User Manual

G32R430

Cortex-M52 core beased 32-bit encoder-specific MCU

Version: V1.0

Contents

1 Introduction and Document Description Rules

1.1 Introduction

This user manual provides application developers with all the information about how to use MCU (micro-controller) system architecture, memory and peripherals.

For information about Arm® Cortex®-M52 core, please refer to Arm® Cortex®-M52 technical reference manual; please refer to the corresponding datasheet for detailed data such as model information, dimensions and electrical characteristics of the device; for all MCU series models, please refer to the corresponding datasheet for memory mapping, peripheral existence and their number.

Note that: Zhuhai Geehy Semiconductor Co., Ltd. is hereinafter referred to as "Geehy".

1.2 Document Description Rules

1.2.1 "Register functional description" rules

(1) Control (CTRL) registers are all "set to 1 and cleared to 0 by software", unless otherwise specified.

(2) The control registers are usually followed by verb abbreviations to make a distinction. The verbs can be: EN-Enable, CFG-Configure, D-Disable, SET-Setup and SEL-Select

(3) The status register abbreviation is usually followed by FLG to make a difference.

(4) The value and data registers usually include V, VALUE, D and DATA, which are not followed by verbs, such as xxPSC and CNT.

1.2.2 Full Name and Abbreviation Description of Terms

Table 1 Abbreviation and Description of R/W Modes

R/W mode	Description	Abbreviation
read/write	The software can read and write this bit.	R/W
read-only	The software can only read this bit.	R
write-only	The software can only write this bit, and after reading this bit, the reset value will be returned.	W
read/clear	The software can read this bit and clear it by writing 1. Writing 0 has no effect on this bit.	RC_W1
read/clear	The software can read this bit and clear it by writing 0. Writing 1 has no effect on this bit.	RC_W0

R/W mode	Description	Abbreviation
read/clear by read	The software can read this bit and reading this bit will automatically clear it to 0, and writing this bit is invalid.	RC_R
read/set	The software can read and set this bit, and writing 0 has no effect on this bit.	R/S
read-only write trigger	The software can read this bit and writing 0 or 1 can trigger an event but has no effect on the value of this bit.	RT_W
toggle	The software can reverse this bit only by writing 1, and writing 0 has no effect on this bit.	T
Read Returns 0s	Read returns 0	R-0
Write 1 to set	Write 1 to set	W1S
Write 1 to clear	Write 1 to clear	W1C
Write once	Write once	WOnce
Write Set once	Write set once	WSonce
Write once	Write this bit once	WO

Table 2 Functional Description and Full Name and Abbreviation of Terms of Commonly Used Registers

Full name in English	English abbreviation
Enable	EN
Disable	D
Clear	CLR
Select	SEL
Configure	CFG
Control	CTRL
Controller	C
Reset	RST
Stop	STOP
Set	SET
Load	LD
Calibration	CAL
Initialize	INIT
Error	ERR
Status	STS
Ready	RDY

Full name in English	English abbreviation
Software	SW
Hardware	HW
Source	SRC
System	SYS
Peripheral	PER
Address	ADDR
Direction	DIR
Clock	CLK
Input	I
Output	O
Interrupt	INT
Data	DATA
Size	SIZE
Divider	DIV
Prescaler	PSC
Multiplier	MUL
Period	PRD

Table 3 Full Name and Abbreviation of Modules

Full name in English	English abbreviation
Static Memory Controller	SMC
Reset and Clock Management Unit	RCM
Power Management Unit	PMU
Nested Vector Interrupt Controller	NVIC
External Interrupt /Event Controller	EINT
Direct Memory Access	DMA
Debug MCU	DBG MCU
General-Purpose Input Output Pin	GPIO
Alternate Function Input Output Pin	AFIO
Timer	TMR
Watchdog Timer	WDT
Independent Watchdog Timer	IWDT
Windows Watchdog Timer	WWDT

Full name in English	English abbreviation
Real-Time Clock	RTC
Universal Synchronous Asynchronous Receiver Transmitter	USART
Inter-integrated Circuit Interface	I2C
Serial Peripheral Interface	SPI
Inter-IC Sound Interface	I2S
Quad Serial Peripheral Interface	QSPI
Controller Area Network	CAN
Secure Digital Input and Output	SDIO
Universal Serial Bus Full-Speed Device	USB
Analog-to-Digital Converter	ADC
Comparator	COMP
Cyclic Redundancy Check Calculation Unit	CRC
Float Point Unit	FPU

2 System Architecture

2.1 Full Name and Abbreviation Description of Terms

Table 4 Full Name and Abbreviation Description of Terms

Full name in English	English abbreviation
Advanced High-Performance Bus	AHB
Advanced Peripheral Bus	APB

2.2 Introduction

G32R430 uses the high-performance and low-power Cortex-M52 based on ARM V8.1-M architecture as the core. It is an analog-digital mixed MCU customized for high-precision magneto electric/photoelectric absolute position encoder applications.

2.3 Main Characteristics

(1) Maximum main frequency 128MHz over the full temperature and voltage range

(2) 4.3 CoreMark/MHz and 1.6 DMIPS/MHz

(3) Support ultra-low power modes

(4) Support TMU trigonometric function calculation acceleration interface

(5) Support I/D TCM interface for tightly coupled memory expansion

(6) Support 4KB I/D Cache

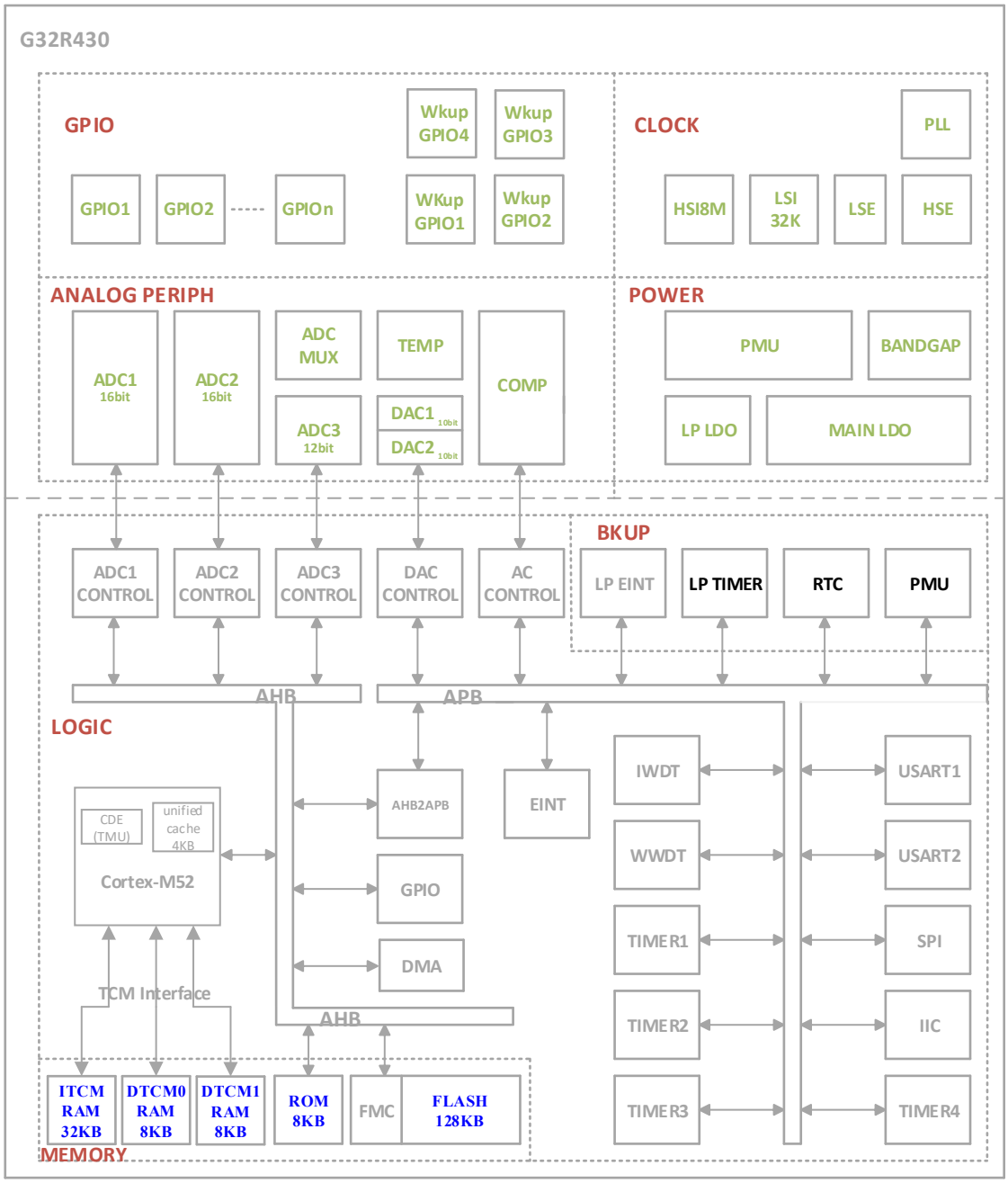
(7) Support 32-bit fixed-point arithmetic

(8) Support nested vector interrupt controller (NVIC)

(9) Support SWD debugging interface

2.4 System Architecture Block Diagram

Figure 1 G32R430 System Architecture Block Diagram



2.5 Address Mapping

Table 5 Memory Mapping and Register Addresses

Bus	Address range	Size	Peripheral name
AHB Bus	0x4002 1800 - 0x4002 1BFF	1KB	GPIO
	0x4002 1400 - 0x4002 17FF	1KB	ADC3（12bit）

Bus	Address range	Size	Peripheral name
	0x4002 1000 - 0x4002 13FF	1KB	ADC2（16bit）
	0x4002 0C00 - 0x4002 0FFF	1KB	ADC1（16bit）
	0x4002 0800 - 0x4002 0BFF	1KB	FLASH Interface
	0x4002 0400 - 0x4002 07FF	1KB	RCM
	0x4002 0000 - 0x4002 03FF	1KB	DMA
APB Bus	0x4000 5800 - 0x4000 5BFF	1KB	DGBMCU
	0x4000 3C00 - 0x4000 3FFF	1KB	TS
	0x4000 3B00 - 0x4000 3BFF	256Bytes	LPTMR
	0x4000 3A00 - 0x4000 3AFF	256Bytes	RTC
	0x4000 3830 - 0x4000 39FF	208Bytes	BKP
	0x4000 3800 - 0x4000 382F	48Bytes	PMU
	0x4000 3400 - 0x4000 37FF	1KB	COMP
	0x4000 3000 - 0x4000 33FF	1KB	DAC2
	0x4000 2C00 - 0x4000 2FFF	1KB	DAC1
	0x4000 2800 - 0x4000 2BFF	1KB	EINT
	0x4000 2400 - 0x4000 27FF	1KB	IWDT
	0x4000 2000 - 0x4000 23FF	1KB	WWDT
	0x4000 1C00 - 0x4000 1FFF	1KB	I2C
	0x4000 1800 - 0x4000 1BFF	1KB	USART2
	0x4000 1400 - 0x4000 17FF	1KB	USART1
	0x4000 1000 - 0x4000 13FF	1KB	SPI
	0x4000 0C00 - 0x4000 0FFF	1KB	TMR4
	0x4000 0800 - 0x4000 0BFF	1KB	TMR3
	0x4000 0400 - 0x4000 07FF	1KB	TMR2
	0x4000 0000 - 0x4000 03FF	1KB	TMR1
Code	0x2000 0000 - 0x2000 3FFF	16KB	DTCM
	0x0800 0000 - 0x0801 FFFF	128KB	FLASH
	0x1FFF 0000 - 0x1FFF 000F	16Bytes	Option Bytes
	0x0010 0000 - 0x0010 1FFF	8KB	System Memory
	0x0000 0000 - 0x0000 7FFF	32KB	ITCM

2.6 Memory Mapping

The assigned addresses of memory mapping include the core (including core peripherals), on-chip Flash (including main memory area, system memory area and option bytes), on-chip SRAM (DTCM), ITCM, and bus peripherals (including AHB and APB peripherals). Please refer to the datasheet of the corresponding model for specific information of various addresses.

2.6.1 TCM

TCM is tightly coupled with the core, provides faster access speeds, and are typically used to store critical codes and data that require high-speed access or low latency. The G32R430 includes 16KB of DTCM and 32KB of ITCM, which can be accessed in byte, half word (16 bits), or full words (32 bits):

- 16KB DTCM, address range: 0x2000 0000 ~ 0x2000 3FFF
- 32KB ITCM, address range: 0x0000 0000 ~ 0x0000 7FFF

These memories can be addressed at the maximum CPU clock frequency with 0 waiting period, and both CPU and DMA can be addressed. When using DMA to address ITCM, an address offset of 0x20004000 must be added. In other words, when addressing the ITCM via DMA, the address range of 0x20004000~0x2000BFFF is the address mapping area of ITCM.

TCM memory is tightly coupled with the core, provides faster access speeds, and is typically used to store critical data that requires high-speed access or low latency.

2.7 Boot Configuration

The G32R430 supports booting from multiple storage areas. Users can select the boot mode by configuring the BOOTADDR [15:0] bits of Option Bytes.

- Boot from main memory
- Boot from BootLoader

Table 6 Startup Mode Configuration and Access Mode

BOOTADDR[15:0] configuration	Boot mode	Access mode
0x4000	Main memory boot (Flash)	Boot from the main memory area
0x0080	System memory	Boot from the system memory area
0x8000	DTCM	Start from DTCM area with an address of 0x20000000
0x0000	ITCM	Start from ITCM area with an address of 0x00000000

Note: To start from BootLoader, users can use serial interface to reprogram the user Flash.

Embedded BootLoader

In embedded BootLoader mode, users can choose to reprogram the Flash through any of the following serial interfaces:

- USART1 (PB6/PB9)
- USART2 (PD5/PD9)

3 Trigonometric Mathematical Unit (TMU)

For information on trigonometric mathematical units (TMUs), please refer to the TMU section in the *G32R430 Data Sheet*.

4 Flash Memory

4.1 Full Name and Abbreviation Description of Terms

Table 7 Full Name and Abbreviation Description of Terms

Full name in English	English abbreviation
Flash Memory Controller	FMC
One-time Programmable	OTP
Flash Accelerator	FACC

4.2 Introduction

The Flash interface manages the access of CPU AHB ICode to Flash. It implements erase and program operations, as well as read/write protection mechanisms for Flash. The flash interface accelerates code execution through instruction prefetching and cache line system.

This chapter mainly introduces the storage structure, read, erase, write, read/write protection, unlock/lock characteristics of Flash, and the involved register functional description.

4.3 Main Characteristics

(1) Flash Memory Structure

The zone division, corresponding NVR, and usage of Flash memory are shown in the following table:

Table 8 FLASH Corresponding Zones

Zone	Size (bytes)	Corresponding NVR	Purpose
Main memory	256 (sector) x 512(byte)=128KB		Main flash memory, storing codes
FLASH TRIM	512byte	NVR0	FLASH TRIM reserved area. Store eflash IP related parameters. When powered on, EFC automatically loads parameters into the eflash IP. Not open to users.
Option byte	512 bytes (first 3 bytes)	NVR1	RDP code protection 1. RDP[7:0] Level 0: 0xAA, without protection Level 1: Values other than 0xCC and 0xAA; the debugging interface cannot access the internal RAM or FLASH program region and Data Flash region;

Zone	Size (bytes)	Corresponding NVR	Purpose
			Level 2: 0xCC; disable the debugging port, and disable BOOT from RAM region When attempting to modify RDP to level 0 under level 1 conditions, the Flash program area will be erased Boot configuration, watchdog enable, and FLASH write protect lock BOOT_ADDR 2. FLASH Main area write protection lock area configuration WLOCK[7:0] The FLASH program area is logically divided into 8 banks. Each bank contains a size of 32 sectors, i.e. 16KB. The space of each bank can be independently set with write and erase protection WLOCK [i] 0: Sector write protection of this area enabled; 1: Sector write protection of this area disabled 3. FLASH Main area write protection lock main switch WLOCKEN 0x33: Disable WLOCK Others: Enable WLOCK 4：IWDTEN/WWDTEN Watchdog enabled 5. PVDEN and PVDVSEL selection
SYSTEM INFO	1K byte	NVR2~3	IP trim parameters, chip information, test mode enable, CP mode enable, FLASHBIST mode enable and other parameter storage areas. After power on, the controller automatically loads the parameter into the register. This area is written and erased through a special sequence.

(2) Functional description

The Eflash controller mainly completes the following functions:

- Read and write protection processing for Flash.
- Realize reading and writing of data between MCU and Flash.
- Perform sector erase and main erase on Flash.
- The main array sector supports sector and main erase.
- NVR sector only supports sector erase.

(3) Automatically load initialization information after power on.

- After power-on reset, the hardware automatically loads NVR0 information to the eflash IP.
- After power-on reset, the hardware automatically loads the option byte information of NVR1 to the eflash IP.
- After power-on reset, the hardware automatically loads the information of NVR2~3 to the system trim related registers.

(4) Flash characteristics:

- Up to 128 kB Flash block
- Flash read operation and 32-bit data width
- Single-page erase and mass erase
- Minimum write/erase cycle is 100000 cycles at T_j=125℃
- Data retention time exceeds 10 years at T_j=125℃

(5) Flash interface characteristics:

- Flash read operation
- Flash write/erase operation
- Read protection (RDP) activated by option bytes
- Two write protection areas selected by option bytes
- Proprietary code read protection area defined by option bytes
- Support prefetch acceleration in ICODE
- Option byte loader
- Low-power mode

4.4 Flash Memory Structure

Figure 2 eflash Address Mapping

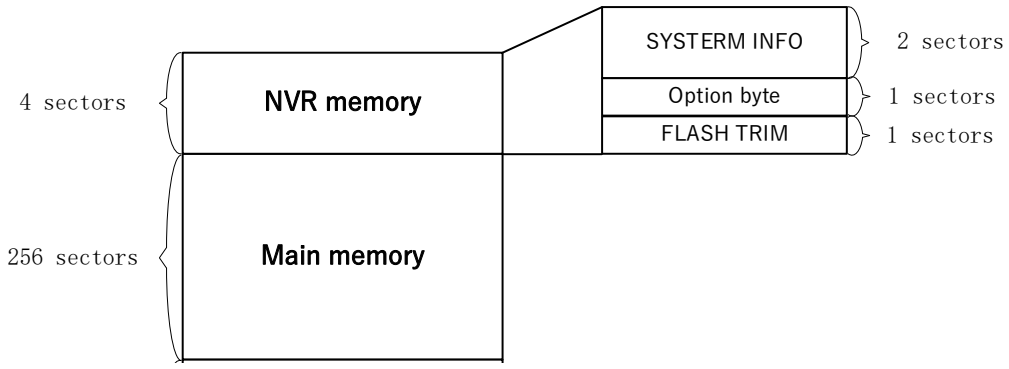


Table 9 Flash Memory Structure

Block	Name	Address range	Size (bytes)	Sector
Main memory block		0x0800 0000 – 0x0800 01FF	512	Sector 0
		0x0800 0200 – 0x0800 03FF	512	Sector 1
		0x0800 0400 – 0x0800 05FF	512	Sector 2
		……	……	……
		0x0801 FE00 – 0x0801 FFFF	512	Sector 255
Information block	Option-bytes (NVR1)	0x1FFF 0000 – 0x1FFF 001F	32	-
	SYSTERM INFO (NVR2~NVR3)	0x0802 0000~0x0802 03FF	1K	-

Block	Name	Address range	Size (bytes)	Sector
	FLASH TRIM	0x1FFF 0200~0x1FFF 03FF	512	-

Figure 3 Module Block Diagram

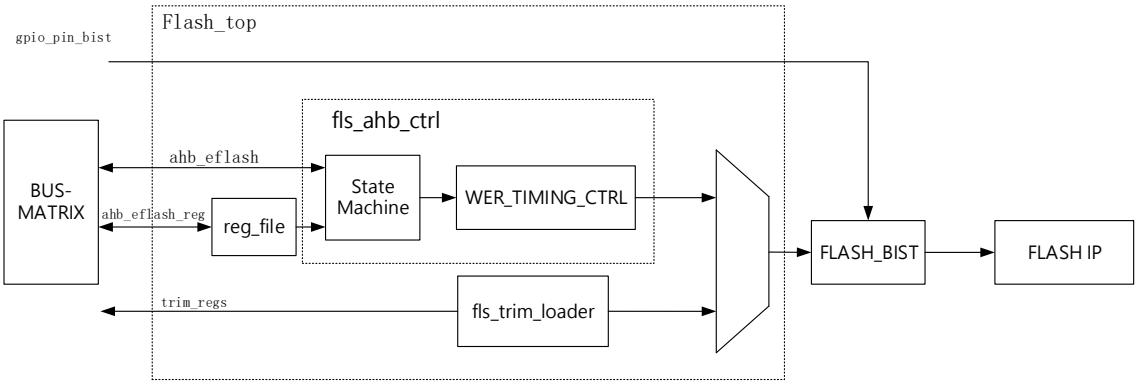
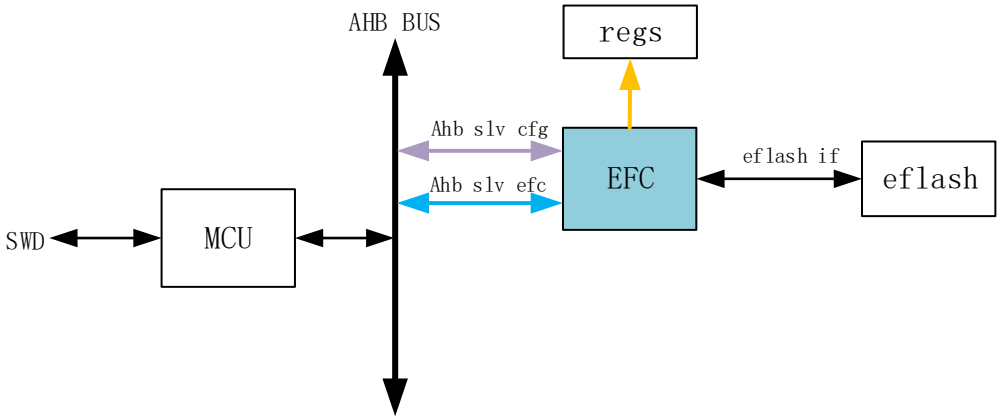


Figure 4 Data Flow Block Diagram



4.5 Function Description of FLASH Memory

4.5.1 Read Flash

The relationship between Flash wait cycle setting and fHCLK:

- 0~32MHz: 0 wait cycle
- 32~64MHz: 1 wait cycles
- 64~96MHz: 2 wait cycles
- 96~128MHz: 3 wait cycles

When increasing the main frequency, it is necessary to first configure the RDWAIT[3:0] bits of the FLASH_CR2 register, and then increase the frequency. On the contrary, when decreasing the main frequency, it is necessary to first decrease the frequency and then configure the RDWAIT[3:0] bits.

4.5.2 Main memory block

During erase/write operations to the main storage area, read operations cannot be performed on the Flash, and the main storage block is erased.

Table 10 Erasure Methods for Different Areas

Main flash	Option bytes(NVR1)
Main/block/Sector	Not supported

4.5.2.1 **Main erase**

This operation only targets the main area. The steps are as follows:

(1) Check that the BUSY bit of the FLASH_SR register is 0 to confirm that the last programming operation has ended

(2) Write the unlock sequence (0x96969696+0x3C3C3C3C) to the FLASH_KEY register to unlock the main area of the Flash

(3) Wait for the MAINUNLOCK bit of the FLASH_SR register to be set to 1 by hardware

(4) Set the ERTYPE[1:0] bit of FLASH_CR1 register to 10 or 11 (Main Erase)

(5) Set the EREQ bit of FLASH_CR1 register to 1

(6) Write 0xA5A5 to any address in the Main area of FLASH

(7) Wait for the BUSY bit of FLASH_SR register to be 0 and the ERD bit to be 1

(8) MCU reads all pages and verifies them

4.5.2.2 **Sector erasure in the main area**

(1) Check the BUSY bit of the FLASH_SR register to confirm that the last programming operation has ended

(2) Write the unlock sequence (0x96969696+0x3C3C3C3C) to the FLASH_KEY register to unlock the main area of the Flash

(3) Wait for the MAINUNLOCK bit of the FLASH_SR register to be set to 1 by hardware

(4) Set the ERTYPE[1:0] bit of FLASH_CR1 register to 00 (Sector Erase)

(5) Set the EREQ bit of FLASH_CR1 register to 1

(6) Write 0xA5A5 to any address in the Main area of FLASH

(7) Wait for the BUSY bit of FLASH_SR register to be 0 and the ERD bit to be 1

(8) MCU reads all pages and verifies them

4.5.2.3 **Write Main Memory Block**

(1) Check that the BUSY bit of the FLASH_SR register is 0 to confirm that the last programming operation has ended

(2) Write the unlock sequence (0x96969696+0x3C3C3C3C) to the FLASH_KEY register to unlock the main area of the Flash

(3) Wait for the MAINUNLOCK bit of the FLASH_SR register to be set to 1 by hardware

(4) Set the PREQ bit of FLASH_CR1 register to 1

(5) Write any data (word) to the target address in the Main area of FLASH

(6) Wait for the BUSY bit of FLASH_SR register to be 0 and the PRD bit to be 1

(7) MCU reads the address data and verifies it

4.5.2.4 **Lock/Unlock**

Table 11 eflash Unlock Collection

		Main flash	**Option bytes0(NVR1)**
FLASH_KEY	KEY1	0x9696_9696	
	KEY2	0x3C3C_3C3C	
FLASH_OPTKEY	KEY1		0xAAAA_BBBB
	KEY2		0x4455_6677

Before performing erase/write operations on the Flash, a KEY verification is required. Each KEY has a sequential write detection state machine. If the write sequence is incorrect or an incorrect value is written, or if an erase or program operation is performed to Flash before the FLASH_KEY/ FLASH_OPTKEY registers are correctly verified, it will enter an error state and generate a corresponding interrupt.

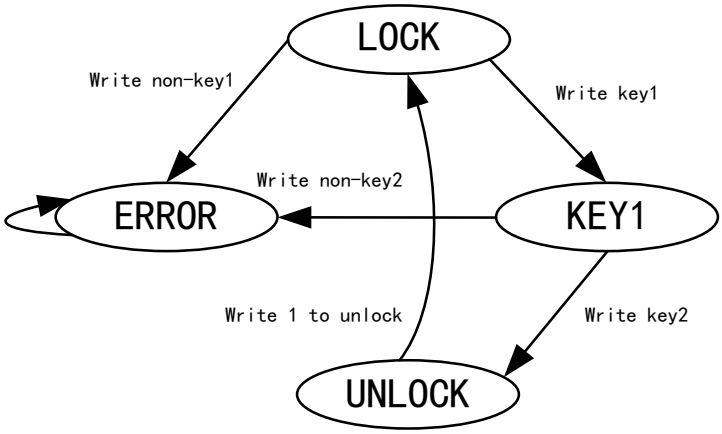
Example: The correct sequence for the main storage area is to first write 0x9696_9696 and then write 0x3C3C_3C3C to the FLASH_KEY register. Where, write sequence error refers to reverse sequence, and write value error refers to writing other values. An error will set the KEYERR bit of the status register.

After a FLASH_KEY/FLASH_OPTKEY register verification error, erase/write operations to the corresponding area will be disabled until the next reset.

Setting the SYSINFOUNLOCK, OPUNLOCK and MAINUNLOC bits of the

FLASH_SR register to 1 will relock the corresponding areas. The state conversion is shown in the following figure:

Figure 5 State Conversion



4.5.2.5 **Interrupt**

An interrupt will occur in any of the following events:

Table 12 FLASH Interrupt Request

Interrupt event	Event flag	Enable control bit
Operation ends	PRD/ERD	EOPIE
Write protection error	WRPRTERR	ERRIE
KEY error	KEYERR	ERRIE
Write and erase error	WADRERR	ERRIE

When the enable control bit is set to 1 and a corresponding interrupt event occurs, an interrupt is generated and the event flag bit is set to 1.

4.5.3 Option byte

All option bytes are stored in the first four 32 bytes of the NVR1 area. Please see the FLASH_OBSR register for default values of option bytes.

Table 13 Options Byte Format

31-24	23-16	15-8	7-0
Option byte 1 complement	Option byte 1	Option byte 0 complement	Option byte 0

Table 14 Instructions for Option Bytes

Address	Field	Option byte	Functional description
0x1FFF 0000	7:0	RDP	Level 0: Unprotected RDP=0xAA nRDP=0x55 Level 1 (default): RDP code protection for any value other than (RDP=0xAA nRDP=0x55) or (RDP=0xCC nRDP=0x33) Level 2: Highest protection level, RDP=0xCC nRDP=0x33
	15:8	nRDP	RDP complement
	23:16	USER	USER[0]: Independent watchdog enable; 0: Watchdog hardware-enabled 1: Requires software enable USER[1]: Window watchdog enable; 0: Watchdog hardware-enabled 1: Requires software enable USER[2]: WLOCK write protection master switch; 0: Disable 1: Enable USER[3] : NMI_DIS 0: After the chip is reset, the NMI pin function is disabled 0: After the chip is reset, the NMI pin function is enabled USER[6:4] : Reserved USER[7] : ADTSLOAD ADCtrim load select configuration bit 0: Disable and trim is not loaded 1: Enable and load trim
	31:24	nUSER	USER complement
0x1FFF 0004	7:0	WLOCK[7:0]	Each bit corresponds to a 32-section region. 0: Write protection enabled for this region 1: Write protection disabled for this region
	15:8	nWLOCK[7:0]	WLOCK complement
	23:16	Reserverd	-
	31:24	Reserverd	-

Address	Field	Option byte	Functional description
0x1FFF C008	7:0	BOOTADDR[7:0]	BOOT_ADD0[15:0] corresponds to the system address BOOT_ADD0[15:0], with missing bits being 0; The boot base address only supports the address range from 0x0000 0000 to 0x0000 0000, with a granularity of 8KB. BOOT_ADD0 = 0x0080,0x0010_0000: Boot from system memory (0x00100000) BOOT_ADD0 = 0x4000: Boot from Flash (0x0800 0000) BOOT_ADD0 = 0x8000: Boot from DTCM RAM (0x2000 0000)
	15:8	nBOOT_ADDR[7:0]	BOOT_ADDR[7:0] complement
	23:16	BOOT_ADDR[15:8]	As above
	31:24	nBOOT_ADDR[15:8]	BOOT_ADDR[15:8] complement

Note: If the original code and its complement do not match, they will be judged for matching on a byte-by-byte basis. If a mismatch is found, the byte is treated as invalid (the default value is loaded), and the TRIMERR bit is set.

After each system reset, the option byte loader (OBL) reads the information block data and stores it in the corresponding option byte registers: FLASH_OPSR1 and FLASH_OPSR2. Each option byte has its complement value stored in the information block for the purpose of verifying the correctness of the option byte. During the option byte loading process, the hardware checks the correction of the option bytes. If an option byte does not match its complement, an Option Byte Error (OPTERR) is generated.

4.5.3.1 **Erase/Write option byte**

The option byte must be unlocked before operations. Option byte programming

(1) Read the RDP (address 0x1FFF 0000) in the option_bytes (NVR1) area of FLASH

(2) Check that the BUSY bit of the FLASH_SR register is 0 to confirm that the last programming operation has ended

(3) Write an unlock sequence (0xAAAABBBB+0x44556677) to 0xAAAABBBB+0x44556677 to unlock the Option bytes (NVR1) area of the flash

(4) Check that the OPUNLOCK bit of FLASH_SR is set to 1 by hardware

(5) Write the cached bits for the corresponding option_byte to the FLASH_OBCR1 register and FLASH_OBCR2 register

(6) Set the OPBWREQ bit of FLASH_OBCR1 register to 1

(7) The hardware automatically erases the NVR1 area, and then updates the cached bit information of FLASH_OBCR20/1 to option_byte

(8) Wait for the BUSY bit of FLASH_SR register to be 0 and the OPBPRD bit to be 1

(9) Power on the system again or set the OPRELOAD bit of FLASH_CR1 register to 1 (writing 1 to this bit requires writing 0xA5A5 to the upper 16 bits of FLASH_CR1 at the same time to succeed), and the status of FLASH_OBSR1 and FLASH_OBSR2 registers will be updated

4.5.3.2 **Option byte erase**

Individual sector erasure of option bytes is not supported to prevent RDP from being rewritten.

4.5.3.3 **Lock/Unlock**

The OPUNLOCK bit of the OPUNLOCK register defaults to 1 to lock the option byte area.

Write the keywords 0xAAAA_BBBB and 0x4455_6677 into the FLASH_OPTKEY register. When the system detects the unlock sequence, it will clear the OPUNLOCK bit to zero, and the option byte will be unlocked.

Setting the OPUNLOCK bit in the FLASH_SR register to 1 will relock the option byte area.

4.5.4 Write protection

Write protection is constrained by the boot mode, read protection level, and WLOCK and WLOCKEN bits of the FLASH_OPTSR1 register.

After power-on, the eflash controller automatically loads WLOCK information from the option byte area into the WLOCK and WLOCKEN bits of the FLASH_ OPTSR1 register.

According to the "Comparison Table of Protection Status, Protection Level, and Operating Mode", the system performs write protection under the premise of read protection.

Example: A user program performs read, erase, and write operations across the entire main memory area. However, before erasing or writing, the WLOCK content must be checked. If the area is write-protected, erase and write operations will fail and the WRPRTERR bit will be set to 1. If a main erase operation is attempted on the main flash program area while WLOCK indicates write protection, the erase will not succeed, and the WRPRTERR bit will be set to 1.

4.5.5 Read Protection

In order to prevent untrusted code from reading Flash data, you can choose to

use the read protection function for the Flash and the read protection level can be selected by configuring the value of the RDP bit. The read protection has three levels, namely, Level 0, Level 1 and Level 2.

The access restriction at different read protection levels is shown in the following table.

Table 15 Comparison Table of Protection Status, Protection Level, and Operating Mode

Zone	RDP protection level	User code execution			Debugging, DTCM startup		
		Read	Write	Erase	Read	Write	Erase
Main flash/Data Flash	0	Y	Y	Y	Y	Y	Y
	1	Y	Y	Y	N	N	N
	2	Y	Y	Y	N	N	N
SYSTEM INFO	0	Y	Y3	Y3	Y	Y3	Y3
	1	Y	N	N	Y	N	N
	2	Y	N	N	N	N	N
Option bytes	0	Y	Y	N4	Y	Y1	N4
	1	Y	Y	N4	Y	Y1	N4
	2	Y	Y	N4	N	N	N
FLASH TRIM	0	Y	N	N	Y	N	N
	1	Y	N	N	Y	N	N
	2	Y	N	N	Y	N	N

Note:

(1) When the encryption level is changed from Level 1 to Level 0, the main flash will be automatically erased.

(2) When Level 2 protection is enabled, the debugging port is disabled.

(3) Individual erasure of Option byte is not supported.

(4) Under Level 1 protection, booting from DTCM cannot read or modify Flash and BKP register contents. Under Level 2 protection, the debugger cannot read Flash or BKP registers, nor read and write DTCM.

4.5.5.1 **Level 0: Unprotected**

When RDP=0xAA and nRDP=0x55, read protection is disabled for the Flash.

4.5.5.2 **Level 1: Read protection**

When RDP=any value (except 0xAA and 0xCC), read protection is set to Level 1. If the protection level is reverted to Level 0, a mass erase is executed, erasing all data in the main memory block. Mass erase only affects user code area, and write-protected other option bytes and OTP will not be affected.

4.5.5.3 **Level 2: No debug function**

When RDP=0xCC and nRDP=0x33, read protection is set to Level 2. Then:

- Reserve the read protection function of Level 1
- It is not allowed to bootstrap from RAM or system memory
- SWV, ETM and boundary scan are disabled

Note: When the read protection is set to Level 2, it can no longer be degraded.

4.5.6 Low-power mode

The Flash IP provides a Deep Powerdown (DPD) mode to reduce power consumption. The FLASH-IP enters low-power mode under the following condition.

When the PMU controls the MCU to enter stop mode, the hardware automatically gates the clock of the FLASH controller, forcing the FLASH into deep power-down mode. When a wake-up source triggers the system to exit low-power mode, the FLASH will automatically exit deep power-down mode.

4.6 Register Address Mapping

Table 16 FMC Register Address Mapping

Register name	Description	Offset address
FLASH_KEY	Key register	0x00
FLASH_OPTKEY	Option byte keyword register	0x04
FLASH_SR	Status register	0x08
FLASH_CR1	Control register 1	0x0C
FLASH_CR2	Control register 2	0x10
FLASH_TMCR	Time base control register	0x18
FLASH_OBSR1	Option byte status register 1	0x20
FLASH_OBSR2	Option byte status register 2	0x24
FLASH_OBCR1	Option byte control register 1	0x28
FLASH_OBCR2	Option byte control register 2	0x2C

4.7 Register Functional Description

4.7.1 Key register (FLASH_KEY)

Offset address: 0x00

Reset value: 0x0000 0000

Field	Name	Description
31:0	FKEY	Flash Key When unlocking, this key needs to be input into this register.

4.7.2 Option byte keyword register (FLASH_OPTKEY)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	Description
31:0	OPTKEY	Option Key When unlocking, this key needs to be input into this register.

4.7.3 Status register (FLASH_SR)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	Description
31:19	Reserved	
18	OPUNLOCK	Option byte area lock flag Write the correct Flash unlock values to FLASH_OPTKEY sequentially. After (0xAAAABBBB+0x44556677), this bit is set to 1 by hardware, and only then are erasing and programming operations allowed on the option byte area. This bit can only be set to 1 by hardware. Writing 1 to this bit via software can clear this bit to zero, and then the option byte area can be relocked.
17	MAINUNLOCK	Main program area lock flag Write the correct Flash unlock values to FLASH_KEY sequentially. After (0x96969696+0x3C3C3C3C), this bit is set to 1 by hardware, and only then is it allowed to erase and program the Flash main program storage area. This bit can only be set to 1 by hardware. Writing 1 to this bit via software can clear this bit to zero, and then the FLASH can be relocked.
16	TRIMERR	Trim check error This bit is set to 1 by hardware when a forward complement code check error or an error that does not comply with the fixed data is defected in trim information during TRIMLOAD process. Writing 1 to this bit by software can clear it to 0. (Applicable only to the FLS_TRIM area)
15	OPTERR	Option_byte forward and complement code check error When option_byte does not comply with the forward and complement code check, this bit is set to 1 by hardware. Writing 1 to this bit by software can clear it to 0.
14	OPBPRD	Option_byte programming completion flag When option_byte completes programming, this bit is set to 1 by hardware. Writing 1 to this bit by software can clear it to 0.
13:6	Reserved	
5	WADRERR	Write And Erase Error During an erasing or programming operation, if an error occurs when writing data to the target address, this bit is set to 1 by hardware, and this erasing or programming operation will not take effect. An error will occur in any of the following situations: 1. If the target area unlocked by FLASH KEY is not unlocked or the unlocked area does not match the actual write area, this bit is automatically set to 1 by hardware.

Field	Name	Description
		2. If the target address for FLASH programming is option_byte0, but the PREQ bit of the FLASH_CR1 register is not set to 1, this bit is automatically set to 1 by hardware. 3. If debugging (SWD) modifies a non-option_byte area (excluding chip erase operations) when RDP is at Level 1, this bit is automatically set to 1 by hardware. 4. If user code execution attempts to modify option byte0, this bit is automatically set to 1 by hardware. 5. If user code execution attempts to perform chip erase, this bit is automatically set to 1 by hardware. 6. During an erase operation, if the last data written to the target address is not 0xA5A5, this bit is automatically set to 1 by hardware. Writing 1 to this bit by software can clear it to 0.
4	WRPRTERR	Write Protection Error When an erase/write operation is operated on the Flash write protection area, this bit is set to 1 by hardware. Writing 1 to this bit by software can clear it to 0. When ERRIE is enabled, setting this bit to 1 by hardware will generate an interrupt.
3	KEYERR	Flash Key Error When a KEY error occurs, this bit is set to 1 by hardware. Writing 1 to this bit by software can clear it to 0.
2	BUSY	Busy When Flash is performing an operation, this bit is set to 1 by hardware. When Flash is idle, this bit is set to 0 by hardware.
1	PRD	Program Done After programming is completed, this bit is set to 1 by hardware. Writing 1 to this bit by software can clear it to 0.
0	ERD	Erase Done After erasing is completed, this bit is set to 1 by hardware. Writing 1 to this bit by software can clear it to 0.

4.7.4 Control register 1 (FLASH_CR1)

Offset address: 0x0C

Reset value: 0x0000 0000

Field	Name	Description
31:16	Reserved	
15	OPRELOAD	Option byte force update This bit can only be written as 1 when the data written in the higher 16 bits at the same time is 0xA5A5. When written as 1, this bit will force the option byte to reload, and this operation will trigger a system reset. 0: Invalid 1: Valid
14:6	Reserved	
5	ERRIE	Error interrupt occur

Field	Name	Description
		Under any of error conditions including WADRERR, WRPRTERR, and KEYERR, enabling this bit will generate an interrupt. 0: Disable 1: Enable
4	EOPIE	Program or erase to end the interrupt enable When the PRD or ERD bit of the FLASH_SR register is set to 1 by hardware, it indicates the end of programming or erasing. Enabling this bit will generate an interrupt. 0: Disable 1: Enable
3:2	ERTYPE[1:0]	Flash erase type configure 00: Sector erase 01: Block erase (8 sectors, a total of 4KB) 1x: Main storage area erase Note: When the BUSY bit of FLASH_SR register is set to 1, this register cannot be written.
1	PREQ	Program Request It is set to 1 by software, and can be automatically cleared to zero after the hardware completes programming Note: When both EREQ and PREQ are set to 1, the hardware considers it as an EREQ request.
0	EREQ	Erase Request It is set to 1 by software, and can be automatically cleared to zero after the hardware completes erasing. Note: When both EREQ and PREQ are set to 1, the hardware considers it as an EREQ request.

4.7.5 Control register 2 (FLASH_CR2)

Offset address: 0x10

Reset value: 0x0000 0001

Field	Name	Description
31:4	Reserved	
3:0	RDWAIT[3:0]	Read Flash-wait time setting Related to the main frequency 0000: No wait required to read flash 0001: Wait for 1 system cycle to read flash 0010: Wait for 2 system cycles to read flash 0011: Wait for 3 system cycles to read flash 0100: Wait for 4 system cycles to read flash 0101: Wait for 5 system cycles to read flash 0110: Wait for 6 system cycles to read flash ... 1111: Wait for 15 system cycles to read flash Note: If SETUPMODE[1] is 0, and RDWAIT writes the value 0000, the internal control of RDWAIT is protected to 1111, and the value read from the bus remains the original value. Main frequency ≤32MHz: RDWAIT=0 32MHz<Main frequency≤50MHz：RDWAIT=1

4.7.6 Time base control register (FLASH_TMCR)

Offset address: 0x18

Reset value: 0x0000 00FF

Field	Name	Description
31:8	Reserved	
7:0	UNIT[7:0]	FLASH erasure time base control Flash 1us unit counter (subtracting 1 configuration) 00000000: 1 cycle 00000001: 2 cycles …… 00000111: 8 cycles …… 11111111: 256 cycles With 128MHz as the reference, 1us requires 128 Cycles, subtracting 1 configuration sets UNIT bit to 0x7F; in actual use, it can be adjusted according to the eFlash clock frequency (the clock cannot be lower than 1MHz in FLASH erase state).

4.7.7 Option byte status register 1 (FLASH_OBSR1)

Offset address: 0x20

Reset value: 0x00FF 03AA

Field	Name	Description
31:24	Reserved	
23:16	WLOCK	Write protection configuration Each bit corresponds to a storage area 0: Enable write protection for this area 1: Disable write protection for this area bit0: Corresponds to the address 0x0800_0000~0x0800_3FFF bit1: Corresponds to the address 0x0800_4000~0x0800_7FFF bit2: Corresponds to the address 0x0800_8000~0x0800_BFFF bit3: Corresponds to the address 0x0800_C000~0x0800_FFFF bit4: Corresponds to the address 0x0801_0000~0x0801_3FFF bit5: Corresponds to the address 0x0801_4000~0x0801_7FFF bit6: Corresponds to the address 0x0801_8000~0x0801_BFFF bit7: Corresponds to the address 0x0801_C000~0x0801_FFFF
15	ADTSLOAD	ADC TRIM value loading configuration 0: Software loading 1: Hardware loading
14:12	Reserved	
11	NMIDIS	NMI pin enable configuration 0: NMI pin disabled after the chip exits reset 1: NMI pin enabled after the chip exits reset
10	WLOCKEN	WLOCK write protection main switch 0: Disable 1: Enable
9	WWDTSW	Window watchdog enable 0: Hardware enabling watchdog

Field	Name	Description
		0: Software enabling watchdog
8	IWDTSW	Independent watchdog enable 0: Hardware enabling watchdog 0: Software enabling watchdog
7:0	RDP	Protection class Level 0: No protection, RDP=0xAA nRDP=0x55 Level 1 (default): RDP code protection for any value other than (RDP=0xAA nRDP=0x55) or (RDP=0xCC nRDP=0x33) Level2: Highest protection level, RDP=0xCC nRDP=0x33

4.7.8 Option byte status register 2 (FLASH_OBSR2)

Offset address: 0x24

Reset value: 0x0000 4000

Field	Name	Description
31:16	Reserved	
15:0	BOOTADDR	Boot base address BOOTADDR[15:0] corresponds to {bit29, bit[27:13]} of the system address, with missing parts all being 0. The base address only supports the address range from 0x00100000 to 0x00100000, with a granularity of 8KB. For example, BOOTADDR= 0x0080, corresponding 0x0010_0000: Boot from the system memory boot loader (0x0010 0000) BOOTADDR = 0x4000: Boot from Flash (0x0800 0000) BOOTADDR = 0x8000: Boot from DTCM RAM (0x2000 0000)

4.7.9 Option byte control register 1 (FLASH_OBCR1)

Offset address: 0x28

Reset value: 0x0000 0300

Field	Name	Description
31:25	Reserved	
24	OPBWREQ	OPT Request
23:16	WLOCKBF	Write protection configuration writes cache Each bit corresponds to a storage area 0: Write protection enabled for this area 1: Write protection disabled for this region bit0: Corresponds to the address 0x0800_0000~0x0800_3FFF bit1: Corresponds to the address 0x0800_4000~0x0800_7FFF bit2: Corresponds to the address 0x0800_8000~0x0800_BFFF bit3: Corresponds to the address 0x0800_C000~0x0800_FFFF bit4: Corresponds to the address 0x0801_0000~0x0801_3FFF bit5: Corresponds to the address 0x0801_4000~0x0801_7FFF bit6: Corresponds to the address 0x0801_8000~0x0801_BFFF bit7: Corresponds to the address 0x0801_C000~0x0801_FFFF
15	ADTSLOADWBF	ADC TRIM value loading configuration 0: Software loading 1: Hardware loading
15:12	Reserved	

Field	Name	Description
11	NMIDISWBF	NMI pin enable configuration 0: NMI pin disabled after the chip exits reset 1: NMI pin enabled after the chip exits reset
10	WLOCKENBF	WLOCK write protection master switch write cache enabled 0: Disable 1: Enable
9	WWDTSWBF	Window watchdog enable write cache 0: Hardware enabling watchdog 0: Software enabling watchdog
8	IWDTSWBF	Independent watchdog enable write cache 0: Hardware enabling watchdog 0: Software enabling watchdog
7:0	RDPWBF	Write cache for protection level Level 0: No protection, RDP=0xAA nRDP=0x55 Level 1 (default): RDP code protection for any value other than (RDP=0xAA nRDP=0x55) or (RDP=0xCC nRDP=0x33) Level2: Highest protection level, RDP=0xCC nRDP=0x33

4.7.10 Option byte control register 2 (FLASH_OBCR2)

Offset address: 0x2C

Reset value: 0x0000 4000

Field	Name	Description
31:16	Reserved	
15:0	BOOTADDRBF	Boot base address cache BOOTADDR[15:0] corresponds to {bit29, bit[27:13]} of the system address, with missing parts all being 0. The base address only supports the address range from 0x00100000 to 0x00100000, with a granularity of 8KB. For example, BOOTADDR= 0x0080, corresponding 0x0010_0000: Boot from the system memory boot loader (0x0010 0000). BOOTADDR = 0x4000: Boot from Flash (0x0800 0000) BOOTADDR = 0x8000: Boot from DTCM RAM (0x2000 0000)

5 Reset and clock (RCM)

5.1 Full Name and Abbreviation Description of Terms

Table 17 Full Name and Abbreviation Description of Terms

Full name in English	English abbreviation
Reset and Clock Management	RCM
Reset	RST
Power-On Reset	POR
Power-Down Reset	PDR
High Speed External Clock	HSECLK
Low Speed External Clock	LSECLK
High Speed Internal Clock	HSICLK
Low Speed Internal Clock	LSICLK
Phase Locked Loop	PLL
Main clock output	MCO
Calibrate	CAL
Trim	TRM
Clock Security System	CSS
Non Maskable Interrupt	NMI

5.2 Reset Management Unit (RMU)

The reset is divided into three forms, namely, system reset, power reset and backup area reset.

5.2.1 System reset

5.2.1.1 **"System reset" reset source**

The reset source is divided into external reset source and internal reset source.

External reset source:

- Low level on NRST pin.

Internal reset source:

- Window watchdog termination count (WWDT reset)
- Independent watchdog termination count (IWDT reset)
- Software reset (SW reset)
- Option byte loading reset (the system reset generated by option byte loading will not reset the registers of FLASH-Controller)

- Abnormal system reset caused by CPU deadlock

A system reset will occur when any of the above events occurs. Besides, the reset event source can be identified by viewing the reset flag bit in RCM_RSTCSR (control/status register).

When the system is reset, all registers except the registers in RCM_RSTCSR (control/status register) reset flag bit and backup area will be reset to the reset state.

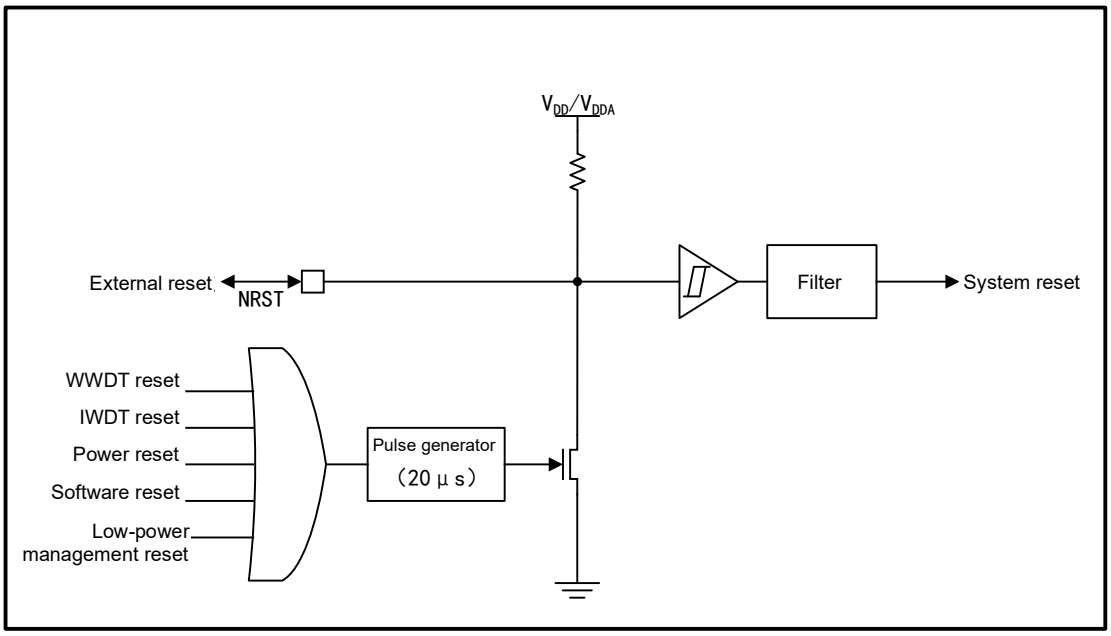
5.2.1.2 **"System reset" reset circuit**

The reset source is used in the NRST pin, which remains low in reset process.

The internal reset source generates a delay of at least 20 μs pulse on the NRST pin through the pulse generator, which causes the NRST to maintain the level to generate reset; the external reset source directly pulls down the NRST pin level to generate reset.

The "system reset" reset circuit is shown in the figure below.

Figure 6 "System Reset" Reset Circuit



5.2.2 Power reset

"Power reset" reset source

"Power reset" reset source is as follows:

- Power-on reset (POR)
- Brown-out reset (BOR)
- Exit (Wake up) from standby mode

A power reset will occur when any of the above events occurs.

Power reset will reset all registers except that in backup area.

5.2.3 Backup domain reset

"Backup domain reset" reset source

"Backup domain reset" reset source is as follows:

- Software reset, set the BDRST bit in the BKP_CR1 register to 1
- After V_{DD} is powered down, V_{DD} is powered on again, resulting in a backup domain reset

A backup domain reset will occur in case of any of the above events.

The BKP domain has two specific resets. When any of the following events occurs, a BKP domain reset will be generated. This reset will reset all registers in the BACKUP domain, including all registers for BKPC/RTC/LPTMR.

5.2.4 Reset peripheral module

With the exception of LPTMR and RTC in the BACKUP domain, which share the BDRST bit in the BKP_CR1 register for software reset, all other AHB and APB peripherals have corresponding software reset control bits. For details, see the description of the RCM_AHBRST and RCM_APBRST registers.

5.3 Clock Management Unit (CMU)

The entire system supports three main clock sources:

- HSICLK: 8MHz low-speed and high-speed RC oscillator
- HSECLK: External high-speed crystal oscillator
- PLL: Up to 128MHz

A 32KHz low-speed internal RC (LSI) provides clocks for independent watchdog, low-power timer LPTMR, and RTC. A 32.768KHz external crystal oscillator (LSE) can provide clocks for RTC, and the BKP is provided with clocks by HSICLK.

Each clock source can be independently enabled or disabled. All peripheral clocks have corresponding register bits to control their enabled or disabled status. Once the independent watchdog (IWDT) is enabled, even if its module clock control bit is disabled, only the IWDT registers become inaccessible, and its work clock cannot be turned off.

FCLK is free-running clock of Cortex-M52. See relevant Arm manuals for details. For the characteristics of the clock source, please refer to the relevant chapter of "Electrical Characteristics" in the datasheet.

5.3.1 External clock source

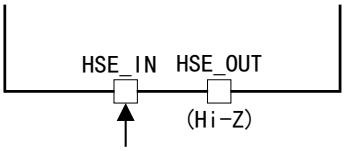
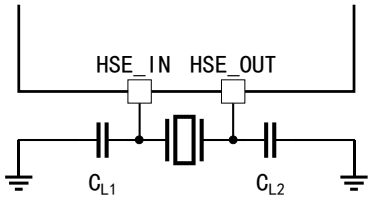
The external clock signal includes HSECLK (high-speed external clock signal) and LSECLK (low-speed external clock signal).

There are two kinds of external clock sources:

- External crystal/ceramic resonator
- External clock of user

The hardware configuration of the two kinds of clock sources is shown in the figure below.

Figure 7 HSECLK/LSECLK Clock Source Hardware Configuration

Clock source	Hardware configuration
External clock	
Crystal/ceramic resonator	

In order to reduce the distortion of clock output and shorten the startup stabilization time, the crystal/ceramic resonator and load capacitor must be as close to the oscillator pin as possible. The value of the load capacitance (C_{L1}, C_{L2}) must be adjusted according to the selected oscillator.

5.3.1.1 **HSECLK (high-speed external clock) signal**

HSECLK signal is generated by two kinds of clock sources, namely HSECLK external crystal/ceramic resonator and HSECLK.

Table 18 Clock Sources Generating HSECLK

Name	Description
External clock source (HSECLK bypass)	The clock is provided to MCU by OSC_IN pin. The signal can be generated by ordinary function signal transmitter (in debugging), crystal oscillator and other signal generators; the waveform can be square wave, sine wave or triangle wave with 50% duty cycle, and the maximum frequency is up to 26MHz. For hardware connection, it must be connected to HSE_IN pin, ensuring HSE_OUT pin is suspended (in high-impedance state); for MCU configuration, the user can select this mode by setting HSEBYPASS and HSEON bits in RCM_RCCR register (clock control register).
External crystal/ceramic resonator (HSECLK crystal)	The clock is provided to MCU by the resonator, and the resonator includes crystal resonator and ceramic resonator. The frequency range is ? MHz. HSE_IN and HSE_OUT need to be connected to a resonator, which can be enabled or disabled by setting the HSEON bit in the RCM_RCCR register of clock control registers. HSERDY bit in the clock control register RCM_RCCR is used to indicate whether the high-speed external oscillator is stable. After it is enabled, the clock is not released until this bit is set to "1" by hardware. If the HSERDYIE bit in the RCM_CIER (clock interrupt enable register) is set to 1 to allow interrupt generation, a corresponding interrupt will be generated. In this case, the HSERDYF bit in the corresponding RCM_CICR (clock interrupt flag register) will be set to 1 by hardware.

The HSERDY bit of the RCM_RCCR register indicates whether HSECLK is stable. The HSECLK clock will be input to the system only when the hardware sets the HSERDY bit to 1. If the HSERDYIE bit of the RCM_CIER register is enabled, the corresponding interrupt will be generated.

HSECLK can be enabled and disabled through the HSEON bit in the RCM_RCCR register. When HSEON is configured to 0, the HSERDY flag bit will be cleared to 0 after 6 hse_clk cycles.

5.3.1.2 **LSECLK low-speed external clock signal**

LSECLK clock signal is generated by LSECLK external crystal/ceramic resonator and LSECLK external clock two kinds of clock sources.

