:0]	RW	Compare/capture register channel 3.	0

15.4.20 Compare/Capture Register3 (TIMx_CH3CVR) (x=4)

Offset address: 0x3C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR3[31：16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15：0]															

Bit	Name	Access	Description	Reset value
[31:0]	CCR3[31:0]	RW	Compare/capture register channel 3.	0

15.4.21 Compare/Capture Register4 (TIMx_CH4CVR) (x=2/3)

Offset address: 0x40

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL4

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15：0]															

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	LEVEL4	RO	The level indicator bit corresponding to the capture value	0
[15:0]	CCR4[15:0]	RW	Compare/capture register channel 4.	0

15.4.22 Compare/Capture Register4 (TIMx_CH4CVR) (x=4)

Offset address: 0x40

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR4[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															

Bit	Name	Access	Description	Reset value
[31:0]	CCR4[31:0]	RW	Compare/capture register channel 4.	0

15.4.23 DMA Control Register (TIMx_DMACFGR) (x=2/3/4)

Offset address: 0x48

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

Bit	Name	Access	Description	Reset value
[15:13]	Reserved	RO	Reserved	0
[12:8]	DBL[4:0]	RW	Length of data that DMA continuously transfers; the actual value is the value of this domain + 1.	0
[7:5]	Reserved	RO	Reserved.	0
[4:0]	DBA[4:0]	RW	These bits define the offset of DMA from the address of control register1 in continuous mode.	0

15.4.24 DMA Address Register in Continuous Mode (TIMx_DMAADR) (x=2/3/4)

Offset address: 0x4C

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

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

DMAB[15:0]

Bit	Name	Access	Description	Reset value
[31:0]	DMAB[31:0]	RW	DMA address in continuous mode.	0

Chapter 16 Low-power Timer (LPTIM)

LPTIM is a 16-bit uplink count timer. LPTIM has a variety of optional clock sources that allow LPTIM to operate in all power modes except standby mode. LPTIM can also run without an internal clock source, so LPTIM can be used as a "pulse counter". In addition, LPTIM can also wake up the system from low-power mode, so LPTIM is very suitable for implementing "timeout" at very low power consumption.

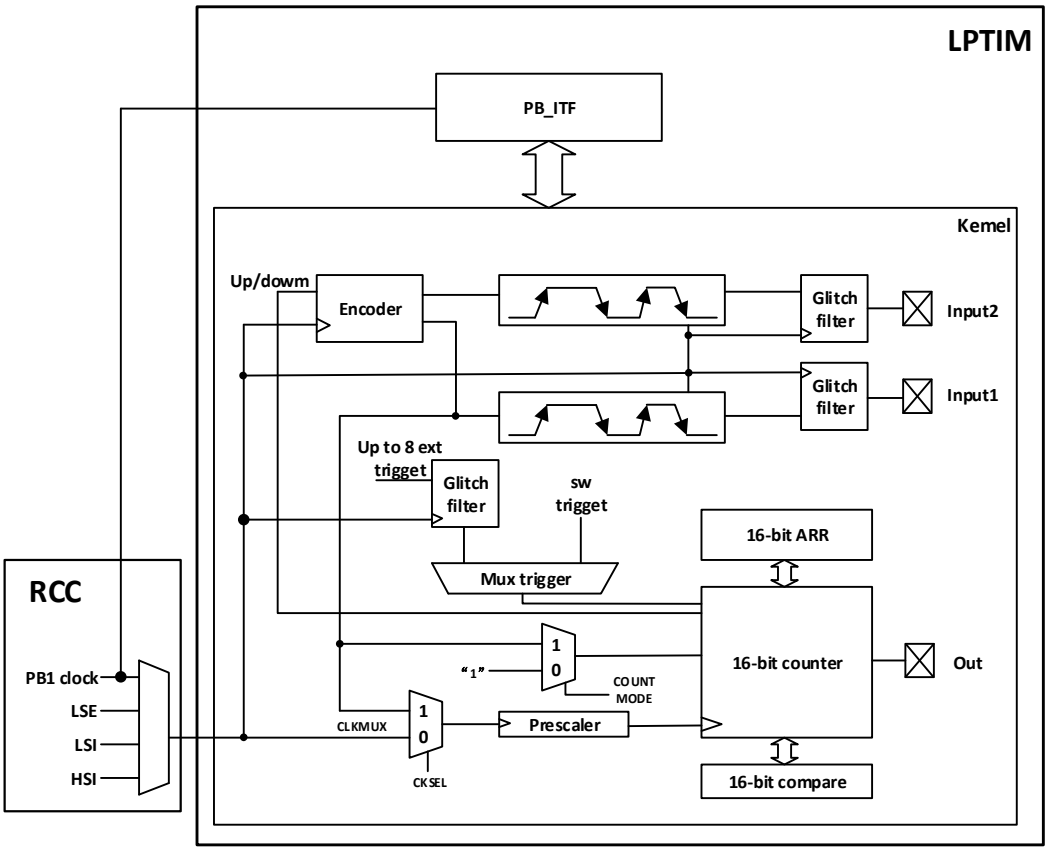
16.1 Main Features

- 16-bit uplink counter
- 3-bit prescaler, supporting 8 frequency division coefficients (1, 2, 4, 8, 16, 32, 64, 128)
- Optional clock source
  Internal clock source: LSE, LSI, HSI or PB1 clock
  External clock source: external clock on LPTIM input
- 16-bit ARR auto-reload register
- 16-bit comparison register
- Continuous / single trigger mode
- Optional software / hardware input trigger
- Programmable digital interference filter
- Configurable output PWM wave
- Configurable I/O polarity
- Encoder mode

16.2 Function Description

16.2.1 LPTIM Structure

Figure 16-1 Block diagram of LPTIM



16.3 LPTIM Trigger Mapping

The information for LPTIM external trigger connections is as follows:

Table 16-1 LPTIM external trigger connection

TRIGSEL[1:0]	External trigger
LPTIM_TRG_00	LPTIM_ETR(PB6/PB14)
LPTIM_TRG_01	RTC_ALARM
LPTIM_TRG_10	TAMP(PC13)

16.3.1 LPTIM Reset and Clock

The reset of the LPTIM module is controlled by the LPTIMRST bit of the RCC_PB1PRSTR register, which has no effect when setting 0 and resets the module when setting 1.

The LPTIM module clock is controlled by the LPTIMEN bit of the RCC_PB1PCENR register. The module clock is turned off at 0 and turned on at 1.

The counting clock of LPTIM is provided by multiple optional clock sources, which can be divided into internal clock sources and external clock sources.

When using the internal clock source count, the internal clock source can select the four PB1, LSI, LSE, HIS clock sources through the CLKMX_SEL bit of the LPTIM_CFGR register. In addition, the LPTIM can be timed using an external clock signal injected on the external input LPTIM_CH1 (PB5/PB12).

Table 16-2 LPTIM internal clock source

Register LPTIM_CFGR[26:25]	Clock source
00	TIM_CLK (From PB1_CLK)
01	HSI_CLK
10	LSE_CLK
11	LSI_CLK

When using external clock source counting, LPTIM may run in one of two configurations:

The first configuration is that the clock is provided to the LPTIM by the external signal, and the internal clock signal is provided by the configurable internal clock source (PB1, LSI, LSE, HSI, etc.).

The second configuration is that the LPTIM only uses the external clock source through its external input channel 1, which realizes the pulse counter function or timeout function when all internal clock sources are turned off after entering the low power mode.

Table 16-3 LPTIM external clock source

Register LPTIM_CFGR[2:1]	Clock source
00	LPTIM_CH1（PB5/PB12）
01	~LPTIM_CH1（PB5/PB12）
Other	None

Programming the CKSEL and COUNTMODE bits controls whether the LPTIM clock uses an external clock source or an internal clock source. When configured to select an external clock source, the CKPOL bit is used to select the effective edge of the external clock signal. If the effective edge is set to the rising edge and the falling edge (double edge), an internal clock signal is also provided. In this case, the internal clock signal frequency should be four times higher than the external clock signal frequency.

16.3.2 Filter

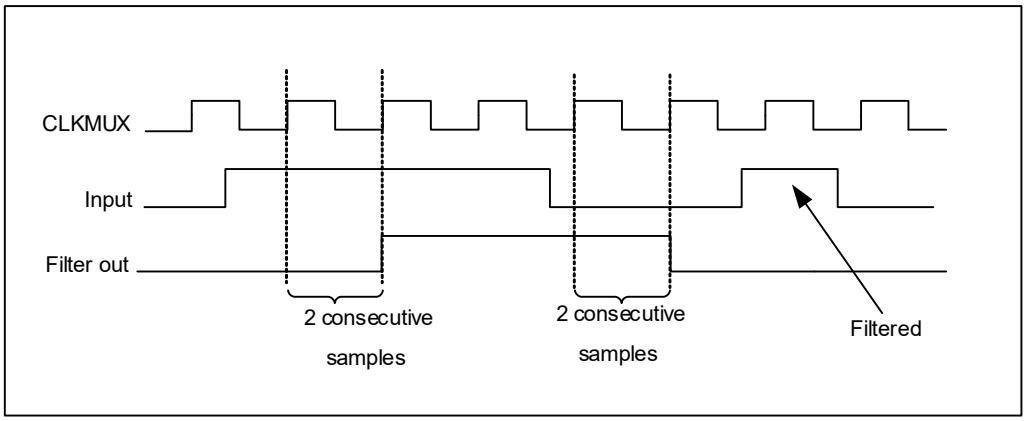
LPTIM inputs, both external and internal, are protected by digital filters to prevent any burr and noise interference from propagating within the LPTIM, thereby avoiding accidental counting or triggering.

Before activating the digital filter, the internal clock source should be provided to the LPTIM to ensure the normal operation of the filter.

Digital filters are divided into 2 groups:
One group protects the LPTIM external input for the digital filter, and the sensitivity of the digital filter is controlled by the CKFLT bit; the other set of digital filter protects the internal trigger input of the LPTIM, and the sensitivity of the digital filter is controlled by the TRGFLT bit.

The sensitivity of the filter will affect the same number of continuous samples, and only when such continuous samples are detected on one of the LPTIM inputs can a signal level change be regarded as an effective switching. The following figure shows an example of interference filter behavior when programming two consecutive samples.

Figure 16-2 Timing block diagram of interference filter



Note: when the internal clock signal is not used, the digital filter must be disabled by zeroing the CKFLT and TRGFLT bits, and using an external analog filter to avoid interference caused by the external input of the LPTIM.

16.3.3 Prescaler

There should be a configurable 2^n prescaler in front of the LPTIM16 bit counter. The division ratio of the prescaler is controlled by the 3-bit field of PRESC [2:0]. All possible division ratios are listed in Table 16-2.

Table 16-4 Frequency division ratio of prescaler

PRESC[2:0]	Division ratio
000	1
001	1/2
010	1/4
011	1/8
100	1/16
101	1/32
110	1/64
111	1/128

16.3.4 Trigger Multiplexer

The LPTIM counter can be started in two ways, one is started by software, and the other is started after more than one valid edge in the trigger input is detected. The trigger mode of LPTIM is controlled by TRIGEN [1:0] and the trigger source is controlled by TRIGSEL [1:0] bit.

Table 16-5 Trigger mode

TRIGEN[1:0]	Trigger mode
00	Invalid
01	Rising edge
10	Falling edge
11	Two-sided edge

Table 16-6 Trigger source

TRIGSEL[1:0]	Trigger source
00	LPTIM_ETR(PB6/PB14)
01	RTC_ALARM
10	TAMP(PC13)
11	Invalid

16.3.5 Operating Mode

LPTIM has two modes of operation:

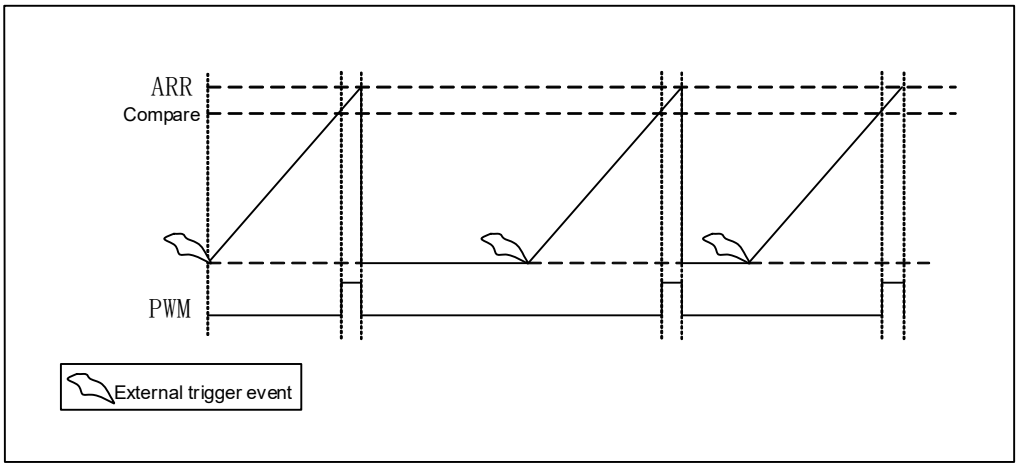
Continuous mode: the timer runs freely, starting from the trigger event and not stopping until the timer is disabled.

Single trigger mode: the timer starts from the trigger event and stops when the ARR value is reached.

In single trigger mode, to enable single counting, you must set SNGSTRT location 1, a new trigger event will restart the timer, and any trigger events that occur after the counter starts until it reaches ARR will be lost.

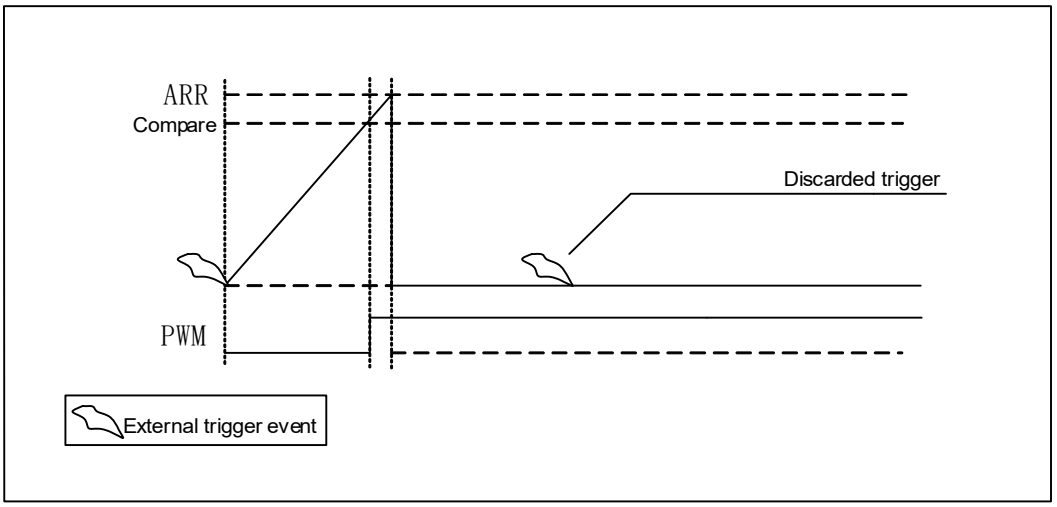
When an external trigger is selected, each external trigger event that arrives after the SNGSTRT bit is set and after the counter register stops starts the counter for a new count loop.

Figure 16-3 LPTIM output waveforms in single count mode configuration



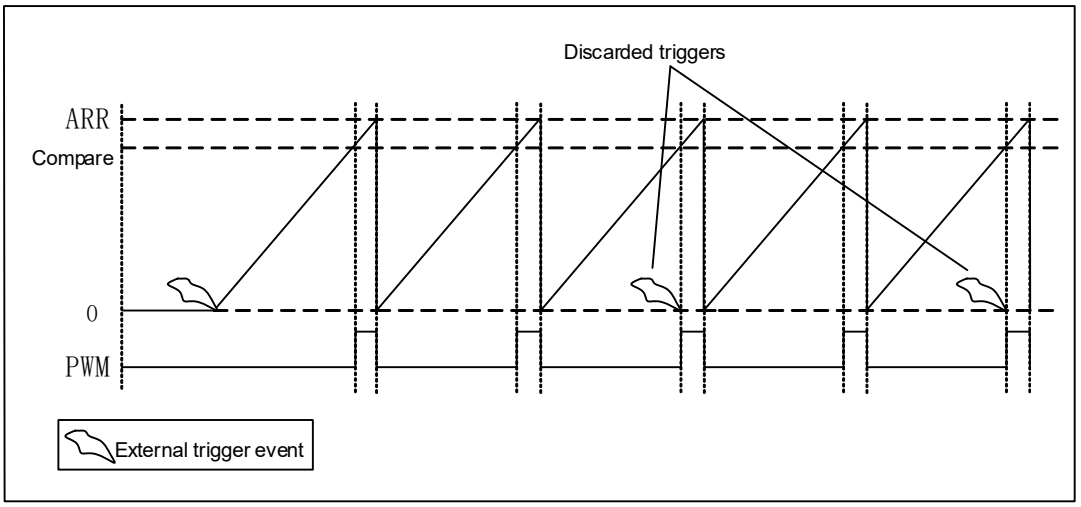
To activate one-trigger setup mode, it should be noted that in one-trigger mode, one-time setup mode is activated when the WAVE bit field in the LPTIMx_CFGR register is set. In this case, the counter is activated only once after the first trigger and any subsequent trigger events are discarded.

Figure 16-4 LPTIM output waveform for single count mode configuration with one setup mode activated



In continuous mode, to enable continuous counting, the CNTSTRT bit must be set, and if an external trigger is selected, external trigger events that arrive after setting CNTSTRT will start the counter for continuous counting. Any subsequent external trigger events will be discarded, and setting CNTSTRT in the case of software startup (TRIGEN=00) will start the counter to count continuously.

Figure 16-5 LPTIM output waveforms in continuous count mode configuration



The SNGSTRT and SNGSTRT bits can only be set when the timer is enabled (ENABLE bit is set to 1). You can change the LPTIM counter mode. If you previously selected continuous mode, setting SNGSTRT will switch LPTIM to single trigger mode, and the counter will stop counting when the ARR value is reached. If you have previously selected single trigger mode, setting CNTSTRT will switch CNTSTRT to continuous mode. Restart as soon as the counter reaches the ARR value.

16.3.6 Timeout Function

The detection of a valid edge on a selected trigger input can be used to reset the counter, which is controlled by the TIMOUT bit. The first trigger event will start the counter, any successive trigger event will reset the counter and the timer will restart, you can achieve low-power timeout function, if no trigger event occurs, the MCU will be awakened by the comparison match event.

16.3.7 Waveform Generation

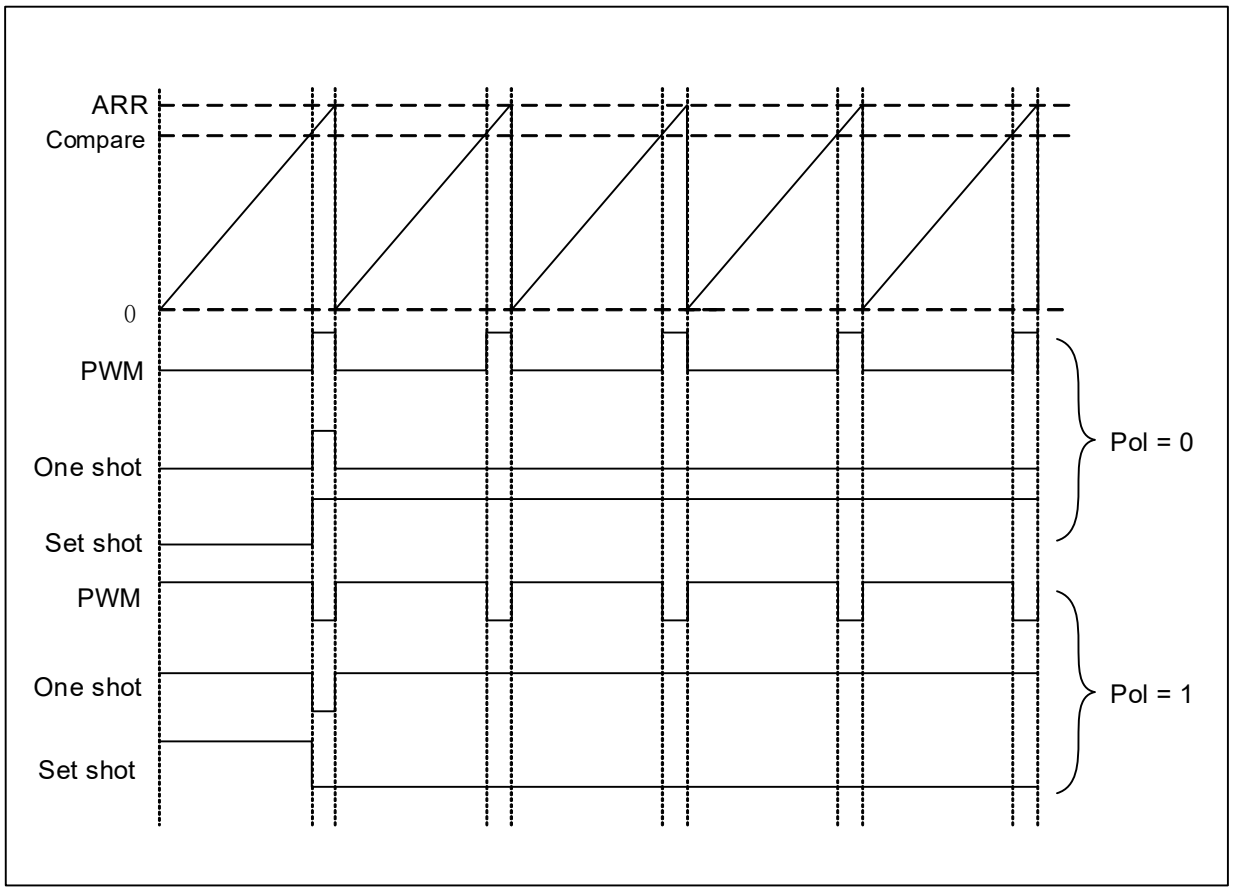
Two 16-bit registers LPTIM_ARR and LPTIM_CMP are used to generate several different waveforms on the LPTIM output.

The timer can generate the following waveforms:
(1) PWM mode: once the counter value in LPTIM_CNT exceeds the comparison value in LPTIM_CMP, the LPTIM output is set. Once the LPTIM_ARR and LPTIM_CNT register values are equal, the LPTIM output is reset.
(2) Mono-pulse mode: the output waveform is similar to the PWM mode of the first pulse, and then permanently reset.
(3) one-time setting mode: the output waveform is similar to the mono-pulse mode, except that the output remains at the last signal level (depending on the polarity of the output configuration).

The above mode requires that the LPTIM_ARR register value is strictly greater than the LPTIM_CMP register value.

The LPTIM output waveform can be configured through the Wave bit as follows:
(1) resetting the WAVE bit to 0 forces LPTIM to generate a PWM waveform or a mono-pulse waveform, depending on the bit set: CNTSTRT or SNGSTRT.
(2) setting the WAVE bit to 1 forces LPTIM to generate a setting mode once.
The WAVPOL bit controls the polarity of the LPTIM output, and the change takes effect immediately, so the output default value changes immediately after the polarity is reconfigured, even before the timer is enabled.
The frequency of the generated signal is as high as the LPTIM clock frequency 2 division. Figure 16-6 shows three waveforms that may be generated on the LPTIM output. In addition, the figure shows the effect of changing polarity through the WAVPOL bit.

Figure 16-6 Waveform generation



16.3.8 Register Update

The LPTIM_ARR register and the LPTIM_CMP register are updated immediately after the PB bus write operation, or if the timer has been started, at the end of the current cycle.

The PRELOAD bit controls how the LPTIM_ARR and LPTIM_CMP registers are updated:

When the PRELOAD bit is set to 0, the LPTIM_ARR and LPTIM_CMP registers are updated immediately after any write access.

When the PRELOAD bit is set to 1, the LPTIM_ARR and LPTIM_CMP registers are updated at the end of the current cycle if the timer has been started.

The LPTIM PB interface and LPTIM logic use different clocks, so there is some delay between PB writes and the values available to counter comparators, during which any additional writes to these registers must be avoided.

The ARROK flag and the CMPOK flag in the LPTIM_ISR register indicate when to complete the writing operation to the LPTIM_ARR register and the LPTIM_CMP register, respectively.

After writing to the LPTIM_ARR register or LPTIM_CMP register, a new write operation to the same register can be performed only when the previous write operation is completed.

Any consecutive writes before setting the ARROK flag or the CMPOK flag will result in unpredictable results.

16.3.9 Counter Mode

LPTIM counters can be used to count external events on the LPTIM_CH1 as well as internal clock cycles. The

CKSEL and count bits control which source will be used to update the counter.

If the LPTIM is configured to count external events on the LPTIM_CH1, the count can be updated after the rising, falling, or double edges based on the values written to the CKPOL [1:0] bit.

Depending on the CKSEL and COUNTMODE values, you can choose the following counting modes:
(1) CKSEL=0: LPTIM clocks are counted by internal clock sources.
The COUNTMODE=0, LPTIM is configured to count the internal clock source, and the LPTIM counter is configured to update after each internal clock pulse.
COUNTMODE=1, sampling the data on the LPTIM_CH1 using the internal clock provided to the LPTIM. Therefore, in order not to miss any events, the frequency of the signal on the LPTIM_CH1 should not exceed the frequency of the internal clock of the LPTIM.
(2) when CKSEL=1: LPTIM is counted by an external clock source, the value of COUNTMODE has no effect.

In this configuration, the LPTIM does not require an internal clock source (unless a digital filter is enabled), and the signal injected on the LPTIM_CH1 is used as the system clock of the LPTIM. This configuration applies to operating modes where the built-in oscillator is not enabled.

For this configuration, the LPTIM counter can be updated at the rising or falling edges of the LPTIM_CH1 clock signal, but not at the double edges. Because the signal injected by LPTIM_CH1 is also used for LPTIM counting, there is some initial delay before the counter is incremented (after LPTIM is enabled). To be exact, after enabling the LPTIM, the first five valid edges of the LPTIM external LPTIM_CH1 will be lost.

16.3.10 Timer Enabled

The ENABLE bit in the LPTIM_CR register is used to enable / disable LPTIM logic, and after setting the ENABLE bit, a delay of two counter clocks is required before LPTIM is actually enabled.

16.3.11 Encoder Mode

This mode allows processing of signals from a quadrature encoder used to detect the position of the rotating element. The encoder interface mode is only used as an external clock with direction selection. This means that the counter counts only continuously between 0 and the automatic overload value in the LPTIM_ARR register. Therefore, LPTIM_ARR. Net must be configured before startup. A clock signal is generated based on two external input signals LPTIM_CH1 and LPTIM_CH2 to time the LPTIM counter. The phase between the two signals determines the counting direction.

Encoder mode is available only if the LPTIM is counted by an internal clock source. The signal frequency on the LPTIM_CH1 and LPTIM_CH2 must not exceed the frequency of the LPTIM internal clock by 4 divisions. The above conditions are met to ensure the normal operation of LPTIM.

The direction change is signaled by 2 up and down flags in the LPTIM_ISR register. In addition, if enabled through the downside bit, interrupts can be generated for events that change in both directions.

To activate encoder mode, the ENC bit must be set to 1, and LPTIM must first be configured in continuous mode.

When the encoder mode is active, the LPTIM counter is automatically modified according to the speed and direction of the incremental encoder. Therefore, its content always represents the location of the encoder. The counting direction is indicated by the up and down marks, which corresponds to the rotation direction of the encoder rotor.

According to the edge sensitivity of CKPOL [1:0] bit configuration, the following possible combination schemes

are obtained, in which LPTIM_CH1 and LPTIM_CH2 do not switch at the same time.

Table 16-7 Configuration of edge sensitivity combination scheme

Active edge (CKPOL[1:0])	The level of the opposite signal (The input for LPTIM_CH1 is LPTIM_CH2, and the input for LPTIM_CH2 is LPTIM_CH1.)	LPTIM_CH1		LPTIM_CH2	
		Rising	Falling	Rising	Falling
Rising edge (00)	High	decrement	non-counting	incremental	non-counting
	Low	incremental	non-counting	decrement	non-counting
Falling edge (01)	High	non-counting	incremental	non-counting	decrement
	Low	non-counting	decrement	non-counting	incremental
Two-sided edge (10)	High	decrement	incremental	incremental	decrement
	Low	incremental	decrement	decrement	Incremental
11	Invalid				

16.3.12 Debug Mode

When MCU enters modal mode (core stops), LPTIM continues to work properly.

16.3.13 LPTIM Low-power Mode

Table 16-8 Effect of low Power Mode on LPTIM

Mode	Description
Sleep mode	No effect, LPTIM interruption will cause the device to exit sleep mode
Stop mode	The LPTIM peripheral is active when it is timed by LSE or LSI, and the LPTIM interrupt causes the device to exit stop mode
Standby mode	LPTIM peripherals are powered off and must be reinitialized after exiting standby mode

16.4 LPTIM Interrupt

An interrupt / wake-up occurs if the following events are enabled through the LPTIM_IER register:

- Compare match.
- Automatic overload matching.
- External trigger event.
- Automatic overload register write complete.
- Comparison register write complete.
- Direction change (encoder mode)

Table 16-9 Interrupt event

Interrupt event	Description
Compare match	An interrupt flag is generated when the value of the counter register LPTIM_CNT is equal to the value of the comparison register LPTIM_CMP
Automatic overload match	An interrupt flag occurs when the value of the counter register LPTIM_CNT

	is equal to that of the automatic reload register LPTIM_ARR
External trigger event	An interrupt flag is generated when an externally triggered event is detected
Automatic overload register write complete	An interrupt flag is generated when the write operation to the LPTIM_ARR register is completed.
Comparison register write complete	An interrupt flag is generated when the write operation to the LPTIM_CMP register is completed.
Change of counting direction	When used in encoder mode, two interrupt flags are embedded in the direction of the signal: Up sign: indicates a change in the direction of the upward count. Downward sign: indicates a change in the downward counting direction.

16.5 Register Description

Table 16-10 LPTIM-related registers

Name	Access	Description	Reset value
R32_LPTIM_ISR	0x40007C00	Interrupt status register	0x00000000
R32_LPTIM_ICR	0x40007C04	Interrupt clear register	0x00000000
R32_LPTIM_IER	0x40007C08	Interrupt enable register	0x00000000
R32_LPTIM_CFGR	0x40007C0C	Configuration register	0x00000000
R32_LPTIM_CR	0x40007C10	Control register	0x00000000
R32_LPTIM_CMR	0x40007C14	Compare register	0x00000000
R32_LPTIM_ARR	0x40007C18	Auto-reload register	0x00000001
R32_LPTIM_CNT	0x40007C1C	Counter register	0x00000000

16.5.1 Interrupt Status Register (LPTIM_ISR)

Offset address: 0x00

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								DIR_SYNC	DOWN	UP	ARROK	CMPOK	EXTTRIG	ARRM	CMPM

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved	0
7	DIR_SYNC	RO	Counter counting direction in encoder mode. 0: Count down. 1: Count up. This bit is only valid in encoder mode, that is, ENC position 1, and the DIR_EXTEN needs to be set to 1 to enable counting direction before use.	0
6	DOWN	RO	Count down.	0

			1: Counting direction from top to bottom. 0: Invalid. In encoder mode, this bit is set by hardware to notify the application of a change in counter direction from top to bottom, which can be cleared by writing 1 to the LPTIM_ICR [6] bit.	
5	UP	RO	Count up. 1: Counting direction from bottom to top. 0: Invalid. In encoder mode, this bit is set by hardware to notify the application of a change in counter direction from bottom to top, which can be cleared by writing 1 to the LPTIM_ICR [5] bit.	0
4	ARROK	RO	Auto-reload register data updated successfully. 1: Data updated successfully. 0: Invalid. This bit is set by the hardware to inform the application that the write operation of the PB bus to the LPTIM_ARR has completed successfully and can be cleared by writing 1 to the LPTIM_ICR [4] bit.	0
3	CMPOK	RO	Compare register data updated successfully. 1: Data updated successfully. 0: Invalid. This bit is set by the hardware to inform the application that the write operation of the PB bus to the LPTIM_CMR has completed successfully and can be cleared by writing 1 to the LPTIM_ICR [3] bit.	0
2	EXTTRIG	RO	External trigger edge event. 1: Valid edge input has occurred. 0: Invalid. This bit is set by hardware to notify the application that valid edge input has occurred on the selected external trigger, and this flag is not set if the trigger is ignored because the timer has been started. 1 can be cleared by writing 1 to the LPTIM_ICR [2] bit.	0
1	ARRM	RO	Automatic reload register data matching with LPTIM_CNT register data. 1: Match successfully. 0: Invalid. This bit is set by the hardware to inform the application that the value of the LPTIM_CNT register reaches the value of the LPTIM_ARR register, which can be cleared by writing 1 to the LPTIM_ICR [1] bit.	0
0	CMPM	RO	The data of the comparison register matches the data of the	0

			LPTIM_CNT register. 1: Match successfully. 0: Invalid. This bit is set by the hardware to inform the application that the value of the LPTIM_CNT register reaches the value of the LPTIM_CMR register, which can be cleared by writing 1 to the LPTIM_ICR [0] bit.	

16.5.2 Interrupt Clear Register (LPTIM_ICR)

Offset address: 0x04

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									DOWNCF	UPNCF	ARROKNCF	CMPOKNCF	EXTTRIGNCF	ARRMNCF	CMPMNCF

Bit	Name	Access	Description	Reset value
[31:7]	Reserved	RO	Reserved	0
6	DOWNCF	W1	Clear downward flag bit. Note: write 0 is invalid, write 1 clear 0 status register corresponding bit	x
5	UPCF	W1	Clear up flag bit. Note: Write 0 is invalid, write 1 clear 0 status register corresponding bit.	x
4	ARROKCF	W1	Clear auto-reload data update flag bit. Note: Write 0 is invalid, write 1 clear 0 status register corresponding bit.	x
3	CMPOKCF	W1	Clear comparator data update flag bit. Note: Write 0 is invalid, write 1 clear 0 status register corresponding bit.	x
2	EXTTRIGCF	W1	Clear external trigger edge event flag bit. Note: write 0 is invalid, write 1 clear 0 status register corresponding bit.	x
1	ARRMCF	W1	Clear auto-reload register matching flag bit. Note: write 0 is invalid, write 1 clear 0 status register corresponding bit.	x
0	CMPMCF	W1	Clear the comparison register matching flag bit. Note: write 0 is invalid, write 1 clear 0 status register corresponding bit.	x

16.5.3 Interrupt Enable Register (LPTIM_IER)

Offset address: 0x08

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									DOWNIE	UPIE	ARROKIE	CMPOKIE	EXTTRIGIE	ARRMIE	CMPMIE

Bit	Name	Access	Description	Reset value
[31:7]	Reserved	RO	Reserved	0
6	DOWNIE	RW	Downwards interrupt enable: 0: Off 1: On	0
5	UPIE	RW	Upwards interrupt enable: 0: Off 1: On	0
4	ARROKIE	RW	Auto-reload register data update successfully interrupt enable. 0: Off 1: On	0
3	CMPOKIE	RW	Compare register data update successfully interrupt enable 0: Off 1: On	0
2	EXTTRIGIE	RW	External trigger edge event input interrupt enable 0: Off 1: On	0
1	ARRMIE	RW	Auto-reload register data matching successfully interrupt enable 0: Off 1: On	0
0	CMPMIE	RW	Compare register data matching successfully interrupt enable 0: Off 1: On	0

16.5.4 Configuration Register (LPTIM_CFGR)

Offset address: 0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				FORCE_PWM	CLKMX_SEL		ENC	COUNTMODE	PRELOAD	WAVPOL	WAVE	TIMOUT	TRIGEN		Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TRIGSEL		Reserved	PRESC			Reserved	TRGFLT		Reserved	CKFLT		CKPOL		CKSEL

Bit	Name	Access	Description	Reset value
[31:28]	Reserved	RO	Reserved	0
27	FORCE_PWM	RW	Force PWM output 0: Invalid. 1: Force PWM output high level	0
[26:25]	CLKMX_SEL	RW	LPTIM counter internal clock source selection 00: TIM_CLK (from PB1_CLK) 01: HSI_CLK 10: LSE_CLK 11: LSI_CLK	0
24	ENC	RW	Encoder mode 0: Off 1: On	0
23	COUNTMODE	RW	Counter mode, select which clock source LPTIM uses to time the counter. 0: The counter increments after each internal clock pulse. 1: When each effective pulse is input outside the LPTIM, the counter increments.	0
22	PRELOAD	RW	Register update mode, control LPTIM_ARR and LPTIM_CMP register update mode. 0: Update the register after each PB bus write access. 1: The register is updated at the end of the current LPTIM cycle.	0
21	WAVPOL	RW	Polarity of PWM waveform. 0: The output reflects the comparison between LPTIM_ARR and LPTIM_CMP registers. 1: The output reflects the inversion of the comparison between the LPTIM_ARR and LPTIM_CMP registers. Note: if the counter value is greater than the comparator value, the comparison result is 1, otherwise it is 0.	0
20	WAVE	RW	PWM waveform. 0: Turn off once mode. 1: Set the primary mode (one pulse waveform).	0
19	TIMOUT	RW	Timeout is enabled to control the timeout function. 0: Trigger events that arrive when the timer has been started are ignored. 1: Trigger events that arrive when the timer has been started will reset and restart the counter.	0
[18:17]	TRIGEN	RW	Trigger enable and polarity control whether the LPTIM counter is started by an external trigger, and if the external trigger option is selected, there are three configurations for the trigger active edge. 00: Software trigger (counter start by software).	0

			01: Rising edge trigger. 10: Falling edge trigger. 11: Double edge trigger.	
[16:15]	Reserved	RO	Reserved	0
[14:13]	TRIGSEL	RW	Trigger source selection 00: LPTIM_ETR(PB6/PB14) 01: RTC_ALARM 10: TAMP(PC13) 11: Invalid.	0
12	Reserved	RO	Reserved	0
[11:9]	PRESC	RW	Clock prescaler, configured with prescaler 000: Divided by 1 001: Divided by 2 010: Divided by 4 011: Divided by 8 100: Divided by 16 101: Divided by 32 110: Divided by 64 111: Divided by 128	0
8	Reserved	RO	Reserved	0
[7:6]	TRGFLT	RW	Configurable digital filter for flip-flop. The TRGFLT value sets the number of consecutive equal samples that should be detected when there is a change on the internal trigger, which is then treated as a valid level conversion. You must have an internal clock source to use this feature. 00: Any change to a trigger is considered a valid trigger. 01: Trigger activation level change must be stable for at least 2 clock cycles before it is considered a valid trigger. 10: Trigger activation level change must be stable for at least 4 clock cycles before it is considered a valid trigger. 11: Trigger activation level change must be stable for at least 8 clock cycles before it is considered a valid trigger.	0
5	Reserved	RO	Reserved	0
[4:3]	CKFLT	RW	Configurable digital filter for external clock. The CKFLT value is used to set the number of consecutive equal samples. Such continuous samples should be detected when there is a level change in the external clock signal before the level change is regarded as an effective level conversion. You must have an internal clock source to use this feature. 00: Any change in the level of the external clock signal is considered an effective conversion. 01: External clock signal level changes must be stable for	0

			at least 2 clock cycles before they are considered valid. 10: External clock signal level changes must be stable for at least 4 clock cycles before they are considered valid. 11: External clock signal level changes must be stable for at least 8 clock cycles before they are considered valid.	
[2:1]	CKPOL	RW	If LPTIM chooses an external clock source count, the CKPOL bit is used to configure the valid edge: 00: Rising is still used in counting. 01: Falling is still used in counting. 10: Bilateral is still used for counting. When both sides of the external clock signal are valid, the LPTIM must also be timed by an internal clock source whose frequency is at least four times the external clock frequency. 11: Invalid. If LPTIM is configured in encoder mode (ENC position 1), the CKPOL bit is used to select the encoder mode: 00: Encoder mode 1 activation. 01: Encoder mode 2 activation. 10: Encoder mode 3 activated. 11: Invalid.	0
0	CKSEL	RW	Clock selector for selecting the clock used by LPTIM. 0: LPTIM uses an internal clock. 1: LPTIM uses an external clock.	0

16.5.5 Control Register (LPTIM_CR)

Offset address: 0x10

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											DIR_EXTEN	OUTEN	CNTSTRT	SNGSTRT	ENABLE

Bit	Name	Access	Description	Reset value
[31:5]	Reserved	RO	Reserved	0
4	DIR_EXTEN	RW	External trigger counting direction enable. 0: Off. 1: On. Note: Use when encoder mode is turned on.	0
3	OUTEN	RW	PWM wave output enable. 0: Off. 1: On. Note: In non-encoder mode, the bit enable output PWM wave requires ENABLE to be enabled at the same time to	0

			be effective.	
2	CNTSTRT	RW	Start in continuous mode. This bit is set by the software and cleared by the hardware. In the case of software startup (TRIGEN==00), set this bit to start the LPTIM in continuous mode, and if the software startup is disabled (TRIGENENCE00), setting this bit will start the LPTIM in continuous mode immediately after an external trigger is detected. If this bit is set when LPTIM is in mono-pulse counting mode, LPTIM will not stop counting when the counter reaches the ARR value. Note: you can only write when you ENABLE=1.	0
1	SNGSTRT	RW	LPTIM starts in single trigger mode. This bit is set by the software and cleared by the hardware. in the case of software startup (TRIGEN==00), set this bit to start the LPTIM in mono-pulse mode, and if the software startup is disabled (TRIGENENCE00), setting this bit will start the LPTIM in mono-pulse mode immediately after an external trigger is detected. If this bit is set when LPTIM is in continuous count mode, LPTIM will stop when the counter reaches the ARR value. Note: write only when you ENABLE=1.	0
0	ENABLE	RW	The timer enables, when the timer enable bit is cleared 0, the internal logic is reset and the LPT_CR [2:1] bit cannot be operated. 0: Low-power timer disables and resets internal logic. 1: Low-power timer is turned on.	0

16.5.6 Compare Register (LPTIM_CMP)

Offset address: 0x14

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

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:0]	CMP	RW	Comparison value, when the value of the counter is equal to the value in the comparison register, the timer comparison flag will be set to 1, and if the corresponding enable signal is turned on before setting 1, it will produce an interrupt source and a low power wake-up signal.	0

16.5.7 Auto-reload Register (LPTIM_ARR)

Offset address: 0x18

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

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:0]	ARR	RW	The count register reloaded count value. When counting up, if the count value is equal to the reloaded count value, the counter starts counting from 0; when counting down, if the counter value equals 0, the count starts from the reloaded count value.	1

16.5.8 Counter Register (LPTIM_CNT)

Offset address: 0x1C

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

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:0]	COUNT	RO	Count value of the current timer counter register. When LPTIM is running with an asynchronous clock, reads to the LPTIM_CNT register may return unreliable values; therefore, in this case, it is necessary to perform two consecutive read accesses and verify that the two returned values are the same. It should be noted that for reliable LPTIM_CNT register read accesses, two consecutive reads must perform the access and be compared.	0

Chapter 17 Universal Synchronous Asynchronous Receiver Transmitter (USART)

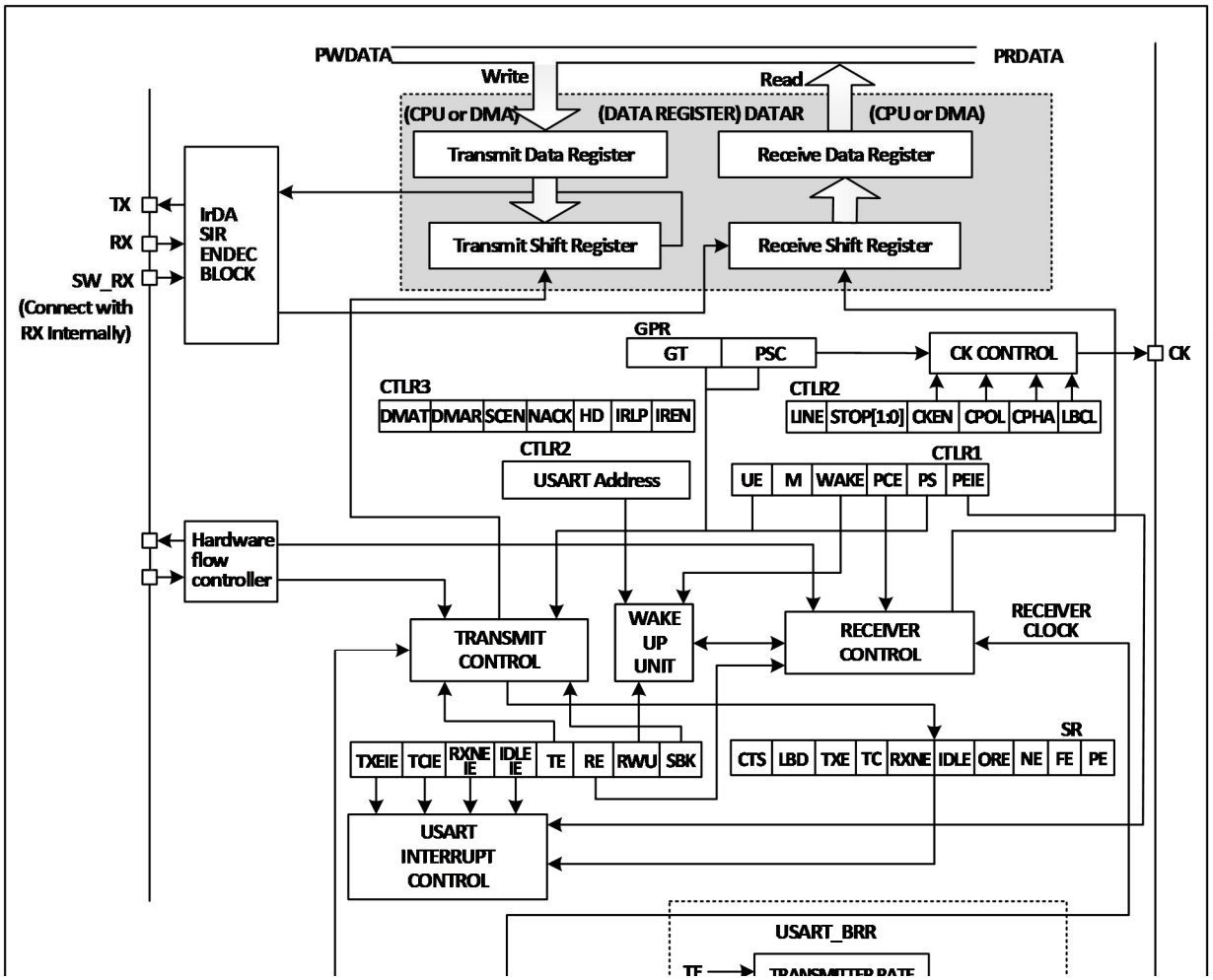
The module contains 4 universal synchronous asynchronous transceivers (USART1/2/3/4).