Table 19 Clock Sources Generating LSECLK

Name	Description
External clock source (LSECLK bypass)	The clock is provided to MCU by OSC32_IN pin. The signal can be generated by ordinary function signal transmitter (in debugging), crystal oscillator and other signal generators; the waveform can be square wave, sine wave or triangle wave with 50% duty cycle, and the signal frequency needs to be 32.768kHz. For hardware connection, it must be connected to LSE_IN pin, ensuring LSE_OUT pin is suspended; for MCU configuration, the user can select this mode by setting LSEBYPASS and LSEON bits in RCM_RCCR register (clock control register).

Name	Description
External crystal/ceramic resonator (LSECLK crystal)	The clock is provided to MCU by the resonator, and the resonator includes crystal resonator and ceramic resonator. The frequency is 32.768kHz. LSE_IN and LSE_OUT need to be connected to a resonator, which can be enabled or disabled by setting the LSEON bit in the RCM_RCCR register (clock control register). LSERDY bit in the clock control register RCM_RCCR is used to indicate whether the high-speed external oscillator is stable. After it is enabled, the clock is not released until this bit is set to "1" by hardware. If the LSERDYIE bit in the RCM_CIER (clock interrupt enable register) is set to 1 to allow interrupt generation, a corresponding interrupt will be generated. In this case, the LSERDYF bit in the corresponding RCM_CICR (clock interrupt flag register) will be set to 1 by hardware.

The LSERDY bit of the RCM_RCCR register indicates whether LSECLK is stable. The LSECLK clock will be input to the system only when the hardware sets the LSERDY bit to 1. If the LSERDYIE bit of the RCM_CIER register is enabled, the corresponding interrupt will be generated.

LSECLK can be enabled and disabled through the LSEON bit in the RCM_RCCR register. When LSEON is configured to 0, the LSERDY flag bit will be cleared to 0 after 2 lse_clk cycles.

5.3.2 Internal Clock Source

The internal clock includes HSICLK (high-speed internal clock signal) and LSICLK (low-speed internal clock signal).

5.3.2.1 **HSICLK (High-speed internal clock) signal**

HSICLK signal is generated by internal 8 MHz RC oscillator.

The RC oscillator frequency of different chips is different, and the frequency of the same chip may also be different with the change of temperature and voltage; the HSICLK clock frequency of each chip has been calibrated to 1% (25℃, VDD =VDDA =3.3V) by the manufacturer before leaving the factory. When the system is reset, the value calibrated by the manufacturer will be loaded to RCM_TRM2 register; in addition, users can further adjust the frequency by setting HSI_TTRIM bit and HSI_FREQTRIM bit in RCM_TRM2 register according to the application environment (temperature and voltage) of the site.

The HSIRDY bit of the RCM_RCCR register indicates whether HSI is stable. The HSICLK clock will be input to the system only when the hardware sets the HSIRDY bit to 1.

If the HSIRDYIE bit of the RCM_CIER register is enabled, the corresponding interrupt will be generated. HSICLK can be enabled and disabled through the HSION bit in the RCM_RCCR register. When HSION is configured to 0, the

HSIRDY flag bit will be cleared to 0 after 6 hsi_clk cycles.

Compared with HSECLK crystal oscillator, RC oscillator can provide system clock without any external device; the start time of RC oscillator is shorter than that of HSECLK crystal oscillator; even after calibration, its clock frequency accuracy is still inferior to that of HSECLK crystal oscillator.

5.3.2.2 **LSICLK low-speed internal clock signal**

LSICLK is located in the BKP domain and is a low-power clock source that can operate in both stop and standby modes, with a clock frequency of 32KHz.

The LSIRDY bit of the BKPC_CR1 register indicates whether LSICLK is stable. The LSI clock will be input to the system only when the hardware sets the LSIRDY bit to 1. If the LSIRDYIE bit of the RCM_CIER register is enabled, the corresponding interrupt will be generated.

LSICLK can be enabled and disabled through the LSION bit in the BKPC_CR1 register. When LSION is configured to 0, the LSIRDY flag bit will be cleared to 0 after 2 lsi_clk cycles.

5.3.3 PLL (phase locked loop)

The PLL clock source can be selected from HSICLK or HSECLK, with a frequency of up to 128MHz.

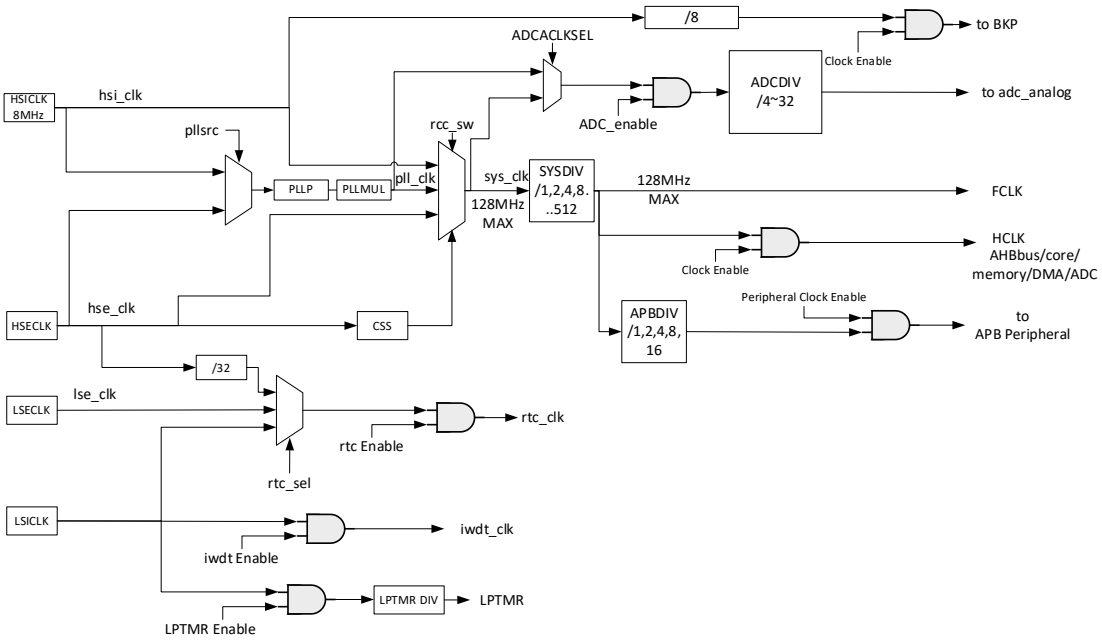
The software must first configure PLL parameters before enabling it, and adjusting PLL parameters halfway is disabled.

When entering stop and standby modes, the PLL will be automatically turned off by hardware, and the PLLON bit of the RCM_RCCR register will be cleared to zero. When exiting low-power mode, the user needs to perform PLL related initialization again.

If the clock source of PLL is HSECLK, and HSECLK experiences a clock fault, PLL will be automatically turned off by hardware, and the PLLON bit value of RCM_RCCR register will remain unchanged.

5.3.4 Clock tree

Figure 8 G32R430 Clock Tree



Note:

(1) HCLK means AHB clock signal.

(2) PLCK refers to the clock signal connected to APB.

(3) FCLK is the free-running clock of the core.

(4) The frequency of AHB and APB domains can be configured by multiple prescalers. The maximum frequency of the AHB domain is 128MHz, and that of the APB domain is 128MHz.

(5) The maximum frequency obtained by the system clock is 128MHz.

(6) Only when the corresponding enable bits are set, can the peripheral obtain the clock signal.

(7) ADC clock can be divided into analog circuit clock and digital circuit clock. The clock management unit provides a prescaler for the analog circuit clock of ADC, so that the ADC can work at the clock frequency of PLLCLK after frequency division, and at this time the clock can be used by all ADC; the clock frequency of digital circuit of ADC is equal to HCLK, and ADC clocks can be enabled respectively through the corresponding bits of RCM_AHBCLKEN register.

(8) SysTick (system timer) can be provided by the clock signal after frequency division of HCLK8. Different clock sources can be selected by setting SysTick control and status register.

(9) For the structure of PLL, please see the chapter of PLL and PLL Configuration Register.

5.3.5 System clock switching

After system reset, HSICLK is used as main system clock by default. When HSICLK directly or indirectly provides clocks to the system through PLL, the application will not be able to reset the HSION bit of RCM_RCCR register through software. The same for HSYON and PLLON.

Switching steps: Under the condition that all system clock sources are ready, clear the CSEN bit of RCM_MCCR, configure the SW bit of RCM_SCCR to select the system clock source, and then enable the CSEN bit of the RCM_MCCR register. If the SWDONE flag bit of the RCM_MCCR register is set to 1 by hardware, it indicates a successful switch. If the SWERR flag bit of the RCM_MCCR register is set to 1 by hardware, it indicates a failed switch.

The main system clock can only be switched successfully when both the source clock and the target clock are ready. If the clock is switched when the clock source is not ready, the main clock switching will be automatically delayed until the clock source is ready. If during the switching, it is found that the software enable for the target clock source is not turned on, the switching fails, and the SWERR bit in the RCM_MCCR register is set to 1 by hardware.

5.3.6 Clock fault detection

The system includes two clock fault detection: HSICLK fault detection for HSECLK, and LSICLK fault detection for LSECLK. When the CSSON, HSEON, and HSION bits in RCM_RCCR are all set to 1, the HSECLK clock fault detection function is activated. When the LSECSS and LSEON bits in RCM_RCCR, and the LSION bit in BKPC_CR1 are all set to 1, the LSECLK clock fault detection function is activated.

When a clock fault is detected, HSECLK/LSECLK will be turned off, and the LSECSSF/HSECSSF flag bit in the RCM_CICR register will be set to 1 by hardware. If the corresponding interrupt is enabled, an interrupt will be generated.

If HSECLK directly or indirectly provides clocks to the system through PLL and detects a clock fault in HSECLK, the system clock will be forced to switch to HSICLK. If the PLL uses HSECLK as the clock source, the PLL will also be turned off.

If HSECLK or LSECLK provides a clock to the RTC, when a clock failure of HSECLK or LSECLK is detected, the RTC clock will not be forced to switch to LSICLK.

5.3.7 RTC clock

The RTCSEL bit of BKP_CR1 can be configured to select the RTC clock. Once the RTC clock source is selected, it is not recommended to modify it directly. You can reset the backup domain by setting the BDRST bit of PMU_CR to 0,

and then configure the RTC clock source. In standby mode, HSECLK and LSECLK cannot provide clocks to RTC.

5.3.8 IWDT clock

If IWDT is activated by software or hardware, the LSICLK will be forcibly enabled and cannot be turned off.

5.3.9 ADCA clock

The ADCA clock is provided to the analog part of the ADC. Users need to configure the ADCADIV divider value according to the sys_clk frequency to make its frequency close to 30MHz. First configure the ADCADIV bit in RCM_SCCR, then configure the ADCACLKEN bit in RCM_SCCR. The ADC can only be used after detecting that the ADCACLKRDY bit is set to 1 by hardware.

5.3.10 Other peripheral clocks

All AHB peripherals and APB peripherals have corresponding clock enable software control bits. For details, see the Description of RCM_AHBCG and RCM_APBCG Registers.

5.3.11 CLKOUT

The internal clock can be output to GPIO by configuring the CLKOUTDIV, CLKOUTEN, and CLKOUTSEL bits of the RCM_SCCR register. Note that the CLKOUTDIV and CLKOUTSEL bits shall be configured first, and CLKOUTEN shall be turned on at last. GPIO output has frequency limitations and cannot output too high frequencies.

5.4 Register Address Mapping

Table 20 RCM Register Address Mapping

Register name	Description	Offset address
RCM_KEY	Unlock register	0x00
RCM_RCCR	Clock control register	0x04
RCM_PLLCR	PLL control register	0x08
RCM_SCCR	System clock control register	0x0C
RCM_MCCR	Master clock control register	0x10
RCM_CIER	Clock interrupt enable register	0x14
RCM_CICR	Clock status register	0x18
RCM_AHBRST	AHB reset register	0x20
RCM_APBRST	APB reset register	0x24
RCM_AHBCG	AHB clock enable register	0x28

Register name	Description	Offset address
RCM_APBCG	APB clock enable register	0x30
RCM_PWRCR	Power control register	0x38
RCM_RSTCSR	Reset status register	0x3C
RCM_ADCCR	ADC control register	0x44
RCM_SYS_ANACR0	Analog control register 0	0x60
RCM_SYS_ANACR1	Analog control register 1	0x64

5.5 Register Functional Description

5.5.1 Unlock Register (RCM_KEY)

Offset address: 0x00

Reset value: 0x0000 0000

Field	Name	Description
31:17	Reserved	
16	KEYST	System register KEY protects status bit After unlocking the system register, the KEYST flag bit will be set, and writing 1 to this bit will lock it. Note: Writing 0 is invalid, and writing 1 will lock it.
15:0	LOCKKEY	Password protection configuration for system register write operations 1. Before operating on the SYSCTRL system register, it must be unlocked by writing 0X87E4 to the KEY value. 2 After unlocking the system register, the KEYST0 flag bit will be set, and writing 1 to this bit will lock it.

5.5.2 Clock control register (RCM_RCCR)

Offset address: 0x04

Reset value: 0x0000 0003

Field	Name	Description
31:28	Reserved	
27	LSEBYPASS	LSEBYPASS control bit enable 0: Disable 1: Enable
26	LSECSSON	LSE clock fault detection enable 0: Disable 1: Enable
25	LSERDY	LSE-READY flag bit
24	LSEON	LSE Enable 0: Disable 1: Enable
23:18	Reserved	
17	PLLRDY	PLL clock READY flag When entering STOP or STANDBY mode, this bit is cleared to zero.

Field	Name	Description
16	PLLON	PLL Enable 0: Disable 1: Enable
15:12	Reserved	
11	CSSON	HSE clock fault detection enable 0: Disable 1: Enable
10	HSEBYPASS	HSEBYPASS
9	HSERDY	HSE Clock READY flag When HSYON is turned off, HSERDY is cleared to zero after 6 HSE-CLK cycles.
8	HSEON	HSE enable 0: Disable 1: Enable
7:2	Reserved	
1	HSIRDY	HSI clock READY flag 0: Disable 1: Enable When HSION is turned off, HSIRDY is cleared to zero after 6 HSI_CLK cycles.
0	HSION	HSI enable 0: Disable 1: Enable

5.5.3 PLL control register (RCM_PLLCR)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	Description
31:22	Reserved	
21:16	PLLMUL	PLL frequency doubling coefficient 000000: Incorrect configuration 000001: Incorrect configuration 000010: 2x frequency multiplier ..... 001001: 9x frequency multiplier 001111: 15x frequency multiplier 010000: 16x frequency multiplier ..... 111111: 63x frequency multiplier Note: Dynamic configuration is not supported. This bit can only be configured when the PLL is disabled.
15:12	Reserved	
11:8	PLLP	PLL input clock frequency division control 0000: No frequency division 0001: No frequency division 0010: 2-divided frequency

Field	Name	Description
		0011: 3-divided frequency 0100: 4-divided frequency ... 1111: 15-divided frequency Note: The minimum clock frequency that the PLL can receive is 6 MHz.
7:2	Reserved	
1:0	PLLSRC	PLL input clock select 0x: No clock 10：HSICLK 11：HSECLK When PLL is disabled, PLLSRC should be configured as 00 or 01.

5.5.4 System clock control register (RCM_SCCR)

Offset address: 0x0C

Reset value: 0x0000 0000

Field	Name	Description
31:23	Reserved	
22:20	CLKOUTDIV	CLKOUT clock frequency division output 000: CLKOUT 1-divide frequency 001: CLKOUT 2-divide frequency 010: CLKOUT 4-divide frequency 011: CLKOUT 8-divide frequency 100: CLKOUT 16-divide frequency
19	CLKOUTEN	Clock output enable 0: Disable 1: Enable
18:16	CLKOUTSEL	CLKOUT clock output select 000: SYSCLK 001: HSICLK 010: HSECLK 011: PLL 100: LSICLK 101: LSECLK Other: Reserve (clock selected by PLLINTERCLKOUT2SEL)
15:11	Reserved	
10:8	PDIV	APB clock frequency division 0xx: No frequency division for HCLK 100: HCLK 2-divided frequency 101: HCLK 4-divided frequency 110: HCLK 8-divided frequency 111: HCLK 16-divided frequency
7:4	HDIV	AHB clock frequency division 0xxx: SYSCLK 1000: SYSCLK 2-divided frequency 1010: SYSCLK 4-divided frequency 1010: SYSCLK 8-divided frequency

Field	Name	Description
		1011: SYSCLK 16-divided frequency 1100: SYSCLK 32-divided frequency 1101: SYSCLK 64-divided frequency 1110: SYSCLK 128-divided frequency 1111: SYSCLK 256-divided frequency
3:2	Reserved	
1:0	SW	System clock select 00: HSICLK is used as main clock 01: HSECLK is used as main clock 1x: PLL is main clock Note: First configure SW to enable CSEN and initiate switching

5.5.5 Master clock control register (RCM_MCCR)

Offset address: 0x10

Reset value: 0x0000 0000

Field	Name	Description
31:5	Reserved	
4	SWERR	Switch error flag bit When clearing CSEN to zero, this flag will also be cleared.
3	SWDONE	Switch complete flag bit When clearing CSEN to zero, this flag will also be cleared.
2:1	SWST	Current master clock status 00: HSICLK is used as main clock 01: HSECLK is used as main clock 1x: PLL is main clock
0	CSEN	Master clock switching enable 0: Disable 1: Enable Note: First configure the SW, then turn on CSEN to enable clock switching. Reading the flag bit SWDONE=1 indicates successful switching; otherwise SWERR=1 indicates switching failure. Finally, the software needs to clear CSEN to zero; otherwise the next switching cannot be initiated;

5.5.6 Clock interrupt enable register (RCM_CIER)

Offset address: 0x14

Reset value: 0x0000 0000

Field	Name	Description
31:7	Reserved	
6	LSECSSIE	LSECLK clock fault enable 0: Disable 1: Enable
5	HSECSSIE	HSECLK clock fault enable 0: Disable 1: Enable
4	PLLRDYIE	PLL-READY interrupt enable 0: Disable

Field	Name	Description
		1: Enable
3	HSERDYIE	HSE-READY interrupt enable 0: Disable 1: Enable
2	HSIRDYIE	HSI-READY interrupt enable 0: Disable 1: Enable
1	LSERDYIE	LSE-READY interrupt enable 0: Disable 1: Enable
0	LSIRDYIE	LSI-READY interrupt enable 0: Disable 1: Enable

5.5.7 Clock status register (RCM_CICR)

Offset address: 0x18

Reset value: 0x0000 0000

Field	Name	Description
31:7	Reserved	
6	LSECSSF	LSE- Clock Fault Flag Bit
5	HSECSSF	HSE- Clock Fault Flag Bit
4	PLLRDYF	PLL-READY flag bit
3	HSERDYF	HSE-READY flag bit
2	HSIRDYF	HSI-READY flag bit
1	LSERDYF	LSE-READY flag bit
0	LSIRDYF	LSI-READY flag bit

5.5.8 AHB reset register (RCM_AHBRST)

Offset address: 0x20

Reset value: 0x0000 01E3

Field	Name	Description
31:9	Reserved	
8	FLASHRST	FLASH reset The FLASH can only be reset when there is no FlASH access. 0: Reset 1: Not reset
7	ADC3RST	12 bit -ADC3 module reset 0: Reset 1: Not reset
6	ADC2RST	16 bit -ADC2 module reset 0: Reset 1: Not reset
5	ADC1RST	16 bit -ADC1 module reset 0: Reset 1: Not reset

Field	Name	Description
4:2	Reserved	
1	GPIORST	GPIO module reset 0: Reset 1: Not reset
0	DMARST	DMA module reset 0: Reset 1: Not reset

5.5.9 APB reset register (RCM_APBRST)

Offset address: 0x24

Reset value: 0x0000 3EFF

Field	Name	Description
31:14	Reserved	
13	COMPRST	COMP module reset 0: Reset 1: Not reset
12	DAC2RST	DAC2 module reset 0: Reset 1: Not reset
11	DAC1RST	DAC1 module reset 0: Reset 1: Not reset
10	EINTRST	EINT module reset 0: Reset 1: Not reset
9	TSRST	TS module reset 0: Reset 1: Not reset
8	Reserved	
7	I2CRST	I2C module reset 0: Reset 1: Not reset
6	USART2RST	USART2 module reset 0: Reset 1: Not reset
5	USART1RST	USART1 module reset 0: Reset 1: Not reset
4	SPIRST	SPI module reset 0: Reset 1: Not reset
3	TMR4RST	TMR4 module reset 0: Reset 1: Not reset
2	TMR3RST	TMR3 module reset

Field	Name	Description
		0: Reset 1: Not reset
1	TMR2RST	TMR2 module reset 0: Reset 1: Not reset
0	TMR1RST	TMR1 module reset 0: Reset 1: Not reset

5.5.10 AHB clock enable register (RCM_AHBCG)

Offset address: 0x28

Reset value: 0x0000 0300

Field	Name	Description
31:10	Reserved	
9	ROMCEN	ROM clock enable 0: Disable 1: Enable The ROM clock can only be turned off when there is no ROM access.
8	FLASHCEN	FLASH clock enable 0: Disable 1: Enable The FLASH clock can only be turned off when there is no FLASH access.
7	ADC3EN	ADC3 clock enable 0: Disable 1: Enable
6	ADC2EN	ADC2 clock enable 0: Disable 1: Enable
5	ADC1EN	ADC1 clock enable 0: Disable 1: Enable
4:2	Reserved	
1	GPIOCEN	GPIOA/B/C/D clock enable 0: Disable 1: Enable
0	DMACEN	DMA clock enable 0: Disable 1: Enable

5.5.11 APB clock enable register (RCM_APBCG)

Offset address: 0x30

Reset value: 0x0000 0000

Field	Name	Description
31:15	Reserved	
14	BKPCEN	BKP domain interface clock enable

Field	Name	Description
		0: Disable 1: Enable
13	COMPCEN	COMP module interface clock enable 0: Disable 1: Enable
12	DAC2CEN	DAC2 module interface clock enable 0: Disable 1: Enable
11	DAC1CEN	DAC1 module interface clock enable 0: Disable 1: Enable
10	EINTCEN	EINT module interface clock enable 0: Disable 1: Enable
9	TSCEN	TS module interface clock enable 0: Disable 1: Enable
8	WWDTCEN	WWDT module interface clock enable 0: Disable 1: Enable
7	I2CCEN	I2C module interface clock enable 0: Disable 1: Enable
6	USART2CEN	USART2 module interface clock enable 0: Disable 1: Enable
5	USART1CEN	USART1 module interface clock enable 0: Disable 1: Enable
4	SPICEN	SPI module interface clock enable 0: Disable 1: Enable
3	TMR4CEN	TMR4 module interface clock enable 0: Disable 1: Enable
2	TMR3CEN	TMR3 module interface clock enable 0: Disable 1: Enable
1	TMR2CEN	TMR2 module interface clock enable 0: Disable 1: Enable
0	TMR1CEN	TMR1 module interface clock enable 0: Disable 1: Enable

5.5.12 Power control register (RCM_PWRCR)

Offset address: 0x38

Reset value: 0x0000 0000

Field	Name	Description
31:9	Reserved	
8	BKPWP	BACKUP area write protection enabled 0: Disable 1: Enable (accessible)
7:2	Reserved	
1	RAM1RET	TCM loses power in STOP mode 0: In 0: STOP mode, ITCM is not powered down 1: In 0: STOP mode, ITCM is powered down
0	LPM	Low-power mode select When the CPU enters DEEPSLEEP mode, this bit selects the low-power mode that the system is in. 0: STOP mode 1: STANDBY mode

5.5.13 Reset status register (RCM_RSTCSR)

Offset address: 0x3C

Reset value: 0x0000 0000

Field	Name	Description
31:6	Reserved	
5	WWDTRSTF	Reset caused by window watchdog 0: Reset caused by non-window watchdog 1: Reset caused by window watchdog
4	IWDTRSTF	Reset caused by independent watchdog 0: Reset caused by non-independent watchdog 1: Reset caused by independent watchdog
3	SFTRSTF	Reset caused by software 0: Reset not caused by software 1: Reset caused by software
2	PORRSTF	Reset caused by power 0: Reset not caused by power supply 1: Reset caused by power supply
1	PINRSTF	Reset caused by external RST pin 0: Reset not caused by external RST pin 1: Reset caused by external RST pin1
0	OPRSTF	Reset caused by OPLOAD 0: Reset not caused by OPLOAD 1: Reset not caused by OPLOA

5.5.14 ADC control register (RCM_ADCCR)

Offset address: 0x44

Reset value: 0x0000 0000

Field	Name	Description
31:9	Reserved	
8	ADCACLKSEL	Clock division source clock selection for ADC1/2/3_ANALOG clock 0: PLL clock 1: SYSCLK clock
7	ADC12ACLKRDY	ADC3_ANALOG clock status bit 0: Not ready 1: Ready
6:5	ADC12ADIV	ADC3_ANALOG clock frequency division ADC3_ANALOG clock (duty cycle 50%); the value of ADC12ADIV can only be changed when ADC12ACLKRDY is 0 and module ADC3 is disabled 00: 4-divided frequency. 01: 8-divided frequency. 10: 16-divided frequency. 11: 32-divided frequency.
4:3	Reserved	
2	ADC16ACLKRDY	ADC1/2_ANALOG clock status bit ADC1/2_ANALOG clock status bit. When ADC1/2 is enabled, ADC can perform the conversion only when this bit becomes 1. 0: Not ready 1: Ready
1:0	ADC16ADIV[1:0]	ADC1/2_ANALOG clock frequency division ADC1/2_ANALOG clock (duty cycle 75%); the value of ADC16ADIV can only be changed when ADC16ACLKRDY is 0 and both modules ADC1 and ADC2 are disabled 0x: 6-divided frequency. 10: 12-divided frequency. 11: 24-divided frequency.

5.5.15 Analog control register 0 (RCM_SYS_ANACR0)

Offset address: 0x60

Reset value: 0x0002 0041

Field	Name	Description
31:24	Reserved	
23:20	LSE_LEAKTRIM	XIN port leakage compensation adjustment control 0000: Maximum leakage compensation of PMOS 1000: default leakage compensation (default) 1111: Maximum leakage compensation of NMOS
19	LSEAUTOTRIMREQ	LSECLK leakage compensation automatic calibration enable It is set to 1 by software, and cleared to 0 automatically by hardware upon completion of calibration (there is no need to enable LSELKDTEN by software). 0: Disable 1: Enable Note: During hardware calibration, the system main frequency should not be lower than 32KHz; if lselkdten is 0, this bit will be cleared to 0 and cannot be written to 1

Field	Name	Description
18	LSELKDTSTA	Leakage detection function status judgment 0: Indicates XOUT>XIN 1: Indicates XOUT<XIN
17	LSEDRV	LSECLK drive capacity select 0: Low drive 1: High drive (default)
16	LSELKDTEN	LSECLK leakage detection function enable 0: Disable detection function 1: Enable detection function The detection function can work only when both LSEON and LSELKDTEN are set to 1. This function is necessary only when testing and trimming the leakage current compensation. In all other states, LSELKDTEN should be configured to 0.
15:8	Reserved	
7:5	CHOPFREQSEL	Chopping clock frequency division 000：50KHz 001：100KHz 010：200KHz 011：400KHz 100：800KHz 101：1.3333MHz 110：2MHz
4	LNBGAUTODONE	LNBG initialization completion flag bit Write 1 to clear to 0. 0: Not completed 1: Completed
3	ADC16BUFEN	LNBG BUF enable 0: Disable 1: Enable
2	CHOPPEREN	LNBG Chopping enable 0: Disable 1: Enable
1	LNBGEN	LNBG enable 0: Disable 1: Enable
0	LNBGAUTO	LNBG hardware automatic initialization enable 0: Disable 1: Enable When this bit is enabled, the software does not need to separately enable LNBGEN, CHOPPEREN, and ADC16BUFEN. When ADC1 or ADC2 is enabled, the LNBG and ADC16BUF are automatically initialized by hardware. When the software determines that LNBGAUTODONE is 1, it indicates that initialization is completed and the ADC can start conversion.

5.5.16 Analog control register 1 (RCM_SYS_ANACR1)

Offset address: 0x64

Reset value: 0x0000 2180

Field	Name	Description
31:22	Reserved	
21:20	DAC2BIAS_TRIM	Bias current of the DAC2 operational amplifier trim_bias[1:0] is enable switch signal, which can adjust the bias current of the DAC2 operational amplifier; trim_bias[1:0]=00 by default
19:18	DAC1BIAS_TRIM	Bias current of the DAC1 operational amplifier trim_bias[1:0] is enable switch signal, which can adjust the bias current of the DAC1 operational amplifier; trim_bias[1:0]=00 by default
17:16	CORELDO_PORVTRIM	Configuration signal of the POR detection threshold of core LDO 00: Threshold 0.9V 01: Threshold 0.8V 10: Threshold 0.75V 11: Threshold 0.7V
15	Reserved	
14:13	ADCBUF_ITRIM	TRIM output stage current of the ADCBUFFER trim_bias[1:0]=01 by default
12:8	PLLICP_TRIM	PLL charge pump ICP current adjustment Only the two lower bits are valid. trim_icp_lv[1:0]=00 ICP current does not increase trim_icp_lv[1:0]=01 ICP current increases by 1 μA (default) trim_icp_lv[1:0]=10 ICP current increases by 2 μA trim_icp_lv[1:0]=11 ICP current increases by 4 μA
7:6	PLLLPFRES0_TRIM	PLL LPF resistor adjustment signal 00: Resistance value 50KΩ 01: Resistance value 40KΩ 10: Resistance value 30KΩ (default) 11 Resistance value 20KΩ
5	PLLLPFCAP1	PLL LPF large capacitor adjustment signal 0: The capacitance value is about 30pF (default) 1: The capacitance value is about 60pF
4	PLLLPFCAP0	PLL LPF small capacitor adjustment signal 0: The capacitance value is about 1pF (default) 1: The capacitance value is about 2pF
3:2	ADC2COMP_TRIM	The common-mode voltage of the LATCH circuit of the ADC2 Trim comparator trim_comp_lv[1:0]=00 by default
1:0	ADC1COMP_TRIM	The common-mode voltage of the LATCH circuit of the ADC1 Trim comparator trim_comp_lv[1:0]=00 by default

6 Power Management Unit (PMU)

6.1 Full Name and Abbreviation Description of Terms

Table 21 Full Name and Abbreviation Description of Terms

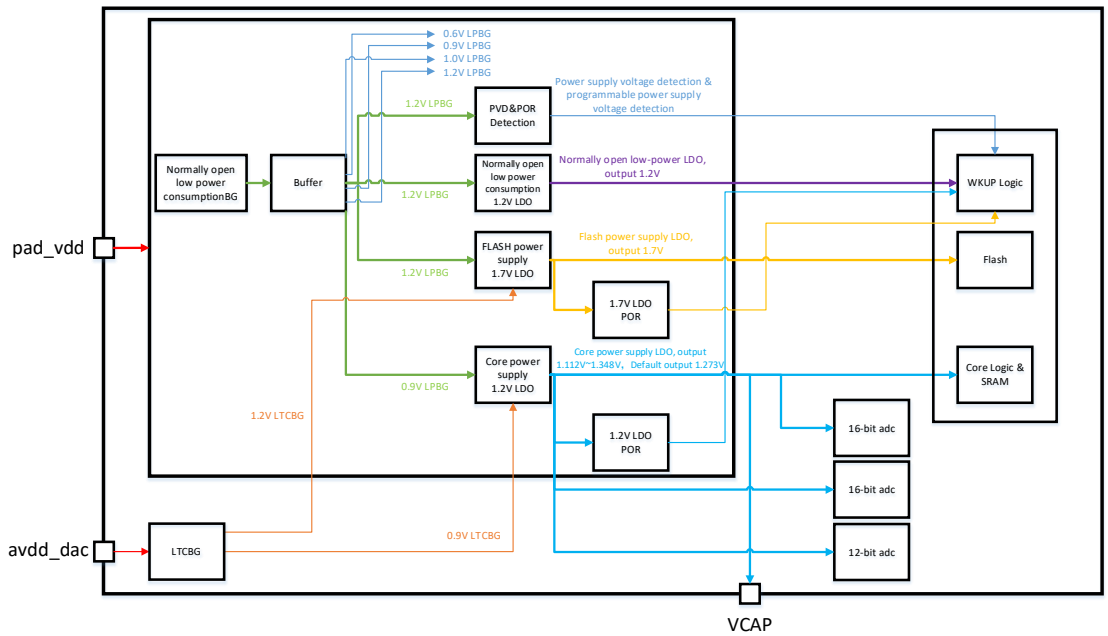
Full name in English	English abbreviation
Power Management Unit	PMU
Power On Reset	POR
Brown-out Reset	BOR
Power Voltage Detector	PVD

6.2 Introduction

The power supply is the foundation for stable operation of a system, with an operating voltage of 1.7 ~ 3.6V, and 1.2V power supply can be provided by the built-in LDO regulator.

6.3 Structure block diagram

Figure 9 Power Supply Structure Block Diagram



6.4 Functional description

6.4.1 Power domain

The power domain of the product includes: V_{DD} power domain, V_{DDA} power domain, 1.2V power domain, and backup power domain.

6.4.1.1 VDD power domain

Power the I/O (see the pin distribution diagram for specific IO) and internal voltage regulator through VDD pin.

LDO Voltage regulator

It powers the 1.5 V power domain in the following operating modes:

- Normal mode: In this mode, 1.2 V power supply area runs at full power
- Stop mode: In this mode, 1.2 V power supply area works in low-power state, all clocks are off, peripherals stop working and the set voltage output level remains unchanged.
- Standby mode: In this mode, 1.2 V power supply area is powered down, the voltage regulator enters a high-impedance output state, the core circuit powers down, and the power consumption of the voltage regulator is zero. Except for the backup circuit, the content of the registers and TCM will be lost.

6.4.1.2 V_{DDA} power domain

Supply power to the analog parts of ADC, DAC, RC oscillator, and PLL. When a 16-bit ADC is used, VDDA shall not be lower than 3.0 V. When HSECLK, TSEN, 12-bit ADC, COMP, or DAC is used, VDDA shall not be lower than 2.7V.

Independent ADC power supply

Independent ADC power supply can improve conversion accuracy, and the specific power pins are as follows:

- V_{DDA} : Power pin of ADC
- V_{SSA}: Independent power ground pin
- V_{REFH}/V_{REFL}: Reference voltage pin of 16-bit ADC

Reference voltage

The 16-bit ADC can be configured with internal reference voltage and VREFH pin input voltage as reference voltage. The reference voltage of 12-bit ADC and DAC can only be VDDA.

6.4.1.3 1.2V power domain

Powered by the LDO regulator, it supplies power to the core, Flash, TCM, and

digital peripherals.

6.4.1.4 **Backup power domain**

In standby mode, the Backup domain can work normally, and is powered by a low-power LDO. The Backup domain includes functional modules such as RTC, LPTMR, LSI, standby-IO wake-up, and BKPC. BKPC includes 32-byte data backup registers. In standby mode, 4 IOs support wake-up. The wake-up method is configured in the BKP_WKPCR register. In standby mode, the wake-up IOs are forced to input mode.

After reset, the registers for RTC, BKPC, and LPTMR are protected to prevent incorrect write. To access the backup domain, conduct configuration according to the steps below.

(1) Configure the BKPCEN bit in the RCM_APBCG to 1 to enable the interface clock for the BKP domain

(2) Configure the BKPWP bit in the RCM_PWRCR to 1 to disable write protection

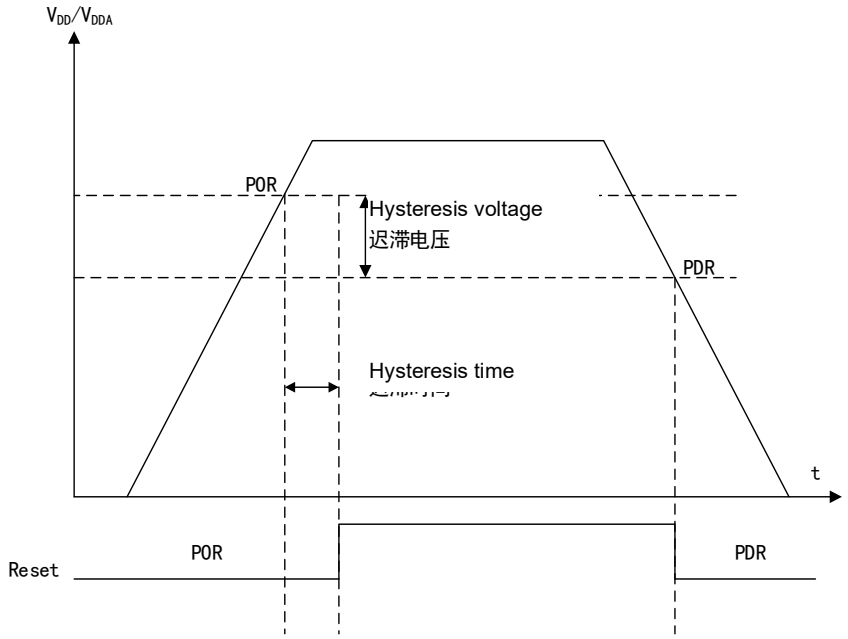
If access to LPTMR or RTC is required, further configure the LPTMRCEN or RTCEN bit in the PMU_CR register to 1.

6.4.2 Power management

6.4.2.1 **Power-on reset (POR)**

When the VDD /VDDA is lower than the threshold voltage VPOR and VPDR , the chip will automatically remain in the reset state. The waveform diagrams of power-on reset and power-down reset are as follows. For POR, PDR, hysteresis voltage and hysteresis time, please refer to the Datasheet.

Figure 10 Power-on Reset and Power-down Reset Oscillogram

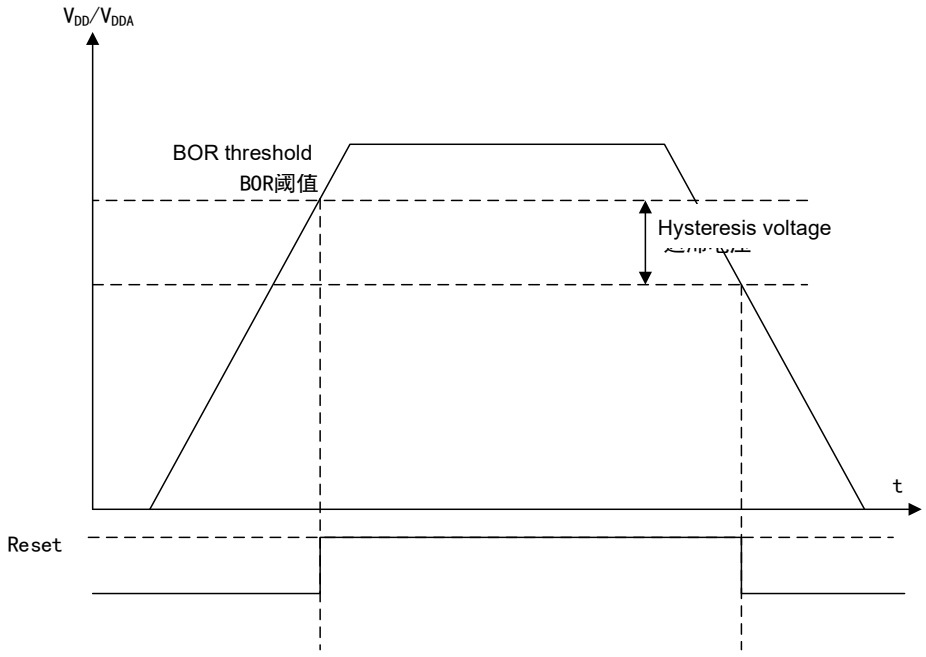


6.4.2.2 **Brown-out reset (BOR)**

When it is detected that V_{DD}/V_{DDA} is lower than the threshold voltage V_{BOR}, the chip will automatically remain in reset state, and V_{BOR} can be configured via option bytes. The followings are 4 thresholds of V_{BOR}:

- V_{BOR0}: BOR is disabled, and the voltage range is 1.70~2.10V
- V_{BOR1}: BOR level is 1, and the voltage range is 2.10~2.40V
- V_{BOR2}: BOR level is 2, and the voltage range is 2.40~2.70V
- V_{BOR3}: BOR level is 3, and the voltage range is 2.70~3.60V

Figure 11 BOR Threshold Oscillogram



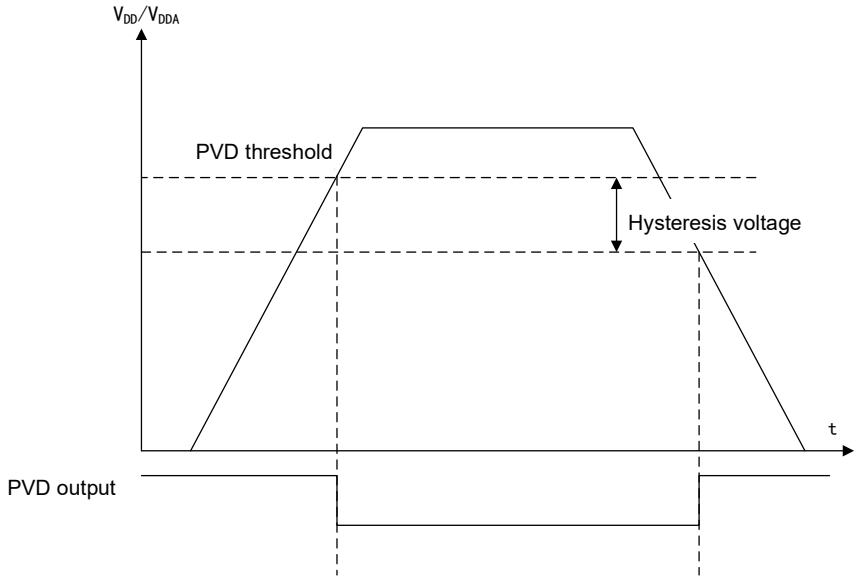
6.4.2.3 **Power voltage detector (PVD)**

The product has a built-in programmable voltage detector (PVD) that can monitor VDD and compare it with V_{PVD} threshold. When VDD is outside the V_{PVD} threshold range and the interrupt is enabled, an interrupt will be generated and the MCU can be set to a safe state through the interrupt service program.

Users can use the PVD to monitor the VDD supply by comparing the PVD with the threshold selected by PVDTHSEL in the PMU_PVDCR register. Enable the PVD by setting the PVDEN bit in the PMU_PVDCR register to 1.

There is a PVDO flag bit in the PMU_SR register, which indicates whether VDD is above or below the PVD threshold. When this event is internally connected to EINT, it can generate an interrupt if enabled via the EINT register. When VDD is below the PVD threshold and/or when VDD is above the PVD threshold, a PVD output interrupt can be generated based on the EINT rising/falling edge configuration.

Figure 12 PVD Threshold Oscillogram



Main power detection module

The chip integrates a dedicated EVS pin internally, which can implement the power-on wake-up function for the main power. The function of this module can be enabled via the VDCPVDEN bit in the PMU_EVSCR register. Setting the VDCPVDSEL[1:0] bits allows detection of different input voltages.

The module integrates filtering function internally to filter out the interference from the input power supply. The filter clock uses LSICLK. By setting the value of VDCPVDFILTER[1:0], different filter times can be obtained.

When the chip is in STANDBY mode, whether to enable the function of the main power detection module can be selected based on the value of the

VDCPVDWKPEN bit.

The main power detection module is connected to external interrupt EINT18 in the chip. An interrupt can be triggered when a rising or falling edge event is detected on the EVS pin.

Note: The rising and falling edge thresholds in VPVDx correspond to the rising and falling edge signals in the EINT module (Stop or Run modes can be configured). When the EVS pin is used for standby wake-up, the wake-up threshold is the threshold corresponding to the rising edge, not the rising or falling edge, which means that as long as the EVS pin level remains above the threshold, it will continuously trigger a wake-up.

6.4.3 Low-power mode

The G32R430 supports stop and standby two low-power modes. These two modes have differences in power consumption, wake-up process, wake-up duration, and wake-up method. The low-power mode can be selected according to actual application needs.

A 32-byte backup register is built in G32R430. In low-power mode, the stored information will not be lost, so it can be used to store important data.

Table 22 Low-power Mode

Mode	Description
Low-power run mode	In the low-power run mode, the main operating frequency is limited in order to reduce the operating power. The code can be executed in DTCM or Flash. When 1.7V≤VDD≤2.4V, the maximum CPU frequency is limited to 8MHz. When 2.4V≤VDD≤3.0V, the maximum CPU frequency is limited to 30MHz. Peripherals with independent clocks can process clocks via HSICLK.
Stop mode	Under the condition that the data of register and some DTCM is not lost, the lowest power can be reached in stop mode; In stop mode, the clock of the internal 1.2V voltage regulator (within the VCORE domain) stops, the HSICLK stops, the HSE crystal oscillator and PLL are powered down. The voltage regulator LDO can be configured as normal or low-power mode. LSECLK and LSICLK can be configured to continue to run or stop. The RTC, backup registers, and low-power timer can remain active. Some peripherals with wake-up capability (see the table below) can wake the chip from Stop mode and enable HSICLK. Any external interrupt line can wake up MCU, and the external interrupt lines include one of the 16 external interrupt lines, PVD output, RTC and low-power timer.
Standby mode	Standby mode is the chip's lowest-power mode; In Standby mode, the internal 1.2 V voltage regulator is turned off, all 1.2V power modules (within the VCORE domain) are powered down, the HSE crystal oscillator, HSICLK, and PLL are powered down. Data in DTCM and remaining registers except backup registers is lost. LSECLK and LSICLK can be configured to continue to run or stop. The RTC, backup registers, and low-power timer can remain active, and the standby circuitry is still working;

Mode	Description
	The external reset signal on NRST, wake-up event on the WKUP pin, BOR reset, RTC or events related to the low-power timer can wake up the MCU.

Note: Stop indicates the module's register information and operating state are retained; Power Down indicates the module's register information and working state are lost.

Table 23 Module Status in Low-Power Mode (1)

		Status in Stop Mode (2)	Wake-up Capability in Stop Mode	Status in Standby Mode (2)	Wake-up Capability in Standby Mode
Core		-	-	-	-
Clock	HSECLK	-	-	-	-
	HSICLK	-	-	-	-
	LSECLK	Y	-	Y	-
	LSICLK	Y	-	Y	-
	PLL	-	-	-	-
Base IP	LDO	-	-	-	-
	Lowpower LDO	Y	-	Y	-
	FLASH	-	-	-	-
	TCM	O (3)	-	-	-
	Backup register	Y	-	Y	-
	DMA	-	-	-	-
	GPIO	O	Y	O (4)	Y (4)
Timer	TMR1/2/3/4	-	-	-	-
	LPTMR	O	Y	O	Y
	IWDT	O	Y	-	-
	WWDT	-	-	-	-
	RTC	O	Y	O	Y
Communication interface	USART1/2	-	-	-	-
	I2C	-	-	-	-
	SPI	-	-	-	-
Analog interface	16-bit ADC1/2	-	-	-	-
	12-bit ADC	-	-	-	-
	COMP1/2/13/14	-	-	-	-
	DAC1/2	-	-	-	-
	Temp Sensor	-	-	-	-

Note:

(1) Y indicates normal operation, O indicates configurable operation (off by default, can be enabled by software), - indicates not supporting operation.

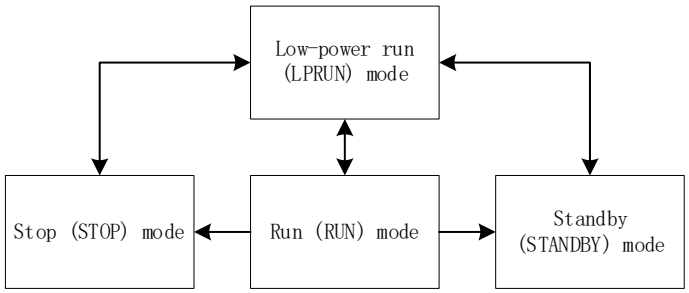
(2) Unless otherwise specified, in STOP mode, the operating status and register information of each IP are reserved. In STANDBY mode, the operating status and register information of each IP are lost.

(3) DTCM (16KB) operates normally and is not powered down.

(4) The four wake-up pins can be configured as internal pull-up, pull-down, or analog mode. The four wake-up pins can be used for pin wake-up in STANDBY mode. The wake-up pins are: PC1\PC2\PD0\PD9

Generally, the CPU is in Run mode after a system or power reset. There are two low-power modes in which the power consumption can be reduced when the CPU does not need to keep running (e.g., while waiting for an external event). Users can choose the mode that offers the best compromise among low power consumption, short start-up time, and available wake-up sources. Direct status transition between the two low-power modes is not possible. The system can only wake up from a low-power mode to the run mode, or enter a low-power mode from the run mode. The specific transition flow is shown in the figure below.

Figure 13 Low-power Mode Transition



Low-power run mode

The characteristics of low-power run mode are shown in the table below:

Table 24 Characteristics of Low-power Run Mode

Characteristics	Description
Enter	Option: Run the program from DTCM and power down the Flash; Reduce the CPU clock frequency to below 8MHz/30MHz according to the VDD voltage level.
Wake-up	Increase the CPU clock frequency.
During running	The CPU runs at a low main frequency.
Delay when switching to run mode	The time required for the CPU clock frequency to increase and stabilize.

Stop mode

The characteristics of stop mode are shown in the table below:

Table 25 Characteristics of STOP Mode

Characteristics	Description
Enter	Set the SLEEPDEEP bit of the core register to 1; Ensure no pending interrupt (WFI) or event (WFE); set the LPM bit in the RCM_PWRCR register to 0; enter the Stop mode immediately when executing the WFI or WFE instruction; When the LDOSTOPCFG bit in the RCM_PWRCR register is set to 0, the voltage regulator operates in normal mode. When it is set to 1, the regulator operates in the low-power mode.
Wake-up	The 8MHz HSICLK oscillator is used as the clock source for the wake-up process. If WFI instruction is executed to enter the Stop mode, wake up by any interrupt; if WFE instruction is executed to enter the Stop mode, wake up by an event.
Stop	The core and the peripheral will stop working, and the data in the core register and memory before stop will be saved.
Wakeup delay	Wake-up time of HSICLK oscillator + wake-up time of voltage regulator from low-power mode.
After wake-up	If the system is awakened by an interrupt, it will first enter the interrupt state, then exit the interrupt, and then execute the program after WFI instruction. If the system is awakened by an event, it will directly execute the program after WFE instruction. When exiting the Sop mode, the system clock is HSICLK.

Standby mode

The characteristics of standby mode are shown in the table below:

Table 26 Standby Mode

Characteristics	Description
Enter	Set the SLEEPDEEP bit of the core register to 1; Confirm there are no pending interrupts (WFI) or events (WFE); Set the LPM bit in the RCM_PWRCR register to 1; Enter the standby mode immediately when executing WFI or WFE instructions.
Wake-up	The 8MHz HSICLK oscillator is used as the clock source for the wake-up process. Wake-up via the rising edge/falling edge/both edges on the WKUP pin, BOR reset, low-power timer wake-up, RTC alarm wake-up, or external reset wake-up on the NRST pin.
Standby	The core and the peripheral will stop working, and the data in the core register and memory will be lost.
Wakeup delay	Chip reset time.
After wake-up	The program starts executing from the beginning. When exiting the standby mode, the system clock is HSICLK.

6.4.3.1 **Reduce the power in run mode**

In the run mode, the operating power consumption can be reduced by reducing the system clock, enabling or disabling the peripheral clock on the APB/AHB bus. RTC alternate function is awakened from low-power mode.

The RTC alternate function can wake up the MCU from Stop or Standby mode.

The RTC provides a programmable time base, facilitating periodic wake-up of the MCU from Stop or Standby mode.

6.5 Register address mapping

Table 27 PMU Register Address Mapping

Register name	Description	Offset address
PMU_CR	Control register	0x00
PMU_WKPCR	Low-power wake-up control register	0x04
PMU_SR	Status register	0x08
PMU_PUCRA	Wake-up IO pull-up control register	0x10
PMU_PDCRA	Wake-up IO pull-down control register	0x14
PMU_EVSCR	Power-down wake-up control register	0x1C
PMU_PVDCR	PVD control register	0x20

6.6 Register functional description

6.6.1 Control register (PMU_CR)

Offset address: 0x00

Reset value: 0x0000 2000

Field	Name	R/W	Description
31:16	Reserved		
15	BDRST	R/W	BKUP area reset 0: Not reset 1: Reset
14	Reserved		
13	NRSTOUTDIS	R/W	Enable the internal reset pulse output of the chip to the NRST pin 0: Disable output 1: Enable output
12	Reserved		
11	LPTMRCEN	R/W	LPTMR clock enable
10	RTCEN	R/W	RTC clock enable
9:8	RTCSEL	R/W	RTC clock select 00: No clock 01：LSECLK 10：LSICLK 11：HSECLK/32
7	Reserved		
6	LSIRDY	R/W	LSICLK ready flag bit 0: Not ready 1: Ready
5	LSION	R/W	LSICLK enable

Field	Name	R/W	Description
			0: Disable 1: Enable
4:0	Reserved		

6.6.2 Low-power wake-up control register (PMU_WKPCR)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15	APC	R/W	When entering the standby mode, select the state of GPIO 0: Analog mode 1: PWR_PUCR/PDCR determines pull-up/pull-down status
14:12	Reserved		
11:10	WKUPPOL3	R/W	PD9-WUIO wake-up polarity 00: Rising edge 01: Falling edge 1x: Rising edge and falling edge
9:8	WKUPPOL2	R/W	PD0-WUIO wake-up polarity 00: Rising edge 01: Falling edge 1x: Rising edge and falling edge
7:6	WKUPPOL1	R/W	PC2-WUIO wake-up polarity 00: Rising edge 01: Falling edge 1x: Rising edge and falling edge
5:4	WKUPPOL0	R/W	PC1-WUIO wake-up polarity 00: Rising edge 01: Falling edge 1x: Rising edge and falling edge
3	WKUPEN3	R/W	Wake up PD9-WUIO enable 0: Disable 1: Enable
2	WKUPEN2	R/W	Wake up PD0-WUIO enable 0: Disable 1: Enable
1	WKUPEN1	R/W	Wake up PC2-WUIO enable 0: Disable 1: Enable
0	WKUPEN0	R/W	Wake up PC1-WUIO enable 0: Disable 1: Enable

6.6.3 Status register (PMU_SR)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:7	Reserved		
6	PVDO	R	PVD output 0: VDD voltage is lower than the set value of PVDTHSEL 1: VDD voltage is higher than the set value of PVDTHSEL
5	EVSWKF	RC_W1	EVS wake-up flag The EVS function can only be used in standby mode. 0: No wake-up occurs 1: Wake up occurs
4	SBF	RC_W1	DEVICE has entered standby mode 0: DEVICE has not entered the standby mode 1: DEVICE has entered the standby mode
3	WKUPF3	RC_W1	PD9-WUIO wake flag
2	WKUPF2	RC_W1	PD0-WUIO wake flag
1	WKUPF1	RC_W1	PC2-WUIO wake flag
0	WKUPF0	RC_W1	PC1-WUIO wake flag

6.6.4 Wake-up IO pull-up control register (PMU_PUCRA)

Offset address: 0x10

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:4	Reserved		
3	WKPU3	R/W	PD9-WUIO pull-up In standby mode, when APC is enabled, this bit is set to 1, and the corresponding IO is in pull-up status.
2	WKPU2	R/W	PD0-WUIO pull-up In standby mode, when APC is enabled, this bit is set to 1, and the corresponding IO is in pull-up status.
1	WKPU1	R/W	PC2-WUIO pull-up In standby mode, when APC is enabled, this bit is set to 1, and the corresponding IO is in pull-up status.
0	WKPU0	R/W	PC1-WUIO pull-up In standby mode, when APC is enabled, this bit is set to 1, and the corresponding IO is in pull-up status.

6.6.5 Wake-up IO pull-down control register (PMU_PDCRA)

Offset address: 0x14

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:4	Reserved		
3	WKPD3	R/W	PD9-WUIO pull-down In standby mode, when APC is enabled, this bit is set to 1, and the corresponding IO is in pull-down status
2	WKPD2	R/W	PD0-WUIO pull-down In standby mode, when APC is enabled, this bit is set to 1, and the corresponding IO is in pull-down status
1	WKPD1	R/W	PC2-WUIO pull-down

Field	Name	R/W	Description
			In standby mode, when APC is enabled, this bit is set to 1, and the corresponding IO is in pull-down status
0	WKPD0	R/W	PC1-WUIO pull-down In standby mode, when APC is enabled, this bit is set to 1, and the corresponding IO is in pull-down status

6.6.6 Power-down wake-up control register (PMU_EVSCR)

Offset address: 0x1C

Reset value: 0x0000 0010

Field	Name	R/W	Description
31:6	Reserved		
5	VDCPVDWKPEN	R/W	EVS pin STANDBY mode wake-up enable 0: Disable wake-up from Standby mode via the main power domain 1: Enable wake-up from Standby mode via the main power domain
4	VDCPVDEN	R/W	EVS module enable 0: Disable 1: Enable
3:2	VDCPVDFILTER	R/W	EVS pin filter configure 00: Not filter 01: 2 LSICLK clocks 10: 4 LSICLK clocks 11: 8 LSICLK clocks
1:0	VDCPVDSEL	R/W	VDC-VPD threshold trimming 00：4.3V 01：4.26V 10：4.35V 11：4.4V Note: The triggering values for specific rising and falling edges can be found in the Datasheet.

6.6.7 PVD control register (PMU_PVDCR)

Offset address: 0x20

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:8	Reserved		
7:5	PVDTHSEL	R/W	Detection threshold switching of the programmable voltage detector For specific data, see the "Reset and Power Control Module Characteristics" in the Datasheet.
4	PVDEN	R/W	PVD enable control bit 0: Disable 1: Enable
3:0	Reserved		

7 Backup Domain Unit (BKP)

The backup domain contains 16 16-bit registers, which can be used to store 32 bytes of data. The backup domain register is powered by a low-power LDO. System resetting, NRST pin resetting, and resetting after the low mode is waken up will not affect the backup domain register. The backup domain registers can be reset by setting the BDRST bit of the PMU_CR register to 1 through software.