17.1 Main Features

- Full-duplex or half-duplex synchronous or asynchronous communication
- NRZ data format
- Fractional baud rate generator, highest 6Mbps
- Programmable data length
- Configurable stop bit
- Support LIN, IrDA encoders, smart cards
- Support DMA
- Multiple interrupt sources

17.2 Overview

Figure 17-1 USART block diagram



When TE (transmission enable bit) is set, the data in the transmitter shift register will be outputted on the TX pin, and the clock will be outputted on the CK pin. During transmission, the lowest significant bit is the first to be shifted out. Each data frame starts with a low-level start bit, and then the transmitter sends 8-bit data or 9-bit data according to the setting of the M (word length) bit, and finally a configurable number of stop bits. If there is a parity check bit, the last bit of the data word is the check bit. After TE is set, an idle frame is sent. The idle frame is 10-bit or 11-bit high level, including the stop bit. The break frame is a 10-bit or 11-bit low level, followed by a stop bit.

17.3 Baud Rate Generator

The baud rate of the transceiver = F_{CLK}/(16*USARTDIV); F_{CLK} is the clock of PBx, i.e., PCLK1 or PCLK2, PCLK2

is used for the USART1 module, and PCLK1 shall be used for the rest. The value of USARTDIV is determined according to the 2 domains: DIV_M and DIV_F in USART_BRR. The specific calculation formula is:

USARTDIV = DIV_M+(DIV_F/16)

It should be noted that the baud rate generated by the baud rate generator may not always generate just the baud rate required by the user, which may be biased. In addition to taking the value as close as possible, the method to reduce the deviation can also be to increase the PBx clock. For example, when the baud rate is set to 115200bps, the value of USARTDIV will be set to 39.0625, and the baud rate of 115200bps can be obtained at the highest frequency, but if you need a baud rate of 921600bps, the calculated USARTDIV will be 4.88, but the actual closest value filled in USART_BRR can only be 4.875. The actual baud rate is 923076bps, with an error of 0.16%.

When the serial port waveform sent by the transmitter is transmitted to the receiver, there is a certain error in the baud rate between the receiver and the sender. The error mainly comes from 3 aspects: the actual baud rate of the receiver and the sender are inconsistent; the clocks of the receiver and the sender have errors; the waveform changes in the circuit. The receiver of the peripheral module has a certain tolerance for receiving. When the sum of the total deviations generated in the above 3 aspects is less than the tolerance limit of the module, the total deviation will not affect the receiving and sending. The tolerance limit of the module is affected by the use of fractional baud rate and M bit (data field word length) or not. The use of fractional baud rate and the use of 9-bit data field length will reduce the tolerance limit, but it shall not be less than 3%.

17.4 Synchronous Mode

The synchronous mode enables the system to output clock signals when the USART module is used. When the synchronous mode is enabled to send data externally, the CK pin will output clock externally at the same time.

To enable synchronous mode, set CLKEN bit in the control register2 (R16_USARTx_CTLR2), but you need to switch off the LIN mode, smart card mode, infrared mode and half-duplex mode at the same time, i.e., to ensure that the SCEN, HDSEL and IREN bits are in the reset status. These 3 bits are in the control register3 (R16_USARTx_CTLR3).

The main point of the synchronous mode is the output control of the clock. Attention shall be paid to the following:

The synchronous mode of the USART module only works in the master mode, i.e., the CK pin only outputs the clock and does not receive input;

The clock signal is outputted only when TX pin outputs data;

The LBCL bit determines whether the clock is outputted when the last data bit is sent. The CPOL bit determines the polarity of the clock, and the CPHA determines the phase position of the clock. These 3 bits are in the control register 2 (R16_USARTx_CTLR2). These 3 bits need to be set when TE and RE are not enabled. The specific difference is shown in Figure 17-2.

In the synchronous mode, the receiver will only sample when outputting the clock, and the slave needs to maintain a certain signal setup time and hold time, specifically as shown in Figure 17-3.

Figure 17-2 Example of USART clock timing (M=0)

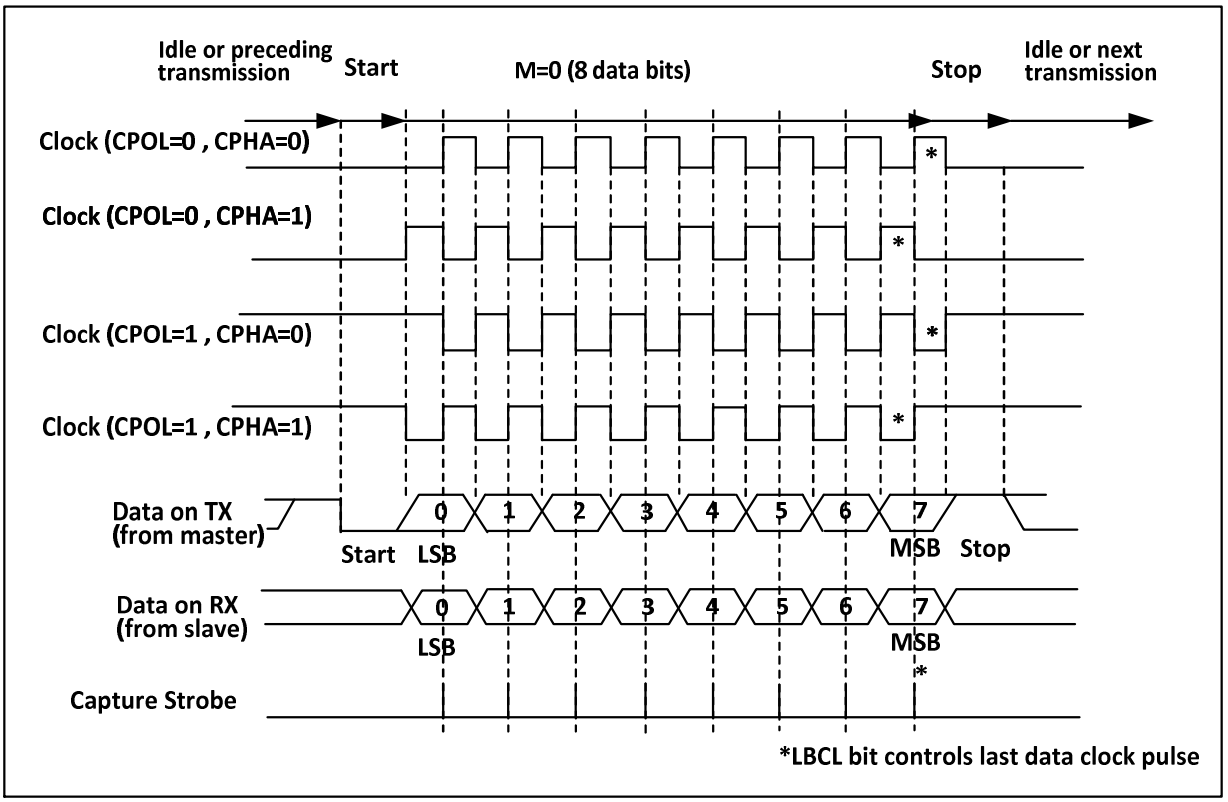
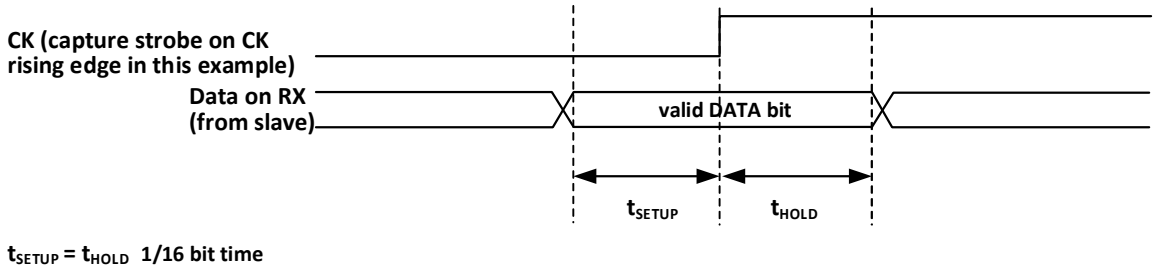


Figure 17-3 Data sample hold time



17.5 1-wire Half-duplex Mode

The half-duplex mode supports the use of a single pin (only TX pin) to receive and transmit, and the TX pin and RX pin are connected inside the chip.

To enable half-duplex mode, set HDSEL bit in the control register 3 (R16_USARTx_CTLR3), but you need to disable LIN mode, smartcard mode, infrared mode and synchronous mode at the same time, i.e., to ensure that the SCEN, CLKEN and IREN bits are in the reset status. These 3 bits are in the control register 2 and the control register 3 (R16_USARTx_CTLR2 and R16_USARTx_CTLR3).

After setting to half-duplex mode, it is needed to set the TX IO port to open-drain output high mode. When TE bit is set, the data will be sent out as long as the data is written to the data register. Special attention shall be paid to the fact that bus conflicts may occur when multiple devices use a single bus to transmit and receive in half-duplex

mode. This requires users to avoid it by software.

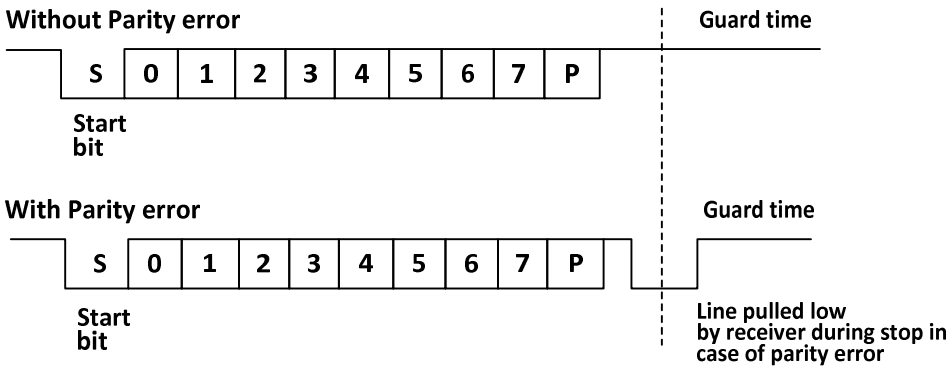
17.6 Smart Card

The smartcard mode supports ISO7816-3 protocol to access the smart card controller.

To enable smartcard mode, set the SCEN bit in the control register 3 (R16_USARTx_CTLR3), but it is needed to disable LIN mode, half-duplex mode and infrared mode at the same time, i.e., to ensure that the LINEN, HDSEL and IREN bits are in the reset status, but CLKEN can be switched on to output the clock. These 3 bits are in the control register 2 and the control register 3 (R16_USARTx_CTLR2 and R16_USARTx_CTLR3).

In order to support smartcard mode, USART shall be set to 8 data bits plus 1 check bit. It is recommended that the stop bit be configured to 1.5 bits for both sending and receiving. The smart card mode is a 1-wire half-duplex protocol, which uses TX line as the data communication and shall be configured as open drain output plus pull-up. When the receiver receives a frame of data and detects a parity check error, it will send a NACK signal at the stop bit, i.e., actively reducing a cycle of TX during the stop bit. After the sender detects the NACK signal, a frame error will be generated, and the application can resend accordingly. Figure 17-4 shows the waveforms on the TX pin under correct conditions and in the event of parity check errors. The TC flag (transmission completion flag) of the USART can delay the generation of GT (protection time) clocks, and the receiver will not recognize the NACK signal set by itself as the start bit.

Figure 17-4 (No) parity check error



In smartcard mode, the output waveform after the CK pin is enabled has nothing to do with the communication. It only provides the clock for the smart card. Its value is the PB clock and then the 5-bit settable clock frequency division (the frequency division value is double of PSC, and the highest is frequency division 62).

17.7 IrDA

USART module supports control IrDA infrared transceiver for physical layer communication. To use IrDA, the LINEN, STOP, CLKEN, SCEN and HDSEL bits must be cleared. NRZ (non-return-to-zero) coding is used between the USART module and the SIR physical layer (infrared transceiver), and the maximum support rate is 115200bps.

IrDA is a half-duplex protocol. If UASRT sends data to the SIR physical layer, the IrDA decoder ignores the newly sent infrared signal. If the USART receives data from SIR, then SIR does not accept USART signal. The level logic sent by USART to SIR and SIR to USART is different. In SIR receive logic, ‘1’ represents high level and ‘0’

represents low level. However, in the SIR transmit logic, '0' represents high level and '1' represents low level.

17.8 DMA

The USART module supports DMA, and can use DMA to implement fast continuous reception and transmission. When DMA is enabled and the TXE bit is set, DMA writes data to the transmit buffer from the set memory space. When DMA is used for reception, DMA transfers the data in the receive buffer to a specific memory space each time the RXNE bit is set.

17.9 Interrupt

The USART module supports multiple interrupt sources, including transmit data register empty (TXE), CTS, transmission complete (TC), received data ready (RXNE), data overrun error (ORE), idle line (IDLE), parity check error (PE), break flag (LBD), noise (NE), multi-buffer communication overrun (ORT) and framing error (FE).

Table 17-1 Interrupts and the corresponding enable bits

Interrupt source	Enable bit
Transmit data register empty (TXE)	TXEIE
Transmission allowed (CTS)	CTSIE
Transmission complete (TC)	TCIE
Received data ready (RXNE)	RXNEIE
Transmit data register empty (TXE)	
Idle line (IDLE)	IDLEIE
Parity error (PE)	PEIE
Break flag (LBD)	LBDIE
Noise (NE)	EIE
Overrun error in multi-buffer communication (ORT)	
Framing error (FE) in multi-buffer communication	

17.10 Register Description

Table 17-2 USART1-related registers

Name	Access address	Description	Reset value
R32_USART1_STATR	0x40013800	UASRT1 status register	0x000000C0
R32_USART1_DATAR	0x40013804	UASRT1 data register	0x000000XX
R32_USART1_BRR	0x40013808	UASRT1 baud rate register	0x00000000
R32_USART1_CTLR1	0x4001380C	UASRT1 control register1	0x00000000
R32_USART1_CTLR2	0x40013810	UASRT1 control register2	0x00000000
R32_USART1_CTLR3	0x40013814	UASRT1 control register3	0x00000000
R32_USART1_GPR	0x40013818	UASRT1 guard time and prescaler register	0x00000000

Table 17-3 USART2-related registers

Name	Access address	Description	Reset value
R32_USART2_STATR	0x40004400	UASRT2 status register	0x000000C0
R32_USART2_DATAR	0x40004404	UASRT2 data register	0x000000XX
R32_USART2_BRR	0x40004408	UASRT2 baud rate register	0x00000000
R32_USART2_CTLR1	0x4000440C	UASRT2 control register1	0x00000000
R32_USART2_CTLR2	0x40004410	UASRT2 control register2	0x00000000
R32_USART2_CTLR3	0x40004414	UASRT2 control register3	0x00000000
R32_USART2_GPR	0x40004418	UASRT2 guard time and prescaler register	0x00000000

Table 17-4 USART3-related registers

Name	Access address	Description	Reset value
R32_USART3_STATR	0x40004800	UASRT3 status register	0x000000C0
R32_USART3_DATAR	0x40004804	UASRT3 data register	0x000000XX
R32_USART3_BRR	0x40004808	UASRT3 baud rate register	0x00000000
R32_USART3_CTLR1	0x4000480C	UASRT3 control register1	0x00000000
R32_USART3_CTLR2	0x40004810	UASRT3 control register2	0x00000000
R32_USART3_CTLR3	0x40004814	UASRT3 control register3	0x00000000
R32_USART3_GPR	0x40004818	UASRT3 guard time and prescaler register	0x00000000

Table 17-5 USART4-related registers

Name	Access address	Description	Reset value
R32_USART4_STATR	0x40004C00	UASRT4 status register	0x000000C0
R32_USART4_DATAR	0x40004C04	UASRT4 data register	0x000000XX
R32_USART4_BRR	0x40004C08	UASRT4 baud rate register	0x00000000
R32_USART4_CTLR1	0x40004C0C	UASRT4 control register1	0x00000000
R32_USART4_CTLR2	0x40004C10	UASRT4 control register2	0x00000000
R32_USART4_CTLR3	0x40004C14	UASRT4 control register3	0x00000000
R32_USART4_GPR	0x40004C18	UASRT4 guard time and prescaler register	0x00000000

17.10.1 USART Status Register (USARTx_STATR) (x=1/2/3/4)

Offset address: 0x00

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CTS	LBD	TXE	TC	RXNE	IDLE	ORE	NE	FE	PE

Bit	Name	Access	Description	Reset value
[31:10]	Reserved	RO	Reserved	0
9	CTS	RW0	CTS status change flag. If the CTSE bit is set, when	0

			the nCTS output state changes, the bit will be set high by the hardware. Zero is cleared by the software. If the CTSIE bit has been set, an interrupt occurs. 1: There is a change in the nCTS state line. 0: There is no change on the nCTS state line.	
8	LBD	RW0	LIN Break detection flag. When LIN Break is detected, this bit will be set by hardware. It is cleared by the software. If the LBDIE bit has been set, an interrupt will be generated. 1: LIN Break detected; 0: No LIN Break detected.	0
7	TXE	RO	Transmission data register empty flag. When data in TDR register is transferred to shift register by hardware, this bit will be set by hardware. If TXEIE bit has been set, an interrupt will be generated, the data register will be written and this bit will be reset. 1: Data is transferred to the shift register; 0: Data is not transferred to the shift register.	1
6	TC	RW0	Transmission complete flag. When a frame containing data is sent and TXE bit is set, the hardware will set this bit. If TCIE is set, a corresponding interrupt will be generated. The software will read this bit and then write the data register to clear this bit. You can also directly write 0 to clear this bit. 1: Transmission completed; 0: Transmission not completed.	1
5	RXNE	RW0	Read data register not empty flag. When the data in the shift register is transferred to the data register, this bit will be set by the hardware. If the RXNEIE bit has been set, the corresponding interrupt will be generated. This bit can be cleared by the write operation of the data register. This bit can be also cleared by directly writing 0. 1: The data is received and can be read; 0: The data is not received.	0
4	IDLE	RO	Idle line flag. When an idle line is detected, the bit will be set by hardware. If IDLEIE bit has been set, the corresponding interrupt will be generated. This bit can be cleared by reading the status register and then reading the data register. 1: The bus is idle now; 0: Idle bus is not detected. *Note: This bit will not be set again until RXNE is set.*	0

3	ORE	RO	Overrun error flag. When the receiving shift register has data that needs to be transferred to the data register, but this bit will be set when there is still data that has not been read in the receiving field of the data register. If the RXNEIE bit is set, the corresponding interrupt will be generated. 1: The overrun error has occurred; 0: No overrun error has occurred. *Note: When an overrun error occurs, the value of the data register will not be lost, but the value of the shift register will be overwritten. If the EIE bit is set, the ORE flag bit will generate an interrupt in the multi-buffer communication mode.*	0
2	NE	RO	Noise error flag. When the noise error flag is detected, it will be set by hardware. This bit can be reset by reading the status register and then reading the data register. 1: The noise is detected; 0: No noise is detected. *Note: This bit will not generate the interrupt. If the EIE bit has been set, the FE flag bit will generate an interrupt in the multi-buffer communication mode.*	0
1	FE	RO	Frame error flag. When a synchronization error, excessive noise or disconnection is detected, this bit will be set by hardware. This bit can be reset by reading the bit and then reading the data register. 1: A frame error is detected; 0: No frame error is detected. *Note: This bit will not generate interrupt. If the EIE bit has been set, the FE flag bit will generate an interrupt in the multi-buffer communication mode.*	0
0	PE	RO	Parity error flag. In the receiving mode, if a parity error occurs, this bit can be set by hardware. This bit can be reset by reading the bit and then reading the data register. Before this bit is cleared, the software must wait for the RXNE flag bit to be set. If PEIE bit has been set before, then the corresponding interrupt will be generated when this bit is set. 1: Parity check error occurs; 0: No parity check error occurs.	0

17.10.2 USART Data Register (USARTx_DATAR) (x=1/2/3/4)

Offset address: 0x04

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

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

Bit	Name	Access	Description	Reset value
[31:9]	Reserved	RO	Reserved	0
[8:0]	DR[8:0]	RW	Data register. This register is actually composed of 2 registers: receive data register (RDR) and transmit data register (TDR). The start of the read and write operations of DR is to read the receive data register (RDR) and write to the transmit data register (TDR).	X

17.10.3 USART Baud Rate Register (USARTx_BRR) (x=1/2/3/4)

Offset address: 0x08

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

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:4]	DIV_Mantissa [11:0]	RW	These 12 bits define the integer portion of the divider division factor.	0
[3:0]	DIV_Fraction[3:0]	RW	These 4 bits define the decimal part of the divider division factor.	0

17.10.4 USART Control Register 1 (USARTx_CTLR1) (x=1/2/3/4)

Offset address: 0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		UE	M	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	RWU	SBK

Bit	Name	Access	Description	Reset value
[31:14]	Reserved	RO	Reserved	0
13	UE	RW	USART enable. When this bit is cleared, the frequency divider and output of USART both stop working after the current byte transmission is completed.	0
12	M	RW	Word length. 1: 9 data bits;	0

			0: 8 data bits.	
11	WAKE	RW	Wake-up. This bit decides the method to wake up USART: 1: Address flag; 0: Idle line.	0
10	PCE	RW	Parity control enable. For the receiver, the parity of the data is performed; for the transmitter, the check bit is inserted. Once this bit is set, the parity control enable takes effect only after the current byte transmission is completed.	0
9	PS	RW	Parity selection. 0 means even parity, and 1 means odd parity. After this bit is set, the parity control enable takes effect only after the current byte transmission is completed.	0
8	PEIE	RW	Parity check interrupt enable. When this bit is set, the parity check error interrupt is allowed to be generated.	0
7	TXEIE	RW	Transmit buffer empty interrupt enable. When this bit is set, the transmit buffer empty interrupt is allowed to be generated.	0
6	TCIE	RW	Transmission completion interrupt enable. When this bit is set, the transmission complete interrupt is allowed to be generated.	0
5	RXNEIE	RW	Receive buffer non-empty interrupt enable. When this bit is set, the receive buffer not empty interrupt is allowed to be generated.	0
4	IDLEIE	RW	Idle line interrupt enable. When this bit is set, the idle line interrupt is allowed to be generated.	0
3	TE	RW	Transmitter enable. When this bit is set, the transmitter is enabled.	0
2	RE	RW	Receiver enable. When this bit is set, the receiver is enabled, and the receiver starts detecting the start bit on the RX pin.	0
1	RWU	RW	Receiver wake-up. This bit decides whether the USART is in mute mode: 1: The receiver is in mute mode; 0: The receiver is in active mode. *Note 1: Before the RWU bit is set, USART needs to receive a data byte firstly. Otherwise, it cannot be woken up by the idle bus in mute mode;* *Note 2: When configured to wake up from address flag, the RWU bit cannot be modified by software when RXNE is set.*	0
0	SBK	RW	Send break character control. This bit is set to transmit a frame break character. For the stop bit of break frame,	0

			the bit is set by hardware. 1: Break character transmitted; 0: No break character transmitted.	

17.10.5 USART Control Register 2 (USARTx_CTLR2) (x=1/2/3/4)

Offset address: 0x10

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	LINEN	STOP		CLKEN	CPOL	CPHA	LBCL	Reserved	LBDIE	LBDL	Reserved	ADD[3:0]			

Bit	Name	Access	Description	Reset value
[31:15]	Reserved	RO	Reserved	0
14	LINEN	RW	LIN mode enable. When this bit is set, the LIN mode is enabled. In LIN mode, you can use the SBK bit to send the LIN synchronization disconnection symbol and detect the LIN synchronization disconnection symbol.	0
[13:12]	STOP	RW	Stop bit setting. These bits are used to set the stop bits. 00: 1 stop bit; 01: 0.5 stop bit; 10: 2 stop bits; 11: 1.5 stop bit.	00b
11	CLKEN	RW	Clock enable. This bit is used to enable CK pin. 1: Enable; 0: Disable.	0
10	CPOL	RW	Clock polarity. In synchronous mode, this bit can be used to select the polarity of the clock output on the SLCK pin, and work with CPHA to generate the required clock/data sampling relationship. 1: High level is maintained on the CK pin when the bus is idle; 0: Low level is maintained on the CK pin when the bus is idle. *Note: This bit cannot be modified after enabling transmission.*	0
9	CPHA	RW	Clock phase position setting. In the synchronization mode, you can use this bit to select the phase position of the clock output on the SLCK pin, and work with CPOL bit to generate the required clock/data sampling relationship. 1: Data capture is performed on the second edge of the clock; 0: Data capture is performed on the first edge of the clock. *Note: This bit cannot be modified after enabling*	0

			transmission.	
8	LBCL	RW	Last bit clock pulse control. In synchronous mode, it is used to control whether to output the clock pulse corresponding to the last data byte sent on the CK pin; 1: The clock pulse of the last bit of data is not output from CK; 0: The clock pulse of the last bit of data is output from CK. *Note: This bit cannot be modified after enabling transmission.*	0
7	Reserved	RW	Reserved.	0
6	LBDIE	RW	LIN Break detection interrupt enable. This bit can enable the interrupt caused by LBD;	0
5	LBDL	RW	LIN Break detection length, used to select 11-bit or 10-bit break character detection. 1: 11-bit break detection; 0: 10-bit break detection.	0
4	Reserved	RW	Reserved.	0
[3:0]	ADD[3:0]	RW	Address of the USART node, used to set the USART node address of the device. When the data is used during mute mode in multi-processor communication, the address flag is used to wake up a certain USART device.	0

17.10.6 USART Control Register 3 (USARTx_CTLR3) (x=1/2/3/4)

Offset address: 0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSEL	IRLP	IREN	EIE

Bit	Name	Access	Description	Reset value
[31:11]	Reserved	RO	Reserved	0
10	CTSIE	RW	CTSIE interrupt enable. When this bit is set, an interrupt is generated when CTS is set.	0
9	CTSE	RW	CTS enable. When this bit is set, the CTS flow control is enabled.	0
8	RTSE	RW	RTS enable. When this bit is set, the RTS flow control is enabled.	0
7	DMAT	RW	DMA transmission enable. When this bit is set, DMA mode is enabled for transmission.	0
6	DMAR	RW	DMA reception enable. When this bit is set, DMA mode is enabled for reception.	0

5	SCEN	RW	Smart card mode enable. When this bit is set, smartcard mode is enabled.	0
4	NACK	RW	Smart card NACK enable. When this bit is set, NACK is transmitted when the check error occurs.	0
3	HDSEL	RW	Half-duplex mode selection. When this bit is set, half-duplex mode is selected.	0
2	IRLP	RW	Infrared low power selection. When this bit is set, low power mode is selected.	0
1	IREN	RW	Infrared enable. When this bit is set, infrared mode is enabled.	0
0	EIE	RW	Error enable interrupt. When this bit is set, and when DMAR is set, an interrupt is generated if the FE or ORE or NE bit is set.	0

17.10.7 USART Guard Time and Prescaler Register (USARTx_GPR) (x=1/2/3/4)

Offset address: 0x18

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

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:8]	GT[7:0]	RW	Guard time domain. These bits specify the guard time in unit of baud rate clock. In smartcard mode, the transmission complete flag is set after the guard time has passed.	0
[7:0]	PSC[7:0]	RW	Prescaler domain. In infrared low-power mode, the source clock is divided by this value (all 8 bits are valid), and a value of 0 means reservation; In infrared normal mode, these bits can only be set to 1; In smartcard mode, the value (the lower 5 bits are valid) is multiplied by 2 to give the division factor of the source clock frequency, to provide the clock to the smart card. A value of 0 means reservation.	0

Chapter 18 Inter-integrated Circuit (I2C) Interface

The internal integrated circuit bus (I2C) is widely used in the communication between microcontrollers, sensors and other off-chip modules. It supports multi-master and multi-slave mode, and can communicate at both 100KHz (standard) and 400KHz (fast) speeds using only two wires (SDA and SCL). I2C bus is also compatible with SMBus protocol. It not only supports I2C timing, but also supports arbitration, timing and DMA, and has CRC check function.

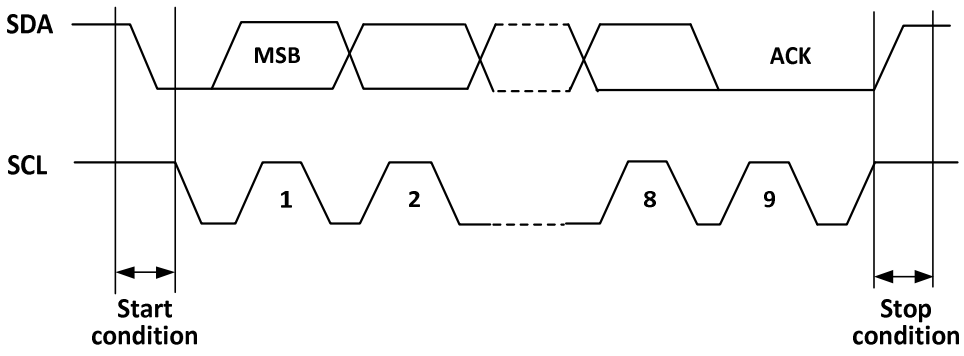
18.1 Main Features

- Master mode and slave mode
- 7-bit or 10-bit address
- Slave device supports dual 7-bit address.
- Two speed modes: 100KHz and 400KHz
- Multiple status modes, multiple error flags
- Optional clock stretching
- 2 interrupt vectors
- DMA capability
- Support PEC
- Compatible with SMBus

18.2 Overview

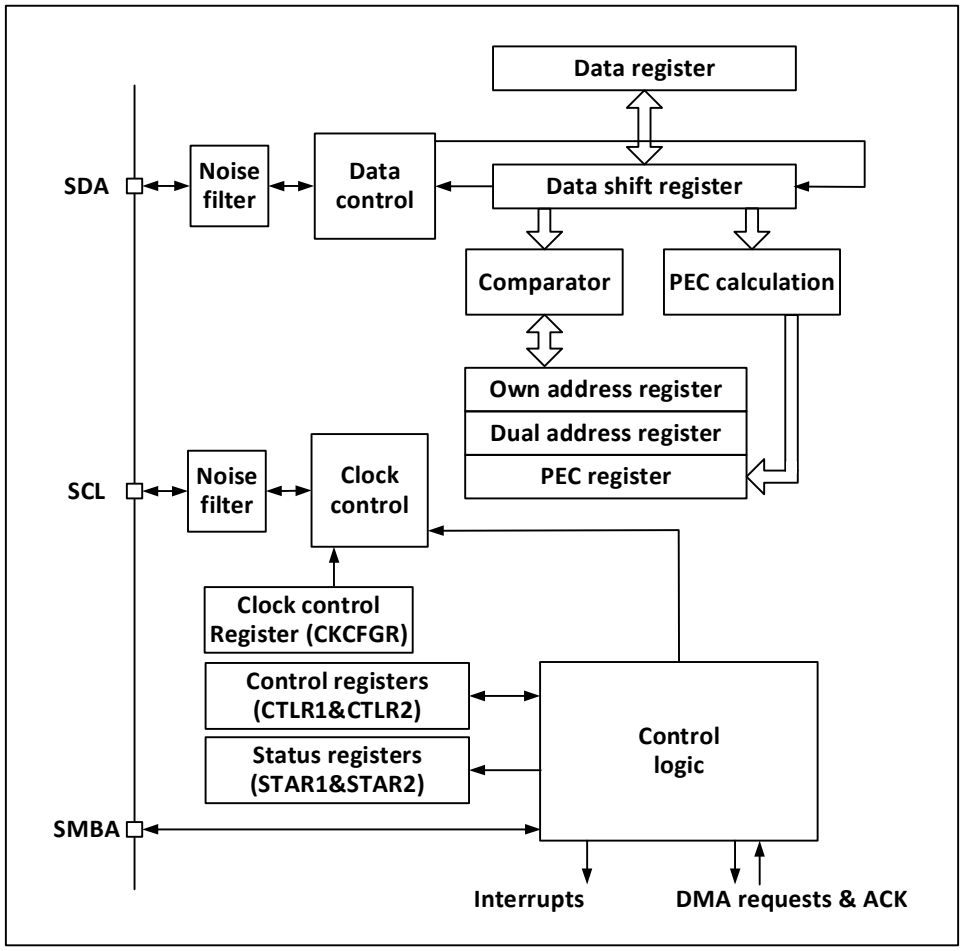
I2C is a half-duplex bus, which can only run in one of the following four modes: master device sending mode, master device receiving mode, slave device sending mode and slave device receiving mode. The I2C module works in slave mode by default. After generating the starting condition, it will automatically switch to the master mode. When the arbitration is lost or a stop signal is generated, it will switch to the slave mode. The I2C module supports multi-host functions. When working in main mode, the I2C module will actively send out data and addresses. Data and addresses are transmitted in 8-bit units, with the high bit in front and the low bit in the back. After the start event is a byte (7-bit address mode) or two-byte (10-bit address mode) address. Every time the host sends 8-bit data or address, the slave needs to reply a reply ACK, that is, pull down the SDA bus, as shown in figure 18-1.

Figure 18-1 I2C timing diagram



For normal use, the correct clock must be input to I2C. In the standard mode, the minimum input clock is 2MHz, while the minimum input clock is 4MHz in the fast mode.
Figure 18-2 shows the block diagram of I2C.

Figure 18-2 I2C block diagram



18.3 Master Mode

In master mode, the I2C module leads the data transmission and outputs the clock signal. The data transfer starts with a Start event and ends with a Stop event. The following is the required operations in master mode:

Set the correct clock in the control register2 (R16_I2Cx_CTLR2) and the clock control register (R16_I2Cx_CKCFGR).

Set a proper rising edge in the rising edge register (R16_I2Cx_RTR).

Set the PE bit in R16_I2Cx_CTLR1 to start the peripheral.

Set the START bit in the control register (R16_I2Cx_CTLR1) to generate a start event.

After the START bit is set, the I2C module automatically switches to master mode, the MSL bit is set, and a Start event is generated. After the start event is generated, the SB bit is set. If the ITEVTEN bit (in R16_I2Cx_CTLR2) is set, an interrupt is generated. In this case, it is needed to read the R16_I2Cx_STAR1 register. After the slave address is written to the data register, the SB bit is automatically cleared.

If the 10-bit address mode is enabled, then write the data register to send the header sequence (the header sequence is 11110xx0b, of which the xx bits are the highest 2 bits of the 10-bit address).

After the header sequence is transmitted, the ADD10 bit in the status register is set. If the ITEVTEN bit is already set, an interrupt will be generated. At this time, read the R16_I2Cx_STAR1 register and write the second address byte to the data register. Then, clear the ADD10 bit.

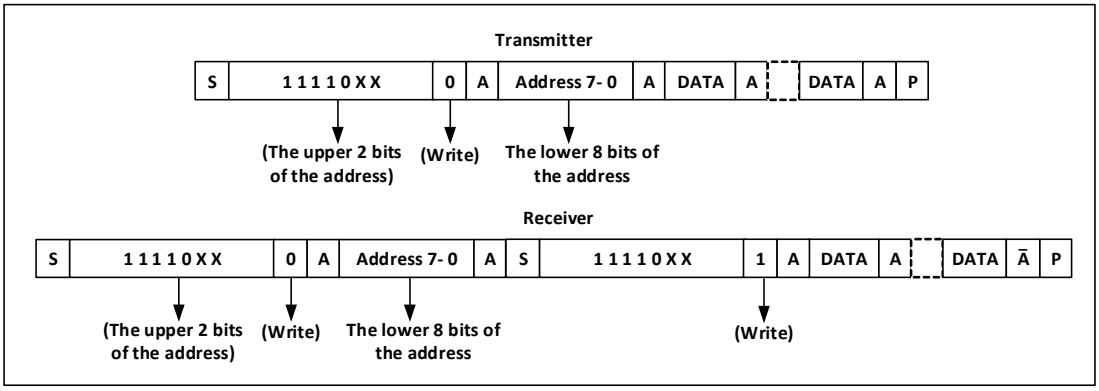
Then, write the data register to send the second address byte. After sending the second address byte, the ADDR bit in the status register is set. If the ITEVTEN bit has been set, an interrupt will be generated. Read the R16_I2Cx_STAR2 register at this time and then read R16_I2Cx_STAR1 register again to clear the ADDR bit;

If the 7-bit address mode is enabled, the write data register transmit the address byte. After the address byte is sent, the ADDR bit in the status register is set. If the ITEVTEN bit has been set, an interrupt will be generated. Read the R16_I2Cx_STAR1 register and then read the R16_I2Cx_STAR2 register again to clear the ADDR bit.

In the 7-bit address mode, the first byte sent is the address byte, the first 7 bits represent the address of the target slave device, the 8th bit determines the direction of the subsequent message, and '0' means the master device writes data to the slave device, '1' means that the master device reads information from the slave device.

In the 10-bit address mode, as shown in Figure 18-3, the first byte is 11110xx0, xx are the highest 2 bits of the 10-bit address, and the second byte is the lower 8 bits of the 10-bit address in the address transmission phase. If you subsequently enter the master transmitter mode, send data continuously. If the device enters the master receiver mode subsequently, a start event needs to be re-sent, and a byte of 11110xx1 will be sent together. Then, enter the master receiver mode.