Before any operation on the backup domain, it is necessary to write the BKPWP bit in the RCM_PWRCR to 1 and enable the BKPCEN bit in the RCM_APBCG to turn on the backup domain clock.

Configurations related to standby mode, and clock enabling and configuration for LSICLK/RTC/LPTMR, are all set within the backup domain unit. See the "Register Function Description" for details.

The backup domain register can be used to cache user data, and be used as a state flag to realize some function application by using its characteristics that the data will remain unchanged after system reset.

7.1 Register Address mapping

Table 28 BKP Register Address Mapping

Register name	Description	Offset address
BKP_DRx	Data register x (x=0~15)	0x00+0x04*x

7.2 Register Functional Description

7.2.1 Data register (BKP_DRx)

Offset address: 0x00+0x04*x (x= 0~15)

Reset value: 0x0000 0000

Field	Name	Description
31:16	Reserved	
15:0	DATA	Backup register data

8 Nested Vector Interrupt Controller (NVIC)

8.1 Full Name and Abbreviation of Terms

Table 29 Full Name and Abbreviation Description of Terms

Full name in English	English abbreviation
Non Maskable Interrupt	NMI
Nested Vectored Interrupt Controller	NVIC

8.2 Introduction

The Arm® Cortex®-M52 core in the product integrates nested vectored interrupt controller (NVIC), which is closely coupled with the core, and can handle exceptions and interrupts and power management control efficiently with low delay.

8.3 Main characteristics

(1) 48 maskable interrupt channels (excluding 16 Arm® Cortex®-M4F interrupt lines)

(2) 16 programmable priority levels (use 4-bit interrupt priority level), with flexible grouping configuration for preemption priority and subpriority.

(3) Low-delay exception and interrupt processing

(4) Power management control

(5) Implementation of system control register

8.4 Interrupt and exception vector table

Table 30 G32R430 Interrupt and Exception Vector Table

Exception type	Vector No.	Priority(1)	Vector address	Description
-	-	-	0x0000 0000	Reserved
Reset	-	-3	0x0000 0004	Reset
NMI	-	-2	0x0000 0008	Non-maskable interrupt
HardFault	-	-1	0x0000 000C	Hardware fault interrupt
MemManage	-	Can set	0x0000 0010	Memory management
BusFault	-	Can set	0x0000 0014	Pre-fetch fault, memory access fault

Exception type	Vector No.	Priority(1)	Vector address	Description
UsageFault	-	Can set	0x0000 0018	Undefined instruction or illegal state
-	-	Can set	0x0000 001C-0x0000 002B	Reserved
SVCall	-	Can set	0x0000 002C	SWI instruction implements system service calls
Debug Monitor	-	Can set	0x0000 0030	Debug monitor
-	-	Can set	0x0000 0034	Reserved
PendSV	-	Can set	0x0000 0038	Pending system service requests
SysTick	-	Can set	0x0000 003C	System tick timer
WWDT	0	Can set	0x0000 0040	Window watchdog interrupt
FLASH	1	Can set	0x0000 0044	Flash global interrupt
RCU	2	Can set	0x0000 0048	Clock reset management interrupt
RTC	3	Can set	0x0000 004C	RTC global interrupt
DMA_CH0	4	Can set	0x0000 0050	DMA1 channel 0 global interrupt
DMA_CH1	5	Can set	0x0000 0054	DMA1 channel 1 global interrupt
DMA_CH2	6	Can set	0x0000 0058	DMA1 channel 2 global interrupt
DMA_CH3	7	Can set	0x0000 005C	DMA1 channel 3 global interrupt
DMA_CH4	8	Can set	0x0000 0060	DMA1 channel 4 global interrupt
DMA_CH5	9	Can set	0x0000 0064	DMA1 channel 5 global interrupt
DMA_CH6	10	Can set	0x0000 0068	DMA1 channel 6 global interrupt
DMA_CH7	11	Can set	0x0000 006C	DMA1 channel 7 global interrupt
ADC1	12	Can set	0x0000 0070	ADC1 global interrupt
ADC2	13	Can set	0x0000 0074	ADC2 global interrupt
ADC3	14	Can set	0x0000 0078	ADC3 global interrupt
TMR1	15	Can set	0x0000 007C	Timer 1 interrupt
TMR2	16	Can set	0x0000 0080	Timer 2 interrupt
TMR3	17	Can set	0x0000 0084	Timer 3 interrupt
TMR4	18	Can set	0x0000 0088	Timer 4 interrupt
SPI	19	Can set	0x0000 008C	SPI global interrupt
USART1	20	Can set	0x0000 0090	USART1 global interrupt
USART2	21	Can set	0x0000 0094	USART2 global interrupt

Exception type	Vector No.	Priority(1)	Vector address	Description
I2C	22	Can set	0x0000 0098	I2C global interrupt
LPTMR	23	Can set	0x0000 009C	LPTMR global interrupt
EINT0	24	Can set	0x0000 00A0	EINT_Line0 interrupt
EINT1	25	Can set	0x0000 00A4	EINT_Line 1 interrupt
EINT2	26	Can set	0x0000 00A8	EINT_Line 2 interrupt
EINT3	27	Can set	0x0000 00AC	EINT_Line 3 interrupt
EINT4	28	Can set	0x0000 00B0	EINT_Line 4 interrupt
EINT5_9	29	Can set	0x0000 00B4	EINT_Line [9:5] interrupt
EINT10_15	30	Can set	0x0000 00B8	EINT_Line [15:10] interrupt
EINT16/PVD	31	Can set	0x0000 00BC	Interrupt and power voltage detection generated through EINT16
EINT17/RTC_ALARM	32	Can set	0x0000 00C0	RTC_ALARM generated through EINT_LINE17
EINT18/EVS	33	Can set	0x0000 00C4	Main power wakeup interrupt
EINT19/COMP1	34	Can set	0x0000 00C8	COMP1 interrupt generated through EINT_Line 19
EINT20/COMP2	35	Can set	0x0000 00CC	COMP2 interrupt generated through EINT_Line 20
EINT21/COMP3	36	Can set	0x0000 00D0	COMP3 interrupt generated through EINT_Line 21
EINT22/COMP4	37	Can set	0x0000 00D4	COMP4 interrupt generated through EINT_Line 22
TS	38	Can set	0x0000 00D8	TS temperature sensor interrupt

Note: (1) There are a total of 16 configurable interrupt priorities, with smaller numbers indicating higher interrupt priorities.

9 External interrupt/event controller (EINT)

9.1 Introduction

The interrupts/events contain internal interrupts/events and external interrupts/events. In this manual, the external interrupt refers to the interrupt/event caused by I/O pin input signal, which is EINTx in interrupt vector table; other interrupts mean internal interrupts/events.

The events can be divided into hardware events and software events. Hardware events are generated by external/core hardware signals, while software events are generated by instructions.

Interrupts need to go through the interrupt handler function to implement the work to be processed, while events do not need to go through interrupt handler function, and the preset work can be triggered by hardware. The external events output pulse through events such as GPIO, while the internal events trigger another TMR to work, for example, through update event of a TMR.

9.2 Main characteristics

(1) Support 23 event/interrupt requests

(2) Each event/interrupt line can be masked independently

(3) Each external event/interrupt line can be triggered independently

(4) Each external interrupt line has dedicated status bit

(5) Detect external signals with pulse width lower than APB clock width

9.3 Functional Description

9.3.1 Classification and difference of "external interrupt and event"

"External interrupt and event" can be classified into external hardware interrupt, external hardware event, external software event and external software interrupt according to trigger source, configuration and execution process. The differences are shown in the table below:

Table 31 Classification and Differences of "External Interrupts and Events"

Name	Trigger source	Configuration and execution process
External hardware interrupt	External signal	(1) Set the trigger mode, allow the interrupt request, and enable corresponding peripheral interrupt line (enable in NVIC); (2) When an edge consistent with the configuration is generated on the external interrupt line, an interrupt request will be generated, and the corresponding pending bit will be set to 1; write 1 to the corresponding bit of the pending register and the interrupt request will be cleared.
External hardware event	External signal	(1) Set the trigger mode and enable the event line; (2) When an edge consistent with the configuration is generated on the external event line, an event request pulse will be generated, and the corresponding pending bit will not be set to 1.
External software event	Software interrupt register/transmit event (SEV) instruction	(1) Enable the event line; (2) Write 1 to the software interrupt event register of the corresponding event line to generate an event request pulse, and the corresponding pending bit will not be set to 1.
External software interrupt	Software interrupt register	(1) Allow interrupt request, and enable the corresponding peripheral interrupt line (enable in NVIC); (2) Write 1 to the software interrupt event register of the corresponding interrupt line to generate an interrupt request, the corresponding pending bit will be set to 1; write 1 to the corresponding bit of the pending register and the interrupt request will be cleared.

9.3.2 Core wake-up

Using WFI and WFE instructions can make stop the core. When WFI instruction is used, any interrupt can wake up the core; when WFE instruction is used, the core can be awakened by an event.

When interrupt is used for wake-up, the interrupt handler function will be triggered, and normal interrupt configuration can wake up the core. When an event is used to wake up the core, the interrupt handler function will not be triggered, which will reduce the wake-up time, and the configuration method is:

(1) Trigger an internal interrupt (internal hardware event) but do not trigger the interrupt handler function for wake-up

- Enable an internal interrupt in the peripheral, but do not enable the corresponding interrupt in NVIC to avoid triggering the interrupt handler function

- Enable SEVONPEND bit in the system controller of the core, and execute WFE instruction to make the core enter sleep mode
- Generate an interrupt to wake up the core; when the core recovers from WFE, it is required to clear the pending bit of corresponding peripheral interrupt and the pending bit of peripheral NVIC interrupt channel (clear the pending register in the NVIC interrupt)

(2) Wake up by EINT line events (external hardware event)

- Configure EINT line as the event mode
- Execute WFE instruction to make the core enter the sleep mode
- Generate an interrupt to wake up the core; after the CPU recovers from WFE, since the pending bit of corresponding event line is not set, it is unnecessary to clear the interrupt pending bit of corresponding peripheral or the pending bit NVIC interrupt channel

9.3.2.1 **Event wake-up**

Trigger an internal interrupt (internal hardware event) but do not trigger the interrupt handler function for wake-up

(1) Enable an internal interrupt in the peripheral, but do not enable the corresponding interrupt in NVIC to avoid triggering the interrupt handler function;
(2) Enable SEVONPEND bit in the system controller of the core, and execute WFE instruction to make the core enter sleep mode;
(3) Generate an interrupt to wake up the core; when the core recovers from WFE, it is required to clear the interrupt pending bit of corresponding peripheral and the pending bit of peripheral NVIC interrupt channel (clear the pending register in the NVIC interrupt).

Wake up by EINT line events (external hardware event)

(1) Configure EINT line as the event mode;
(2) Execute WFE instruction to make the core enter the sleep mode;
(3) Generate an interrupt to wake up the core; when the CPU recovers from WFE, since the pending bit of corresponding event line is not set, it is unnecessary to clear the interrupt pending bit of corresponding peripheral or the pending bit of the NVIC interrupt channel.

9.3.3 External interrupt and event line mapping

Table 32 External Interrupt and Event Line Mapping

External Interrupt and Event Channel Name	External Interrupt and Event Line No.
PA0/PB0/PC0/PD0	EINT 0
PA1/PB1/PC1/PD1	EINT 1
…	…

External Interrupt and Event Channel Name	External Interrupt and Event Line No.
PA9/PB9/PC9/PD9	EINT 9
PB10/PC10/PD10	EINT 10
PB11/PC11/PD11	EINT 11
PB12/PC12/PD12	EINT 12
PB13/PD13	EINT 13
PD14	EINT 14
PD15	EINT 15
PVD output	EINT 16
ALARM event	EINT 17
EVS	EINT 18
COMP1	EINT 19
COMP2	EINT 20
COMP3	EINT 21
COMP4	EINT 22

9.4 Register address mapping

Table 33 EINT Register Address Mapping

Register name	Description	Offset Address
EINT_IMASK	Interrupt mask register	0x00
EINT_EMASK	Event mask register	0x04
EINT_RTEN	Enable the rising edge to trigger the register	0x08
EINT_FTEN	Enable the falling edge to trigger the register	0x0C
EINT_SWINTE	Software interrupt event register	0x10
EINT_IPEND	Interrupt pending register	0x14
EINT_IOSELR1	GPIO port selection register 1	0x18
EINT_IOSELR2	GPIO port selection register 2	0x1C

9.5 Register functional description

9.5.1 Interrupt mask register (EINT_IMASK)

Offset address: 0x00

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:23	Reserved		

Field	Name	R/W	Description
22:0	IMASKx	R/W	Interrupt Request Mask on Line x (x=0~22) 0: Mask 1: Open

9.5.2 Event mask register (EINT_EMASK)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:23	Reserved		
22:0	EMASKx	R/W	Event Request Mask on Line x (x=0~22) 0: Mask 1: Open

9.5.3 Enable the rising edge to trigger the register (EINT_RTEN)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:23	Reserved		
22:0	RTENx	R/W	Rising Trigger Event and Interrupt Enable of Line x (x=0~22) 0: Disable 1: Enable

Note: Since the external wake-up lines are edge-triggered, there should be no glitch signal on these lines; when writing EINT_RTEN register, if the rising edge signal is on the external interrupt line, it will not be recognized and the pending bit will not be set; on the same interrupt line, the rising edge trigger and falling edge trigger can be set at the same time.

9.5.4 Enable the falling edge to trigger the register (EINT_FTEN)

Offset address: 0x0C

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:23	Reserved		
22:0	FTENx	R/W	Falling Trigger Event and Interrupt Enable of Line x (x=0~22) 0: Disable 1: Enable

Note: Since the external wake-up lines are edge-triggered, there should be no glitch signal on these lines; when writing EINT_FTEN register, if the rising edge signal is on the external interrupt line, it will not be recognized and the pending bit will not be set; on the same interrupt line, the rising edge trigger and falling edge trigger can be set at the same time.

9.5.5 Software interrupt event register (EINT_SWINTE)

Offset address: 0x10

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:23	Reserved		
22:0	SWINTEx	R/W	Software Interrupt Event on Line x (x=0~22) Set 1 by software, write 1 or clear 0 for the corresponding bit of EINT_IPEND. When this bit is 0, the pending bit of EINT_IPEND can be set by writing 1. If EINT_IMASK (EINT_EMASK) is set to open the interrupt (event) request, an interrupt (event) will be generated. 0: No effect 1: Software generates an interrupt (event)

9.5.6 Interrupt pending register (EINT_IPEND)

Offset address: 0x14

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:23	Reserved		
22:0	IPENDx	RC_W1	Interrupt Pending Occur of Line x Flag (x=0~22) When an edge-trigger request corresponding to EINT_RTEN/EINT_FTEN occurs on the external interrupt line, the hardware sets the bit to 1; it can be cleared to 0 either by changing the edge-detection polarity, or by writing 1 to this bit to clear it.

9.5.7 GPIO port selection register 1 (EINT_IOSELR1)

Offset address: 0x18

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	EINTx[3:0]	R/W	EINTx Configuration (x=0~7) Write through software to choose the source input for EINTx external interrupts. 0000：PA[x] 0001：PB[x] 0010：PC[x] 0011：PD[x] Other：Reserved

9.5.8 GPIO port selection register 2 (EINT_IOSELR2)

Offset address: 0x1C

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	EINT(x+8)[3:0]	R/W	EINTx Configuration (x=0~7) Write through software to choose the source input for EINTx external interrupts. 0000：PA[x] 0001：PB[x] 0010：PC[x] 0011：PD[x] Other：Reserved

10 Direct Memory Access (DMA)

10.1 Introduction

DMA (Direct Memory Access) can realize high-speed data transmission between peripheral devices and memory or between memory and memory without CPU intervention, thus saving CPU resources for other operations.

The product has one DMA controller, with 8 data streams. Each data stream corresponds to 8 channels, but only 1 channel can be used for each data stream at the same time. Each data stream can set priority, and the arbiter can coordinate the priority of corresponding DMA requests of each data stream according to the priority of the data stream .

10.2 Main Characteristics

(1) The DMA has 8 data streams, and each data stream has 8 channels

(2) Dual AHB main interfaces; one is memory interface, and the other is peripheral interface

(3) There are three data transfer modes: peripheral to memory, memory to peripheral, and memory to memory (Flash cannot be written as the target memory)

(4) Each data stream has a special hardware DMA request for connection

(5) Support software priority and hardware priority when multiple requests occur at the same time
- Software priority: The four levels are highest, high, medium, and low
- Hardware priority: Supported by hardware arbiter. Request 1 takes priority over Request 2

(6) Each data stream has 5 event flags and independent interrupts

(7) The configurable source and target transmission width is byte, half word or word

(8) Support circular transmission mode

(9) Support source and target incremental modes

(10) Supports DMA flow control (hardware supports peripheral flow control)

(11) The configurable burst increment size is single time, 4, 8 or 16 ticks

(12) The number of data for transmission is programmable, up to 65535

10.3 Functional Description

10.3.1 DMA request

If the peripheral or memory needs to transmit data using DMA, it is required to first transmit DMA request and after it is approved by DMA, data transmission can be started.

DMA has 8 data streams in total. Each data stream is connected with different peripheral channels, and each data stream has five event flags (DMA half transmission, DMA transmission completion, DMA transmission error, DMA FIFO error, and direct mode error). The logic of the five event flags may become a separate interrupt request, and they all support software trigger.

When multiple peripherals request the same data stream, it is required to configure the corresponding register to turn on or off the request of each peripheral, so as to ensure that one data stream can only turn on one peripheral request.

Table 34 DMA Request Mapping Table

Peripheral request	Data stream 0	Data stream 1	Data stream 2	Data stream 3	Data stream 4	Data stream 5	Data stream 6	Data stream 7
Channel 0	ADC1	ADC2	ADC3	-	-	-	-	-
Channel 1	-	-	-	USART1_TX	USART1_RX	USART2_TX	USART2_RX	-
Channel 2	-	-	-	-			SPI_TX	SPI_RX
Channel 3	-	-	-	-	I2C_TX	I2C_RX	-	-
Channel 4	TMR1_UP	TMR1_CH1	-	-	-	TMR1_TRIG	-	-
Channel 5	-	TMR2_UP	TMR2_CH1	-	TMR2_CH3	-	TMR2_TRIG	-
Channel 6	-	-	TMR3_UP	-	TMR3_CH2	-	TMR3_CH4	TMR3_TRIG
Channel 7	TMR4_UP	TMR4_CH1	-	TMR4_CH3	-	TMR4_TRIG	-	-

10.3.2 Arbiter

When multiple DMA channel requests occur, an arbiter is needed to manage the response sequence. Management is divided into two stages: the first stage is software stage, which is divided into the highest, high, medium and low priority; the second stage is hardware stage, and under the condition of the same

software priority, the lower the data stream number is, the higher the priority is.

10.3.3 FIFO

FIFO is used to temporarily store data before the source data is transmitted to the destination address. Each data stream has an independent 4-word FIFO, and the FIFO threshold can be controlled by software to be 1/4, 1/2, 3/4 or full.

There are two DMA transmission modes. The first is direct mode, in which a single transmission will be started to the memory immediately after each peripheral request. If DMA is configured to transmit data from the memory to the peripheral, DMA will store a data in FIFO, and once the peripheral triggers the DMA request, it will transmit the data. The direct mode requires the same data width configuration for the source and destination addresses, and does not support burst mode or memory-to-memory transmission mode. The second is FIFO mode, in which, FIFO threshold is configured first, and when the data storage reaches the threshold, FIFO content will be transmitted to the destination address; FIFO mode is applicable when the data width of source address and destination address is different, and it supports burst mode; FIFO can store the data first and output them as required.

10.3.4 Port

The DMA controller performs data transmission with memory and peripherals through the memory port and peripheral port. Both the memory port and peripheral port of DMA are connected to the AHB matrix bus. DMA memory and peripherals can access internal Flash, internal TCM, AHB peripherals, and APB peripherals.

10.3.5 DMA initialization parameter configuration

10.3.5.1 **Transmission Mode**

DMA supports three transfer modes: peripheral to memory, memory to peripheral, and memory to memory (including Flash and TCM). The transmission mode can be controlled through DIRCFG bit of DMA_SCFG register.

10.3.5.2 **Increment mode**

The increment mode of peripheral and memory is controlled through PERIM and MEMIM bits of DMA_SCFG register. When both bits are set to 1, it is configured as the increment mode and the increment is the value of PERSIZECFG and MENSIZECFG bits of DMA_SCFG register. The PERSIZECFG and MENSIZECFG bits are used to set the data size of peripheral and memory to byte, half word or word.

10.3.5.3 **Single transmission and burst mode**

Burst transmission refers to the high-speed transmission that increases the data

volume transmitted each time at the transmission stage so as to improve the transmission speed. In the process of burst transmission, AHB bus will be occupied.

Single and burst transmissions can be controlled through the PBCFG and MBCFG bits of DMA_SCFG register, and it can be configured as single transmission, incremental burst transmission of 4 ticks, incremental burst transmission of 8 ticks and incremental burst transmission of 16 ticks. This increment is determined by the value of PERSIZECFG and MENSIZECFG bits. The burst mode can be enabled only when the increment mode is supported.

The burst mode shall be used in combination with FIFO, and the selected FIFO threshold shall be suitable for the burst size of memory, as shown in the table below.

Table 35 FIFO Threshold Configuration

<table>
<tr><th>MENSIZECFG</th><th>FIFO threshold</th><th>MBCFG=01</th><th>MBCFG=10</th><th>MBCFG=11</th></tr>
<tr><td rowspan="4">Byte</td><td>1/4</td><td>One-time burst of 4 ticks</td><td>Disable</td><td rowspan="3">Disable</td></tr>
<tr><td>1/2</td><td>Two-time burst of 4 ticks</td><td>One-time burst of 8 ticks</td></tr>
<tr><td>3/4</td><td>Three-time burst of 4 ticks</td><td>Disable</td></tr>
<tr><td>Full</td><td>Four-time burst of 4 ticks</td><td>Two-time burst of 8 ticks</td><td>One-time burst of 16 ticks</td></tr>
<tr><td rowspan="4">Half word</td><td>1/4</td><td>Disable</td><td rowspan="3">Disable</td><td rowspan="8">Disable</td></tr>
<tr><td>1/2</td><td>One-time burst of 4 ticks</td></tr>
<tr><td>3/4</td><td>Disable</td></tr>
<tr><td>Full</td><td>Two-time burst of 4 ticks</td><td>One-time burst of 8 ticks</td></tr>
<tr><td rowspan="4">Word</td><td>1/4</td><td rowspan="3">Disable</td><td rowspan="4">Disable</td></tr>
<tr><td>1/2</td></tr>
<tr><td>3/4</td></tr>
<tr><td>Full</td><td>One-time burst of 4 ticks</td></tr>
</table>

10.3.5.4 **Circular mode**

The circular mode is used to process the circular buffer area and continuous data stream. The circular mode will automatically configure the number of data items as the initial value after the transmission ends, and continue the data transmission.

The circular mode can be controlled through CIRCMEN bit of DMA_SCFG register.

10.3.5.5 **Double-buffer mode**

Set DBM of DMA_SCFG register to 1 to enable the double- buffer mode and automatically activate the circular mode. In the double-buffer mode, the DMA_M1ADDR register is activated, and when the corresponding memory area of the address pointer of DMA_M0ADDR register finishes transmission, it will switch to the corresponding memory area of the address pointer of DMA_M1ADDR register to continue to transmit and be called circularly. When DMA accesses the DMA_M1ADDR, CTARG bit of DMA_SCFG register will be set to 1 and data can be written or read to DMA_M0ADDR register.

This mode does not support memory-to-memory transmission.

10.3.5.6 **Stream controller**

The stream controller can be configured as DMA or peripheral through PERFC bit of DMA_SCFG register.

When DMA is used as the stream controller, configure DMA_NDATA register before enabling data stream, and set the number of data items to be transmitted.

When the peripheral is used as the stream controller, the number of transmitted data items is unknown, and the hardware will force the value of DMA_NDATA register to 0xFFFF for execution. After the transmission is completed, the peripheral will send instructions to DMA through hardware, and then read the value of the register. The number of transmitted data=0xFFFF-DMA_NDATA.

When the peripheral is used as the stream controller, the circular mode is disabled. When the memory-to-memory mode is selected, the PERFC bit will be forced to be cleared to zero by the hardware, and only DMA can be selected as the stream controller.

10.3.6 Interrupt

Each data stream has five types of interrupt events: half transmission, transmission completion, transmission error, FIFO error and direct mode error.

Table 36 DMA Interrupt Request

Interrupt event	Event flag bit	Enable interrupt bit
Half transmission	HTXIFLGx	HTXIEN
Transmission completed	TXCIFLGx	TXCIEN
Transmission error	TXEIFLGx	TXEIEN
FIFO error	FEIFLGx	FEIEN
Direct mode error	DMEIFLGx	DMEIEN

10.4 Register Address Mapping

Table 37 DMA Register Address Mapping

Register name	Description	Offset address
DMA_LINTSTS	DMA low interrupt status register	0x00
DMA_HINTSTS	DMA high interrupt status register	0x04
DMA_LIFCLR	DMA low interrupt flag clear register	0x08
DMA_HIFCLR	DMA high interrupt flag clear register	0x0C
DMA_SCFG	DMA data stream x configuration register	0x10+0x18× (data stream number)
DMA_NDATA	DMA data stream x data item number register	0x14+0x18× (data stream number)
DMA_PADDR	DMA data stream x peripheral address register	0x18+0x18× (data stream number)
DMA_M0ADDR	DMA data stream x memory 0 address register	0x1C+0x18× (data stream number)
DMA_M1ADDR	DMA data stream x memory 1 address register	0x20+0x18× (data stream number)
DMA_FCTRL	DMA data stream x FIFO control register	0x24+0x18× (data stream number)

10.5 Register Functional Description

10.5.1 DMA low interrupt status register (DMA_LINTSTS)

Offset address: 0x00

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:28, 15:12	Reserved		
27, 21, 11, 5	TXCIFLGx	R	Stream x Transfer Complete Interrupt Flag (x=0…3) These bits are set to 1 by hardware; write 1 and set to 0 by software on the corresponding bit of DMA_LIFCLR register. 0: No transmission completion event 1: Transmission completion event is generated.
26, 20, 10, 4	HTXIFLGx	R	Stream x Half Transfer Interrupt Flag (x=0…3) These bits are set to 1 by hardware; write 1 and set to 0 by software on the corresponding bit of DMA_LIFCLR register. 0: No half-transmission event 1: Half-transmission event occurs
25, 19, 9, 3	TXEIFLGx	R	Stream x Transfer Error Interrupt Flag (x=0…3) These bits are set to 1 by hardware; write 1 and set to 0 by software on the corresponding bit of DMA_LIFCLR register. 0: No transmission error 1: Transmission error occurs

Field	Name	R/W	Description
24, 18, 8, 2	DMEIFLGx	R	Stream x Direct Mode Error Interrupt Flag (x=0…3) These bits are set to 1 by hardware; write 1 and set to 0 by software on the corresponding bit of DMA_LIFCLR register. 0: No direct mode error 1: Direct mode error occurs
23, 17, 7, 1	Reserved		
22, 16, 6, 0	FEIFLGx	R	Stream x FIFO Error Interrupt Flag (x=0…3) These bits are set to 1 by hardware; write 1 and set to 0 by software on the corresponding bit of DMA_LIFCLR register. 0: No FIFO error event 1: FIFO error event occurs

10.5.2 DMA high interrupt status register (DMA_HINTSTS)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:28, 15:12	Reserved		
27, 21, 11, 5	TXCIFLGx	R	Stream x Transfer Complete Interrupt Flag (x=4…7) These bits are set to 1 by hardware; write 1 and clear to 0 by software on the corresponding bit of DMA_HIFCLR register. 0: No transmission completion event 1: Transmission completion event is generated.
26, 20, 10, 4	HTXIFLGx	R	Stream x Half Transfer Interrupt Flag (x=4…7) These bits are set to 1 by hardware; write 1 and clear to 0 by software on the corresponding bit of DMA_HIFCLR register. 0: No half-transmission event 1: Half-transmission event occurs
25, 19, 9, 3	TXEIFLGx	R	Stream x Transfer Error Interrupt Flag (x=4…7) These bits are set to 1 by hardware; write 1 and clear to 0 by software on the corresponding bit of DMA_HIFCLR register. 0: No transmission error 1: Transmission error occurs
24, 18, 8, 2	DMEIFLGx	R	Stream x Direct Mode Error Interrupt Flag (x=4…7) These bits are set to 1 by hardware; write 1 and clear to 0 by software on the corresponding bit of DMA_HIFCLR register. 0: No direct mode error 1: Direct mode error occurs
23, 17, 7, 1	Reserved		
22, 16, 6, 0	FEIFLGx	R	Stream x FIFO Error Interrupt Flag (x=4…7) These bits are set to 1 by hardware; write 1 and clear to 0 by software on the corresponding bit of DMA_HIFCLR register. 0: No FIFO error event 1: FIFO error event occurs

10.5.3 DMA low interrupt flag clear register (DMA_LIFCLR)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:28, 15:12	Reserved		
27, 21, 11, 5	CTXCIFLGx	W	Stream x Clear Transfer Complete Interrupt Flag (x=0…3) 0: Invalid 1: The corresponding TXCIFLGx flag in DMA_LINTSTS register is cleared to 0
26, 20, 10, 4	CHTXIFLGx	W	Stream x Clear Half Transfer Interrupt Flag (x=0…3) 0: Invalid 1: The corresponding HTXIFLGx flag in DMA_LINTSTS register is cleared to 0
25, 19, 9, 3	CTXEIFLGx	W	Stream x Clear Transfer Error Interrupt Flag (x=0…3) 0: Invalid 1: The corresponding TXEIFLGx flag in DMA_LINTSTS register is cleared to 0
24, 18, 8, 2	CDMEIFLGx	W	Stream x Clear Direct Mode Error Interrupt Flag (x=0…3) 0: Invalid 1: The corresponding DMEIFLGx flag in DMA_LINTSTS register is cleared to 0
23, 17, 7, 1	Reserved		
22, 16, 6, 0	CFEIFLGx	W	Stream x Clear FIFO Error Interrupt Flag (x=0…3) 0: Invalid 1: The corresponding FEIFLGx flag in DMA_LINTSTS register is set to 0

10.5.4 DMA high interrupt flag clear register (DMA_HIFCLR)

Offset address: 0x0C

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:28, 15:12	Reserved		
27, 21, 11, 5	CTXCIFLGx	W	Stream x Clear Transfer Complete Interrupt Flag (x=4…7) 0: Invalid 1: The corresponding TXCIFLGx flag in DMA_HINTSTS register is cleared to 0
26, 20, 10, 4	CHTXIFLGx	W	Stream x Clear Half Transfer Interrupt Flag (x=4…7) 0: Invalid 1: The corresponding HTXIFLGx flag in DMA_HINTSTS register is cleared to 0
25, 19, 9, 3	CTXEIFLGx	W	Stream x Clear Transfer Error Interrupt Flag (x=4…7) 0: Invalid 1: The corresponding TXEIFLGx flag in DMA_HINTSTS register is cleared to 0

Field	Name	R/W	Description
24, 18, 8, 2	CDMEIFLGx	W	Stream x Clear Direct Mode Error Interrupt Flag (x=4…7) 0: Invalid 1: The corresponding DMEIFLGx flag in DMA_HINTSTS register is cleared to 0
23, 17, 7, 1	Reserved		
22, 16, 6, 0	CFEIFLGx	W	Stream x Clear FIFO Error Interrupt Flag (x=4…7) 0: Invalid 1: The corresponding FEIFLGx flag in DMA_HINTSTS register is cleared to 0

10.5.5 DMA data stream x configuration register (DMA_SCFG) (x=0…7)

Offset address: 0x10+0x18× (data stream number)

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:28	Reserved		
27:25	CHSEL	R/W	Channel Selection 000: Select Channel 0 001: Select Channel 1 010: Select Channel 2 011: Select Channel 3 100: Select Channel 4 101: Select Channel 5 110: Select Channel 6 111: Select Channel 7
24:23	MBCFG	R/W	Memory Burst Transfer Configure 00: Single transmission 01: INCR4 (4-tick increment burst transmission) 10: INCR8 (8-tick increment burst transmission) 11: INCR16 (16-tick increment burst transmission) This bit can be written only when EN bit is 0. In direct mode, these bits will be forced to 0.
22:21	PBCFG	R/W	Peripheral Burst Transfer Configure 00: Single transmission 01: INCR4 (4-tick increment burst transmission) 10: INCR8 (8-tick increment burst transmission) 11: INCR16 (16-tick increment burst transmission) This bit can be written only when EN bit is 0. In direct mode, these bits will be forced to 0.
20	Reserved		
19	CTARG	R/W	Current Target (only in double buffer mode) This bit can be set to 1 or cleared to 0 by hardware, or be written by software. 0: The current target memory is Memory 0 1: The current target memory is Memory 1

Field	Name	R/W	Description
18	DBM	R/W	Double Buffer Mode 0: Do not switch the buffer when the transmission ends 1: Switch the target memory when DMA transmission ends This bit can be written only when EN bit is 0.
17:16	PRILCFG	R/W	Priority Level Configure 00: Low 01: Medium 10: High 11: Very high These bits can be written only when EN bit is 0.
15	PERIOSIZE	R/W	Peripheral increment offset size 0: The offset used to calculate the peripheral address is related to PERSIZECFG 1: The offset used to calculate the peripheral address is fixed to be 4 If PERIM bit is 0, this bit is meaningless, and it can be written only when EN bit is 0. If the direct mode is selected or the PBCFG bit is not configured to 00, and the data stream is enabled, this bit will be forced to low level by hardware.
14:13	MEMSIZECFG	R/W	Memory Data Size Configure 00: Byte (8 bits) 01: Half word (16 bits) 10: Word (32 bits) 11: Reserved These bits can be written only when EN bit is 0. In direct mode, when EN bit is 1, MEMSIZECFG bit will be forced to be of the same value as that of PERSIZECFG bit.
12:11	PERSIZECFG	R/W	Peripheral Data Size Configure 00: Byte (8 bits) 01: Half word (16 bits) 10: Word (32 bits) 11: Reserved These bits can be written only when EN bit is 0.
10	MEMIM	R/W	Memory Increment Mode 0: The memory address pointer is fixed 1: After each data transmission, the memory address pointer will increase This bit can be written only when EN bit is 0.
9	PERIM	R/W	Peripheral Increment Mode 0: The peripheral address pointer is fixed 1: After each data transmission, the peripheral address pointer will increase This bit can be written only when EN bit is 0.

Field	Name	R/W	Description
8	CIRCMEN	R/W	Circular Mode Enable This bit can be set to 1 or 0 by software, or be set to 0 by hardware. 0: Disable 1: Enable If the peripheral is set as the stream controller and the data stream is enabled, this bit will be automatically forced to 0 by hardware. If DMA transmission is ended, switch the target memory area, enable the data stream, and this bit will be automatically forced to 1 by hardware.
7:6	DIRCFG	R/W	Data Transfer Direction Configure 00: From peripheral to memory 01: From memory to peripheral 10: From memory to memory 11: Reserved These bits can be written only when EN bit is 0.
5	PERFC	R/W	Peripheral Flow Controller 0: DMA is stream controller 1: The peripheral is stream controller This bit can be written only when the EN bit is 0; when the memory-to-memory mode is selected, this bit will be automatically forced to zero by the hardware.
4	TXCIEN	R/W	Transfer Complete Interrupt Enable 0: Disable 1: Enable
3	HTXIEN	R/W	Half Transfer Interrupt Enable 0: Disable 1: Enable
2	TXEIEN	R/W	Transfer Error Interrupt Enable 0: Disable 1: Enable
1	DMEIEN	R/W	Direct Mode Error Interrupt Enable 0: Disable 1: Enable
0	EN	R/W	Stream Enable 0: Disable 1: Enable This bit shall be set to 0 by hardware in any of the following situations: When DMA transmission ends When transmission error occurs to AHB main bus When the FIFO threshold on the memory AHB port is incompatible with the burst size

10.5.6 DMA data stream x data item number register (DMA_NDATA) (x=0…7)

Offset address: 0x14+0x18× (data stream number)

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15:0	NDATA	R/W	Number Of Data Items To Transfer The number of data items to be transmitted is 0-65535. This register can be operated only when the data stream is disabled. After the data stream is enabled, this register is read-only to indicate the number of remaining data items to be transmitted. After each DMA transmission, this register will decrease. This register is 0 after completion of transmission, and the initial value will be automatically reloaded in any of the following circumstances: Configure the data stream in circular mode Re-enable the data stream

10.5.7 DMA data stream x peripheral address register (DMA_PADDR) (x=0…7)

Offset address: 0x18+0x18× (data stream number)

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	PADDR	R/W	Peripheral Address Base address of peripheral data register of read/write data. This bit can be written only when EN bit is 0.

10.5.8 DMA data stream x memory 0 address register (DMA_M0ADDR) (x=0…7)

Offset address: 0x1C+0x18× (data stream number)

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	M0ADDR	R/W	Memory 0 Address Base address of memory 0 of read/write data. These bits are write-protected, and can be written only in any of the following circumstances: Disable data stream Enable the data stream and set CTARG bit of DMA_SCFG register to 1

10.5.9 DMA data stream x memory 1 address register (DMA_M1ADDR) (x=0…7)

Offset address: 0x20+0x18× (data stream number)

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	M1ADDR	R/W	Memory 1 Address Base address of memory 1 of read/write data. This register is only applicable to double-buffer mode. These bits are write-protected, and can be written only in any of the following circumstances: Disable data stream Enable the data stream and set CTARG bit of DMA_SCFG register to 0

10.5.10 DMA data stream x FIFO control register (DMA_FCTRL) (x=0…7)

Offset address: 0x24+0x18× (data stream number)

Reset value: 0x0000 0020

Field	Name	R/W	Description
31:8	Reserved		
7	FEIEN	R/W	FIFO Error Interrupt Enable 0: Disable 1: Enable
6	Reserved		
5:3	FSTS	R	FIFO Status 000: 0<fifo_level<1/4 001: 1/4<fifo_level<1/2 010: 1/2<fifo_level<3/4 011: 3/4<fifo_level<full 100: FIFO is empty 101: FIFO is full Others: Meaningles These bits are invalid in direct mode.
2	DMDEN	R/W	Direct Mode Disable 0: Enable direct mode 1: Disable direct mode This bit can be written only when the EN bit is 0; when the memory-to-memory mode is selected and EN bit is 1, this bit will be set to 1 by hardware.
1:0	FTHSEL	R/W	FIFO Threshold Select 00: 1/4 of FIFO capacity 01: 1/2 of FIFO capacity 10: 3/4 of FIFO capacity 11: Full FIFO capacity In direct mode, these bits are invalid, and they can be written only when EN bit is 1.

11 Debug MCU (DBGMCU)

11.1 Full Name and Abbreviation of Terms

Table 38 Full Name and Abbreviation Description of Terms

Full name in English	English abbreviation
Frame Clock	FCLK
Serial Wire Debug Port	SW-DP

11.2 Introduction

G32R430 MCU series uses Arm® Cortex®-M52 core, and Arm® Cortex®-M52 core includes hardware debug module and supports complex debugging operation. During debugging, the module can make the running core stop at breakpoint, and achieve the effect of querying the internal state of the core and the external state of the system, and after the query is completed, the core and peripheral operation can be restored to continue to execute the program.

Debugging interfaces is serial interface.

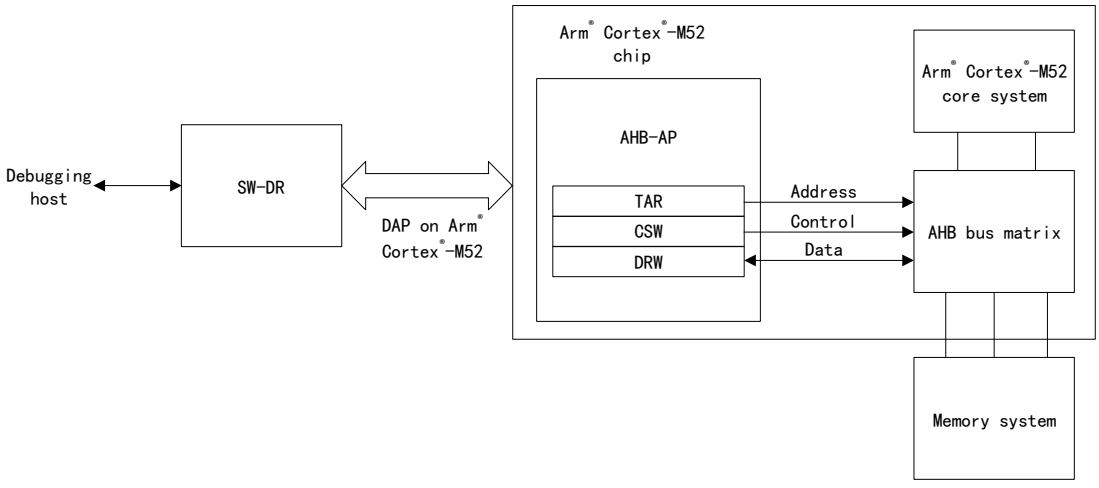
11.3 Main characteristics

(1) Able to replace the core to access AHB bus matrix

(2) Flexible debug pin assignment

(3) MCU debug box (support low-power mode, control peripheral clock, etc.)

Figure 14 Debugging Block Diagram



11.4 Functional Description

11.4.1 Debug pin function configuration

(1) Achieve the on-line programming and debugging of the chip

(2) Using KEIL/IAR and other software to implement on-line debugging, downloading and programming

(3) Flexible implementation of the offline programmer

Table 39 SWD Pin Description

SW-DP Name	Type	Description	GPIO
SWDIO	IO	SW-DP data input and output	PB10
SWCLK	I	SW-DP clock	PB5

11.4.2 ID code

11.4.2.1 MCU device ID code

G32R430 MCU series includes a MCU ID code. It can be accessed with SW debugging interface or user code.

11.5 Register address mapping

Table 40 Register Address Mapping

Register name	Description	Address
DBGMCU_ID	ID register	0x00
DBGMCU_APB	APB freeze register	0x0C

11.6 Register functional description

11.6.1 ID register (DBGMCU_ID)

Offset address: 0x00

Reset value: 0x1000 0479

Field	Name	R/W	Description
31:16	REV_ID[15:0]	R	Chip Version
15:0	DEV_ID[15:0]	R	Equipment ID

11.6.2 APB freeze register (DBGMCU_APB)

Offset address: 0x0C

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:18	Reserved		
17	DBG_IWDT_STOP	R/W	IWDT Counter Stopped When Core is Halted in Debug Mode

Field	Name	R/W	Description
			0: When the core stops, IWDT counter continues running 1: When the core stops, IWDT counter stops
16	DBG_WWDT_STOP	R/W	WWDT Counter Stopped When Core is Halted in Debug Mode 0: When the core stops, WWDT counter continues running 1: When the core stops, WWDT counter stops
15	DBG_RTC_STOP	R/W	RTC Counter Stopped When Core is Halted in Debug Mode 0: When the core stops, RTC counter continues running 1: When the core stops, RTC counter stops
14:13	Reserved		
12	DBG_LPTMR_STOP	R/W	LPTMR Counter Stopped When Core is Halted in Debug Mode 0: When the core stops, LPTMR counter continues running 1: When the core stops, LPTMR counter stops
11:7	Reserved		
6	DBG_TMR4_STOP	R/W	TMR4 Counter Stopped When Core is Halted in Debug Mode 0: When the core stops, TMR4 counter continues running 1: When the core stops, TMR4 counter stops
5	DBG_TMR3_STOP	R/W	TMR3 Counter Stopped When Core is Halted in Debug Mode 0: When the core stops, TMR3 counter continues running 1: When the core stops, TMR3 counter stops
4	DBG_TMR2_STOP	R/W	TMR2 Counter Stopped When Core is Halted in Debug Mode 0: When the core stops, TMR2 counter continues running 1: When the core stops, TMR2 counter stops
3:1	Reserved		
0	DBG_TMR1_STOP	R/W	TMR1 Counter Stopped When Core is Halted in Debug Mode 0: When the core stops, TMR1 counter continues running 1: When the core stops, TMR1 counter stops

12 General-Purpose Input/Output Pin (GPIO)

12.1 Full Name and Abbreviation Description of Terms

Table 41 Full Name and Abbreviation Description of Terms

Full name in English	English abbreviation
General-purpose Input/Output	GPIO

12.2 Main characteristics

GPIO ports have four 32-bit configuration registers (GPIOx_MODER, GPIOx_OTYPER, GPIOx_OSPEEDR, and GPIOx_PUPDR) and two 32-bit data registers (GPIOx_IDR, GPIOx_ODR). All GPIO have a 32-bit alternate function selection register (GPIOx_AFR).

(1) Input mode
- Input states: Analog input, floating input, pull-up/pull-down input
- Input data from the input data register (GPIOx_IDR) or from a peripheral (alternate function input)

(2) Output mode
- Output states: Push-pull output (with pull-up/pull-down function) or open-drain output (with pull-up/pull-down function)
- The output rate of each I/O can be configured.
- Output data from the output data register (GPIOx_ODR) or from a peripheral (alternate function output)

(3) Multiplexing function
- Push-pull multiplexing function
- Open-drain multiplexing function
- The alternate function can be used for digital peripherals

(4) Analog function
- Analog input and output can be used for analog peripherals and in low-power modes

(5) The RX alternate-function pin for USART supports 5V-tolerant I/O

(6) Upon entering the STOP/STANDBY mode, the GPIO states are latched

(7) Four GPIO support wake-up on rising edge, falling edge, or both edges in low-power STANDBY mode.

(8) All GPIO can be used as an external interrupt/STOP wake-up line

(9) It supports fast switching of locked I/O configuration functions, which change every two clock cycles.

(10) I/O can be used as GPIOs or as one of multiple peripheral functions via pin multiplexing.

12.3 Structure block diagram

Figure 15 GPIO Structure Block Diagram

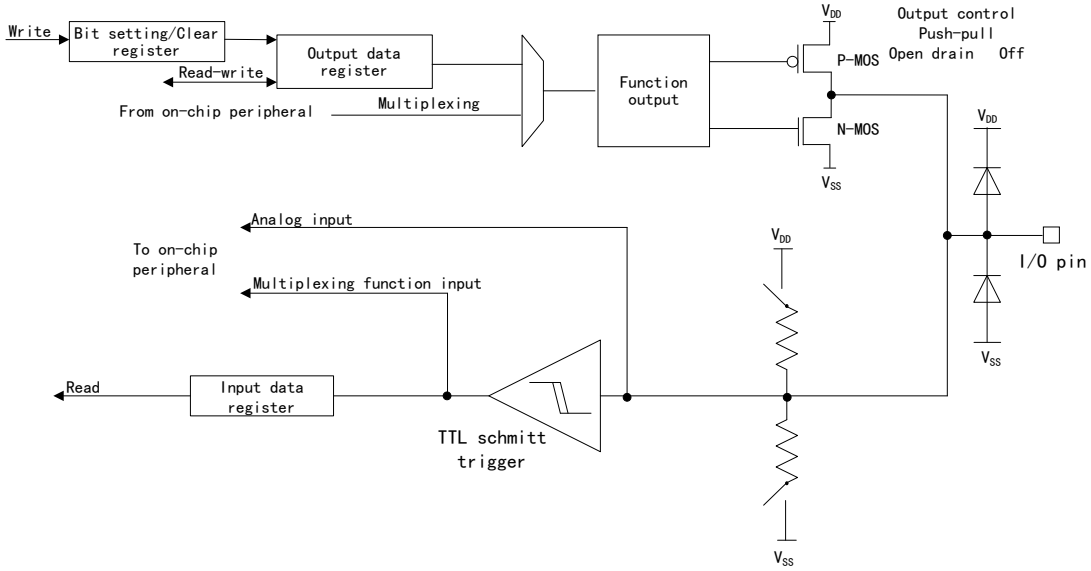
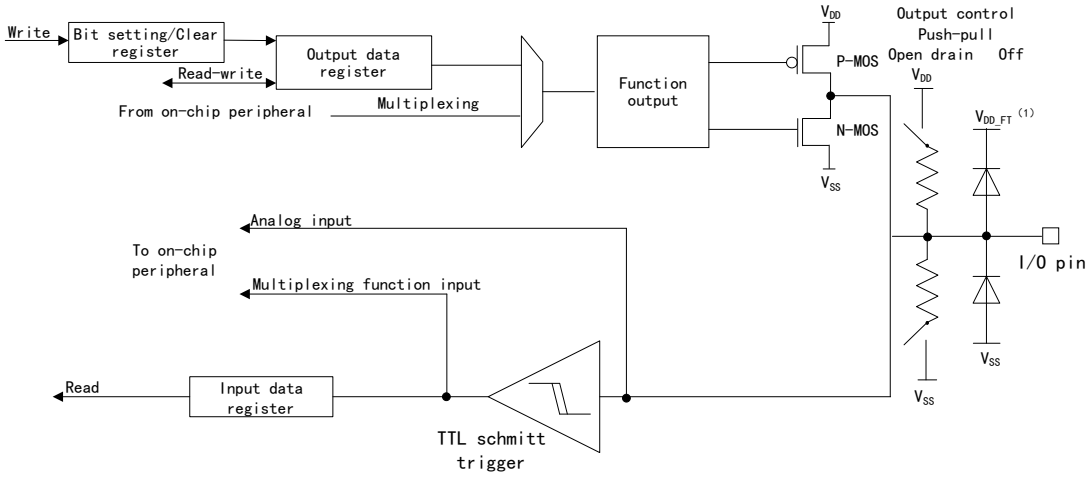


Figure 16 5V-compatible GPIO Structure Block Diagram



Note: Different from V_{DD}, V_{DD_FT} is special for 5V-tolerant GPIO pins.

12.4 Functional description

Each bit of the GPIO port can be individually configured into multiple modes by software:

- Input floating

- Input pull-up
- Input pull-down
- Analog input
- Push-pull output with pull-up/pull-down
- Open-drain output with pull-up/pull-down
- Push-pull alternate function with pull-up/pull-down
- Open-drain alternate function with pull-up/pull-down

All I/O port bits can be freely programmed, but the I/O port registers must be accessed in 32-bit words, half words, or bytes. This avoids the risk of IRQ between read and change accesses.

All GPIO interfaces have external interrupt capability.

Table 42 IO Port Configuration

GPIO_MDR[2i+1:2i]	GPIO_OTR[i]	GPIO_PUPDR[2i+1:2i]	I/O configuration
10	0	x0	Push-pull output
		01	Push-pull output pull-up
		11	Push-pull output pull-down
	1	x0	Open-drain output
		01	Open-drain output pull-up
		11	Open-drain output pull-down
11	0	x0	Alternate function push-pull output
		01	Alternate function push-pull output pull-up
		11	Alternate function push-pull output pull-down
	1	x0	Alternate function open-drain output
		01	Alternate function push-pull open-drain pull-up
		11	Alternate function open-drain output pull-down
01	-	x0	Input floating
		01	Input pull-up
		11	Input pull-down
00	-	x0	Analog

12.4.1 IO status during reset and just after reset

If the alternate function is not enabled during and after GPIO reset, the I/O port will be configured as analog mode, and in such case the pull-up/pull-down resistor is disabled in input mode.

When the pin is configured as output, the value written to the output data register (GPIOx_ODR) will be output onto the I/O pin. The output driver can be

set to the push-pull mode or open-drain mode (only drives low level, and high level is high-impedance state).

The input data register (GPIOx_IDR) captures the data on I/O pins in each AHB clock cycle. All GPIO pins have weak internal pull-up and pull-down resistors, which can be activated or not based on the value in the GPIOx_PUPDR register.

12.4.2 IO pin multiplexer and mapping

The I/O pin can be connected to on-board peripherals or modules via a multiplexer. The multiplexer allows only one peripheral's alternate function (AF) to be connected to a single I/O pin at a time, avoiding conflicts between peripherals available on the same I/O pin.

Each I/O pin can be configured via the GPIOx_AFR register to have up to 4 alternate function inputs (AF0~AF3):

- After reset, the multiplexer selects AF0. The I/O can be configured as alternate function mode via the GPIOx_MODER register.
- Please refer to the Datasheet for specific alternate function information for each pin.

Furthermore, to optimize the number of peripherals available in smaller device packages, some alternate functions can be remapped to some other I/O pins. If an I/O needs to be used in a given configuration, operations shall be performed according to the following steps:

(1) Debug function: After device reset, these pins are assigned as alternate function pins and can be used immediately by the debugger host;

(2) GPIO: Configure the required I/O as output, input, or analog input in the GPIOx_MODER register;

(3) Peripheral alternate function:
- Connect I/O to the required AFx in the GPIOx_AFRL or GPIOx_AFRH.
- Use the GPIOx_OTYPER, GPIOx_PUPDR and GPIOx_OSPEEDER registers to select the type, pull-up/pull-down, and output speed, respectively.
- Configure the required I/O as the alternate function in the GPIOx_MODER register.

(4) Other functions:
- For ADC, DAC, and COMP, configure the required I/O in analog mode, write to the GPIOx_MODER register, and configure the required function in the ADC, DAC, and COMP registers.

Note: These functions are controlled for output by the corresponding peripherals. Therefore, before enabling other function outputs of the peripheral control registers, be careful to select the analog function of the I/O port

For other functions such as RTC, WKUPx, and oscillators, configure the required functions in the relevant RTC, PMU, and RCM registers. These functions take precedence over the configuration in the standard GPIO registers.

12.4.3 Input mode

In the input mode, it can be set as pull-up, pull-down, floating and analog input.

When GPIO is configured as input mode, all GPIO pins have an internal weak pull-up and pull-down resistor, which can be activated or broken.

Pull-up, pull-down, and floating modes

In (pull-up, pull-down, floating) input mode

- Schmitt trigger is enabled
- Disable output buffer
- Connect the pull-up and pull-down resistors according to the value of the GPIOx_PUPDR register.
- The input data register GPIOx_IDR captures the data on I/O pins in each AHB clock cycle
- Read the I/O status by the input data register GPIOx_IDR

The initial level state of the floating input mode is uncertain and is easy to be disturbed by the outside; when connecting the equipment, it is determined by the external input level (except for the very high impedance).

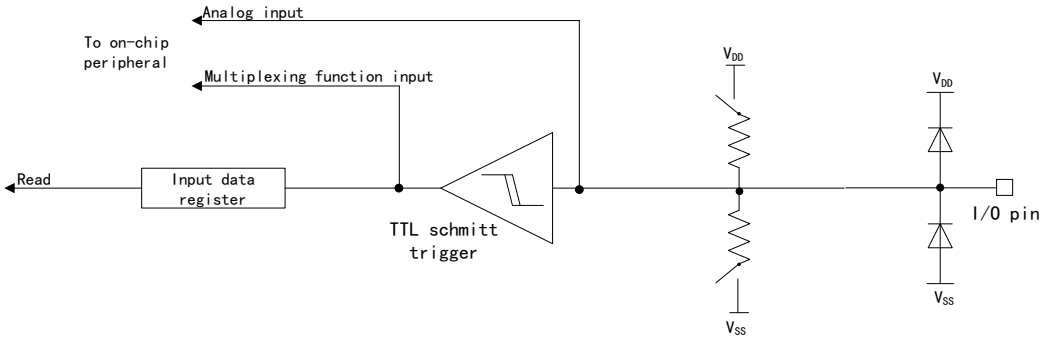
The initial level state of pull-up/pull-down input mode is high if pull-up, and low if pull-down; when connecting the equipment, it is determined by the external input level and load impedance.

Analog input mode

In analog input mode

- Disable output buffer
- The input of Schmitt trigger is disabled, and the output value of Schmitt trigger is forced to be 0
- Weak pull-up and pull-down resistors are disabled
- The value of port input state register is 0

Figure 17 Input Mode Structure



12.4.4 Output mode

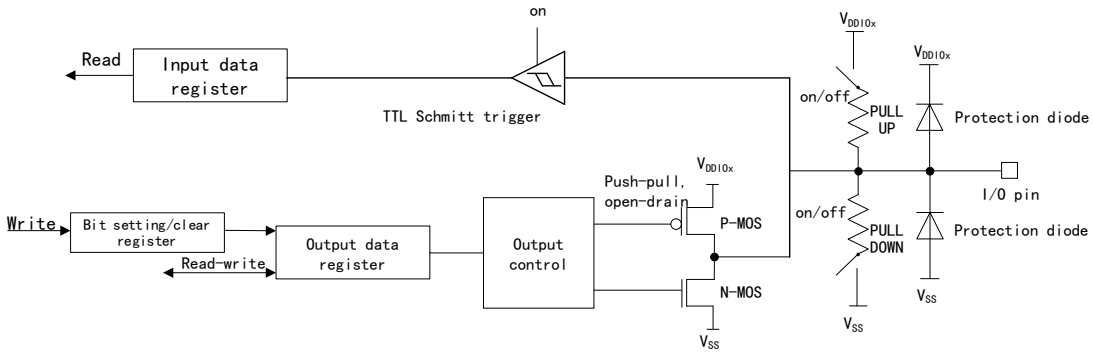
In the output mode, it can be set as push-pull output and open-drain output.

When GPIO is configured as the output pin, the output speed of the port can be configured and the output drive mode (push-pull/open-drain) can be selected.

In output mode

- Activate output buffer
- Push-pull mode: The double MOS transistors work alternately. When the value of the output data register is "0", the N-MOS transistor is activated; when the value of the output data register is "1", the P-MOS transistor is activated.
- Open-drain mode: Only the N-MOS transistor works. When the value of the output data register is "0", the N-MOS transistor is activated; when the value of the output data register is "1", the port remains in a high-impedance state.
- Activate Schmitt trigger input
- Connect the pull-up and pull-down resistors according to the value of the GPIOx_PUPDR register.
- The input data register GPIOx_IDR captures the data on I/O pins in each AHB clock cycle. Read the actual I/O status through the input data register GPIOx_IDR
- Read the finally written value through the output data register GPIOx_ODR

Figure 18 Output Mode I/O Structure



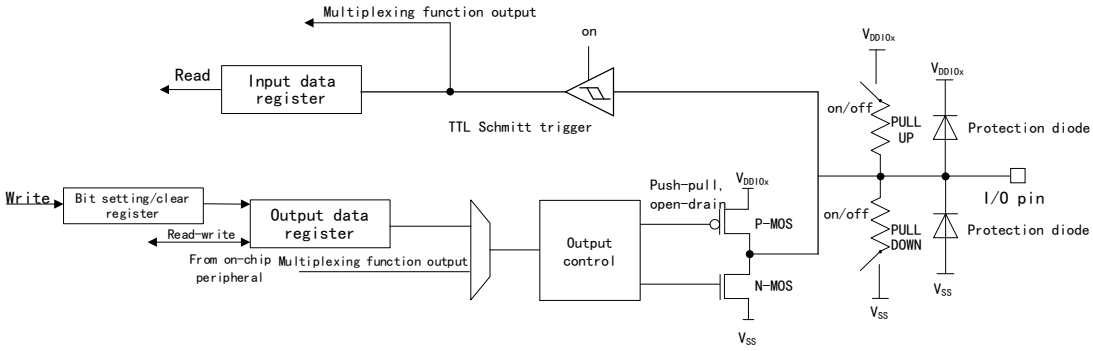
12.4.5 Multiplexing mode

In multiplexing mode, it can be set as push-pull multiplexing and open-drain multiplexing.

In push-pull/open drain multiplexing mode

- Enable the output buffer
- Output buffer is driven by peripheral
- Activate Schmitt trigger input
- Activate/Deactivate the weak pull-up and pull-down resistors according to the value of the GPIOx_PUPDR register
- The input data register GPIOx_IDR captures the data on I/O pins in each AHB clock cycle
- Read the actual I/O status through the input data register GPIOx_IDR
- Read the finally written value through the output data register GPIOx_ODR

Figure 19 Alternate Function Mode I/O Structure



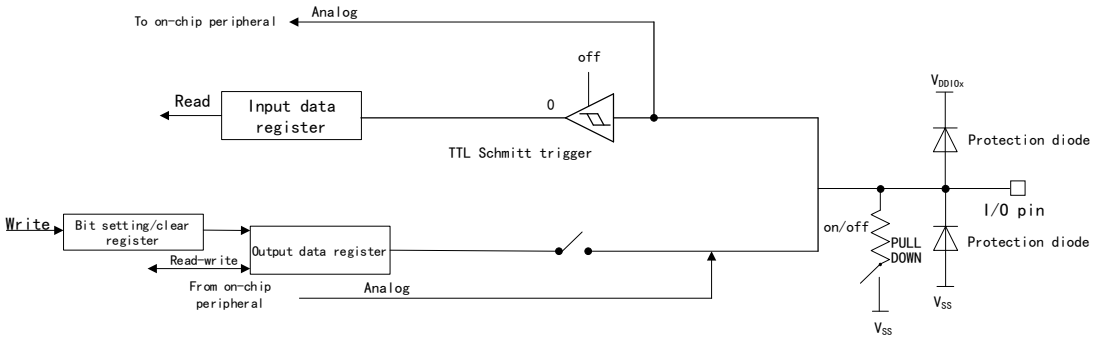
12.4.6 Analog mode

In analog mode

- Disable output buffer

- The Schmitt trigger input is disabled, providing zero power consumption for each analog value of the I/O pin. The output value of the Schmitt trigger is forced to a fixed value of 0.
- Weak pull-up is disabled by hardware.
- Weak pull-down is configurable.
- The value in the GPIOx_IDR register is 0.

Figure 20 High-impedance Analog Mode I/O Structure



12.4.7 Port control register

Each GPIO port has four 32-bit configuration registers (GPIOx_MODER, GPIOx_OTYPER, GPIOx_OSPEEDR and GPIOx_PUPDR) to configure up to 16 I/Os.

- GPIOx_MODER: Port mode register, used to select input/output/alternate/analog mode.
- GPIOx_OTYPER: Port output mode register, used to select push-pull/open-drain mode.
- GPIOx_OSPEEDR: Port output speed register.
- GPIOx_PUPDR: Port pull-up/pull-down register (regardless of whether the I/O direction is set as input or output)

12.4.8 Port data register

Each GPIO port has two 32-bit data registers (GPIOx_IDR, GPIOx_ODR).

- GPIOx_IDR: Port bit input data register, which is a read-only register for storing data input through the I/O
- GPIOx_ODR: Port bit output data register, which can be read, written and accessed for storing the data to be output.

12.4.9 Bit set and bit clear

GPIO has a 32-bit bit set/reset register (GPIOx_BSRR), allowing the application to set and reset each individual bit in the output data register (GPIOx_ODR).

The GPIOx_BSRR set/reset register is twice the size of GPIOx_ODR. Each bit in GPIOx_ODR corresponds to two control bits in GPIOx_BSRR: BS(i) and BR(i):

- When 1 is written, the BS(i) bit sets the corresponding ODR(i) bit.

- When 1 is written, the BR(i) bit resets the corresponding ODR(i) bit.
- When 0 is written, it has no effect on the bits in GPIOx_ODR. Note: If both BS(i) and BR(i) bits are set, BS(i) has priority.

Using the GPIOx_BSRR register for bit operation of GPIOx_ODR is single. It does not lock the bits of GPIOx_ODR, which can always be accessed directly.

When individual bits of GPIOx_ODR are programmed, the software does not need to disable interrupts. In a single AHB write operation, only one or more bits can be changed.

12.4.10 GPIO alternate function input and output

Each GPIO has GPIOx_AFRL and GPIOx_AFRH alternate function registers, which can connect the alternate function to other pins as required by the application. Since the alternate function input and output share the AF selection signal, a single channel is selected for the alternate function input/output of the given I/O. Please consult this chip's Datasheet for specific alternate function information for each GPIO pin.

12.4.11 External interrupt/wake-up line

All GPIO ports have external interrupt function. If you want to use external interrupt line, the port must be configured as input mode.

12.4.12 Other cases of pins being used as GPIO

HSECLK/LSECLK pin is used as GPIO

- When the HSECLK or LSECLK oscillator is disabled (the default state after reset), the relevant oscillator pins can be used as a general GPIO.
- When the HSECLK or LSECLK oscillator is enabled (by setting the [HSEON] or [LSEON] in the RCM_RCCR register), the oscillator controls its associated pins, and the GPIO configuration of these pins is ineffective.
- When the oscillator is configured in user external clock mode, only the OSC_IN or OSC32_IN pin is reserved for clock input, while the OSC_OUT or OSC32_OUT pin can still be used as a general GPIO.

12.5 Register address mapping

Table 43 GPIO Register Address Mapping

Register name	Description	Offset address
GPIO_MODERx	Port mode selection register x	0x00+0x04*x
GPIO_PUPDRx	Port pull-up/pull-down control register x	0x10+0x04*x
GPIO_OTYPER1	Port output mode register 1	0x20

GPIO_OTYPER2	Port output mode register 2	0x24
GPIO_OSPEEDRx	Port output speed register x	0x28+0x04*x
GPIO_IDR1	Port input data register 1	0x38
GPIO_IDR2	Port input data register 2	0x3C
GPIO_ODR1	Port output data register 1	0x40
GPIO_ODR2	Port output data register 2	0x44
GPIO_AFSELRx	Port alternate function selection register x	0x48+0x04*x
GPIO_FILTERR1	Filter enable register 1	0x58
GPIO_FILTERR2	Filter enable register 2	0x5C
GPIO_BSRRx	Port set/reset register x	0x60+0x04*x
GPIO_BRRx	Port reset register x	0x70+0x04*x
GPIO_SWR1	Port analog switch register 1	0x80
GPIO_SWR2	Port analog switch register 2	0x84

12.6 Register functional description

The value of x is: 0/1/2/3, corresponding respectively to GPIOA/GPIOB/GPIOC/GPIOD respectively.

The values of y is: 0~5/0~12/0~12/0~15, corresponding respectively to GPIOA/GPIOB/GPIOC/GPIOD.

PA/PB/PC/PD correspond respectively to GPIOA/GPIOB/GPIOC/GPIOD.

12.6.1 Port mode selection register (GPIO_MODERx)

Offset address: 0x00+0x04*x

Reset value: 0x00F0 0C00 (GPIOB)

0x0000 000 (for other ports)

Field	Name	R/W	Description
31:0	MODEy[1:0]	R/W	Port x Pin y mode configuration (y=0…15) 00: Analog mode (state after reset) 01: Analog input mode 10: General output mode 11: Alternate function mode For these bits, the I/O port mode is configured through write operation of software.

12.6.2 Port pull-up/pull-down control register (GPIO_PUPDRx)

Offset address: 0x10+0x04*x

Reset value: 0x0050 0C00 (GPIOB)

0x0000 000 (for other ports)

Field	Name	R/W	Description
31:16	PUPDSELy	R/W	Port x Pin y Pull-up/Pull-down enable (y=0…15) 0: Pull up 1: Pull down
15:0	PUPDENy	R/W	Port x Pin y Pull-up/Pull-down enable (y=0…15) 0: Disable pull-up/pull-down function 1: Enable pull-up/pull-down function

12.6.3 Port output mode register (GPIO_OTYPER1)

Offset address: 0x20

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	OTy	R/W	Port B Pin y output mode configuration (y=0…15) 0: Push-pull output (reset state) 1: Open-drain output
15:0	OTy	R/W	Port A Pin y output mode configuration (y=0…15) 0: Push-pull output (reset state) 1: Open-drain output

12.6.4 Port output mode register (GPIO_OTYPER2)

Offset address: 0x24

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	OTy	R/W	Port D Pin y output mode configuration (y=0…15) 0: Push-pull output (reset state) 1: Open-drain output
15:0	OTy	R/W	Port C Pin y output mode configuration (y=0…15) 0: Push-pull output (reset state) 1: Open-drain output

12.6.5 Port output speed register (GPIO_OSPEEDRx)

Offset address: 0x28+0x04*x

Reset value: 0x0000 0AAA (GPIOA)

0x02BA AAAA (GPIOB)

0x02AA AAAA (GPIOC)

0xAAAA AAAA (GPIOD)

Field	Name	R/W	Description
31:0	OSPEEDy[1:0]	R/W	Port x Pin y Output Speed Select (y=0…15) 0x: Low speed 10: Medium speed 11: High speed The speed of I/O port is written by software.

12.6.6 Port input data register (GPIO_IDR1)

Offset address: 0x38

Reset value: 0xXXXX XXXX

Field	Name	R	Description
31:16	IDy	R	Port B Pin y Input Data (y=0…15) These bits can only be read to store the input values of the corresponding I/O ports.
15:0	IDy	R	Port A Pin y Input Data (y=0…15) These bits can only be read to store the input values of the corresponding I/O ports.

12.6.7 Port input data register (GPIO_IDR2)

Offset address: 0x3C

Reset value: 0xXXXX XXXX

Field	Name	R	Description
31:16	IDy	R	Port D Pin y Input Data (y=0…15) These bits can only be read to store the input values of the corresponding I/O ports.
15:0	IDy	R	Port C Pin y Input Data (y=0…15) These bits can only be read to store the input values of the corresponding I/O ports.

12.6.8 Port output data register (GPIO_ODR1)

Offset address: 0x40

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	ODy	W	Port B Pin y Output Data (y=0…15) Read and write operations can be performed by software. For atomic bit setting/clearing, the ODy bit can be set separately by writing to GPIOx_BSRR register.
15:0	ODy	W	Port A Pin y Output Data (y=0…15) Read and write operations can be performed by software. For atomic bit setting/clearing, the ODy bit can be set separately by writing to GPIOx_BSRR register.

12.6.9 Port output data register (GPIO_ODR2)

Offset address: 0x44

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	ODy	W	Port D Pin y Output Data (y=0…15) Read and write operations can be performed by software. For atomic bit setting/clearing, the ODy bit can be set separately by writing to GPIOx_BSRR register.
15:0	ODy	W	Port C Pin y Output Data (y=0…15) Read and write operations can be performed by software. For atomic bit setting/clearing, the ODy bit can be set separately by writing to GPIOx_BSRR register.

12.6.10 Port alternate function selection register (GPIO_AFSELRx)

Offset address: 0x48+0x04*x

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	AFSELy[1:0]	R/W	Select the alternate function of Port x Pin y (y=0….15) These bits can be read by software to configure the multiplexing function of the port. AFSELy selection: 00: AF0 01: AF1 10: AF2 11: AF3

12.6.11 Filter enable register (GPIO_FILTERR1)

Offset address: 0x58

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	FLTENy	R/W	Port B Pin y filter enable (y=0…15) 0: Disable filter function for Port B Pin y 1: Enable filter function for Port B Pin y
15:0	FLTENy	R/W	Port A Pin y filter enable (y=0…15) 0: Disable filter function for Port A Pin y 1: Enable filter function for Port A Pin y

12.6.12 Filter enable register (GPIO_FILTERR2)

Offset address: 0x5C

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	FLTENy	R/W	Port D Pin y filter enable (y=0…15) 0: Disable filter function for Port D Pin y 1: Enable filter function for Port D Pin y
15:0	FLTENy	R/W	Port C Pin y filter enable (y=0…15) 0: Disable filter function for Port C Pin y 1: Enable filter function for Port C Pin y

12.6.13 Port set/reset register (GPIO_BSRRx)

Offset address: 0x60+0x04*x

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	BRy	R	Port x Pin y reset (y=0…15) 0: No effect on Port x Pin y 1: Reset Port x Pin y Note: If both BS and BR bits for the corresponding pin are set to 1, the BS bit has higher priority.
15:0	BSy	R	Port x Pin y set (y=0…15) 0: No effect on Port x Pin y 1: Reset Port x Pin y

12.6.14 Port reset register (GPIO_BRRx)

Offset address: 0x70+0x04*x

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15:0	BRy	R/W	Port x Pin y reset (y=0…15) 0: No effect on Port x Pin y 1: Reset Port x Pin y

12.6.15 Port analog switch register (GPIO_SWR1)

Offset address: 0x80

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	SWy	R/W	Port B Pin y analog switch control (y=0…15) For 3.3V standard I/O: 0: The analog switch at pin y is open 1: The analog switch at pin y is closed For example, when configuring the PB2 pin for ADC function, the corresponding register position needs to be set to 1. For 5V tolerant I/O: 0: Digital 5V tolerance input 1: Simulate 3.3V input and output For example, when configuring the PB4 pin for the DAC function, the corresponding register position needs to be set to 1.
15:0	SWy	R/W	Port x Pin y analog switch control (y=0…15) For 3.3V standard I/O: 0: The analog switch at pin y is open 1: The analog switch at pin y is closed For 5V tolerant I/O: 0: Digital 5V tolerance input 1: Simulate 3.3V input and output

12.6.16 Port analog switch register (GPIO_SWR2)

Offset address: 0x84

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	SWy	R/W	Port D Pin y analog switch control (y=0…15) For 3.3V standard I/O: 0: The analog switch at pin y is open 1: The analog switch at pin y is closed For 5V tolerant I/O: 0: Digital 5V tolerance input 1: Simulate 3.3V input and output
15:0	SWy	R/W	Port C Pin y analog switch control (y=0…15) For 3.3V standard I/O: 0: The analog switch at pin y is open 1: The analog switch at pin y is closed For example, when configuring the PC4 pin for ADC function, the corresponding register position needs to be set to 1. For 5V tolerant I/O: 0: Digital 5V tolerance input

Field	Name	R/W	Description
			1: Simulate 3.3V input and output For example, when configuring the PC12 pin for ADC function, the corresponding register position needs to be set to 1.

13 Timer overview

13.1 Full Name and Abbreviation of Terms

Table 44 Full Name and Abbreviation Description of Terms

Full name in English	English abbreviation
Timer	TMR
Update	U
Request	R
Event	EV
Compare	C
Compare	C
Length	LEN

13.2 Timer categories and main differences

The product features embedded the 16-bit advanced timer, 16-bit general-purpose timers, independent watchdog timer, window watchdog timer, low-power timer, and system tick timer (the watchdog timers are described in detail in another chapter).

The advanced timer can be used to measure input signal pulse length (input capture), brake input, encoder interface, ETR input, or generate output waveforms (output compare, single pulse output, complementary PWM with deadband insertion, or ordinary PWM output), or serve as a repetitive timer base.

The general-purpose timer can also be used as a simple time base. Its functions are less complex than those of the advanced timer, with main differences in the number of non-complementary output channels, complementary output channel groups, brake functions, deadband insertion, and repetitive counting capabilities.

The watchdog timer is used to detect whether the program is running normally. The low-power timer can operate and wake up the system in stop mode.

The system tick timer is an external peripheral of the core, equipped with auto-reload functionality. When the counter reaches zero, it can generate a maskable system interrupt, making it suitable for real-time operating systems and ordinary delays.

The main differences of timers included in the products are shown in the table below:

Table 45 Main Differences among Timers Included in the Products

Item	Specific content/Category	Advanced timer	General-purpose timer	Low-power consumption timer	SysTick Timer
Name		TMR1	TMR2/3/4	LPTMR	SysTick Timer
Clock source	Internal clock	Yes	Yes	Yes	Yes
	External input	Yes	Yes	No	No
	External trigger	Yes	Yes	No	No
	Internal trigger	Yes	Yes	No	No
Internal trigger output		1	1	0	0
Timebase unit	Counter resolution	16-bit	16-bit	16-bit	24-bit
	Counting mode	Count up Count down Count up/down	Count up Count down Count up/down	Count up	Count down
	Prescaler coefficient	Any integer between 1~65536	Any integer between 1~65536	Any integer between 1~65536	No
Pin Characteristics	External trigger signal input pin	1 channel	1 channel	0	0
	Brake input pin	2 channels	0	0	0
	Complementary channel pin	3 groups	0	0	0
	Non-complementary channel pin	1 channel	4 channels	0	0
Channel	Capture/Compare channel	4	4	0	0
	Complementary output channel	3	0	0	0
Function	Generate DMA request	Yes	Yes	No	No
	Counting function	Yes	Yes	-	-
	PWM input	Yes	Yes	-	-
	Brake input	Yes	-	-	-
	Encoder interface	Yes	Yes	-	-
	ETR input	Yes	Yes	-	-
	PWM output	Yes	Yes	-	-

Item	Specific content/Category	Advanced timer	General-purpose timer	Low-power consumption timer	SysTick Timer
	Forced output	Yes	Yes	-	-
	Single pulse output	Yes	Yes	-	-
	Complementary PWM with deadband insertion	Yes	-	-	-
	Synchronization or event linking function	Yes	Yes	-	-
	Counter can be frozen in debug mode	Yes	Yes	-	-
	Comparator output can be connected to a timer for counting	Yes	Yes	-	-
	Run and wake-up system in stop mode	-	-	Yes	-
	Auto-reload function	Yes	Yes	Yes	Yes

Timer terms

Table 46 Definitions and Terms of Pins

Name	Description
TMRx_ETR	External trigger signal of Timer x
TMRx_CH1, TMRx_CH2, TMRx_CH3, TMRx_CH4	Channel 1/2/3/4 of Timer x
TMRx_ChyN	Complementary output channel y of Timer x
TMRx_BKIN	Braking signal of Timer x

Table 47 Definitions and Terms of Internal Signals

Name	Description
ETR	TMRx_ETR external trigger signal
ETRF	External trigger filter
ETRP	External trigger prescaler
-	

Name	Description
ITR，ITR0，ITR1	Internal trigger
TRGI	Clock/Trigger/Slave mode controller trigger input
TIF_ED	Timer input filter edge detection
-	
CK_PSC	Prescaler clock
CK_CNT	Counter clock
PSC	Prescaler
CNT	Counter
AUTORLD	Autoload register
-	
Tix，TI1	Timer input
TixF，TI1F	Timer input filter
TI1_ED	Timer input edge detection
TixFPx，TI1FP1	Timer input filter polarity
Icx，IC1	Input capture
IcxPS，IC1PS	Input capture prescaler
TRC	Trigger capture
BRK	Braking signal
-	
Ocx，OC1	Timer output compare channel
OCxREF，OC1REF	Output compare reference signal
-	
TGI	Trigger interrupt
BI	Braking interrupt
CcxI，CC1I	Capture/Compare interrupt
UEV	Update event
UIFLG	Update interrupt flag

14 Advanced Timers (TMR1)

14.1 Introduction

The advanced timer takes the time base unit as the core, and has the functions of input capture, output compare and braking input, including a 16-bit automatic loading counter. Compared to other timers, the advanced timer adds functionalities like complementary outputs, repetitive counting, and programmable deadband insertion. It can be used not only for controlling high-precision optical/magnetic absolute encoders but also for motor control applications.

14.2 Main characteristics

(1) Timebase unit
- Counter: 16-bit counter, count-up, count-down and center-aligned count
- Prescaler: 16-bit programmable prescaler
- Repeat counter: 16-bit repeat counter
- Autoreload function

(2) Clock source selection
- Internal clock
- External input
- External trigger
- Internal trigger

(3) Input capture function
- Counting function
- PWM input mode (measurement of pulse width, frequency and duty cycle)
- Encoder interface mode

(4) Output compare function
- PWM output mode
- Forced output mode
- Single-pulse mode
- Complementary output and dead zone insertion

(5) Timing function

(6) Braking function

(7) Master/Slave mode controller of timer
- Timers can be synchronized and cascaded
- Support multiple slave modes and synchronization signals

(8) Interrupt output and DMA request event

- Update event (counter overrun/underrun, counter initialization)
- Trigger event (counter start, stop, internal/external trigger)
- Capture/Compare event
- Braking signal input event

(9) The timer has an independent DMA request mechanism

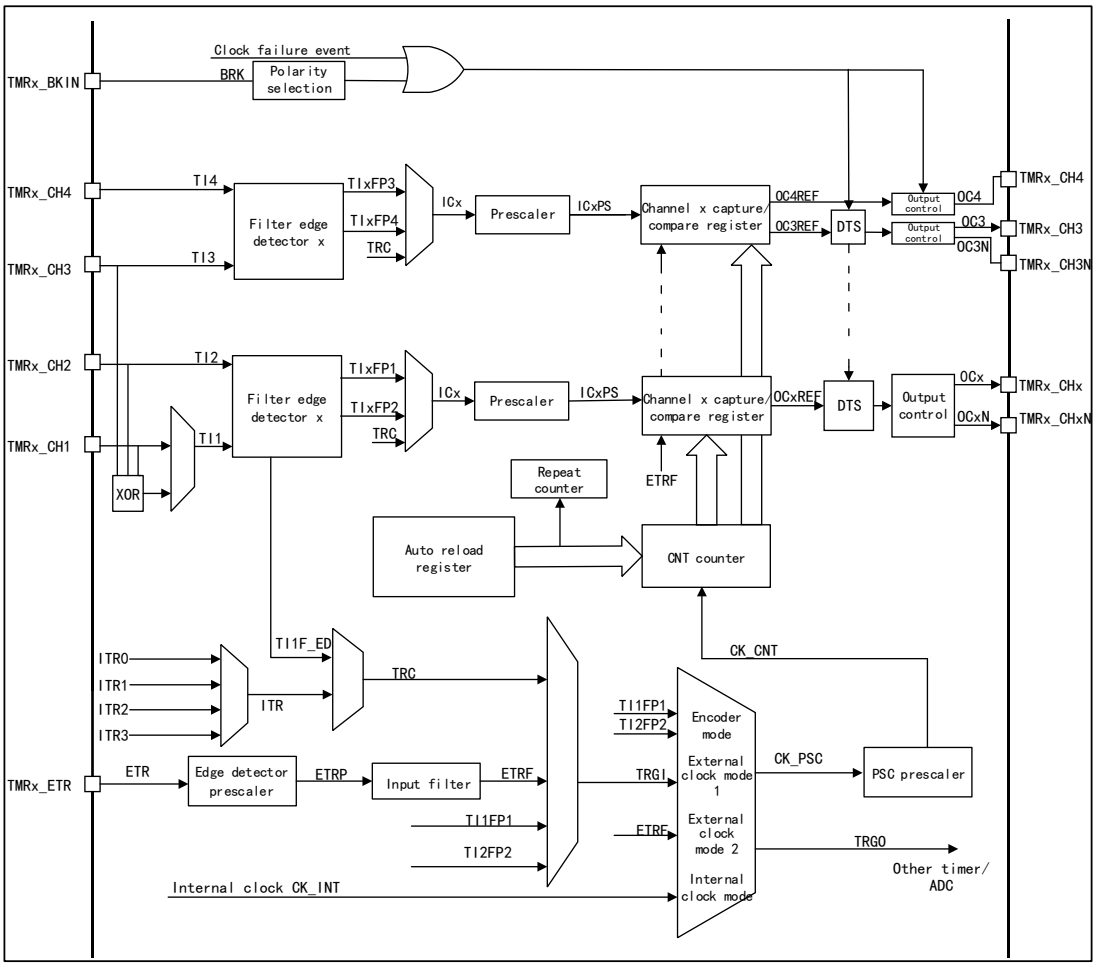
(10) The comparator output can be connected to the timer ETR or input channels for counting

(11) Support incremental (quadrature) encoders and Hall sensor circuits for positioning

(12) Support ETR input (external trigger input) functionality, which can be used as an external clock or for cycle-by-cycle current management

14.3 Structure block diagram

Figure 21 Advanced Timer Structure Block Diagram



14.4 Functional Description

14.4.1 Clock source selection

The advanced timer has four clock sources.

Internal clock

It is TMRx_CLK from RCM, namely the driving clock of the timer; when the slave mode controller is disabled, the clock source CK_PSC of the prescaler is driven by the internal clock CK_INT.

External clock mode 1

The trigger signal generated from the input channel TI1/2/3/4 of the timer after polarity selection and filtering is connected to the slave mode controller to control the work of the counter. Besides, the pulse signal generated by the input of Channel 1 after double-edge detection of the rising edge and the falling edge is logically equal or the future signal is TI1F_ED signal, namely double-edge signal of TIF_ED. Especially the PWM input can only be input by TI1/2.

External clock mode 2

After polarity selection, frequency division and filtering, the signal from external trigger interface (ETR) is connected to the slave mode controller through trigger input selector to control the work of the counter.

Internal trigger input

The timer is set to work in slave mode, and the clock source is the output signal of other timers. At this time, the clock source has no filtering, and the synchronization or cascading between timers can be realized. The master mode timer can reset, start, stop or provide clock for the slave mode timer.

14.4.2 Timebase unit

The timebase unit in the advanced timer contains four registers

- 16-bit counter register (CNT)
- 16-bit auto reload register (AUTORLD)
- 16-bit prescaler register (PSC)
- 8-bit repetition count register (REPCNT)

Repetition register is unique to advanced timer.

Counter CNT

There are three count modes for the counter in the advanced timer

- Count-up mode

- Count-down mode
- Center-aligned mode

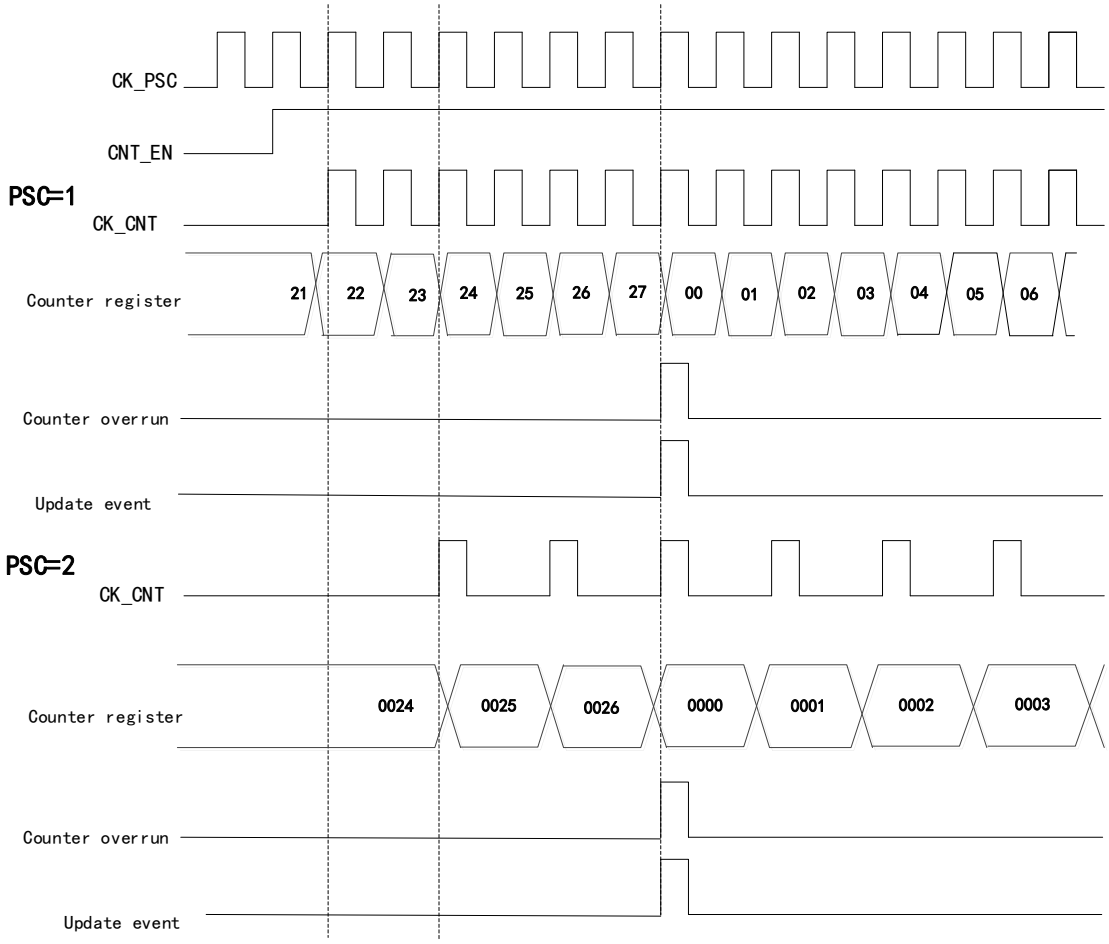
Count-up mode

Set to the count-up mode by configuring CNTDIR bit of control register (TMRx_CTRL1).

When the counter is in count-up mode, the counter will count up from 0; every time a pulse is generated, the counter will increase by 1 and when the value of the counter (TMRx_CNT) is equal to the value of the auto reload (TMRx_AUTORLD), the counter will start to count from 0 again, a count-up overrun event will be generated, and the value of the auto reload (TMRx_AUTORLD) is written in advance. If a repeat counter is used, an update event will be generated when the number of count-up repetitions reaches the number in the repeat counter register plus one time (TMRx_REPCNT+1). Otherwise, an update event will be generated every time the counter overruns. At this time, the repeat count shadow register, the auto reload shadow register and the prescaler buffer will be updated. The update event can be disabled by configuring UD bit of control register TMRx_CTRL1.

The figure below is the timing diagram of count-up mode when the division factor is 1 or 2

Figure 22 Timing Diagram of Count-up Mode when Division Factor is 1 or 2

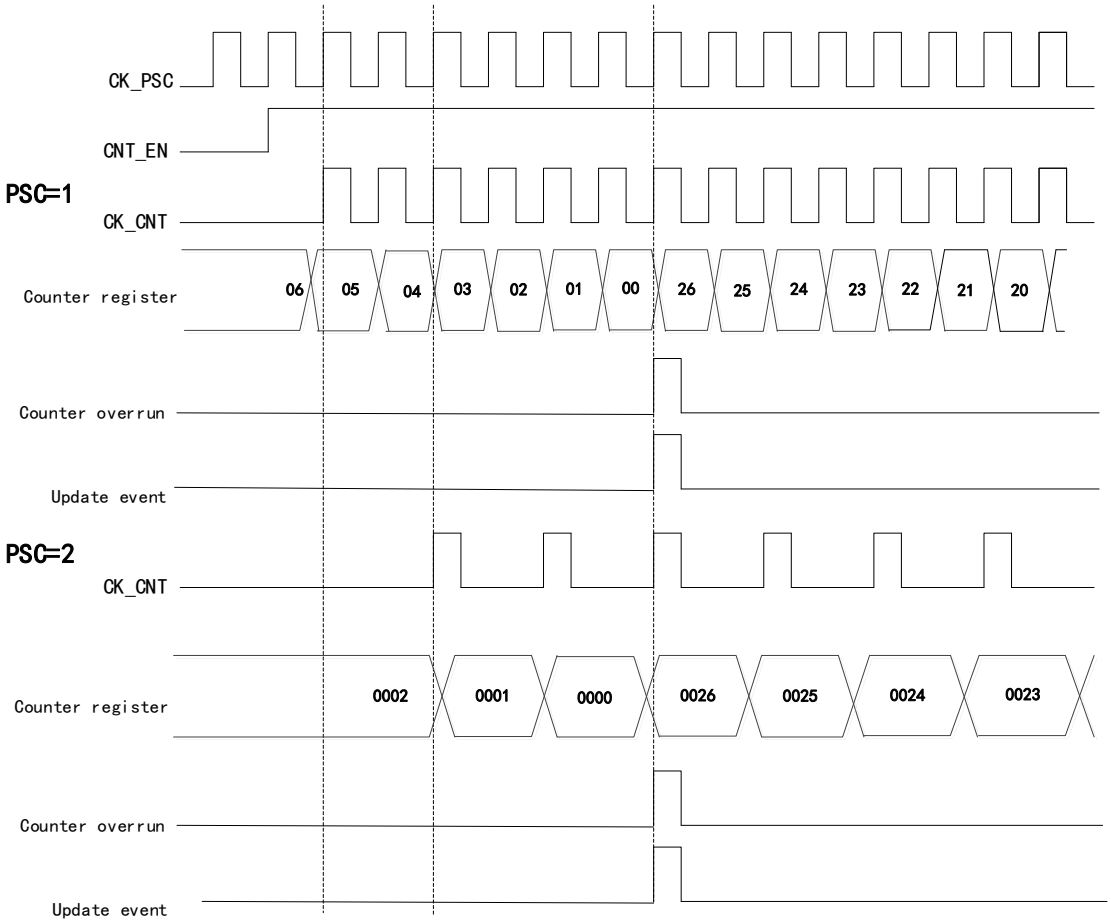


Count-down mode

Set to the count-down mode by configuring CNTDIR bit of control register (TMRx_CTRL1).

When the counter is in count-down mode, it will start to count down from the value of the auto reload (TMRx_AUTORLD); every time a pulse is generated, the counter will decrease by 1 and when it becomes 0, the counter will start to count again from (TMRx_AUTORLD), meanwhile, a count-down overrun event will be generated, and the value of the auto reload (TMRx_AUTORLD) is written in advance. If a repeat counter is used, an update event will be generated when the number of count-down repetitions reaches the number in the repeat counter register plus one time (TMRx_REPCNT+1). Otherwise, an update event will be generated every time the counter underruns. At this time, the repeat count shadow register, the auto reload shadow register and the prescaler buffer will be updated. The update event can be disabled by configuring the UD bit of the TMRx_CTRL1 register.

Figure 23 Timing Diagram of Count-down Mode when Division Factor is 1 or 2

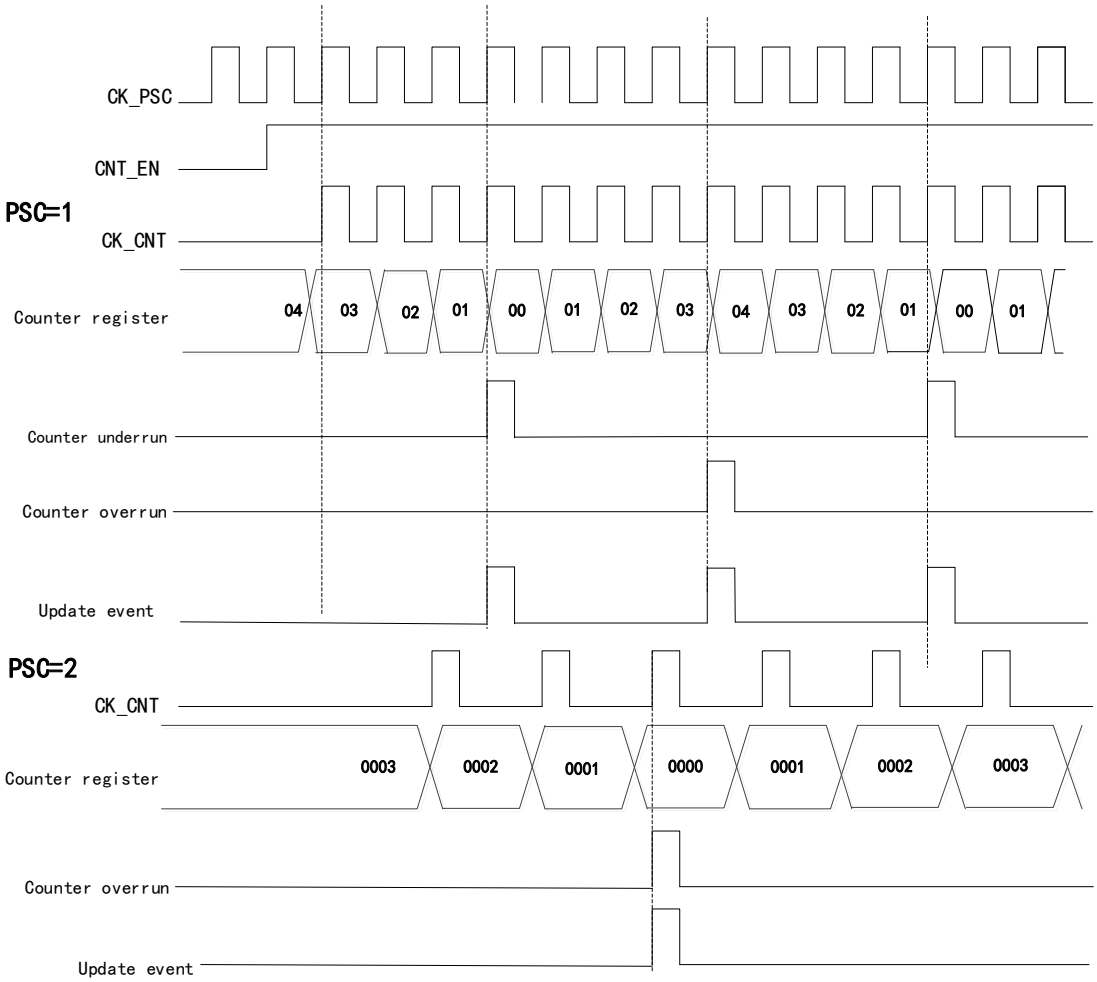


Center-aligned mode

Set to center-aligned mode by configuring CAMSEL bit of control register (TMRx_CTRL1).

When the counter is in center-aligned mode, the counter counts up from 0 and when it reaches the value of auto reload (TMRx_AUTORLD), it counts down to 0 from the value of the auto reload (TMRx_AUTORLD), which will repeat; in counting up, when the counter value is (AUTORLD-1), a counter overrun event will be generated; in counting down, when the counter value is 1, a counter underrun event will be generated.

Figure 24 Timing Diagram of Center-aligned Mode when Division Factor is 1 or 2



Repeat counter REPCNT

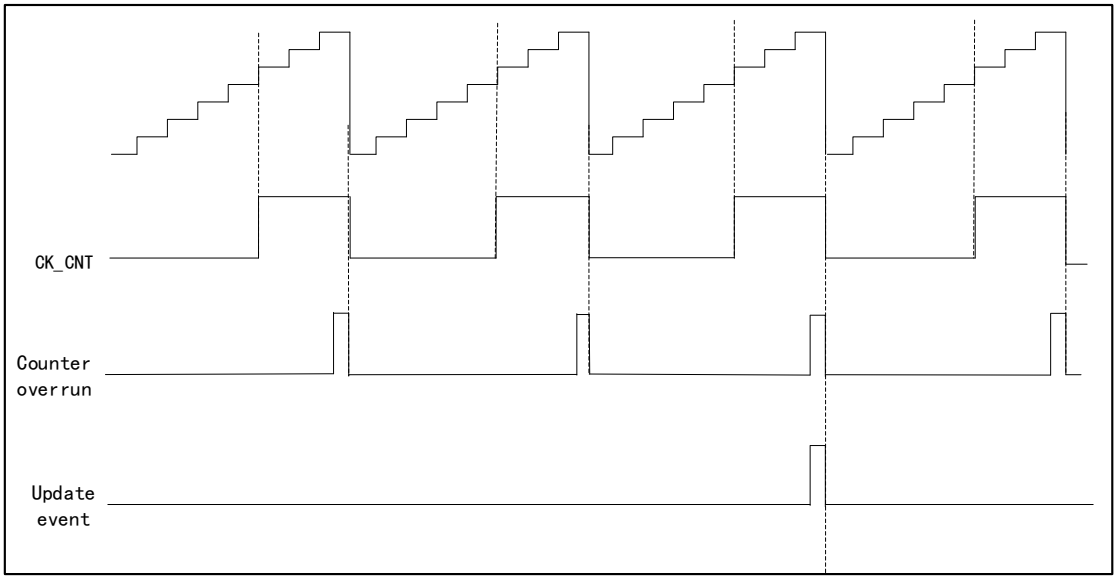
There is no repeat counter REPCNT in the general-purpose timer, which means that when an overrun event or underrun event occurs in the general-purpose timer, an update event will be generated directly; while in the advanced timer, because of the existence of the repeat counter, when an overrun/unerrrun event occurs to the advanced timer, the update event will be generated only when the value of the repeat counter is 0.

For example, if the advanced timer needs to generate an update event when an overrun/underrun event occurs, the value of the repeat counter should be set to 0.

If the repeat counter function is used in the count-up mode, every time the counter counts up to AUTORLD, an overrun event will occur. At this time, the value of the repeat counter will decrease by 1, and an update event will be generated when the value of the repeat counter is 0.

That is, when N+1 (N is the value of repeat counter) overrun/underrun events occur, an update event will be generated.

Figure 25 Timing Diagram of Count-up Mode when Setting REPCNT=2



Prescaler PSC

The prescaler is 16 bits and programmable, and it can divide the clock frequency of the counter to any value within 1~65536 (controlled by TMRx_PSC register), and after frequency division, the clock will drive the counter CNT to count. The prescaler has a buffer, which can be changed during running.

14.4.3 Input capture

Input capture channel

The advanced timer has four independent capture/compare channels, each of which is surrounded by a capture/compare register.

In the input capture, the measured signal will enter from the external pin TI1/2/3/4 of the timer, first pass through the edge detector and input filter, and then enter the capture channels. Each capture channel has a corresponding capture register. When the capture occurs, the value of the counter CNT will be latched in the capture register CCx. Before entering the capture register, the signal will pass through the prescaler to set how many events to capture at a time.

Input capture application

Input capture is used to capture external events, and can give the time flag to indicate the occurrence time of the event and measure the pulse jump edge events (measure the frequency or pulse width), for example, if the selected edge appears on the input pin, the TMRx_CCx register will capture the current value of the counter and the CCxIFLG bit of the status register TMRx_STS will be set to 1; if CCxIEN=1, an interrupt will be generated.

In capture mode, the timing, frequency, cycle and duty cycle of a waveform can be measured. In the input capture mode, the edge selection is set to rising edge detection. When the rising edge appears on the capture channel, the first capture occurs, at this time, the value of the counter CNT will be latched in the capture register CCx; at the same time, it will enter the capture interrupt, a capture will be recorded in the interrupt service program and the value will be recorded. When the next rising edge is detected, the second capture occurs, the value of counter CNT will be latched in capture register CCx again, at this time, it will enter the capture interrupt again; read the value of capture register and the cycle of this pulse signal will be obtained by capture.

14.4.4 Output compare

There are eight modes of output compare: freeze, channel x is valid level when matching, channel x is invalid level when matching, flip, force is invalid, force is valid, PWM mode 1 and PWM mode 2, which are configured by OCxMOD bit in TMRx_CCMx register and can control the waveform of output signal in output compare mode.

Output compare application

In the output compare mode, the position, polarity, frequency and time of the pulse generated by the timer can be controlled.

When the value of the counter is equal to that of the capture/compare register, the channel output can be set as high level, low level or flip by configuring the OCxMOD bit in TMRx_CCMx register and the CCxPOL bit in the output polarity TMRx_CCEN register.

When CCxIFLG=1 in TMRx_STS register, if CCxIEN=1 in TMRx_DIEN register, an interrupt will be generated; if CCDSEL=1 in TMRx_CTRL2 register, a DMA request will be generated.

14.4.5 PWM output mode

PWM mode is pulse signal that can be adjusted by external output of the timer. The pulse width of the signal is determined by the value of the compare register CCx, and the cycle is determined by the value of the auto reload AUTORLD.

PWM output mode contains PWM mode 1 and PWM mode 2; PWM mode 1 and PWM mode 2 are divided into count-up, count-down and center alignment counting; in PWM mode 1, if the value of the counter CNT is less than the value of the compare register CCx, the output level will be valid; otherwise, it will be invalid.

Set the timing diagram of PWM mode 1 when CCx=5, AUTORLD=7

Figure 26 Timing Diagram of PWM1 Count-up Mode

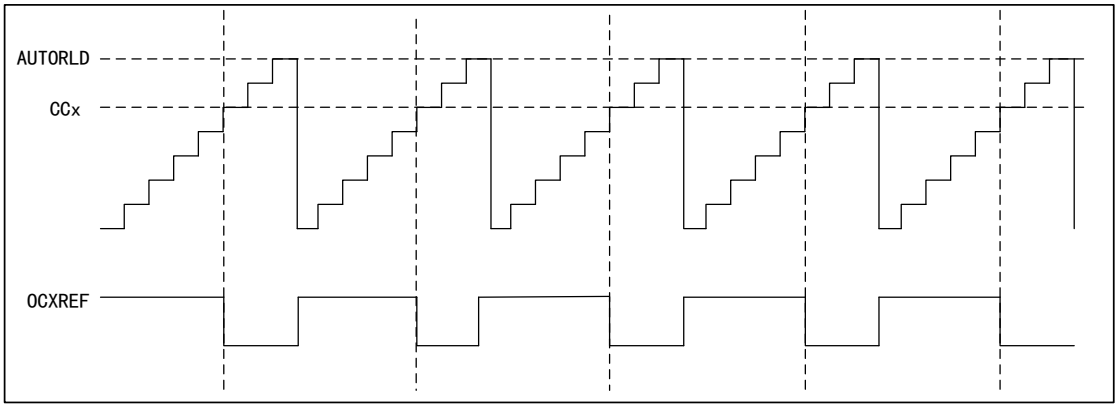


Figure 27 Timing Diagram of PWM1 Count-down Mode

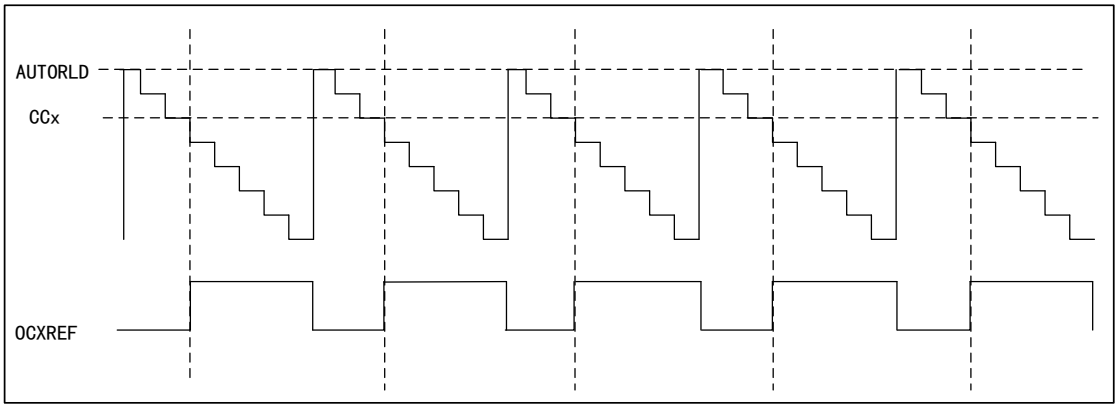
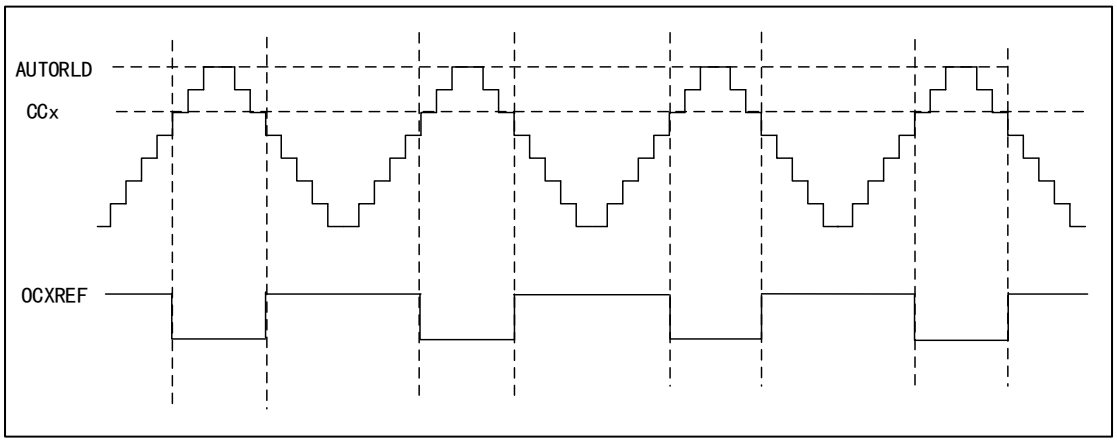


Figure 28 Timing Diagram of PWM1 Center-aligned Mode



In PWM mode 2, if the value of the counter CNT is less than that of the compare register CCx, the output level will be invalid; otherwise, it will be valid.

Set the timing diagram of PWM mode 2 when CCx=5, AUTORLD=7

Figure 29 Timing Diagram of PWM2 Count-up Mode

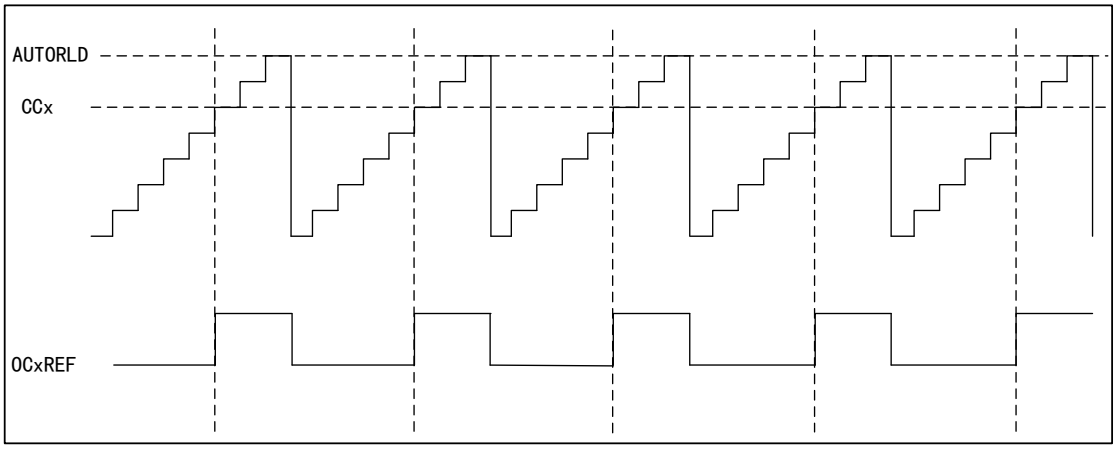


Figure 30 Timing Diagram of PWM2 Count-down Mode

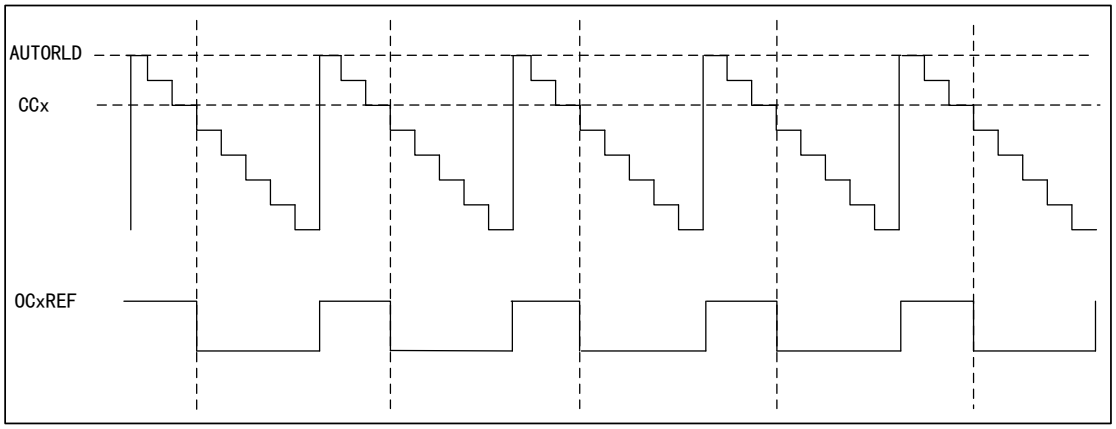
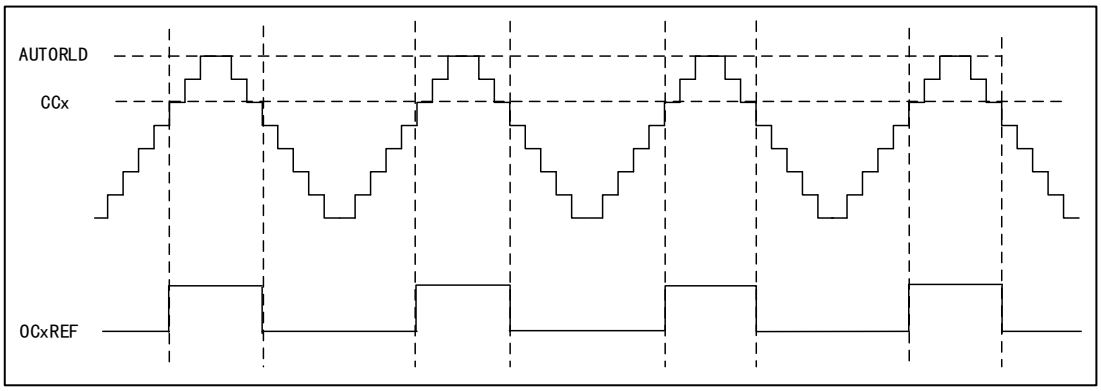


Figure 31 Timing Diagram of PWM2 Center-aligned Mode



14.4.6 PWM input mode

PWM input mode is a particular case of input capture.

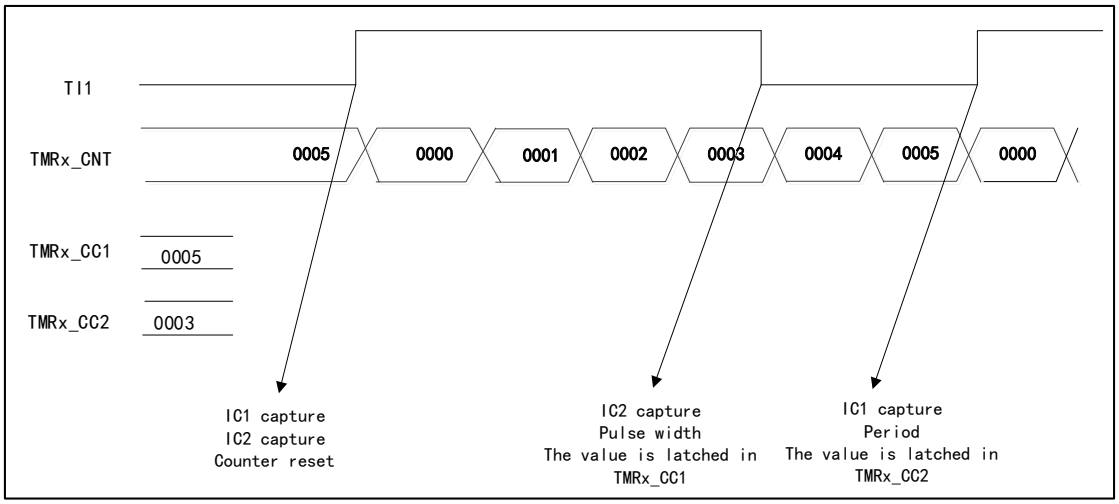
In PWM input mode, as only TI1FP1 and TI1FP2 are connected to the slave mode controller, input can be performed only through the channels TMRx_CH1

and TMRx_CH2, which need to occupy the capture registers of CH1 and CH2.

In the PWM input mode, the PWM signal enters from TMRx_CH1, and the signal will be divided into two channels, one can measure the cycle and the other can measure the duty cycle. In the configuration, it is only required to set the polarity of one channel, and the other will be automatically configured with the opposite polarity.