Figure 18-3 Master receive/transmits data in 10-bit address mode



When transmitting mode, the shift register inside the main device transmits data from the data register to the SDA line. When the main device receives the ACK, the TxE of the status register 1 (R16_I2Cx_STAR1) is set, and an interrupt occurs if the ITEVTEN and ITBUFEN are set. Writing data to the data register clears the TxE bit. If the TxE bit is set and no new data is written to the data register before the last data is sent, then the BTF bit will be set, the SCL will remain low until it is cleared, and after reading the R16_I2Cx_STAR1, writing data to the data register will clear the BTF bit.

In the receiving mode, the I2C module receives data from the SDA line and writes it into the data register through the shift register. After each byte, if the ACK bit is set, the I2C module will issue a low level of reply, while the RxNe bit will be set, and if ITEVTEN and ITBUFEN are set, there will be an interruption. If the RxNE is set and the original data is not read before the new data is received, the BTF bit will be set, the SCL will remain low before clearing the BTF, and reading the R16_I2Cx_STAR1 and then reading the data register will clear the BTF bit.

When the master device ends sending data, it will actively send an end event, that is, setting the STOP bit. In the receive mode, the master device needs to NAK at the answer location of the last data bit. Note that after the NAK is generated, the I2C module will switch to slave mode.

18.4 Slave Mode

From the mode, the I2C module can recognize its own address and broadcast call address. The software can control the identification of broadcast call addresses on or off. Once the start event is detected, the I2C module compares the SDA data through the shift register with its own address (the number of bits depends on ENDUAL and ADDMODE) or the broadcast address (when ENGC is set). If there is a mismatch, it will be ignored until a new start event is generated. If it matches the header sequence, an ACK signal is generated and waits for the address of the second byte; if the address of the second byte also matches, or if the address of the whole segment matches in the case of a 7-bit address, then:

First an ACK reply is generated; the ADDR bit is set, and if the ITEVTEN bit is set, there will be a corresponding interrupt.

If you are using dual-address mode (the ENDUAL bit is set), you also need to read the DUALF bit to determine which address the host is calling.

The slave mode defaults to receive mode, and when the last bit of the received header sequence is 1, or the last bit of the 7-bit address is 1 (depending on whether the header sequence is received for the first time or a normal 7-bit address), the I2C module will enter sender mode, and the TRA bit will indicate whether it is currently in receiver or sender mode.

When sending mode, after clearing the ADDR bit, the I2C module sends bytes from the data register to the SDA line through the shift register. Upon receipt of a reply ACK, the TXE bit is set, and an interrupt occurs if ITEVTEN and ITBUFEN are set. If TxE is set but no new data is written to the data register before the end of the next data transmission, the BTF bit will be set. The SCL will remain low until the BTF is cleared, and after reading the status register 1 (R16_I2Cx_STAR1), writing data to the data register will clear the BTF bit.

In the receiving mode, after the ADDR is cleared, the I2C module stores the data on the SDA into the data register through the shift register. After each byte is received, the I2C module will set an ACK bit and parallel the RxNE bit. If ITEVTEN and ITBUFEN are set, an interrupt will be generated. If the RxNE is set and the old data is not read out before the new data is received, then the BTF will be set. The SCL stays low until the BTF bit is cleared. Reading the status register 1 (R16_I2Cx_STAR1) and reading the data in the data register clears the BTF bit.

When the I2C module detects a stop event, the STOPF bit is set, and if the ITEVFEN bit is set, an interrupt occurs. The user needs to read the status register (R16_I2Cx_STAR1) and then write the control register (such as the reset control word SWRST) to clear.

18.5 Error

18.5.1 Bus Error (BERR)

When the I2C module detects an external start or stop event during address or data transmission, a bus error occurs. When a bus error is generated, the BERR bit is set, and an interrupt occurs if ITERREN is set. In slave mode, the data is discarded and the hardware releases the bus. If it is a start signal, the hardware will consider it as a restart

signal and start waiting for the address or stop signal; if it is a stop signal, it will operate according to the normal stop condition in advance. In main mode, the hardware does not release the bus and does not affect the current transmission, and it is up to the user code to decide whether to abort the transmission.

18.5.2 Acknowledge Failure (AF)

When the I2C module detects no response after a byte, a reply error occurs. When a reply error is generated: AF will be set, and an interrupt will be generated if ITERREN is set; if an AF error is encountered, if the I2C module works in slave mode, the hardware must release the bus, and if it is in master mode, the software must generate a stop event.

18.5.3 Arbitration Lost (ARLO)

When the I2C module detects the arbitration loss, an arbitration loss error occurs. When an arbitration loss error occurs: the ARLO bit is set and an interrupt is generated if ITERREN is set; the I2C module switches to slave mode and no longer responds to slave-initiated transfers against it unless a host initiates a new start event; the hardware releases the bus.

18.5.4 Overrun/Underrun Error (OVR)

- Overload error:

In slave mode, if clock extension is prohibited, the I2C module is receiving data. If one byte of data has been received, but the last received data has not been read out, an overload error will occur. When an overload error occurs, the last received byte is discarded and the sender should resend the last sent byte.

- Underload error:

In slave mode, if clock extension is prohibited, the I2C module is sending data, and if new data is not written to the data register before the next byte clock arrives, an underload error will occur. When an underload error occurs, the data in the previous data register will be sent twice. If an underload error occurs, the receiver should discard the repeatedly received data. In order not to generate underload errors, the I2C module should write data to the data register before the first rising edge of the next byte.

18.6 Clock Extension

If clock extension is prohibited, there is a possibility of overload / underload errors. But if the clock is extended when enabled:

- In transmit mode, if TxE is set and BTF is set, SCL will always be low, waiting for the user to read the status register and write the data to be sent to the data register.
- In receive mode, if RxNE is set and BTF is set, the SCL will remain low after receiving the data until the user reads the status register and reads the data register.

Thus it can be seen that the extension of the enable clock can avoid overload / underload errors.

18.7 SMBus

SMBus is also a 2-wire interface, which is generally used between the system and power management. SMBus and I2C have many similarities. For example, SMBus uses the same 7-bit address mode as I2C.

Similarities between SMBus and I2C:

1) Master-slave communication mode; the host provides the clock and supports multiple masters and multiple slaves;
2) 2-wire communication structure, of which a warning line can be selected for SMBus;
3) Support 7-bit address format.

Differences between SMBus and I2C:

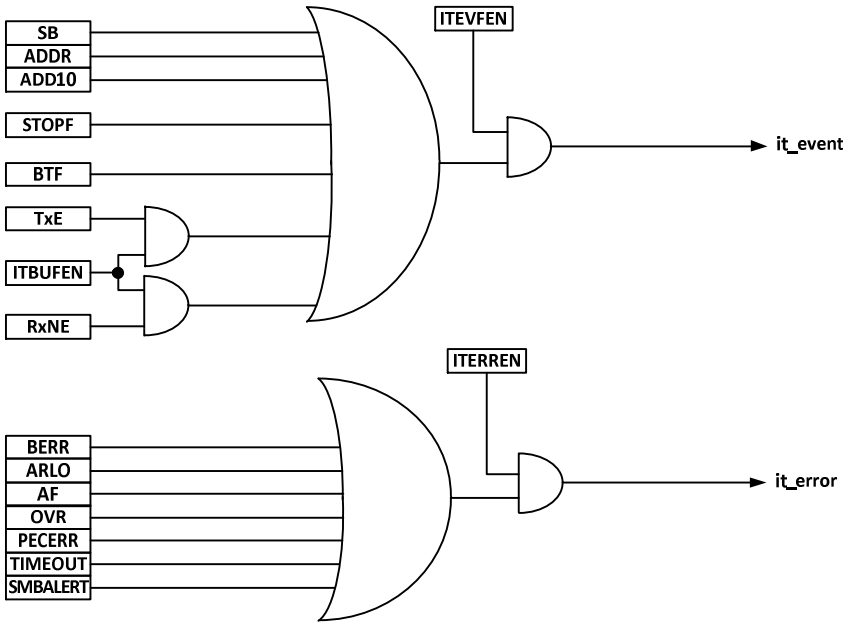
1) I2C supports the maximum speed of 400 KHz, while SMBus supports the maximum speed of 100 KHz, and SMBus has the minimum speed limit of 10 KHz;
2) When the SMBus clock is lower than 35mS, it will report a timeout, but I2C has no such limitation;
3) SMBus has a fixed logic level, but I2C does not, depending on VDD;
4) SMBus has a bus protocol, but I2C does not.

SMBus also includes device identification, address resolution protocol, unique device identifier, SMBus reminder, and various bus protocols. For details, please refer to SMBus specification version 2.0. When SMBus is used, only the SMBus bit of the control register needs to be set, and the SMBTYPE and ENAARP bits need to be configured as needed.

18.8 Interrupt

Each I2C module has 2 interrupt vectors: event interrupt and error interrupt. 2 types of interrupts support the interrupt sources as shown in Figure 18-4.

Figure 18-4 I2C interrupt request



18.9 DMA

DMA can be used to receive/transmit bulk data. When DMA is used, the ITBUFEN bit of the control register cannot be set.

- DMA is used for transmission

The DMA mode can be activated by setting the DMAEN bit in the CTLR2 register. As long as the TxE bit is set, the data can be loaded into the I2C data register from the set memory by DMA. The following settings are required to allocate channels for I2C.

1) Set the I2Cx_DATAR register address to the DMA_PADDRx register, and set the memory address in the DMA_MADDRx register, so that the data will be sent from the memory to the I2Cx_DATAR register after each TxE event.
2) Set the required number of transferred bytes in the DMA_CNTRx register. After each TxE event, this value will be reduced progressively.
3) The channel priority is configured using the PL[0:1] bits in the DMA_CFGRx register.
4) Set the DIR bit in the DMA_CFGRx register, and it can be configured to issue an interrupt request according to application requirements when the entire transmission is half or wholly completed.
5) Activate the channel by setting the EN bit in the DMA_CFGRx register.

When the number of data transfer bytes set in the DMA controller has been completed, the DMA controller will send an EOT/EOT_1 signal indicating the end of the transmission to the I2C interface. When the interrupt is allowed, a DMA interrupt will be generated.

- DMA is used for reception

After the DMAEN bit in the CTLR2 register is set, DMA receiver mode can be started. When DMA is used for reception, DMA transfers the data in the data register to the preset memory area. The following steps are required to allocate channels for I2C.

1) Set the I2Cx_DATAR register address to the DMA_PADDRx register, and set the memory address in the DMA_MADDRx register, so that the data will be written into the memory from the I2Cx_DATAR register after each RxNE event.
2) Set the required number of transferred bytes in the DMA_CNTRx register. After each RxNE event, this value will be reduced progressively.
3) The channel priority is configured by the PL[0:1] bits in the DMA_CFGRx register.
4) Clear the DIR bit in the DMA_CFGRx register, and it can be configured to issue an interrupt request according to application requirements when the data transmission is half or wholly completed.
5) Activate the channel by setting the EN bit in the DMA_CFGRx register.

When the number of data transfer bytes set in the DMA controller has been completed, the DMA controller will send an EOT/EOT_1 signal indicating the end of the transmission to the I2C interface. When the interrupt is allowed, a DMA interrupt will be generated.

18.10 Packet Error Checking

Packet error checking (PEC) is a CRC8 check step added to provide the transmission reliability. Each bit of serial data can be calculated through the following polynomial:

$$C=X^8+X^2+X+1$$

PEC calculation is activated by the ENPEC bit in the control register, and all information bytes are calculated, including address and read/write bits. During transmission, enabling PEC will add a byte of CRC8 calculation result after the last byte of data. While in receiver mode, the last byte is considered to be the CRC8 check result. If it does not match the internal calculation result, it will reply with a NAK. For the master receiver, it will reply with a NAK regardless of whether the check result is correct or not.

18.11 Debug Mode

After the system enters the debug mode, the DBG_I2Cx_SMBUS_TIMEOUT bit in the DEBUG module can be used to determine whether to continue operating or stop the time-out control of I2CSMBus.

18.12 Register Description

Table 18-1 I2C1 registers

Name	Access address	Description	Reset value
R16_I2C1_CTLR1	0x40005400	I2C1 control register 1	0x0000
R16_I2C1_CTLR2	0x40005404	I2C1 control register 2	0x0000
R16_I2C1_OADDR1	0x40005408	I2C1 address register 1	0x0000
R16_I2C1_OADDR2	0x4000540C	I2C1 address register 2	0x0000
R16_I2C1_DATAR	0x40005410	I2C1 data register	0x0000
R16_I2C1_STAR1	0x40005414	I2C1 status register 1	0x0000
R16_I2C1_STAR2	0x40005418	I2C1 status register 2	0x0000
R16_I2C1_CKCFGR	0x4000541C	I2C1 clock register	0x0000
R16_I2C1_RTR	0x40005420	I2C1 rise time register	0x0002

Table 18-2 I2C2 registers

Name	Access address	Description	Reset value
R16_I2C2_CTLR1	0x40005800	I2C2 control register 1	0x0000
R16_I2C2_CTLR2	0x40005804	I2C2 control register 2	0x0000
R16_I2C2_OADDR1	0x40005808	I2C2 address register 1	0x0000
R16_I2C2_OADDR2	0x4000580C	I2C2 address register 2	0x0000
R16_I2C2_DATAR	0x40005810	I2C2 data register	0x0000
R16_I2C2_STAR1	0x40005814	I2C2 status register 1	0x0000
R16_I2C2_STAR2	0x40005818	I2C2 status register 2	0x0000
R16_I2C2_CKCFGR	0x4000581C	I2C2 clock register	0x0000
R16_I2C2_RTR	0x40005820	I2C2 rise time register	0x0002

18.12.1 I2C Control Register (I2Cx_CTLR1) (x=1/2)

Offset address: 0x00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWRST	Reserved	ALERT	PEC	POS	ACK	STOP	START	NOSTRETCH	ENGC	ENPEC	ENARP	SMBTYPE	Reserved	SMBUS	PE

Bit	Name	Access	Description	Reset value
15	SWRST	RW	Software reset. Setting this bit by user code will reset the I2C peripheral. Before reset, make sure that the	0

			pins of the I2C bus are released and the bus is idle. *Note: This bit can reset the I2C module when no stop condition is detected on the bus but the busy bit is 1.*	
14	Reserved	RO	Reserved.	0
13	ALERT	RW	SMBus alert. This bit can be set or cleared by the user code. When PE is set, this bit can be cleared by hardware. 1: Drive the SMBusALERT pin to make it low, and the response address header shall closely follow the ACK signal; 0: Release the SMBusALERT pin to make it high, and the response address header shall closely follow the NACK signal.	0
12	PEC	RW	Packet error check enable. Set this bit to enable data packet error detection. This bit can be set or cleared by the user code. When the PEC is transmitted, or a start or end signal is generated, or the PE bit is cleared to 0, the bit can be cleared by hardware. 1: Provided with PEC; 0: Not provided with PEC. *Note: PEC will fail when the arbitration is lost.*	0
11	POS	RW	ACK and PEC position setting. This bit can be set and cleared by user code, and it can be cleared by hardware after PE is cleared; 1: The ACK bit controls the ACK or NAK of the next byte received in the shift register. The next byte received in the PEC shift register is PEC; 0: The ACK bit controls the ACK or NAK of the byte currently being received in the shift register. The PEC bit indicates that the byte of the shift register before the current bit is PEC. *Note: The usage of POS bit in 2-byte data reception is as follows: It must be configured before receiving. For the second byte of NACK, the ACK bit must be cleared immediately after the ADDR bit is cleared; in order to detect the PEC of the second byte, the PEC bit must be set after the ADDR event occurs following the POS bit.*	0
10	ACK	RW	Acknowledge enable. This bit can be set or cleared by user code. When PE bit is set, this bit can be cleared by hardware; 1: Acknowledge returned after a byte is received; 0: No acknowledge is returned.	0
9	STOP	RW	Stop event generation. It can be set or cleared by user	0

			code, or cleared by hardware when a stop event is detected, or set by hardware when a timeout error is detected. In master mode: 1: A stop event is generated after the current byte transfer or the current start condition is issued; 0: No stop event occurs. In slave mode: 1: Release the SCL and SDA lines after the current byte transfer; 0: No stop event occurs.	
8	START	RW	Start event generation. This bit can be set or cleared by the user code. When the start condition is issued or PE is cleared, it can be cleared by hardware. In master mode: 1: A start event is generated repeatedly; 0: No start event is generated. In slave mode: 1: When the bus is idle, a start event is generated; 0: No start event is generated.	0
7	NOSTRETCH	RW	Clock stretching disable. This bit is used to disable clock stretching in slave mode when the ADDR or BTF flag bit is set until it is cleared by software. 1: Clock stretching disabled; 0: Clock stretching enabled.	0
6	ENGC	RW	General call enable. Set this bit to enable the general call, and respond to general address 00h.	0
5	ENPEC	RW	PEC enable. Set this bit to enable PEC calculation.	0
4	ENARP	RW	ARP enable. Set this bit to enable the ARP. If SMBTYPE=0, the default address of the SMBus device is used. If SMBTYPE=1, the main address of the SMBus is used.	0
3	SMBTYPE	RW	SMBus device type: 1: SMBus master device; 0: SMBus slave device.	0
2	Reserved	RO	Reserved.	0
1	SMBUS	RW	SMBus mode selection. 1: SMBus mode; 0: I2C mode.	0
0	PE	RW	I2C peripheral enable. 1: I2C module enabled; 0: I2C module disabled.	0

18.12.2 I2C Control Register 2 (I2Cx_CTLR2) (x=1/2)

Offset address: 0x04

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			LAST	DMAEN	ITBUFEN	ITEVTEN	ITERREN	Reserved		FREQ[5:0]					

Bit	Name	Access	Description	Reset value
[15:13]	Reserved	RO	Reserved	0
12	LAST	RW	Last transfer setting of DMA. 1: Next DMA EOT is the last transfer; 0: Next DMA EOT is not the last transfer. *Note: This bit is used in master receiver mode and can generate a NAK when the data is received at the last time.*	0
11	DMAEN	RW	DMA request enable. Set this bit to enable DMA request when TxE or RxEN bit is set.	0
10	ITBUFEN	RW	Buffer interrupt enable. 1: When the TxE bit is set, or when the RxEN bit is set, an event interrupt is generated; 0: When the TxE bit is set, or when the RxEN bit is set, no interrupt is generated.	0
9	ITEVTEN	RW	Event interrupt enable. Set this bit to enable event interrupt. Under the following conditions, the interrupt can be generated: SB=1 (master mode); ADDR=1(master and slave modes); ADDR10=1 (master mode); STOPF=1 (slave mode); BTF=1, but no TxE or RxEN event occurs; If ITBUFEN=1, TxE event is 1; If ITBUFEN=1, RxNE event is 1.	0
8	ITERREN	RW	Error interrupt enable. When the bit is set, the error interrupt is enabled. Under the following conditions, the interrupt can be generated: BERR=1; ARLO=1; AF=1; OVR=1; PECERR=1; TIMEOUT=1; SMBAlert=1.	0
[7:6]	Reserved	RO	Reserved.	0
[5:0]	FREQ[5:0]	RW	I2C module clock frequency domain, must input the correct clock frequency to produce the correct timing, the allowable range is between 4-60MHz. Must be set between 000100b and 111100b in MHz.	0

18.12.3 I2C Address Register 1 (I2Cx_OADDR1) (x=1/2)

Offset address: 0x08

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDMODE	Reserved							ADD[9:8]		ADD[7:1]					ADD0

Bit	Name	Access	Description	Reset value
15	ADDMODE	RW	Addressing mode. 1: 10-bit slave address (7-bit address not acknowledged); 0: 7-bit slave address (10-bit address not acknowledged).	0
[14:10]	Reserved	RO	Reserved.	0
[9:8]	ADD[9:8]	RW	Interface address, which is 9-8 bits when using 10-bit addresses and ignored when 7-bit addresses are used.	0
[7:1]	ADD[7:1]	RW	Interface address, bit 7-1.	0
0	ADD0	RW	Interface address. Bit0 when a 10-bit address is used. It is ignored when a 7-bit address is used.	0

18.12.4 I2C Address Register 2 (I2Cx_OADDR2) (x=1/2)

Offset address: 0x0C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								ADD2[7:1]							ENDUAL

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved	0
[7:1]	ADD2[7:1]	RW	Interface address. Bit7 to bit1 of address in dual-address mode.	0
0	ENDUAL	RW	Dual addressing mode enable. When this bit is set, the ADD2 can be identified.	0

18.12.5 I2C Data Register (I2Cx_DATAR) (x=1/2)

Offset address: 0x10

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

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved.	0
[7:0]	DR[7:0]	RW	Data register. This domain is used to store received data or store data to be transmitted to the bus.	0

18.12.6 I2C Status Register 1 (I2Cx_STAR1) (x=1/2)

Offset address: 0x14

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMBALERT	TIMEOUT	Reserved	PECERR	OVR	AF	ARLO	BERR	TxE	RxNE	Reserved	STOPF	ADD10	BTF	ADDR	SB

Bit	Name	Access	Description	Reset value
15	SMBALERT	RW0	SMBus alert. It can be reset by user writing 0, or reset by hardware when PE becomes low. In master mode of SMBus: 1: SMBus alert is generated on the pin; 0: No SMBus alert. In slave mode of SMBus: 1: SMBAlert response address header to SMBAlert LOW received; 0: No SMBAlert response address header.	0
14	TIMEOUT	RW0	Timeout or Tlow error flag. It can be reset by user writing 0, or reset by hardware when PE becomes low. 1: SCL is low and has reached 25mS, or the accumulated clock expansion time of the master device low level exceeds 10mS, or the accumulated time of the slave device low level exceeds 25mS; 0: No timeout error. *Note: When this bit is set in slave mode, the slave device resets the communication and the hardware releases the bus. When this bit is set in master mode, hardware issues a stop condition.*	0
13	Reserved	RO	Reserved	0
12	PECERR	RW0	PEC error flag occurs during reception. This bit can be reset by user writing 0, or reset by hardware when PE becomes low. 1: PEC error. After PEC is received, NAK is returned; 0: No PEC error.	0
11	OVR	RW0	Overrun and underrun flag. 1: Overrun or underrun event occurs: In case of NOSTRETCH=1, when a new byte is received in the receiver mode and the content in the data register has not been read, the newly received byte will be lost. In the transmitter mode, no new data is written into the data register, and the same byte will be sent twice;	0

			0: No overrun and underrun event.	
10	AF	RW0	Acknowledge failure flag. This bit can be reset by user writing 0, or reset by hardware when PE becomes low. 1: Acknowledge error; 0: Normal acknowledge.	0
9	ARLO	RW0	Arbitration lost flag. It can be reset by user writing 0, or reset by hardware when PE becomes low. 1: Arbitration lost is detected and the module loses control of the bus; 0: Normal arbitration.	0
8	BERR	RW0	Bus error flag. It can be reset by user writing 0, or reset by hardware when PE becomes low. 1: Start or stop condition error; 0: Normal.	0
7	TxE	RO	Data register empty flag, which can be cleared by writing data to the data register, or it is automatically cleared by hardware after a start or stop bit is generated, or when PE is 0. 1: When the data is transmitted, the transmit data register is empty; 0: The data register is non-empty.	0
6	RxNE	RO	The data register is not an empty log bit, which will be cleared by reading and writing to the data register, or by the hardware when PE is 0. 1: When receiving data, the data register is not empty. 0: Normal.	0
5	Reserved	RO	Reserved.	0
4	STOPF	RO	Stop event flag. After the user reads the status register1, writing to the control register1 will clear this bit, or when PE is 0, the hardware will clear this bit. 1: After the response, the slave device will detect a stop event on the bus; 0: No stop event is detected.	0
3	ADD10	RO	10-bit address header sent flag. After the user reads the status register1, writing to the control register1 will clear this bit, or when PE is 0, the hardware will clear this bit. 1: In 10-bit address mode, the master device has sent the first address byte; 0: None	0
2	BTF	RO	Byte transmission end flag. After the user reads	0

			the status register1, reading and writing to the data register will clear this bit. During transmission, after a start or stop event is initiated, or when PE is 0, this bit will be cleared by hardware. 1: Byte transmission completed. In case of NOSTRETCH=0: when a new data is sent and the data register has not been written with new data during transmission; when a new byte is received but the data register has not been read; 0: None	
1	ADDR	RW0	Address transmitted/matched flag. After the user reads the status register 1, the read operation of the status register2 will clear this bit, or when PE is 0, the hardware will clear this bit. Master mode: 1: End of address transmission: In 10-bit address mode, the bit will be changed to be set after the ACK of the second byte of the address is received; in 7-bit address mode, the bit will be set after the ACK of the address is received; 0: The address transmission is not finished. Slave mode: 1: The received address matches; 0: The address does not match or no address is received.	0
0	SB	RO	Start bit transmission flag. After reading the status register 1, the operation of writing the data register will clear this bit, or when PE is 0, the hardware will clear this bit. 1: The start bit has been transmitted; 0: The start bit has not been transmitted.	0

18.12.7 I2C Status Register 2 (I2Cx_STAR2) (x=1/2)

Offset address: 0x18

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PEC[7:0]								DUALF	SMBHOST	SMBDEFAULT	GENCALL	Reserved	TRA	BUSY	MSL

Bit	Name	Access	Description	Reset value
[15:8]	PEC[7:0]	RO	Packet error checking. When PEC is enabled (ENPEC is set), this domain stores the value of PEC.	0
7	DUALF	RO	Matched detection flag. When the stop bit or start bit is generated, or when PE=0, the hardware will clear	0

			this bit. 1: The received address matched with OAR2; 0: The received address matched with OAR1.	
6	SMBHOST	RO	SMBus host header flag. When the stop bit or start bit is generated, or when PE=0, the hardware will clear this bit. 1: When SMBTYPE=1 and ENARP=1, the SMBus host address will be received; 0: SMBus host address is not received.	0
5	SMBDEFAULT	RO	SMBus device default address flag. When the stop bit or start bit is generated, or when PE=0, the hardware will clear this bit. 1: When ENARP=1, the default address of the SMBus device is received; 0: No address is received.	0
4	GENCALL	RO	General call address flag. When the stop bit or start bit is generated, or when PE=0, the hardware will clear this bit. 1: When ENGC=1, the address of general call is received; 0: No general call address is received.	0
3	Reserved	RO	Reserved.	0
2	TRA	RO	Transmitter/receiver flag, cleared by hardware when a stop event (STOPF=1) is detected, repeated start condition or bus arbitration is lost (ARLO=1) or PE=0. 1: Data has been sent; 0: Data is received. This bit is determined by R/W bit of address byte.	0
1	BUSY	RO	Bus busy flag. This bit is cleared when a stop bit is detected. When the interface is disabled (PE=0), the information is still updated. 1: Busy bus: SDA or SCL has a low level; 0: The bus is idle and does not have communication.	0
0	MSL	RO	Master/slave mode indication. When the interface is in master mode (SB=1), the hardware will set this bit. When the bus detects a stop bit and the arbitration is lost, or PE=0, the hardware will clear this bit.	0

18.12.8 I2C Clock Register (I2Cx_CKCFGR) (x=1/2)

Offset address: 0x1C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F/S	DUTY	Reserved		CCR[11:0]											

Bit	Name	Access	Description	Reset value
15	F/S	RW	Master mode selection. 1: Fast mode; 0: Standard mode.	0
14	DUTY	RW	Duty cycle in the fast mode: 1: $T_{Low\ level}/T_{High\ level}$=16/9; 0: $T_{Low\ level}/T_{High\ level}$=2.	0
[13:12]	Reserved	RO	Reserved.	0
[11:0]	CCR[11:0]	RW	Clock frequency division factor. These bits determine the frequency waveform of the SCL clock.	0

18.12.9 I2C Rise Time Register (I2Cx_RTR) (x=1/2)

Offset address: 0x20

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

Bit	Name	Access	Description	Reset value
[15:6]	Reserved	RO	Reserved	0
[5:0]	TRISE[5:0]	RW	Maximum rise time domain. The rise time of SCL in master mode is set at this bit. The maximum rising edge time is equal to TRISE-1 clock cycle. This bit can only be set when PE is cleared. For example, if the input clock cycle of the IIC module is 125nS and the value of TRISE is 9h, then the maximum rising edge time is (9-1)*125nS, i.e., 1000nS.	000010b

Chapter 19 Serial Peripheral Interface (SPI)

SPI supports data exchange in three-wire synchronous serial mode, plus chip line selection supports hardware switching between master and slave modes, and supports communication with a single data line.

19.1 Main Features

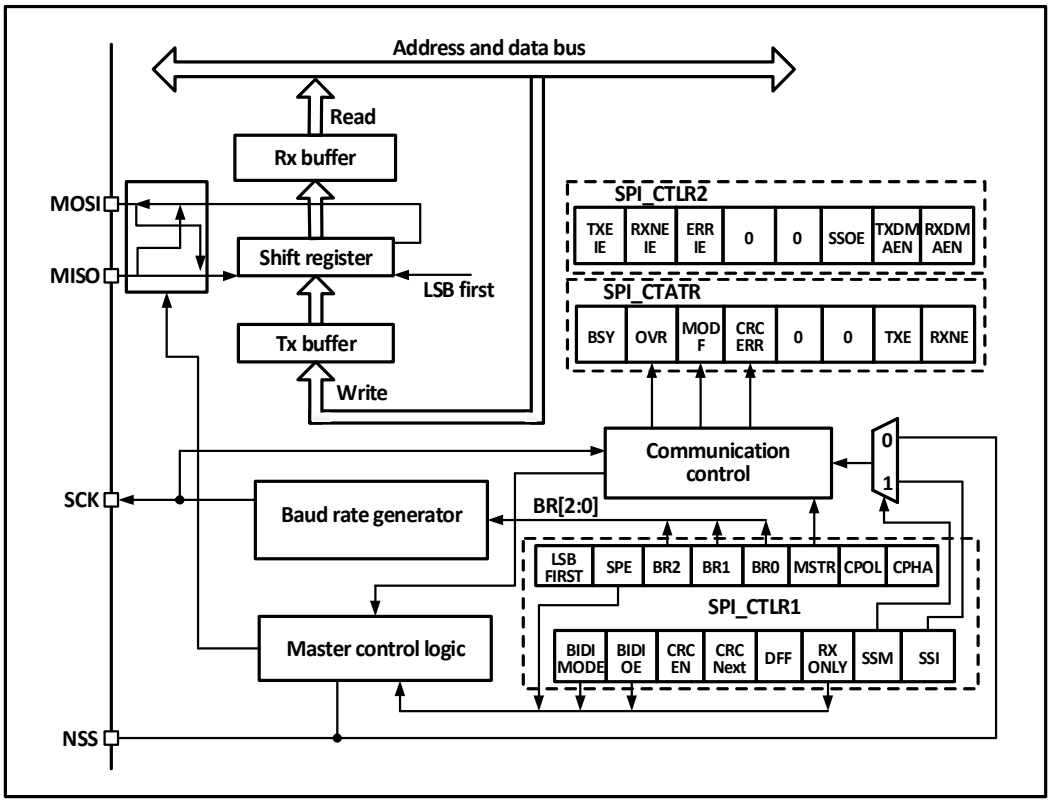
19.1.1 SPI Features

- Support full-duplex synchronous serial mode
- Support 1-wire half-duplex mode
- Support master mode and slave mode, multi-slave mode
- Support 8-bit or 16-bit data structures
- The highest clock frequency is supported to half of Fpclk
- Data ordering supports MSB or LSB first
- Support hardware or software control of the NSS pin
- Transceiver supports hardware CRC
- Transceiver buffer supports DMA transfer
- Support modifying clock phase and polarity

19.2 SPI Functional Description

19.2.1 Overview

Figure 19-1 SPI structure diagram



As can be seen from Figure 19-1, the 4 pins related to SPI are MISO, M0SI, SCK and NSS. The MISO pin is the data input pin when the SPI module works in the master mode; it is the data output pin when it works in the slave mode. When the MOSI pin works in master mode, it is a data output pin; when it works in slave mode, it is a data input pin. SCK is the clock pin, the clock signal is always output by the master, and the slave receives the clock signal and synchronizes data transmission and reception. The NSS pin is a chip select pin and has the following uses:
1) NSS is controlled by software: At this time, SSM is set, and the internal NSS signal is determined by SSI to output high or low. This situation is generally used in SPI master mode;
2) NSS is controlled by hardware: When the NSS output is enabled, i.e., when SSOE is set, the NSS pin will be actively pulled down when the SPI host sends the output outward, and a hardware error will be generated if the NSS pin cannot be successfully pulled down and pulled down, indicating that there are other host devices on the main line that are communicating; SSOE If it is not set, it can be used in multi-master mode. If it is pulled low, it will be forced into slave mode, and the MSTR bit will be automatically cleared.

The working mode of the SPI can be configured through CPHA and CPOL. If CPHA is set, it means that the module samples the data on the second edge of the clock, and the data is latched. If CPHA is not set, it means that the SPI module samples on the first edge of the clock, and the data is latched. CPOL indicates whether the clock remains high or low when there is no data.

The host and device need to be set to the same SPI mode, and the SPE bit needs to be cleared before configuring SPI mode. The DEF bit can determine whether the single data length of the SP is 8 or 16 bits. LSBFIRST can control whether a single data word is high or low.

Table 19-1 SPI mode differences

	Mode 0	Mode 1	Mode 2	Mode 3
CPOL	0	1	1	1
CPHA	0	0	0	1

19.2.2 Master Mode

When the SPI module works in master mode, the serial clock is generated by SCK. To configure into master mode do the following steps:
Configure the BR[2:0] bits in the control register to determine the clock;
Configure the CPOL and CPHA bits to determine the SPI mode;
Configure DEF to determine the data word length;
Configure LSBFIRST to determine the frame format;
Configure the NSS pin, such as setting the SSOE bit to let the hardware reset NSS. You can also set the SSM bit and set the SSI bit high;
To set the MSTR bit and the SPE bit, it is necessary to ensure that NSS is already high at this time.

When you need to send data, you only need to write the data to be sent to the data register. SPI will send data from the transmit buffer to the shift register in parallel, and then send the data from the shift register according to the setting of LSBFIRST. When the data has reached the shift register, the TXE flag will be set. If it has been set TXEIE, then an interrupt will be generated. If the TXE flag bit is set, data needs to be filled into the data register to maintain a complete data flow.

When the receiver receives data, when the last sampling clock edge of the data word arrives, the data is transferred from the shift register to the receive buffer in parallel, the RXNE bit is set, and if the RXNEIE bit was previously

set, it will also generate interrupt. At this time, the data register should be read as soon as possible to remove the data.

Figure 19-2 SPI master mode read/write mode 0

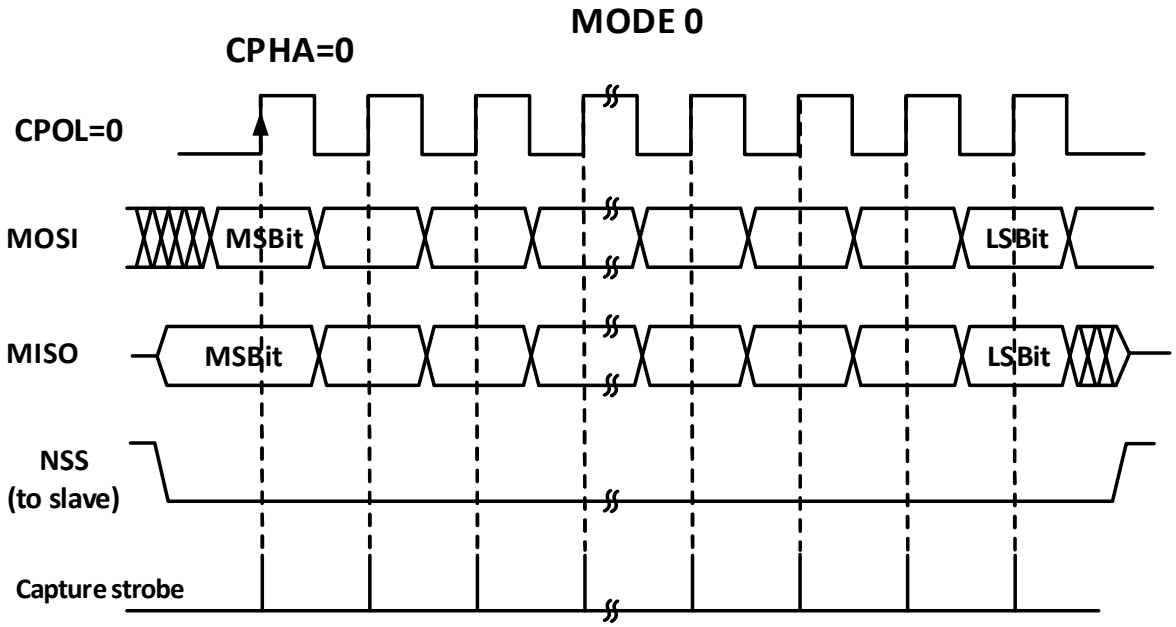


Figure 19-3 SPI master mode read/write mode 1

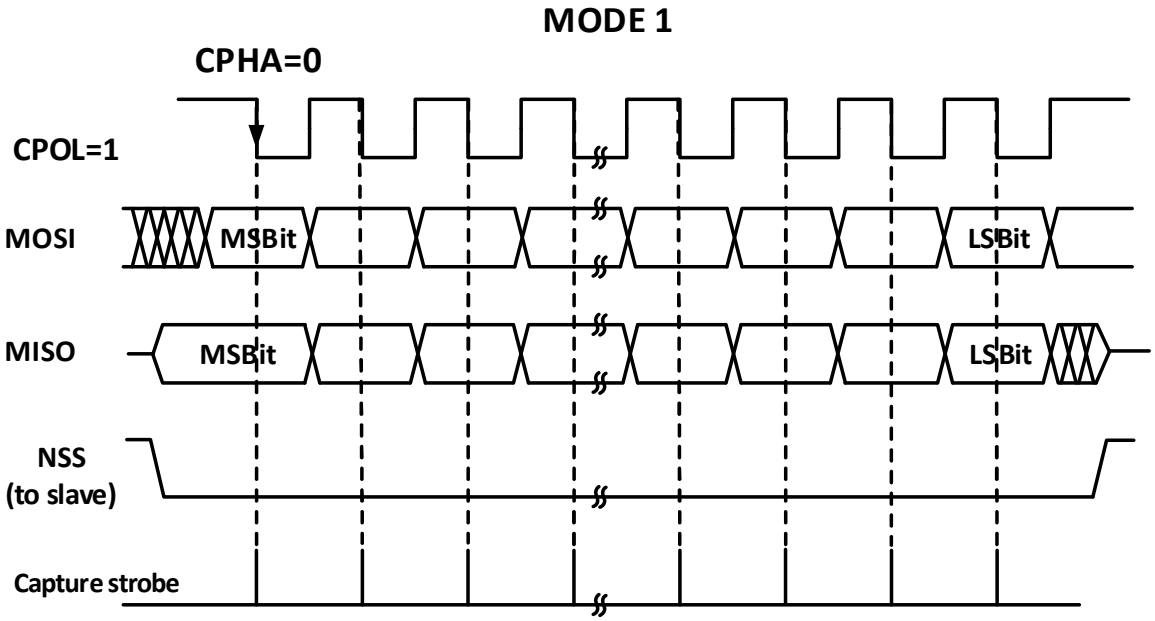


Figure 19-4 SPI master mode read/write mode 2

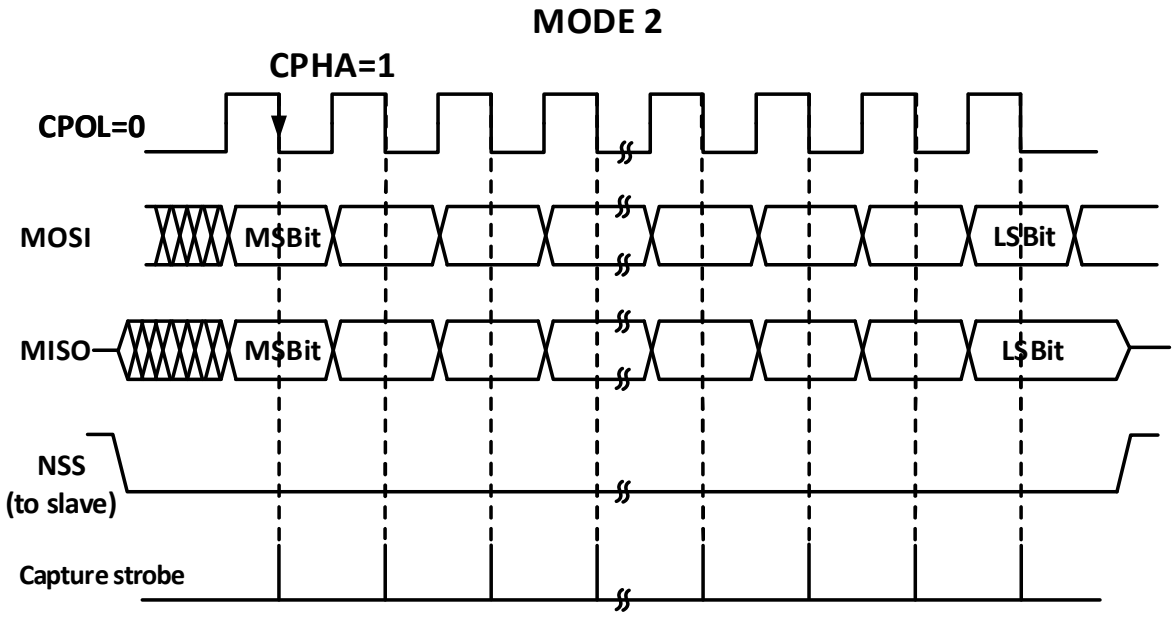
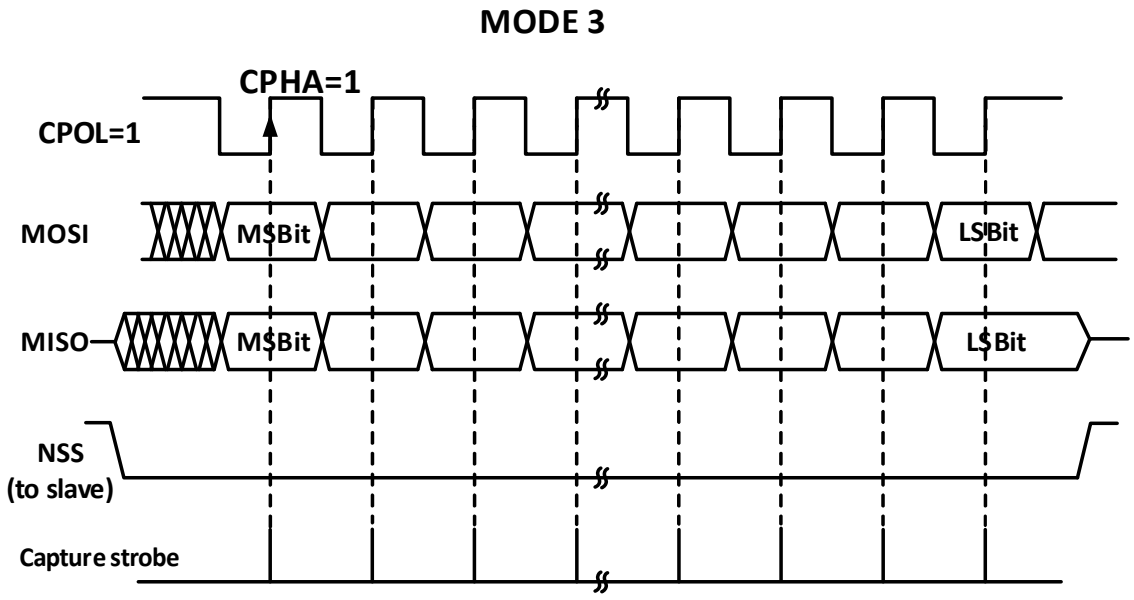


Figure 19-5 SPI master mode read/write mode 3



19.2.3 Slave Mode

When the SPI module works in slave mode, SCK is used to receive the clock sent by the host, and its own baud rate setting is invalid. The steps to configure into slave mode are as follows:

Configure the DEF bit to set the data bit length;

Configure CPHA to match the host operating mode;

Determine the SPI operating mode based on the transmitting and receiving configurations and CPOL;

If transmitting in slave mode is required, CPOL needs to be set and configured as mode 2 or mode 3, and the host changes the configuration as required;

If only receiving in slave mode is required, only the host CPOL mode needs to be matched;

Configure LSBFIRST to match the host data frame format;
In hardware management mode, the NSS pin needs to be kept at a low level. If NSS is set to software management (SSM is set), then please keep SSI not set;

Clear the MSTR bit and set the SPE bit to enable SPI mode.