In this mode, the slave mode controller should be configured as the reset mode (SMFSEL bit of TMRx_SMCTRL register)

Figure 32 Timing Diagram in PWM Input Mode



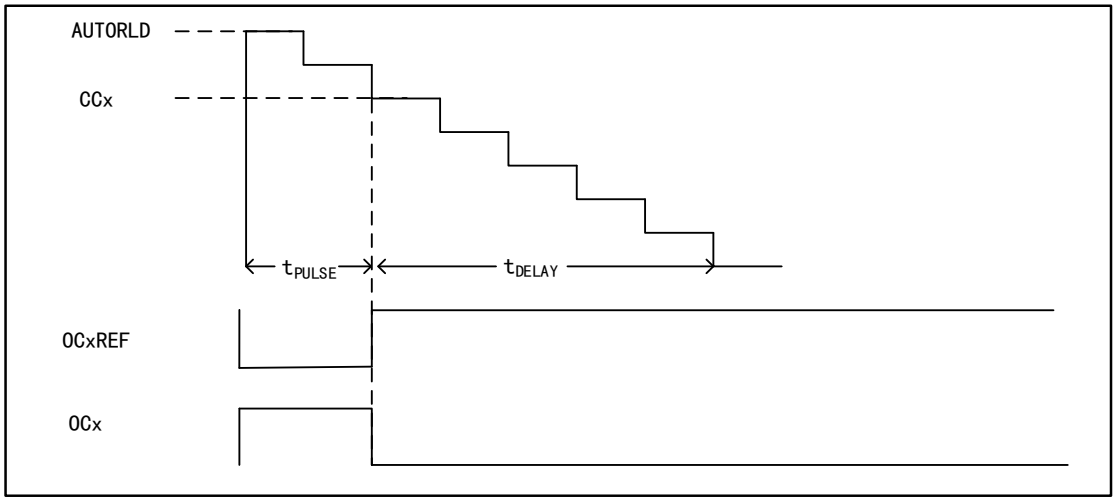
14.4.7 Single-pulse mode

The single-pulse mode is a special case of timer compare output, and is also a special case of PWM output mode.

Set SPMEN bit of TMRx_CTRL1 register, and select the single-pulse mode. After the counter is started, a certain number of pulses will be output before the update event occurs. When an update event occurs, the counter will stop counting, and the subsequent PWM waveform output will no longer be changed.

After a certain controllable delay, a pulse with controllable pulse width is generated in single-pulse mode through the program. The delay time is defined by the value of TMRx_CCx register; in the count-up mode, the delay time is CCx and the pulse width is AUTORLD-CCx; in the count-down mode, the delay time is AUTORLD-CCx and the pulse width is CCx.

Figure 33 Timing Diagram of Single-pulse Mode



14.4.8 Impact of the register on output waveform

The following registers will affect the level of the timer output waveform. For details, please refer to "Register Functional Description".

(1) CcxEN and CCxNEN bits in TMRx_CCEN register

- CCxNEN=0 and CCxEN=0: The output is disabled (output disabled, invalid)
- CCxNEN=1 and CCxEN=1: The output is enabled (output enabled, normal output)

(2) MOEN bit in TMRx_BDT register

- MOEN=0: Idle mode
- MOEN=1: Run mode

(3) OCxOIS and OCxNOIS bits in TMRx_CTRL2 register

- OCxOIS=0 and OCxNOIS=0: When idle (MOEN=0), the output level after the dead zone is 0
- OCxOIS=1 and OCxNOIS=1: When idle (MOEN=0), the output level after the dead zone is 1

(4) RMOS bit in TMRx_BDT register

- Application environment of RMOS: In run mode of corresponding complementary channel and timer (MOEN=1), the timer is not working (CCxEN=0, CCxNEN=0) or is working (CCxEN=1, CCxNEN=1)

(5) IMOS bit in TMRx_BDT register

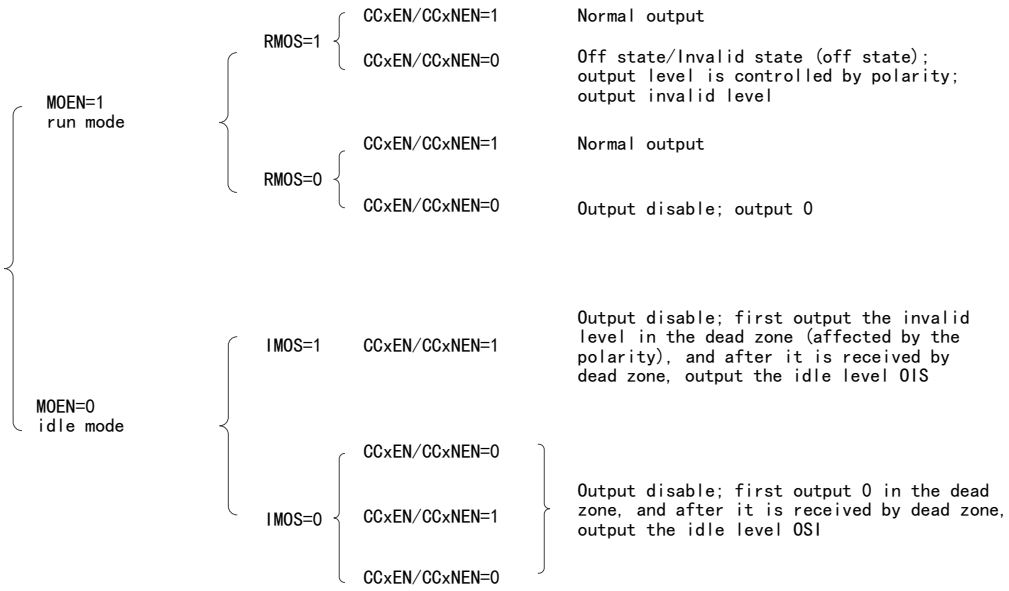
- Application environment of IMOS: In idle mode of corresponding complementary channel and timer (MOEN=0), the timer is not working (CCxEN=0, CCxNEN=0) or is working (CCxEN=1, CCxNEN=1)

(6) CCxPOL and CCxNPOL bits of TMRx_CCEN register

- CCxPOL=0 and CCxNPOL=0: Output polarity, valid at high level
- CCxPOL=1 and CCxNPOL=1: Output polarity, valid at low level

The following figure lists the register structural relationships that affect the output waveform

Figure 34 Register Structural Relationship Affecting Output Waveform



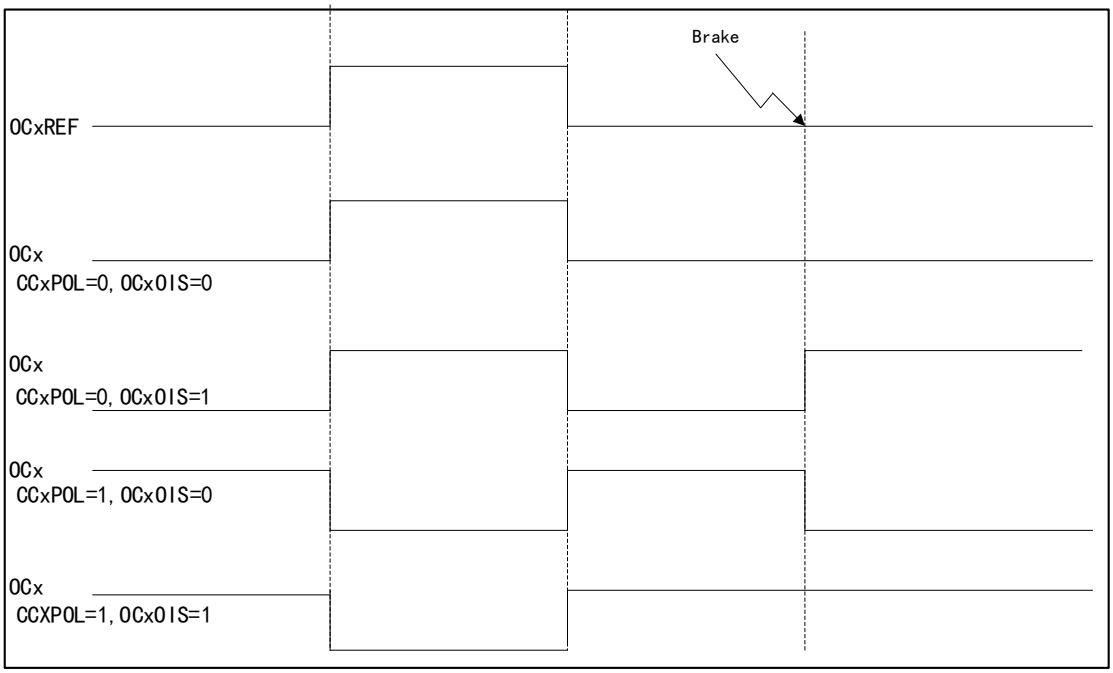
14.4.9 Braking function

The signal source of braking is clock fault event and external input interface.

Besides, the BRKEN bit in TMRx_BDT register can enable the braking function, and the BRKPOL bit can configure the polarity of braking input signal.

When a braking event occurs, the output pulse signal level can be modified according to the state of the relevant control bit.

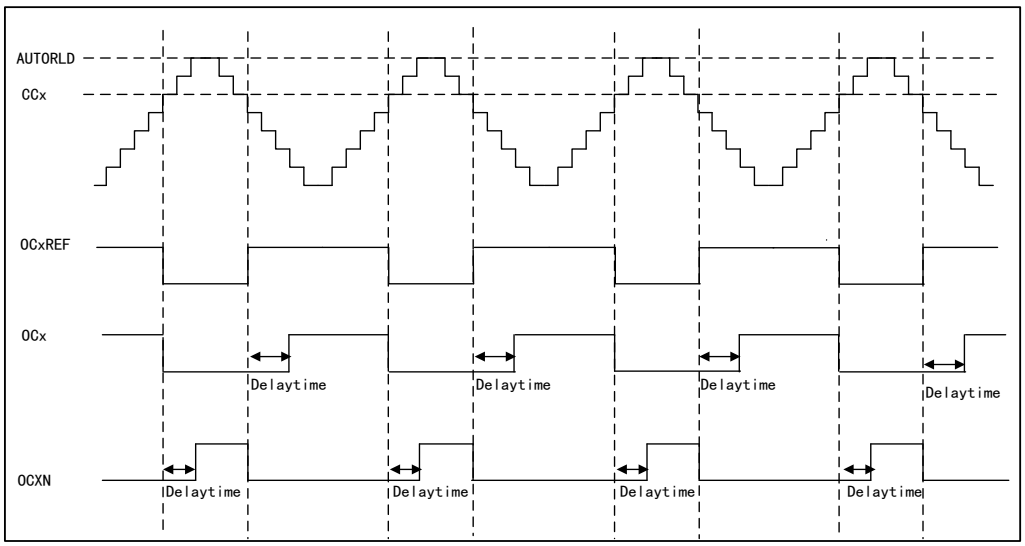
Figure 35 Braking Event Timing Diagram



14.4.10 Complementary output and dead zone insertion

Complementary output is particular output of advanced timer, and the advanced timer has three groups of complementary output channels. The insertion dead time is used to generate complementary output signals to ensure that the two-way complementary signals of channels will not be valid at the same time. The dead time is set according to the output device connected to the timer and its characteristics. The duration of the dead zone can be controlled by configuring DTS bit of TMRx_BDT register.

Figure 36 Complementary Output with Dead Zone Insertion



14.4.11 Forced output mode

In the forced output mode, the comparison result is ignored, and the corresponding level is directly output according to the configuration instruction.

- CCxSEL=00 for TMRx_CCMx register, set CCx channel as output
- OCxMOD=100/101 for TMRx_CCMx register, set to force OCxREF signal to invalid/valid

In this mode, the corresponding interrupt and DMA request will still be generated.

14.4.12 Encoder interface mode

The encoder interface mode is equivalent to an external clock with direction selection. In the encoder interface mode, the content of the timer can always indicate the position of the encoder.

The method of selecting encoder interface is as follows:

- By setting SMFSEL bit of TMRx_SMCTRL register, set the counter to count on the edge of TI1 channel /TI2 channel, or count on the edge of TI1 and TI2 at the same time.
- Select the polarity of TI1 and TI2 by setting the CC1POL and CC2POL bits of TMRx_CCEN register.
- Select to filter or not by setting the IC1F and IC2F bits of TMRx_CCM1 register.
- The two input TI1 and TI2 can be used as the interface of incremental encoder. The counter is driven by the effective jump of the signals TI1FP1 and TI2FP2 after filtering and edge selection in TI1 and TI2.
- The count pulse and direction signal are generated according to the input signals of TI1 and TI2
- The counter will count up/down according to the jumping sequence of the input signal.
- Set CNTDIR of control register TMRx_CTRL1 to be read-only (CNTDIR will be re-calculated due to jumping of any input end).

The change mechanism of counter count direction is shown in the figure below

Table 48 Relationship between Count Direction and Encoder

Effective edge		**Count only in TI1**		**Count only in TI2**		**Count in both TI1 and TI2**	
Level of relative signal		High	Low	High	Low	High	Low
TI1FP1	Rising edge	—		Count down	Count up	Count down	Count up
	Falling edge			Count up	Count down	Count up	Count down

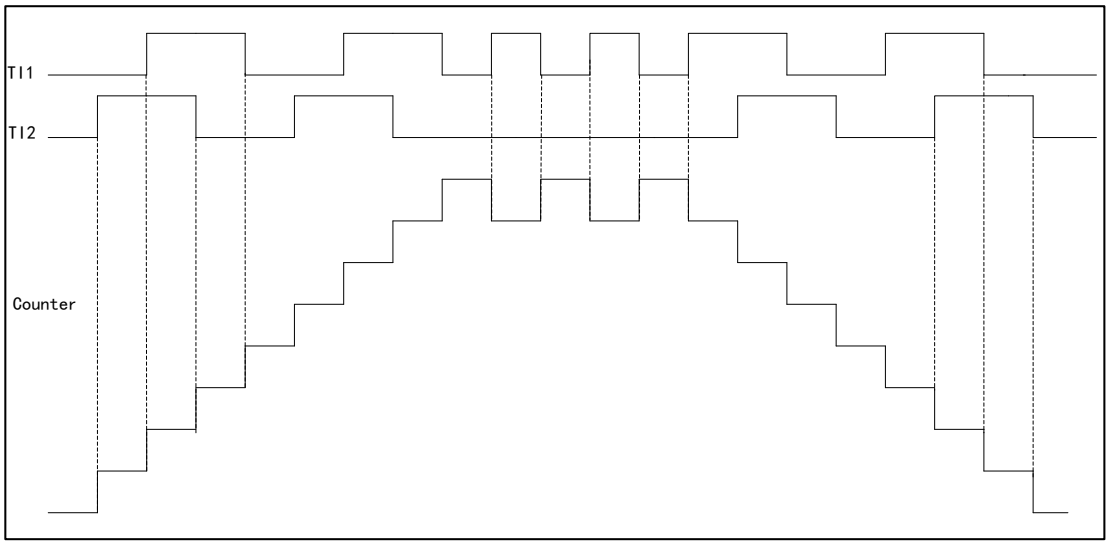
Effective edge		Count only in TI1		Count only in TI2	Count in both TI1 and TI2	
TI2FP2	Rising edge	Count up	Count down	—	Count up	Count down
	Falling edge	Count down	Count up		Count down	Count up

The external incremental encoder can be directly connected with MCU, not needing external interface logic, so the comparator is used to convert the differential output of the encoder to digital signal to increase the immunity to noise interference.

Among the following examples:

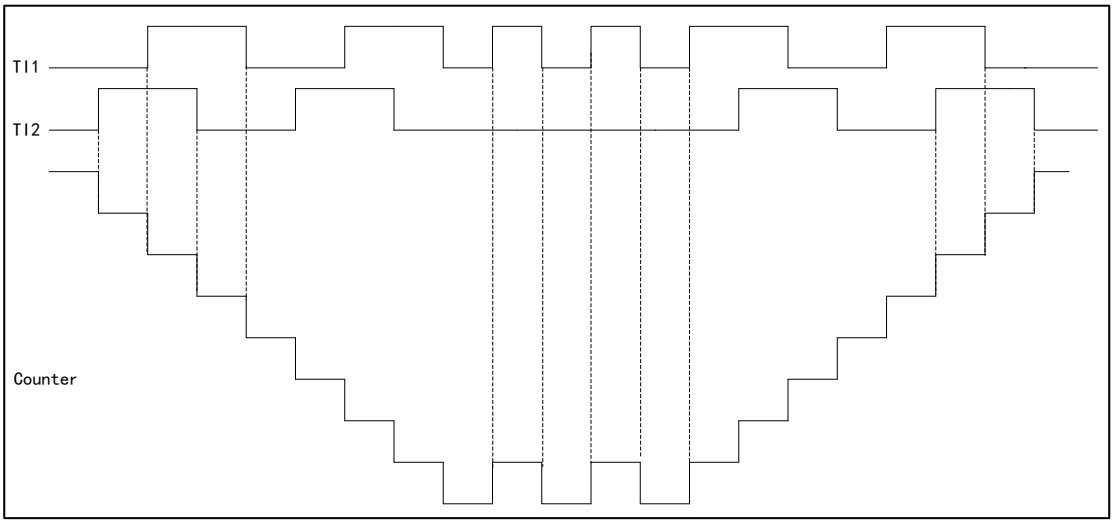
- IC1FP1 is mapped to TI1
- IC2FP2 is mapped to TI2
- Neither IC1FP1 nor IC2FP2 is phase-inverting
- The input signal is valid at the rising edge and falling edge
- Enable the counter

Figure 37 Counter Operation Example in Encoder Mode



For example, when TI1 is at low level, and TI2 is in rising edge state, the counter will count up.

Figure 38 Example of Encoder Interface Mode of IC1FP1 Reversed Phase



For example, when TI1 is at low level, and the rising edge of TI2 jumps, the counter will count down.

14.4.13 Slave Mode

TMRx timer can synchronize external trigger

- Reset mode
- Gated mode
- Trigger mode

SMFSEL bit in TMRx_SMCTRL register can be set to select the mode.

SMFSEL=100 set the reset mode, SMFSEL=101 set the gated mode, and SMFSEL=110 set the trigger mode.

In the reset mode, when a trigger input event occurs, the counter and prescaler will be initialized, and the rising edge of the selected trigger input (TRGI) will reinitialize the counter and generate a signal to update the register.

In the gated mode, the enable of the counter depends on the high level of the selected input end. When the trigger input is high, the clock of the counter will be enabled. Once the trigger input becomes low, the counter will stop (but not be reset). The start and stop of the counter are controlled.

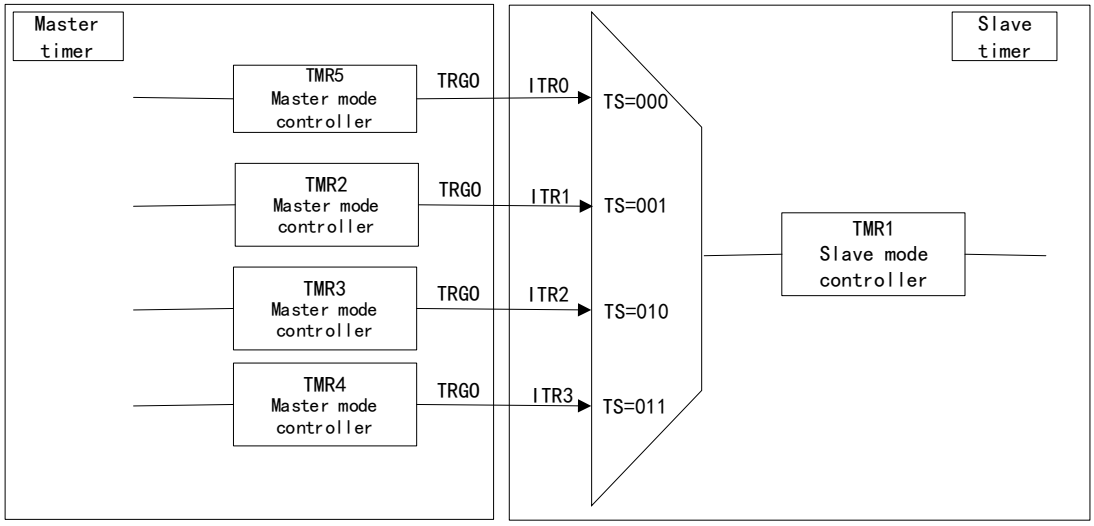
In the trigger mode, the enable of the counter depends on the event on the selected input, the counter will be enabled at the rising edge of the trigger input (but not be reset), and only the start of the counter is controlled.

14.4.14 Timer interconnection

Each timer of TMRx can be connected with each other to realize synchronization or cascading between timers. It is required to configure one timer in master mode and the other timer in slave mode.

When the timer is in master mode, it can reset, start, stop and provide clock source for the counter of the slave mode timer.

Figure 39 Timer 1 Master/Slave Mode Example



When the timers are interconnected:

- A timer can be used as the prescaler of other register
- Start the other register by the enable signal of a timer
- Start the other register by the update event of a timer
- Select the other register by the enable of a timer
- Two timers can be synchronized by an external trigger

14.4.15 Interrupt and DMA request

The timer can generate an interrupt when an event occurs during operation

- Update event (counter overrun/underrun, counter initialization)
- Trigger event (counter start, stop, internal/external trigger)
- Capture/Compare event
- Braking signal input event

Some internal interrupt events can generate DMA requests, and special interfaces can enable or disable trigger DMA requests.

14.4.16 Debug mode

The TMR1 can be configured in debug mode and choose to stop or continue to work. It depends on the DBG_TIM1_STOP bit of DBGMCU_APBF register in DBGMCU module.

14.4.17 Clear OCxREF signal when an external event occurs

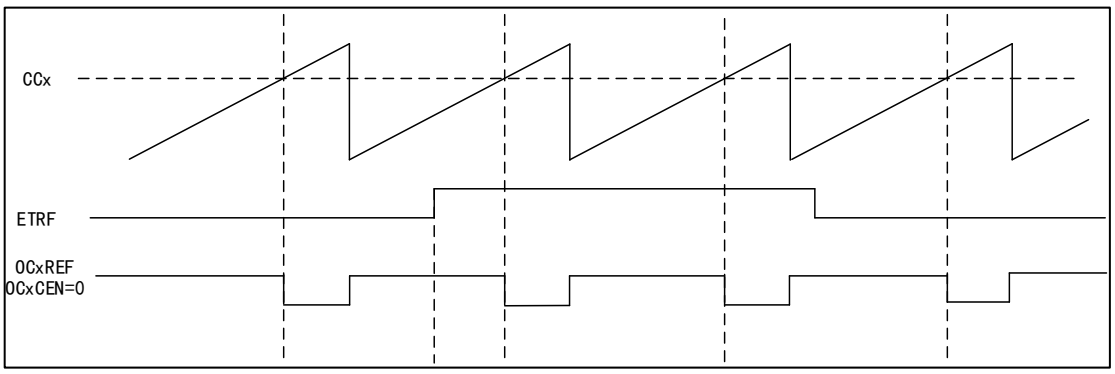
This function is used for output compare and PWM mode.

In one channel, the high level of ETRF input port will reduce the signal of OCxREF to low level, and the OCxCEN bit in capture/compare register TMRx_CCMx is set to 1, and OCxREF signal will remain low until the next

update event occurs.

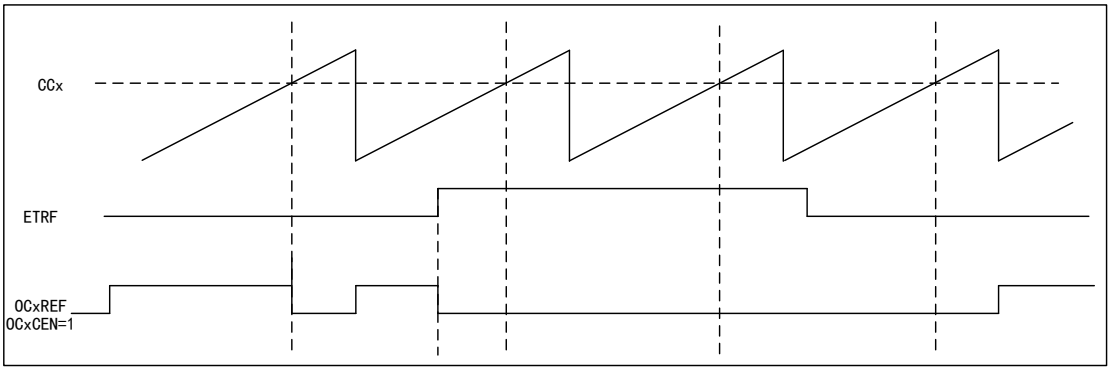
Set TMRx to PWM mode, disable the external trigger prescaler, and disable the external trigger mode 2; when ETRF input is high, set OCxCEN=0, and the output OCxREF signal is shown in the figure below.

Figure 40 OCxREF Timing Diagram



Set TMRx to PWM mode, disable the external trigger prescaler, and disable the external trigger mode 2; when ETRF input is high, set OCxCEN=1, and the output OCxREF signal is shown in the figure below.

Figure 41 OCxREF Timing Diagram



14.5 Register address mapping

In the following table, all registers of the advanced timer are mapped to a 16-bit addressable (addressing) space.

Table 49 Advanced Timer Register Address Mapping

Register name	Description	Offset Address
TMRx_CTRL1	Control register 1	0x00
TMRx_CTRL2	Control register 2	0x04
TMRx_SMCTRL	Slave mode control register	0x08
TMRx_DIEN	DMA/Interrupt enable register	0x0C
TMRx_STS	Status register	0x10

Register name	Description	Offset Address
TMRx_CEG	Control event generation register	0x14
TMRx_CCM1	Capture/Compare mode register 1	0x18
TMRx_CCM2	Capture/Compare mode register 2	0x1C
TMRx_CCEN	Capture/Compare enable register	0x20
TMRx_CNT	Counter register	0x24
TMRx_PSC	Prescaler register	0x28
TMRx_AUTORLD	Auto reload register	0x2C
TMRx_REPCNT	Repeat count register	0x30
TMRx_CC1	Channel 1 capture/compare register	0x34
TMRx_CC2	Channel 2 capture/compare register	0x38
TMRx_CC3	Channel 3 capture/compare register	0x3C
TMRx_CC4	Channel 4 capture/compare register	0x40
TMRx_BDT	Brake and dead zone register	0x44
TMRx_DCTRL	DMA control register	0x48
TMRx_DMADDR	DMA address register of continuous mode	0x4C

14.6 Register functional description

14.6.1 Control register 1 (TMRx_CTRL1) (x=1)

Offset address: 0x00

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:10	Reserved		
9:8	CLKDIV	R/W	Clock Division For the configuration of dead zone and digital filter, CK_INT provides the clock, and the dead time and the clock of the digital filter can be adjusted by this bit. 00: $t_{DTS}=t_{CK\_INT}$ 01: $t_{DTS}=2\times t_{CK\_INT}$ 10: $t_{DTS}=4\times t_{CK\_INT}$ 11: Reserved
7	ARPEN	R/W	Auto-reload Preload Enable When the buffer is disabled, modification of TMRx_AUTORLD by program will immediately lead to modification of the values loaded to the counter; when the buffer is enabled, modification of TMRx_AUTORLD by program will lead to modification of the values loaded to the counter at the next update event. 0: Disable 1: Enable

Field	Name	R/W	Description
6:5	CAMSEL	R/W	Center Aligned Mode Select In the center-aligned mode, the counter counts up and down alternately; otherwise, it will only count up or down. Different center-aligned modes affect the timing of setting the output compare interrupt flag bit of the output channel to 1; when the counter is disabled (CNTEN=0), select the center-aligned mode. 00: Edge-aligned mode 01: Center-aligned mode 1 (the output compare interrupt flag bit of output channel is set to 1 when counting down) 10: Center-aligned mode 2 (the output compare interrupt flag bit of output channel is set to 1 when counting up) 11: Center-aligned mode 3 (the output compare interrupt flag bit of output channel is set to 1 when counting up/down)
4	CNTDIR	R/W	Counter Direction This bit is read-only when the counter is configured as center-aligned mode or encoder mode. 0: Count up 1: Count down
3	SPMEN	R/W	Single Pulse Mode Enable When an update event is generated, the output level of the channel can be changed; in this mode, the CNTEN bit will be cleared, the counter will stop, and the subsequent output level of the channel will no long be changed. 0: Disable 1: Enable
2	URSSEL	R/W	Update Request Source Select If interrupt or DMA is enabled, the update event can generate update interrupt or DMA request. Different update request sources can be selected by this bit. 0: The counter overruns or underruns Set UEG bit; Update generated by slave mode controller. 1: The counter overruns or underruns
1	UD	R/W	Update Disable Update event can cause AUTORLD, PSC and CCx to generate the value of update setting. 0: Enable update event (UEV) An update event can occur in any of the following situations: The counter overruns/underruns; Set UEG bit; Update generated by slave mode controller. 1: Disable update event

Field	Name	R/W	Description
0	CNTEN	R/W	Counter Enable 0: Disable 1: Enable When the timer is configured as external clock, gated mode and encoder mode, it is required to write 1 to the bit by software to start regular work; when it is configured as the trigger mode, it can write 1 by hardware.

14.6.2 Control register 2 (TMRx_CTRL2) (x=1)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:15	Reserved		
14	OC4OIS	R/W	Configure OC4 output idle state. Refer to OC1OIS bit
13	OC3NOIS	R/W	Configure OC3N output idle state. Refer to OC1NOIS bit
12	OC3OIS	R/W	Configure OC3 output idle state. Refer to OC1OIS bit
11	OC2NOIS	R/W	Configure OC2N output idle state. Refer to OC1NOIS bit
10	OC2OIS	R/W	Configure OC2 output idle state. Refer to OC1OIS bit
9	OC1NOIS	R/W	OC1N Output Idle State Configure Only the level state after the dead time of OC1 is affected when MOEN=0 and OC1N is realized. 0:OC1N=0 1:OC1N=1 Note: When PLOCKCFG bit in TMRx_BDT register is at the Level 1, 2 or 3, this bit cannot be modified.
8	OC1OIS	R/W	OC1 Output Idel State Configure Only the level state after the dead time of OC1 is affected when MOEN=0 and OC1N is realized. 0:OC1=0 1:OC1=1 Note: When LOCKCFG bit in TMRx_BDT register is at the Level 1, 2 or 3, this bit cannot be modified.
7	TI1SEL	R/W	Timer Input 1 Selection 0: TMRx_CH1 pin is connected to TI1 input 1: TMRx_CH1, TMRx_CH2 and TMRx_CH3 pins are connected to TI1 input after exclusive

Field	Name	R/W	Description
6:4	MMSEL	R/W	Master Mode Signal Select The signals of timers working in master mode can be used for TRGO, so as to affect the work of timers in slave mode and cascaded with the master timer, and the specific impact is related to the configuration of slave mode timer. 000: Reset; the reset signal of master mode timer is used for TRGO 001: Enable; the counter enable signal of master mode timer is used for TRGO 010: Update; the update event of master mode timer is used for TRGO 011: Compare pulses; when the master mode timer captures/compares successfully (CCxIFLG=1), a pulse signal is output for TRGO 100: Compare mode 1; OC1REF is used to trigger TRGO 101: Compare mode 2; OC2REF is used to trigger TRGO 110: Compare mode 3; OC3REF is used to trigger TRGO 111: Compare mode 4; OC4REF is used to trigger TRGO
3	CCDSEL	R/W	Capture/Compare DMA Select 0: Transmit DMA request of CCx when CCx event occurs 1: Transmit DMA request of CCx when an update event occurs
2	CCUSEL	R/W	Capture/compare Control Update Select It works only when the capture/compare preload is enabled (CCPEN=1), and it works only for complementary output channel. 0: It can only be updated by setting COMG bit 1: It can be updated by setting COMG bit or rising edge on TRGI
1	Reserved		
0	CCPEN	R/W	Capture/Compare Preloaded Enable This bit affects the change of CcxEN, CCxNEN and OCxMOD values. When preloading is disabled, the program modification will immediately affect the timer setting; when preloading is enabled, it is only updated after COMG is set, so as to affect the setting of the timer; this bit only works on channels with complementary output. 0: Disable 1: Enable

14.6.3 Slave mode control register (TMRx_SMCTRL) (x=1)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15	ETPOL	R/W	External Trigger Polarity Configure This bit decides whether the external trigger ETR is phase-inverting. 0: The external trigger ETR is not phase-inverting, and the high level or rising edge is valid 1: The external trigger ETR is phase-inverting, and the low level or falling edge is valid
14	ECEN	R/W	External Clock Enable Mode2

Field	Name	R/W	Description
			0: Disable 1: Enable Setting ECEN bit has the same function as selecting external clock mode 1 to connect TRGI to ETRF; slave mode (reset, gating, trigger) can be used at the same time with external clock mode 2, but TRGI cannot be connected to ETRF in such case; when external clock mode 1 and external clock mode 2 are enabled at the same time, the input of external clock is ETRF.
13:12	ETPCFG	R/W	External Trigger Prescaler Configure The ETR (external trigger input) signal becomes ETRP after frequency division. The signal frequency of ETRP is at most 1/4 of TMRxCLK frequency; when ETR frequency is too high, the ETRP frequency must be reduced through frequency division. 00: Disable the prescaler; 01: ETR signal 2 divided frequency 10: ETR signal 4 divided frequency 11: ETR signal 8 divided frequency
11:8	ETFCFG	R/W	External Trigger Filter Configure 0000: Disable filter, sampled by f_{DTS} 0001:DIV=1，N=2 0010:DIV=1，N=4 0011:DIV=1，N=8 0100:DIV=2，N=6 0101:DIV=2，N=8 0110:DIV=4，N=6 0111:DIV=4，N=8 1000:DIV=8，N=6 1001:DIV=8，N=8 1010:DIV=16，N=5 1011:DIV=16，N=6 1100:DIV=16，N=8 1101:DIV=32，N=5 1110:DIV=32，N=6 1111:DIV=32，N=8 Sampling frequency=timer clock frequency/DIV; the filter length=N, and a jump is generated by every N events.
7	MSMEN	R/W	Master/slave Mode Enable 0: Invalid 1: Enable the master/slave mode
6:4	TRGSEL	R/W	Trigger Input Signal Select In order to avoid generating false edge detection when changing the value of this bit, it must be changed when SMFSEL=0. 000: Internal trigger ITR0 001: Internal trigger ITR1 010: Internal trigger ITR2 011: Internal trigger ITR3 100: Channel 1 input edge detector TIF_ED 101: Channel 1 post-filtering timer input TI1FP1

Field	Name	R/W	Description
			110: Channel 2 post-filtering timer input TI2FP2 111: External trigger input (ETRF)
3	Reserved		
2:0	SMFSEL	R/W	Slave Mode Function Select 000: Disable the slave mode, the timer can be used as master mode timer to affect the work of slave mode timer; if CTRL1_CNTEN=1, the prescaler is directly driven by the internal clock. 001: Encoder mode 1; according to the level of TI1FP1, the counter counts at the edge of TI2FP2. 010: Encoder mode 2; according to the level of TI2FP2, the counter counts at the edge of TI1FP1. 011: Encoder mode 3; according to the input level of the other signal, the counter counts at the edge of TI1FP1 and TI2FP2. 100: Reset mode; the slave mode timer resets the counter after receiving the rising edge signal of TRGI and generates the signal to update the register. 101: Gated mode; when the slave mode timer receives the TRGI high level signal, the counter will start to work; when it receives TRGI low level signal, the counter will stop working; when it receives TRGI high level signal again, the timer will continue to work; the counter is not reset during the whole period. 110: Trigger mode; the slave mode timer drives the counter to work after receiving the rising edge signal of TRGI. 111: External clock mode 1; select the rising edge signal of TRGI as the clock source to drive the counter to work.

14.6.4 DMA/Interrupt enable register (TMRx_DIEN) (x=1)

Offset address: 0x0C

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:15	Reserved		
14	TRGDEN	R/W	Trigger DMA Request Enable 0: Disable 1: Enable
13	COMDEN	R/W	COM DMA Request Enable 0: Disable 1: Enable
12	CC4DEN	R/W	Capture/Compare Channel4 DMA Request Enable 0: Disable 1: Enable
11	CC3DEN	R/W	Capture/Compare Channel3 DMA Request Enable 0: Disable 1: Enable

Field	Name	R/W	Description
10	CC2DEN	R/W	Capture/Compare Channel2 DMA Request Enable 0: Disable 1: Enable
9	CC1DEN	R/W	Capture/Compare Channel1 DMA Request Enable 0: Disable 1: Enable
8	UDIEN	R/W	Update DMA Request Enable 0: Disable 1: Enable
7	BRKIEN	R/W	Break Interrupt Enable 0: Disable 1: Enable
6	TRGIEN	R/W	Trigger Interrupt Enable 0: Disable 1: Enable
5	COMIEN	R/W	COM Interrupt Enable 0: Disable 1: Enable
4	CC4IEN	R/W	Capture/Compare Channel4 Interrupt Enable 0: Disable 1: Enable
3	CC3IEN	R/W	Capture/Compare Channel3 Interrupt Enable 0: Disable 1: Enable
2	CC2IEN	R/W	Capture/Compare Channel2 Interrupt Enable 0: Disable 1: Enable
1	CC1IEN	R/W	Capture/Compare Channel1 Interrupt Enable 0: Disable 1: Enable
0	UIEN	R/W	Update Interrupt Enable 0: Disable 1: Enable

14.6.5 Status register (TMRx_STS) (x=1)

Offset address: 0x10

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:13	Reserved		
12	CC4RCFLG	RC_W0	Captuer/Compare Channel4 Repetition Capture Flag Refer to STS_CC1RCFLG
11	CC3RCFLG	RC_W0	Captuer/compare Channel3 Repetition Capture Flag Refer to STS_CC1RCFLG

Field	Name	R/W	Description
10	CC2RCFLG	RC_W0	Captuer/Compare Channel2 Repetition Capture Flag Refer to STS_CC1RCFLG
9	CC1RCFLG	RC_W0	Captuer/Compare Channel1 Repetition Capture Flag 0: Repeated capture does not occur 1: Repeated capture occurs The value of the counter is captured to TMRx_CC1 register, and CC1IFLG=1 in TMRx_STS register; this bit can be set to 1 by hardware and cleared to 0 by software only when the channel is configured as input capture.
8	Reserved		
7	BRKIFLG	RC_W0	Brake Event Interrupt Generate Flag 0: No brake event occurs 1: Brake event occurs When brake input is valid, this bit is set to 1 by hardware; when brake input is invalid, this bit can be cleared to 0 by software.
6	TRGIFLG	RC_W0	Trigger Event Interrupt Generate Flag 0: No trigger event interrupt occurs 1: Trigger event interrupt occurs When a trigger event is generated, this bit is set to 1 by hardware and cleared to 0 by software.
5	COMIFLG	RC_W0	COM Event Interrupt Generate Flag 0: No COM event occurs 1: COM interrupt waits for response After COM event is generated, this bit is set to 1 by hardware and cleared to 0 by software.
4	CC4IFLG	RC_W0	Captuer/Compare Channel4 Interrupt Flag Refer to STS_CC1IFLG
3	CC3IFLG	RC_W0	Captuer/Compare Channel3 Interrupt Flag Refer to STS_CC1IFLG
2	CC2IFLG	RC_W0	Capture/Compare Channel2 Interrupt Flag Refer to STS_CC1IFLG
1	CC1IFLG	RC_W0	Captuer/Compare Channel1 Interrupt Flag When the capture/compare channel 1 is configured as output: 0: No matching occurs 1: The value of TMRx_CNT matches the value of TMRx_CC1 When the capture/compare channel 1 is configured as input: 0: No input capture occurs 1: Input capture occurs When a capture event occurs, it is set to 1 by hardware; it can be cleared to 0 by software or cleared to 0 when reading TMRx_CC1 register.

Field	Name	R/W	Description
0	UIFLG	RC_W0	Update Event Interrupt Generate Flag 0: No update event interrupt occurs 1: Update event interrupt occurred When the counter value is reloaded or reinitialized, an update event will be generated. The bit is set to 1 by hardware and cleared to 0 by software; update events are generated in the following situations: (1) UD=0 on TMRx_CTRL1 register, and when the value of the repeat counter overruns/underruns, an update event will be generated; (2) URSSEL=0 and UD=0 on TMRx_CTRL1 register, configure UEG=1 on TMRx_CEG register to generate an update event, and the counter needs to be initialized by software; (3) URSSEL=0 and UD=0 on TMRx_CTRL1 register, and an update event will be generated when the counter is initialized by trigger event.

14.6.6 Control event generation register (TMRx_CEG) (x=1)

Offset address: 0x14

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:8	Reserved		
7	BEG	W	Generate brake event (Brake Event Generate) 0: Invalid 1: Generate brake event This bit is set to 1 by software and cleared to 0 automatically by hardware.
6	TEG	W	Trigger Event Generate 0: Invalid 1: Generate trigger event This bit is set to 1 by software and cleared to 0 automatically by hardware.
5	COMG	W	Capture/Compare Control Update Event Generate 0: Invalid 1: Generate capture/Compare update event This bit is set to 1 by software and cleared to 0 automatically by hardware. Note: COMG bit is valid only in complementary output channel.
4	CC4EG	W	Capture/Compare Channel4 Event Generation Refer to CC1EG description
3	CC3EG	W	Capture/Compare Channel3 Event Generation Refer to CC1EG description
2	CC2EG	W	Capture/Compare Channel2 Event Generation Refer to CC1EG description
1	CC1EG	W	Capture/Compare Channel1 Event Generation 0: Invalid

Field	Name	R/W	Description
			1: Generate capture/compare event This bit is set to 1 by software and cleared to 0 automatically by hardware. If Channel 1 is in output mode: When CC1IFLG=1 in TMRx_STS register, if CC1IEN and CC1DEN bits in TMRx_DIEN are set, the corresponding interrupt and DMA request will be generated. If Channel 1 is in input mode: The value of the capture counter is stored in TMRx_CC1 register. Configure CC1IFLG=1 in TMRx_STS register, and if CC1IEN and CC1DEN bits in the TMRx_DIEN register are also set, the corresponding interrupt and DMA request will be generated; at this time, if CC1IFLG=1, it is required to configure CC1RCFLG=1.
0	UEG	W	Update Event Generate 0: Invalid 1: Initialize the counter and generate an update event This bit is set to 1 by software, and cleared to 0 by hardware. Note: When an update event is generated, the counter of the prescaler will be cleared to 0, but the prescaler factor remains unchanged. In the count-down mode, the counter will read the value of TMRx_AUTORLD; in center-aligned mode or count-up mode, the counter will be cleared to 0.

14.6.7 Capture/Compare mode register 1 (TMRx_CCM1) (x=1)

Offset address: 0x18

Reset value: 0x0000 0000

The timer can be configured as input (capture mode) or output (compare mode) by CCxSEL bit. The functions of other bits of the register are different in input and output modes, and the functions of the same bit are different in output mode and input mode. The OCX in the register describes the function of the channel in the output mode, and the Icx in the register describes the function of the channel in the input mode.

Output compare mode:

Field	Name	R/W	Description
31:16	Reserved		
15	OC2CEN	R/W	Output Compare Channel2 Clear Enable
14:12	OC2MOD	R/W	Output Compare Channel2 Mode
11	OC2PEN	R/W	Output Compare Channel2 Buffer Enable
10	OC2FEN	R/W	Output Compare Channel2 Preload Enable

Field	Name	R/W	Description
9:8	CC2SEL	R/W	Capture/Compare Channel2 Select This bit defines the input/output direction and selects the input pin. 00: CC2 channel is output 01: CC2 channel is input, and IC2 is mapped on TI2 10: CC2 channel is input, and IC2 is mapped on TI1 11: CC2 channel is input, and IC2 is mapped on TRC, and only works in internal trigger input Note: This bit can be written only when the channel is closed (TMRx_CCEN register CC2EN=0).
7	OC1CEN	R/W	Output Compare Channel1 Clear Enable 0: OC1REF is unaffected by ETRF input. 1: When high level of ETRF input is detected, OC1REF=0
6:4	OC1MOD	R/W	Output Compare Channel1 Mode Configure 000: Freeze The output compare has no effect on OC1REF 001: The output value is high when matching. When the value of counter CNT matches the value CCx of capture compare register, OC1REF will be forced to be at high level 010: The output value is low when matching. When the value of the counter matches the value of the capture/compare register, OC1REF will be forced to be low 011: Output flaps when matching. When the value of the counter matches the value of the capture compare register, flap the level of OC1REF 100: The output is forced to be low. Force OC1REF to be low 101: The output is forced to be high. Force OC1REF to be high 110: PWM mode 1 (set to high when the counter value<output compare value; otherwise, set to low) 111: PWM mode 2 (set to high when the counter value>output compare value; otherwise, set to low) Note: When the protection level is 3 and the channel is configured as output, this bit cannot be modified. In PWM modes 1 and 2, the OC1REF level changes when the comparison result changes or when the output compare mode changes from freeze mode to PWM mode.
3	OC1PEN	R/W	Output Compare Channel1 Preload Enable 0: Disable preloading function; write the value of TMRx_CC1 register through the program and it will work immediately. 1: Enable preloading function; write the value of TMRx_CC1 register through the program and it will work after an update event is generated. Note: When the protection level is 3 and the channel is configured as output, this bit cannot be modified. When the preload register is uncertain, PWM mode can be used only in single-pulse mode (SPMEN=1); otherwise, the following output compare result is uncertain.
2	OC1FEN	R/W	Output Compare Channel1 Fast Enable 0: Disable 1: Enable This bit is used to improve the response of the capture/compare output to the trigger input event.

Field	Name	R/W	Description
1:0	CC1SEL	R/W	Capture/Compare Channel 1 Selection This bit defines the input/output direction and selects the input pin. 00: CC1 channel is output 01: CC1 channel is input, and IC1 is mapped on TI1 10: CC1 channel is input, and IC1 is mapped on TI2 11: CC1 channel is input, and IC1 is mapped on TRC, and only works in internal trigger input Note: This bit can be written only when the channel is closed (TMRx_CCEN register CC1EN=0).

Input capture mode:

Field	Name	R/W	Description
31:16	Reserved		
15:12	IC2F	R/W	Input Capture Channel 2 Filter Configuration
11:10	IC2PSC	R/W	Input Capture Channel2 Perscaler Configuration
9:8	CC2SEL	R/W	Capture/Compare Channel 2 Select 00: CC2 channel is output 01: CC2 channel is input, and IC2 is mapped on TI2 10: CC2 channel is input, and IC2 is mapped on TI1 11: CC2 channel is input, and IC2 is mapped on TRC, and only works in internal trigger input Note: This bit can be written only when the channel is closed (TMRx_CCEN register CC2EN=0).
7:4	IC1F	R/W	Input Capture Channel 1 Filter Configuration 0000: Disable filter, sampled by f_{DTS} 0001：DIV=1，N=2 0010：DIV=1，N=4 0011：DIV=1，N=8 0100：DIV=2，N=6 0101：DIV=2，N=8 0110：DIV=4，N=6 0111：DIV=4，N=8 1000：DIV=8，N=6 1001：DIV=8，N=8 1010：DIV=16，N=5 1011：DIV=16，N=6 1100：DIV=16，N=8 1101：DIV=32，N=5 1110：DIV=32，N=6 1111：DIV=32，N=8 Sampling frequency=timer clock frequency/DIV; the filter length=N, indicating that a jump is generated by every N events.
3:2	IC1PSC	R/W	Input Capture Channel1 Perscaler Configure 00:PSC=1 01:PSC=2 10:PSC=4 11:PSC=8

Field	Name	R/W	Description
			PSC is prescaler factor; capture is triggered once by every PSC events.
1:0	CC1SEL	R/W	Capture/Compare Channel 1 Select 00: CC1 channel is output 01: CC1 channel is input, and IC1 is mapped on TI1 10: CC1 channel is input, and IC1 is mapped on TI2 11: CC1 channel is input, and IC1 is mapped on TRC, and only works in internal trigger input Note: This bit can be written only when the channel is closed (TMRx_CCEN bit CC1EN=0).

14.6.8 Capture/Compare mode register 2（TMRx_CCM2）(x=1)

Offset address: 0x1C

Reset value: 0x0000 0000

Refer to the description of the above CCM1 register.

Output compare mode:

Field	Name	R/W	Description
31:16	Reserved		
15	OC4CEN	R/W	Output Compare Channel4 Clear Enable
14:12	OC4MOD	R/W	Output Compare Channel4 Mode Configure
11	OC4PEN	R/W	Output Compare Channel4 Buffer Enable
10	OC4FEN	R/W	Output Compare Channel4 Preload Enable
9:8	CC4SEL	R/W	Capture/Compare Channel 4 Selection This bit defines the input/output direction and selects the input pin. 00: CC4 channel is output 01: CC4 channel is input, and IC4 is mapped on TI4 10: CC4 channel is input, and IC4 is mapped on TI3 11: CC4 channel is input, and IC4 is mapped on TRC, and only works in internal trigger input Note: This bit can be written only when the channel is closed (TMRx_CCEN register CC4EN=0).
7	OC3CEN	R/W	Output Compare Channel3 Clear Enable 0: OC3REF is unaffected by ETRF input. 1: When high level of ETRF input is detected, OC1REF=0
6:4	OC3MOD	R/W	Output Compare Channel3 Mode Configure)
3	OC3PEN	R/W	Output Compare Channel3 Preload Enable
2	OC3FEN	R/W	Output Compare Channel3 Fast Enable 0: Disable 1: Enable This bit is used to improve the response of the capture/compare output to the trigger input event.

Field	Name	R/W	Description
1:0	CC3SEL	R/W	Capture/Compare Channel 3 Selection This bit defines the input/output direction and selects the input pin. 00: CC3 channel is output 01: CC3 channel is input, and IC3 is mapped on TI3 10: CC3 channel is input, and IC3 is mapped on TI4 11: CC3 channel is input, and IC3 is mapped on TRC, and only works in internal trigger input Note: This bit can be written only when the channel is closed (TMRx_CCEN register CC3EN=0).

Input capture mode:

Field	Name	R/W	Description
31:16	Reserved		
15:12	IC4F	R/W	Input Capture Channel 4 Filter Configuration
11:10	IC4PSC	R/W	Input Capture Channel4 Perscaler Configuration
9:8	CC4SEL	R/W	Capture/Compare Channel 4 Select 00: CC4 channel is output 01: CC4 channel is input, and IC4 is mapped on TI4 10: CC4 channel is input, and IC4 is mapped on TI3 11: CC4 channel is input, and IC4 is mapped on TRC, and only works in internal trigger input Note: This bit can be written only when the channel is closed (TMRx_CCEN register CC4EN=0).
7:4	IC3F	R/W	Input Capture Channel 3 Filter Configuration
3:2	IC3PSC	R/W	Input Capture Channel 3 Perscaler Configuration 00：PSC=1 01：PSC=2 10：PSC=4 11：PSC=8 PSC is prescaler factor; capture is triggered once by every PSC events.
1:0	CC3SEL	R/W	Capture/Compare Channel 3 Select 00: CC3 channel is output 01: CC3 channel is input, and IC3 is mapped on TI3 10: CC3 channel is input, and IC3 is mapped on TI4 11: CC3 channel is input, and IC3 is mapped on TRC, and only works in internal trigger input Note: This bit can be written only when the channel is closed (TMRx_CCEN register CC3EN=0).

14.6.9 Capture/Compare enable register (TMRx_CCEN) (x=1)

Offset address: 0x20

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:14	Reserved		
13	CC4POL	R/W	Capture/Compare Channel4 Output Polarity Configure

Field	Name	R/W	Description
			Refer to CCEN_CC1POL
12	CC4EN	R/W	Capture/Compare Channel4 Output Enable Refer to CCEN_CC1EN
11	CC3NPOL	R/W	Capture/Compare Channel3 Complementary Output Polarity Configure Refer to CCEN_CC1NPOL
10	CC3NEN	R/W	Capture/Compare Channel3 Complementary Output Enable Refer to CCEN_CC1NEN
9	CC3POL	R/W	Capture/Compare Channel3 Output Polarity Configure Refer to CCEN_CC1POL
8	CC3EN	R/W	Capture/Compare Channel3 Output Enable Refer to CCEN_CC1EN
7	CC2NPOL	R/W	Capture/Compare Channel2 Complementary Output Polarity Configure Refer to CCEN_CC1NPOL
6	CC2NEN	R/W	Capture/Compare Channel2 Complementary Output Enable Refer to CCEN_CC1NEN
5	CC2POL	R/W	Capture/Compare Channel2 Output Polarity Configure Refer to CCEN_CC1POL
4	CC2EN	R/W	Capture/Compare Channel2 Output Enable Refer to CCEN_CC1EN
3	CC1NPOL	R/W	Capture/Compare Channel1 Complementary Output Polarity 0: OC1N is active high 1: OC1N is active low Note: When the protection level is 2 or 3, this bit cannot be modified
2	CC1NEN	R/W	Capture/Compare Channel1 Complementary Output Enable 0: Disable 1: Enable
1	CC1POL	R/W	Capture/Compare Channel1 Output Polarity Configure When CC1 channel is configured as output: 0: OC1 is active high 1: OC1 is active low When CC1 channel is configured as input: 0: Phase not reversed: capture at the rising edge of IC1; phase not reversed when IC1 is used as external trigger. 1: Phase reversed, capture at the falling edge of ICC1; phase reversed when IC1 is used as external trigger. Note: When the protection level is 2 or 3, this bit cannot be modified
0	CC1EN	R/W	Capture/Compare Channel1 Output Enable When the capture/compare channel 1 is configured as output: 0: Disable output 1: Enable output When the capture/compare channel 1 is configured as input: This bit determines whether the value CNT of the counter can be captured and enter TMRx_CC1 register 0: Disable capture 1: Enable capture

14.6.10 Counter register (TMRx_CNT) (x=1)

Offset address: 0x24

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15:0	CNT	R/W	Counter Value

14.6.11 Prescaler register (TMRx_PSC) (x=1)

Offset address: 0x28

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15:0	PSC	R/W	Prescaler Value Clock frequency of counter (CK_CNT)=f_{CK_PSC}/(PSC+1)

14.6.12 Auto reload register (TMRx_AUTORLD) (x=1)

Offset address: 0x2C

Reset value: 0x0000 FFFF

Field	Name	R/W	Description
31:16	Reserved		
15:0	AUTORLD	R/W	Auto Reload Value When the value of auto reload is empty, the counter will not count.

14.6.13 Repeat count register (TMRx_REPCNT) (x=1)

Offset address: 0x30

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:8	Reserved		
7:0	REPCNT	R/W	Repetition Counter Value When the count value of the repeat counter is reduced to 0, an update event will be generated, and the counter will start counting again from the REPCNT value; the new value newly written to this register is valid only when an update event occurs in next cycle.

14.6.14 Channel 1 capture/compare register (TMRx_CC1) (x=1)

Offset address: 0x34

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		

Field	Name	R/W	Description
15:0	CC1	R/W	Capture/Compare Channel 1 Value When the capture/compare channel 1 is configured as input mode: CC1 contains the counter value transmitted by the last input capture channel 1 event. When the capture/compare channel 1 is configured as output mode: CC1 contains the value currently loaded in the capture/compare register Compare the value CC1 of the capture and compare channel 1 with the value CNT of the counter to generate the output signal on OC1. When the output compare preload is disabled (OC1PEN=0 for TMRx_CCM1 register), the written value will immediately affect the output comparison results; If the output compare preload is enabled (OC1PEN=1 for TMRx_CCM1 register), the written value will affect the output compare result when an update event is generated.

14.6.15 Channel 2 capture/compare register (TMRx_CC2) (x=1)

Offset address: 0x38

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15:0	CC2	R/W	Capture/Compare Channel2 Value Refer to TMRx_CC1

14.6.16 Channel 3 capture/compare register (TMRx_CC3) (x=1)

Offset address: 0x3C

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15:0	CC3	R/W	Capture/Compare Channel 3 Value Refer to TMRx_CC1

14.6.17 Channel 4 capture/compare register (TMRx_CC4) (x=1)

Offset address: 0x40

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15:0	CC4	R/W	Capture/Compare Channel 4 Value Refer to TMRx_CC1

14.6.18 Brake and dead zone register (TMRx_BDT) (x=1)

Offset address: 0x44

Reset value: 0x0000 0000

Note: According to the lock setting, AOEN, BRKPOL, BRKEN, IMOS, RMOS and DTS[7:0] bits all can be write-protected, and it is necessary to configure them when writing to TMRx_BDT register for the first time.

Field	Name	R/W	Description
31:16	Reserved		
15	MOEN	R/W	PWM Main Output Enable 0: Disable the output of OCx and OCxN or force the output of idle state 1: When CcxEN and CCxNEN bits of the TMRx_CCEN register are set, enable Ocx and OcxN output When the brake input is valid, it is cleared to 0 by hardware asynchronously. Note: Setting 1 by software or setting 1 automatically depends on AOEN bit of the TMRx_BDT register.
14	AOEN	R/W	Automatic Output Enable 0: MOEN can only be set to 1 by software 1: MOEN can be set to 1 by software or be automatically set to 1 at the next update event (braking input is invalid) Note: When the protection level is 1, this bit cannot be modified.
13	BRKPOL	R/W	Brake Polarity Configure 0: The brake input BRK is valid at low level 1: The brake input BRK is valid at high level Note: When the protection level is 1, this bit cannot be modified. Writing to this bit requires an APB clock delay before use.
12	BRKEN	R/W	Brake Function Enable 0: Disable 1: Enable Note: When the protection level is 1, this bit cannot be modified.
11	RMOS	R/W	Run Mode Off-state Configure Run mode means MOEN=1; disable means CcxEN=0; this bit describes the impact of different values for this bit on the output waveform when MOEN=1 and CcxEN changes from 0 to 1. 0: Disable Ocx/OcxN output 1: Ocx/OcxN first outputs invalid level (the specific level value is affected by the polarity configuration)
10	IMOS	R/W	Idle Mode Off-state Configure Idle mode means MOEN=0; disable means CcxEN=0; this bit describes the impact of different values for this bit on the output waveform when MOEN=0 and CcxEN changes from 0 to 1. 0: Disable Ocx/OcxN output 1: If CCxEN=1, the invalid level is output during the dead time (the specific level value is affected by the polarity configuration), and the idle level is output after the dead time
9:8	LOCKCFG	R/W	Lock Write Protection Mode Configuration 00: No Lock write protection; it cannot be written to the register directly 01: Lock write protection level 1 It cannot be written to DTS, BRKEN, BRKPOL and AOEN bits of TMRx_BDT, and OCxOIS and OCxNOIS bits of TMRx_CTRL2 register. 10: Lock write protection level 2

Field	Name	R/W	Description
			It cannot be written to all bits of protection level 1, CCxPOL and OCxNPOL bits in TMRx_CCEN register, and RMOS and IMOS bits in TMRx_BDT register. 11: Lock write protection level 3 It cannot be written to all bits of protection level 2, and OCxMOD and OCxPEN bits of TMRx_CCMx register. Note: After system reset, the lock write protect bit can only be written once.
7:0	DTS	R/W	Dead Time Setup DT is the dead zone duration, and the relationship between DT and register DTS is as follows: DTS[7:5]=0xx=>DT=DTS[7:0]×T_{DTS}，T_{DTS}=TDTS； DTS[7:5]=10x=>DT=（64+DTS[5:0]）×T_{DTS}，T_{DTS}=2×T_{DTS}； DTS[7:5]=110=>DT=（32+DTS[4:0]）×T_{DTS}，T_{DTS}=8×T_{DTS}； DTS[7:5]=111=>DT=（32+DTS[4:0]）×T_{DTS}，T_{DTS}=16×T_{DTS}； For example: assuming T_{DTS} =125ns (8MHz), the dead time setting is as follows: If the step time is 125ns, the dead time can be set from 0 to 15875ns; If the step time is 250ns, the dead time can be set from 16us to 31750ns; If the step time is 1us, the dead time can be set from 32us to 63us; If the step time is 2us, the dead time can be set from 64us to 126us. Note: Once LOCK level (LOCKCFG bit in TMRx_BDT register) is set to 1, 2 or 3, these bits cannot be modified.

14.6.19 DMA control register (TMRx_DCTRL) (x=1)

Offset address: 0x48

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:13	Reserved		
12:8	DBLEN	R/W	DMA Burst Transfer Length Setup These bits define the transmission length and transmission times of DMA in continuous mode. The data transmitted can be 16 bits and 8 bits. When reading/writing TMRx_DMADDR register, the timer will conduct a continuous transmission; 00000: Transmission once 00001: Transmission twice 00010: Transmission for three times …… 10001: Transmission for 18 times The transmission address formula is as follows: Transmission address=TMRx_CTRL1 address (slave address) +DBADDR+DMA index; DMA index=DBLEN For example: DBLEN=7, DBADDR=TMR2_CTRL1 (slave address) means the address of the data to be transmitted, while the address +DBADDR+7 of TMRx_CTRL1 means the address of the data to be written/read,

Field	Name	R/W	Description
			Data transmission will occur to: TMRx_CTRL1 address + seven registers starting from DBADDR. The data transmission will change according to different DMA data length: When the transmission data is set to 16 bits, the data will be transmitted to seven registers When the transmission data is set to 8 bits, the data of the first register is the MSB bit of the first data, the data of the second register is the LSB bit of the first data, and the data will still be transmitted to seven registers.
7:5	Reserved		
4:0	DBADDR	R/W	DMA Base Address Setup These bits define the base address of DMA in continuous mode (when reading or writing TMRx_DMADDR register), and DBADDR is defined as the offset from the address of TMRx_CTRL1 register: 00000：TMRx_CTRL1 00001：TMRx_CTRL2 00010：TMRx_SMCTRL …….

14.6.20 DMA address register of continuous mode (TMRx_DMADDR) (x=1)

Offset address: 0x4C

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	DMADDR	R/W	DMA Register for Burst Transfer Read or write operation access of TMRx_DMADDR register may lead to access to the register in the following address: TMRx_CTRL1 address + (DBADDR+DMA index) ×4 Where: "TMRx_CTRL1 address" is the address of control register 1 (TMRx_CTRL1); "DBADDR" is the base address defined in TMRx_DCTRL register; "DMA index" is the offset automatically controlled by DMA, and it depends on DBLEN defined in TMRx_DCTRL register.

15 General-purpose Timer (TMR2/3/4)

15.1 Introduction

The general-purpose timer takes the time base unit as the core, and has the functions of input capture and output compare, and can be used to measure the pulse width, frequency and duty cycle, and generate the output waveform. It includes a 16-bit auto reload counter (achieve count-up, count-down and center-aligned count).

The timers are independent of each other, and they can achieve synchronization and cascading.

15.2 Main characteristics

(1) Timebase unit
- Counter: 16-bit counter, supporting count-up, count-down and count-up/down (center-aligned) count.
- Prescaler: 16-bit programmable prescaler
- Autoreload function

(2) Clock source selection
- Internal clock
- External input
- External trigger
- Internal trigger

(3) Input capture function
- Counting function
- PWM input
- Encoder interface mode

(4) Output compare function
- PWM output mode
- Forced output mode
- Single-pulse mode

(5) Master/Slave mode controller of timer
- Timers can be synchronized and cascaded
- Support multiple slave modes and synchronization signals

(6) Interrupt and DMA request event
- Update event (counter overrun/underrun, counter initialization)
- Trigger event (counter start, stop, internal/external trigger)
- Input capture
- Output compare

(7) The timer has an independent DMA request mechanism

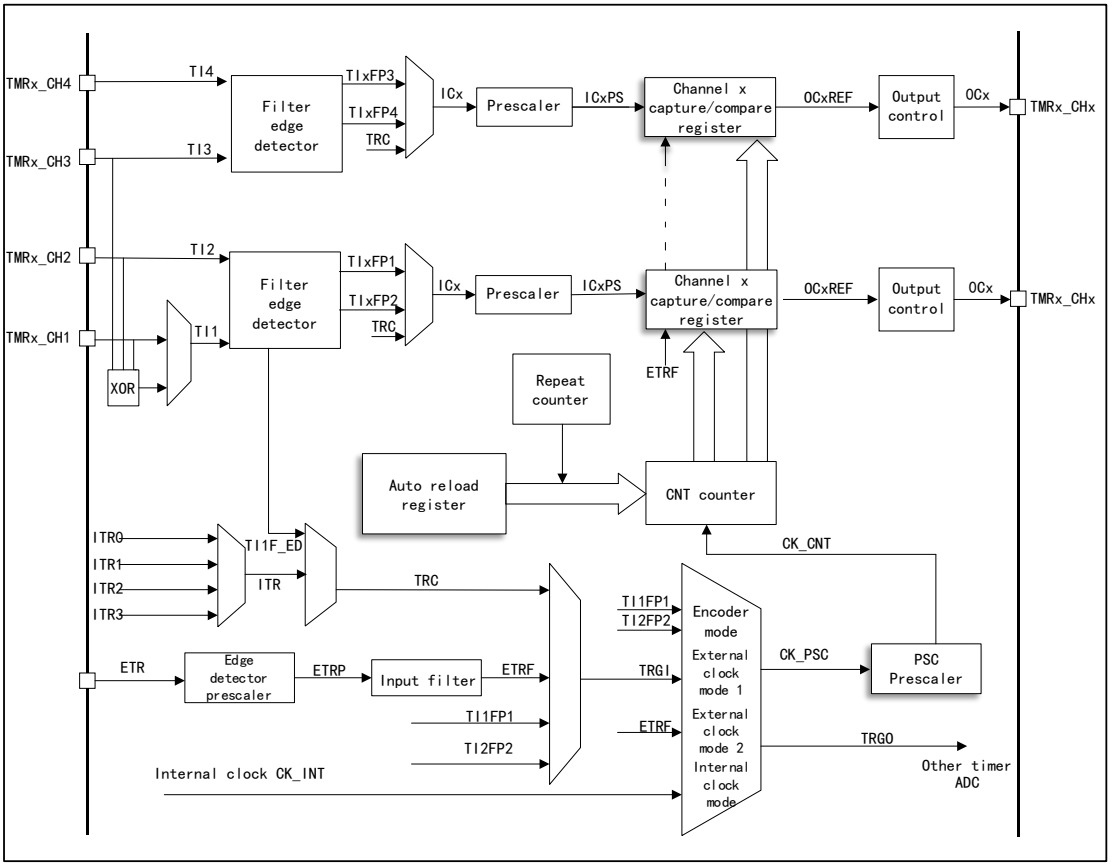
(8) The comparator output can be connected to the timer ETR or input channels for counting

(9) Support incremental (quadrature) encoders and Hall sensor circuits for positioning

(10) Support ETR input (external trigger input) functionality, which can be used as an external clock or for cycle-by-cycle current management

15.3 Structure block diagram

Figure 42 General-purpose Timer Structure Block Diagram



15.4 Functional Description

15.4.1 Clock source selection

The general-purpose timer has four clock sources.

Internal clock

It is TMRx_CLK from RCM, namely the driving clock of the timer; when the slave mode controller is disabled, the clock source CK_PSC of the prescaler is

driven by the internal clock CK_INT.

External clock mode 1

The trigger signal generated from the input channel TI1/2/3/4 of the timer after polarity selection and filtering is connected to the slave mode controller to control the work of the counter. Besides, the pulse signal generated by the input of Channel 1 after double-edge detection of the rising edge and the falling edge is logically equal or the future signal is TI1F_ED signal, namely double-edge signal of TIF_ED. Especially the PWM input can only be input by TI1/2.

External clock mode 2

After polarity selection, frequency division and filtering, the signal from external trigger interface (ETR) is connected to the slave mode controller through trigger input selector to control the work of the counter.

Internal trigger input

The timer is set to work in slave mode, and the clock source is the output signal of other timers. At this time, the clock source has no filtering, and the synchronization or cascading between timers can be realized. The master mode timer can reset, start, stop or provide clock for the slave mode timer.

15.4.2 Timebase unit

The time base unit in the general-purpose timer contains three registers

- 16-bit counter register (CNT)
- 16-bit auto reload register (AUTORLD)
- 16-bit prescaler register (PSC)

Counter CNT

There are three counting modes for the counter in the general-purpose timer

- Count-up mode
- Count-down mode
- Center-aligned mode

Count-up mode

Set to the count-up mode by configuring CNTDIR bit of control register (TMRx_CTRL1).

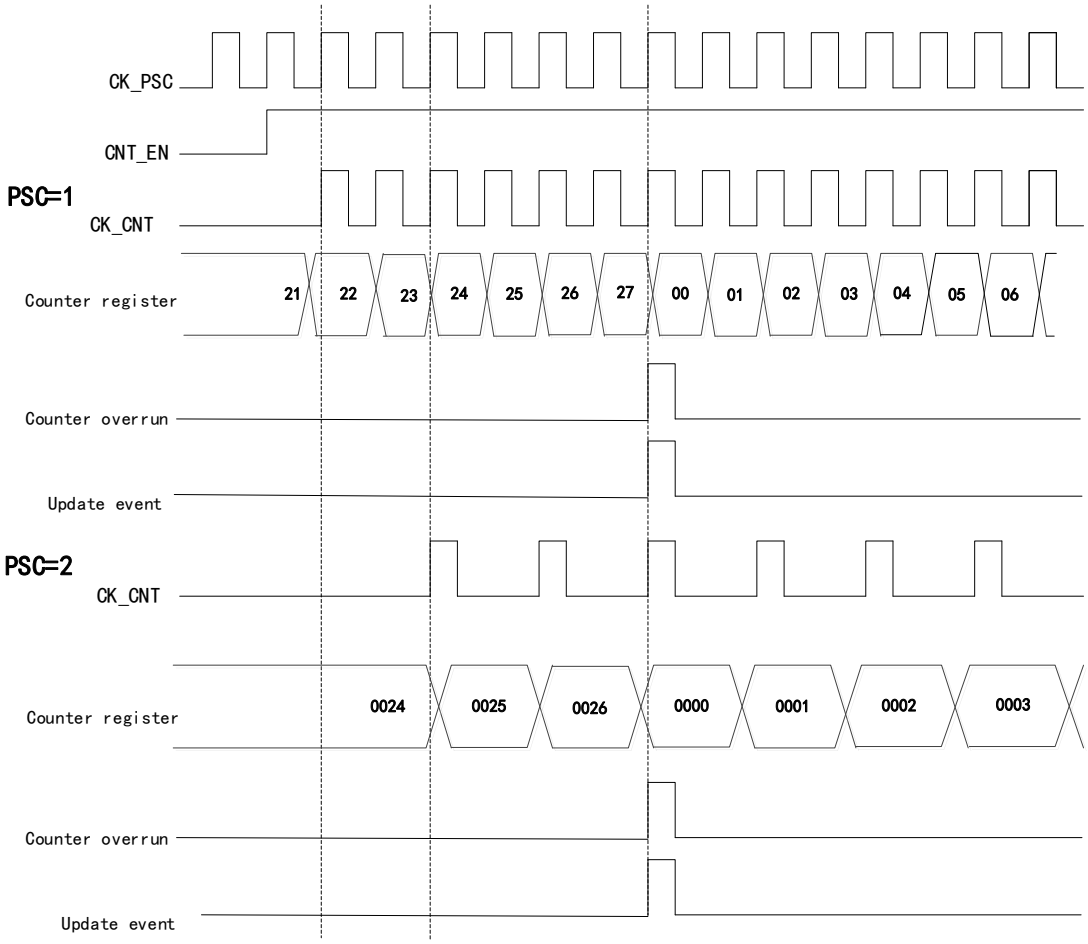
When the counter is in count-up mode, the counter will count up from 0; every time a pulse is generated, the counter will increase by 1 and when the value of the counter (TMRx_CNT) is equal to the value of the auto reload (TMRx_AUTORLD), the counter will start to count from 0 again, a count-up overrun event will be generated, and the value of the auto reload

(TMRx_AUTORLD) is written in advance.

When the counter overruns, an update event will be generated. At this time, the auto reload shadow register and the prescaler buffer will be updated. The update event can be disabled by configuring UD bit of control register TMRx_CTRL1.

The figure below is the timing diagram of count-up mode when the division factor is 1 or 2

Figure 43 Timing Diagram of Count-up Mode when Division Factor is 1 or 2



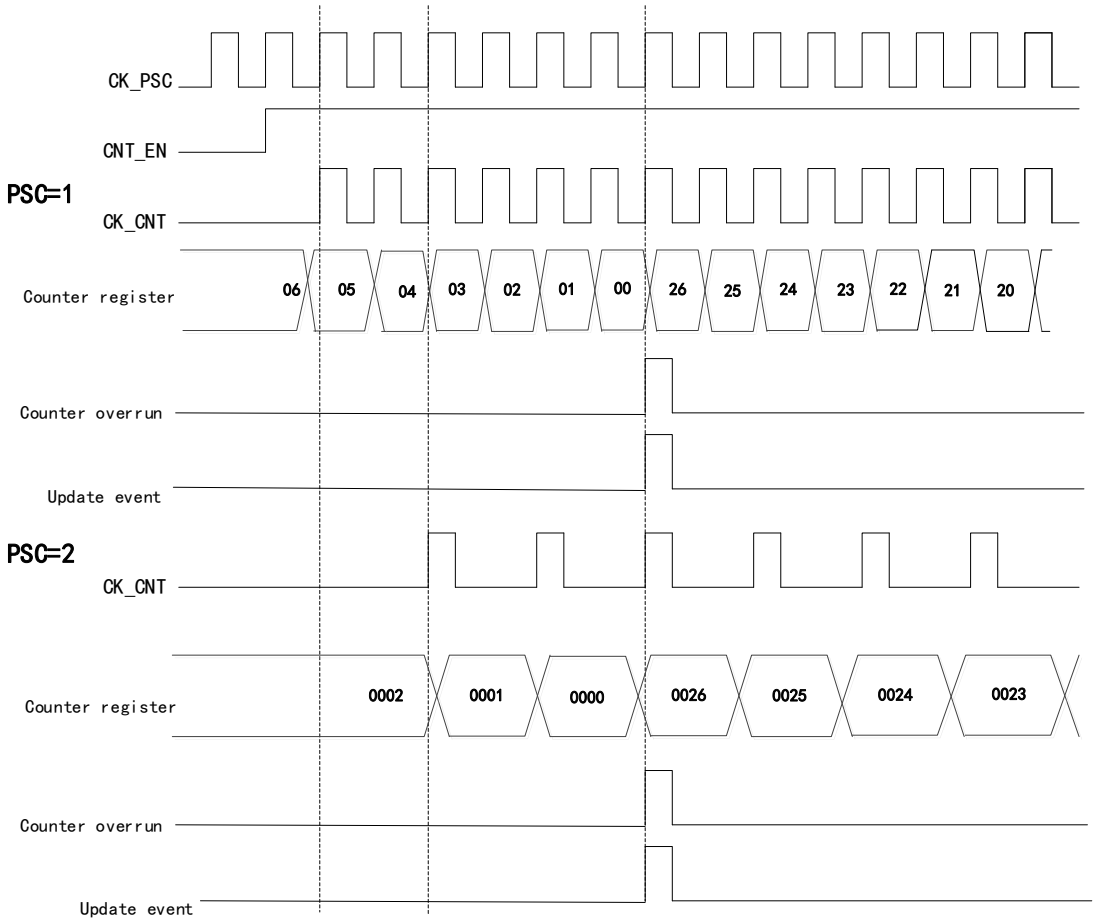
Count-down mode

Set to the count-down mode by configuring CNTDIR bit of control register (TMRx_CTRL1).

When the counter is in count-down mode, it will start to count down from the value of the auto reload (TMRx_AUTORLD); every time a pulse is generated, the counter will decrease by 1 and when it becomes 0, the counter will start to count again from (TMRx_AUTORLD), meanwhile, a count-down overrun event will be generated, and the value of the auto reload (TMRx_AUTORLD) is written in advance.

When the counter overruns, an update event will be generated. At this time, the auto reload shadow register and the prescaler buffer will be updated. The update event can be disabled by configuring the UD bit of the TMRx_CTRL1 register.

Figure 44 Timing Diagram of Count-down Mode when Division Factor is 1 or 2

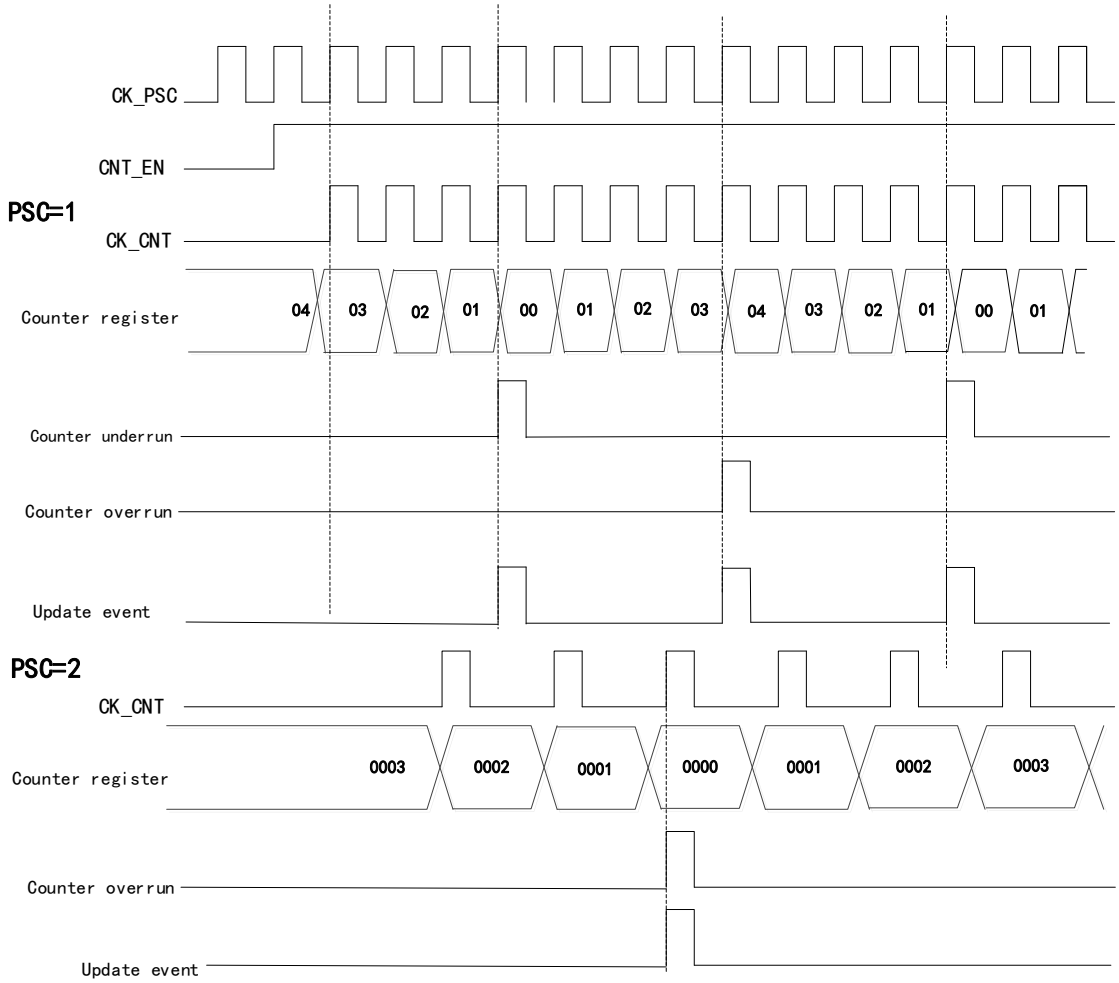


Center-aligned mode

Set to the center-aligned mode by configuring CNTDIR bit of control register (TMRx_CTRL1).

When the counter is in center-aligned mode, the counter counts up from 0 and when it reaches the value of auto reload (TMRx_AUTORLD), it counts down to 0 from the value of the auto reload (TMRx_AUTORLD), which will repeat; in counting up, when the counter value is (AUTORLD-1), a counter overrun event will be generated; in counting down, when the counter value is 1, a counter underrun event will be generated.

Figure 45 Timing Diagram of Center-aligned Mode when Division Factor is 1 or 2



Prescaler PSC

The prescaler is 16 bits and programmable, and it can divide the clock frequency of the counter to any value within 1~65536 (controlled by TMRx_PSC register), and after frequency division, the clock will drive the counter CNT to count. The prescaler has a buffer, which can be changed during running.

15.4.3 Input capture

Input capture channel

The general-purpose timer has four independent capture/compare channels, each of which is surrounded by a capture/compare register.

In the input capture, the measured signal will enter from the external pin TI1/2/3/4 of the timer, first pass through the edge detector and input filter, and then enter the capture channels. Each capture channel has a corresponding capture register. When the capture occurs, the value of the counter CNT will be latched in the capture register CCx. Before entering the capture register, the signal will pass through the prescaler to set how many events to capture at a

time.

Input capture application

Input capture is used to capture external events, and can give the time flag to indicate the occurrence time of the event and measure the pulse jump edge events (measure the frequency or pulse width), for example, if the selected edge appears on the input pin, the TMRx_CCx register will capture the current value of the counter and the CCxIFLG bit of the status register TMRx_STS will be set to 1; if CCxIEN=1, an interrupt will be generated.

In capture mode, the timing, frequency, cycle and duty cycle of a waveform can be measured. In the input capture mode, the edge selection is set to rising edge detection. When the rising edge appears on the capture channel, the first capture occurs, at this time, the value of the counter CNT will be latched in the capture register CCx; at the same time, it will enter the capture interrupt, a capture will be recorded in the interrupt service program and the value will be recorded. When the next rising edge is detected, the second capture occurs, the value of counter CNT will be latched in capture register CCx again, at this time, it will enter the capture interrupt again; read the value of capture register and the cycle of this pulse signal will be obtained by capture.

15.4.4 Output compare

There are eight modes of output compare: freeze, channel x is valid level when matching, channel x is invalid level when matching, flip, force is invalid, force is valid, PWM mode 1 and PWM mode 2, which are configured by OCxMOD bit in TMRx_CCMx register and can control the waveform of output signal in output compare mode.

Output compare application

In the output compare mode, the position, polarity, frequency and time of the pulse generated by the timer can be controlled.

When the value of the counter is equal to that of the capture/compare register, the channel output can be set as high level, low level or flip by configuring the OCxMOD bit in TMRx_CCMx register and the CCxPOL bit in the output polarity TMRx_CCEN register.

When CCxIFLG=1 in TMRx_STS register, if CCxIEN=1 in TMRx_DIEN register, an interrupt will be generated; if CCDSEL=1 in TMRx_CTRL2 register, a DMA request will be generated.

15.4.5 PWM output mode

PWM mode is pulse signal that can be adjusted by external output of the timer. The pulse width of the signal is determined by the value of the compare register CCx, and the cycle is determined by the value of the auto reload AUTORLD.

PWM output mode contains PWM mode 1 and PWM mode 2; PWM mode 1 and PWM mode 2 are divided into count-up, count-down and center alignment counting; in PWM mode 1, if the value of the counter CNT is less than the value of the compare register CCx, the output level will be valid; otherwise, it will be invalid.

Set the timing diagram in PWM mode 1 when CCx=5, AUTORLD=7:

Figure 46 Timing Diagram of PWM1 Count-up Mode

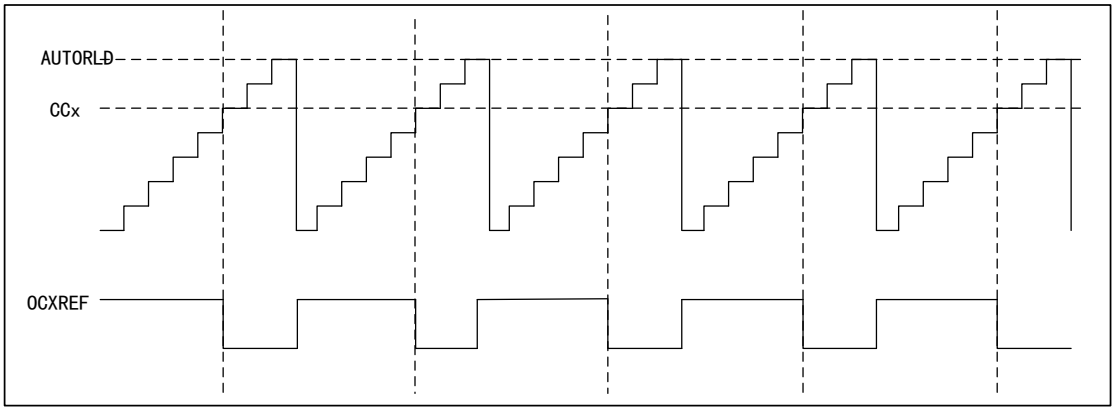


Figure 47 Timing Diagram of PWM1 Count-down Mode

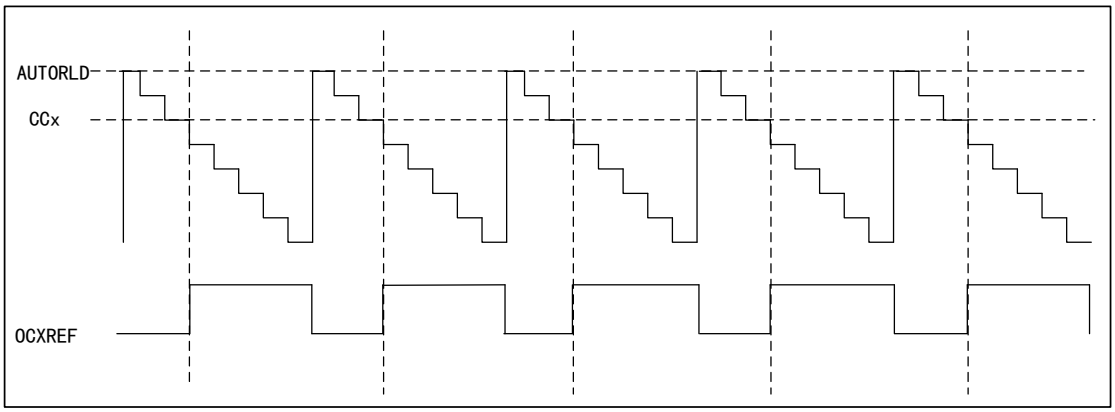
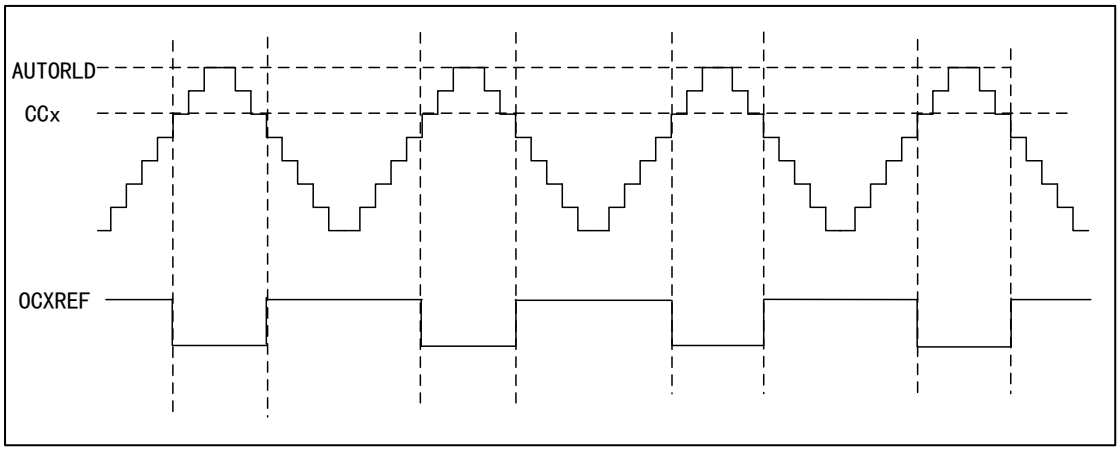


Figure 48 Timing Diagram of PWM1 Center-aligned Mode



In PWM mode 2, if the value of the counter CNT is less than that of the compare register CCx, the output level will be invalid; otherwise, it will be valid.

Set the timing diagram of PWM mode 2 when CCx=5, AUTORLD=7

Figure 49 Timing Diagram of PWM2 Count-up Mode

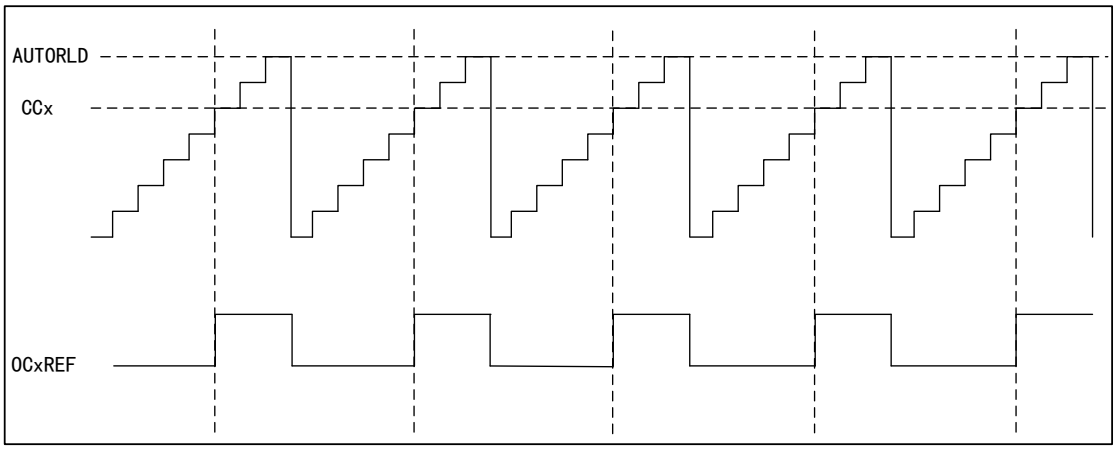


Figure 50 Timing Diagram of PWM2 Count-down Mode

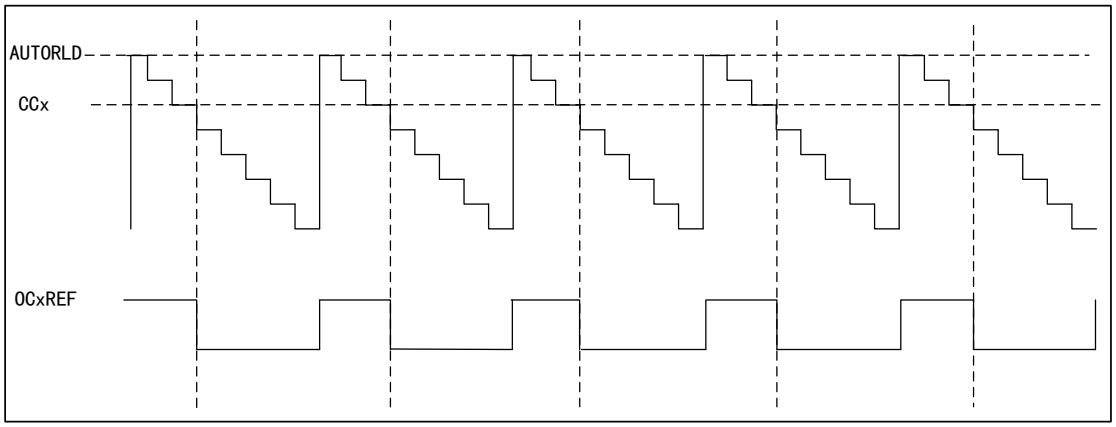
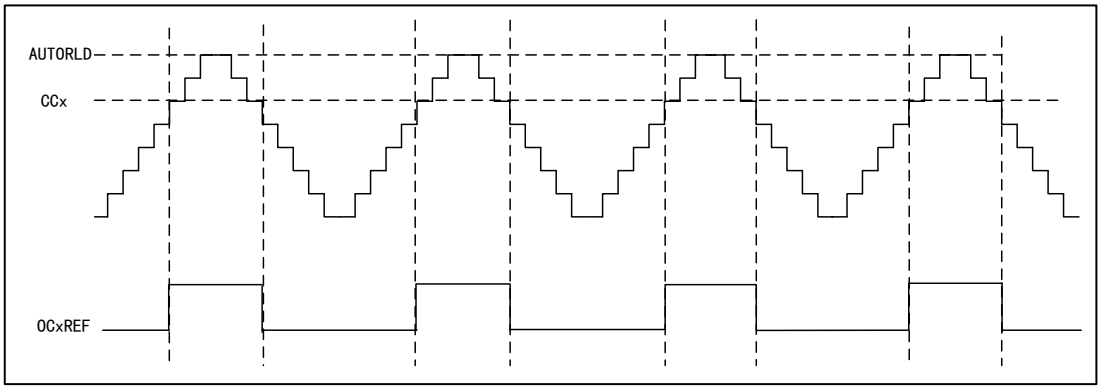


Figure 51 Timing Diagram of PWM2 Center-aligned Mode



15.4.6 PWM input mode

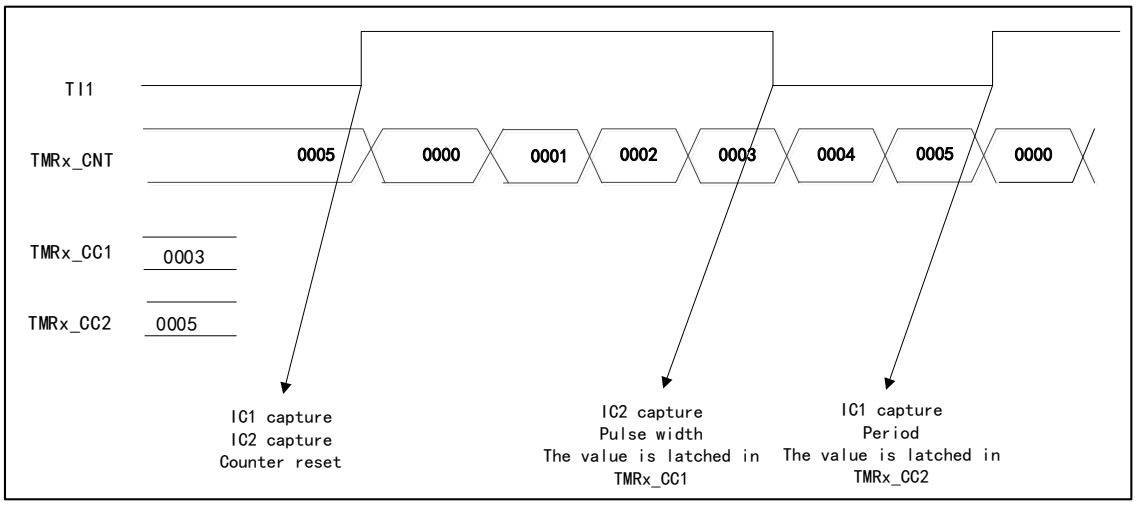
PWM input mode is a particular case of input capture.

In PWM input mode, as only TI1FP1 and TI1FP2 are connected to the slave mode controller, input can be performed only through the channels TMRx_CH1 and TMRx_CH2, which need to occupy the capture registers of CH1 and CH2.

In PWM input mode, the PWM signal enters from TMRx_CH1, and the signal will be divided into two channels, one can measure the period, and the other can measure the duty cycle. In the configuration, it is only required to set the polarity of one channel, and the other will be automatically configured with the opposite polarity.

In this mode, the slave mode controller should be configured as the reset mode (SMFSEL bit of TMRx_SMCTRL register)

Figure 52 Timing Diagram of PWM Input Mode



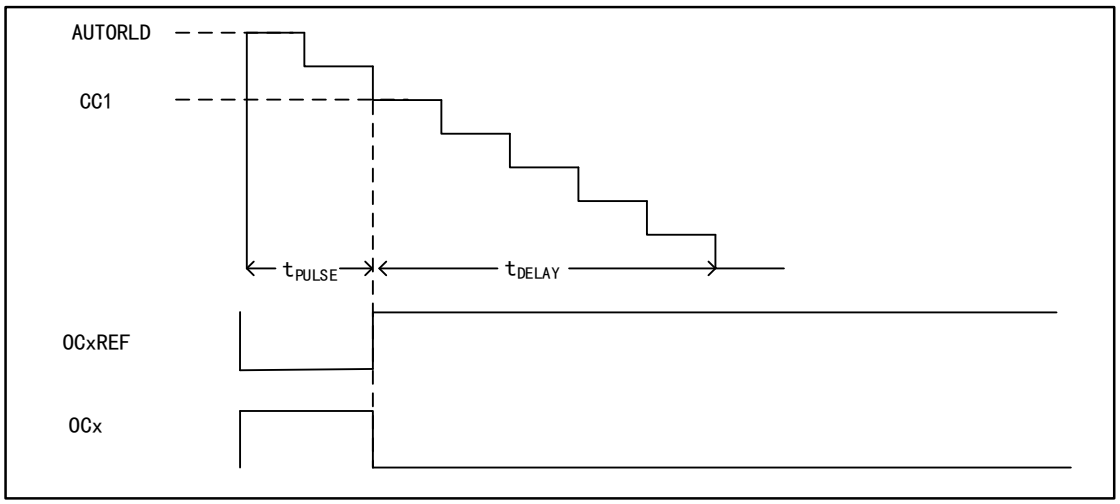
15.4.7 Single-pulse mode

The single-pulse mode is a special case of timer compare output, and is also a special case of PWM output mode.

Set SPMEN bit of TMRx_CTRL1 register, and select the single-pulse mode. After the counter is started, a certain number of pulses will be output before the update event occurs. When an update event occurs, the counter will stop counting, and the subsequent PWM waveform output will no longer be changed.

After a certain controllable delay, a pulse with controllable pulse width is generated in single-pulse mode through the program. The delay time is defined by the value of TMRx_CCx register; in the count-up mode, the delay time is CCx and the pulse width is AUTORLD-CCx; in the count-down mode, the delay time is AUTORLD-CCx and the pulse width is CCx.

Figure 53 Timing Diagram of Single-pulse Mode



15.4.8 Forced output mode

In the forced output mode, the comparison result is ignored, and the corresponding level is directly output according to the configuration instruction.

- CCxSEL=00 for TMRx_CCMx register, set CCx channel as output
- OCxMOD=100/101 for TMRx_CCMx register, set to force OCxREF signal to invalid/valid

In this mode, the corresponding interrupt and DMA request will still be generated.

15.4.9 Encoder interface mode

The encoder interface mode is equivalent to an external clock with direction selection. In the encoder interface mode, the content of the timer can always indicate the position of the encoder.

The method of selecting encoder interface is as follows:

- By setting SMFSEL bit of TMRx_SMCTRL register, set the counter to count on the edge of TI1 channel /TI2 channel, or count on the edge of TI1 and TI2 at the same time.
- Select the polarity of TI1 and TI2 by setting the CC1POL and CC2POL bits of TMRx_CCEN register.
- Select to filter or not by setting the IC1F and IC2F bits of TMRx_CCM1 register.
- The two input TI1 and TI2 can be used as the interface of incremental encoder. The counter is driven by the effective jump of the signals TI1FP1 and TI2FP2 after filtering and edge selection in TI1 and TI2.
- The count pulse and direction signal are generated according to the input signals of TI1 and TI2
- The counter will count up/down according to the jumping sequence of the input signal

- Set CNTDIR of control register TMRx_CTRL1 to be read-only (CNTDIR will be re-calculated due to jumping of any input end)

The change mechanism of counter count direction is shown in the figure below:

Table 50 Relationship between Count Direction and Encoder

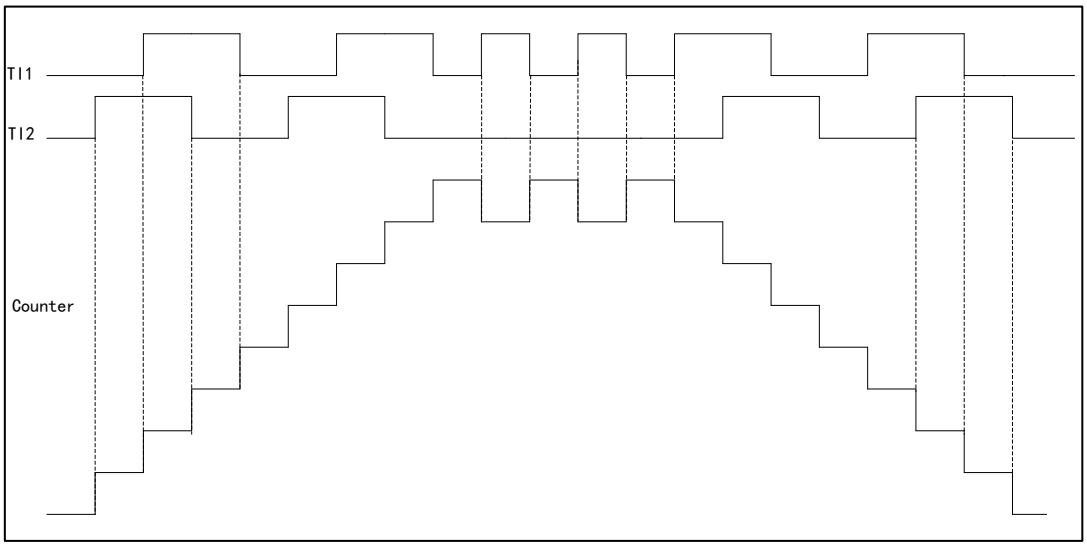
Effective edge		**Count only in TI1**		**Count only in TI2**		**Count in both TI1 and TI2**	
Level of relative signal		High	Low	High	Low	High	Low
TI1FP1	Rising edge	—		Count down	Count up	Count down	Count up
	Falling edge			Count up	Count down	Count up	Count down
TI2FP2	Rising edge	Count up	Count down	—		Count up	Count down
	Falling edge	Count down	Count up			Count down	Count up

The external incremental encoder can be directly connected with MCU, not needing external interface logic, so the comparator is used to convert the differential output of the encoder to digital signal to increase the immunity to noise interference.

Among the following examples:

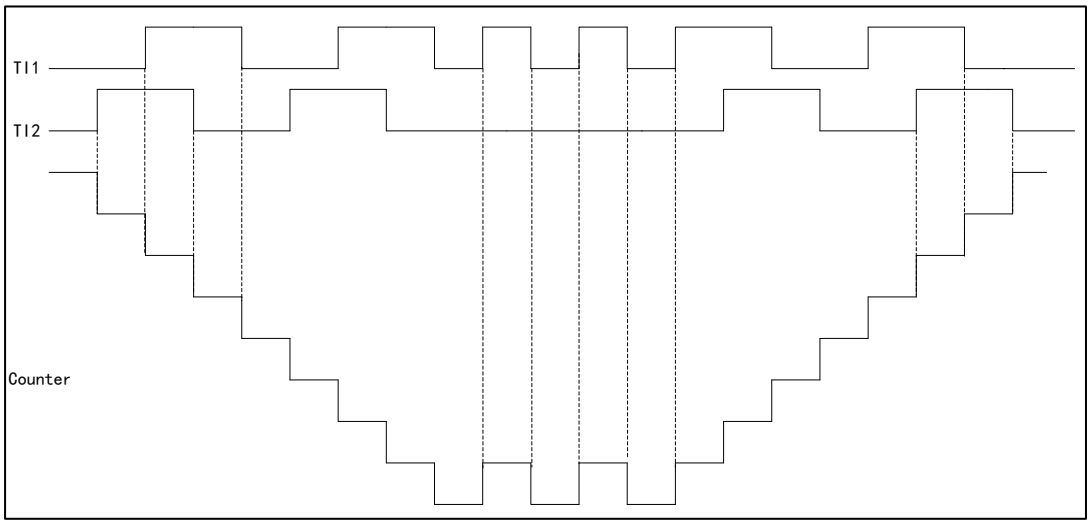
- IC1FP1 is mapped to TI1
- IC2FP2 is mapped to TI2
- Neither IC1FP1 nor IC2FP2 is phase-inverting
- The input signal is valid at the rising edge and falling edge
- Enable the counter

Figure 54 Counter Operation Example in Encoder Mode



For example, when TI1 is at low level, and TI2 is in rising edge state, the counter will count up.

Figure 55 Example of Encoder Interface Mode of IC1FP1 Reversed Phase



For example, when TI1 is at low level, and the rising edge of TI2 jumps, the counter will count down.

15.4.10 Slave Mode

TMRx timer can synchronize external trigger

- Reset mode
- Gated mode
- Trigger mode

SMFSEL bit in TMRx_SMCTRL register can be set to select the mode

SMFSEL=100 set the reset mode, SMFSEL=101 set the gated mode, and SMFSEL=110 set the trigger mode.

In the reset mode, when a trigger input event occurs, the counter and prescaler will be initialized, and the rising edge of the selected trigger input (TRGI) will reinitialize the counter and generate a signal to update the register.

In the gated mode, the enable of the counter depends on the high level of the selected input end. When the trigger input is high, the clock of the counter will be enabled. Once the trigger input becomes low, the counter will stop (but not be reset). The start and stop of the counter are controlled.

In the trigger mode, the enable of the counter depends on the event on the selected input, the counter will be enabled at the rising edge of the trigger input (but not be reset), and only the start of the counter is controlled.

15.4.11 Timer interconnection

See the chapter of "14.4.14 Timer Interconnection" for details.

15.4.12 Interrupt and DMA request

The timer can generate an interrupt when an event occurs during operation

- Update event (counter overrun/underrun, counter initialization)
- Trigger event (counter start, stop, internal/external trigger)
- Capture/Compare event

Some internal interrupt events can generate DMA requests, and special interfaces can enable or disable trigger DMA requests.

15.4.13 Debug mode

The TMR2/3/4 can be configured in debug mode and choose to stop or continue to work. It depends on the DBG_TIMx_STOP (x=2,3,4) bit of DBGMCU_APBF register in DBGMCU module.

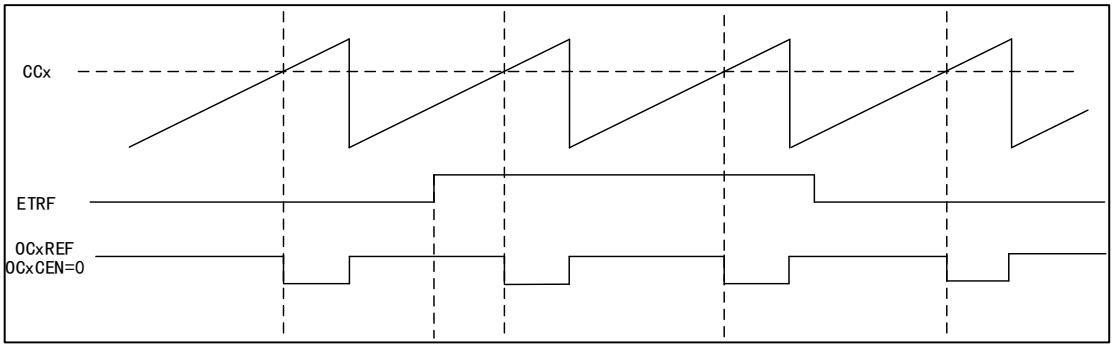
15.4.14 Clear OCxREF signal when an external event occurs

This function is used for output compare and PWM mode.

In one channel, the high level of ETRF input port will reduce the signal of OCxREF to low level, and the OCxCEN bit in capture/compare register TMRx_CCMx is set to 1, and OCxREF signal will remain low until the next update event occurs.

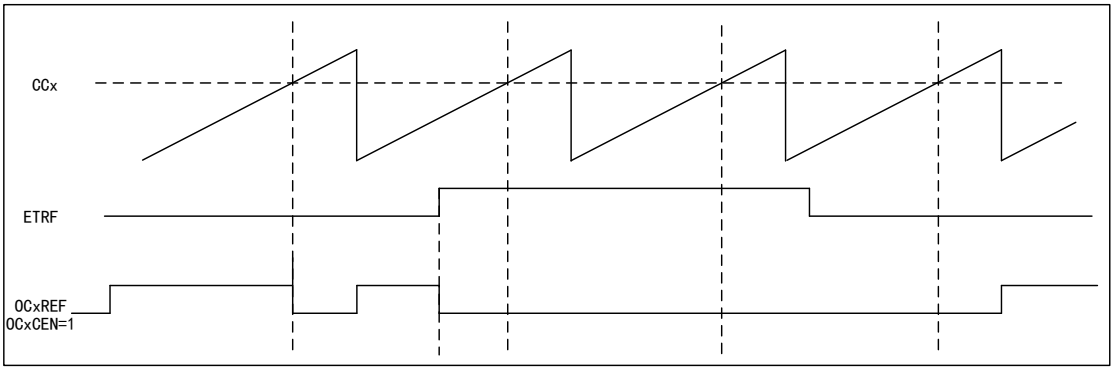
Set TMRx to PWM mode, disable the external trigger prescaler, and disable the external trigger mode 2; when ETRF input is high, set OCxCEN=0, and the output OCxREF signal is shown in the figure below.

Figure 56 OCxREF Timing Diagram



Set TMRx to PWM mode, disable the external trigger prescaler, and disable the external trigger mode 2; when ETRF input is high, set OCxCEN=1, and the output OCxREF signal is shown in the figure below.

Figure 57 OCxREF Timing Diagram



15.5 Register address mapping

In the following table, all registers of the general-purpose timer are mapped to a 16-bit addressable (address) space.

Table 51 General-purpose Timer Register Address Mapping

Register name	Description	Offset Address
TMRx_CTRL1	Control register 1	0x00
TMRx_CTRL2	Control register 2	0x04
TMRx_SMCTRL	Slave mode control register	0x08
TMRx_DIEN	DMA/Interrupt enable register	0x0C
TMRx_STS	Status register	0x10
TMRx_CEG	Control event generation register	0x14
TMRx_CCM1	Capture/Compare mode register 1	0x18
TMRx_CCM2	Capture/Compare mode register 2	0x1C
TMRx_CCEN	Enable capture/compare channel register	0x20
TMRx_CNT	Counter register	0x24
TMRx_PSC	Prescaler register	0x28
TMRx_AUTORLD	Auto reload register	0x2C
TMRx_CC1	Channel 1 capture/compare register	0x34
TMRx_CC2	Channel 2 capture/compare register	0x38
TMRx_CC3	Channel 3 capture/compare register	0x3C
TMRx_CC4	Channel 4 capture/compare register	0x40
TMRx_DCTRL	DMA control register	0x48
TMRx_DMADDR	DMA address register of continuous mode	0x4C

15.6 Register functional description

15.6.1 Control register 1 (TMRx_CTRL1)

Offset address: 0x00

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:10	Reserved		
9:8	CLKDIV	R/W	Clock Division For the configuration of digital filter, CK_INT provides the clock, and the clock of the digital filter can be adjusted by this bit. 00: $t_{DTS}=t_{CK\_INT}$ 01: $t_{DTS}=2\times t_{CK\_INT}$ 10: $t_{DTS}=4\times t_{CK\_INT}$ 11: Reserved
7	ARPEN	R/W	Auto-reload Preload Enable When the buffer is disabled, modification of TMRx_AUTORLD by program will immediately lead to modification of the values loaded to the counter; when the buffer is enabled, modification of TMRx_AUTORLD by program will lead to modification of the values loaded to the counter at the next update event. 0: Disable 1: Enable
6:5	CAMSEL	R/W	Center Aligned Mode Select In the center-aligned mode, the counter counts up and down alternately; otherwise, it will only count up or down. Different center-aligned modes affect the timing of setting the output compare interrupt flag bit of the output channel to 1; when the counter is disabled (CNTEN=0), select the center-aligned mode. 00: Edge-aligned mode 01: Center-aligned mode 1 (the output compare interrupt flag bit of output channel is set to 1 when counting down) 10: Center-aligned mode 2 (the output compare interrupt flag bit of output channel is set to 1 when counting up) 11: Center-aligned mode 3 (the output compare interrupt flag bit of output channel is set to 1 when counting up/down)
4	CNTDIR	R/W	Counter Direction This bit is read-only when the counter is configured as center-aligned mode or encoder mode. 0: Count up 1: Count down
3	SPMEN	R/W	Single Pulse Mode Enable When an update event is generated, the output level of the channel can be changed; in this mode, the CNTEN bit will be cleared, the counter will stop, and the subsequent output level of the channel will no long be changed. 0: Disable 1: Enable

Field	Name	R/W	Description
2	URSSEL	R/W	Update Request Source Select If interrupt or DMA is enabled, the update event can generate update interrupt or DMA request. Different update request sources can be selected by this bit. 0: The counter overruns or underruns Set UEG bit Update generated by slave mode controller 1: The counter overruns or underruns
1	UD	R/W	Update Disable Update event can cause AUTORLD, PSC and CCx to generate the value of update setting. 0: Enable update event (UEV) An update event can occur in any of the following situations: The counter overruns/underruns; Set UEG bit; Update generated by slave mode controller. 1: Disable update event
0	CNTEN	R/W	Counter Enable 0: Disable 1: Enable When the timer is configured as external clock, gated mode and encoder mode, it is required to write 1 to the bit by software to start regular work; when it is configured as the trigger mode, it can write 1 by hardware.

15.6.2 Control register 2 (TMRx_CTRL2)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:8	Reserved		
7	TI1SEL	R/W	Timer Input 1 Selection 0: TMRx_CH1 pin is connected to TI1 input 1: TMRx_CH1, TMRx_CH2 and TMRx_CH3 pins are connected to TI1 input after exclusive
6:4	MMSEL	R/W	Master Mode Signal Select The signals of timers working in master mode can be used for TRGO, so as to affect the work of timers in slave mode and cascaded with the master timer, and the specific impact is related to the configuration of slave mode timer. 000: Reset; the reset signal of master mode timer is used for TRGO 001: Enable; the counter enable signal of master mode timer is used for TRGO 010: Update; the update event of master mode timer is used for TRGO

Field	Name	R/W	Description
			011: Compare pulses; when the master mode timer captures/compares successfully (CCxIFLG=1), a pulse signal is output for TRGO 100: Compare mode 1; OC1REF is used to trigger TRGO 101: Compare mode 2; OC2REF is used to trigger TRGO 110: Compare mode 3; OC3REF is used to trigger TRGO 111: Compare mode 4; OC4REF is used to trigger TRGO
3	CCDSEL	R/W	Capture/Compare DMA Select 0: Transmit DMA request of CCx when CCx event occurs 1: Transmit DMA request of CCx when an update event occurs
2:0	Reserved		

15.6.3 Slave mode control register (TMRx_SMCTRL)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15	ETPOL	R/W	External Trigger Polarity Configure This bit decides whether the external trigger ETR is phase-inverting. 0: The external trigger ETR is not phase-inverting, and the high level or rising edge is valid 1: The external trigger ETR is phase-inverting, and the low level or falling edge is valid
14	ECEN	R/W	External Clock Enable Mode2 0: Disable 1: Enable Setting ECEN bit has the same function as selecting external clock mode 1 to connect TRG1 to ETRF; slave mode (reset, gating, trigger) can be used at the same time with external clock mode 2, but TRGI cannot be connected to ETRF in such case; when external clock mode 1 and external clock mode 2 are enabled at the same time, the input of external clock is ETRF.
13:12	ETPCFG	R/W	External Trigger Prescaler Configure The ETR (external trigger input) signal becomes ETRP after frequency division. The signal frequency of ETRP is at most 1/4 of TMRxCLK frequency; when ETR frequency is too high, the ETRP frequency must be reduced through frequency division. 00: Disable the prescaler; 01: ETR signal 2 divided frequency 10: ETR signal 4 divided frequency 11: ETR signal 8 divided frequency
11:8	ETFCFG	R/W	External Trigger Filter Configure 0000: Disable filter, sampled by f_{DTS} 0001: DIV=1，N=2 0010: DIV=1，N=4 0011: DIV=1，N=8 0100: DIV=2，N=6

Field	Name	R/W	Description
			0101: DIV=2，N=8 0110: DIV=4，N=6 0111: DIV=4，N=8 1000: DIV=8，N=6 1001: DIV=8，N=8 1010: DIV=16，N=5 1011: DIV=16，N=6 1100: DIV=16，N=8 1101: DIV=32，N=5 1110: DIV=32，N=6 1111: DIV=32，N=8 Sampling frequency=timer clock frequency/DIV; the filter length=N, and a jump is generated by every N events.
7	MSMEN	R/W	Master/slave Mode Enable 0: Invalid 1: Enable the master/slave mode
6:4	TRGSEL	R/W	Trigger Input Signal Select In order to avoid generating false edge detection when changing the value of this bit, it must be changed when SMFSEL=0. 000: Internal trigger ITR0 001: Internal trigger ITR1 010: Internal trigger ITR2 011: Internal trigger ITR3 100: Channel 1 input edge detector TIF_ED 101: Channel 1 post-filtering timer input TI1FP1 110: Channel 2 post-filtering timer input TI2FP2 111: External trigger input (ETRF)
3	Reserved		
2:0	SMFSEL	R/W	Slave Mode Function Select 000: Disable the slave mode, the timer can be used as the master mode timer to affect the work of slave mode timer; if CTRL1_CNTEN=1, the prescaler is directly driven by the internal clock. 001: Encoder mode 1; according to the level of TI1FP1, the counter counts at the edge of TI2FP2. 010: Encoder mode 2; according to the level of TI2FP2, the counter counts at the edge of TI1FP1. 011: Encoder mode 3; according to the input level of the other signal, the counter counts at the edge of TI1FP1 and TI2FP2. 100: Reset mode; the slave mode timer resets the counter after receiving the rising edge signal of TRGI and generates the signal to update the register. 101: Gated mode; when the slave mode timer receives the TRGI high level signal, the counter will start to work; when it receives TRGI low level signal, the counter will stop working; when it receives TRGI high level signal again, the timer will continue to work; the counter is not reset during the whole period. 110: Trigger mode, the slave mode timer starts the counter to work after receiving the rising edge signal of TRGI.