When transmitting, when the first slave receive sampling edge occurs on SCK, the slave starts to transmit. The process of sending is to move the data in the send buffer to the send shift register. When the data in the send buffer is moved to the shift register, the TXE flag will be set. If the TXEIE bit was previously set, an interrupt will be generated.

When receiving, after the last clock sampling edge, the RXNE bit is set, the byte received by the shift register is transferred to the receive buffer, and the read operation of the read data register can obtain the data in the receive buffer. An interrupt will be generated if RXNEIE is set before RXNE is set.

Figure 19-6 SPI slave mode read mode 0

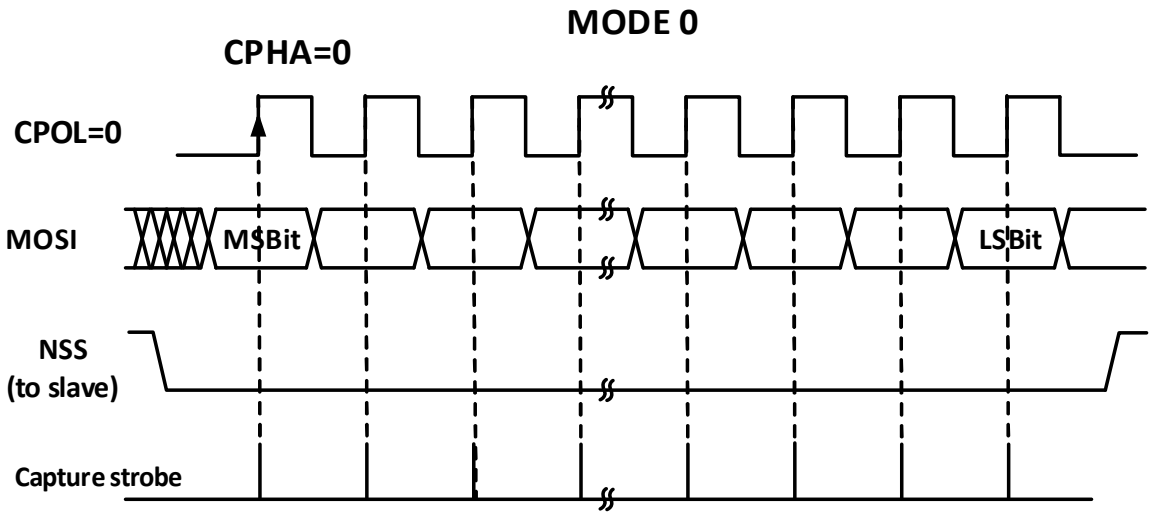


Figure 19-7 SPI slave mode read mode 1

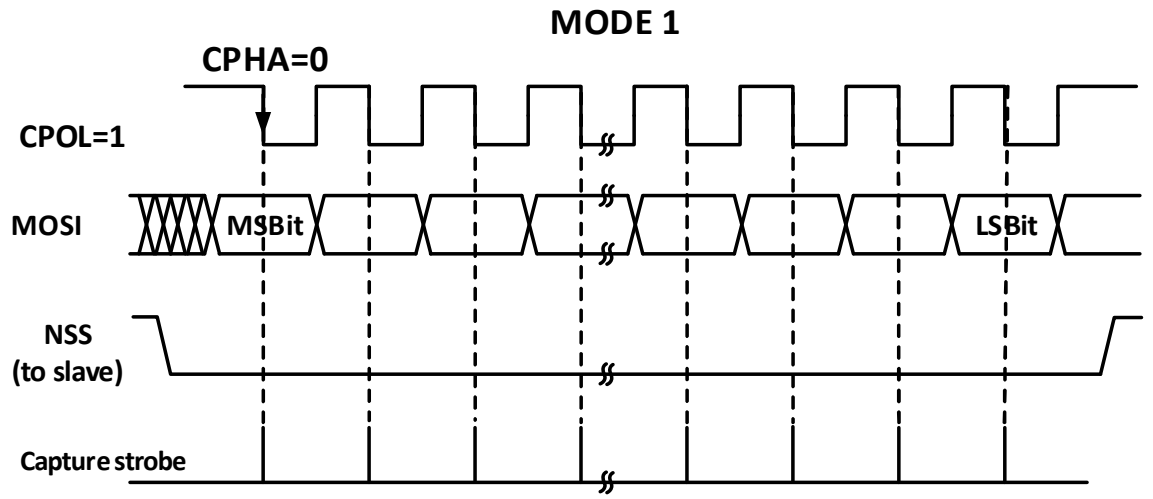


Figure 19-8 SPI slave mode read/write mode 2

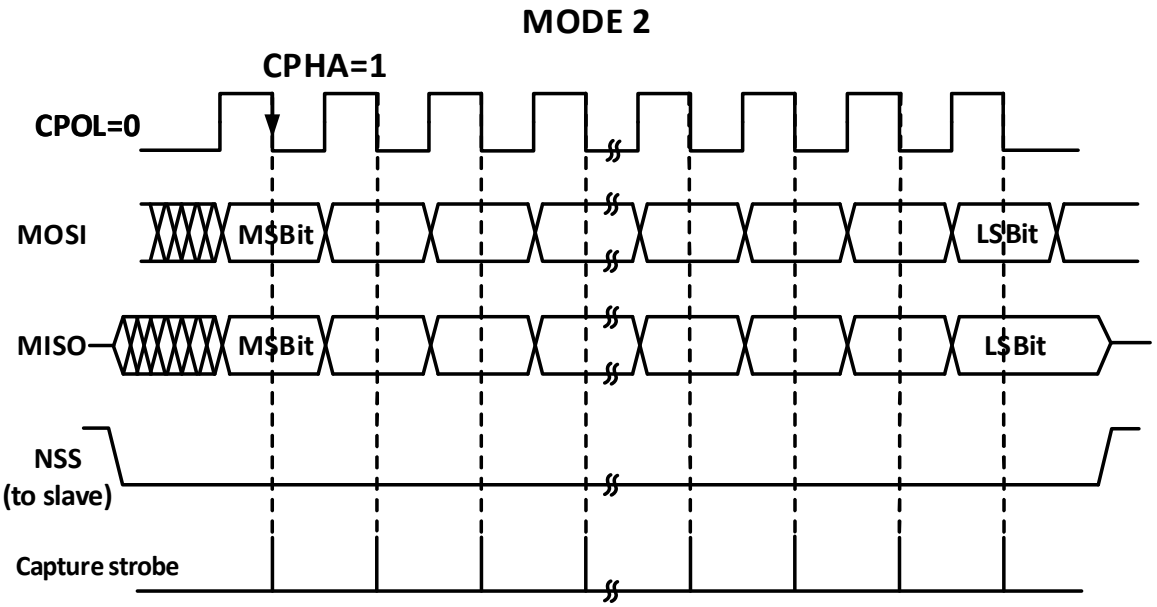
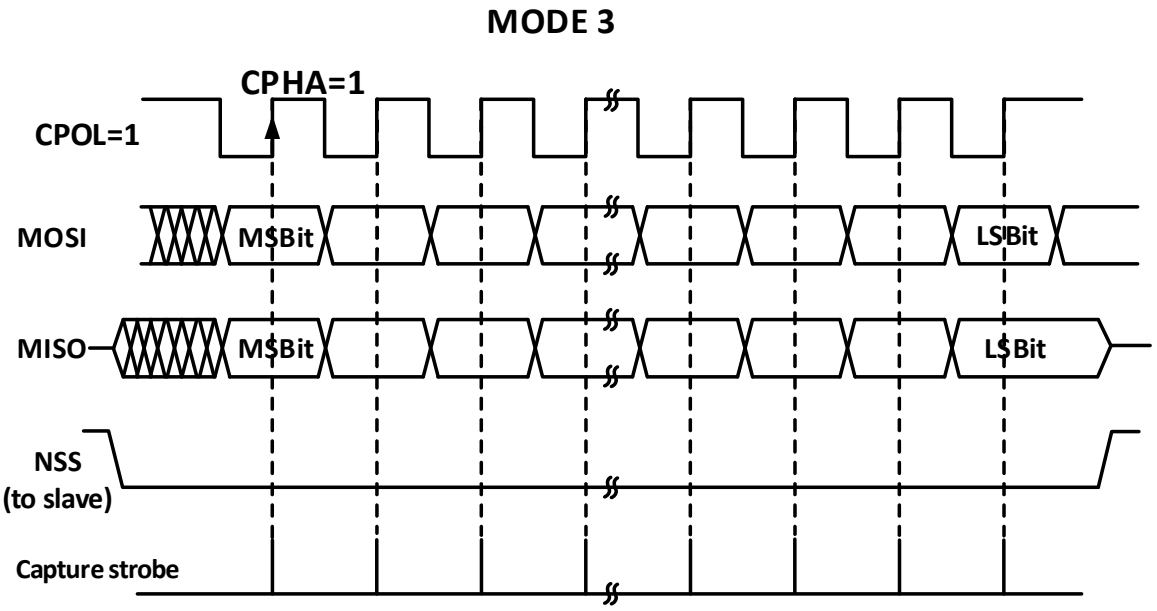


Figure 19-9 SPI slave mode read/write mode 3



19.2.4 Simplex Mode

The SPI interface can work in half-duplex mode, that is, the master device uses the MOSI pin, and the slave device uses the MISO pin for communication. When using half-duplex communication, BIDIMODE needs to be set, and BIDIOE is used to control the transmission direction.

Setting the RXONLY bit in normal full-duplex mode can set the SPI module to receive-only simplex mode. After RXONLY is set, a data pin will be released. The pins released in master mode and slave mode are not the same. It is also possible to ignore the received data and set the SPI to transmit-only mode.

19.2.5 CRC

The SPI module uses CRC to ensure the reliability of full-duplex communication, and separate CRC calculators are used for data transmission and reception. The polynomial of the CRC calculation is determined by the polynomial register. For 8-bit data width and 16-bit data width, different calculation methods are used respectively.

Setting the CRCEN bit enables CRC and resets the CRC calculator. After the last data byte is sent, setting the CRCNEXT bit will send the calculation result of the TXCRCR calculator after the current byte is sent. At the same time, if the last received value of the receive shift register is not the same as the locally calculated value of RXCRCR, then the CRCERR bit will be set. To use the CRC, you need to set the polynomial calculator and set the CRCEN bit when configuring the SPI working mode, and set the CRCNEXT bit in the last word or half word to send the CRC and check the received CRC. Note that the CRC calculation polynomials of the sender and receiver should be unified.

19.2.6 DMA

The SPI module supports the use of DMA to speed up data communication. You can use DMA to fill in data into the send buffer, or use DMA to take data from the receive buffer in time. DMA will take or send data in time with RXNE and TXE as signals. DMA can also work in simplex or CRC mode.

19.2.7 Error

- Master mode failure error

When the SPI works in the NSS pin hardware management mode, an external operation of pulling down the NSS pin occurs; or in the NSS pin software management mode, the SSI bit is cleared; or the SPE bit is cleared, causing the SPI to be turned off; or the MSTR bit is cleared, the SPI enters slave mode. An interrupt will also be generated if the ERRIE bit has been set. Clear MODF bit steps: first perform a read or write operation to R16_SPI1_STATR, and then write to R16_SPI1_CTLR1.

- Overflow error

If the master sends data and there is unread data in the slave's receive buffer, an overrun error occurs, the OVR bit is set, and an interrupt is generated if ERRIE is set. Sending an overflow error should restart the current transfer. Reading the data register followed by the status register will clear this bit.

- CRC error

When the received CRC word does not match the value of RXCRCR, a CRC error will occur, and the CRCERR bit will be set.

19.2.8 Interrupt

The interrupt of the SPI module supports 5 interrupt sources. The 2 events of the send buffer empty and the receive buffer non-empty will set TXE and RXNE respectively, and an interrupt will be generated when the TXEIE and RXNEIE bits are set respectively. In addition, the 3 errors mentioned above will also generate interrupts, namely MODF, OVR and CRCERR. After the ERRIE bit is enabled, these 3 errors will also generate error interrupts.

19.3 Register Description

Table 19-2 SPI1-related registers

Name	Access address	Description	Reset value

R16_SPI1_CTLR1	0x40013000	SPI1 control register 1	0x0000
R16_SPI1_CTLR2	0x40013004	SPI1 control register 2	0x0000
R16_SPI1_STATR	0x40013008	SPI1 status register	0x0002
R16_SPI1_DATAR	0x4001300C	SPI1 data register	0x0000
R16_SPI1_CRCR	0x40013010	SPI1 polynomial register	0x0007
R16_SPI1_RCRCR	0x40013014	SPI1 receive CRC register	0x0000
R16_SPI1_TCRCR	0x40013018	SPI1 transmit CRC register	0x0000
R16_SPI1_HSCR	0x40013024	SPI1 high-speed control register	0x0000

Table 19-3 SPI2-related registers

Name	Access address	Description	Reset value
R16_SPI2_CTLR1	0x40003800	SPI2 control register 1	0x0000
R16_SPI2_CTLR2	0x40003804	SPI2 control register 2	0x0000
R16_SPI2_STATR	0x40003808	SPI2 status register	0x0002
R16_SPI2_DATAR	0x4000380C	SPI2 data register	0x0000
R16_SPI2_CRCR	0x40003810	SPI2 polynomial register	0x0007
R16_SPI2_RCRCR	0x40003814	SPI2 receive CRC register	0x0000
R16_SPI2_TCRCR	0x40003818	SPI2 transmit CRC register	0x0000
R16_SPI2_HSCR	0x40003824	SPI2 high-speed control register	0x0000

19.3.1 SPI Control Register 1 (SPIx_CTLR1) (x=1/2)

Offset address: 0x00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIDIMODE	BIDIOE	CRCEN	CRCNEXT	DFF	RXONLY	SSM	SSI	LSBFIRST	SPE	BR[2:0]			MSTR	CPOL	CPHA

Bit	Name	Access	Description	Reset value
15	BIDIMODE	RW	Unidirectional data mode enable. 1: 1-wire bidirectional mode; 0: 2-wire bidirectional mode.	0
14	BIDIOE	RW	Single-line output enable, used in conjunction with BIDIMODE. 1: Enable output, only transmit; 0: Disable output, only receive.	0
13	CRCEN	RW	Hardware CRC check enable, this bit can only be written when SPE is 0, this bit can only be used in full duplex mode. 1: Enable CRC calculation; 0: Disable CRC calculation.	0
12	CRCNEXT	RW	Value of the transmit CRC register after next data transfer. This bit should be set immediately after writing the last data to the data register.	0

			1: Transmit the CRC result; 0: Continue to transmit the data of the data register.	
11	DFF	RW	Data frame length, this bit can only be written when SPE is 0. 1: Use 16-bit data length to transmit and receive; 0: Use 8-bit data length for transmission and reception.	0
10	RXONLY	RW	Rx only in 2-wire mode. It is used in conjunction with BIDIMODE. Setting this bit allows the device to only receive and not transmit. 1: Receive only, simplex mode; 0: Full-duplex mode.	0
9	SSM	RW	CS pin management, this bit determines whether the level of the NSS pin is controlled by hardware or software. 1: Software control NSS pin; 0: Hardware controls NSS pin.	0
8	SSI	RW	CS pin control. When SSM is set, this bit determines the level of the NSS pin. 1: NSS is high level; 0: NSS is low level.	0
7	LSBFIRST	RW	Frame format control. This bit cannot be modified during communication. 1: Transmit LSB first; 0: Transmit MSB first.	0
6	SPE	RW	SPI enable. 1: Enable SPI; 0: Disable SPI.	0
[5:3]	BR[2:0]	RW	Baud rate setting domain, this domain cannot be modified during communication. 000: FPCLK/2; 001: FPCLK/4; 010: FPCLK/8; 011: FPCLK/16; 100: FPCLK/32; 101: FPCLK/64; 110: FPCLK/128; 111: FPCLK/256. *Note: This bit applies only if the HSRXEN bit is 0; when the HSRXEN bit is 1, the SCK frequency is FPCLK/(BR+2).*	000b
2	MSTR	RW	Master/slave setting. It cannot be modified during communication. 1: Configure as the master device; 0: Configured as the slave device.	0
1	CPOL	RW	Clock polarity selection. It cannot be modified during communication.	0

			1: In idle state, SCK remains high; 0: In idle state, SCK remains low.	
0	CPHA	RW	Clock phase setting. It cannot be modified during communication. 1: Data sampling starts from the second clock edge; 0: Data sampling starts from the first clock edge.	0

19.3.2 SPI Control Register 2 (SPIx_CTLR2) (x=1/2)

Offset address: 0x04

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TXEIE	RXNEIE	ERRIE	Reserved		SSOE	TXDMAEN	RXDMAEN

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved	0
7	TXEIE	RW	Transmit buffer empty interrupt enable. Setting this bit allows an interrupt to be generated when TXE is set.	0
6	RXNEIE	RW	Receive buffer not empty interrupt enable. Setting this bit allows an interrupt to be generated when RXNE is set.	0
5	ERRIE	RW	Error interrupt enable. Setting this bit enables an interrupt to be generated when an error occurs (CRCERR, OVR, MODF).	0
[4:3]	Reserved	RO	Reserved	0
2	SSOE	RW	SS output enable. Disable SS output to work in multi-master mode. 1: Enable SS output; 0: Disable SS output in master mode.	0
1	TXDMAEN	RW	Transmit buffer DMA enable. 1: Enable transmit buffer DMA; 0: Disable transmit buffer DMA.	0
0	RXDMAEN	RW	Receive buffer DMA enable. 1: Enable receive buffer DMA; 0: Disable receive buffer DMA.	0

19.3.3 SPI Status Register (SPIx_STATR) (x=1/2)

Offset address: 0x08

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BSY	OVR	MODF	CRCERR	Reserved		TXE	RXNE

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved	0
7	BSY	RO	Busy flag. Set and reset by hardware. 1: SPI is communicating, or the transmit buffer is not empty; 0: SPI is not communicating.	0
6	OVR	RW0	Overflow flag. Set by hardware, and reset by software. 1: An overflow error occurred; 0: No overflow error occurred.	0
5	MODF	RO	Mode error flag. Set by hardware, and reset by software. 1: A mode error occurred; 0: No mode error occurred.	0
4	CRCERR	RW0	CRC error flag. Set by hardware, and reset by software. 1: The received CRC value is inconsistent with the RCRCR value; 0: The received CRC value is the same as the RCRCR value.	0
[3:2]	Reserved	RO	Reserved	0
1	TXE	RO	Transmit buffer empty flag. 1: The transmit buffer is empty; 0: The transmit buffer is not empty.	1
0	RXNE	RO	Receive buffer not empty flag. 1: The receive buffer is not empty; 0: The receive buffer is empty.	0

19.3.4 SPI Data Register (SPIx_DATAR) (x=1/2)

Offset address: 0x0C

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

Bit	Name	Access	Description	Reset value
[15:0]	DR[15:0]	RW	Data register. The data register is used to store the received data or pre-store the data to be sent, so the read and write of the data register is actually corresponding to the different operation area, where the read corresponds to the receiving buffer and the write corresponds to the sending buffer. The reception and transmission of data can be 8-bit or 16-bit, and how many bits of data need to be determined before transmission. When using 8 bits	0

			for data transmission, only the lower 8 bits of the data register are used, and the high 8 bits are forced to 0 when received. The use of 16-bit data structures causes all 16-bit data registers to be used.	

19.3.5 SPI Polynomial Register (SPIx_CRCR) (x=1/2)

Offset address: 0x10

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

Bit	Name	Access	Description	Reset value
[15:0]	CRCPOLY[15:0]	RW	CRC polynomial. This domain defines the polynomial used for the CRC calculation.	0007h

19.3.6 SPI Receive CRC Register (SPIx_RCRCR) (x=1/2)

Offset address: 0x14

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

Bit	Name	Access	Description	Reset value
[15:0]	RXCRC[15:0]	RO	Rx CRC value. Stores the result of the calculated CRC check of the received bytes. Setting CRCEN will reset this register. The calculation method uses the polynomial used by CRCPOLY. In 8-bit mode, only the lower 8 bits participate in the calculation, and in 16-bit mode, all 16 bits will participate in the calculation. This register needs to be read when BSY is 0.	0

19.3.7 SPI Transmit CRC Register (SPIx_TCRCR) (x=1/2)

Offset address: 0x18

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

Bit	Name	Access	Description	Reset value
[15:0]	TXCRC[15:0]	RO	Send a CRC value. Stores the calculated results of the CRC check of the bytes that have been sent. Setting CRCEN resets the register. The calculation method uses the polynomials used by CRCPOLY. In 8-bit mode, only the lower 8 bits participate in the calculation, while in 16-bit mode, all 16 bits participate in the calculation. You need to read this	0

			register when BSY is 0.	

19.3.8 SPI High-speed Control Register (SPIx_HSCR) (x=1/2)

Offset address: 0x24

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

Bit	Name	Access	Description	Reset value
[15:1]	Reserved	RO	Reserved	0
0	HSRXEN	RW	Read enable in SPI high-speed mode. 1: High-speed read mode enabled; 0: High-speed read mode disabled.	0

Chapter 20 USB Full-speed Host/Device Controller (USBFS)

20.1 USB Controller Introduction

Embedded USB2.0 controller and USB-PHY, with dual roles of host controller and USB device controller. When used as a host controller, it supports low-speed, full-speed and high-speed USB devices / HUB. When used as a device controller, it can be flexibly set to low-speed, full-speed or high-speed mode to adapt to a variety of applications.

USB controller and transceiver are embedded in the chip, and the characteristics are as follows:

- Dual-role device controller, supporting USB Host function and USB Device function.
- Both host and device modes support USB2.0 full-speed 12Mbps or low-speed 1.5Mbps.
- Support software HNP and SRP protocols.
- Support USB control transfer, batch transfer, interrupt transmission and synchronous / real-time transmission.
- Support packets up to 64 bytes, built-in FIFO, interrupt and DMA.

20.2 Register Description

The USB registers are divided into 3 parts, and some registers are multiplexed in the host and device modes.

- USB global registers
- USB device control registers
- USB host control registers

20.2.1 Global Register Description

Table 20-1 USBFS-related registers

Name	Access address	Description	Reset value
R8_USB_CTRL	0x50000000	USB control register	0x06
R8_USB_INT_EN	0x50000002	USB interrupt enable register	0x00
R8_USB_DEV_AD	0x50000003	USB device address register	0x00
R8_USB_MIS_ST	0x50000005	USB miscellaneous status register	0xXX
R8_USB_INT_FG	0x50000006	USB interrupt flag register	0x20
R8_USB_INT_ST	0x50000007	USB interrupt status register	0xXX
R16_USB_RX_LEN	0x50000008	USB receive length register	0xXX

20.2.1.1 USB Control Register (R8_USB_CTRL)

Bit	Name	Access	Description	Reset value
7	RB_UC_HOST_MODE	RW	USB operating mode selection: 1: Host mode; 0: Device mode.	0
6	RB_UC_LOW_SPEED	RW	USB bus signal transfer rate selection: 1: 1.5Mbps; 0: 12Mbps.	0
5	RB_UC_DEV_PU_EN	RW	The USB device enable and internal pull-up resistor control bits in USB Device mode, a	

			value of 1 enables USB device transfer and enables the internal pull-up resistor.	
[5:4]	MASK_UC_SYS_CTRL	RW	See the table below to configure USB system.	0
3	RB_UC_INT_BUSY	RW	The USB transmission completion interrupt flag automatically pauses the enable bit before zero is cleared: 1: Automatically pause before the interrupt flag UIF_TRANSFER is cleared, answer busy NAK in device mode, and automatically pause subsequent transmission in host mode. 0: No pause.	0
2	RB_UC_RST_SIE	RW	USB protocol processor reset control bits: 1: Disable USB and reset the USB protocol processor (SIE), PA11/PA12 as normal GPIOs; 0: Enable USB, PA11/PA12 as USB dedicated pins.	1
1	RB_UC_CLR_ALL	RW	USB FIFO and interrupt flags clear: 1: Forced clearing and clearing; 0: Not clear.	1
0	RB_UC_DMA_EN	RW	DMA and DMA interrupt control for USB: 1: Enable DMA function and DMA interrupt; 0: Disable DMA.	0

When RB_UC_RST_SIE=0, the USB system control combination is composed of RB_UC_HOST_MODE and MASK_UC_SYS_CTRL:

Table 20-2 USB system control combination

RB_UC_HOST_MODE	MASK_UC_SYS_CTRL	USB system control description
0	00	Disable the USB device function, turn off the internal pull-up resistor.
0	01	Enable the USB device function, close the internal pull-up resistor, and need to add an external pull-up.
0	1x	Enable USB device function, enable internal 1.5K pull-up resistor. This pull-up resistor takes precedence over the pull-down resistor and can also be used in GPIO mode.
1	00	USB host mode, normal working state.
1	01	USB host mode, forcing DP/DM to output SE0 state.
1	10	USB host mode, forces DP/DM to output J state.
1	11	USB host mode, force DP/DM to output K state/wake up.

20.2.1.2 USB Interrupt Enable Register (R8_USB_INT_EN)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved	0
6	RB_UIE_DEV_NAK	RW	USB device mode, receive NAK interrupt: 1: Enable interrupt; 0: Disable interrupt.	0
5	RB_UID_1_WIRE	RW	USB 1-wire mode enable 1: Enable 0: Disable	0
4	RB_UIE_FIFO_OV	RW	FIFO overflow interrupt: 1: Enable interrupt; 0: Disable interrupt.	0
3	RB_UIE_HST_SOF	RW	USB host mode, SOF timing interrupt: 1: Enable interrupt; 0: Disable interrupt.	0
2	RB_UIE_SUSPEND	RW	USB bus suspend or wakeup event interrupt: 1: Enable interrupt; 0: Disable interrupt.	0
1	RB_UIE_TRANSFER	RW	USB transfer complete interrupt: 1: Enable interrupt; 0: Disable interrupt.	0
0	RB_UIE_DETECT	RW	USB host mode, USB device connect or disconnect event interrupt: 1: Enable interrupt; 0: Disable interrupt.	0
	RB_UIE_BUS_RST	RW	USB device mode, USB bus reset event interrupt: 1: Enable interrupt; 0: Disable interrupt.	0

20.2.1.3 USB Device Address Register (R8_USB_DEV_AD)

Bit	Name	Access	Description	Reset value
7	RB_UDA_GP_BIT	RW	USB general-purpose flag, user-defined.	0
[6:0]	MASK_USB_ADDR	RW	Host mode: current operated USB device address; Device Mode: USB own address.	0

20.2.1.4 USB Miscellaneous Status Register (R8_USB_MIS_ST)

Bit	Name	Access	Description	Reset value
7	RB_UMS_SOF_PRES	RO	SOF packet presage status in USB host mode: 1: SOF packet is about to be transmitted, if there are other USB packets at this time, it will be automatically delayed; 0: No SOF packet is transmitted.	x
6	RB_UMS_SOF_ACT	RO	SOF packet transfer status in USB host mode: 1: SOF packet is being sent; 0: Send complete or idle.	x

5	RB_UMS_SIE_FREE	RO	USB protocol handler free: 1: The protocol device is free; 0: Busy, USB transfer in progress.	1
4	RB_UMS_R_FIFO_RDY	RO	USB receive FIFO data ready: 1: The receive FIFO is not empty; 0: The receive FIFO is empty.	0
3	RB_UMS_BUS_RST	RO	USB bus reset: 1: The current USB bus is in reset state; 0: The current USB bus is in a non-reset state.	x
2	RB_UMS_SUSPEND	RO	USB suspend: 1: The USB bus is in a suspended state, and there is no USB activity for a period of time; 0: The USB bus is in a non-suspend state.	0
1	RB_UMS_DM_LEVEL	RO	In USB host mode, level state of the DM pin when the device is just connected to the USB port is used to judge the speed: 1: High level/low-speed; 0: Low level/full-speed.	0
0	RB_UMS_DEV_ATTACH	RO	USB device attach status for the port in USB host mode: 1: The port has been connected to a USB device; 0: The port has no USB device connected.	0

20.2.1.5 USB Interrupt Flag Register (R8_USB_INT_FG)

Bit	Name	Access	Description	Reset value
7	RB_U_IS_NAK	RO	In USB device mode, NAK response status: 1: Respond to NAK during the current USB transfer process; 0: No NAK response.	0
6	RB_U_TOG_OK	RO	Current USB transfer DATA0/1 sync flag match status: 1: Synchronized; 0: Not synchronized.	0
5	RB_U_SIE_FREE	RO	USB Protocol Handler free: 1: USB idle; 0: Busy, USB transfer in progress.	1
4	RB_UIF_FIFO_OV	RW	SOF timer interrupt flag in USB host mode, write 1 to clear: 1: SOF packet transmission completion trigger; 0: No event.	0
3	RB_UIF_HST_SOF	RW	USB bus suspend or wake-up event interrupt	0

			flag, write 1 to clear: 1: Triggered by USB suspend event or wake-up event; 0: No event.	
2	RB_UIF_SUSPEND	RW	USB bus suspend or wake-up event interrupt flag, write 1 to clear: 1: Triggered by USB suspend event or wake-up event; 0: No event.	0
1	RB_UIF_TRANSFER	RW	USB transfer completion interrupt flag, write 1 to clear: 1: Trigger when a USB transfer is completed; 0: No event.	0
0	RB_UIF_DETECT	RW	In USB host mode, USB device connect or disconnect event interrupt flag, write 1 to clear: 1: Detected USB device connection or disconnection trigger; 0: No event.	0
	RB_UIF_BUS_RST	RW	In USB device mode, USB bus reset event interrupt flag bit, write 1 to clear: 1: USB bus reset event trigger; 0: No event.	0

20.2.1.6 USB Interrupt Status Register (R8_USB_INT_ST)

Bit	Name	Access	Description	Reset value
7	RB_UIS_SETUP_ACT	RO	In USB device mode, 1 indicates that an 8-byte SETUP request packet has been successfully received. SETUP tokens do not affect RB_UIS_TOG_OK, MASK_UIS_TOKEN, MASK_UIS_ENDP, and RB_USB_RX_LEN.	0
6	RB_UIS_TOG_OK	RO	The current USB transport DATA0/1 synchronization flag matches the status bit, which is the same as RB_U_TOG_OK: 1: Synchronous; 0: Out of sync.	0
[5:4]	MASK_UIS_TOKEN	RO	Token PID identity of the current USB transport transaction in device mode.	x
[3:0]	MASK_UIS_ENDP	RO	The endpoint number of the current USB transport transaction in device mode.	x
	MASK_UIS_H_RES	RO	In host mode, the reply PID identification of the current USB transport transaction, 0000 indicates that the device does not answer or times out; other values indicate the reply PID.	x

MASK_UIS_TOKEN is used to identify the token PID of the current USB transfer transaction in the USB device mode: 00 means OUT packet; 01 reserved; 10 means IN packet; 11 means SETUP packet.

MASK_UIS_H_RES is only valid in host mode. In the host mode, if the host sends the OUT/SETUP token packet, the PID is the handshake packet ACK/NAK/STALL, or the device has no response/timeout. If the host sends an IN token packet, the PID is the PID of the data packet (DATA0/DATA1) or the PID of the handshake packet.

20.2.1.7 USB Receive Length Register (R16_USB_RX_LEN)

Bit	Name	Access	Description	Reset value
[15:10]	Reserved	RO	Reserved	0
[9:0]	R16_USB_RX_LEN	RO	The current number of data bytes received by the USB endpoint	x

20.2.2 Device Register Description

The USBFS module provides a total of 8 sets of bidirectional endpoints from endpoints 0-7 in USB device mode. The maximum packet length for all endpoints except endpoint 3 is 64 bytes, and the maximum packet length for endpoint 3 is 1023 bytes.

- Endpoint 0 is the default endpoint that supports controlling transmission, sending and receiving sharing a 64-byte data buffer.
- endpoints 1-7 each include a sending endpoint IN and a receiving endpoint OUT, each with an independent data buffer that supports batch transmission, interrupt transmission and real-time / synchronous transmission.
- Endpoint 0 has a separate DMA address, which is shared by both transceiver and transceiver. The sending and receiving of Endpoint 1-7 each has an DMA address. The mode of the data buffer can be set to double buffering or single buffering by UEPn_BUF_MOD. If dual-buffer mode is used, the endpoint can only be transmitted in one direction.
- Each set of endpoints has transceiver control registers R16_UEPn_CTRL and send length registers R16_UEPn_T_LEN and R32_UEPn_DMA (n=0~7), which are used to configure the synchronous trigger bit of the endpoint, the response to OUT and IN transactions, and the length of transmitted data, etc.

The USB bus pull-up resistor necessary for the USB device can be set by the software at any time. When the RB_UC_DEV_PU_EN in the USB control register R8_USB_CTRL is set to 1, the controller connects the pull-up resistor for the DP/DM pin of the USB bus internally according to the speed setting of the RB_UC_SPEED_TYPE, and enables the USB device function.

When a USB bus reset, a USB bus hang or wake-up event is detected, or when the USB successfully processes data transmission or data reception, the USB protocol processor will set the corresponding interrupt flag, and if the interrupt is enabled, a corresponding interrupt request will be generated. The application program can query directly or query and analyze the interrupt flag register R8_USB_INT_FG in the USB interrupt service program, and deal with it according to RB_UIF_BUS_RST and RB_UIF_SUSPEND. And, if the RB_UIF_TRANSFER is valid, then you need to continue to analyze the USB interrupt status register R8_USB_INT_ST and process it accordingly according to the current endpoint number MASK_UIS_ENDP and the current transaction token PID identification MASK_UIS_TOKEN. If the synchronization trigger bit RB_UEP_R_TOG of the OUT transaction of each endpoint is set in advance, then whether the synchronization trigger bit of the currently received data packet matches the synchronization trigger bit of the endpoint can be determined by RB_U_TOG_OK or RB_UIS_TOG_OK. If the data is synchronized, the data is valid; if the data is not synchronized, the data should be discarded. After each

interrupt of USB transmission or reception, the synchronization trigger bit of the corresponding endpoint should be modified correctly to detect whether the data packet sent or received next time is synchronized; in addition, setting RB_UEP_T_AUTO_TOG or RB_UEP_R_AUTO_TOG can automatically modify the corresponding synchronization trigger bit (flip or self-subtraction) after successful transmission or reception.

The data to be sent by each endpoint is in its own buffer, and the length of the data to be sent is independently set in the R16_UEPn_T_LEN; the data received by each endpoint is in the respective buffer, but the length of the received data is in the USB receiving length register R16_USB_RX_LEN, which can be distinguished according to the current endpoint number when the USB receives the interrupt.

Table 20-3 Device-related registers

Name	Access address	Description	Reset value
R8_UDEV_CTRL	0x50000001	USB device physical port control register	0xX0
R8_UEP4_1_MOD	0x5000000C	Endpoint 1/4 mode control register	0x00
R8_UEP2_3_MOD	0x5000000D	Endpoint 2/3 mode control register	0x00
R8_UEP5_6_MOD	0x5000000E	Endpoint 5/6 mode control register	0x00
R8_UEP7_MOD	0x5000000F	Endpoint 7 mode control register	0x00
R32_UEP0_DMA	0x50000010	Endpoint 0 buffer start address	0x0000XXXX
R32_UEP1_DMA	0x50000014	Endpoint 1 buffer start address	0x0000XXXX
R32_UEP2_DMA	0x50000018	Endpoint 2 buffer start address	0x0000XXXX
R32_UEP3_DMA	0x5000001C	Endpoint 3 buffer start address	0x0000XXXX
R32_UEP4_DMA	0x50000020	Endpoint 4 buffer start address	0x0000XXXX
R32_UEP5_DMA	0x50000024	Endpoint 5 buffer start address	0x0000XXXX
R32_UEP6_DMA	0x50000028	Endpoint 6 buffer start address	0x0000XXXX
R32_UEP7_DMA	0x5000002C	Endpoint 7 buffer start address	0x0000XXXX
R32_USB_EP0_CTRL	0x50000030	Endpoint 0 transmit length and control register	0x000000XX
R16_UEP0_T_LEN	0x50000030	Endpoint 0 transmit length register	0x00XX
R16_UEP0_CTRL	0x50000032	Endpoint 0 control register	0x0000
R32_USB_EP1_CTRL	0x50000034	Endpoint 1 transmit length and control register	0x000000XX
R16_UEP1_T_LEN	0x50000034	Endpoint 1 transmit length register	0x00XX
R16_UEP1_CTRL	0x50000036	Endpoint 1 control register	0x0000
R32_USB_EP2_CTRL	0x50000038	Endpoint 2 transmit length and control register	0x000000XX
R16_UEP2_T_LEN	0x50000038	Endpoint 2 transmit length register	0x00XX
R16_UEP2_CTRL	0x5000003A	Endpoint 2 control register	0x0000
R32_USB_EP3_CTRL	0x5000003C	Endpoint 3 transmit length and control register	0x000000XX
R16_UEP3_T_LEN	0x5000003C	Endpoint 3 transmit length register	0x00XX
R16_UEP3_CTRL	0x5000003E	Endpoint 3 control register	0x0000
R32_USB_EP4_CTRL	0x50000040	Endpoint 4 transmit length and control register	0x000000XX
R16_UEP4_T_LEN	0x50000040	Endpoint 4 transmit length register	0x00XX
R16_UEP4_CTRL	0x50000042	Endpoint 4 control register	0x0000
R32_USB_EP5_CTRL	0x50000044	Endpoint 5 transmit length and control register	0x000000XX
R16_UEP5_T_LEN	0x50000044	Endpoint 5 transmit length register	0x00XX
R16_UEP5_CTRL	0x50000046	Endpoint 5 control register	0x0000

R32_USB_EP6_CTRL	0x50000048	Endpoint 6 transmit length and control register	0x000000XX
R16_UEP6_T_LEN	0x50000048	Endpoint 6 transmit length register	0x00XX
R16_UEP6_CTRL	0x5000004A	Endpoint 6 control register	0x0000
R32_USB_EP7_CTRL	0x5000004C	Endpoint 7 transmit length and control register	0x000000XX
R16_UEP7_T_LEN	0x5000004C	Endpoint 7 transmit length register	0x00XX
R16_UEP7_CTRL	0x5000004E	Endpoint 7 control register	0x0000

20.2.2.1 USB Device Physical Port Control Register (R8_UDEV_CTRL)

Bit	Name	Access	Description	Reset value
7	RB_UD_PD_DIS	RW	USB device port UDP/UDM pin internal pull-down resistance control bit: 1: Disable internal drop-down. 0: Enable internal pull-down, can also be used in GPIO mode to provide pull-down resistors. *Note: the MODE and CNF and GPIOC_OUTDR settings of GPIOC_CFGXR have been changed to drop-down, this bit is reserved.*	1
6	Reserved	RO	Reserved	0
5	RB_UD_DP_PIN	RO	Current UDP pin status: 1: High level. 0: Low level.	X
4	RB_UD_DM_PIN	RO	Current UDM pin status: 1: High level. 0: Low level.	X
3	Reserved	RO	Reserved	0
2	RB_UD_LOW_SPEED	RW	USB device physical port low-speed mode enable bit: 1: Select 1.5Mbps low-speed mode. 0: Select 12Mbps full-speed mode.	0
1	RB_UD_GP_BIT	RW	USB device mode common flag bit, user-defined.	0
0	RB_UD_PORT_EN	RW	USB device physical port enable bit: 1: Enable physical port. 0: Disable physical port.	0

20.2.2.2 Endpoint 1/4 Mode Control Register (R8_UEP4_1_MOD)

Bit	Name	Access	Description	Reset value
7	RB_UEP1_RX_EN	RW	1: Enable endpoint 1 to receive (OUT). 0: Disable endpoint 1 to receive.	0
6	RB_UEP1_TX_EN	RW	1: Enable endpoint 1 to transmit (IN). 0: Disable endpoint 1 to transmit.	0
5	Reserved	RO	Reserved	0

4	RB_UEP1_BUF_MOD	RW	Endpoint 1 data buffer mode control bit.	0
3	RB_UEP4_RX_EN	RW	1: Enable endpoint 4 to receive (OUT). 0: Disable endpoint 4 to receive.	0
2	RB_UEP4_TX_EN	RW	1: Enable endpoint 4 to transmit (IN). 0: Disable endpoint 4 to transmit.	0
1	Reserved	RO	Reserved	0
0	RB_UEP4_BUF_MOD	RW	Endpoint 4 data buffer mode control bit.	0

20.2.2.3 Endpoint 2/3 Mode Control Register (R8_UEP2_3_MOD)

Bit	Name	Access	Description	Reset value
7	RB_UEP3_RX_EN	RW	1: Enable endpoint 3 to receive (OUT). 0: Disable endpoint 3 to receive.	0
6	RB_UEP3_TX_EN	RW	1: Enable endpoint 3 to transmit (IN). 0: Disable endpoint 3 to transmit.	0
5	Reserved	RO	Reserved	0
4	RB_UEP3_BUF_MOD	RW	Endpoint 3 data buffer mode control bit.	0
3	RB_UEP2_RX_EN	RW	1: Enable endpoint 2 to receive (OUT). 0: Disable endpoint 2 to receive.	0
2	RB_UEP2_TX_EN	RW	1: Enable endpoint 2 to transmit (IN). 0: Disable endpoint 2 to transmit.	0
1	Reserved	RO	Reserved	0
0	RB_UEP2_BUF_MOD	RW	Endpoint 2 data buffer mode control bit.	0

20.2.2.4 Endpoint 5/6 Mode Control Register (R8_UEP5_6_MOD)

Bit	Name	Access	Description	Reset value
7	RB_UEP6_RX_EN	RW	1: Enable endpoint 6 to receive (OUT). 0: Disable endpoint 6 to receive.	0
6	RB_UEP6_TX_EN	RW	1: Enable endpoint 6 to transmit (IN). 0: Disable endpoint 6 to transmit.	0
5	Reserved	RO	Reserved	0
4	RB_UEP6_BUF_MOD	RW	Endpoint 6 data buffer mode control bit.	0
3	RB_UEP5_RX_EN	RW	1: Enable endpoint 5 to receive (OUT). 0: Disable endpoint 5 to receive.	0
2	RB_UEP5_TX_EN	RW	1: Enable endpoint 5 to transmit (IN). 0: Disable endpoint 5 to transmit.	0
1	Reserved	RO	Reserved	0
0	RB_UEP5_BUF_MOD	RW	Endpoint 5 data buffer mode control bit.	0

20.2.2.5 Endpoint 7 Mode Control Register (R8_UEP7_MOD)

Bit	Name	Access	Description	Reset value
[7:4]	Reserved	RO	Reserved	0
3	RB_UEP7_RX_EN	RW	1: Enable endpoint 7 to receive (OUT). 0: Disable endpoint 7 to receive.	0

2	RB_UEP7_TX_EN	RW	1: Enable endpoint 7 to transmit (IN). 0: Disable endpoint 7 to transmit.	0
1	Reserved	RO	Reserved	0
0	RB_UEP7_BUF_MOD	RW	Endpoint 7 data buffer mode control bit.	0

The data buffer modes of USB endpoints 1-15 are configured by the combination of RB_UEPn_RX_EN and RB_UEPn_TX_EN and RB_UEPn_BUF_MOD, respectively, with specific reference to Table 20-4. Among them, in the double-64-byte buffer mode, the USB data transfer will select the first 64-byte buffer according to RB_UEP_*_TOG=0 In the dual 64-byte buffer mode, USB data transfer will select the back 64-byte buffer according to RB_UEP_*_TOG=1, and setting RB_UEP_AUTO_TOG=1 can realize automatic switching.

Table 20-4 Endpoint n buffer mode (n=1-7)

RB_UEPn_RX_EN	RB_UEPn_TX_EN	RB_UEPn_BUF_MOD	Description: Arranged from low to high with R16_UEPn_DMA as the starting address
0	0	X	The endpoint is disabled and the R16_UEPn_DMA buffer is not used.
1	0	0	Single 64-byte receive buffer (OUT).
1	0	1	Dual 64-byte receive buffer (OUT), selected by RB_UEP_R_TOG.
0	1	0	Single 64-byte transmit buffer (IN).
0	1	1	Dual 64-byte transmit buffer (IN) , selected by RB_UEP_T_TOG.
1	1	0	Single 64-byte receive buffer (OUT), single 64-byte transmit buffer (IN).
1	1	1	Dual 64-byte receive buffer (OUT), selected via RB_UEP_R_TOG. Dual 64-byte transmit buffer (IN), selected via RB_UEP_T_TOG. All 256 bytes are arranged as follows: UEPn_DMA+0 address: endpoint receive address with RB_UEP_R_TOG=0; UEPn_DMA+64 address: endpoint receive address when RB_UEP_R_TOG=1; UEPn_DMA+128 address: endpoint transmit address with RB_UEP_T_TOG=0; UEPn_DMA+192 address: endpoint transmit address when RB_UEP_T_TOG=1.

20.2.2.6 Endpoint n Buffer start address (R32_UEPn_DMA) (n=0-7)

Bit	Name	Access	Description	Reset value
[31:15]	Reserved	RO	Reserved	0
[14:0]	UEPn_DMA	RW	The starting address of the endpoint n buffer. The lower 15 bits are valid and the address must be 4 bytes aligned.	x

20.2.2.7 Endpoint n Transmit Length and Control Register (R32_USB_EPn_CTRL) (n=0-7)

Bit	Name
[31:16]	R16_UEPn_CTRL
[15:0]	R16_UEPn_T_LEN

20.2.2.8 Endpoint n transmit length register (R16_UEPn_T_LEN) (n=0-1)

Bit	Name	Access	Description	Reset value
[15:7]	Reserved	RO	Reserved	0

[6:0]	R8_UEPn_T_LEN	RW	Set the number of data bytes n=0, 1 that USB endpoint n is ready to send.	x

20.2.2.9 Endpoint n transmit length register (R16_UEPn_T_LEN) (n=2)

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved	0
7	HOST_PID3	RW	PID in host mode [3].	0
[6:0]	R8_UEPn_T_LEN	RW	Set the number of data bytes n=2 that USB endpoint n is ready to send.	x

20.2.2.10 Endpoint n transmit length register (R16_UEPn_T_LEN) (n=3)

Bit	Name	Access	Description	Reset value
[15:10]	Reserved	RO	Reserved	0
[9:0]	R8_UEPn_T_LEN	RW	Set the number of data bytes n=3 that USB endpoint n is ready to send.	x

20.2.2.11 Endpoint n transmit length register (R16_UEPn_T_LEN) (n=4-7)

Bit	Name	Access	Description	Reset value
[15:7]	Reserved	RO	Reserved	0
[6:0]	R8_UEPn_T_LEN	RW	Set the number of data bytes n=4, 5, 6, 7 that USB endpoint n is ready to send.	x

20.2.2.12 Endpoint n control register (R16_UEPn_CTRL) (n=0-7)

Bit	Name	Access	Description	Reset value
[15:12]	Reserved	RO	Reserved	0
11	RB_UEP_R_AUTO_TOG	RW	Synchronous trigger bit auto flip enable control bit: 1: Automatically flip the corresponding synchronous trigger bit after successful data reception. 0: Do not flip automatically, you can switch manually. *Note: this bit of endpoint 0 is reserved.*	0
10	MASK_UEP_R_TOG	RW	The desired synchronous trigger bit of the receiver of the USB endpoint n (processing OUT transactions): 1: Expected DATA1; 0: expected DATA0.	0
[9:8]	MASK_UEP_R_RES	RW	Response control of the receiver of endpoint n to the OUT transaction: 00: Reply ACK. 01: Timeout / no response for real-time / synchronous transmission of non-endpoint 0.	00b

			10: Answer NAK or busy. 11: Answer STALL or error.	
[7:4]	Reserved	RO	Reserved	0
3	RB_UEP_T_AUTO_TOG	RW	Synchronous trigger bit auto flip enable control bit: 1: Automatically flip the corresponding synchronous trigger bit after the data is sent successfully. 0: Do not flip automatically, you can switch manually. *Note: this bit of endpoint 0 is reserved.*	0
2	RB_UEP_T_TOG	RW	Synchronous trigger bits prepared by the sender of USB endpoint n (processing IN transactions): 1: Transmit DATA1;0: Transmit DATA0.	0
[1:0]	MASK_UEP_T_RES	RW	Response control of the sender of endpoint n to the IN transaction: 00: DATA0/DATA1 data is ready and expects ACK. 01: Reply DATA0/DATA1 and expect no response for real-time / synchronous transmission of non-endpoint 0. 10: Answer NAK or busy. 11: Answer STALL or error.	00b

20.2.3 USB Host Register

In USB host mode, the chip provides a set of bi-directional host endpoints, including a sending endpoint OUT and a receiving endpoint IN. The maximum length of a packet is 1024 bytes (synchronous transmission), supporting control transmission, interrupt transmission, batch transmission and real-time / synchronous transmission.

Every USB transaction initiated by the host endpoint automatically sets the RB_UIF_TRANSFER interrupt flag at the end of the processing. The application program can directly query or query and analyze the interrupt flag register R8_USB_INT_FG in the USB interrupt service program, and carry out corresponding processing according to each interrupt flag; moreover, if the RB_UIF_TRANSFER is valid, then it is necessary to continue to analyze the USB interrupt status register R8_USB_INT_ST and carry out corresponding processing according to the reply PID identification MASK_UIS_H_RES of the current USB transmission transaction.

If the synchronization trigger bit (RB_UH_R_TOG) of the IN transaction of the host receiving endpoint is set in advance, then whether the synchronization trigger bit of the currently received data packet matches the synchronization trigger bit of the host receiving endpoint can be determined by RB_U_TOG_OK or RB_UIS_TOG_OK. If the data is synchronized, the data is valid; if the data is not synchronized, the data should be discarded. After each interrupt of USB transmission or reception, the synchronization trigger bit of the corresponding host endpoint should be correctly modified to synchronize the next sent data packet and detect whether the next received data packet is synchronized; in addition, by setting RB_UH_T_AUTO_TOG and RB_UH_R_AUTO_TOG, the corresponding synchronization trigger bit can be flipped automatically after

successful transmission or reception.

The USB host token setting register R8_UH_EP_PID is used to set the terminal number of the target device to be operated and the token PID packet identification of the USB transmission transaction. The data corresponding to the SETUP token and the OUT token are provided by the host sending endpoint, the data to be sent is in the R16_UH_TX_DMA buffer zone, and the length of the data to be sent is set in the R16_UH_TX_LEN; the data corresponding to the IN token is returned by the target device to the host receiving endpoint, the received data is stored in the R16_UH_RX_DMA buffer zone, and the received data length is stored in the R16_USB_RX_LEN.

Table 20-5 Host-related registers

Name	Access address	Description	Reset value
R8_UHOST_CTRL	0x50000001	USB host physical port control register	0xX0
R8_UH_EP_MOD	0x5000000D	USB host endpoint mode control register	0x00
R16_UH_RX_DMA	0x50000018	USB host receive buffer start address	X
R16_UH_TX_DMA	0x5000001C	USB host transmit buffer start address	X
R16_UH_SETUP	0x50000036	USB host auxiliary setting register	0x0000
R8_UH_EP_PID	0x50000038	USB host token setting register	0x00
R8_UH_RX_CTRL	0x5000003B	USB host receive endpoint control register	0x00
R16_UH_TX_LEN	0x5000003C	USB host transmit length register	X
R8_UH_TX_CTRL	0x5000003E	USB host transmit endpoint control register	0x00

20.2.3.1 USB Host Physical Port Control Register (R8_UHOST_CTRL)

Bit	Name	Access	Description	Reset value
7	RB_UH_PD_DIS	RW	USB host port UD+/UD- pin internal 15K pull-down resistance control bit: 1: Disable internal drop-down. 0: Enable internal drop-down. Also can be used in GPIO mode to provide 15K pull-down resistors.	1
6	Reserved	RO	Reserved	0
5	RB_UH_DP_PIN	RO	Current UD+ pin status: 1: High; 0: Low.	x
4	RB_UH_DM_PIN	RO	Current UD- pin status: 1: High; 0: Low.	x
3	Reserved	RO	Reserved	0
2	RB_UH_LOW_SPEED	RW	USB host port low-speed mode enable bit: 1: Select 1.5Mbps low-speed mode. 0: Select 12Mbps full-speed mode.	0
1	RB_UH_BUS_RESET	RW	USB host mode bus reset control bit: 1: Force output USB bus reset. 0: End the output.	0
0	RB_UH_PORT_EN	RW	USB host port enable bit: 1: Enable host port. 0: Disable host port. This bit is automatically cleared 0 when the	0

			USB device is disconnected.	

20.2.3.2 USB Host Endpoint Mode Control Register (R8_UH_EP_MOD)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved	0
6	RB_UH_EP_TX_EN	RW	Host send endpoint transmit (SETUP/OUT) enable bit: 1: Enable endpoint to transmit. 0: Enable endpoint to transmit.	0
5	Reserved	RO	Reserved	0
4	RB_UH_EP_TBUF_MOD	RW	Host transmitting endpoint transmits data buffer mode control bits.	0
3	RB_UH_EP_RX_EN	RW	Host receive endpoint receive (IN) enable bit: 1: Enable endpoint to receive. 0: Disable endpoint to receive.	0
[2:1]	Reserved	RO	Reserved	00b
0	RB_UH_EP_RBUF_MOD	RW	Host receiving endpoint receives the data buffer mode control bits.	0

The host transmit endpoint data buffer mode is controlled by the combination of RB_UH_EP_TX_EN and RB_UH_EP_TBUF_MOD, refer to the following table.

Table 20-6 Host transmit buffer mode

RB_UH_EP_TX_EN	RB_UH_EP_TBUF_MOD	Description: start address with R16_UH_TX_DMA
0	X	Endpoints are disabled and the R16_UH_TX_DMA buffer is not used.
1	0	Single 64-byte transmit buffer (SETUP/OUT).
1	1	Dual 64-byte transmit buffer, selected by RB_UH_T_TOG: The first 64-byte buffer is selected when RB_UH_T_TOG=0; Back 64-byte buffer selected when RB_UH_T_TOG=1.

The host receives endpoint data buffer mode by the combination of RB_UH_EP_RX_EN and RB_UH_EP_RBUF_MOD, refer to the following table.

Table 20-7 Host receive buffer mode

RB_UH_EP_RX_EN	RB_UH_EP_RBUF_MOD	Description: start address with R16_UH_TX_DMA
0	X	Endpoints are disabled and the R16_UH_RX_DMA buffer is not used.
1	0	Single 64-byte receive buffer (IN).
1	1	Dual 64-byte receive buffer, selected by RB_UH_R_TOG: The first 64-byte buffer is selected when RB_UH_R_TOG=0; Back 64-byte buffer selected when RB_UH_R_TOG=1.

20.2.3.3 USB Host Receive Buffer Start Address (R16_UH_RX_DMA)

Bit	Name	Access	Description	Reset value
[15:0]	R16_UH_RX_DMA	RW	Host endpoint data receive buffer starting address. The lower 15 bits are valid and the address must be 4 bytes aligned.	XXXXh

20.2.3.4 USB Host Transmit Buffer Start Address (R16_UH_TX_DMA)

Bit	Name	Access	Description	Reset value
[15:0]	R16_UH_TX_DMA	RW	Host endpoint data transmit buffer starting address. The lower 15 bits are valid and the address must be 4 bytes aligned.	XXXXh

20.2.3.5 USB Host Auxiliary Setting Register (R16_UH_SETUP)

Bit	Name	Access	Description	Reset value
[15:11]	Reserved	RO	Reserved	0
10	RB_UH_PRE_PID_EN	RW	PREPID enable level of low-speed pilot package: 1: Enable, used to communicate with low-speed USB devices through external HUB. 0: Disable the low-speed pilot package.	0
[9:3]	Reserved	RO	Reserved	0
2	RB_UH_SOF_EN	RW	Automatically generate SOF packet enable bit: 1: The host automatically generates SOF packets. 0: It is not automatically generated, but can be generated manually.	0
[1:0]	Reserved	RO	Reserved	0

20.2.3.6 USB Host Token Setting Register (R8_UH_EP_PID)

Bit	Name	Access	Description	Reset value
[7:4]	MASK_UH_TOKEN	RW	Set the token PID package identity for this USB transport transaction.	0000b
[3:0]	MASK_UH_ENDP	RW	Set the endpoint number of the target device to be operated this time.	0000b

20.2.3.7 USB Host Receive Endpoint Control Register (R8_UH_RX_CTRL)

Bit	Name	Access	Description	Reset value
[7:4]	Reserved	RO	Reserved	0

3	RB_UH_R_AUTO_TOG	RW	Synchronous trigger bit auto flip enable control bit: 1: Automatically flip the corresponding expected synchronous trigger bit (RB_UH_R_TOG) after successful data reception. 0: Do not flip automatically, you can switch manually.	0
2	RB_UH_R_TOG	RW	The synchronization trigger bits expected by the USB host receiver (processing IN transactions): 1: Expect DATA1. 0: Expect DATA0.	0
1	Reserved	RO	Reserved	0
0	RB_UH_R_RES	RW	Host receiver response control bits for IN transactions: 1: No response for real-time / synchronous transmission of non-zero endpoints. 0: Reply ACK.	0

20.2.3.8 USB Host Transmit Length Register (R16_UH_TX_LEN)

Bit	Name	Access	Description	Reset value
[15:0]	R8_UH_TX_LEN	RW	Sets the number of bytes of data that the USB host transmitting endpoint is ready to transmit.	XXh

20.2.3.9 USB Host Transmit Endpoint Control Register (R8_UH_TX_CTRL)

Bit	Name	Access	Description	Reset value
[7:4]	Reserved	RO	Reserved	0
3	RB_UH_T_AUTO_TOG	RW	Synchronous trigger bit auto flip enable control bit: 1: Automatically flip the corresponding synchronous trigger bit (RB_UH_T_TOG) after the data is sent successfully. 0: Do not flip automatically, you can switch manually.	0
2	RB_UH_T_TOG	RW	Synchronous trigger bits prepared by the USB host transmitter (processing SETUP/OUT transactions): 1: Transit DATA1. 0: Transit DATA0.	0
1	Reserved	RO	Reserved	0
0	RB_UH_T_RES	RW	The response control bits of the USB host transmitter to the SETUP/OUT transaction:	0

			1: No response is expected for real-time / synchronous transmission of non-zero endpoints. 0: Expect to reply ACK.	

Chapter 21 USB PD Controller (USBPD)

21.1 USB PD Controller Introduction

The chip has built-in USB Power Delivery controller and PD transceiver PHY, supports USB Type-C master-slave detection, automatic BMC codec and CRC, hardware edge control, supports USB PD2.0 and PD3.0 power transmission control, supports fast charging, supports UFP/DFP/DRD/DRP and PDUSB, and supports PD power receiving and PD power supply applications.

- Built-in USB Type-C interface, support master-slave detection, support DRP, Sink/Consumer and Source/Provider.
- Built-in USB PD transceiver PHY, integrated hardware edge slope control.
- Built-in USB Power Delivery controller, automatic BMC codec, 4b5b codec and CRC.
- Support PD packages such as SOP, SOP', SOP '', and supports hardware reset of USB PD reset signal frames.
- Support a maximum packet length of 510 bytes and supports DMA.
- Support USB PD 2.0 and 3.0 power transmission protocols, and USB ports support charging protocols such as BC.

21.2 Register Description

Table 21-1 USBPD-related registers

Name	Access address	Description	Reset value
R32_USBPD_CONFIG	0x40027000	PD configuration register	0x00000X02
R16_CONFIG	0x40027000	PD interrupt enable register	0x0X02
R16_BMC_CLK_CNT	0x40027002	BMC sampling clock counter	0x0000
R32_USBPD_CONTROL	0x40027004	PD control register	0x00000000
R16_CONTROL	0x40027004	PD transceiver control register	0x0000
R8_CONTROL	0x40027004	PD transceiver enable register	0x00
R8_TX_SEL	0x40027005	PD transmit SOP selection register	0x00
R16_BMC_TX_SZ	0x40027006	PD transmit length register	0x0000
R32_USBPD_STATUS	0x40027008	PD status register	0x000000XX
R16_STATUS	0x40027008	PD interrupt and data register	0x00XX
R8_DATA_BUF	0x40027008	DMA cache data register	0xXX
R8_STATUS	0x40027009	PD interrupt flag register	0x00
R16_BMC_BYTE_CNT	0x4002700A	Byte counter	0x0000
R32_USBPD_PORT	0x4002700C	Port control register	0x00030003
R16_PORT_CC1	0x4002700C	CC1 port control register	0x0003
R16_PORT_CC2	0x4002700E	CC2 port control register	0x0003
R32_USBPD_DMA	0x40027010	DMA cache address register	0x0000XXXX
R16_DMA	0x40027010	PD buffer start address register	0xXXXX

21.2.1 PD Configuration Register (R16_CONFIG)

Offset address: 0x00

Bit	Name

[31:16]	R16_BMC_CLK_CNT
[15:0]	R16_CONFIG

21.2.2 PD Interrupt Enable Register (R16_CONFIG)

Offset address: 0x00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IE_TX_END	IE_RX_RESET	IE_RX_ACT	IE_RX_BYTE	IE_RX_BYTE	IE_PD_IO	RTX_BIT0	MULTI_0	Reserved		WAKE_POLAR	PD_RST_EN	PD_DMA_EN	CC_SEL	PD_ALL_CLR	Reserved

Bit	Name	Access	Description	Reset value
15	IE_TX_END	RW	End-of-transmit interrupt enable.	0
14	IE_RX_RESET	RW	Receive reset interrupt enable.	0
13	IE_RX_ACT	RW	Receive completion interrupt enable.	0
12	IE_RX_BYTE	RW	Receive byte interrupt enable.	0
11	IE_RX_BIT	RW	Receive bit interrupt enable.	0
10	IE_PD_IO	RW	PD IO interrupt enable.	0
9	RTX_BIT0	RO	Current value of send/receive shift register bit 0: 0: The value of bit 0 at the current time is 1. 1: The value of bit 0 at the current time is 0.	x
8	MULTI_0	RO	Multiple consecutive bits of the 0 indication signal have been received. This bit is 1, indicating that 5 consecutive bits of the 0 value have been received.	0
[7:6]	Reserved	RO	Reserved	0
5	WAKE_POLAR	RW	PD port wake-up level: 0: Low level effective. 1: High level is effective.	0
4	PD_RST_EN	RW	The PD mode reset command enable: 0: Invalid. 1: Reset.	0
3	PD_DMA_EN	RW	Enable the DMA of USBPD, which must be set to 1 in normal transfer mode: 0: Disable DMA. 1: Enable DMA function and DMA interrupt.	0
2	CC_SEL	RW	Select the current PD communication port: 0: Use CC1 port to communicate. 1: Use CC2 port to communicate.	0
1	PD_ALL_CLR	RW	PD mode clears all interrupt flag bits: 0: Invalid. 1: Clear the interrupt flag bit.	1
0	Reserved	RO	Reserved	0

21.2.3 BMC Sampling Clock Counter (R16_BMC_CLK_CNT)

Offset address: 0x02

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

Bit	Name	Access	Description	Reset value
[15:9]	Reserved	RO	Reserved	0
[8:0]	BMC_CLK_CNT	RW	BMC transmits or receives sampling clock counters.	0

21.2.4 PD Control Register (R32_USBPD_CONTROL)

Offset address: 0x04

Bit	Name
[31:16]	R16_BMC_TX_SZ
[15:0]	R16_CONTROL

21.2.5 PD Transceiver Control Register (R16_CONTROL)

Offset address: 0x04

Bit	Name
[15:8]	R8_TX_SEL
[7:0]	R8_CONTROL

21.2.6 PD Transceiver Enable Register (R8_CONTROL)

Offset address: 0x04

7	6	5	4	3	2	1	0
RX_ST_H	RX_ST_L	DATA_FLAG	Reserved			BMC_START	PD_TX_EN

Bit	Name	Access	Description	Reset value
7	RX_ST_H	RO	The receiving status is high.	0
6	RX_ST_L	RO	The receiving status is low.	0
5	DATA_FLAG	RO	Valid flag bits for cached data.	0
[4:2]	Reserved	RO	Reserved	0
1	BMC_START	RW	BMC transmits a start signal.	0
0	PD_TX_EN	RW	USBPD transceiver mode and transmit enable: 0: PD receive enable. 1: PD transmit enable.	0

21.2.7 PD Transmit SOP Selection Register (R8_TX_SEL)

Offset address: 0x05

7	6	5	4	3	2	1	0

TX_SEL4	TX_SEL3	TX_SEL2	Reserved	TX_SEL1

Bit	Name	Access	Description	Reset value
[7:6]	TX_SEL4	RW	Select the K-CODE4 type in PD transmitting mode: 00: SYNC2; 01: SYNC3; 1x: RST2.	0
[5:4]	TX_SEL3	RW	Select the K-CODE3 type in PD transmitting mode: 00: SYNC1; 01: SYNC3; 1x: RST1.	0
[3:2]	TX_SEL2	RW	Select the K-CODE2 type in PD transmitting mode: 00: SYNC1; 01: SYNC3; 1x: RST1.	0
1	Reserved	RO	Reserved	0
0	TX_SEL1	RW	Select the K-CODE1 type in PD transmitting mode: 0: SYNC1; 1: RST1.	0

21.2.8 PD Transmit Length Register (R16_BMC_TX_SZ)

Offset address: 0x06

Bit	Name	Access	Description	Reset value
[15:9]	Reserved	RO	Reserved	0
[8:0]	BMC_TX_SZ	RW	The total length transmitted in PD mode.	0

21.2.9 PD Status Register (R32_USBPD_STATUS)

Offset address: 0x08

Bit	Name
[31:16]	R16_BMC_BYTE_CNT
[15:0]	R16_STATUS

21.2.10 PD Interrupt and Data Register (R16_STATUS)

Offset address: 0x08

Bit	Name
[15:8]	R8_STATUS
[7:0]	R8_DATA_BUF

21.2.11 DMA Cache Data Register (R8_DATA_BUF)

Offset address: 0x08

7	6	5	4	3	2	1	0
DATA_BUF							

Bit	Name	Access	Description	Reset value
[7:0]	DATA_BUF	RO	DMA cache data	X

21.2.12 PD Interrupt Flag Register (R8_STATUS)

Offset address: 0x09

7	6	5	4	3	2	1	0
IF_TX_END	IF_RX_RESET	IF_RX_ACT	IF_RX_BYTE	IF_RX_BIT	BUF_ERR	BMC_AUX	

Bit	Name	Access	Description	Reset value
7	IF_TX_END	RW1	Transfer completed interrupt flag, write 1 clear 0, write 0 invalid.	0
6	IF_RX_RESET	RW1	Receive reset interrupt flag, write 1 clear 0, write 0 invalid.	0
5	IF_RX_ACT	RW1	Transfer completed interrupt flag, write 1 clear 0, write 0 invalid.	0
4	IF_RX_BYTE	RW1	Receive byte or SOP interrupt flag, write 1 clear 0, write 0 is invalid.	0
3	IF_RX_BIT	RW1	Receive bit or 5bit interrupt flag, write 1 clear 0, write 0 invalid.	0
2	BUF_ERR	RW1	BUFFER or DMA error interrupt flag, write 1 clear 0, write 0 invalid.	0
[1:0]	BMC_AUX	RO	Indicates the current PD status: When the PD is received or after the reception is completed, the status is as follows: 00: Receiving idle or no valid packet received; 01: SOP received i.e. SOP0; 10: SOP' received i.e. SOP1 or Hard Reset; 11: SOP' received i.e. SOP2 or Cable Reset. When PD is transmitting, the status is as follows: 00: CRC32[7:0] is being transmitted; 01: CRC32[15:8] is being transmitted; 10: CRC32[23:16] is being transmitted; 11: CRC32[31:24] is being transmitted.	00

21.2.13 Byte Counter (R16_BMC_BYTE_CNT)

Offset address: 0x0A

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

Bit	Name	Access	Description	Reset value
[15:9]	Reserved	RO	Reserved	0

[8:0]	BMC_BYTE_CNT	RO	Byte counter.	0

21.2.14 Port Control Register (R32_USBPD_PORT)

Offset address: 0x0C

Bit	Name
[15:8]	R16_PORT_CC2
[7:0]	R16_PORT_CC1

21.2.15 CC1 Port Control Register (R16_PORT_CC1)

Offset address: 0x0C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CC1_CE			CC1_LVE	CC1_PU		CC1_PD	PA_CC1_AI

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved	0
[7:5]	CC1_CE	RW	Enable the CC1 port voltage comparator: 000: Off. 001: Reserved 010: 0.22V; 011: 0.43V; 100: 0.55V; 101: 0.66V; 110: 0.96V; 111: 1.23V.	000
4	CC1_LVE	RW	The CC1 port outputs low voltage enable: 0: Normal VDD voltage drive output. 1: Low voltage drive output.	0
[3:2]	CC1_PU	RW	CC1 port pull-up current selection: 00: No pull-up current 01: 330μA; 10: 180μA; 11: 80μA. *Note: (1) The pull-up current of the CC port is independent of the GPIO and can be controlled separately.* *(2) Chips with a CHIPID penultimate digit of 1 have the port configured as a pull-up input when used as a Source.*	00
1	CC1_PD	RW	CC1 port pull-down resistor Rd enable: 0: Disable the pull-down resistor (Note: there is still about 800kΩ weak pull-down after closing). 1: Enable Rd pull-down resistor, about 5.1K Ω. *Note: The pull-down resistor of CC port is*	1

			independent of GPIO and can be controlled separately, but some package forms of the chip do not have built-in Rd, refer to the selection table in CH32L103DS0.PDF.	
0	PA_CC1_AI	RO	The CC1 port comparator simulates input.	1

21.2.16 CC2 Port Control Register (R16_PORT_CC2)

Offset address: 0x0E

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CC2_CE			CC2_LVE	CC2_PU		CC2_PD	PA_CC2_AI

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved	0
[7:5]	CC2_CE	RW	Enable the CC2 port voltage comparator: 000: Off. 001: Reserved 010: 0.22V; 011: 0.43V; 100: 0.55V; 101: 0.66V; 110: 0.96V; 111: 1.23V.	000
4	CC2_LVE	RW	The CC2 port outputs low voltage enable: 0: Normal VDD voltage drive output. 1: Low voltage drive output.	0
[3:2]	CC2_PU	RW	CC2 port pull-up current selection: 00: No pull-up current 01: 330μA; 10: 180μA; 11: 80μA. *Note: (1) The pull-up current of the CC port is independent of the GPIO and can be controlled separately.* *(2) Chips with a CHIPID penultimate digit of 1 have the port configured as a pull-up input when used as a Source.*	00
1	CC2_PD	RW	CC2 port pull-down resistor Rd enable: 0: Disable pull-down resistor (note: about 800kΩ weak pull-down remains after disabling); 1: Enable Rd pull-down resistor, about 5.1KΩ. *Note: The CC port pull-down resistor is independent of the GPIO and can be controlled separately, but some package forms of the chip*	1

			do not have built-in Rd, specifically refer to the selection table in CH32L103DS0.PDF	
0	PA_CC2_AI	RO	The CC2 port comparator Analog input.	1

21.2.17 DMA Cache Address Register (R32_USBPD_DMA)

Offset address: 0x10

Bit	Name
[31:15]	Reserved
[15:0]	R16_DMA

21.2.18 PD Buffer Start Address Register (R16_DMA)

Offset address: 0x10

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

Bit	Name	Access	Description	Reset value
[15:0]	USBPD_DMA_ADDR	RW	USBPD_DMA cache address. The low 16-bit is valid and the address must be 4-byte aligned.	X

Chapter 22 Controller Area Network (CAN)

Controller Area Network is a high-performance communication protocol for serial data communication. The CAN controller provides a complete implementation of the CAN protocol, supporting CAN protocols 2.0A and 2.0B. The CAN controller can be used to construct a powerful local area network to realize safe distributed real-time control, process a large number of data packets with a small CPU load, and has a wide range of applications in the industrial and automotive fields.

22.1 Main Features

- Compatible with CAN specification 2.0A and 2.0B
- Programmable transfer rate up to 8Mbit/s
- Support time-triggered communication function to avoid low-priority message blocking
- Support 3 sending mailboxes, the priority of sending messages can be determined by the message identifier or the order of sending requests, and can record the time stamp of the SOF moment of sending the message
- 2 receiving FIFO,14 packet filter groups supporting 3-level mailbox depth can be configured, each filter group can be configured in 32 or 16-bit mode, shielded bit or identifier list mode, which can minimize the interference of software in message filtering. FIFO overflow processing is flexible, and can record the time stamp of receiving message SOF.
- Occupies 4 interrupt vectors, each interrupt source can be configured independently

22.2 CAN Controller Operating Mode

The CAN controller can operate the SLEEP or INRQ bits in the register CAN_CTLR to switch between the initialization mode, sleep mode and normal mode.

22.2.1 Initialization Mode

After reset, CAN works in sleep mode by default to reduce power consumption. At this time, the sending and receiving of messages is prohibited, the internal pull-up resistor of the TX pin is enabled, and the TX pin outputs a recessive bit. The INRQ bit in the register CAN_CTLR is set to 1, and the CAN controller is requested to enter the initialization mode. When the INAK bit of the register CAN_STATR is automatically set to 1, the initialization state is successfully entered. Similarly, clear the INRQ bit in the register CAN_CTLR to request the CAN controller to exit the initialization mode. When the INAK bit of the register CAN_STATR is automatically cleared to 0, the initialization state is successfully exited.

The filter group can be initialized in the non-initialization mode, but the FINIT bit of the register CAN_FCTLR must be set to 1, and the reception of the message is prohibited at this time.

24.2.2 Sleep Mode

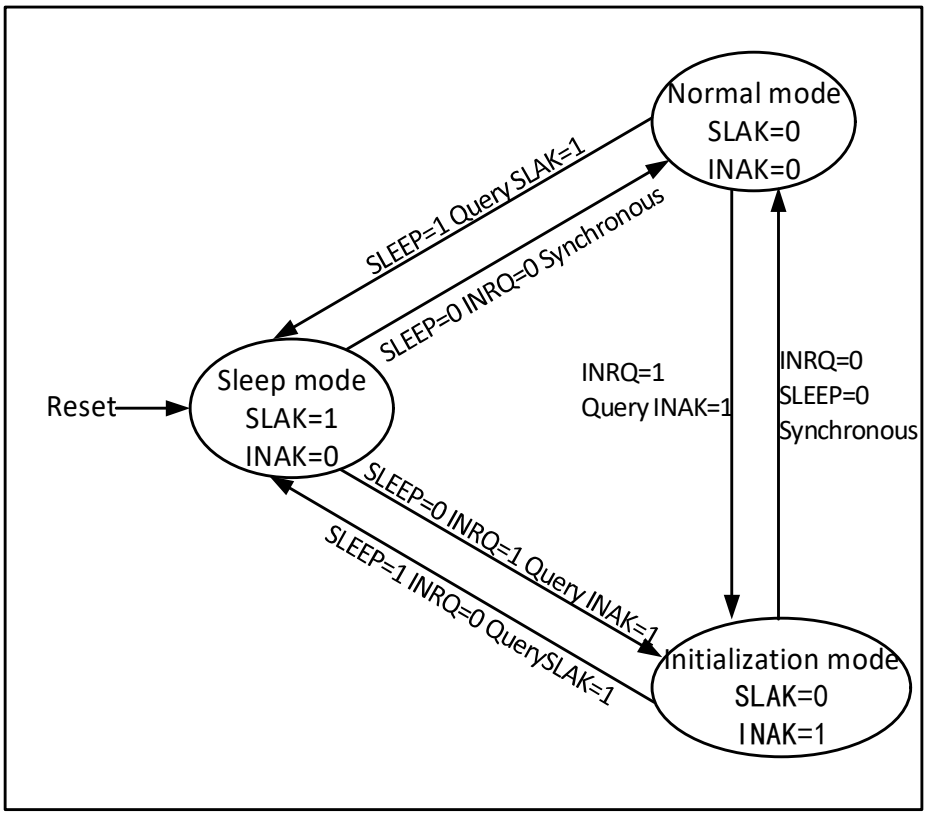
For the SLEEP position 1 in the register CAN_CTLR, the CAN controller is requested to enter the sleep mode. When the SNAK bit of the register CAN_STATR is automatically set to 1, the CAN successfully enters the sleep mode, and the clock of the CAN controller stops, but the mailbox register can still be accessed.

From sleep mode to initialization mode, the SLEEP bit of CAN_CTLR must be cleared to position 1, and when the

INAK bit of register CAN_STATR is automatically set to 1, it will be switched to initialization state.

To enter the normal mode from sleep mode, the SLEEP bit of CAN_CTLR must be cleared 0, and when the SNAK bit of register CAN_STATR is automatically cleared 0, it will enter normal mode.

Figure 22-1 CAN operating mode switch



22.3 CAN Controller Test Mode

In the initialization mode, operate the SILM and LBKM bits of the register CAN_BTIMR to select a test mode, and then exit the initialization mode and enter the test mode by clearing the INRQ bit of the register CAN_CTLR. There are 3 test modes: silent mode, loopback mode and silent loopback mode.

22.3.1 Silent Mode

Setting the SILM bit in register CAN_BTIMR to 1 can optionally enter silent mode. In this mode, the CAN controller can receive, but cannot send messages to the outside world. It is always in a recessive bit to the outside world, which can avoid affecting the bus, but the message can be received by the controller of the node where it is located. Usually, silent mode is used for the status analysis of the CAN bus.

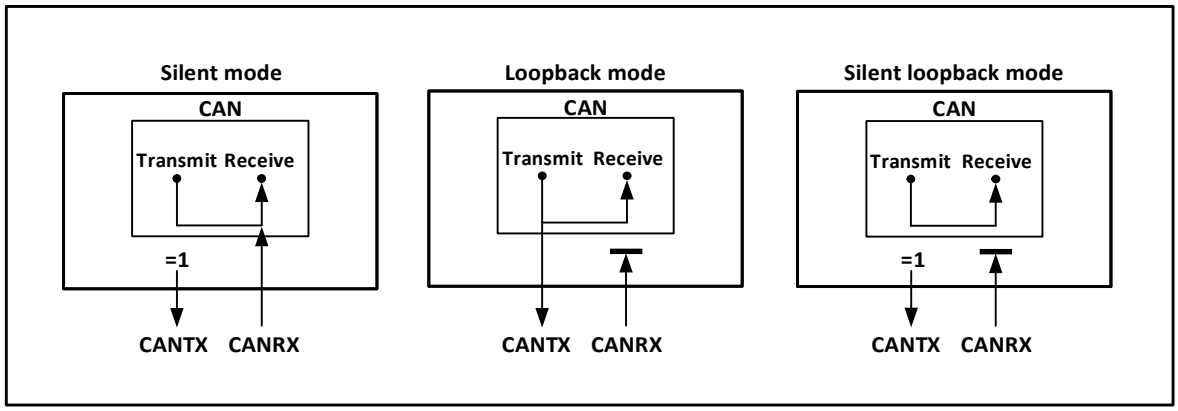
22.3.2 Loopback Mode

By setting the LBKM bit of the register CAN_BTIMR to 1, the loopback mode can be selected. In this mode, the CAN controller can send external messages, but cannot receive external messages, but the sent messages can be received by the controller of the node where it is located, and the reception filtering mechanism is effective. Usually, loopback mode is used for transceiver testing of CAN controllers.

22.3.3 Silent Loopback Mode

Setting the SILM and LBKM bits in register CAN_BTIMR to 1 can optionally enter silent loopback mode. This mode is usually used for the closed self-test of the CAN controller. In this mode, it has no effect on the CAN bus, the RX pin is disconnected from the bus, and the TX pin is set to a recessive bit.

Figure 22-2 CAN 3 test modes of the bus



22.4 MCU Operating State of CAN Controller in Debug Mode

When the MCU enters the debug mode, the core is in a suspended state, but it can be determined whether the CAN controller is in a normal operation or a stop state through the configuration bits in the debug module.

22.5 CAN Controller Functional Description

22.5.1 Transmit Processing Flow

The transmit processing flow is as follows: If there are vacant mailboxes among the 3 sending mailboxes, the application layer software only has write access to the registers of the vacant mailboxes, and operates the registers CAN_TXMIRx, CAN_TXMDTRx, CAN_TXMDLRx and CAN_TXMDHRx, and can set the message identifier, message length, time stamp, and message data. After the data is ready, the TXRQ bit of the register CAN_TXMIRx is set to 1 to request transmission, the mailbox enters the registered state, and the priority is queued; once it becomes the highest priority mailbox, it becomes the scheduled transmission state and waits for the CAN bus to be idle; when the CAN bus is idle When the message is scheduled to be sent, the message of the mailbox will enter the sending state immediately; after the message is sent, the mailbox will become a vacant mailbox again, and the RQCP and TXOK bits of the register CAN_TSTATR are set to 1 to indicate that the sending is successful; if the arbitration fails during sending, the ALST of the register CAN_TSTATR Set to 1, TERR set to 1 if an error is sent.

22.5.2 Transmit Priority

The transmit priority can be determined by the identifier or the order of sending requests. The TXFP position of the register CAN_CTLR is 1 and sent according to the order of sending requests. According to the order of sending requests, it is mainly used for segmented sending; the TXFP bit is cleared to 0 and sent according to the priority of the identifier. In order, the smaller the identifier is, the higher the priority is. In the case of the same identifier, the mailbox with the lower number has a higher priority.