Field	Name	R/W	Description
			111: External clock mode 1; select the rising edge signal of TRGI as the clock source to drive the counter to work.

15.6.4 DMA/Interrupt enable register (TMRx_DIEN)

Offset address: 0x0C

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:15	Reserved		
14	TRGDEN	R/W	Trigger DMA Request Enable 0: Disable 1: Enable
13	Reserved		
12	CC4DEN	R/W	Capture/Compare Channel4 DMA Request Enable 0: Disable 1: Enable
11	CC3DEN	R/W	Capture/Compare Channel3 DMA Request Enable 0: Disable 1: Enable
10	CC2DEN	R/W	Capture/Compare Channel2 DMA Request Enable 0: Disable 1: Enable
9	CC1DEN	R/W	Capture/Compare Channel1 DMA Request Enable 0: Disable 1: Enable
8	UDIEN	R/W	Update DMA Request Enable 0: Disable 1: Enable
7	Reserved		
6	TRGIEN	R/W	Trigger Interrupt Enable 0: Disable 1: Enable
5	Reserved		
4	CC4IEN	R/W	Capture/Compare Channel4 Interrupt Enable 0: Disable 1: Enable
3	CC3IEN	R/W	Capture/Compare Channel3 Interrupt Enable 0: Disable 1: Enable
2	CC2IEN	R/W	Capture/Compare Channel2 Interrupt Enable 0: Disable 1: Enable

Field	Name	R/W	Description
1	CC1IEN	R/W	Capture/Compare Channel1 Interrupt Enable 0: Disable 1: Enable
0	UIEN	R/W	Update Interrupt Enable 0: Disable 1: Enable

15.6.5 Status register (TMRx_STS)

Offset address: 0x10

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:13	Reserved		
12	CC4RCFLG	RC_W0	Captuer/compare Channel4 Repetition Capture Flag Refer to STS_CC1RCFLG
11	CC3RCFLG	RC_W0	Captuer/compare Channel3 Repetition Capture Flag Refer to STS_CC1RCFLG
10	CC2RCFLG	RC_W0	Captuer/compare Channel2 Repetition Capture Flag Refer to STS_CC1RCFLG
9	CC1RCFLG	RC_W0	Captuer/Compare Channel1 Repetition Capture Flag 0: Repeated capture does not occur 1: Repeated capture occurs The value of the counter is captured to TMRx_CC1 register, and CC1IFLG=1; this bit can be set to 1 by hardware and cleared to 0 by software only when the channel is configured as input capture.
8:7	Reserved		
6	TRGIFLG	RC_W0	Trigger Event Interrupt Generate Flag 0: No trigger event interrupt occurs 1: Trigger event interrupt occurs When a trigger event is generated, this bit is set to 1 by hardware and cleared to 0 by software.
5	Reserved		
4	CC4IFLG	RC_W0	Captuer/Compare Channel4 Interrupt Flag Refer to STS_CC1IFLG
3	CC3IFLG	RC_W0	Captuer/Compare Channel3 Interrupt Flag Refer to STS_CC1IFLG
2	CC2IFLG	RC_W0	Capture/Compare Channel2 new Interrupt Flag Refer to STS_CC1IFLG
1	CC1IFLG	RC_W0	Captuer/Compare Channel1 Interrupt Flag When the capture/compare channel 1 is configured as output: 0: No matching occurs 1: The value of TMRx_CNT matches the value of TMRx_CC1 When the capture/compare channel 1 is configured as input: 0: No input capture occurs 1: Input capture occurs

Field	Name	R/W	Description
			When a capture event occurs, it is set to 1 by hardware; it can be cleared to 0 by software or cleared to 0 when reading TMRx_CC1 register.
0	UIFLG	RC_W0	Update Event Interrupt Generate Flag 0: No update event interrupt occurs 1: Update event interrupt occurred When the counter value is reloaded or reinitialized, an update event will be generated. The bit is set to 1 by hardware and cleared to 0 by software; update events are generated in the following situations: (1) UD=0 on TMRx_CTRL1 register, and when overruns or underruns, an update event will be generated; (2) URSEL=0 and UD=0 on TMRx_CTRL1 register, configure UG = 1 on TMRx_CEG register to generate update event, and the counter needs to be initialized by software; (3) URSEL=0 and UD=0 on TMRx_CTRL1 register, generate update event when the counter is initialized by trigger event.

15.6.6 Control event generation register (TMRx_CEG)

Offset address: 0x14

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:7	Reserved		
6	TEG	W	Trigger Event Generate 0: Invalid 1: Generate trigger event This bit is set to 1 by software and cleared to 0 automatically by hardware.
5	Reserved		
4	CC4EG	W	Capture/Compare Channel4 Event Generation Refer to CC1EG description
3	CC3EG	W	Capture/Compare Channel3 Event Generation Refer to CC1EG description
2	CC2EG	W	Capture/Compare Channel2 Event Generation Refer to CC1EG description
1	CC1EG	W	Capture/Compare Channel1 Event Generation 0: Invalid 1: Generate capture/compare event This bit is set to 1 by software and cleared to 0 automatically by hardware. If Channel 1 is in output mode When CC1IFLG=1, if CC1IEN and CC1DEN bits are set, the corresponding interrupt and DMA request will be generated. If Channel 1 is in input mode The value of the capture counter is stored in TMRx_CC1 register; configure CC1IFLG=1, and if CC1IEN and CC1DEN bits are also set, the corresponding interrupt and DMA request will be generated; at this time, if CC1IFLG=1, it is required to configure CC1RCFLG=1.

Field	Name	R/W	Description
0	UEG	W	Update Event Generate 0: Invalid 1: Initialize the counter and generate an update event This bit is set to 1 by software, and cleared to 0 by hardware. Note: When an update event is generated, the counter of the prescaler will be cleared to 0, but the prescaler factor remains unchanged. In the count-down mode, the counter will read the value of TMRx_AUTORLD; in center-aligned mode or count-up mode, the counter will be cleared to 0.

15.6.7 Capture/Compare mode register 1 (TMRx_CCM1)

Offset address: 0x18

Reset value: 0x0000 0000

The timer can be configured as input (capture mode) or output (compare mode) by CCxSEL bit. The functions of other bits of the register are different in input and output modes, and the functions of the same bit are different in output mode and input mode. The OCX in the register describes the function of the channel in the output mode, and the Icx in the register describes the function of the channel in the input mode.

Output compare mode:

Field	Name	R/W	Description
31:16	Reserved		
15	OC2CEN	R/W	Output Compare Channel2 Clear Enable
14:12	OC2MOD	R/W	Output Compare Channel2 Mode
11	OC2PEN	R/W	Output Compare Channel2 Buffer Enable
10	OC2FEN	R/W	Output Compare Channel2 Preload Enable
9:8	CC2SEL	R/W	Capture/Compare Channel2 Select This bit defines the input/output direction and selects the input pin. 00: CC2 channel is output 01: CC2 channel is input, and IC2 is mapped on TI2 10: CC2 channel is input, and IC2 is mapped on TI1 11: CC2 channel is input, and IC2 is mapped on TRC, and only works in internal trigger input Note: This bit can be written only when the channel is closed (TMRx_CCEN register CC2EN=0).
7	OC1CEN	R/W	Output Compare Channel1 Clear Enable 0: OC1REF is unaffected by ETRF input. 1: When high level of ETRF input is detected, OC1REF=0
6:4	OC1MOD	R/W	Output Compare Channel1 Mode Configure 000: Freeze The output compare has no effect on OC1REF 001: The output value is high when matching. When the value of counter CNT matches the value CCx of capture/compare register, OC1REF will be forced to be high

Field	Name	R/W	Description
			010: The output value is low when matching. When the value of the counter matches the value of the capture/compare register, OC1REF will be forced to be low 011: Output flaps when matching. When the value of the counter matches the value of the capture/compare register, flap the level of OC1REF 100: The output is forced to be low. Force OC1REF to be low 101: The output is forced to be high. Force OC1REF to be high 110: PWM mode 1 (set to high when the counter value<output compare value; otherwise, set to low) 111: PWM mode 2 (set to high when the counter value>output compare value; otherwise, set to low) Note: When the protection level is 3 and the channel is configured as output, this bit cannot be modified. In PWM modes 1 and 2, the OC1REF level changes when the comparison result changes or when the output compare mode changes from freeze mode to PWM mode.
3	OC1PEN	R/W	Output Compare Channel1 Preload Enable 0: Disable preloading function; write the value of TMRx_CC1 register through the program and it will work immediately. 1: Enable preloading function; write the value of TMRx_CC1 register through the program and it will work after an update event is generated. Note: When the protection level is 3 and the channel is configured as output, this bit cannot be modified. When the preload register is uncertain, PWM mode can be used only in single-pulse mode (SPMEN=1); otherwise, the following output compare result is uncertain.
2	OC1FEN	R/W	Output Compare Channel1 Fast Enable 0: Disable 1: Enable This bit is used to improve the response of the capture/compare output to the trigger input event.
1:0	CC1SEL	R/W	Capture/Compare Channel 1 Selection This bit defines the input/output direction and selects the input pin. 00: CC1 channel is output 01: CC1 channel is input, and IC1 is mapped on TI1 10: CC1 channel is input, and IC1 is mapped on TI2 11: CC1 channel is input, and IC1 is mapped on TRC, and only works in internal trigger input Note: This bit can be written only when the channel is closed (TMRx_CCEN register CC1EN=0).

Input capture mode:

Field	Name	R/W	Description
31:16	Reserved		
15:12	IC2F	R/W	Input Capture Channel 2 Filter Configuration
11:10	IC2PSC	R/W	Input Capture Channel2 Perscaler Configuration
9:8	CC2SEL	R/W	Capture/Compare Channel 2 Select 00: CC2 channel is output 01: CC2 channel is input, and IC2 is mapped on TI2

Field	Name	R/W	Description
			10: CC2 channel is input, and IC2 is mapped on TI1 11: CC2 channel is input, and IC2 is mapped on TRC, and only works in internal trigger input Note: This bit can be written only when the channel is closed (TMRx_CCEN register CC2EN=0).
7:4	IC1F	R/W	Input Capture Channel 1 Filter Configuration 0000: Disable filter, sampled by f_{DTS} 0001：DIV=1，N=2 0010：DIV=1，N=4 0011：DIV=1，N=8 0100：DIV=2，N=6 0101：DIV=2，N=8 0110：DIV=4，N=6 0111：DIV=4，N=8 1000：DIV=8，N=6 1001：DIV=8，N=8 1010：DIV=16，N=5 1011：DIV=16，N=6 1100：DIV=16，N=8 1101：DIV=32，N=5 1110：DIV=32，N=6 1111：DIV=32，N=8 Sampling frequency=timer clock frequency/DIV; the filter length=N, indicating that a jump is generated by every N events.
3:2	IC1PSC	R/W	Input Capture Channel1 Perscaler Configure 00：PSC=1 01：PSC=2 10：PSC=4 11：PSC=8 PSC is prescaler factor; capture is triggered once by every PSC events.
1:0	CC1SEL	R/W	Capture/Compare Channel 1 Select 00: CC1 channel is output 01: CC1 channel is input, and IC1 is mapped on TI1 10: CC1 channel is input, and IC1 is mapped on TI2 11: CC1 channel is input, and IC1 is mapped on TRC, and only works in internal trigger input Note: This bit can be written only when the channel is disabled (TMRx_CCEN register CC1EN=0).

15.6.8 Capture/Compare mode register 2（TMRx_CCM2）

Offset address: 0x1C

Reset value: 0x0000 0000

Refer to the description of the above CCM1 register.

Output compare mode:

Field	Name	R/W	Description
31:16	Reserved		

Field	Name	R/W	Description
15	OC4CEN	R/W	Output Compare Channel4 Clear Enable
14:12	OC4MOD	R/W	Output Compare Channel4 Mode Configure
11	OC4PEN	R/W	Output Compare Channel4 Buffer Enable
10	OC4FEN	R/W	Output Compare Channel4 Preload Enable
9:8	CC4SEL	R/W	Capture/Compare Channel 4 Selection This bit defines the input/output direction and selects the input pin. 00: CC4 channel is output 01: CC4 channel is input, and IC4 is mapped on TI4 10: CC4 channel is input, and IC4 is mapped on TI3 11: CC4 channel is input, and IC4 is mapped on TRC, and only works in internal trigger input Note: This bit can be written only when the channel is closed (TMRx_CCEN register CC4EN=0).
7	OC3CEN	R/W	Output Compare Channel3 Clear Enable 0: OC3REF is unaffected by ETRF input. 1: When high level of ETRF input is detected, OC1REF=0
6:4	OC3MOD	R/W	Output Compare Channel3 Mode Configure)
3	OC3PEN	R/W	Output Compare Channel3 Preload Enable
2	OC3FEN	R/W	Output Compare Channel3 Fast Enable 0: Disable 1: Enable This bit is used to improve the response of the capture/compare output to the trigger input event.
1:0	CC3SEL	R/W	Capture/Compare Channel 3 Selection This bit defines the input/output direction and selects the input pin. 00: CC3 channel is output 01: CC3 channel is input, and IC3 is mapped on TI3 10: CC3 channel is input, and IC3 is mapped on TI4 11: CC3 channel is input, and IC3 is mapped on TRC, and only works in internal trigger input Note: This bit can be written only when the channel is closed (TMRx_CCEN register CC3EN=0).

Input capture mode:

Field	Name	R/W	Description
31:16	Reserved		
15:12	IC4F	R/W	Input Capture Channel 4 Filter Configuration
11:10	IC4PSC	R/W	Input Capture Channel4 Perscaler Configuration
9:8	CC4SEL	R/W	Capture/Compare Channel 4 Select 00: CC4 channel is output 01: CC4 channel is input, and IC4 is mapped on TI4 10: CC4 channel is input, and IC4 is mapped on TI3 11: CC4 channel is input, and IC4 is mapped on TRC, and only works in internal trigger input

Field	Name	R/W	Description
			Note: This bit can be written only when the channel is closed (TMRx_CCEN register CC4EN=0).
7:4	IC3F	R/W	Input Capture Channel 3 Filter Configuration
3:2	IC3PSC	R/W	Input Capture Channel 3 Perscaler Configuration 00：PSC=1 01：PSC=2 10：PSC=4 11：PSC=8 PSC is prescaler factor; capture is triggered once by every PSC events.
1:0	CC3SEL	R/W	Capture/Compare Channel 3 Select 00: CC3 channel is output 01: CC3 channel is input, and IC3 is mapped on TI3 10: CC3 channel is input, and IC3 is mapped on TI4 11: CC3 channel is input, and IC3 is mapped on TRC, and only works in internal trigger input Note: This bit can be written only when the channel is closed (TMRx_CCEN register CC3EN=0).

15.6.9 Enable capture/compare channel register (TMRx_CCEN)

Offset address: 0x20

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15	CC4NPOL	R/W	Capture/Compare Channel4 Output Polarity Configure Refer to CCEN_CC1NPOL
14	Reserved		
13	CC4POL	R/W	Capture/Compare Channel4 Output Polarity Refer to CCEN_CC1POL
12	CC4EN	R/W	Capture/Compare Channel4 Output Enable Refer to CCEN_CC1EN
11	CC3NPOL	R/W	Capture/Compare Channel3 Output Polarity Configure Refer to CCEN_CC1NPOL
10	Reserved		
9	CC3POL	R/W	Capture/Compare Channel3 Output Polarity Configure Refer to CCEN_CC1POL
8	CC3EN	R/W	Capture/Compare Channel3 Output Enable Refer to CCEN_CC1EN
7	CC2NPOL	R/W	Capture/Compare Channel2 Output Polarity Configure Refer to CCEN_CC1NPOL
6	Reserved		
5	CC2POL	R/W	Capture/Compare Channel2 Output Polarity Configure Refer to CCEN_CC1POL
4	CC2EN	R/W	Capture/Compare Channel2 Output Enable

Field	Name	R/W	Description
			Refer to CCEN_CC1EN
3	CC1NPOL	R/W	Capture/Compare Channel1 Output Polarity Configure When CC1 channel is configured as output: CC1NPOL remains in cleared state all the time When CC1 channel is configured as input: This bit and CC1POL control the polarity of the triggered or captured signals TI1FP1 and TI2FP1 at the same time.
2	Reserved		
1	CC1POL	R/W	Capture/Compare Channel1 Output Polarity Configure When CC1 channel is configured as output: 0: OC1 is active high 1: OC1 is active low When CC1 channel is configured as input: 0: Phase not reversed: capture at the rising edge of IC1; phase not reversed when IC1 is used as external trigger. 1: Phase reversed, capture at the falling edge of ICC1; phase reversed when IC1 is used as external trigger. Note: When the protection level is 2 or 3, this bit cannot be modified
0	CC1EN	R/W	Capture/Compare Channel1 Output Enable When the capture/compare channel 1 is configured as output: 0: Disable output 1: Enable output When the capture/compare channel 1 is configured as input: This bit determines whether the value CNT of the counter can be captured and enter TMRx_CC1 register 0: Disable capture 1: Enable capture

15.6.10 Counter register (TMRx_CNT)

Offset address: 0x24

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15:0	CNT	R/W	Counter Value

15.6.11 Prescaler register (TMRx_PSC)

Offset address: 0x28

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15:0	PSC	R/W	Prescaler Value Clock frequency of counter (CK_CNT)=f_{CK_PSC}/(PSC+1)

15.6.12 Auto reload register (TMRx_AUTORLD)

Offset address: 0x2C

Reset value: 0x0000 FFFF

Field	Name	R/W	Description
31:16	Reserved		
15:0	AUTORLD	R/W	Auto Reload Value When the value of auto reload is empty, the counter will not count.

15.6.13 Channel 1 capture/compare register (TMRx_CC1)

Offset address: 0x34

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15:0	CC1	R/W	Capture/Compare Channel 1 Value When the capture/compare channel 1 is configured as input mode: CC1 contains the counter value transmitted by the last input capture channel 1 event. When the capture/compare channel 1 is configured as output mode: CC1 contains the value currently loaded in the capture/compare register Compare the value CC1 of the capture and compare channel 1 with the value CNT of the counter to generate the output signal on OC1. When the output compare preload is disabled (OC1PEN=0 for TMRx_CCM1 register), the written value will immediately affect the output comparison results; If the output compare preload is enabled (OC1PEN=1 for TMRx_CCM1 register), the written value will affect the output compare result when an update event is generated.

15.6.14 Channel 2 capture/compare register (TMRx_CC2)

Offset address: 0x38

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15:0	CC2	R/W	Capture/Compare Channel 2 Value Refer to TMRx_CC1

15.6.15 Channel 3 capture/compare register (TMRx_CC3)

Offset address: 0x3C

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15:0	CC3	R/W	Capture/Compare Channel 3 Value Refer to TMRx_CC1

15.6.16 Channel 4 capture/compare register (TMRx_CC4)

Offset address: 0x40

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15:0	CC4	R/W	Capture/Compare Channel 4 Value Refer to TMRx_CC1

15.6.17 DMA control register (TMRx_DCTRL)

Offset address: 0x48

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:13	Reserved		
12:8	DBLEN	R/W	DMA Burst Transfer Length Setup These bits define the transmission length and transmission times of DMA in continuous mode. The data transmitted can be 16 bits and 8 bits. When reading/writing TMRx_DMADDR register, the timer will conduct a continuous transmission; 00000: Transmission once 00001: Transmission twice 00010: Transmission for three times …… 10001: Transmission for 18 times The transmission address formula is as follows: Transmission address=TMRx_CTRL1 address (slave address) +DBADDR+DMA index; DMA index=DBLEN For example: DBLEN=7, DBADDR=TMR2_CTRL1 (slave address) means the address of the data to be transmitted, while the address +DBADDR+7 of TMRx_CTRL1 means the address of the data to be written/read, Data transmission will occur to: TMRx_CTRL1 address + seven registers starting from DBADDR. The data transmission will change according to different DMA data length: When the transmission data is set to 16 bits, the data will be transmitted to seven registers When the transmission data is set to 8 bits, the data of the first register is the MSB bit of the first data, the data of the second register is the LSB bit of the first data, and the data will still be transmitted to seven registers.
7:5	Reserved		
4:0	DBADDR	R/W	DMA Base Address Setup These bits define the base address of DMA in continuous mode (when reading or writing TMRx_DMADDR register), and DBADDR is defined as the offset from the address of TMRx_CTRL1 register: 00000：TMRx_CTRL1 00001：TMRx_CTRL2 00010：TMRx_SMCTRL …….

15.6.18 DMA address register of continuous mode (TMRx_DMADDR)

Offset address: 0x4C

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15:0	DMADDR	R/W	DMA Register for Burst Transfer Read or write operation access of TMRx_DMADDR register may lead to access to the register in the following address: TMRx_CTRL1 address + (DBADDR+DMA index) ×4 Where: "TMRx_CTRL1 address" is the address of control register 1 (TMRx_CTRL1); "DBADDR" is the base address defined in TMRx_DCTRL register; "DMA index" is the offset automatically controlled by DMA, and it depends on DBLEN defined in TMRx_DCTRL register.

16 Low-power Timer (LPTMR)

16.1 Introduction

The low-power timer is a 16-bit timer that uses an internal low-frequency clock to drive the counter. It can maintain counting function and generate interrupts in low-power modes, thereby waking up the system from the low-power mode.

16.2 Main characteristics

- 16-bit count-up counter
- 16-bit auto-reload function period register
- 4-bit programmable prescaler can realize 16 different division factors (1, 2, 4, 8...32768)
- Independent internal low-frequency clock LSICLK serves as the driving clock for the counter.
- When an update event occurs, it can be configured to generate an update interrupt.

16.3 Functional description

16.3.1 Clock

The low-power timer module has two clocks: the APB clock and the internal low-frequency clock LSICLK. The APB clock serves as the read/write clock for the registers, while LSICLK serves as the drive clock for the counter.

16.3.2 Timebase unit

The timebase unit in the low-power timer contains three registers:

- 16-bit counter
- 16-bit auto-reload register (LPTMR_WVR)
- 4-bit programmable prescaler counter

The 16-bit counter is driven by the internal low-frequency clock LSICLK. When the EN bit in the LPTMR_CR register is enabled, the counter starts counting up from its initial value. When the count value equals the period register value, an update event is generated, and the WK_STS bit in the LPTMR_SR register is set to 1. If the IREN bit in the LPTMR_CR register is 1, an interrupt is generated.

Prescaler

The low-power timer has a 4-bit programmable prescaler, which can realize 16 different division factors, namely 1, 2, 4, 8...32768. The prescaler can be modified while the timer is running. The modification takes effect upon the

occurrence of an update event.

Period register

The period register has an internal buffer register. What actually takes effect while the timer is running is the value of the buffer register. When an update event occurs, the value of the period register will be updated into the internal buffer register.

16.4 Register address mapping

In the following table, all registers of the low-power timer are mapped to a 16-bit addressable (address) space.

Table 52 LPTMR Register Address Mapping

Register name	Description	Offset address
LPTMRx_CR	Control register	0x00
LPTMRx_PSC	Prescale register	0x04
LPTMRx_WVR	Period register	0x08
LPTMRx_SR	Status register	0x0C

16.5 Register functional description

16.5.1 Control register (LPTMR_CR)

Offset address: 0x00

Reset value: 0x0000 0000

Field	Name	Description
31:2	Reserved	
1	IREN	Interrupt enable 0: Disable interrupt 1: Enable interrupt
0	EN	Module enable 0: Disable the module 1: Enable the module

16.5.2 Prescaler register (LPTMR_PSC)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	Description
31:4	Reserved	
3:0	PS	Prescaler value 0000: No frequency division 0001: Divide by 2 0010: Divide by 4 0011: Divide by 8 0100: Divide by 16

Field	Name	Description
		0101: Divide by 32 0110: Divide by 64 0111: Divide by 128 1000: Divide by 256 1001: Divide by 512 1010: Divide by 1024 1011: Divide by 2048 1100: Divide by 4096 1101: Divide by 8192 1110: Divide by 16384 1111: Divide by 32768

16.5.3 Period register (LPTMR_WVR)

Offset address: 0x08

Reset value: 0x0000 FFFF

Field	Name	Description
31:16	Reserved	
15:0	RV	Cycle value

16.5.4 Status register (LPTMR_SR)

Offset address: 0x0C

Reset value: 0x0000 0000

Field	Name	Description
31:3	Reserved	
2	RU	Reload value update flag This bit is set to 1 by hardware after software writes to LPTMR_WVR, and is automatically cleared to 0 by hardware after the write value update is completed. Note: The update time is 3 LSICLK cycles.
1	PU	Pre-division value update flag This bit is set to 1 by hardware after software writes to LPTMR_PSC, and is automatically cleared to 0 by hardware after the write value update is completed. Note: The update time is 3 LSICLK cycles.
0	WK_STS	Interrupt status register 0: No update event occurs 1: Update event occurs Write 0 to clear this bit.

17 Watchdog timer (WDT)

17.1 Introduction

The watchdog is used to monitor system faults caused by software errors. There are two watchdog devices on the chip: independent watchdog and window watchdog, which improve the security, and make the time more accurate and the use more flexible.

The independent watchdog will reset only when the counter is reduced to 0, and the value of refresh counter will not be reset until it is not reduced to 0.

The window watchdog will reset when the counter decreases to 0x3F. When the count value of the counter is before the window value of the configuration register, the counter will be reset after refresh.

Table 53 Comparison between IWDT and WWDT

Name	Counter Resolution	Counter Type	Prescaler Coefficient	Function Description
Independent Watchdog (IWDT)	12-bit	Count down	Any integer between 1~256	The clock is provided by an independent 32KHz RC oscillator. As the RC oscillator is independent of the main clock, the IWDT can operate in stop mode.
				In case of a fault, it can reset the entire system.
				It can serve as a free timer to provide timeout management for applications.
				It can be configured via option bytes to be started by software or hardware.
				In debug mode, the counter can be frozen.
Window Watchdog (WWDT)	7-bit	Count down	-	Driven by the main clock, it features an early warning interrupt function.
				In case of a problem, it can reset the entire system.
				It can be configured to run freely.
				In debug mode, the counter can be frozen.

17.2 Independent watchdog timer (IWDT)

17.2.1 Introduction

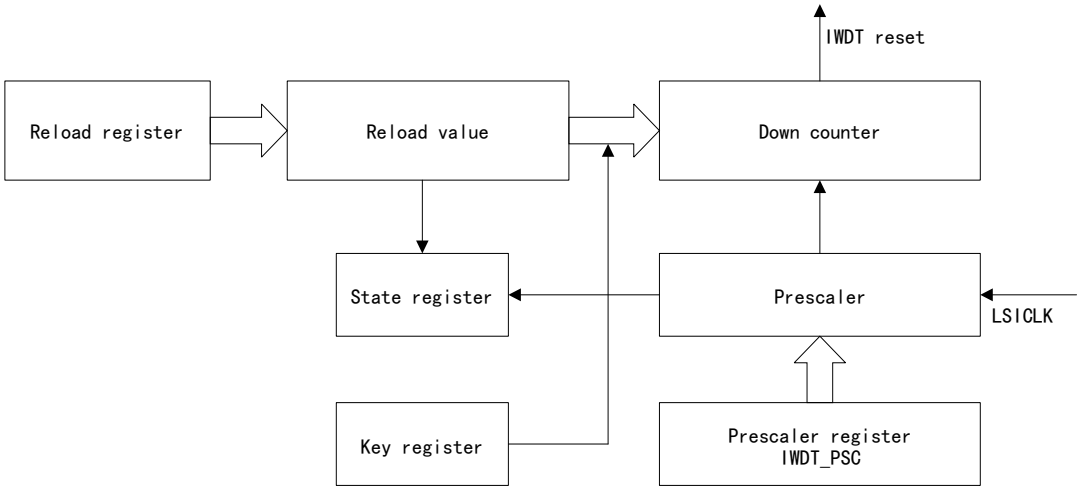
The independent watchdog consists of an 8-bit prescaler IWDT_PSC, 12-bit count-down counter, 12-bit reload register IWDT_CNTRLD, key register IWDT_KEY and state register IWDT_STS.

The independent watchdog has an independent clock source, and even if the master clock fails, it is still valid.

The independent watchdog is applicable when an independent environment is required but the accuracy requirement is not high.

17.2.2 Structure block diagram

Figure 58 Independent Watchdog Structure Block Diagram



Note: The watchdog function is in the V_{DD} power supply area and can work normally in stop or standby mode.

17.2.3 Functional Description

17.2.3.1 **Key register**

Write 0xCCCC in the key register to enable the independent watchdog, then the counter starts to count down, and when the counter counts to 0x000, a reset will be generated.

Write 0xAAAA in the key register, and the value of the reload register will be reloaded to the counter to prevent the watchdog from resetting.

Write 0x5555 to the key register to rewrite the value of the prescaler register and the reload register.

17.2.3.2 **Register access protection**

The prescaler register and reload register have write protection function. To rewrite these two registers, it is necessary to write 0X5555 to the key register. If other value is written to the key register, the protection of the register will be enabled again.

Write 0xAAAA to the key register and the write protection function will also be enabled.

17.2.3.3 Hardware watchdog

After the "hardware watchdog" function is enabled, and the system is powered on and reset, the watchdog will run automatically. If 0xAAAA is not written to the key register, reset will be generated after the counter finishes counting.

17.2.3.4 Debug mode

The independent watchdog can be configured in debug mode and choose to stop or continue to work. This depends on the DBG_IWDT_STOP bit of DBGMCU_APBF register.

Table 54 Minimum/Maximum timeout value when LSICLK=32kHz

PSC	Minimum timeout value	Maximum timeout value
0	0.125ms	512ms
1	0.25ms	1024ms
2	0.5ms	2048ms
3	1ms	4096ms
4	2ms	8192ms
5	4ms	16384ms
6	-	32768ms

17.3 Window watchdog timer (WWDT)

17.3.1 Introduction

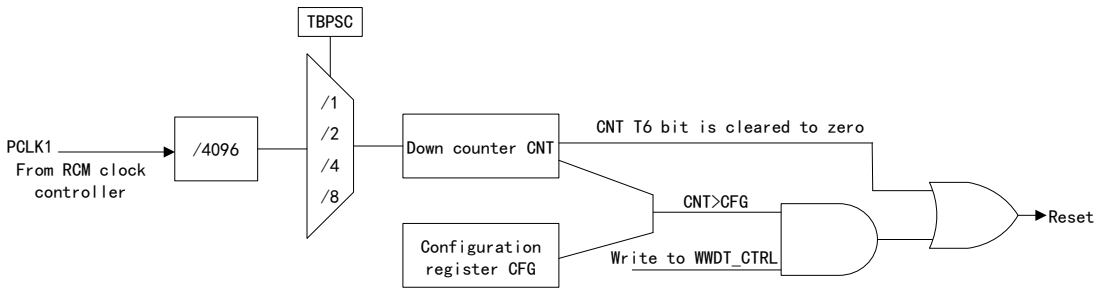
The window watchdog contains a 7-bit free-running down counter, prescaler and control register WWDT_CTRL, configuration register WWDT_CFG and status register WWDT_STS.

The window watchdog clock comes from PCLK1, and the counter clock is obtained from the CK counter clock through frequency division by prescaler (configured by the configuration register).

The window watchdog is applicable when precise timing is needed.

17.3.2 Structure block diagram

Figure 59 Window Watchdog Structure Block Diagram



17.3.3 Functional Description

Enable window watchdog timer, and the reset conditions are:

- When the counter count is less than 0x40, a reset will be generated.
- The reload counter will be reset before the counter counts to the value of the window register.

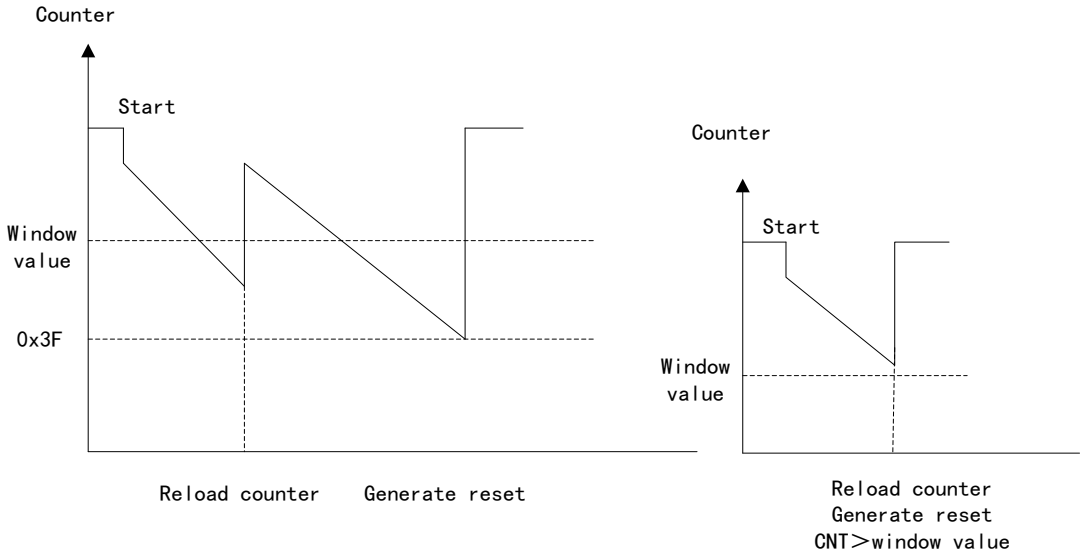
After reset, the watchdog is always closed and the watchdog can be enabled only by setting the WWDTEN bit of WWDT_CTRL control register.

The counter of window watchdog is in free state. When the watchdog is disabled, the counter will continue to count down. The counter must be reloaded between the value of window register and 0x40 to avoid resetting.

Setting the EWIEN bit of the WWDT_CFG configuration register can enable the early wake-up interrupt. When the count reaches 0x40, an interrupt will be generated. Enter the interrupt service program (ISR) to prevent the window watchdog from resetting. EWIEN interrupt can be cleared by writing 0 in the status register.

The unique window of the window watchdog timer can effectively monitor whether the program is faulty. For example, assuming that the running time of a program segment is T, the value of the window register is set to slightly less than (TR-T), if there is no reload register in the window, it means that the program is faulty, and when the counter counts to 0x3F, a reset will be generated.

Figure 60 Window Watchdog Timing Diagram



The calculation formula of window watchdog timer timeout is as follows:

$$T_{WWDT}=T_{PCLK1}\times 4096\times 2^{TBPSC}\times（CNT[5:0]+1）$$

Where:

- T_{WWDT}: WWDT timeout
- T_{PCLK1}: Clock cycle of APB1 (in ms)

Minimum/Maximum timeout when PCLK1=30MHz:

Table 55 Minimum/Maximum timeout value when PCLK1=30MHz

TBPSC	Minimum timeout value	Maximum timeout value
0	136.53μs	8.74ms
1	273.07μs	17.48ms
2	546.13μs	34.95ms
3	1092.27μs	69.91ms

17.4 IWDT register address mapping

Table 56 IWDT Register Address Mapping

Register name	Description	Offset Address
IWDT_KEY	Key register	0x00
IWDT_PSC	Prescaler register	0x04
IWDT_CNTRLD	Counter reload register	0x08
IWDT_STS	Status register	0x0C

17.5 IWDT register functional description

These peripheral registers can be operated by half word (16 bits) or word (32 bits).

17.5.1 Key register (IWDT_KEY)

Offset address: 0x00

Reset value: 0x0000 0000 (reset in standby mode)

Field	Name	R/W	Description
31:16	Reserved		
15:0	KEY	W	Allow Access IWDT Register Key Value Writing 0x5555 means enabled access to IWDT_PSC and IWDT_CNTRLD registers; When the software writes 0xAAAA, it means to execute the reload counter, and a certain interval is required to prevent the watchdog from resetting. Write 0xCCCC and the watchdog will be enabled (the hardware watchdog is unrestricted by this command word); This register is write-only and the read-out value is 0x0000.

17.5.2 Prescaler register (IWDT_PSC)

Offset address: 0x04

Reset value: 0x0000 0007

Field	Name	R/W	Description
31:3	Reserved		
2:0	PSC	R/W	Prescaler Factor Configure Support the write protection function; when writing 0x5555 to the IWDT_KEY register, it is allowed to access the register; in the process of writing to this register, only when PSCUFLG=0 for IWDT_STS register, can the prescaler factor be changed; in the process of reading this register, only when PSCUFLG=0, can the read-out value of PSC register be valid. 000：PSC=4 001：PSC=8 010：PSC=16 011：PSC=32 100：PSC=64 101：PSC=128 110：PSC=256 111：PSC=256

17.5.3 Counter reload register (IWDT_CNTRLD)

Offset address: 0x08

Reset value: 0x0000 0FFF (reset in standby mode)

Field	Name	R/W	Description
31:12	Reserved		
11:0	CNTRLD	R/W	Watchdog Counter Reload Value Setup It supports write protection function and defines the value loaded to the watchdog counter when 0xAAAA is written to IWDT_KEY register; in the process of writing this register, this register can be modified only when CNTUFLG=0. In the process of reading this register, only when CNTUFLG=0 in IWDT_STS register, can the read value be valid. The watchdog timeout cycle can be calculated by the reload value and clock prescaler value.

17.5.4 Status register (IWDT_STS)

Offset address: 0x0C

Reset value: 0x0000 0000 (not reset in standby mode)

Field	Name	R/W	Description
31:2	Reserved		
1	CNTUFLG	R	Watchdog Counter Reload Value Update Flag When the counter reload value is updated, it will be set to 1 by hardware; after the counter reload value is updated, it will be cleared to 0 by hardware; the counter reload value is updated only when the CNTUFLG bit is cleared to 0.
0	PSCUFLG	R	Watchdog Prescaler Factor Update Flag When the prescaler factor is updated, it will be set to 1 by hardware; after the prescaler factor is updated, it will be cleared to 0 by hardware; the prescaler factor is updated only when the PSCUFLG bit is cleared to 0.

17.6 WWDT register address mapping

Table 57 WWDT Register Address Mapping

Register name	Description	Offset Address
WWDT_CTRL	Control register	0x00
WWDT_CFG	Configuration Register	0x04
WWDT_STS	Status register	0x08

17.7 WWDT register functional description

These peripheral registers can be operated by half word (16 bits) or word (32 bits).

17.7.1 Control register (WWDT_CTRL)

Offset address: 0x00

Reset value: 0x0000 007F

Field	Name	R/W	Description
31:8	Reserved		
7	WWDTEN	R/S	Window Watchdog Enable This bit is set to 1 by software and can be cleared by hardware only after reset. When WWDTEN=1, WWDT can generate a reset. 0: Disable 1: Enable
6:0	CNT	R/W	Counter Value Setup This counter is 7 bits, and CNT6 is the most significant bit These bits are used to store the counter value of the watchdog. When the count value decreases from 0x40 to 0x3F, WWDT reset will be generated.

17.7.2 Configuration register (WWDT_CFG)

Offset address: 0x04

Reset value: 0x0000 01FF

Field	Name	R/W	Description
31:10	Reserved		
9	EWIEN	R/S	Early Wakeup Interrupt Enable 0: No effect 1: When the counter value reaches 0x40, an interrupt will be generated; this interrupt is cleared by hardware after reset.
8:7	TBPSC	R/W	Timer Base Prescaler Factor Configure Divide the frequency on the basis of PCLK1/4096 00: No frequency division 01: 2 divided frequency 10: 4 divided frequency 11: 8 divided frequency

Field	Name	R/W	Description
6:0	WIN	R/W	Window Value Setup This window value is 7 bits, which is used to compare with the down counter.

17.7.3 Status register (WWDT_STS)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:1	Reserved		
0	EWIFLG	RC_W0	Early Wakeup Interrupt Occur Flag 0: Not occur 1: When the counter value reaches 0x40, it is set to 1 by hardware; if the interrupt is not enabled, the bit will also be set to 1; it can be cleared by writing 0 by software.

18 Real-time clock (RTC)

18.1 Full Name and Abbreviation of Terms

Table 58 Full Name and Abbreviation Description of Terms

Full name in English	English abbreviation
Second	SEC
Alarm	ALR
Overflow	OVR
Prescaler	PSC
Time Basic Clock	TBCLK

18.2 Introduction

The Real-Time Clock (RTC) is an independently operating timing module in the chip, which can provide precise date, time and alarm functions. Its core components include the time base unit, frequency divider and counter. Through configuration, multiple interrupts can be set to be generated, and it also supports low-power wake-up function.

18.3 Main characteristics

(1) Timebase unit

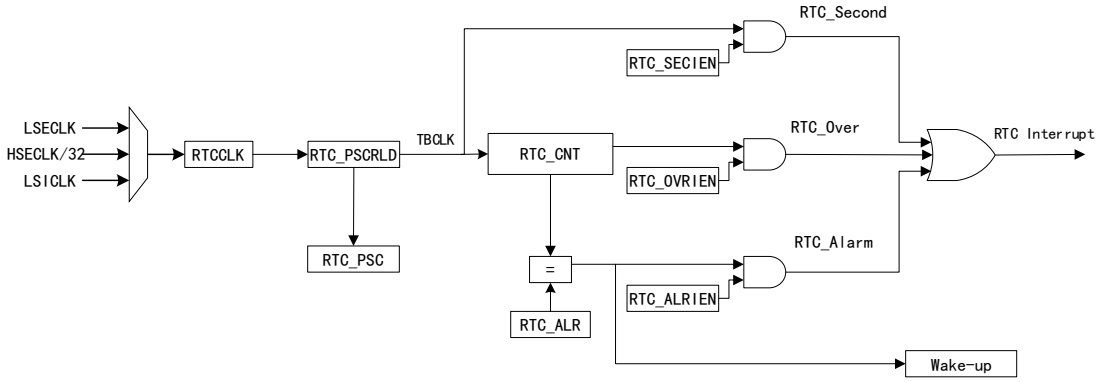
(2) A programmable 32-bit counter

(3) Multiple interrupt control

(4) Automatic wakeup from low-power mode

18.4 Structure block diagram

Figure 61 RTC Structure Block Diagram



18.5 Functional Description

18.5.1 Timebase unit

Clock Source

RTC has three clock sources:

- External LSECLK crystal oscillator
- External HSECLK crystal oscillator divided by 32
- Internal LSICLK (32kHz)

Select different clock sources by configuring the register of the clock controller RCM.

Prescaler

The RTC prescaler contains a 20-bit programmable divider, which can generate an RTC time base with a maximum period of 1 second.

18.5.2 RTC Register Configuration

To prevent unexpected counting errors caused by accidental writing to RTC registers, the RTC adopts a write protection mechanism. Only after disabling the write protection can registers with write protection features be operated on.

When configuring the RTC clock, the BKPWP bit in the power control register (RCM_PWRCR) must be set to "1" before configuration. To put the RTC into configuration mode, set the CFGMFLG bit in the RTC_CSTS register. Afterwards, can configure the RTC_PSCRLD, RTC_CNT, and RTC_ALR registers. After configuration, clear the CFGMFLG bit in RTC_CSTS to exit configuration mode.

Any write operation to RTC registers must occur after the previous write

operation has completed, which can be verified by checking the OCFLG flag in RTC_CSTS.

18.5.3 Programmable alarm

As a real-time clock, the RTC integrates an alarm function, which mainly operates through the alarm register and the counter, by configuring the alarm time via the RTC_ALR register.

After the alarm function is enabled, and when the counter value equals the alarm value, an alarm flag is set. If the alarm interrupt is enabled, an interrupt is triggered. By configuring external line 17 interrupt, the RTC alarm can be used to wake up the system from low-power consumption mode.

18.5.4 Interrupts

The RTC can generate seconds interrupts, alarm interrupts, and overflow interrupts. When the 20-bit prescaler overflows, an alarm event occurs, or the 32-bit counter overflows, the corresponding status flags are set. By configuring the RTC_CTRL register, the respective interrupts can be enabled.

18.6 Register address mapping

Table 59 RTC Register Address Mapping

Register name	Description	Offset Address
RTC_CTRL	Control register	0x00
RTC_CSTS	Control/Status register	0x04
RTC_PSCRLDH	Prescaler reload high register	0x08
RTC_PSCRLDL	Prescaler reload low register	0x0C
RTC_DIVH	Prescaler remainder high register	0x10
RTC_DIVL	Prescaler remainder low register	0x14
RTC_CNTH	Counter high register	0x18
RTC_CNTL	Counter low register	0x1C
RTC_ALRH	Alarm high register	0x20
RTC_ALRL	Alarm low register	0x24

18.7 Register functional description

18.7.1 Control register (RTC_CTRL)

Offset address: 0x00

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:3	Reserved		
2	OVRIEN	R/W	Overflow Interrupt Enable 0: Disable 1: Enable
1	ALRIEN	R/W	Alarm Interrupt Enable 0: Disable 1: Enable
0	SECIEN	R/W	Second Interrupt Enable 0: Disable 1: Enable

18.7.2 Control/Status register (RTC_CSTS)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:6	Reserved		
5	OCFLG	RC_W0	RTC Operation Complete Flag Indicates the status of the last write to the RTC register. 0: Not completed, unable to perform the next write operation 1: Completed, ready for the next write operation
4	CFGMFLG	RC_W0	Configure Mode Enable Flag Writing 1 by software enters configuration mode, allowing write operations to the RTC_CNT, RTC_ALR, or RTC_PSCRLD registers; writing 0 exits configuration mode. 0: Exit configuration mode (start updating RTC registers) 1: Enable configuration mode
3	RSYNCFLG	RC_W0	Registers Synchronized Flag When the RTC_CNT, RTC_PSCRLD, and RTC_ALR registers have been synchronized, it is set to 1 by hardware; it can only be cleared to 0 via software. After APB clock reset or stop, this bit must be cleared to 0 by software. User programs need to wait until hardware sets it to 1 before correctly reading the values of RTC_CNT, RTC_PSCRLD, and RTC_ALR. 0: Not synchronized 1: Synchronized
2	OVRFLG	RC_W0	Overflow Occur Flag When the counter overflows, it is set to 1 by hardware; it can only be cleared to 0 via software. 0: No overflow 1: 32-bit programmable counter overflow
1	ALRFLG	RC_W0	Alarm Occur Flag When the counter reaches the RTC_ALR value, it is set to 1 by hardware; it can only be cleared to 0 via software. 0: No alarm 1: Alarm is active

Field	Name	R/W	Description
0	SECFLG	RC_W0	Second Signal Condition Met Flag This flag provides a periodic signal (typically 1 second) for the RTC counter. When the 32-bit programmable prescaler overflows, it is set to 1 by hardware, and the RTC counter increments by 1. It can only be cleared to 0 via software. 0: No second flag 1: Second flag is set

18.7.3 Prescaler reload high register (RTC_PSCRLDH)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:4	Reserved		
3:0	PSCRLDH	W	RTC Prescaler Reload Value High Setup According to the following formula, these bits are used to define the base clock frequency: f_{TBCLK}=f_{RTCCLK}/（RLD [19:0]+1）

18.7.4 Prescaler reload low register (RTC_PSCRLDL)

Offset address: 0x0C

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15:0	PSCRLDL	W	RTC Prescaler Reload Value Low Setup According to the following formula, these bits are used to define the base clock frequency: f_{TBCLK}=f_{RTCCLK}/（RLD [19:0]+1）

18.7.5 Prescaler remainder high register (RTC_DIVH)

Offset address: 0x10

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:4	Reserved		
3:0	DIVH	R	RTC Clock Prescaler Remainder High Setup

18.7.6 Prescaler remainder low register (RTC_DIVL)

Offset address: 0x14

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15:0	DIVL	R	RTC Clock Prescaler Remainder Low Setup

18.7.7 Counter high register (RTC_CNTH)

Offset address: 0x18

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:4	Reserved		
3:0	CNTH	R/W	RTC Counter High Setup

18.7.8 Counter low register (RTC_CNTL)

Offset address: 0x1C

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15:0	CNTL	R/W	RTC Counter Low Setup

18.7.9 Alarm register (RTC_ALR)

Write operations are allowed when the OCFLG value is 1.

Alarm high register (RTC_ALRH)

Offset address: 0x20

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15:0	ALRH	W	RTC Alarm Value High Setup

Alarm low register (RTC_ALRL)

Offset address: 0x24

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15:0	ALRL	W	RTC Alarm Value Low Setup

19 Universal synchronous/asynchronous transceiver (USART)

19.1 Full Name and Abbreviation of Terms

Table 60 Full Name and Abbreviation Description of Terms

Full name in English	English abbreviation
Clear to Send	CTS
Request to Send	RTS
Most Significant Bit	MSB
Least Significant Bit	LSB
Guard	GRD
Overrun	OVR

19.2 Introduction

USART (universal synchronous/asynchronous receiver transmitter) is a serial communication device that can flexibly exchange full-duplex and half-duplex data with external devices, and meets the requirements of external devices for industry standard NRZ asynchronous serial data format. USART also provides a wide range of baud rate and supports multiprocessor communication.

USART not only supports standard asynchronous transceiver mode, but also supports some other serial data exchange modes, such as LIN protocol, smart card protocol, IrDA SIR ENDEC specification and hardware flow control mode.

USART also supports DMA function to realize high-speed data communication.

19.3 Main characteristics

(1) Full-duplex asynchronous communication

(2) Single-line half-duplex communication

(3) NRZ standard format

(4) Communication rate: 4800 / 9600 / 115200 / 2.5M / 4M / 5M / 8M / 10 Msps

(5) Characteristics of programmable serial port:
- Data bit: 8 or 9 bits
- Check bits: Even parity check, odd parity check, no check
- Support 0.5, 1, 1.5 and 2 stop bits

(6) Parity control
- Transmit the parity bit
- Check the received data

(7) Status flag bit:
- Transmission detection flag: The transmit register is empty, the receive register is not empty, and transmission is completed
- Error detection flag: Overrun error, noise error, parity error, frame error

(8) Select speed and clock tolerance with programmable 8 or 16-time oversampling rate

(9) Independent transmitter and receiver enable bit

(10) Configure the baud rate generator using a 16-bit frequency division coefficient

(11) DMA can be used for continuous communication

(12) Multiprocessor communication:
- If the address does not match, enter the mute mode
- Wake up from mute mode through idle bus detection or address flag detection

(13) Synchronous transmission mode

(14) Generation and detection of LIN break frame

(15) Support smart card interface of ISO7816-3 standard

(16) Support IrDA protocol

(17) Support hardware flow control

(18) Support automatic control of RS485 transmit enable I/O

(19) Multiple interrupt sources:
- The transmit register is empty
- Transmission Completed
- CTS changed
- The receive register is not empty
- Overrun error
- Bus idle
- Parity check error
- LIN break detection
- Noise error
- Overrun error
- Frame error

19.4 Functional Description

Table 61 USART Pin Description

Pin	Type	Description
USART_RX	Input	Data receiving
USART_TX	Output I/O (single-line mode/smart card mode)	Data transmission When the transmitter is enabled and does not transmit data, the default is high
USART_CK	Output	Clock output
USART_nRTS	Input	Request to transmit in hardware flow control mode
USART_nCTS	Output	Clear to send in hardware flow control mode
IrDA_RDI	Input	Data input in IrDA mode
IrDA_TDO	Output	Data output in IrDA mode

19.4.1 Single-line half-duplex communication

HDEN bit of USART_CTRL3 register determines whether to enter the single-line half-duplex mode.

When USART enters single-line half-duplex mode:

- The CLKEN and LINMEN bits of USART_CTRL2 register, and IREN and SCEN bits of USART_CTRL3 register must be cleared to 0.
- RX pin is disabled.
- TX pin should be configured as open-drain output and connected with RX pin inside the chip.
- Transmitting data and receiving data can not be carried out at the same time. The data cannot be received before they are transmitted. If needing to receive data, enable receiving can be turned on only after TXCFLG bit of USART_STS register is set to 1.
- If there is data collision on the bus, software is required to manage the distributed communication process.

19.4.2 Frame format

The frame format of data frame is controlled by USART_CTRL1 register

- DBLCFG bit controls the character length, which can be set to 8 or 9 bits.
- PCEN bit controls whether to enable the check bit or not.
- PCFG bit controls whether the parity bit is odd or even.

Table 62 Frame Format

DBLCFG bit	PCEN bit	USART data frame
0	0	Start bit+8-bit data+stop bit

DBLCFG bit	PCEN bit	USART data frame
0	1	Start bit+7-bit data+parity check bit+stop bit
1	0	Start bit+9-bit data+stop bit
1	1	Start bit+8-bit data+ parity check bit+stop bit

Configurable stop bit

Four different stop bits can be configured through STOPCFG bit of USART_CTRL2 register.

- 1 stop bit: Default stop bit.
- 0.5 stop bit: Used when receiving data in smart card mode.
- 2 stop bits: Used in normal mode, single-line mode and hardware flow control mode.
- 1.5 stop bits: Used when transmitting and receiving data in smart card mode.

Parity bit

PCFG bit of USART_CTRL1 determines the parity check bit; when PCFG=0, it is even parity check, on the contrary, it is odd parity check.

- Even parity check: When the number of frame data and check bit 1 is even, the even parity check bit is 0; otherwise, it is 1.
- Odd parity check: When the number of frame data and check bit 1 is even, the odd parity check bit is 1; otherwise, it is 0.

19.4.3 Transmitter

When TXEN bit of the register USART_CTRL1 is set, the transmit shift register will output data through TX pin and the corresponding clock pulses will be output through CK pin.

19.4.3.1 Character transmission

During transmission period of USART, the least significant bit of the data will be moved out by TX pin first. In this mode, USART_ DATA register has a buffer between the internal bus and the transmit shift register.

A data frame is composed of the start bit, character and stop bit, so there is a low-level start bit in front of each character; then there is a high-level stop bit who number is configurable.

Transmission configuration steps

(1) Set UEN bit of USART_CTRL1 register to enable USART.

(2) Decide the word length by setting DBLCFG bit of USART_CTRL1 register.

(3) Decide the number of stop bits by setting STOPCFG bit of USART_CTRL2 register.

(4) If multi-buffer communication is selected, DMA should be enabled in USART_CTRL3 register.

(5) Set the baud rate of communication in USART_BR register.

(6) Enable TXEN bit in USART_CTRL1 register, and transmit an idle frame.

(7) Wait for TXBEFLG bit of USART_STS register to be set to 1.

(8) Write data to USART_DATA register (if DMA is not enabled, repeat steps 7-8 for each byte to be transmitted).

(9) Wait for TXCFLG bit of USART_STS register to be set to 1, indicating transmission completion.

Note: TXEN bit cannot be reset during data transmission; otherwise, the data on TX pin will be destroyed, which is because if the baud rate generator stops counting, the data being transmitted will be lost.

19.4.3.2 **Single-byte communication**

TXBEFLG bit can be cleared to 0 by writing USART_DATA register. When the TXBEFLG bit is set by hardware, the shift register will receive the data transferred from the data transmit register, then the data will be transmitted, and the data transmit register will be emptied. The next data can be written in the data register without overwriting the previous data.

(1) If TXBEIEN in USART_CTRL1 register is set to 1, an interrupt will be generated.

(2) If USART is in the state of transmitting data, write to the data register to save the data to the DATA register, and transfer the data to the shift register at the end of the current data transmission.

(3) If USART is in idle state, write to the data register, put the data into the shift register, start transmitting data, and set TXBEFLG bit to 1.

(4) When a data transmission is completed and TXBEFLG bit is set, TXCFLG bit will be set to 1; at this time if TXCIEN bit in USART_CTRL1 register is set to 1, an interrupt will be generated.

(5) After the last data is written in the USART_DATA register, before entering the low-power mode or before disabling the USART module, wait to set TXCFLG to 1.

19.4.3.3 **Break frame**

The break frames are considered to receive 0 in a frame period. Setting TXBF bit of USART_CTRL1 register can transmit a break frame, and the length of the

break frame is determined by DBLCFG bit of USART_CTRL1 register. If the TXBF bit is set, after completion of transmission of current data, the TX line will transmit a break frame, and after completion of transmission of break frame, the TXBF bit will be reset. At the end of the break frame, the transmitter inserts one or two stop bits to respond to the start bit.

Note: If the TXBF bit is reset before transmission of the break frame starts, the break frame will not be transmitted. To transmit two consecutive break frames, the TXBF bit should be set after the stop bit of the previous break symbol.

19.4.3.4 **Idle frame**

The idle frame is regarded as a complete data frame composed entirely of 1, followed by the start bit of the next frame containing the data. Set TXEN bit of USART_CTRL1 register to 1 and one idle frame can be transmitted before the first data frame.

19.4.4 Receiver

19.4.4.1 **Character receiving**

During receiving period of USART, RX pin will first introduce the least significant bit of the data. In this mode, USART_DATA register has a buffer between the internal bus and the receive shift register. The data is transmitted to the buffer bit by bit. When the data is fully received, the corresponding receive register is not empty, then the user can read USART_ DATA.