22.5.3 Transmit Abort Processing

If the ABRQ bit in the register CAN_TSTATR is set, the transmission request can be aborted. When the mailbox status is registered or scheduled to send, the sending request is directly aborted; when the mailbox is in the sending state, the abort request may succeed (stop sending) or fail (sending complete), and the result can be queried by the TXOK bit in the CAN_TSTATR register.

22.5.4 Time-based Trigger Mode

When the traditional CAN communication bus is busy, it is easy to cause low-priority messages to be blocked for a long time, and even cannot meet the requirements of its time limit. In order to solve the bottleneck, related protocols based on time-triggered mode have been introduced. Such protocols have a certain scale of application in the industry, and the functions based on time-triggered mode are the application of such protocols.

There are 2 modes to choose from in the time-triggered mode. To use this mode, the automatic retransmission function needs to be turned off. The default mode and enhanced mode are selected by configuring the MODE bit of the CAN_TTCTLR register. Set the TTCM and NART bits of the register CAN_CTLR to 1, enable the time-triggered mode and disable automatic retransmission. The MODE bit of the CAN_TTCTLR register defaults to 0. At this time, it works in the default mode, and the internal timer is activated to generate timestamps of the transmit and receive mailboxes. The timer accumulates at the CAN bit time, and the internal timer is sampled and generates a timestamp at the sample point position of the received and transmitted frame start bits. If the enhanced mode is used, the MODE bit of the CAN_TTCTLR register needs to be set to 1 to enable the enhanced mode. Using this mode, there must be 3 or more nodes in the entire CAN network, one of which sends the time reference, and the other nodes receive the time stamp of the reference node, they reset the internal counter by writing 1 to the TIMRST bit of the CAN_TTCTLR register to synchronize the internal counter. So that in addition to the node that sends the time reference, the rest of the CAN nodes achieve time synchronization. Afterwards, write the data to be sent to the sending mailbox, configure the time trigger count value (TIMCNT of the CAN_TTCNT register) and the internal counter count end value (TIMCMV of the CAN_TTCTLR register) of each node in turn. The time-triggered count value and the final count value of the internal counter are determined by the CAN nodes, the CAN communication rate and the number of data bits in a frame. After the configuration is completed, each node waits for the internal counter to count to the time-triggered count value, and then triggers the transmitting action.

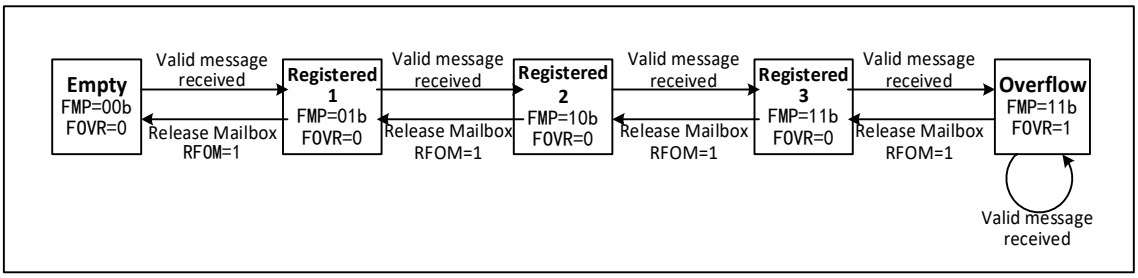
22.5.5 Receive Processing Flow

The reception of CAN bus messages is completed by the controller hardware without the intervention of the MCU, which reduces the processing load of the MCU. The received messages are stored in 2 FIFOs with 3-level mailbox depth according to the setting of the register CAN_FAFIFOR. If the application layer needs to obtain the message, it can only read the valid received message through the receiving FIFO mailbox.

Initially, the receiving FIFO is empty, and the value of FMR[1:0] in the receiving FIFO register CAN_RFIFOx is binary 00b. After receiving a valid receiving message, it becomes the registered 1 state, and the controller automatically sets the FMR[1:0] in the receiving FIFO register CAN_RFIFOx to binary 01b; if the mailbox data registers CAN_RXMDLRx and CAN_RXMDHRx are read at this time, the mailbox is released by setting the RFOM bit of the receiving FIFO register CAN_RFIFOx to 1, and the receiving FIFO state becomes empty again; if the mailbox is not released in the registered 1 state, after the next valid receiving message is received, the receiving FIFO state switches to the registered 2 state; at this time, the FMR[1:0] of the receiving FIFO register CAN_RFIFOx is automatically set to binary 10b. If the mailbox data register is read and the mailbox is released, then the state returns to registered 1; if the mailbox is not released in the registered 2 state, the receiving FIFO enters the registered

3 state; also in the registered 3 state, the message is read and the mailbox is released, then the registered 2 state is returned; if the registered 3 state is not If the mailbox is released, when the next valid packet is received, packet loss will inevitably occur.

Figure 22-3 Receive FIFO state switching diagram



In the case of message loss above, that is, the receiving FIFO is full, and the message overflow causes the message to be lost. The FOVR bit of the receiving FIFO register CAN_RFIFOx will be automatically set to 1 by hardware for overflow query. When the RFLM bit of the register CAN_CTLR is set to 1, the receiving FIFO locking function is enabled, and the discarded message is a new received message; when the RFLM bit of the register CAN_CTLR is cleared to 0, the receiving FIFO locking function is disabled. Among the 3 original messages of the receiving FIFO, the last received message will be overwritten by the new message.

When the relevant bit of the register CAN_INTENR is set, an interrupt can be generated when the state of the receiving FIFO is switched, so as to process the received message more efficiently, see section 22.6 CAN interrupt for details.

22.5.6 Receive Message Identifier Filtering

There are up to 14 filter groups in the module. By setting the filter group, each CAN node can receive the packets that meet the filtering rules, and the packets that do not meet the filtering rules are discarded by hardware without software intervention.

Each filter bank consists of 2 32-bit registers CAN_FxR0 and CAN_FxR1. The bit width of the filter group can be independently configured as a 32-bit filter or 2 16-bit filters by setting each bit of the register CAN_FSCFGR. Each filter group can be configured as mask bit or identifier list mode by setting each bit of register CAN_FMCFGR, and each filter group can be enabled or disabled by setting each bit of register CAN_FWR. Setting each bit of the register CAN_FAFIFOR can select which receive FIFO the message that passes the filter is stored in.

As shown in Table 22-1 below, in the mask bit mode, the 2 registers are the identifier register and the mask register, which need to be used together. Each bit of the identifier register indicates that the expected value of the corresponding bit is dominant or recessive. Each bit of the register indicates whether the corresponding bit needs to be consistent with the expected value of the corresponding identifier register bit.

Table 22-1 32-bit mask bit mode

Identifier register	CAN_FxR1[31:24]	CAN_FxR1[23:16]		CAN_FxR1[15:8]	CAN_FxR1[7:0]			
Mask bit register	CAN_FxR2[31:24]	CAN_FxR2[23:16]		CAN_FxR2[15:8]	CAN_FxR2[7:0]			
Map	STID[10:3]	STID[2:0]	EXID[17:13]	EXID[12:5]	EXID[4:0]	IDE	RTR	0

In the identifier list mode, both registers are used as identifier registers, and each bit of the received message identifier must be consistent with one of the registers to pass the filter.

Table 22-2 32-bit identifier list mode

Identifier register	CAN_FxR1[31:24]	CAN_FxR1[23:16]		CAN_FxR1[15:8]	CAN_FxR1[7:0]			
Mask bit register	CAN_FxR2[31:24]	CAN_FxR2[23:16]		CAN_FxR2[15:8]	CAN_FxR2[7:0]			
Map	STID[10:3]	STID[2:0]	EXID[17:13]	EXID[12:5]	EXID[4:0]	IDE	RTR	0

In 16-bit mode, the register group is divided into 4 registers, and the mask bit mode of each group of filters can have 2 filters, each of which contains a 16-bit identifier register and a 16-bit mask register; all 4 registers are used as identifier registers in identifier list mode.

Table 22-3 16-bit mask bit mode

Identifier register n	CAN_FxR1[15:8]	CAN_FxR1[7:0]			
Mask bit register n	CAN_FxR1[31:24]	CAN_FxR1[23:16]			
Identifier register n+1	CAN_FxR2[15:8]	CAN_FxR2[7:0]			
Mask bit register n+1	CAN_FxR2[31:24]	CAN_FxR2[23:16]			
Map	STID[10:3]	STID[2:0]	RTR	IDE	EXID[17:15]

Table 22-4 16-bit identifier list mode

Identifier register n	CAN_FxR1[15:8]	CAN_FxR1[7:0]			
Mask bit register n	CAN_FxR1[31:24]	CAN_FxR1[23:16]			
Identifier register n+1	CAN_FxR2[15:8]	CAN_FxR2[7:0]			
Mask bit register n+1	CAN_FxR2[31:24]	CAN_FxR2[23:16]			
Map	STID[10:3]	STID[2:0]	RTR	IDE	EXID[17:15]

When the message enters the FIFO mailbox, it will be read and stored by the application program. Usually, the application program distinguishes the message data according to the message identifier. The CAN controller provides a filter number for the messages filtered by different filters in the receiving FIFO, and the number is stored in FMI[7:0] of the CAN_RXMDTRx register, regardless of whether the filter group is enabled or not. The numbering scheme is detailed in the example in Figure 22-4.

When a packet can be filtered by multiple filters, the filter number stored in the receiving mailbox determines which filter number is stored according to the filter priority rules. The filter priority rules are as follows:

- All 32-bit filters have higher priority than 16-bit filters
- For filters of the same width, the filter of the identifier list has higher priority than the filter of the mask bit pattern
- Filters with the same width and pattern, filters with smaller numbers have higher priority

As shown in Figure 22-5: When receiving a message, the identifier is first matched and filtered with the 32-bit identifier list pattern filter. If there is no match, the 32-bit masked bit pattern filter will be matched and filtered. If there is no match, the 16-bit identifier list pattern filter will be matched and filtered. If there is no match, the 16-bit masked bit pattern filter will be matched and filtered. Finally, if there is no match, the message will be discarded. If there is a match, the message will be stored in the mailbox of the receiving FIFO. The identifier number is stored in the FMI in the CAN_RXMDTRx register.

Figure 22-4 Example of filter number

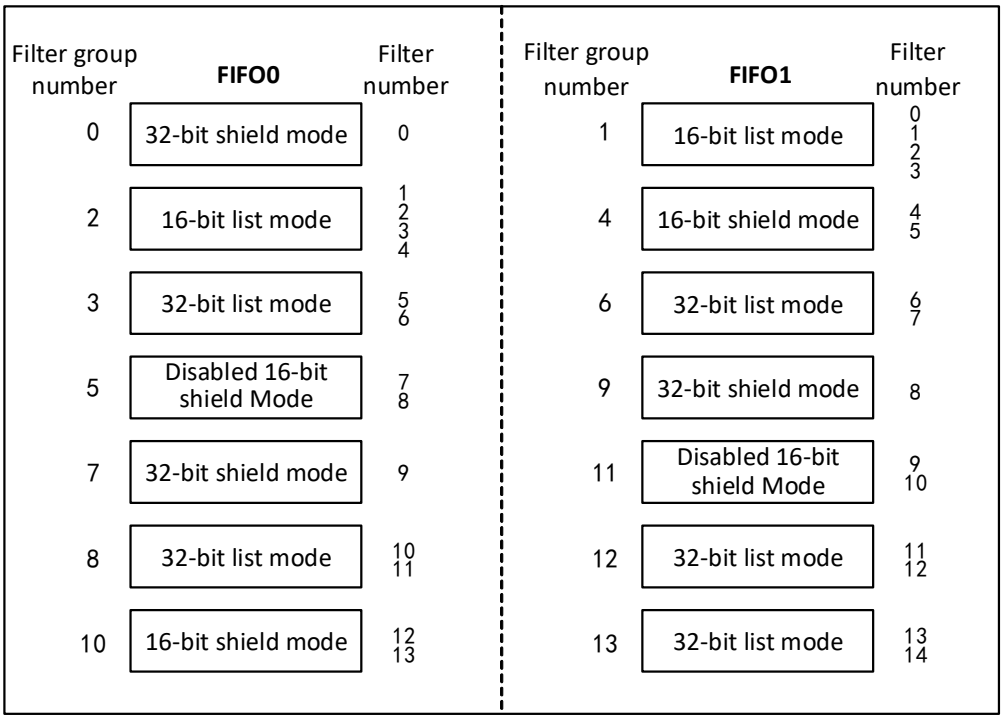
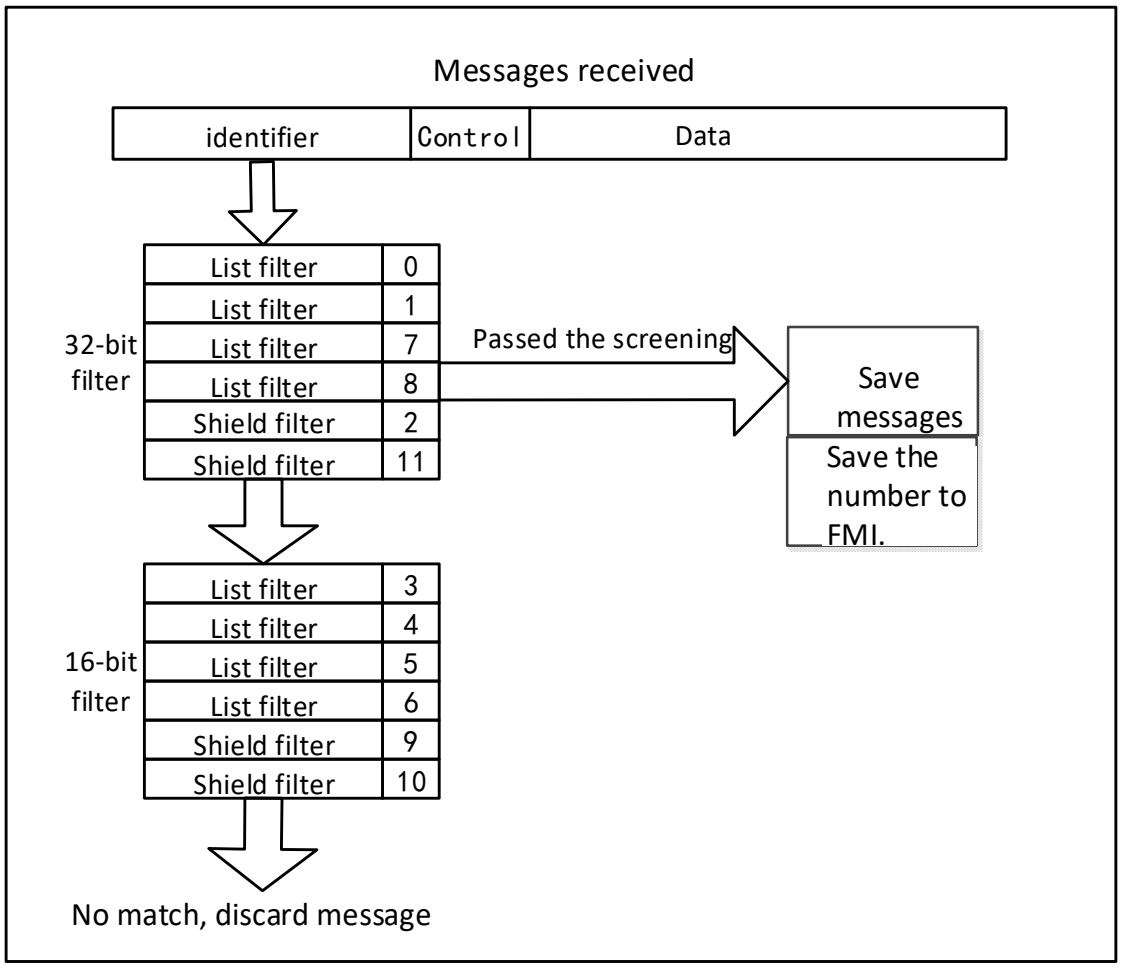


Figure 22-5 Filter example



24.5.7 Error Handling

The CAN controller relies on the state error register CAN_ERRSR to manage errors on the bus. The TEC and REC in the state error register CAN_ERRSR represent the sending and receiving error count values respectively. According to the increase of the sending and receiving errors and the decrease of the sending and receiving success, the stability of the CAN bus can be judged according to their values.

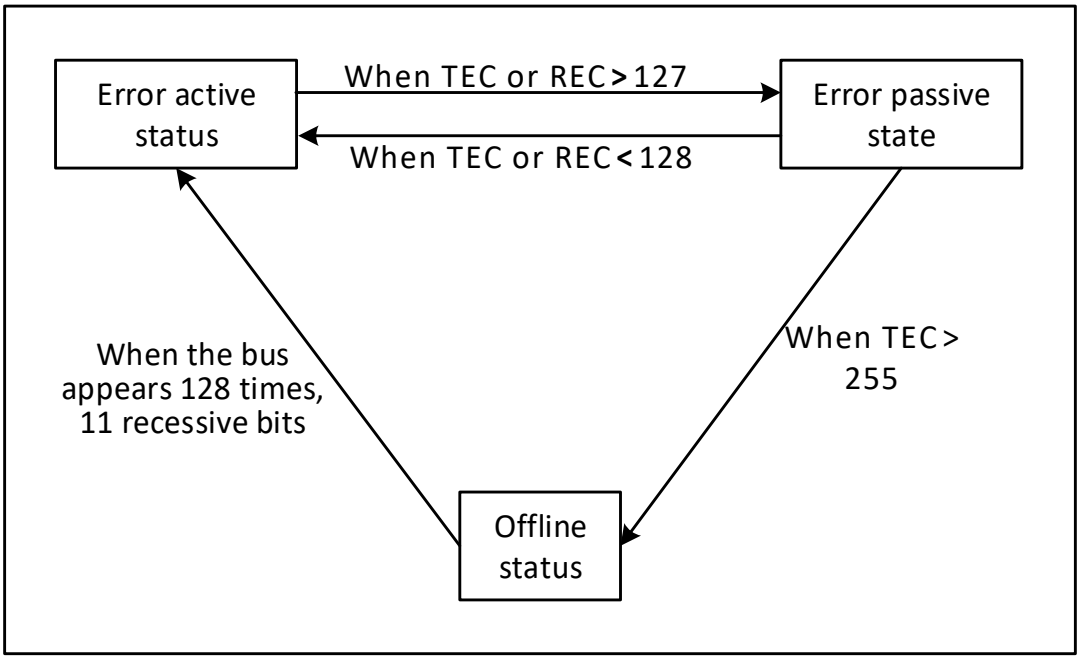
When the TEC and REC in the state error register CAN_ERRSR are less than 128, the current CAN node is in an error active state, can normally participate in bus communication, and issue an active error flag when an error is detected.

When the TEC and REC in the state error register CAN_ERRSR are greater than 127, the current CAN node is in an error passive state, and when an error is detected, it is not allowed to issue an active error flag, but can only issue a passive error flag.

When the TEC in the status error register CAN_ERRSR is greater than 255, the current CAN node goes offline.

When the bus monitors 11 consecutive hidden bits for 128times, it returns to the error active state, and the recovery mode is affected by the ABOM bit in the main control register CAN_CTLR. If ABOM is set to 1, the hardware automatically exits the offline state. If ABOM is 0, the software needs to operate the INRQ bit to enter the initialization mode, and then exit the initialization before exiting the offline state.

Figure 22-6 CAN error state switch diagram



22.5.8 Bit Timing

According to the standard of CAN bus, each bit time is divided into four segments: synchronization segment, propagation time segment, phase buffer segment 1 and phase buffer segment 2. These segments consist of the minimum time unit Tq. The CAN controller monitors the CAN bus change by sampling and synchronizes through the edge of the start bit of the frame.

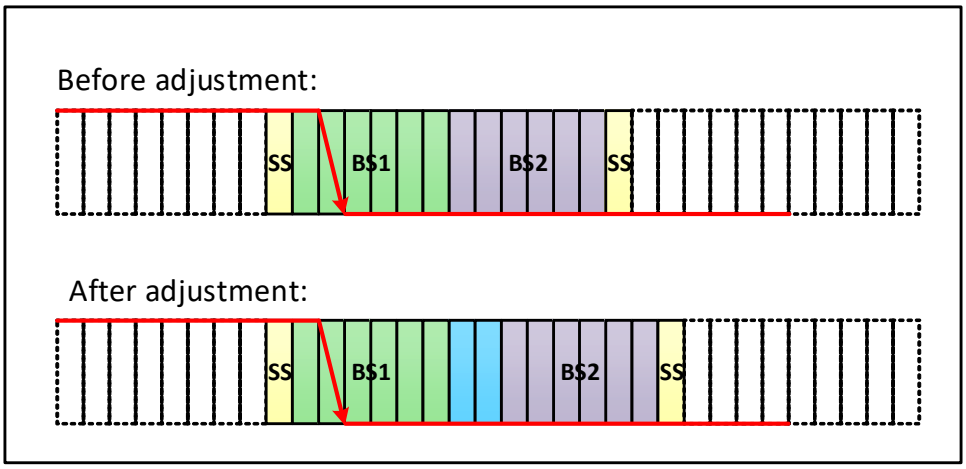
The CAN controller re-divides the above four segments into three segments, which are:

- Synchronization segment (SS): the synchronization segment in the CAN standard is fixed as a minimum time unit, and the expected bit jump normally occurs within this period of time.
- Time period 1 (BS1): contains the propagation time period and phase buffer section 1 in the CAN standard, which can be set to contain 1 to 16 minimum time units and can be automatically extended to compensate for the positive phase drift caused by frequency accuracy errors of different nodes on the CAN bus. The end of the time period is the location of the sampling point.
- Time period 2 (BS2): phase buffer section 2 in the CAN standard can be set to 1 to 8 minimum time units and can be automatically shortened to compensate for the negative phase drift caused by frequency accuracy errors of different nodes on the CAN bus.

Resynchronization jump width (SJW) is the upper limit of the minimum number of time units that can be extended and reduced per person, and the range can be set to 1 to 4 minimum time units.

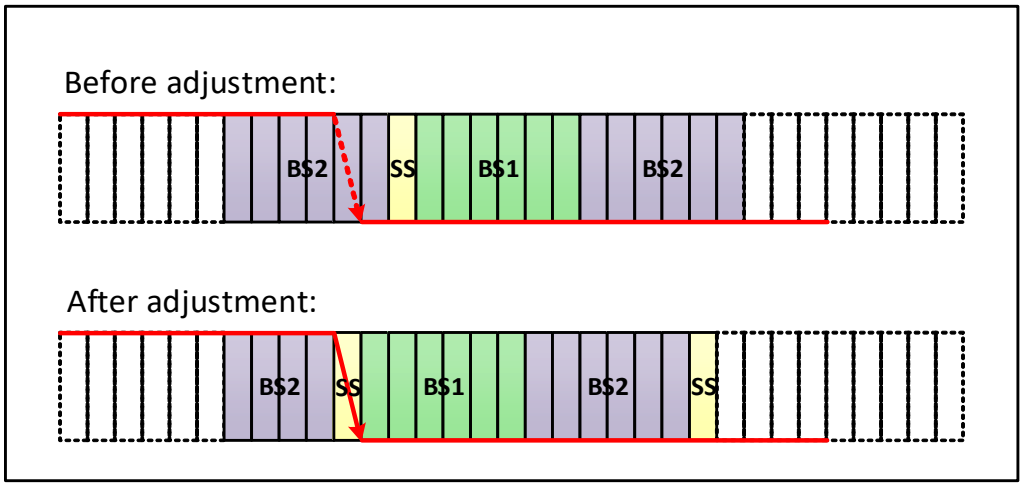
The above parameters can be configured in the CAN bus timing register CAN_BTIMR.

Figure 22-7 The jump appears in BS1



If the SJW of figure 22-7 is 2, and the bus level jump is detected in time period 1, the length of time period 1 needs to be extended, with a maximum extension of SJW, thus delaying the position of the sampling point.

Figure 22-8 The jump appears in BS2



If the SJW of figure 22-8 is 2 and the bus level jump is detected in time period 2, it is necessary to reduce the length of time period 2 and the maximum SJW, so as to advance the position of the sampling point.

22.6 CAN Interrupt

The CAN controller has four interrupt vectors, which are send interrupt, FIFO_0 interrupt, FIFO_1 interrupt, error and state change interrupt.

Set the CAN interrupt allow register CAN_INTENR to allow or disable each interrupt source.

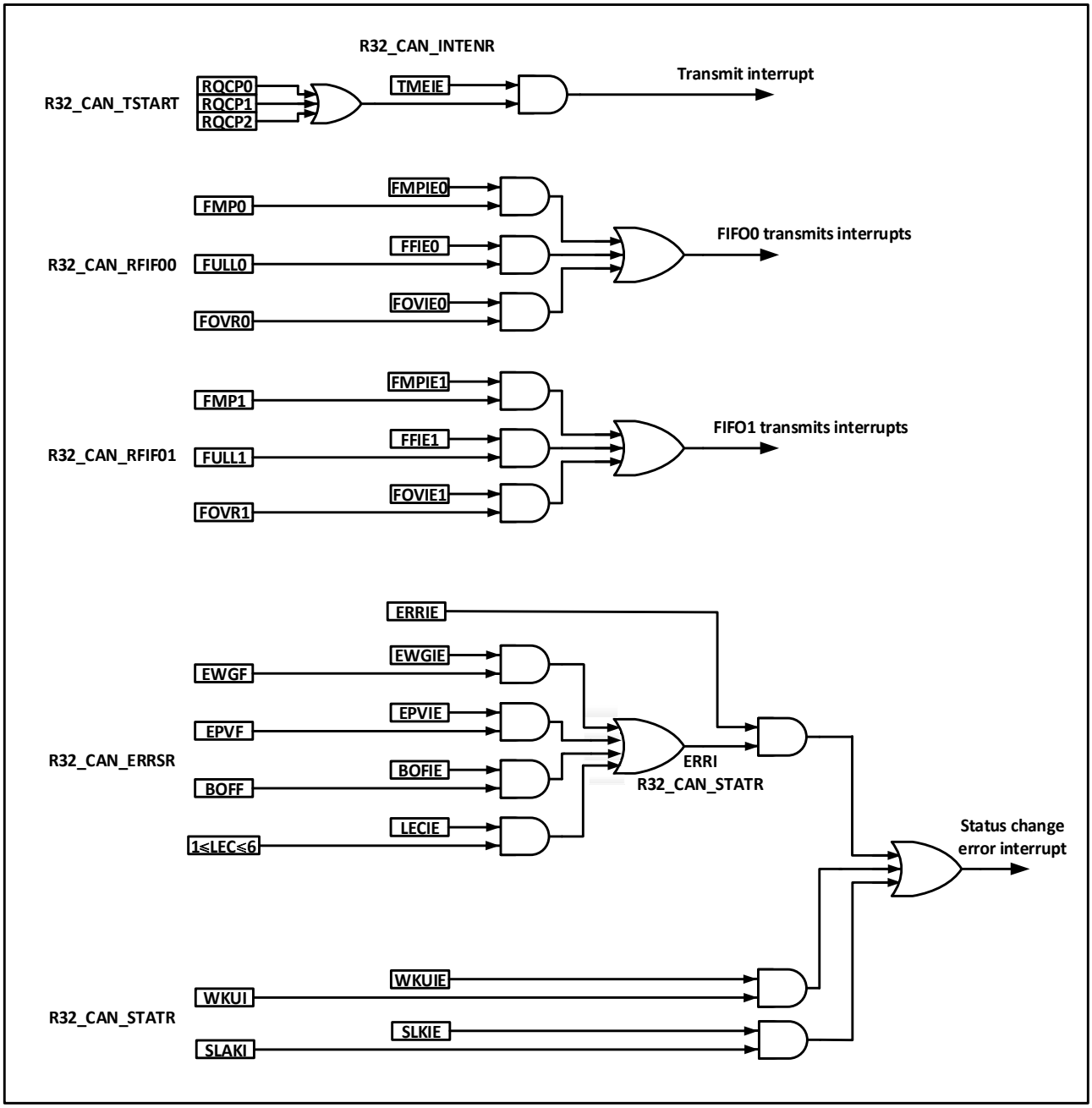
The sending interrupt is caused by the sending mailbox empty event. After the interrupt is generated, the RQCP0, RQCP1 and RQCP2 bits of the register CAN_TSTATR are queried to determine which mailbox empty event is generated.

FIFO0 interrupts are caused by receiving new messages, receiving mailbox fullness and overflow events. After the interrupts are generated, the FMP0, FULL0 and FOVER0 bits of register CAN_RFIFO0 are queried to determine which mailbox emptiness event is generated.

FIFO1 interrupts are caused by receiving new messages, receiving mailbox fullness and overflow events. After the interrupts are generated, the FMP1, FULL1 and FOVER1 bits of register CAN_RFIFO1 are queried to determine which mailbox emptiness event is generated.

Errors and state change interruptions are caused by errors, arousal, and sleep events.

Figure 22-9 CAN interrupt logic diagram



22.7 CAN FD Function Description

22.7.1 FD Frame Operation

1) Transmitting FD frames:

Just configure the [7:0] bit of the CANFD_CR register as 0x0F, fill the CANFD_DMA_T0/1/2 buffer of the

corresponding mailbox into the sending data, and configure the DMA address, you can send out the frame in FD format, and other registers can be configured as needed.
2) Receiving FD frames:
Simply configure the CANFD_DMA_R0/1 address corresponding to the receiving FIFO and configure the CANFD_BTR register to the correct bit rate to receive the FD frame.

In the FDCAN format, the coding of DLC differs from the standard CAN format. The coding of DLC codes 0 through 8 is the same as that of standard CAN, while the coding of codes 9 through 15 (which all encode an 8-byte data field in standard CAN) is different from that of standard CAN, as shown in the table.

Table 22-5 DLC Coding in CAN FD Mode

DLC	9	10	11	12	13	14	15
Data byte count	12	16	20	24	32	48	64

Note: the FIFO depth received by the FD frame is 1, that is, every time the data is received to the DMA_R0/1, the software must take the data packet in the RAM in time, otherwise the data will be overwritten.

22.7.2 Transmitter Delay Compensation

CAN FD provides a sender delay timing filtering function. When using this function, it is necessary to set the TDC_FILTER bit of the CANFD_TDCT register to define the minimum delay of the internal delay counter and prevent the dominant bit burr on the RX from causing the delay counter to end prematurely.

The offset of the transmitter delay can be compensated by the TDCO bit, and the size is FD_TDCO+1. The CANFD_PSR register TDCV [23:16] bit is read-only and returns the actual value; the actual value of FD transmission delay compensation is the value of the internal delay counter plus FD_TDCO.

22.8 Register Description

The registers related to the CAN controller must be manipulated in 32-bit words. In order to avoid the influence of the current node on the entire CAN bus, the application software can only modify the bit timing register CAN_BTIMR in the initialization mode.

Table 22-6 CAN-related registers

Name	Access address	Description	Reset value
R32_CAN_CTLR	0x40006400	CAN main control register	0x00010002
R32_CAN_STATR	0x40006404	CAN main status register	0x00000C02
R32_CAN_TSTATR	0x40006408	CAN transmit status register	0x1C000000
R32_CAN_RFIFO0	0x4000640C	CAN receive FIFO0 control and status registers	0x00000000
R32_CAN_RFIFO1	0x40006410	CAN receive FIFO1 control and status registers	0x00000000
R32_CAN_INTENR	0x40006414	CAN interrupt enable register	0x00000000
R32_CAN_ERRSR	0x40006418	CAN error status register	0x00000000
R32_CAN_BTIMR	0x4000641C	CAN bit timing register	0x01230000
R32_CAN_TTCTLR	0x40006420	CAN time trigger control register	0x0000FFFF
R32_CAN_TTCNT	0x40006424	CAN time trigger count value register	0x00000000

R32_CAN_TERR_CNT	0x40006428	CAN offline recovery error counter	0x00000000
R32_CANFD_CR	0x4000642C	CANFD control register	0x0000000E
R32_CANFD_BTR	0x40006430	CANFD timing register	0x60800637
R32_CANFD_TDCT	0x40006434	CANFD transmit delay compensation register	0x00000002
R32_CANFD_PSR	0x40006438	CANFD transmit delay compensation value register	0x00000000
R32_CANFD_DMA_T0	0x4000643C	CANFD DMA transmit mailbox 0 cache register	0x00000000
R32_CANFD_DMA_T1	0x40006440	CANFD DMA transmit mailbox 1 cache register	0x00000000
R32_CANFD_DMA_T2	0x40006444	CANFD DMA transmit mailbox 2 cache register	0x00000000
R32_CANFD_DMA_R0	0x40006448	CANFD DMA receive mailbox 0 cache register	0x00000000
R32_CANFD_DMA_R1	0x4000644C	CANFD DMA receive mailbox 1 cache register	0x00000000

Table 22-7 CAN mailbox-related registers

Name	Access address	Description	Reset value
R32_CAN_TXMIR0	0x40006580	CAN Tx mailbox 0 identifier register	X
R32_CAN_TXMDTR0	0x40006584	CAN Tx mailbox 0 data length and timestamp register	X
R32_CAN_TXMDLR0	0x40006588	CAN Tx mailbox 0 data low register	X
R32_CAN_TXMDHR0	0x4000658C	CAN Tx mailbox 0 data high register	X
R32_CAN_TXMIR1	0x40006590	CAN Tx mailbox 1 identifier register	X
R32_CAN_TXMDTR1	0x40006594	CAN Tx mailbox 1 data length and timestamp register	X
R32_CAN_TXMDLR1	0x40006598	CAN Tx mailbox 1 data low register	X
R32_CAN_TXMDHR1	0x4000659C	CAN Tx mailbox 1 data high register	X
R32_CAN_TXMIR2	0x400065A0	CAN Tx mailbox 2 identifier register	X
R32_CAN_TXMDTR2	0x400065A4	CAN Tx mailbox 2 data length and timestamp register	X
R32_CAN_TXMDLR2	0x400065A8	CAN Tx mailbox 2 data low register	X
R32_CAN_TXMDHR2	0x400065AC	CAN Tx mailbox 2 data high register	X
R32_CAN_RXMIR0	0x400065B0	CAN Rx FIFO 0 mailbox identifier register	X
R32_CAN_RXMDTR0	0x400065B4	CAN Rx FIFO 0 mailbox data length and timestamp register	X
R32_CAN_RXMDLR0	0x400065B8	CAN Rx FIFO 0 mailbox data low register	X
R32_CAN_RXMDHR0	0x400065BC	CAN Rx FIFO 0 mailbox data high register	X
R32_CAN_RXMIR1	0x400065C0	CAN Rx FIFO1 mailbox identifier register	X
R32_CAN_RXMDTR1	0x400065C4	CAN Rx FIFO1 mailbox data length and timestamp register	X
R32_CAN_RXMDLR1	0x400065C8	CAN Rx FIFO1 mailbox data low register	X
R32_CAN_RXMDHR1	0x400065CC	CAN Rx FIFO1 mailbox data high register	X

Table 22-8 CAN Filter registers

Name	Access address	Description	Reset value
R32_CAN_FCTLR	0x40006600	CAN filter main control register	0x2A1C0E01
R32_CAN_FMCFGR	0x40006604	CAN filter mode register	0x00000000
R32_CAN_FSCFGR	0x4000660C	CAN filter bit width register	0x00000000
R32_CAN_FAFIFOR	0x40006614	CAN filter FIFO association register	0x00000000
R32_CAN_FWR	0x4000661C	CAN filter activation register	0x00000000
R32_CAN_F0R1	0x40006640	CAN filter group 0 register 1	X

R32_CAN_F0R2	0x40006644	CAN filter group 0 register 2	X
R32_CAN_F1R1	0x40006648	CAN filter group 1 register 1	X
R32_CAN_F1R2	0x4000664C	CAN filter group 1 register 2	X
R32_CAN_F2R1	0x40006650	CAN filter group 2 register 1	X
R32_CAN_F2R2	0x40006654	CAN filter group 2 register 2	X
R32_CAN_F3R1	0x40006658	CAN filter group 3 register 1	X
R32_CAN_F3R2	0x4000665C	CAN filter group 3 register 2	X
R32_CAN_F4R1	0x40006660	CAN filter group 4 register 1	X
R32_CAN_F4R2	0x40006664	CAN filter group 4 register 2	X
R32_CAN_F5R1	0x40006668	CAN filter group 5 register 1	X
R32_CAN_F5R2	0x4000666C	CAN filter group 5 register 2	X
R32_CAN_F6R1	0x40006670	CAN filter group 6 register 1	X
R32_CAN_F6R2	0x40006674	CAN filter group 6 register 2	X
R32_CAN_F7R1	0x40006678	CAN filter group 7 register 1	X
R32_CAN_F7R2	0x4000667C	CAN filter group 7 register 2	X
R32_CAN_F8R1	0x40006680	CAN filter group 8 register 1	X
R32_CAN_F8R2	0x40006684	CAN filter group 8 register 2	X
R32_CAN_F9R1	0x40006688	CAN filter group 9 register 1	X
R32_CAN_F9R2	0x4000668C	CAN filter group 9 register 2	X
R32_CAN_F10R1	0x40006690	CAN filter group 10 register 1	X
R32_CAN_F10R2	0x40006694	CAN filter group 10 register 2	X
R32_CAN_F11R1	0x40006698	CAN filter group 11 register 1	X
R32_CAN_F11R2	0x4000669C	CAN filter group 11 register 2	X
R32_CAN_F12R1	0x400066A0	CAN filter group 12 register 1	X
R32_CAN_F12R2	0x400066A4	CAN filter group 12 register 2	X
R32_CAN_F13R1	0x400066A8	CAN filter group 13 register 1	X
R32_CAN_F13R2	0x400066AC	CAN filter group 13 register 2	X

22.8.1 CAN Main Control Register (CAN_CTLR)

Offset address: 0x00

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RST	Reserved							TTCM	ABOM	AWUM	NART	RFLM	TXFP	SLEEP	INRQ

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	DBF	RW	Debug whether to disable CAN bus from working 1: During debugging, the CAN transceiver is prohibited, but the control and read and write operations of the receiving FIFO are normal; 0: When debugging, the CAN controller works	1

			normally.	
15	RST	RW1	CAN controller software reset request. Writing 0 to this bit is invalid 1: Reset the CAN controller. After reset, the controller enters sleep mode, and then the hardware automatically clears it to 0; 0: The CAN controller is in normal state.	0
[14:8]	Reserved	RO	Reserved	0
7	TTCM	RW	Whether to allow time trigger mode 1: Enable time trigger mode; 0: Disable time trigger mode. The time trigger mode is mainly used with the TTCAN protocol.	0
6	ABOM	RW	Offline automatic exit control 1: The hardware detects 11 consecutive implicit bits 128 times and automatically exits the offline state; 0: The INRQ bit of the software operation register CAN_CTLR is required to be set to 1 and then cleared to 0. After 11 consecutive implicit bits are detected 128 times, the offline state is exited.	0
5	AWUM	RW	CAN controller automatic wake-up enable 1: When a message is detected, the hardware automatically wakes up, and the SLEEP and SLAK bits of the register CAN_STATR are automatically cleared to 0; 0: Software operation is required to clear the SLEEP bit of the CAN_CTLR register to wake up the CAN controller.	0
4	NART	RW	The automatic packet retransmission function is disabled 1: No matter whether the transmission is successful or not, the message can only be sent once; 0: The CAN controller keeps retransmitting until the transmission is successful.	0
3	RFLM	RW	Receive FIFO message lock mode enable. 1: When the received FIFO overflows, the received mailbox message is not read, and the mailbox is not released, the newly received message is discarded. 0: When the received FIFO overflows, the received mailbox message is not read, and when the mailbox is not released, the newly received message will overwrite the original message. *Note: this bit is only used for traditional CAN.*	0
2	TXFP	RW	Transmit mailbox priority method selection	0

			1: The priority is determined by the order in which the requests are sent; 0: The priority is determined by the message identifier.	
1	SLEEP	RW	Sleep mode request 1: Set to 1 to request the CAN controller to enter sleep mode. After the current activity is completed, the controller enters sleep mode. If the AWUM bit is set to 1, the controller clears the SLEEP bit to 0 when a message is received; 0: After the software clears to 0, the controller exits the sleep mode.	1
0	INRQ	RW	Initialize mode request 1: Set to 1 to request the CAN controller to enter the initialization mode. After the current activity is completed, the controller enters the initialization mode, and the hardware sets the INAK bit of the register CAN_STATR to 1; 0: Set to 0 to request the CAN controller to exit the initialization mode and enter the normal mode, and the hardware clears the INAK bit of the CAN_STATR register to 0.	0

22.8.2 CAN Main Status Register (CAN_STATR)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				RX	SAMP	RXM	TXM	Reserved			SLAKI	WKUI	ERRI	SLAK	INAK

Bit	Name	Access	Description	Reset value
[31:12]	Reserved	RO	Reserved	0
11	RX	RO	Current actual level of CAN controller Rx pin.	1
10	SAMP	RO	Level of a receive bit on the CAN controller RX pin	1
9	RXM	RO	Rx mode query 1: CAN controller is in Rx mode currently; 0: CAN controller is not in Rx mode currently.	0
8	TXM	RO	Tx mode query 1: CAN controller is in Tx mode currently; 0: CAN controller is not in Tx mode currently.	0
[7:5]	Reserved	RO	Reserved	0

4	SLAKI	RW1	Sleep interrupt enable, that is, interrupt generation flag when the SLKIE bit in the CAN_INTENR register is set to 1, write 1 to clear it, and writing 0 is invalid. 1: When entering sleep mode, an interrupt is generated and the hardware is set to 1; 0: When exiting sleep mode, it can be cleared by hardware or by software.	0
3	WKUI	RW1	Wake-up interrupt flag. When the WKUI bit in the CAN_INTENR register is set to 1, if the SOF bit is detected when the CAN controller is in sleep mode, the hardware will set it to 1. Set to 1 by software to clear to 0, and set to 0 is invalid.	0
2	ERRI	RW1	Error interrupt. When the ERRIE bit in the CAN_INTENR register is set to 1, an error and status change interrupt is generated. This bit is set to 1 and cleared to 0 by software, and set to 0 is invalid.	0
1	SLAK	RO	Sleep mode indication. 1: CAN controller is in sleep mode; 0: CAN controller is not in sleep mode.	1
0	INAK	RO	Initialization mode indication. 1: CAN controller is in initialization mode; 0: CAN controller is not in initialization mode.	0

22.8.3 CAN Transmit Status Register (CAN_TSTATR)

Offset Address: 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOW2	LOW1	LOW0	TME2	TME1	TME0	CODE[1:0]		ABRQ2	Reserved			TERR2	ALST2	TXOK2	RQCP2

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABRQ1	Reserved			TERR1	ALST1	TXOK1	RQCP1	ABRQ0	Reserved			TERR0	ALST0	TXOK0	RQCP0

Bit	Name	Access	Description	Reset value
31	LOW2	RO	Lowest priority flag for Tx mailbox2 1: The priority of Tx mailbox2 is the lowest; 0: The priority of Tx mailbox2 is not the lowest.	0
30	LOW1	RO	Lowest priority flag for Tx mailbox1 1: The priority of Tx mailbox1 is the lowest; 0: The priority of Tx mailbox1 is not the lowest.	0
29	LOW0	RO	Lowest priority flag for Tx mailbox0 1: The priority of Tx mailbox0 is the lowest;	0

			0: The priority of Tx mailbox0 is not the lowest.	
28	TME2	RO	Indicates the empty flag bit of sending mailbox 2. 1: Indicates that mailbox 2 does not have a message waiting to be sent. 0: Indicates that mailbox 2 is waiting to send a message.	1
27	TME1	RO	Indicates the empty flag bit of sending mailbox 1. 1: Indicates that mailbox 1 does not have a message waiting to be sent. 0: Indicates that mailbox 1 is waiting to send a message.	1
26	TME0	RO	Indicates the empty flag bit of sending mailbox 0. 1: Indicates that mailbox 0 does not have a message waiting to be sent. 0: Indicates that mailbox 0 is waiting to send a message.	1
[25:24]	CODE[1:0]	RO	Mailbox number. When more than one mailbox is empty, the next mailbox number is empty; when the mailbox is empty, it means the mailbox number with the lowest priority.	00
23	ABRQ2	RW1	Transmit a send abort request for mailbox 2. Software setting 1 can abort the transmitting request of mailbox 2, and the hardware is cleared when transmitting messages are cleared. If mailbox 2 is empty, software setting 1 is invalid.	0
[22:20]	Reserved	RO	Reserved	0
19	TERR2	RW1	Tx mailbox2 transmit error. When Tx mailbox2 fails, this bit is automatically set to 1. Set to 1 by software to clear, software write 0 is invalid.	0
18	ALST2	RW1	Tx mailbox 2 arbitration failure flag. When the Tx mailbox 2 has a low arbitration priority and fails to send, this bit is automatically set to 1. Set to 1 by software to clear, software write 0 is invalid.	0
17	TXOK2	RW1	Tx mailbox 2 transmit OK. 1: The last transmission was successful; 0: The last transmission failed. Set to 1 by software to clear, software write 0 is invalid.	0
16	RQCP2	RW1	Tx mailbox 2 request completion flag, this bit is automatically set to 1 when the sending or aborting request of sending mailbox 2 is completed. Set to 1 by software to clear, software	0

			write 0 is invalid.	
15	ABRQ1	RW1	Tx a transmit abort request for mailbox 1. When the software is set to 1, the transmitting request of mailbox 1 can be aborted and the hardware will clear 0 when the message is cleared. The software write 0 is invalid.	0
[14:12]	Reserved	RO	Reserved	0
11	TERR1	RW1	Tx mailbox1 transmit error. When Tx mailbox1 fails, this bit is automatically set to 1. Set to 1 by software to clear, software write 0 is invalid.	0
10	ALST1	RW1	Tx mailbox1 arbitration failure flag. When the Tx mailbox1 has a low arbitration priority and fails to send, this bit is automatically set to 1.	0
9	TXOK1	RW1	Tx mailbox1 transmit OK. 1: The last transmission was successful; 0: The last transmission failed. Set to 1 by software to clear, software write 0 is invalid.	0
8	RQCP1	RW1	Tx mailbox1 request completion flag, this bit is automatically set to 1 when the sending or aborting request of sending mailbox 1 is completed. Set to 1 by software to clear, software write 0 is invalid.	0
7	ABRQ0	RW1	Tx a transmit abort request for mailbox 0. Set to 1 by software to abort the transmitting request of mailbox 0, and reset to 0 by hardware when the sent message is cleared. Software write 0 is invalid.	0
[6:4]	Reserved	RO	Reserved	0
3	TERR0	RW1	Tx mailbox0 transmit error. When Tx mailbox0 fails, this bit is automatically set to 1. Set to 1 by software to clear, software write 0 is invalid.	0
2	ALST0	RW1	Tx mailbox0 arbitration failure flag. When Tx mailbox0 has a low arbitration priority and fails to send, this bit is automatically set to 1. Set to 1 by software to clear, software write 0 is invalid.	0
1	TXOK0	RW1	Tx mailbox0 transmit OK. 1: The last transmission was successful; 0: The last transmission failed. Set to 1 by software to clear, software write 0 is invalid.	0
0	RQCP0	RW1	Tx mailbox0 request completion flag. When the send or abort request of Tx mailbox0 is	0

			completed, this bit is automatically set to 1. Set to 1 by software to clear, software write 0 is invalid.	

22.8.4 CAN Receive FIFO 0 Status Register (CAN_RFIFO0)

Offset address: 0x0C

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										RFOM0	FOVR0	FULL0	Reserved	FMP0[1:0]	

Bit	Name	Access	Description	Reset value
[31:6]	Reserved	RO	Reserved	0
5	RFOM0	RW1	When the software sets this bit to 1, it releases the current mailbox message of the receiving FIFO_0, and automatically clears it to 0 after the release, and software write 0 is invalid.	0
4	FOVR0	RW1	Receive FIFO_0 overflow flag. When there are 3 messages in FIFO_0, a new message is received, and the hardware is set to 1. This bit needs software to be set to 1 and cleared to 0, and software write 0 is invalid.	0
3	FULL0	RW1	Receive FIFO_0 full flag. Set by hardware when there are 3 messages in FIFO_0. This bit needs software to be set to 1 and cleared to 0, and software write 0 is invalid.	0
2	Reserved	RO	Reserved	0
[1:0]	FMP0[1:0]	RO	Number of received FIFO_0 messages.	0

22.8.5 CAN Receive FIFO 1 Status Register (CAN_RFIFO1)

Offset address: 0x10

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										RFOM1	FOVR1	FULL1	Reserved	FMP1[1:0]	

Bit	Name	Access	Description	Reset value
[31:6]	Reserved	RO	Reserved	0
5	RFOM1	RW1	When the software sets this bit to 1, the current mailbox message of the receiving FIFO_1 is released, and it is automatically cleared to 0 after	0

			the release, and software write 0 is invalid.	
4	FOVR1	RW1	Receive FIFO_1 overflow flag. When there are 3 messages in FIFO_1, a new message is received, and the hardware is set to 1. This bit needs software to be set to 1 and cleared to 0, and software write 0 is invalid.	0
3	FULL1	RW1	Receive FIFO_1 full flag. Set to 1 by hardware when there are 3 messages in FIFO_1. This bit needs software to be set to 1 and cleared to 0, and software write 0 is invalid.	0
2	Reserved	RF	Reserved	0
[1:0]	FMP1[1:0]	RO	Number of received FIFO_1 messages.	0

22.8.6 CAN Interrupt Enable Register (CAN_INTENR)

Offset address: 0x14

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIE	Reserved			LECIE	BOFIE	EPVIE	EWGIE	Reserved	FOVIE1	FFIE1	FMPIE1	FOVIE0	FFIE0	FMPIE0	TMEIE

Bit	Name	Access	Description	Reset value
[31:18]	Reserved	RO	Reserved	0
17	SLKIE	RW	Sleep interrupt enable. 1: When entering the sleep state, an interrupt is generated; 0: No interrupt is generated when entering sleep state.	0
16	WKUIE	RW	Wake-up interrupt enable. 1: When the CAN controller is woken up, an interrupt is generated; 0: No interrupt is generated when the CAN controller wakes up.	0
15	ERRIE	RW	Error interrupt enable, CAN error interrupt always enable. 1: When the CAN controller generates an error, an interrupt is generated; 0: No interrupt is generated when the CAN controller generates an error.	0
[14:12]	Reserved	RF	Reserved.	0
11	LECIE	RW	Last error number interrupt enable. 1: When an error is detected, the hardware	0

			updates LEC[2:0], updates the ERRI bit to 1, and triggers an error interrupt; 0: When an error is detected, the hardware updates LEC[2:0], does not update the ERRI bit, and does not trigger an error interrupt.	
10	BOFIE	RW	Offline interrupt enable. 1: When entering the offline state, update the ERRI bit to 1, triggering an error interrupt; 0: When entering the offline state, the ERRI bit will not be updated, and the error interrupt will not be triggered.	0
9	EPVIE	RW	Error passive interrupt enable. 1: When entering the error passive state, update the ERRI bit to 1, triggering an error interrupt; 0: When entering the error passive state, the ERRI bit is not updated and the error interrupt is not triggered.	0
8	EWGIE	RW	Error warning interrupt enable. 1: When the number of errors reaches the warning threshold, update the ERRI bit to 1, triggering an error interrupt; 0: When the number of errors reaches the warning threshold, the ERRI bit will not be updated, and the error interrupt will not be triggered.	0
7	Reserved	RF	Reserved	0
6	FOVIE1	RW	Receive FIFO_1 overflow interrupt enable. 1: When FIFO_1 overflows, trigger FIFO_1 interrupt; 0: When FIFO_1 overflows, do not trigger FIFO_1 interrupt.	0
5	FFIE1	RW	Receive FIFO_1 full interrupt enable. 1: When FIFO_1 is full, trigger FIFO_1 interrupt; 0: When FIFO_1 is full, do not trigger FIFO_1 interrupt.	0
4	FMPIE1	RW	Receive FIFO_1 message registration interrupt enable. 1: When FIFO_1 updates the FMP bit and is not 0, trigger FIFO_1 interrupt; 0: When FIFO_1 updates the FMP bit, and it is not 0, the FIFO_1 interrupt is not triggered.	0
3	FOVIE0	RW	Receive FIFO_0 overflow interrupt enable. 1: When FIFO_0 overflows, trigger FIFO_0 interrupt; 0: When FIFO_0 overflows, not trigger FIFO_0	0

			interrupt.	
2	FFIE0	RW	Receive FIFO_0 full interrupt enable. 1: When FIFO_0 is full, trigger FIFO_0 interrupt; 0: When FIFO_0 is full, not trigger FIFO_0 interrupt.	0
1	FMPIE0	RW	Receive FIFO_0 message registration interrupt enable. 1: When FIFO_0 updates the FMP bit and is not 0, trigger FIFO_0 interrupt; 0: When FIFO_0 updates the FMP bit, and it is not 0, the FIFO_0 interrupt is not triggered.	0
0	TMEIE	RW	Tx mailbox empty interrupt. 1: When the Tx mailbox is empty, an interrupt is generated; 0: No interrupt is generated when the Tx mailbox is empty.	0

22.8.7 CAN Error Status Register (CAN_ERRSR)

Offset Address: 0x18

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REC[7:0]								TEC[7:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									LEC[2:0]			Reserved	BOFF	EPVF	EWGF

Bit	Name	Access	Description	Reset value
[31:24]	REC[7:0]	RO	Receive error counter. When the CAN receives an error, according to the error condition, the counter is incremented by 1 or 8; after successful reception, the counter is decreased by 1 or set to 120 (the error count value is greater than 127). When the counter value exceeds 127, the CAN enters the error passive state.	0
[23:16]	TEC[7:0]	RO	Transmit error counter. When the CAN sends an error, according to the error condition, the counter is incremented by 1 or 8; after the transmission is successful, the counter is decremented by 1 or set to 120 (the error count value is greater than 127). When the counter value exceeds 127, the CAN enters the error passive state.	0
[15:7]	Reserved	RO	Reserved	0
[6:4]	LEC{2:0}	RW	Last error code. When detecting the sending error on the CAN bus,	0

			the controller will set according to the error condition, and set 000b when sending and receiving the message correctly. 000: no error; 001: Bit stuffing error; 010: FORM format error; 011: ACK confirmation error; 100: recessive bit error; 101: Dominant bit error; 110: CRC error; 111: Software settings. Usually when the application software reads the error, the code name is set to 111b, and the code name update can be detected.	
3	Reserved	RO	Reserved	0
2	BOFF	RO	Offline status flag. When the CAN controller enters the offline state, the hardware automatically sets it to 1; when it exits the offline state, the hardware automatically clears it to 0.	0
1	EPVF	RO	Error passive flag. When the transceiver error counter reaches the error passive threshold, that is, greater than 127, the hardware is set to 1.	0
0	EWGF	RO	Error warning flag bit. When the sending and receiving error counter reaches the warning threshold, that is, greater than or equal to 96, the hardware is set to 1.	0

22.8.8 CAN Bit Timing Register (CAN_BTIMR)

Offset address: 0x1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SILM	LBKM	Reserved		SJW[3:0]				TS2[3:0]				TS1[3:0]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BTR_TS1_T				Reserved		BRP[9:0]									

Bit	Name	Access	Description	Reset value
31	SILM	RW	Silent mode setting. 1: Enter silent mode; 0: Exit silent mode.	0
30	LBKM	RW	Loopback mode setting. 1: Enter loopback mode;	0

			0: Exit loopback mode.	
[29:28]	Reserved	RO	Reserved	0
[27:24]	SJW[3:0]	RW	Defines the resync jump width setting value. When implementing resynchronization, the upper limit of the minimum number of time units that can be extended and reduced in the bit, the actual value is (SJW[1:0]+1), and the range can be set to 1 to 4 minimum time units.	0001b
[23:20]	TS2[3:0]	RW	Time period 2 set value. It defines how many minimum time units are occupied by time period 2, and the actual value is (TS2[1:0]+1).	0010b
[19:16]	TS1[3:0]	RW	Time period 1 set value. It defines how many minimum time units are occupied by time period 1, and the actual value is (TS1[1:0]+1).	0011b
[15:12]	BTR_TS1_T	RW	For legacy CAN, CLAS_LONG_TS1=0, then TS1 is TS[3:0] (4bit); CLAS_LONG_TS1=1, then TS1 is TS[1:0] + BTR_TS1_T[15:12] (6bit).	0
[11:10]	Reserved	RO	Reserved	0
[9:0]	BRP[9:0]	RW	Minimum time unit length setting value Tq = (BRP[9:0]+1) $\times t_{pclkS}$	0

22.8.9 CAN Time Trigger Control Register (CAN_TTCTLR)

Offset address: 0x20

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

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

Bit	Name	Access	Description	Reset value
[31:18]	Reserved	RO	Reserved	0
17	MODE	RW	Time-triggered mode selection. 1: Enhanced mode; 0: Default mode.	0
16	TIMRST	WZ	Internal counter reset control. Write 1 to reset the internal counter, the hardware will automatically clear 0	0
[15:0]	TIMCMV[15:0]	RW	Internal counter count end value	ffffh

22.8.10 CAN Time Trigger Count Value Register (CAN_TTCNT)

Offset address: 0x24

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

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

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:0]	TIMCNT[15:0]	RW	Time-triggered count value	0

22.8.11 CAN Offline Recovery Error Counter (CAN_TERR_CNT)

Offset address: 0x28

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

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

Bit	Name	Access	Description	Reset value
[31:9]	Reserved	RO	Reserved	0
[8:0]	TX_ERR_CNT	RW	Currently, the error count value is recovered offline. If you modify the count value, you can recover it immediately from offline.	0

22.8.12 CANFD Control Register (CANFD_CR)

Offset address: 0x2C

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						RESTRICT_MODE	CLAS_LONG_TS1	RES_EXCEPT	USER_ESI_B		TX_BRS_B				TX_FD

Bit	Name	Access	Description	Reset value
[31:10]	Reserved	RO	Reserved	0
9	RESTRICT_MODE	RW	Restricted operating mode enable of CANFD. 1: The error counter will not increase or decrease, and can only send hidden error frames and overloaded frames, and reception will not be affected. 0: Invalid	0

8	CLAS_LONG_TS1	RW	Selecting the TS1 phase length for CAN 1: 6bit 0: 4bit	0
7	RES_EXCEPT	RW	The FD frame RES bit protocol exception enables: 1: Receive an implicit RES bit will not cause an error. 0: Receive the implicit RES bit produces a form error.	0
[6:4]	USER_ESI_B	RW	When transmitting FD frames, there are three ESI bits of the transmitting mailbox: 1: The software forces the sending of hidden ESI bits. 0: Automatic hardware configuration, send explicit ESI when the error is active, and implicit ESI when the error is passive	0
[3:1]	TX_BRS_B	RW	When sending FD frames, there are three BRS bits of the sending mailbox: 1: The corresponding channel enables bit rate switching. 0: The corresponding channel does not enable bit rate switching.	111b
0	TX_FD	RW	Transmit FD frame enable bit: 1: Transmit FD frame. 0: Transmit traditional frames.	0

22.8.13 CANFD Timing Register (CANFD_BTR)

Offset address: 0x30

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								TDCE	Reserved		BTR_BRP_FD				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			BTR_TS1_FD					BTR_TS2_FD				BTR_SJW_FD			

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RO	Reserved	0
23	TDCE	RW	FD frame transmission delay compensation enable	1
[22:21]	Reserved	RO	Reserved	0
[20:16]	BTR_BRP_FD	RW	Minimum time cell length setting value Tq = (BRP+1) × tpclk Note: The CAN baud rate calculation formula is: CANbps = PCLK1/((TSI+1+TS2+1+1)*(BPR+1))	0
[15:13]	Reserved	RO	Reserved	0

[12:8]	BTR_TS1_FD	RW	Time period 1 setup value: Defines how many minimum time units are occupied by time period 1, the actual value is TS1+1	00110b
[7:4]	BTR_TS2_FD	RW	Time period 2 setup value: Defines how many minimum time units are occupied by time period 2, the actual value is TS2+1	0011b
[3:0]	BTR_SJW_FD	RW	Resynchronization jump width setting value: The upper limit on the number of minimum time units that can be extended and reduced in the bit when resynchronization is implemented, the actual value is FD_SJW+1 and the range can be set from 1 to 4 minimum time units.	0111b

22.8.14 CANFD Transmit Delay Compensation Register (CANFD_TDCT)

Offset address: 0x34

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

Bit	Name	Access	Description	Reset value
[31:14]	Reserved	RO	Reserved	0
[13:8]	TDC_FILTER	RW	Transmitter delay timing filtering: Define the minimum delay of the internal delay counter to prevent the dominant bit burr on the RX from causing the delay counter to end prematurely.	0
[7:6]	Reserved	RO	Reserved	0
[5:0]	TDCO	RW	Transmitter delay compensation offset: FD_TDCO+1	000010b

22.8.15 CANFD Transmit Delay Compensation Value Register (CANFD_PSR)

Offset address: 0x38

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

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RO	Reserved	0
[23:16]	TDCV	RO	Actual value of FD transmission delay compensation: value of internal delay counter	0

			plus FD_TDCO	
[15:0]	Reserved	RO	Reserved	0

22.8.16 CANFD DMA Transmit Mailbox 0 Cache Register (CANFD_DMA_T0)

Offset address: 0x3C

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	DMA_ADDR_T0														

Bit	Name	Access	Description	Reset value
[31:15]	Reserved	RO	Reserved	0
[14:0]	DMA_ADDR_T0	RW	The FD frame sends the transmit buffer corresponding to mailbox 0. The address must be 4-byte aligned.	0

22.8.17 CANFD DMA Transmit Mailbox 1 Cache Register (CANFD_DMA_T1)

Offset address: 0x40

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	DMA_ADDR_T1														

Bit	Name	Access	Description	Reset value
[31:15]	Reserved	RO	Reserved	0
[14:0]	DMA_ADDR_T1	RW	The FD frame sends the transmit buffer corresponding to mailbox 1. The address must be 4-byte aligned.	0

22.8.18 CANFD DMA Transmit Mailbox 2 Cache Register (CANFD_DMA_T2)

Offset address: 0x44

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	DMA_ADDR_T2														

Bit	Name	Access	Description	Reset value
[31:15]	Reserved	RO	Reserved	0

[14:0]	DMA_ADDR_T2	RW	The FD frame sends the transmit buffer corresponding to mailbox 2. The address must be 4-byte aligned.	0

22.8.19 CANFD DMA Receive Mailbox 0 Cache Register (CANFD_DMA_R0)

Offset address: 0x48

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	DMA_ADDR_R0														

Bit	Name	Access	Description	Reset value
[31:15]	Reserved	RO	Reserved	0
[14:0]	DMA_ADDR_R0	RW	FD frame receive FIFO0 corresponds to the receive buffer, the address must be 4-byte aligned.	0

22.8.20 CANFD DMA Receive Mailbox 1 Cache Register (CANFD_DMA_R1)

Offset address: 0x4C

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	DMA_ADDR_R1														

Bit	Name	Access	Description	Reset value
[31:15]	Reserved	RO	Reserved	0
[14:0]	DMA_ADDR_R1	RW	FD frame receive FIFO1 corresponds to the receive buffer, the address must be 4-byte aligned.	0

22.8.21 CAN Tx Mailbox Identifier Register (CAN_TXMIRy) (y=0/1/2)

Offset address: 0x180,0x190,0x1A0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STID[10:0]/EXID[28:18]											EXID[17:13]				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXID[12:0]													IDE	RTR	TXRQ

Bit	Name	Access	Description	Reset value

[31:21]	STID[10:0] /EXID[28:18]	RW	The upper 11 bits of a standard or extended identifier.	x
[20:3]	EXID[17:0]	RW	The lower 18 bits of the extended identifier.	x
2	IDE	RW	Identifier selection flag. 1: Extended identifier; 0: Standard identifiers.	x
1	RTR	RW	Remote frame selection flag. 1: Remote frame; 0: Data frame.	x
0	TXRQ	RW	Data transmission request flag. When the software is set to 1, the data in the mailbox is requested to be sent. When the mailbox is empty after sending, the hardware is cleared to 0.	0

22.8.22 CAN Tx Mailbox Data Length and Timestamp Register (CAN_TXMDTRy) (y=0/1/2)

Offset address: 0x184,0x194,0x1A4

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

Bit	Name	Access	Description	Reset value
[31:16]	TIME[15:0]	RW	The 16-bit timer value used to send the message SOF time.	x
[15:9]	Reserved	RO	Reserved	0
8	TGT	RW	Message timestamp transmission selection flag. This bit is valid when TTCM is set to 1 and the message length is 8. 1: Transmit timestamp, the value is the immediate value of TIME[15:0], replacing the last 2 bytes of the 8-byte message; 0: No timestamp is transmitted.	x
[7:4]	Reserved	RO	Reserved	0
[3:0]	DLC[3:0]	RW	The data length of the data frame or the data length of the remote frame request. The data length can be set from 0 to 64.	0

22.8.23 CAN Tx Mailbox Data Low Register (CAN_TXMDLRy) (y=0/1/2)

Offset address: 0x184,0x194,0x1A4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA3[7:0]								DATA2[7:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA1[7:0]								DATA0[7:0]							

Bit	Name	Access	Description	Reset value
[31:24]	DATA3[7:0]	RW	Tx the content of data byte 3.	x
[23:16]	DATA2[7:0]	RW	Tx the content of data byte 2.	x
[15:8]	DATA1[7:0]	RW	Tx the content of data byte 1.	x
[7:0]	DATA0[7:0]	RW	Tx the contents of data byte 0.	x

22.8.24 CAN Tx Mailbox Data High Register (CAN_TXMDHRy) (y=0/1/2)

Offset address: 0x18C,0x19C,0x1AC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA7[7:0]								DATA6[7:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA5[7:0]								DATA4[7:0]							

Bit	Name	Access	Description	Reset value
[31:24]	DATA7[7:0]	RW	Tx the content of data byte 7.	x
[23:16]	DATA6[7:0]	RW	Tx the content of data byte 6.	x
[15:8]	DATA5[7:0]	RW	Tx the content of data byte 5.	x
[7:0]	DATA4[7:0]	RW	Tx the contents of data byte 4.	x

22.8.25 CAN Rx Mailbox Identifier Register (CAN_RXMIRy) (y=0/1)

Offset address: 0x1B0,0x1C0

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

Bit	Name	Access	Description	Reset value
[31:21]	STID[10:0] /EXIDH[28:18]	RO	The upper 11 bits of a standard or extended identifier.	x
[20:3]	EXIDL[17:0]	RO	The lower 18 bits of the extended identifier.	x
2	IDE	RO	Identifier selection flag. 1: Select extended identifier;	x

			0: Use standard identifiers.	
1	RTR	RO	Remote frame selection flag. 1: Remote frame; 0: Data frame.	x
0	FDF	RO	Reserved	0

22.8.26 CAN Rx Mailbox Data Length and Timestamp Register (CAN_RXMDTRy) (y=0/1)

Offset address: 0x1B4,0x1C4

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FMI[7:0]								Reserved	RES	ESI	BRS	DLC[3:0]			

Bit	Name	Access	Description	Reset value
[31:16]	TIME[15:0]	RO	16-bit timer value used to receive the SOF time of the message.	0
[15:8]	FMI[7:0]	RO	The filter number matched by the packet.	x
7	Reserved	RO	Reserved	0
6	RES	RO	The RES bit of the currently received frame	0
5	ESI	RO	The ESI bit of the currently received frame	0
4	BRS	RO	The BRS bit of the currently received frame	0
[3:0]	DLC[3:0]	RO	Received message data length. DLC=0-8: Standard CAN is the same as CANFD, indicating data frame lengths of 0 through 8 and 0 for remote frames. DLC=9-15: Unique to CANFD, indicates that the data frame length is detailed in Table 22-5.	x

22.8.27 CAN Rx Mailbox Data Low Register (CAN_RXMDLRy) (y=0/1)

Offset address: 0x1B8,0x1C8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA3[7:0]								DATA2[7:0]							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA1[7:0]								DATA0[7:0]							

Bit	Name	Access	Description	Reset value
[31:24]	DATA3[7:0]	RO	The data byte 3 that receives the message.	x
[23:16]	DATA2[7:0]	RO	The data byte 2 that receives the message.	x
[15:8]	DATA1[7:0]	RO	The data byte 1 that receives the message.	x
[7:0]	DATA0[7:0]	RO	The data byte 0 that receives the message.	x

22.8.28 CAN Rx Mailbox Data High Register (CAN_RXMDHRy) (y=0/1)

Offset address: 0x1BC,0x1CC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA7[7:0]								DATA6[7:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA5[7:0]								DATA4[7:0]							

Bit	Name	Access	Description	Reset value
[31:24]	DATA7[7:0]	RO	The data byte 7 that receives the message.	x
[23:16]	DATA6[7:0]	RO	The data byte 6 that receives the message.	x
[15:8]	DATA5[7:0]	RO	The data byte 5 that receives the message.	x
[7:0]	DATA4[7:0]	RO	The data byte 4 that receives the message.	x

22.8.29 CAN Filter Main Control Register (CAN_FCTLR)

Offset address: 0x200

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

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0x2A1C
[15:1]	Reserved	RO	Reserved	0
0	FINIT	RW	Filter initialization mode enable flag. 1: Filter bank is in initialization mode; 0: Filter bank is in normal mode.	1

22.8.30 CAN Filter Mode register (CAN_FMCFGR)

Offset address: 0x204

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		FBM13	FBM12	FBM11	FBM10	FBM9	FBM8	FBM7	FBM6	FBM5	FBM4	FBM3	FBM2	FBM1	FBM0

Bit	Name	Access	Description	Reset value
[31:14]	Reserved	RO	Reserved	0
[13:0]	FBMx	RW	Working mode control of filter bank x. Only can be written when FINIT is 1. 0: The register of filter bank x is in mask bit	0

			mode; 1: The register of filter bank x is in identifier list mode.	

22.8.31 CAN Filter Bit Width Register (CAN_FSCFGR)

Offset address: 0x20C

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		FSC13	FSC12	FSC11	FSC10	FSC9	FSC8	FSC7	FSC6	FSC5	FSC4	FSC3	FSC2	FSC1	FSC0

Bit	Name	Access	Description	Reset value
[31:14]	Reserved	RO	Reserved	0
[13:0]	FSCx	RW	Bit width control bit of filter bank x. Only can be written when FINIT is 1. 1: The register of filter bank x is a single 32-bit; 0: The register of filter bank x is 2 16-bit.	0

22.8.32 CAN Filter FIFO Association Register (CAN_FAFIFOR)

Offset address: 0x214

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		FFA13	FFA12	FFA11	FFA10	FFA9	FFA8	FFA7	FFA6	FFA5	FFA4	FFA3	FFA2	FFA1	FFA0

Bit	Name	Access	Description	Reset value
[31:14]	Reserved	RO	Reserved	0
[13:0]	FFAx	RW	Filter bank x activation control bit. Only can be written when FINIT is 1. 1: Filter bank x active; 0: Filter bank x disabled.	0

22.8.33 CAN Filter Activation Register (CAN_FWR)

Offset address: 0x21C

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		FACT13	FACT12	FACT11	FACT10	FACT9	FACT8	FACT7	FACT6	FACT5	FACT4	FACT3	FACT2	FACT1	FACT0

Bit	Name	Access	Description	Reset value
[31:14]	Reserved	RO	Reserved	0
[13:0]	FACTx	RW	Filter bank x activation control bit. 1: Filter bank x active; 0: Filter bank x disabled.	0

22.8.34 CAN Filter Group register (CAN_FiRx) (i=0-13, x=1/2)

Offset address: 0x240-0x31C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FB31	FB30	FB29	FB28	FB27	FB26	FB25	FB24	FB23	FB22	FB21	FB20	FB19	FB18	FB17	FB16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FB15	FB14	FB13	FB12	FB11	FB10	FB9	FB8	FB7	FB6	FB5	FB4	FB3	FB2	FB1	FB0

Bit	Name	Access	Description	Reset value
[31:0]	FB	RW	The flag bit of the register in the filter group can only be written if FINIT is 1. Identifier pattern. 1: The expected level of the corresponding position is recessive. 0: The expected level of the corresponding bit is the dominant bit. Shielded bit mode. 1: Must be consistent with the corresponding identifier register bit. 0: Does not need to be consistent with the corresponding identifier register bit.	0

Chapter 23 Operational Amplifier (OPA) and Comparator (CMP)

The module consists of an independently configurable operational amplifier (OPA or PGA) and three independently configurable voltage comparators (CMP). The operational amplifier (OPA or PGA) supports gain selection or can be used as a voltage comparator.

The input and output of each operational amplifier are connected to the I/O port, and the input pin or gain is optional, and the output pin can be optionally configured to the universal I/O port or multiplexed as an ADC sampling channel. The external analog small signal is amplified into the ADC to achieve small-signal ADC conversion.

The input and output of each voltage comparator are connected to the I/O port, and the input pins are optional, and the output pins can be optionally configured to the universal I/O port or multiplexed as an TIM internal sampling channel (without using the I/O pin).

23.1 Main Features

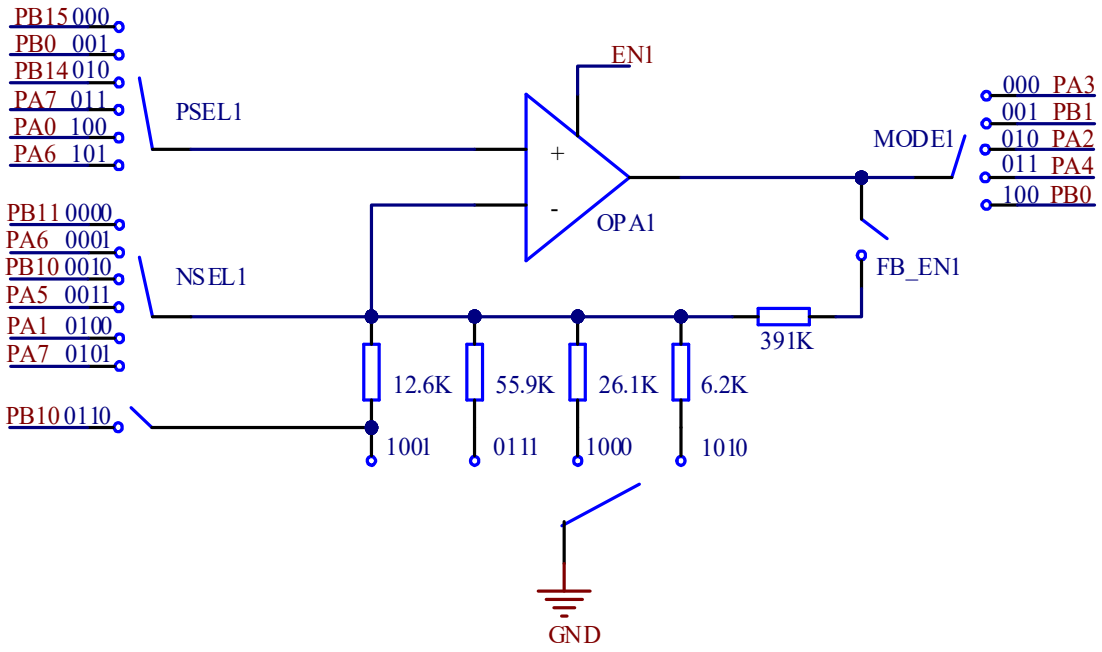
- OPA input pin or channel can be selected.
- OPA output pin can choose universal I/O port or ADC sampling channel.
- OPA supports front-end input polling.
- OPA supports PGA gain selection.
- CMP input pin is optional, negative input channel optional common pin.
- CMP output pin can choose universal I/O port or TIM internal sampling channel.
- OPA interrupt wakes system from sleep mode
- CMP interrupt wakes the system from sleep and stop modes

23.2 Function Description

23.2.1 OPA

Set the EN1 in the OPA_CTLR1 register to enable the corresponding OPA1, configure the MODE1 in the OPA_CTLR1 register to choose the output channel of the OPA1 as the ADC sampling channel or the ordinary Imax O port, configure the PSEL1 in the OPA_CTLR1 register, select the positive input pin of the OPA1, configure the NSEL1 in the OPA_CTLR1 register, choose the negative input channel of the OPA1, or be used as the gain when the PGA is used.

Figure 23-1 OPA Structure Diagram



23.2.2 OPA Positive Input Polling

Each OPA's P terminal can be selected from OPA_P0/OPA_P1/OPA_P2/OPA_P3/OPA_P4/OPA_P5, and the polling function of the OPA can be realized by selecting OPA_P0/OPA_P1/OPA_P2/OPA_P3/OPA_P4/OPA_P5 sequentially at regular intervals to take turns to select all P terminals; the OPA1 with polling enabled can be selected by configuring the POLL_EN bit in the OPA_CFGR1 register. (Note: The P-side polling order cannot be changed; see the POLL1_NUM[2:0] bits in the OPA_CFGR2 register for details on setting the polling order)

The number of polled channels can be configured by POLL1_NUM[2:0] in the OPA_CFGR2 register, the polling interval can be configured by POLL_VLU[8:0] in the OPA_CFGR2 register, and the polled P-terminal query can be configured by POLL1_CNT in the OPA_CFGR2 register; for example, if POLL1_CNT = 001b, the corresponding P-terminal is OPA_P1, and so on. For example, if POLL1_CNT=001b, the corresponding P-end is OPA_P1, and so on, the corresponding polling P-end can be queried.

23.2.3 OPA Interrupt

The OPA interrupt can only wake up SLEEP mode, OPA interrupt configuration:

1) Configure the OPA interrupt in the kernel's PFIC to ensure that it can respond correctly;

2) Configure POLL_EN=1 to enable the OPA polling function, and configure POLL_VLU[8:0] to set the time to enter the interrupt.

3) Enable OPA to enable the OPA interrupt.

Interrupt Configuration:

(1) OPA interrupt

Set IE_OUT=1 to turn on OPA1 interrupt enable and enter the interrupt when OPA outputs high level.

(2) OPA polling interrupt

Set IE_CNT=1 to turn on the OPA1 polling interval end interrupt enable and enter the interrupt when the P terminal of OPA is polled once.

(3) OPANMI

Set NMI_EN=1 to turn on the OPA1 NMI interrupt enable, and enter the NMI interrupt when the OPA output is

high.

23.2.4 OPA Reset

After enabling the OPA function, set RST_EN=1 to turn on the OPA reset function, and the system is reset when the OPA output is high.

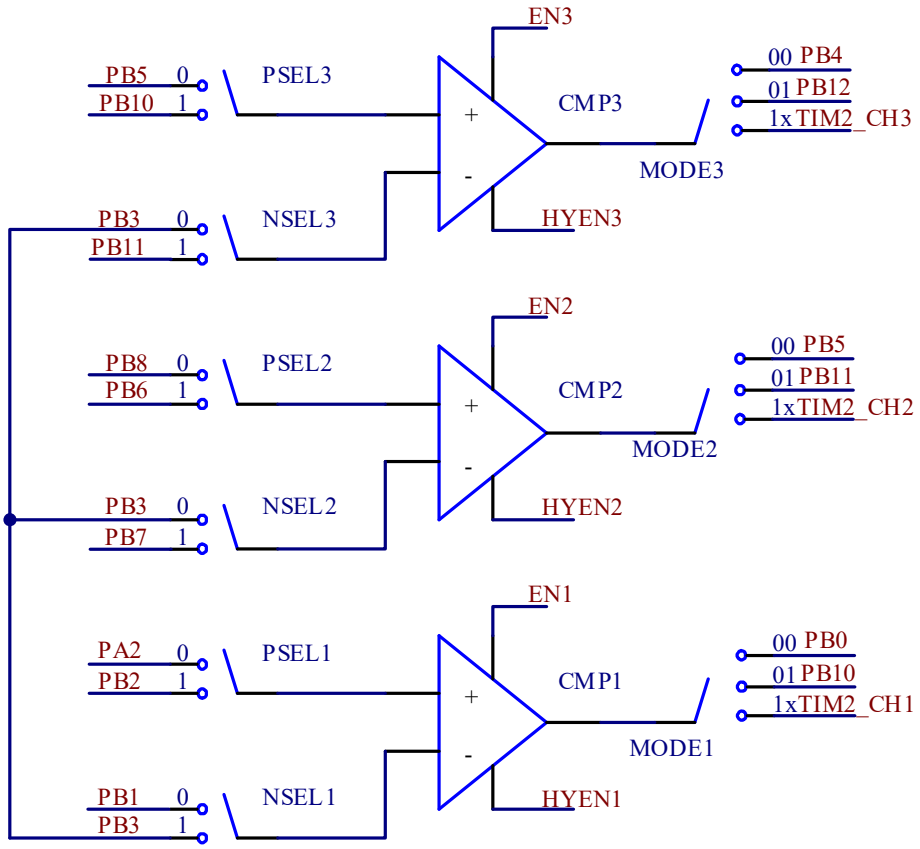
23.2.5 OPA Brake

The brake signal source can be selected by setting the BKIN_EN bit in the OPA_CFGR1 register. When BKIN_EN=1, the brake source of TIM1 comes from OPA, and at this time, it is invalid to use the IO pin for braking; the brake input polarity is valid when and only when the OPA output is high.

23.2.6 CMP

Set the ENx in the OPA_CTLR2 register to enable the corresponding CMPx, and configure the MODEx in the OPA_CTLR2 register to choose the output channel of the CMPx as the ordinary Icano port or the internal timer channel. Configure the PSELx in the OPA_CTLR2 register, select the positive input pin of the CMPx, configure the NSELx in the OPA_CTLR2 register, and select the negative input pin of the CMPx.

Figure 23-2 CMP Structure Diagram



Note: If the comparator output uses a timed channel, capture via timer is used to view the comparator output status.

23.2.7 CMP Interrupt

External Interrupt (EXTI) Line Exclusive to Comparator - EXTI22 (COMP Wake-Up Event), which generates an interrupt or event, can be used for wake-up in SLEEP, STOP low-power modes.

The required conditions when using external interrupt wake-up are:
1) Configure the event enable bit (EXTI_EVENR) of the corresponding external interrupt channel;
2) Configure the comparator output level trigger edge, selecting rising edge trigger, falling edge trigger or double edge trigger;
3) Configure the EXTI interrupt in the kernel's PFIC to ensure that it can respond correctly.
4) Enable CMP.

When using events, the required conditions are:
1) Configure the event enable bit (EXTI_EVENR) of the corresponding external interrupt channel;
2) Configure the comparator output level trigger edge, selecting rising edge trigger, falling edge trigger, or double edge trigger;
3) Enable CMP.

Select the wake-up source for the CMP output signal by configuring WKUP_MD[1:0].

Table 23-1 Wake-up source selection for CMP output signals

Set WKUP_MD[1:0]	01	10	11	00
Wake-up CMP output signal level	double-edge	Rising edge	Falling edge	invalid

Note: When the CMP will generate an interrupt request, the corresponding Interrupt Line 22 flag bit will also be set. Writing a 1 to the flag bit clears it.

23.2.8 CMP Low-power Mode

Setting the LPx bit, CMPx enters the low-power mode, the current corresponds to a reduction of about 3.2μA, and it will prolong the comparison time of CMP and increase the output offset voltage. Please refer to the CH32L103DS0 manual for the corresponding parameter values.

23.3 Register Description

Table 23-2 OPA-related registers

Name	Access address	Description	Reset value
R16_OPA_CFGR1	0x40026000	OPA configuration register 1	0x0080
R16_OPA_CFGR2	0x40026002	OPA configuration register 2	0x0000
R32_OPA_CTLR1	0x40026004	OPA control register 1	0x01D801D8
R32_OPA_CTLR2	0x40026008	OPA control register 2	0x00000000
R32_OPCM_KEY	0x40026014	POLL unlock key register	0xXXXXXXXX

23.3.1 OPA Configuration Register 1 (OPA_CFGR1)

Offset address: 0x00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	IF_CNT	Reserved	IF_OUT	NMI_EN	IE_CNT	Reserved	IE_OUT	OPCM_LOCK	Reserved		RST_EN	Reserved	BKIN_EN	Reserved	POLL_EN

Bit	Name	Access	Description	Reset value
15	Reserved	RO	Reserved	0
14	IF_CNT	RW0	Interrupt flag for end of OPA polling interval: 0: Invalid; 1: End of polling interval. Write 0 to clear, write 1 to invalidate.	0
13	Reserved	RO	Reserved	0
12	IF_OUT	RW0	Interrupt flag for polling to OPA1 output high: 0: Invalid; 1: Polling to OPA1 output high. Write 0 clear, write 1 invalid.	0
11	NMI_EN	RW	OPA connection NMI interrupt enable: 0: Off; 1: On.	0
10	IE_CNT	RW	Interrupt enable for end of OPA polling interval: 0: Turn off interrupt enable; 1: Turn on interrupt enable.	0
9	Reserved	RO	Reserved	0
8	IE_OUT	RW	OPA1 interrupt enable: 0: Turn off interrupt enable; 1: Turn on interrupt enable.	0
7	OPCM_LOCK	RW	OPA and CMP lock (write 1 to lock, write 0 to invalidate) 0: Unlocked, can write to other bits of the configuration registers 1: Locked, cannot write to other bits of the configuration registers	1
[6:5]	Reserved	RO	Reserved	0
4	RST_EN	RW	OPA1 reset system enable: 0: Turns off reset enable; 1: Turn on reset enable.	0
3	Reserved	RO	Reserved	0
2	BKIN_EN	RW	BKIN input source selection bit for TIM1: 0: TIM1's BKIN comes from the IO; 1: BKIN of TIM1 comes from OPA output.	0
1	Reserved	RO	Reserved	0
0	POLL_EN	RW	OPA1 positive polling enable 0: Off 1: On	0

23.3.2 OPA Configuration Register 2 (OPA_CFGR2)

Offset address: 0x02

15	14 13 12	11 10	9 8 7 6 5 4 3 2 1 0
Reserved	POLL1_CNT[2:0]	POLL1_NUM[2:0]	POLL_VLU[8:0]

Bit	Name	Access	Description	Reset value
15	Reserved	RO	Reserved	0
[14:12]	POLL1_CNT[2:0]	RO	Query the number of positive ends polled by OPA1: 000: 01P0 001: 01P1 010: 01P2 011: 01P3 100: 01P4 101: 01P5 Other: Reserved	0
[11:9]	POLL1_NUM[2:0]	RW	Configure the number of positive ends to be polled by OPA1 000: 1, 01P0 001: 2, 01P0+01P1 010: 3, 01P0+01P1+01P2 011: 4, 01P0+01P1+01P2+01P3 100: 5, 01P0+01P1+01P2+01P3+01P4 101: 6, 01P0+01P1+01P2+01P3+01P4+01P5 Others: Reserved	0
[8:0]	POLL_VLU[8:0]	RW	Configure the OPA1 polling sample interval time Polling interval = (POLL_VLU+1)*1us.	0

23.3.3 OPA Control Register 1 (OPA_CTLR1)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				ITRIMN[5:0]				Reserved		ITRIMP[5:0]					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			LP1	NSEL1[3:0]				FB_EN1	PSEL1[2:0]			MODE1[2:0]			EN1

Bit	Name	Access	Description	Reset value
[31:30]	Reserved	RO	Reserved	0
[29:24]	INTRIMN[5:0]	RW	OPA1 low level trim voltage	0
[23:22]	Reserved	RO	Reserved	0
[21:16]	ITRIMP[5:0]	RW	OPA1 high level trim voltage	0
[15:13]	Reserved	RO	Reserved	0
12	LP1	RW	OPA1 low-power mode selection 0: Off 1: On	0
[11:8]	NSEL1[3:0]	RW	OPA1 Negative Channel Selection and PGA Gain Selection 0000: PB11 0001: PA6	1111b

			0010: PB10 0011: PA5 0100：PA1 0101: PA7 0110: PB10, PGA mode, internal gain of 32, feedback resistor 391kΩ 0111: PGA mode, no negative input channel, internal gain 8 1000: PGA mode, no negative input channel, internal gain 16 1001: PGA mode, no negative input channel, internal gain 32 1010: PGA mode, no negative input channel, internal gain 64 1111: None selected, negative channel in high resistance state Others: Reserved	
7	FB_EN1	RW	OPA1 internal feedback resistor enable 0: Disable 1: Enable	0
[6:4]	PSEL1[2:0]	RW	OPA1 positive input selection 000: PB15 001: PB0 010: PB14 011: PA7 100: PA0 101: PA6 Others: None selected, positive channel in high resistance state	111b
[3:1]	MODE1[2:0]	RW	OPA1 output channel selection 000: Output signal via PA3 001: Output signal via PB1 010: Output signal via PA2 011: Output signal via PA4 100：Output signal through PB0 Others: None selected, output channels are in high resistance state.	111b
0	EN1	RW	OPA1 enable 0: Off 1: On	0

23.3.4 OPA Control Register 2 (OPA_CTLR2)

Offset address: 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						WKUP_MD[1:0]		Reserved	LP3	HYEN3	PSEL3	NSEL3	MODE3[1:0]		EN3

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	LP2	HYEN2	PSEL2	NSE2L	MODE2[1:0]		EN2	Reserved	LP1	HYEN1	PSEL1	NSEL1	MODE1[1:0]		EN1

Bit	Name	Access	Description	Reset value
[31:26]	Reserved	RO	Reserved	0
[25:24]	WKUP_MD[1:0]	RW	CMP wake-up level configuration: 00: Disables the CMP wake-up function; 01: Both rising and falling edges of the output of the comparator wake up the system; 10: Wake-up system on the rising edge of the output of the comparator; 11: The falling edge of the comparator's output wakes up the system.	0
23	Reserved	RO	Reserved	0
22	LP3	RW	CMP3 low-power switch 0: Off 1: On, 3.2μA	0
21	HYEN3	RW	CMP3 comparator hysteresis function selection 0: Off 1: On, ±15mV	0
20	PSEL3	RW	CMP3 positive input channel selection 0: PB5 1: PB10	0
19	NSEL3	RW	CMP3 negative input channel selection 0: PB3 1: PB11	0
[18:17]	MODE3[1:0]	RW	CMP3 output channel selection 00: Output channel is PB4 01: Output channel is PB12 1x: Output channel is internal channel TIM2_CH3	0
16	EN3	RW	CMP3 enable 0: Disable CMP3 1: Enable CMP3	0
15	Reserved	RO	Reserved	0
14	LP2	RW	CMP2 low-power switch 0: Off 1: On, 3.2μA	0
13	HYEN2	RW	CMP2 comparator hysteresis function selection 0: Off 1: On, ±15mV	0
12	PSEL2	RW	CMP2 positive input channel selection 0: PB8 1: PB6	0
11	NSE2L	RW	CMP2 negative input channel selection 0: PB3 1: PB7	0
[10:9]	MODE2[1:0]	RW	CMP2 output channel selection	0

			00: Output channel is PB5 01: Output channel is PB11 1x: Output channel is internal channel TIM2_CH2	
8	EN2	RW	CMP2 enable 0: Disable CMP2 1: Enable CMP2	0
7	Reserved	RO	Reserved	0
6	LP1	RW	CMP1 low-power switch 0: Off 1: On, 3.2μA	0
5	HYEN1	RW	CMP1 comparator hysteresis function selection 0: Off 1: On, ±15mV	0
4	PSEL1	RW	CMP1 positive input channel selection 0: PA2 1: PB2	0
3	NSEL1	RW	CMP1 negative input channel selection 0: PB1 1: PB3	0
[2:1]	MODE1[1:0]	RW	CMP1 output channel selection 00: Output channel is PB0 01: Output channel is PB10 1x: Output channel is internal channel TIM2_CH1	0
0	EN1	RW	CMP1 enable 0: Disable CMP1 1: Enable CMP1	0

23.3.5 OPA/CMP Unlock Key Register (OPCM_KEY)

Offset address: 0x14

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

Bit	Name	Access	Description	Reset value
[31:0]	OPCM_KEY[31:0]	RW	The OPA/CMP key, used to enter the OPA/CMP's unlock key includes: KEY1 = 0x45670123; KEY2 = 0xCDEF89AB.	X

Chapter 24 Flash Memory and User Option Bytes

24.1 Flash Memory Organization

The flash memory organization inside the chip is as follows:

Table 24-1 Flash memory organization

Block	Name	Address range	Size (byte)
Main memory	Page 0	0x0800 0000 – 0x0800 00FF	256
	Page 1	0x0800 0100 – 0x0800 01FF	256
	Page 2	0x0800 0200 – 0x0800 02FF	256
	Page 3	0x0800 0300 – 0x0800 03FF	256
	…	…	…
	Page 255	0x0800 FF00 – 0x0800 FFFF	256
Information block	Boot program code	0x1FFF 0000 – 0x1FFF 0CFF	3K+256
	User option bytes	0x1FFF F800 – 0x1FFF F8FF	256
	Vendor configuration words	0x1FFF F700 – 0x1FFF F7FF	256

Note: 1) The above main memory area is used for the user's application storage, and the write protection is divided in 2K bytes (8 pages) units; except that the "manufacturer configuration word" area is factory locked, the user is inaccessible. Other areas can be operated by users under certain conditions.

24.2 Flash Memory Programming and Safety

24.2.1 Program/Erase Methods

- Quick Programming: This method uses page operation. After a specific sequence of unlocking, it performs a single 256-byte programming and 256-byte erasing, 2K-byte erasing and whole chip erasing.

24.2.2 Security-preventing against Illegal Access (read, write and erase)

- Page write protection
- Read protection

When the chip is in read protection:
1) the main memory 0-7 pages (2K bytes) automatically write-protected state, not controlled by the FLASH_WPR register; unprotected state, all main storage pages are controlled by the FLASH_WPR register.
2) the boot code area, SDI mode and RAM area of the system can not erase or program the main memory, except the whole chip erase. The user option bytes area can be erased or programmed. If you try to unprotect the read (programming user bytes), the chip will automatically erase the entire user area.
Note: The internal RC oscillator (HSI) must be turned on when programming/erasing flash memory.

24.3 Register Description

Table 24-1 FLASH-related registers

Name	Access address	Description	Reset value
R32_FLASH_ACTLR	0x40022000	Access control register	0x00000000
R32_FLASH_KEYR	0x40022004	FPEC key register	0xXXXXXXXX
R32_FLASH_OBKEYR	0x40022008	OBKEY register	0xXXXXXXXX
R32_FLASH_STATR	0x4002200C	Status register	0x00000000
R32_FLASH_CTLR	0x40022010	Control register	0x00008080
R32_FLASH_ADDR	0x40022014	Address register	0x00000000
R32_FLASH_OBR	0x4002201C	Option byte register	0x0XXXXXFE
R32_FLASH_WPR	0x40022020	Write protection register	0xXXXXXXXX
R32_FLASH_MODEKEYR	0x40022024	Extension key register	0xXXXXXXXX

24.3.1 Access Control Register (FLASH_ACTLR)

Offset address: 0x00

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

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

Bit	Name	Access	Description	Reset value
[31:2]	Reserved	RO	Reserved	0
[1:0]	LATENCY	RW	FLASH wait state number 00: 0 wait (HCLK<=40MHz) 01: 1 wait (40MHz<HCLK<=72MHz) 10：2 wait (HCLK>72MHz) Others: Invalid.	00b

24.3.2 FPEC Key Register (FLASH_KEYR)

Offset address: 0x04

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

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

Bit	Name	Access	Description	Reset value
[31:0]	KEYR[31:0]	WO	FPEC key, the unlock key used to enter FPEC includes: KEY1 = 0x45670123; KEY2 = 0xCDEF89AB.	x

24.3.3 OBKEY Register (FLASH_OBKEYR)

Offset address: 0x08

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

Bit	Name	Access	Description	Reset value
[31:0]	OBKEYR[31:0]	WO	Selection word key for entering a selection word key to disarm OBWRE. KEY1 = 0x45670123; KEY2 = 0xCDEF89AB. (Note: FLASH needs to be unlocked first)	x

24.3.4 Status Register (FLASH_STATR)

Offset address: 0x0C

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

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved	0
7	TURBO	RO	TURBO 1: Setting to enter TURBO mode increases the quiescent current by about 100μA, which can accelerate the process of waking up after SLEEP 0: No effect.	0
6	FWAKE_FLAG	RWO	FLASH wake-up flag, write 0 to clear it 1: FALSH is woken up. 0: No effect	0
5	EOP	RW1	Indicates the end of the operation, write 1 clears. Hardware is reset each time a successful erase or program is performed.	0
4	WRPRTERR	RW1	Indicates a write-protect error, write 1 cleared. Hardware is reset if and when a write-protected address is programmed.	0
[3:1]	Reserved	RO	Reserved	0
0	BSY	RO	Indicates busy status: 1: Indicates that a flash operation is in progress; 0: The operation is finished.	0

Note: To perform the programming operation, you need to make sure that the STRT bit of the FLASH_CTLR register is 0.

24.3.5 Control Register (FLASH_CTLR)

Offset address: 0x10

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

Reserved								BER32	Reserved			BUFRST	BUFLOAD	FTER	FTPG
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLOCK	Reserved	FWAKEIE	EOPIE	Reserved	ERRIE	OBWRE	Reserved	LOCK	STRT	OBER	Reserved		MER	PER	Reserved

Bit	Name	Access	Description	Reset value
[31:25]	Reserved	RO	Reserved	0
23	BER32	RW	Perform block erase 32KB	0
[22:20]	Reserved	RO	Reserved	0
19	BUFRST	RW	BUF reset operation	0
18	BUFLOAD	RW	Cache data into BUF	0
17	FTER	RW	Performs a fast page (256Byte) erase operation.	0
16	FTPG	RW	Performs quick page programming operations.	0
15	FLOCK	RW1	Quick programming lock. Only '1' can be written. When this bit is '1' it indicates that the Quick Program/Erase mode is not available. Hardware clears this bit to '0' after the correct unlock sequence is detected. Software sets it to '1' to re-engage the lock.	1
14	Reserved	RO	Reserved	0
13	FWAKEIE	RW	FLASH Wake-up interrupt enable 1: Interrupt generation is enabled 0: Interrupt generation is disabled	0
12	EOPIE	RW	Operation completes interrupt control (EOP is set in FLASH_STATR register): 1: Interrupts are enabled. 0: Interrupts are disabled.	0
11	Reserved	RO	Reserved	0
10	ERRIE	RW	Error status interrupt control (PGERR/WRPRTERR set in FLASH_STATR register): 1: Interrupts are enabled. 0: Interrupts are disabled.	0
9	OBWRE	RW0	The user chooses the word lock, and the software clears 0: 1: Indicates that the words selected by the user can be programmed. The correct sequence needs to be written in the FLASH_OBKEYR register and then set by the hardware. 0: The user option bytes are re-locked after the software is cleared.	0
8	Reserved	RO	Reserved	0
7	LOCK	RW1	Lock. You can only write'1'. When the bit is'1', it	1

			means that FPEC and FLASH_CTLR are locked unwritable. After the correct unlock sequence is detected, the hardware clears this bit as'0'. After an unsuccessful unlock operation, the bit will not change until the next time the system is reset.	
6	STRT	RW1	Start. Setting 1 initiates an erase action, and the hardware automatically clears 0 (BSY becomes '0').	0
5	OBER	RW	Performs user option bytes erasure	0
[4:3]	Reserved	RO	Reserved	0
2	MER	RW	Performs a full erase operation (erases the entire user area).	0
1	PER	RW	Performs a sector erase	0
0	Reserved	RO	Reserved	0

24.3.6 Address Register (FLASH_ADDR)

Offset address: 0x14

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

Bit	Name	Access	Description	Reset value
[31:0]	FAR	RW	The flash address, which is the programming address for programming and the starting address for erasure. The register cannot be written when the BSY bit in the FLASH_STATR register is'1'.	0

24.3.7 Option Byte Register (FLASH_OBR)

Offset address: 0x1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						DATA1								DATA0	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA0						FIX_11		CFGCANM	Reserved		STANDYRST	STOPRST	IWDGSW	RDPRT	OBERR

Bit	Name	Access	Description	Reset value
[31:26]	Reserved	RO	Reserved	0
[25:18]	DATA1	RO	Data byte 1	X
[17:10]	DATA0	RO	Data byte 0	X
[9:8]	FIX_11	RO	Fixed to 11	11b

7	USER	CFGCANM	RO	Configuring CAN offline recovery time. 1: Recovery from offline to normal is a bit faster, 0: Recovery from offline to normal in accordance with the CAN protocol	1
[6:5]		Reserved	RO	Reserved	11b
4		STANDY_RST	RO	System reset control in Standby mode, active low.	1
3		STOP_RST	RO	System reset control in Stop mode, active low.	1
2		IWDG_SW	RO	Independent Watchdog Dog (IWDG) hardware enable bit, active low.	1
1	RDPRT		RO	Read protection status. 1: Indicates that the flash memory is currently read-protected.	1
0	OBERR		RO	Selection word error. 1: Indicates a mismatch between the selection word and its inverse code.	0

Note: USER and RDPRT are loaded from the User Option Bytes area after a system reset.

24.3.8 Write Protection Register (FLASH_WPR)

Offset address: 0x20

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

Bit	Name	Access	Description	Reset value
[31:0]	WRP	RO	Flash write-protect status. 1: Write protection disabled; 0: Write protection active. Each bit represents 2K bytes (8 pages) of storage write protection status.	X

Note: The WPR is loaded from the option byte area after a system reset.

24.3.9 Extension Key Register (FLASH_MODEKEYR)

Offset address: 0x24

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

Bit	Name	Access	Description	Reset value
[31:0]	MODEKEYR	WO	Enter the following sequence to unlock the	X

			Quick Program/Erase mode: KEY1 = 0x45670123; KEY2 = 0xCDEF89AB.	

24.4 Flash Operation Procedure

24.4.1 Read Operation

Direct addressing is in the general address space, and the user can access the content of the flash memory module and get the corresponding data through any read operation of 8/16/32-bit data.

24.4.2 Flash Memory Unlock

After the system reset, the flash memory controller (FPEC) and FLASH_CTLR register will be locked and cannot be accessed. The flash memory controller module can be unlocked by writing the sequence to the FLASH_KEYR register.

Unlock sequence:

1) Write KEY1 = 0x45670123 to the FLASH_KEYR register (must operate KEY1 first);

2) Write KEY2 = 0xCDEF89AB to the FLASH_KEYR register (must operate KEY2 secondly).

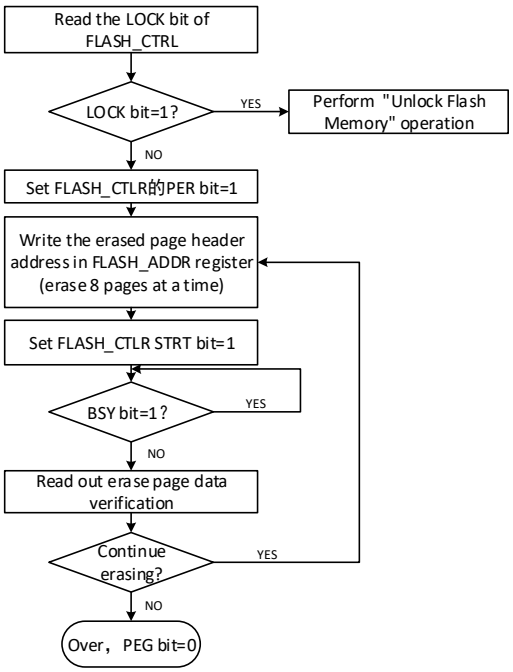
The above operations must be performed sequentially and continuously. Otherwise, it is an error operation, which will lock the FPEC module and FLASH_CTLR register and generate a bus error until the next system reset.

The flash memory controller (FPEC) and the FLASH_CTLR register can be locked again by setting the "LOCK" bit in the FLASH_CTLR register to 1.

24.4.3 Main Memory Standard Erase

Flash memory can be erased in standard pages (2K bytes) or as a whole chip.

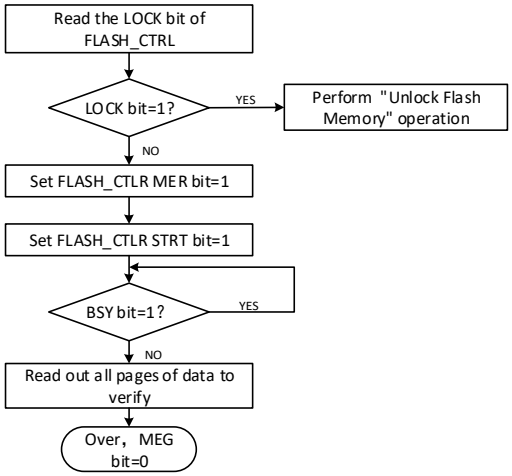
Figure 24-1 FLASH page erasure



1) Check the LOCK bit in the FLASH_CTLR register. If it is 1, you need to perform the "Release Flash Memory

Lock" operation.
2) Set the PEG bit in the FLASH_CTLR register to ‘1’ to enable the standard page erasure mode.
3) Write the page heading address of the page to be erased to the FLASH_ADDR register.
4) Set the STAT bit in the FLASH_CTLR register to ‘1’ to start an erase action.
5) When the BYS bit changes to '0' or the EOP bit in the FLASH_STATR register to be '1', it indicates the end of erasure. Clear the EOP bit to 0.
6) Read the page of erasure page for verification.
7) To erase the standard page continuously, you can repeat steps 3-5 to end erasing and clear the PEG bit to 0.
Note: After erasing successfully, read the word-0xFF.

Figure 24-2 FLASH whole chip erase



1) Check the LOCK bit in the FLASH_CTLR register. If it is 1, you need to perform the "Release Flash Memory Lock" operation.
2) Set the MEG bit in the FLASH_CTLR register to ‘1’ to enable the whole chip erasure mode.
3) Set the STAT bit in the FLASH_CTLR register to ‘1’ to start an erasure action.
4) When the BYS bit changes to '0' or the EOP bit in the FLASH_STATR register to be '1', it indicates the end of erasure. Clear the EOP bit to 0.
5) Read the data of the erasure page for verification.
6) Clear the MER bit to 0.

24.4.4 Fast Programming Mode Unlock

Fast programming mode operation can be unlocked by writing a sequence to the FLASH_MODEKEYR register. After unlocking, the block bit of the FLASH_CTLR register will be cleared to 0, indicating that quick erasure and programming operations can be performed. Lock again by setting the "FLOCK" bit of the FLASH_CTLR register to 1.

Unlock sequence:
1) write KEY1=0x45670123 to the FLASH_MODEKEYR register.
2) write KEY2=0xCDEF89AB to the FLASH_MODEKEYR register.
The above operations must be performed sequentially and continuously, otherwise the error operation will be locked and cannot be unlocked until the next system reset.
Note: Fast programming requires unlocking the "LOCK" and "FLOCK" layers.

24.4.5 Main Memory Fast Programming

The fast programming (256 bytes) is made according to the page.

1) Check the LOCK bit of the FLASH_CTLR register. If it is' 1blank, you need to perform the "unlock flash" operation.
2) Check the FLASH_CTLR register FLOCK bit, if it is' 1percent, you need to perform a "quick programming mode unlock" operation.
3) Check the BSY bit of the FLASH_STATR register to confirm that there are no other programming operations in progress.
4) Set the FTPG bit of the FLASH_CTLR register to '1blank to enable fast page programming mode.
5) Set the BUFRST bit of the FLASH_CTLR register and perform the operation of clearing the internal 256-byte cache.
6) Wait for the BYS bit to become'0' or the EOP bit of the FLASH_STATR register to be'1' means that the erasure is over, and the EOP bit is cleared to 0.
7) Write data to a FLASH address in a 32-bit manner, such as.
* (uint32_t*) 0x80000000=0x12345678.
8) Then set the BUFLOAD bit of the FLASH_CTLR register and perform the load into the cache.
9) Wait for the WR_BSY of the FLASH_STATR register to be '0percent, and write the next data.
10) Repeat steps 7-9 64 times to load 256 bytes of data into the cache (the addresses of the main 64 rounds of operations should be continuous).
11) Write the first address of the quick programming page to the FLASH_ADDR register.
12) Set the STRT bit of the FLASH_CTLR register to '1percent to start fast page programming.
13) Waiting for the BSY bit to change to'0' or the EOP bit of the FLASH_STATR register to'1' means that a quick page programming is completed and the EOP bit is cleared to 0.
14) Query the FLASH_STATR register to see if there are any errors, or read the programming address data check.
15) Continue fast page programming to repeat steps 5-14 and end the programming to clear the FTPG bit 0.

24.4.6 Main Memory Fast Erasure

Fast erase is erased by page (256 bytes).
1) Check the FLASH_CTLR register LOCK bit, if it is 1, you need to perform the "unlock flash memory" operation.
2) Check the FLASH_CTLR register FLOCK bit, if it is 1, you need to perform a "quick programming mode unlock" operation.
3) Check the BSY bit of the FLASH_STATR register to confirm that there are no other programming operations in progress.
4) Set the FTER bit of the FLASH_CTLR register to '1bytes, and turn on the fast page erasure mode function.
5) Write the first address of the quick erase page to the FLASH_ADDR register.
6) Set the STAT bit of the FLASH_CTLR register to '1destroy to start a quick page erase (256byte) action.
7) Wait for the BSY bit to change to'0' or the EOP bit of the FLASH_STATR register to'1' means the erasure is over, and the EOP bit is cleared to 0.
8) Query the FLASH_STATR register to see if there are any errors, or read the erase page address data check.
9) Continue to quickly erase the page and repeat steps 5-8 to end the erase to clear the FTER bit 0.
Note: After erasing successfully, read the word-0xFF.

Fast erase is erased in blocks (32K bytes).
1) Check the FLASH_CTLR register LOCK bit, if it is 1, you need to perform the "unlock flash memory" operation.

2) Check the FLASH_CTLR register FLOCK bit, if it is 1, you need to perform a "fast programming mode unlock" operation.
3) Check the BSY bit of the FLASH_STATR register to confirm that there are no other programming operations in progress.
4) Set the BER32 bit of the FLASH_CTLR register to '1bytes, and turn on the fast block erase (32K bytes) mode function.
5) Write the first address of the quick erase block to the FLASH_ADDR register.
6) Set the STAT bit of the FLASH_CTLR register to '1bytes, and start a fast block erase (32K bytes) action.
7) Wait for the BYS bit to change to'0' or the EOP bit of the FLASH_STATR register to'1' means the erasure is over, and the EOP bit is cleared to 0.
8) Query the FLASH_STATR register to see if there are any errors, or read the erase page address data check.
9) continue to quickly erase the page and repeat steps 5-8 to end the erase to clear the BER32 bit 0.
Note: After erasing successfully, read the word-0xFF.

24.5 User Option Bytes

The User Option Bytes are solidified in FLASH and will be reloaded into the corresponding register after the system reset, and the user can erase and program at will. The user option word information block has a total of 8 bytes (4 bytes for write protection, 1 byte for read protection, 1 byte for configuration options, 2 bytes for user data storage), and each bit has the inverted code bit for checking during loading. The structure and meaning of the selected word information are described below.

Table 24-3 32-bit option bytes format division

[31:24]	[23:16]	[15:8]	[7:0]
Inverse code of option bytes 1	Option bytes 1	Inverse code of option bytes 0	Option bytes 0

Table 24-4 User option bytes information structure

Address / Bit	[31:24]	[23:16]	[15:8]	[7:0]
0x1FFFF800	nUSER	USER	nRDPR	RDPR
0x1FFFF804	nData1	Data1	nData0	Data0
0x1FFFF808	nWRPR1	WRPR1	nWRPR0	WRPR0
0x1FFFF80C	Reserved	Reserved	Reserved	Reserved

Name/Byte	Description	Reset value
RDPR	Read protection control. It configures whether the code in the flash memory can be read. 0xA5: If this byte is 0xA5 (nRDP must be 0x5A), it means that the current code is in a non-read protected state and can be read; Other values: Code read protection status, unreadable; pages 0 to 31 pages (4K) will be automatically write-protected and not controlled by WRPR0.	0xA5

USER	[7:6]	Reserved	Reserved	11b
	5	CFGCANM	Configure CAN offline recovery time: 1: It's faster to get back to normal offline. 0: Restore from offline to normal and comply with CAN protocol	1
	[4:3]	Reserved	Reserved	11b
	2	STANDYRST	System reset control in Standby mode: 1: Do not enable, enter Standby mode system does not reset. 0: Enabled, enter Standby mode to generate system reset.	1
	1	STOPRST	System reset control in Stop mode: 1: Do not enable, enter Stop mode system does not reset. 0: Enable, enter the Stop mode to generate a system reset.	1
	0	IWDGSW	Independent watchdog (IWDG) hardware enable bit: 1: IWDG function is enabled by software, but hardware is not allowed. 0: IWDG function is turned on by the hardware (determined by the LSI clock).	1
Data0–Data1			Stores 2 bytes of user data.	FFFFh
WRPR0 - WRPR3			Write protection control bit. Each bit is used to control the write protection status of 2 sectors (2K bytes / sector) in the main memory: 1: Disable write protection. 0: Enable write protection. 2 bytes are used to protect a total of 64K bytes of main memory. WRPO: Sector 0-15 storage write protection control. WRP1: 16-31 sector storage write protection control. WRP2: Reserved. WRP3: Reserved.	FFFFh

24.5.1 User Option Bytes Unlock

The user option bytes operation can be unlocked by writing the sequence to the FLASH_OBKEYR register. After unlocking, the OBWRE bit in the FLASH_CTLR register will be set to 1, indicating that user option bytes can be erased and programmed. By setting the OBWRE bit in the FLASH_CTLR register, it will be cleared to 0 by software to lock again.

Unlocking sequence

1) Write KEY1 = 0x45670123 to the FLASH_OBKEYR register;

2) Write KEY2 = 0xCDEF89AB to the FLASH_OBKEYR register.

Note: The user needs to unlock the 2 layers: "LOCK" and "OBWRE" for word selection.

24.5.2 User Option Bytes Programming

1) Check the LOCK bit of the FLASH_CTLR register. If it is' 1blank, you need to perform the "unlock flash" operation.

2) Check the FLASH_CTLR register FLOCK bit, if it is' 1percent, you need to perform a "quick programming mode unlock" operation.
3) Check the BSY bit of the FLASH_STATR register to confirm that there are no other programming operations in progress.
4) Set the FTPG bit of the FLASH_CTLR register to '1blank to enable fast page programming mode.
5) Set the BUFRST bit of the FLASH_CTLR register and perform the operation of clearing the internal 256-byte cache.
6) Wait for the BYS bit to become'0' or the EOP bit of the FLASH_STATR register to be'1' means that the erasure is over, and the EOP bit is cleared to 0.
7) Write data to a FLASH address in a 32-bit manner, such as.
* (uint32_t*) 0x1FFFF804=0x5AA55AA5.
8) Then set the BUFLOAD bit of the FLASH_CTLR register and perform the load into the cache.
9) Wait for the WR_BSY of the FLASH_STATR register to be '0percent, and write the next data.
10) Repeat steps 7-9 for a total of 4 times to load all 16 bytes of data into the cache (the main 4 rounds of operation address should be consecutive), and the remaining bytes of the current page use the cache to reset the default value.
11) Write the first address of the quick programming page to the FLASH_ADDR register.
12) Set the STRT bit of the FLASH_CTLR register to '1percent to start fast page programming.
13) Waiting for the BSY bit to change to'0' or the EOP bit of the FLASH_STATR register to'1' means that a quick page programming is completed and the EOP bit is cleared to 0.
14) Query the FLASH_STATR register to see if there are any errors, or read the programming address data check.
15) End the programming to clear the FTPG bit 0.
Note: when "read protected" in the modified selection word is changed to "unprotected" state, an entire erase main storage area operation is automatically performed. If you modify a selection other than read Protection, the whole erase operation will not occur.

24.5.3 User Option Bytes Erasure

Erase the entire 256-byte user option bytes area directly.

1) Check the FLASH_CTLR register LOCK bit, if it is 1, you need to perform the "unlock flash memory" operation.
2) Check the BSY bit of the FLASH_STATR register to confirm that there is no programming operation in progress.
3) Check the OBWRE bit of the FLASH_CTLR register. If it is 0, you need to perform the "user option bytes unlock" operation.
4) Set the Ober bit of the FLASH_CTLR register to '1percent, then set the STAT bit of the FLASH_CTLR register to' 1percent, and turn on erasure of user option bytes.
5) Wait for the BYS bit to change to'0' or the EOP bit of the FLASH_STATR register to be'1' means the erasure is over, and the EOP bit is cleared to 0.
6) Read erase address data check.
7) Clear the OBER bit 0 at the end.
Note: After erasing successfully, read the word-0xFF.

24.5.4 Read Protection Release

Whether the flash memory is read-protected or not is determined by the user option byte. Read the FLASH_OBR register. When the RDPRT bit is'1', it indicates that the current flash memory is in read-protected state, and the flash operation is protected by a series of read-protected states. The process of unprotecting read is as follows:
1) Erase the entire user option byte area, read the protection field RDPR, and the read protection is still valid.

2) The user chooses words to program and writes the correct RDPR code 0xA5 to remove the read protection of the flash memory. (This step will first cause the system to automatically erase the entire piece of flash memory.).
3) A power-on reset is performed to reload the selected bytes (including the new RDPR code), and the read protection is removed.

24.5.5 Write Protection Release

Whether the flash memory is write-protected or not is determined by the user option byte. Read the FLASH_WPR register, each bit represents 2K bytes of flash space, when the bit is'1' for non-write-protected state, for'0' for write-protected. The process of unprotecting a write is as follows:
1) Erase the entire user option byte area.
2) Write the correct RDPR code 0xA5 to allow read access.
3) Perform a system reset, reload the selected bytes (including the new WRPR [3:0] bytes), and write protection is removed.

Chapter 25 Extended Configuration (EXTEN)

25.1 Extended Configuration

The system provides an EXTEN extension configuration unit (EXTEN_CTR register). The unit uses an HB clock and performs a reset action only when the system is reset. It mainly includes the following extended control bit functions:

1) Adjust the core voltage: select the default values for the LDOTRIM and ULLDOTRIM fields, which can be modified when adjusting performance and power consumption.

2) PLL clock selection: the HSIPRE field, together with the original clock configuration register, provides the HSI clock with or without frequency division as the input clock of the PLL.

3) Lock-up function monitoring: if the LKUPEN field is enabled, the Lock-up monitoring of the system will be turned on. In the event of Lock-up, the system will reset and set the LKUPRST field to 1. After reading, you can write 1 to clear this flag.

25.2 Register Description

Table 25-1 EXTEN-related register

Name	Access address	Description	Reset value
R32_EXTEN_CTR	0x40023800	Configuration extended control register	0x00002440

25.2.1 Configure Extended Control Register (EXTEN_CTR)

Offset address: 0x00

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		LDOTRIM		Reserved	ULLDOTRIM			LKUP RST	LKUP EN	Reserved	HSI PRE	Reserved			

Bit	Name	Access	Description	Reset value
[31:14]	Reserved	RO	Reserved	0
[13:12]	LDOTRIM[1:0]	RW	Adjust digital core voltage value, LDO voltage value	10b
11	Reserved	RO	Reserved	0
[10:8]	ULLDOTRIM[2:0]	RW	Adjust ULLDO voltage value in low-power mode	100b
7	LKUPRST	RW1	LOCKUP reset: 1: LOCKUP occurs and causes system reset. Write 1 to clear it. 0: Normal.	0
6	LKUPEN	RW	LOCKUP monitor function: 1: Enable. System reset occurs and set the	1

			LKUPRST bit when lock-up occurs. 0: Disable.	
5	Reserved	RO	Reserved	0
4	HSIPRE	RW	HSI clock: (Only can be written when PLL is disabled.) 1: HSI clock selected as PLL input clock. 0: HSI clock divided by 2 selected as PLL input clock.	0
[3:0]	Reserved	RO	Reserved	0

Chapter 26 Debug Support (DBG)

26.1 Main Features

This register allows the MCU to be configured in the debug state. Includes:

- Counters supporting Independent Watchdog (IWDG)
- Counters supporting Window Watchdog (WWDG)
- Counter supporting timer
- Support for I2CSMBus timeout control
- Support for bxCAN communication

26.2 Register Description

26.2.1 Debug MCU Configuration Register (DBGMCU_CR)

Offset address: 0x7C0(CSR)

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIM4_STOP	TIM3_STOP	TIM2_STOP	TIM1_STOP	I2C2_SMBUS_TIMEOUT	I2C1_SMBUS_TIMEOUT	WWDG_STOP	IWDG_STOP	Reserved					STANDBY	STOP	SLEEP

Bit	Name	Access	Description	Reset value
[31:21]	Reserved	RO	Reserved	0
20	CAN_STOP	RW	CAN debug stop bit. CAN stops when the core enters the debug state. 1: CAN's receive register does not continue to receive data. 0: CAN still operates normally.	0
[19:16]	Reserved	RO	Reserved	0
15	TIM4_STOP	RW	Timer 4 debug stop bit. The counter stops when the core enters the debug state. 1: Timer 4's counter stops working. 0: Timer 4's counter is still working normally.	0
14	TIM3_STOP	RW	Timer 3 debug stop bit. The counter stops when the core enters the debug state. 1: Timer 3's counter stops working. 0: Timer 3's counter is still working normally.	0
13	TIM2_STOP	RW	Timer 2 debug stop bit. The counter stops when the core enters the debug state. 1: Timer 2's counter stops working. 0: Timer 2's counter is still working normally.	0

12	TIM1_STOP	RW	Timer 1 debug stop bit. The counter stops when the core enters the debug state. 1: Timer 1's counter stops working. 0: Timer 1's counter is still working normally.	0
11	I2C2_SMBUS_TIMEOUT	RW	SMBUS timeout mode debug stop bit. Stops SMBUS timeout mode when the core enters debug state. 1: Freezes the SMBUS timeout control. 0: Same as normal mode operation.	0
10	I2C1_SMBUS_TIMEOUT	RW	SMBUS timeout mode debug stop bit. Stops SMBUS timeout mode when the core enters debug state. 1: Freezes the SMBUS timeout control. 0: Same as normal mode operation.	0
9	WWDG_STOP	RW	WWDG debug stop bit. The debug WWDG stops working when the core enters the debug state. 1: WWDG counter stops working. 0: WWDG counter is still working normally.	0
8	IWDG_STOP	RW	IWDG debug stop bit. The debug IWDG stops working when the core enters the debug state. 1: IWDG counter stops working. 0: IWDG counter is still working normally.	0
[7:3]	Reserved	RW	Reserved	0
2	STANDBY	RW	Debug standby mode bit. 1: (FCLK on, HCLK on) the digital circuit does not power down, and the FCLK and HCLK clocks are provided by internal RL oscillators. In addition, the microcontroller exits STANDBY mode by generating a system reset, which is the same as the reset. 0: (FCLK off, HCLK off) the whole digital circuit is powered off. From a software point of view, exiting STANDBY mode is the same as resetting (except that some status bits indicate that the microcontroller has just exited from the STANDBY state).	0
1	STOP	RW	Debug stop mode bit. 1: (FCLK on, HCLK on) in stop mode, the FCLK and HCLK clocks are provided by the internal RC oscillator. When exiting the stop mode, the software must reconfigure the clock system to start PLL, crystal oscillator, etc. (the same operation as configuring this bit to 0). 0: (FCLK off, HCLK off) when in stop mode, the clock controller disables all clocks (including HCLK and FCLK). When exiting from STOP mode, the clock is configured the same as after the reset (the microcontroller is provided by the 8MHz's internal RC oscillator (HSI)).	0

			Therefore, the software must reconfigure the clock control system to start PLL, crystal oscillator and so on.	
0	SLEEP	RW	Debug sleep mode bits. 1: (FCLK on, HCLK on) in sleep mode, both the FCLK and HCLK clocks are provided by the previously configured system clock. 0: (FCLK on, HCLK off) in sleep mode, the FCLK is provided by the previously configured system clock and the HCLK is turned off. Because sleep mode does not reset the configured clock system, the software does not need to reconfigure the clock system when exiting from sleep mode.	0

Chapter 27 Electronic Signature (ESIG)

Electronic Signature contains chip identification information: Flash memory area capacity and unique identification. It is burned into the system storage area of the memory module by the manufacturer when leaving the factory, and can be read through SWD (SDI) or application code.

27.1 Function Description

Flash memory area capacity: Indicates the size that the current chip user application can use.

Unique identification: 96-bit binary code, unique to any microcontroller, users can only read and access it and cannot modify it. This unique identification information can be used as the security password, decryption key, product serial number, etc. of the microcontroller (product) to improve the system security mechanism or indicate identity information.

Users of the above content can perform read access in 8/16/32 bits.

27.2 Register Description

Table 27-1 ESIG-related register

Name	Access address	Description	Reset value
R16_ESIG_FLACAP	0x1FFFF7E0	Flash capacity register	0xXXXX
R32_ESIG_UNIID1	0x1FFFF7E8	UID register 1	0xXXXXXXXX
R32_ESIG_UNIID2	0x1FFFF7EC	UID register 2	0xXXXXXXXX
R32_ESIG_UNIID3	0x1FFFF7F0	UID register 3	0xXXXXXXXX

27.2.1 Flash Capacity Register (ESIG_FLACAP)

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

Bit	Name	Access	Description	Reset value
[15:0]	F_SIZE	RO	Flash memory capacity in Kbyte. Example: 0x0080 = 128 Kbytes	X

27.2.2 UID Register (ESIG_UNIID1)

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

Bit	Name	Access	Description	Reset value
[31:0]	U_ID[31:0]	RO	Bits 0-31 of the UID.	X

27.2.3 UID Register (ESIG_UNIID2)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U_ID[63:48]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U_ID[47:32]															

Bit	Name	Access	Description	Reset value
[31:0]	U_ID[63:32]	RO	Bits 32-63 of the UID.	X

27.2.4 UID Register (ESIG_UNIID3)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U_ID[95:80]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U_ID[79:64]															

Bit	Name	Access	Description	Reset value
[31:0]	U_ID[95:64]	RO	Bits 64-95 of the UID.	X