Receiving configuration steps

(1) Set UEN bit of USART_CTRL1 register to enable USART.

(2) Decide the word length by setting DBLCFG bit of USART_CTRL1 register.

(3) Decide the number of stop bits by setting STOPCFG bit of USART_CTRL2 register.

(4) If multi-buffer communication is selected, DMA should be enabled in USART_CTRL3 register.

(5) Set the baud rate of communication in USART_BR register.

(6) Set RXEN bit of USART_CTRL1 to enable receiving.

Note:

(1) RXEN bit cannot be reset during data receiving period; otherwise, the bytes being received will be lost.

(2) In the process of the receiver receiving a data frame, if an overrun error, noise error or frame error is detected, the error flag will be set to 1.

(3) When data is transferred from the shift register to USART_DATA register, the RXBNEFLG bit of USART_STS will be set by hardware.

(4) An interrupt will be generated if RXBNEIEN bit is set.

(5) In single buffer mode, the RXBNEFLG bit can be cleared by reading USART_DATA register by software or by writing 0.

(6) In multi-buffer mode, after each byte is received, the RXBNEFLG bit of USART_STS register will be set to 1, and can be cleared to 0 by reading the data register by DMA.

19.4.4.2 **Break frame**

When the receiver receives a break frame, USART will handle it as receiving a frame error.

19.4.4.3 **Idle frame**

When the receiver receives an idle frame, USART will handle it as receiving an ordinary data frame; if IDLEIEN bit of USART_CTRL1 is set, an interrupt will be generated.

19.4.4.4 **Oversampling rate**

OSMCFG bit of USART_CTRL1 register determines the oversampling rate.

If the oversampling rate is 8 times the baud rate, the speed is higher, but the clock tolerance is smaller. If it is 16 times, the speed is lower, but the clock tolerance is bigger.

19.4.4.5 **Overrun error**

When RXBNEFLG bit of USART_STS register is set to 1 and a new character is received at the same time, an overrun error will be caused. Only after RXEN is reset, can the data be transferred from the shift register to DATA register. RXBNEFLG bit will be set to 1 after receiving the byte. This bit needs to be reset before receiving the next data or serving the previous DMA request; otherwise, an overrun error will be caused.

When an overrun error occurs

- USART_STS OVREFLG bit is set to 1.
- The data in DATA register will not be lost.
- The data in the shift register received before will be overwritten, but the data received later will not be saved.
- If RXBNEIEN bit of USART_CTRL1 is set to 1, an interrupt will be generated.
- When OVREFLG bit is set to 1, it means that the data has been lost. There are two possibilities:
  - When RXBNEFLG=1, the previous valid data is still on DATA register, and can be read.

- When RXBNEFLG=0, there is no valid data in DATA register.

- The OVREFLG bit can be reset by reading USART_STS and USART_DATA registers successively.

19.4.4.6 **Noise error**

When noise is detected in receiving process of the receiver:

- Set NE flag on the rising edge of RXBNEFLG bit of USART_STS register.
- Invalid data is transmitted from the shift register to USART_DATA register.

Note: 8-time oversampling ratio cannot be used in LIN, smart card and IrDA modes.

19.4.4.7 **Frame error**

If the stop bit is not received and recognized at the expected receiving time due to excessive noise or lack of synchronization, a frame error will be detected.

When a frame error is detected in receiving process of the receiver:

(1) Set the FEFLG bit of USART_STS register.

(2) Invalid data is transmitted from the shift register to USART_DATA register.

(3) In single byte communication, there is no interrupt, but in multi-buffer communication, an interrupt will be generated by setting the ERRIEN bit of USART_CTRL3 register.

19.4.5 Baud rate generator

The baud rate division factor (USARTDIV) is a 16-digit number consisting of 12-digit integer part and 4-digit decimal part. Its relationship with the system clock:

Baud rate= FCLK / 16 × (USARTDIV)

The system clock for USART1/2 is FCLK. The USART can only be enabled after enabling the system clock in the clock control unit.

19.4.6 Multiprocessor communication

In multiprocessor communication, multiple USART are connected to form a network. In this network, two devices communicate with each other, and the mute mode can be enabled for other devices not participating in the communication to reduce the burden of USART. In mute mode, no receive state bit will be set and all receive interrupts are disabled.

When mute mode is enabled, there are two ways to exit the mute mode:

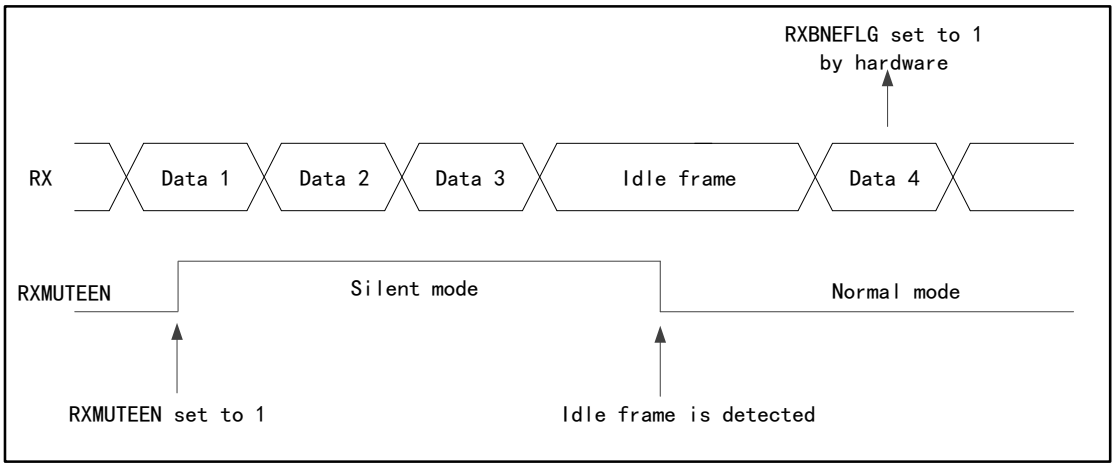
- Exit the mute mode when WUPMCFG bit in USART_CTRL1 register is cleared and the bus is idle.

- Exit the mute mode when WUPMCFG bit in USART_CTRL1 register is set and the address flag is received.

Idle bus detection (WUPMCFG=0)

When RXMUTEEN is set to 1, USART enters the mute mode, and it can be waken up from the mute mode when an idle frame is detected, meanwhile, the RXMUTEEN bit will be cleared to 0 by hardware. RXMUTEEN can also be cleared to 0 by software.

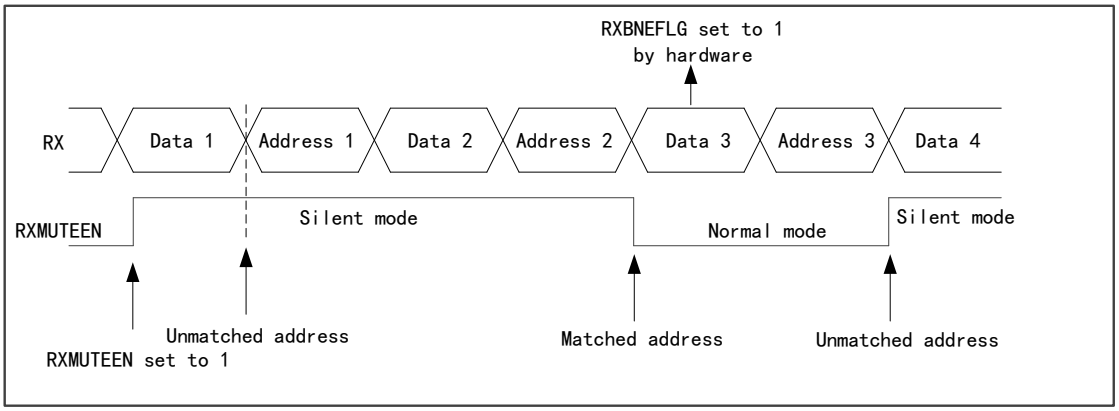
Figure 62 Idle Bus Exits Mute Mode



Address flag detection (WUPMCFG=1)

If the address flag bit is 1, this byte is regarded as the address. The address bytes are low four-byte storage address. When the receiver receives the address byte, it will be compared with its own address. If the addresses do not match, the receiver will enter the mute mode. If the addresses match, the receiver will wake up from the mute mode and be ready to receive the next byte. If the address byte is received again after exiting the mute mode, but the address does not match its own address, the receiver will enter the mute mode again.

Figure 63 Address Flag Exits Mute Mode



19.4.7 Synchronous mode

The synchronous mode supports full-duplex synchronous serial communication in master mode, and has one more signal line USART_CK which can output synchronous clock than the asynchronous mode.

CLKEN bit of USART_CTRL2 register decides whether to enter the synchronous mode.

When USART enters the synchronous mode:

- The LINMEN bit of USART_CTRL2 register, and IREN, HDEN and SCEN bits of USART_CTRL3 register must be cleared to 0.
- The start bit and stop bit of data frame have no clock output.
- Whether the last data bit of data frame generates USART_CK clock is decided by the LBCPOEN bit of USART_CTRL2 register.
- The clock polarity of USART_CK is decided by CPOL bit of USART_CTRL2 register.
- The phase of USART_CK is decided by the CPHA bit of USART_CTRL2 register.
- The external CK clock cannot be activated when the bus is idle or the break frame appears.

Figure 64 USART Synchronous Transmission Timing Diagram (DBLCFG=0)

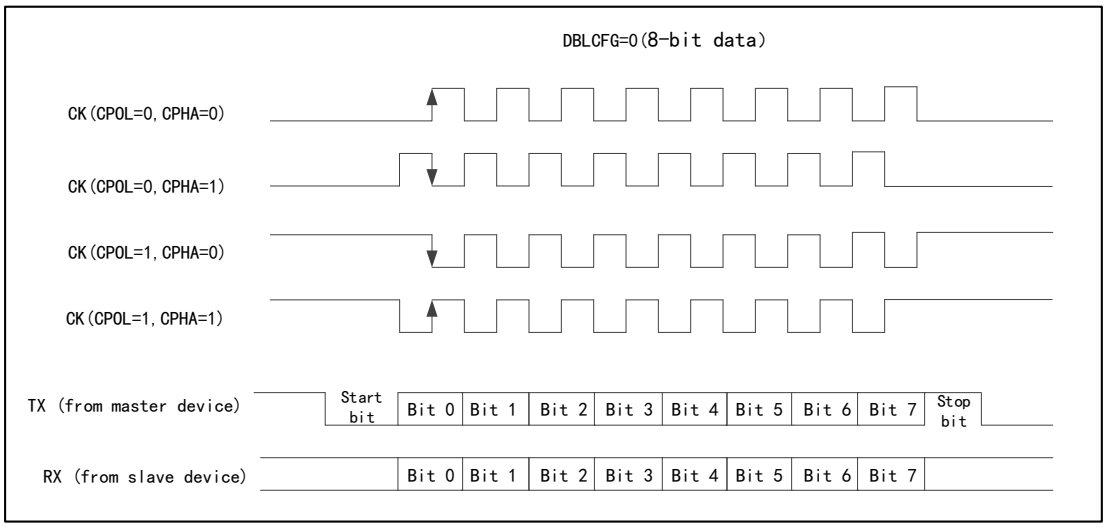
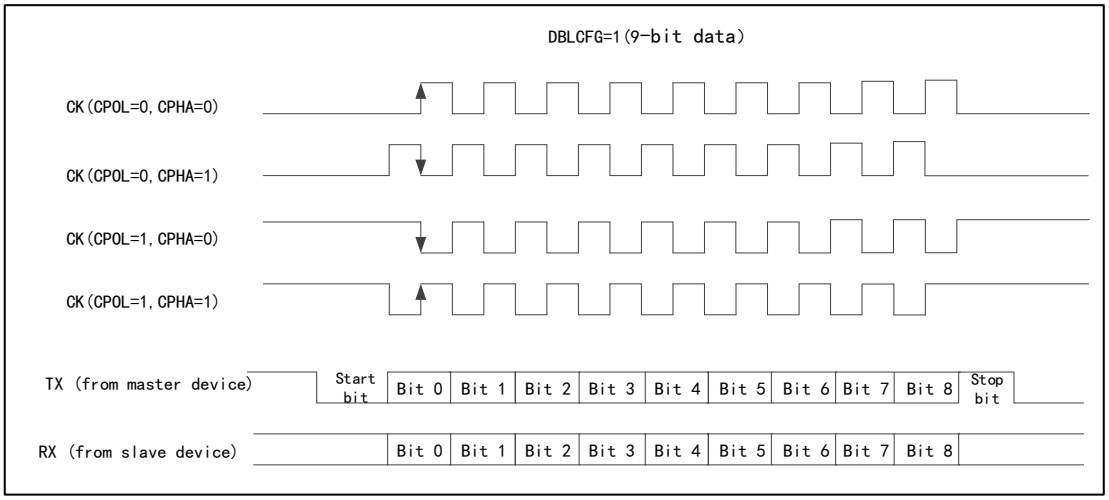


Figure 65 USART Synchronous Transmission Timing Diagram (DBLCFG=1)



19.4.8 LIN mode

LINMEN bit of USART_CTRL2 register decides whether to enter LIN mode.

When entering LIN mode:

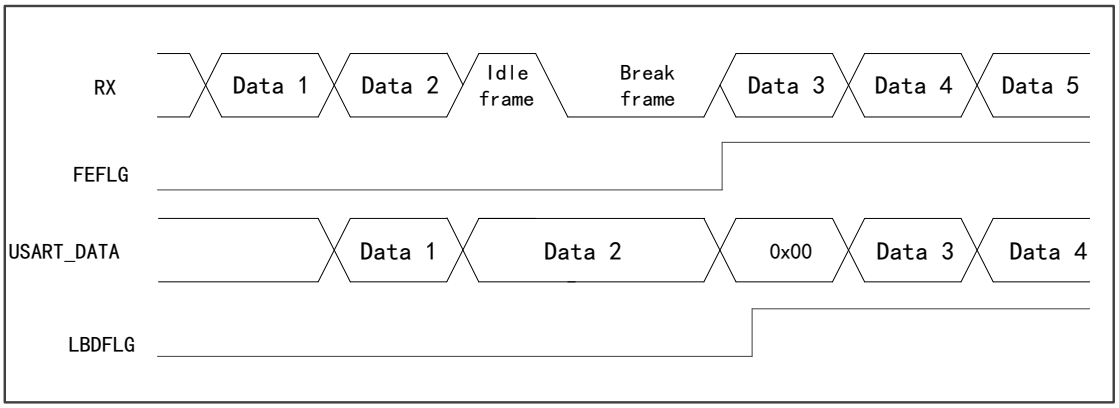
- All data frames are 8 data bits and 1 stop bit.
- The CLKEN bit and STOPCF bit of USART_CTRL2 register and IREN bit, HDEN bit and SCEN bit of USART_CTRL3 register need to be cleared to 0.

In LIN master mode, USART can generate break frame, and the detection length of break frame can be set to 10 or 11 bits through LBDLCFG bit of USART_CTRL2. The break frame detection circuit is independent of USART receiver, and no matter in idle state or in data transmission state, RX pin can detect the break frame, and LBDFLG bit of USART_STS register is set to 1; at this time, if LBDIEN bit of USART_CTRL2 is enabled, an interrupt will be generated.

Detection of break frame in idle state

In idle state, if a break frame is detected on RX pin, the receiver will receive a data frame of 0 and generate FEFLG.

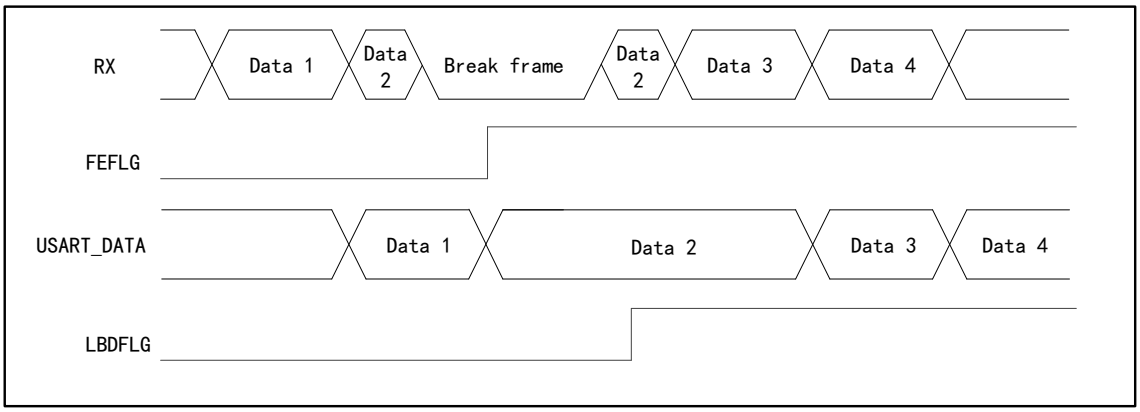
Figure 66 Break Frame Detection in Idle State



Break frame detection in data transmission state

In the process of data transmission, if the RX pin detects the break frame, the currently transmitted data frame will generate FEFLG.

Figure 67 Break Frame Detection in Data Transmission State



19.4.9 Smart card mode

Smart card mode is a single-line half-duplex communication mode. The interface supports ISO7816-3 standard protocol and can control the reading and writing of smart cards that meet the standard protocol.

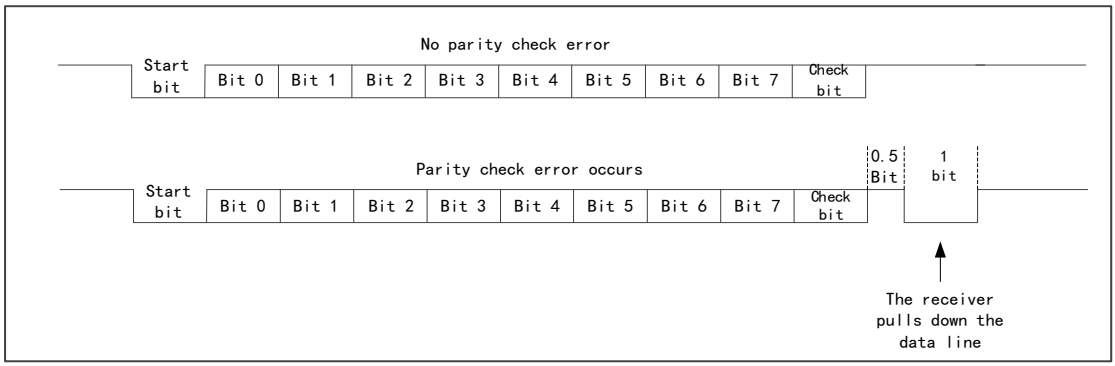
SCEN bit of USART_CTRL3 register decides whether to enter the smart card mode.

When USART enters the smart card mode:

- The LINMEN bit of USART_CTRL2 register, and IREN and HDEN bits of USART_CTRL3 register must be cleared to 0.

- The data frame format is 8 data bits and 1 check bit, and 0.5 or 1.5 stop bits are used. (To avoid switching between two configurations, it is recommended that 1.5 stop bits should be used when transmitting and receiving data)
- CLKEN bit of USART_CTRL2 register can be set to provide clocks for smart card.
- During the communication, when the receiver detects a parity check error, in order to inform the transmitter that the data has not been received successfully, the data line will be pulled down after half a baud rate clock, and keep pulling down for one baud rate clock.
- The break frame has no meaning in smart card mode. A 00h data with frame error will be regarded as a data instead of break symbol.

Figure 68 ISO7816-3 Standard Protocol



19.4.10 Infrared (IrDA SIR) function mode

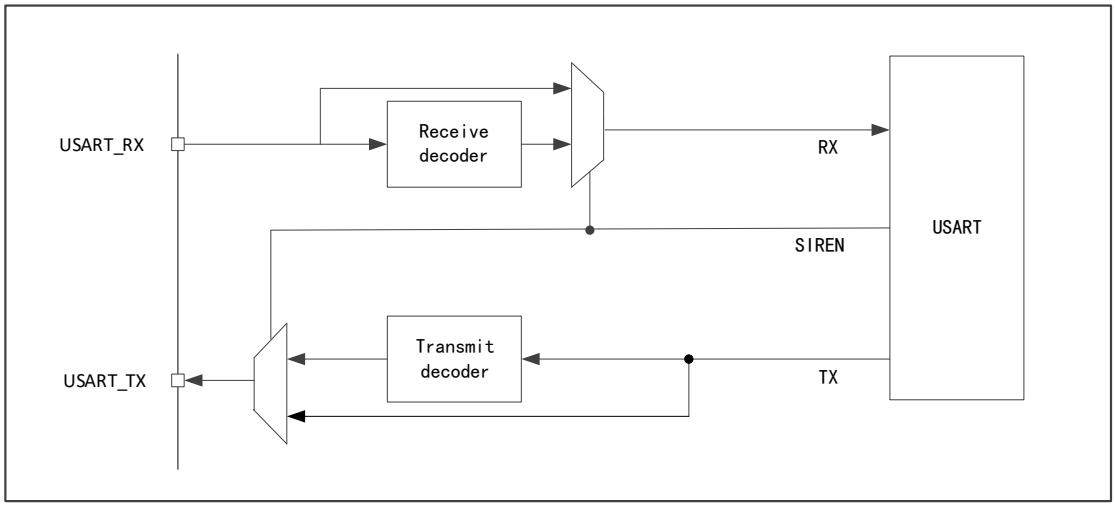
IrDA mode is a half-duplex protocol, transmitting and receiving data can not be carried out at the same time, and the delay between data transmitting and receiving should be more than 10ms.

IREN bit of USART_CTRL3 register decides whether to enter the IrDA mode.

When USART enters the IrDA mode:

- The CLKEN bit, STOPCF bit and LINMEN bit of USART_CTRL2 register and HDEN bit and SCEN bit of USART_CTRL3 register must be cleared to 0.
- The data frame uses 1 stop bit and the baud rate is less than 115200Hz.
- Using infrared pulse (RZI) indicates logic 0, so in normal mode, its pulse width is 3/16 baud rate cycle. In IrDA low-power mode, it is recommended that the pulse width be greater than 3 DIV frequency division clocks to ensure that this pulse can be detected by IrDA normally.

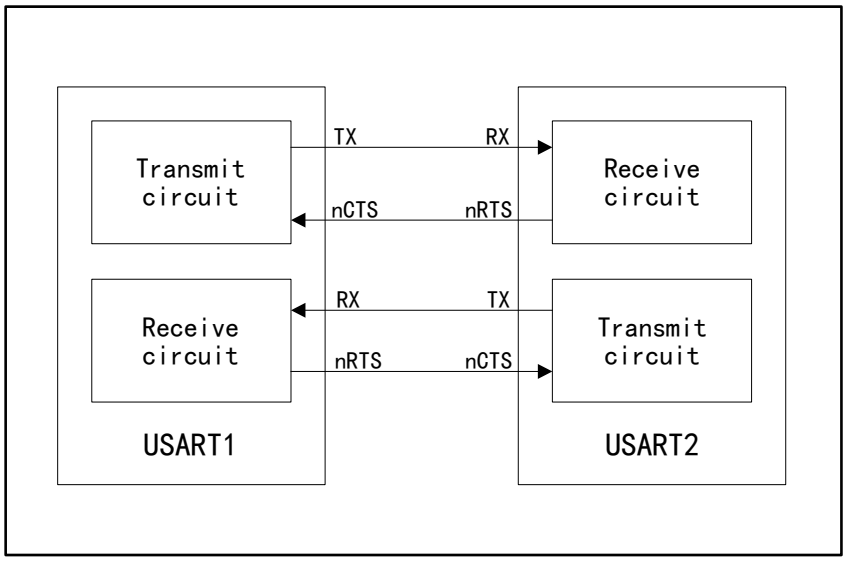
Figure 69 IrDA Mode Block Diagram



19.4.11 Hardware flow control

The function of hardware flow control is to control the serial data stream between two devices through nCTS pin and nRTS pin.

Figure 70 Hardware Flow Control between Two USART



CTS flow control

CTSEN bit of USART_CTRL3 register determines whether to enable CTS flow control. If CTS flow control is enabled, the transmitter will detect whether the data frame of nCTS pin can be transmitted. If TXBEFLG bit=0 for USART_STS register and nCTS is pulled to low, the data frame can be transmitted. If nCTS becomes high during transmission, the transmitter will stop transmitting after the current data frame is transmitted.

RTS flow control

RTSEN bit of USART_CTRL3 register determines whether to enable RTS flow control. If RTS flow control is enabled, when the receiver receives data, nRTS will be pulled to low. When a data frame is received, nRTS will become high to inform the transmitter to stop transmitting data frame.

19.4.12 RS485 driver enable

The driver enable function can be enabled by setting the DEM bit in the USART_CR3 control register to 1. This allows the user to activate the external transceiver control via the DE (Driver Enable) signal. The enabling time is the time interval between the activation of the DE signal and the START bit start time. The disable time can be programmed via the DEAT[4:0] field in the USART_CR1 control register. The disable time is the interval between the end of the last stop bit in the transmitted message and the de-assertion of the DE signal. The disable time can also be programmed via the DEDT[4:0] field in the USART_CR1 control register. The polarity of the DE signal can be configured using the DEP bit in the USART_CR3 control register. In the USART, DEAT and DEDT are expressed in sampling time units (1/8 or 1/16 bit time, depending on the oversampling rate).

19.4.13 DMA multi-buffer communication

USART can access the data buffer in DMA mode to reduce the burden of processors.

Transmission in DMA mode

DMATXEN bit of USART_CTRL3 register determines whether to transmit in DMA mode. When transmitting by DMA, the data in the designated TCM will be transmitted to the buffer by DMA.

Configuration steps of transmission by DMA:

(1) Clear the TXCFLG bit of USART_STS register to zero.

(2) Set the address of TCM memory storing data as DMA source address.

(3) Set the address of USART_DATA register as DMA destination address.

(4) Set the number of data bytes to be transmitted.

(5) Set channel priority.

(6) Set interrupt enable.

(7) Enable DMA channel.

(8) Wait for TXCFLG bit of USART_STS register to be set to 1, indicating transmission completion.

Receive by DMA

DMARXEN bit of USART_CTRL3 register determines whether to use DMA mode to receive; when DMA is used to receive, every time one byte is received, the data of receive buffer will be transmitted to the specified TCM area by DMA.

Configuration steps of receiving by DMA:

(1) Set the address of USART_DATA register as DMA source address.

(2) Set the address of TCM memory storing data as DMA destination address.

(3) Set the number of data bytes to be transmitted.

(4) Set channel priority.

(5) Set interrupt enable.

(6) Enable DMA channel.

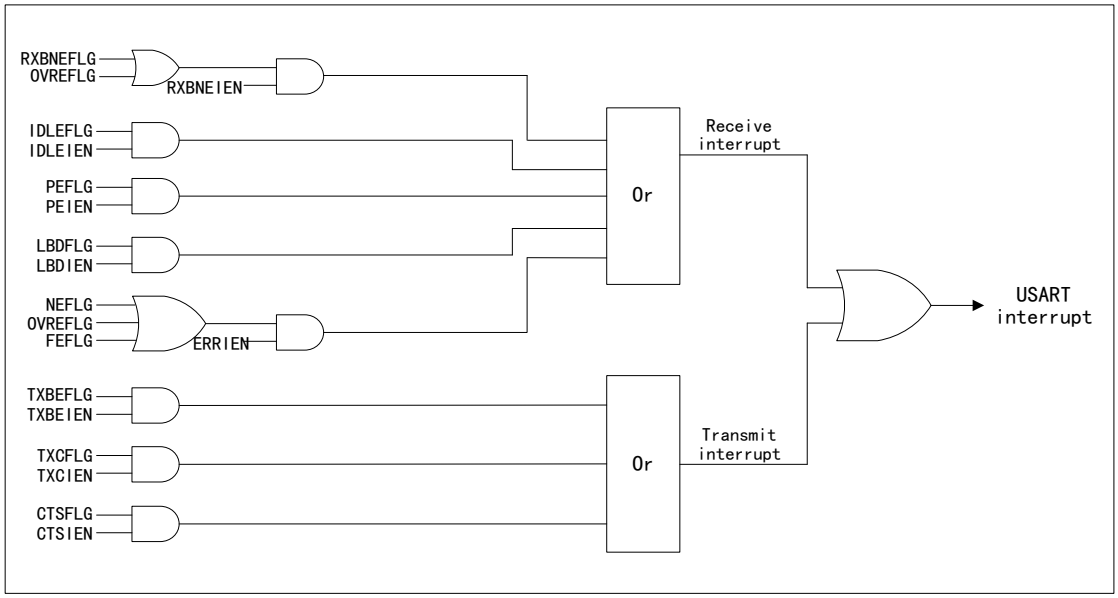
19.4.14 Interrupt request

Table 63 USART Interrupt Request

Interrupt event		Event flag bit	Enable bit
The receive register is not empty		RXBNEFLG	RXBNEIEN
Overrun error		OVREFLG	
Idle line is detected		IDLEFLG	IDLEIEN
Parity check error		PEFLG	PEIEN
LIN break frame flag		LBDFLG	LBDIEN
Receiving error in DMA mode	Noise error	NEFLG	ERRIEN
	Overrun error	OVREFLG	
	Frame error	FEFLG	
Data transmit register is empty		TXBEFLG	TXBEIEN
Transmission Completed		TXCFLG	TXCIEN
CTS flag		CTSFLG	CTSIEN

All interrupt requests of USART are connected to the same interrupt controller, and the interrupt requests have logical or relation before they are transmitted to the interrupt controller.

Figure 71 USART Interrupt Mapping



19.4.15 Comparison of USART supporting functions

Table 64 Comparison of USART Supporting Functions

USART mode	USART1	USART2
Half duplex (single-line mode)	√	√
Multi-processor communication	√	√
Synchronize	√	√
Asynchronous mode	√	√
LIN	√	√
Smart card	√	√
IrDA	√	√
Hardware flow control	√	√
Multi-buffer communication (DMA)	√	√

Note: "√" means this function is supported.

19.5 Register address mapping

Table 65 USART Register Address Mapping

Register name	Description	Offset Address
USART_STS	Status register	0x00
USART_DATA	Data register	0x04
USART_BR	Baud rate register	0x08
USART_CTRL1	Control register 1	0x0C
USART_CTRL2	Control register 2	0x10

Register name	Description	Offset Address
USART_CTRL3	Control register 3	0x14
USART_GTPSC	Protection time and prescaler register	0x18

19.6 Register functional description

19.6.1 State register (USART_STS)

Offset address: 0x00

Reset value: 0x0000 00C0

Field	Name	R/W	Description
31:10	Reserved		
9	CTSFLG	RC_W0	CTS Change Flag 0: No change on nCTS state line 1: There is change on nCTS state line If the CTSEN bit is set, when switching to the nCTS input, set to 1 by hardware; This bit can be cleared to 0 by software; or cleared by writing 0 to this bit. If CTSIEN in USART_CTRL3 is set to 1, an interrupt is generated.
8	LBDFLG	RC_W0	LIN Break Detected Flag 0: LIN break is not detected 1: LIN break is detected When LIN disconnection is detected, set to 1 by hardware; This bit can be cleared to 0 by software; or cleared by writing 0 to this bit. If LBDIEN equals 1, an interrupt should be generated when LBDFLG is 1.
7	TXBEFLG	R	Transmit Data Buffer Empty Flag 0: The transmit data buffer is not empty 1: The transmit data buffer is empty When the shift register receives the data transmitted by the transmit data register, it will be set to 1 by hardware; This bit can be cleared to 0 by software; write USART_DATA register to clear to 0. If the TXBEIEN bit in the USART_CTRL1 register is set to 1, an interrupt is generated. This bit is used in single-buffer transfers.
6	TXCFLG	RC_W0	Transmit Data Complete Flag 0: Transmitting data is not completed 1: Transmitting data is completed After the last frame of data is transmitted and the TXBEFLG is set, it will be set to 1 by hardware; If the TXCIEN bit in USART_CTRL1 is set to 1, an interrupt is generated. This bit can be cleared to 0 by software; first read USART_STS register, and then write USART_DATA register to clear to 0; or this bit can be cleared by writing 0 to it.

Field	Name	R/W	Description
			This clear procedure is recommended only in multi-buffered (multi-cache) communication.
5	RXBNEFLG	RC_W0	Receive Data Buffer Not Empty Flag 0: The receive data buffer is empty 1: The receive data buffer is not empty When the data register receives the data transmitted by the receive shift register, it will be set to 1 by hardware; If the RXBNEIEN bit in the USART_CTRL1 register is set to 1, an interrupt is generated. This bit can be cleared to 0 by software; read USART_DATA to clear to 0, or write 0 to this bit to clear it. This clear procedure is recommended only in multi-buffered (multi-cache) communication.
4	IDLEFLG	R	IDLE Line Detected Flag 0: Idle bus is not detected 1: Idle bus is detected When idle bus is detected, it will be set to 1 by hardware; If the IDLEIEN bit in USART_CTRL1 is set to 1, an interrupt is generated. This bit can be cleared to 0 by software; first read USART_STS register, and then read USART_DATA register to clear to 0. The IDLEFLG bit will not be set to 1 again unless RXBNEFLG is set (when the idle bus is detected again).
3	OVREFLG	R	Overrun Error Occur Flag 0: No overrun error 1: Overrun error occurs When the RXBNEFLG bit is set and the data in the shift register is to be transmitted to the receive register, it will be set to 1 by hardware; If the RXNEIE bit in USART_CTRL1 is set to 1, an interrupt is generated. This bit can be cleared to 0 by software; first read the USART_STS register, then read the USART_DATA register. When this bit is set, the value in the receive data register will not be lost, but the data in the shift register will be overwritten. If the ERRIEN bit of USART_CTRL3 is set, in multi-buffer communication mode an interrupt will be generated when the OVREFLG flag is set.
2	NEFLG	R	Noise Error Occur Flag 0: No noise 1: There is noise error When there is noise error, set to 1 by hardware; This bit can be cleared to 0 by software; first read USART_STS register, and then read USART_DATA register to clear to 0. The bit does not generate an interrupt, because it appears together with the RXBNEFLG; the hardware will generate an interrupt when the RXBNEFLG flag is set. In multi-buffer communication mode, if the ERRIEN bit of USART_CTRL3 is set, an interrupt will be generated when the NEFLG flag is set.

Field	Name	R/W	Description
1	FEFLG	R	Frame Error Occur Flag 0: No frame error 1: A frame error or break symbol appears When there is synchronous dislocation, too much noise or break symbol, set to 1 by hardware; This bit can be cleared to 0 by software; first read USART_STS register, and then read USART_DATA register to clear to 0. This bit does not generate an interrupt, because it appears together with RXBNEFLG; the hardware will generate an interrupt when the RXBNEFLG flag is set. If the currently transmitted data incurs both a frame error and an overrun error, the hardware will still continue transmitting that data and will only set the OVREFLG flag. In multi-buffer communication mode, if the ERRIEN bit is set, an interrupt will be generated when the FEFLG flag is set.
0	PEFLG	R	Parity Error Occur Flag 0: No error 1: Parity check error occurs In the receiving mode, when a parity check error occurs, set to 1 by hardware; If the PEIEN bit in USART_CTRL1 is set to 1, an interrupt is generated. This bit can be cleared to 0 by software; after setting of RXBNEFLG, first read USART_STS register, and then read USART_DATA register to clear to 0.

19.6.2 Data register (USART_DATA)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:9	Reserved		
8:0	DATA	R/W	Data Value Transmit or receive the data value; read data when receiving data, and write data to the register when transmitting data. It is composed of two registers, one for transmit (TDR) and one for receive (RDR). TDR provides a parallel interface between the internal bus and the output shift register; RDR provides a parallel interface between the input shift register and the internal bus. When the parity bit is enabled, for 9 data bits, the 8 bit of DATA is parity bit; for 8 data bits, the 7 bit of DATA is parity bit.

19.6.3 Baud rate register (USART_BR)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		

Field	Name	R/W	Description
15:4	IBR[15:4]	R/W	Integer of USART Baud Rate Divider Factor The integral part of USART baud rate division factor is determined by these 12 bits. If TXEN and RXEN are disabled respectively, the baud counter stops counting.
3:0	FBR[3:0]	R/W	Fraction of USART Baud Rate Divider Factor The decimal part of USART baud rate division factor is determined by these four bits. If TXEN and RXEN are disabled respectively, the baud counter stops counting.

19.6.4 Control register 1 (USART_CTRL1)

Offset address: 0x0C

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:26	Reserved		
25:21	DEAT	R/W	Driver Enable Assertion Time This 5-bit value defines the time interval between the activation of DE (Driver Enable) signal and the start bit start time. This time interval is expressed in sampling time units (1/8 or 1/16 bit time, depending on the oversampling rate). This field can only be written when USART is disabled (UEN=0). Note: If the Driver Enable feature is not supported, this bit field is reserved and must retain its reset value.
20:16	DEDT	R/W	Driver Enable De-assertion Time This 5-bit value defines the time interval between the end of the last stop bit in a transmitted message and the de-assertion of the DE (Driver Enable) signal. This time interval is expressed in sampling time units (1/8 or 1/16 bit time, depending on the oversampling rate). If a write operation to the TDR register occurs within the DEDT time, the new data will be transmitted only after the time of DEDT and DEAT has elapsed. This field can only be written when USART is disabled (UEN=0). Note: If the Driver Enable feature is not supported, this bit field is reserved and must retain its reset value.
15	OSMCFG	R/W	Oversampling Mode Configure 0: 16-time oversampling 1: 8-time oversampling This bit can be set only when USART is not enabled. 8 times oversampling is unavailable in Smart Card, IrDA, and LIN modes: when SCEN=1, IREN=1, or LINMEN=1, the hardware forcefully sets OSMCFG to 0.
14	Reserved		
13	UEN	R/W	USART Enable 0: Disable USART frequency divider and output 1: Enable USART module

Field	Name	R/W	Description
12	DBLCFG	R/W	Data Bits Length Configure 0: 1 start bit, 8 data bits, n stop bits 1: 1 start bit, 9 data bits, n stop bits During data transfer (either sending or receiving), this bit must not be modified.
11	WUPMCFG	R/W	Wakeup Method Configure 0: Idle bus wakeup 1: Address tag wakeup
10	PCEN	R/W	Parity Control Enable 0: Disable 1: Enable If this bit is set, a check bit will be inserted in the most significant bit when transmitting data; when receiving data, check whether the check bit of the received data is correct. The check control will not take effect until the current transmission of bytes is completed.
9	PCFG	R/W	Odd/Even Parity Configure 0: Even parity check 1: Odd parity check The selection will not take effect until the current transmission of bytes is completed.
8	PEIEN	R/W	Parity Error interrupt Enable 0: Disable interrupt generation 1: An interrupt will be generated when PEFLG is set
7	TXBEIEN	R/W	Transmit Buffer Empty Interrupt Enable 0: Disable interrupt generation 1: An interrupt will be generated when TXBEFLG is set
6	TXCIEN	R/W	Transmit Complete Interrupt Enable 0: Disable 1: An interrupt will be generated when TXCFLG is set
5	RXBNEIEN	R/W	Receive Buffer Not Empty Interrupt Enable 0: Disable 1: An interrupt will be generated when OVREFLG or RXBNEFLG is set
4	IDLEIEN	R/W	IDLE Interrupt Enable 0: Disable 1: An interrupt will be generated when IDLEFLG is set
3	TXEN	R/W	Transmit Enable 0: Disable 1: Enable Except in smart card mode, if there is a 0 pulse on this bit at any time of transmitting data, an idle bus will be transmitted after the current data is transmitted. After this bit is set, the data will be transmitted after delay of one-bit time.

Field	Name	R/W	Description
2	RXEN	R/W	Receive Enable 0: Disable 1: Enable, and start to detect the start bit on RX pin
1	RXMUTEEN	R/W	Receive Mute Mode Enable 0: Normal working mode 1: Mute mode This bit is set or cleared to 0 by software, or cleared to 0 by hardware when wakeup sequence is detected. USART must receive a data before it is put in the mute mode, so that it can be detected and awakened by idle bus. In the wake-up of address flag detection, if the RXBNEFLG bit is set, the RXMUTEEN bit cannot be modified by software.
0	TXBF	R/W	Transmit Break Frame 0: Not transmit 1: Will transmit This bit can be set by software and cleared to 0 by hardware when the stop bit of the break frame is transmitted.

19.6.5 Control register 2 (USART_CTRL2)

Offset address: 0x10

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:15	Reserved		
14	LINMEN	R/W	LIN Mode Enable 0: Disable 1: Enable In LIN mode, the TXBF bit in the USART_CTRL1 register can be used to send the LIN break field (low 13 bits) and to detect the LIN break field.
13:12	STOPCFG	R/W	STOP Bit Configure 00: 1 stop bit 01: 0.5 stop bit 10: 2 stop bits 11: 1.5 stop bits
11	CLKEN	R/W	Clock Enable (CK pin) 0: Disable 1: Enable
10	CPOL	R/W	Clock Polarity Configure The state of CK pin when USART is in idle state 0: Low level 1: High level This bit is valid only in synchronous mode.
9	CPHA	R/W	Clock Phase Configure This bit indicates on the edge of which clock sampling is conducted 0: The first 1: The second

Field	Name	R/W	Description
			This bit is valid only in synchronous mode.
8	LBCPOEN	R/W	Last Bit Clock Pulse Output Enable This bit indicates whether a clock pulse corresponding to the last transmitted data byte (MSB) is output on the CK pin. 0: Not output from CK 1: Output from CK This bit is valid only in synchronous mode.
7	Reserved		
6	LBDIEN	R/W	LIN Break Detection Interrupt Enable 0: Disable 1: An interrupt is generated when LBDFLG bit is set
5	LBDLCFG	R/W	LIN Break Detection Length Configure 0: 10 bits 1: 11 bits
4	Reserved		
3:0	ADDR[3:0]	R/W	USART Device Node Address Setup This bit is valid only in the mute mode of multiprocessor communication, and decides to enter the mute mode or wake up depending on whether the detected address tags are consistent.

Note: These three bits (CPOL, CPHA and LBCPOEN) cannot be changed after transmission is enabled.

19.6.6 Control register 3 (USART_CTRL3)

Offset address: 0x14

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:13	Reserved		
12	DEM	R/W	Driver Enable Mode This bit is used to activate external transceiver control via the DE signal. 0: DE function disabled. 1: DE function enabled. The DE signal is output on the chip's DE pin (active under high level). This field can only be written when USART is disabled (UEN=0). Note: If the Driver Enable feature is not supported, this bit is reserved and must retain its reset value.
11	SAMCFG	R/W	Sample Method Configure 0: Sampling for three times 1: Single sampling
10	CTSIEN	R/W	CTS Interrupt Enable 0: Disable 1: An interrupt will be generated when CTSFLG is set

Field	Name	R/W	Description
9	CTSEN	R/W	CTS Hardware Flow Control Function Enable 0: Disable 1: Enable CTS: Clear To Send, which is input signal Only when CTS input signal is low, the data can be transmitted; otherwise, the data cannot be transmitted; if CTS signal is pulled high during data transmission, the data transmission will be stopped after the data transmission is completed; if write operation is performed for the data register when CTS is high, the data will not be transmitted until CTS is valid.
8	RTSEN	R/W	RTS Hardware Flow Control Function Enable 0: Disable 1: Enable RTS interrupt RTS: Require To Send, which is output signal, indicating it has been ready to receive. Request is made to receive data only when there is space in the receive buffer; when data can be received, RTS output is pulled to low.
7	DMATXEN	R/W	DMA Transmit Enable 0: Disable 1: Enable
6	DMARXEN	R/W	DMA Receive Enable 0: Disable 1: Enable
5	SCEN	R/W	Smartcard Function Enable 0: Disable 1: Enable
4	SCNACKEN	R/W	NACK Transmit Enable During Parity Error in Smartcard Function 0: When a parity error occurs, not transmit NACK 1: When a parity error occurs, transmit NACK
3	HDEN	R/W	Half-duplex Mode Enable 0: Disable 1: Enable
2	IRLPEN	R/W	IrDA Low-power Mode Enable 0: Normal mode 1: Low-power mode
1	IREN	R/W	IrDA Function Enable 0: Disable 1: Enable
0	ERRIEN	R/W	Error interrupt Enable 0: Disable 1: Enable; when DMARXEN is set and one among FEFLG, OVREFLG or NEFLG is set, an interrupt will be generated.

19.6.7 Protection time and prescaler register (USART_GTPSC)

Offset address: 0x18

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15:8	GRDT	R/W	Guard Time Value Setup After transmitting data, TXCFLG can be set only after the protection time; the time unit is baud clock; it can be applied to smart card mode.
7:0	PSC	R/W	Prescaler Factor Setup Divide the frequency of the system clock and provide the clock; in different working modes, the significant bits of PSC have difference, specifically as follows: In infrared low-power mode: PSC[7:0] is significant. To obtain the frequency in low-power mode, divide the system clock: 00000000: Reserved 00000001: 1 divided frequency 00000010: 2 divided frequency …… 11111111: 255 divided frequency In infrared normal mode: PSC can only be set to 00000001 In smart card mode: PSC[7:5] is insignificant, PSC[4:0] is significant To provide the clock to the smartcard, divide the system clock: 00000: Reserved 00001: 2 divided frequency 00010: 4 divided frequency 00011: 6 divided frequency …… 11111: 62 divided frequency

20 Internal integrated circuit interface (I2C)

20.1 Full Name and Abbreviation of Terms

Table 66 Full Name and Abbreviation Description of Terms

Full name in English	English abbreviation
Serial Data	SDA
Serial Clock	SCL
System Management Bus	SMBus
Clock	CLK
Serial Clock High	SCLH
Serial Clock Low	SCLL
Address Resolution Protocol	ARP
Negative Acknowledgement	NACK

Full name in English	English abbreviation
Packet Error Checking	PEC
Address Resolution Protocol	ARP

20.2 Introduction

I2C is a short-distance bus communication protocol. In physical implementation, I2C bus is composed of two signal lines (SDA and SCL) and a ground wire. These two signal lines can be used for bidirectional transmission.

- Two signal lines, SCL clock line and SDA data line. SCL provides timing for SDA, and SDA transmits/receives data in series
- Both SCL and SDA signal lines are bidirectional
- The ground is common when the two systems use I2C bus for communication

20.3 Main characteristics

(1) Multi-master function
- The master can generate the clock, start bit and stop bit
- Can control all I2C bus–specific sequencing, protocol, arbitration, and timing

(2) Slave function
- Programmable I2C address detection
- Double-address mode
- Detection of stop bit

(3) 7-bit and 10-bit addressing mode

(4) Response to broadcast

(5) Two communication speeds
- Standard mode
- Fast mode

(6) Programmable clock extension

(7) State flag
- Transmitter/Receiver mode flag
- Flag for end of byte transmission
- Busy bus flag

(8) Error flag
- Arbitration loss
- Acknowledgment error
- Detection of wrong start bit or stop bit

(9) Interrupt source
- Address/Data communication succeeded
- Error interrupt

(10) Support DMA function

(11) Support waking up from Stop mode when the address matches

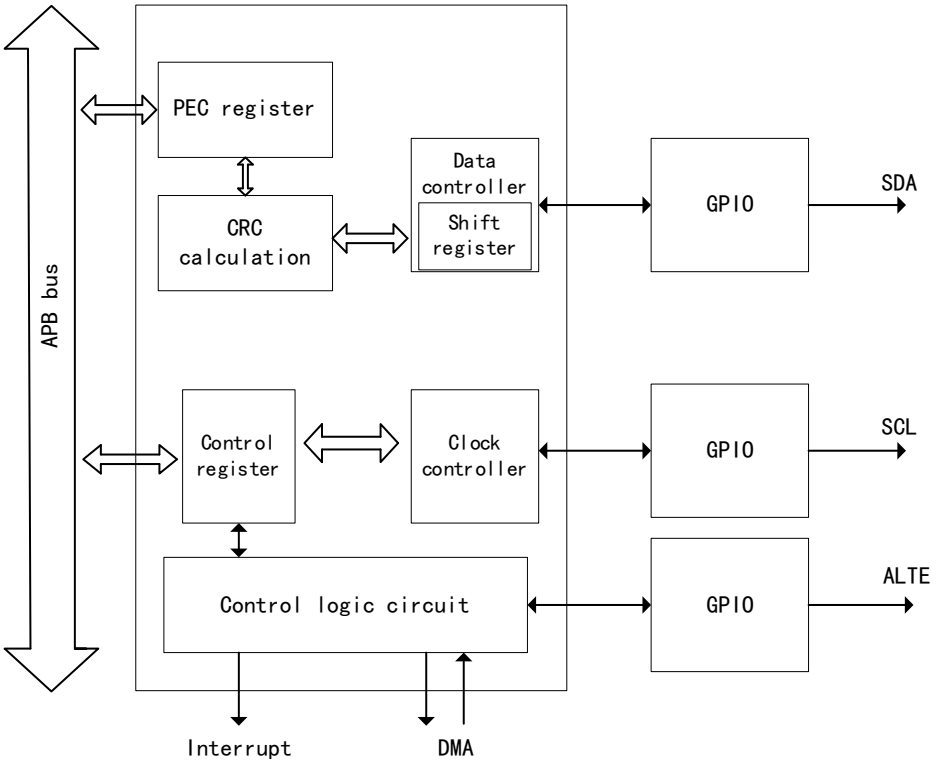
(12) Programmable digital noise filter

(13) Programmable PEC
- Final transmission in transmission mode
- PEC error check after the last byte is received

(14) SMBus specific function
- Hardware PEC
- Address resolution protocol

20.4 Structure block diagram

Figure 72 I2C Function Structure Diagram



The interface can be configured to the following modes:
- Slave transmitting
- Slave receiving
- Master transmitting
- Master receiving

In the initial state of I2C interface, the working mode is slave mode. After I2C interface transmits the start signal, it will automatically switch from slave mode to master mode.

20.5 Functional Description

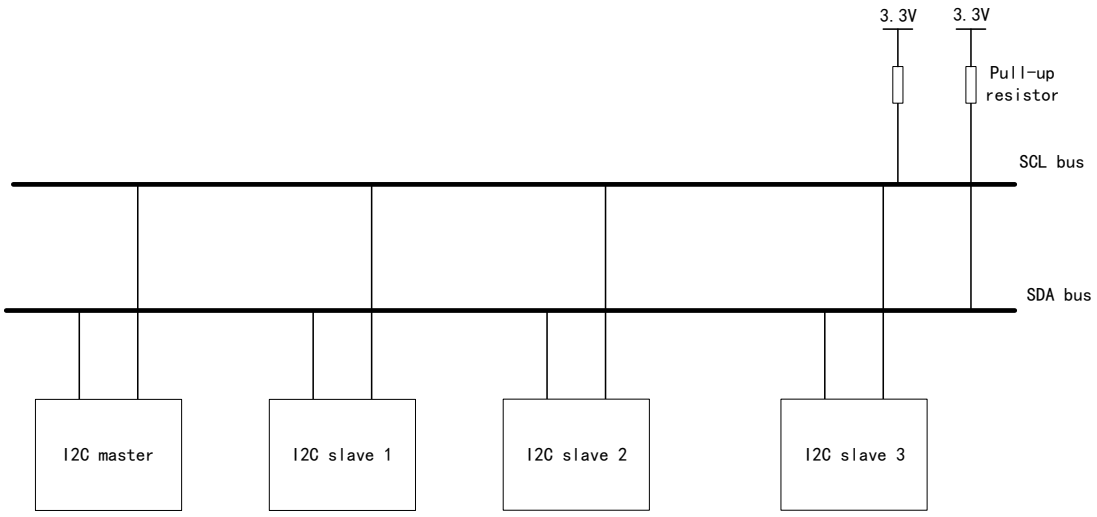
Table 67 Description of Proper Noun of I2C Bus

Proper nouns	Description
Transmitter	Device transmitting data to the bus
Receiver	Device receiving data from the bus
Master	Device that initiates data transmission, generates clock signals and ends data transmission
Slave	Device addressed by master
Multiple masters	Multiple masters that control the bus at the same time without destroying information
Synchronize	The process of synchronizing the clock signals between two or more devices
Arbitration	If more than one master tries to control the bus at the same time, only one master can control the bus, and the information of the controlled master will not be destroyed

20.5.1 I2C physical layer

The commonly used connection modes between I2C communication devices are shown in the figure below:

Figure 73 Common I2C Communication Connection Diagram



Characteristics of physical layer:

(1) It supports the buses of multiple devices (signal line shared by multiple devices). In I2C communication bus, can connect multiple communication masters and communication slaves.

(2) An I2C bus only uses two bus lines, namely, a bidirectional serial data line (SDA) and a serial clock line (SCL). The data line is used for data transmission, and the clock line is used for synchronous receiving and transmission of data.

(3) Each device connected to the bus has an independent address (seven or ten bits), and the master addresses and accesses the slave devices according to the address of the device.

(4) The bus needs to connect the pull-up resistor to the power supply. When I2C bus is idle, the output is in high-impedance state. When all devices are idle, the output is in high-impedance state, and the pull-up resistor pulls the bus to high.

(5) Three communication modes: Standard mode, Fast mode.

(6) When the bus is used by multiple masters at the same time, to prevent data collision, the bus arbitration mode is adopted to determine which device occupies the bus.

(7) Able to program the setup and hold time, and program the high-level time and low-level time of SCL in I2C.

20.5.2 I2C protocol layer

Characteristics of protocol layer

(1) Data is transmitted in the form of frame, and each frame is composed of 1 byte (8 bits).

(2) In the rising edge phase of SCL, SDA needs to keep stable and SDA changes when SCL is low.

(3) In addition to data frame, I2C bus also has start signal, stop signal and acknowledgment signal.

- Start bit: During the stable high level period of SCL, a falling edge of SDA starts transmission.
- Stop bit: During the stable high level period of SCL, a rising edge of SDA stops transmission.
- Acknowledge bit: Used to indicate successful transmission of one byte. After the bus transmitter (regardless of the master or slave) transmits 8-bit data, SDA will release (from output to input). During the ninth clock pulse period, the receiver will pull down SDA to acknowledge receiving of data.

I2C communication reading and writing process

Figure 74 Master Writes Data to Slave

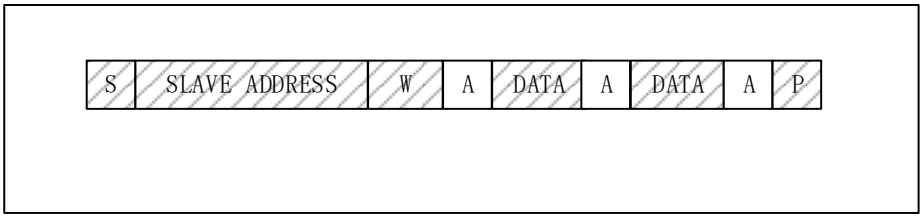
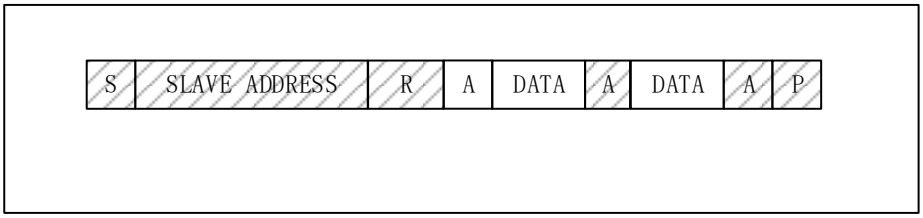


Figure75 Master Reads Data from Slave



Remarks:

(1) : This data is transmitted from master to slave

(2) S: Start signal

(3) SLAVE ADDRESS: Slave address

(4) : This data is transmitted from slave to master

(5) R/W: Selection bit of transmission direction

(6) 1 means read

(7) 0 means write

(8) P: Stop signal

After the start signal is generated, all slaves will wait for the slave address signal transmitted by the master. In I2C bus, the address of each device is unique. When the address signal matches the device address, the slave will be selected, and the unselected slave will ignore the future data signal.

When the master direction is writing data

After broadcasting the address and receiving the acknowledge signal, the master will transmit data to the slave, the data length is one byte, and every time the master transmits one byte of data, it needs to wait for the acknowledge signal transmitted by the slave. After all the bytes have been transmitted, the master will transmit a stop signal (STOP) to the slave, indicating that the

transmission is completed.

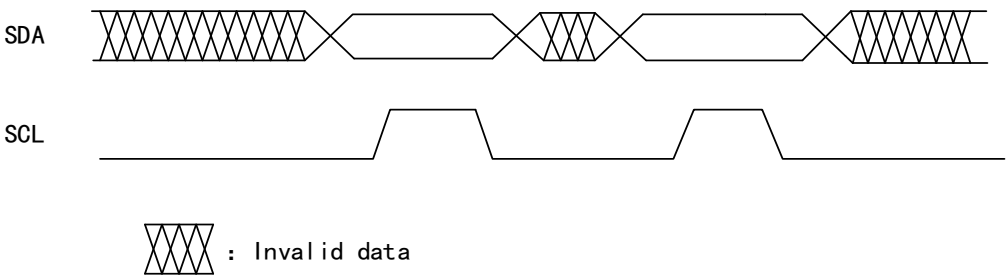
When the master direction is reading data

After broadcasting the address and receiving the acknowledge signal, the slave will transmit the data to the master. The size of the data package is 8 bits. Every time the slave transmits one byte of data, it needs to wait for the acknowledge signal of the master. When the master needs to stop receiving data, it needs to return a non-acknowledge signal to the slave, then the slave will stop transmitting the data automatically.

20.5.3 Data validity

In the process of data transmission, the data on SDA line must be stable when the clock signal SCL is at high level. Only when the SCL is at the low level, can the level state of SDA be changed, and the bit transmission of each data needs a clock pulse.

Figure 76 SDA Timing Diagram



20.5.4 Start and stop signals

All data transmission must have start signal (START) and stop signal (STOP).

Figure 77 START signal is defined as: when SCL is at high level, SDA will convert from high level to low level

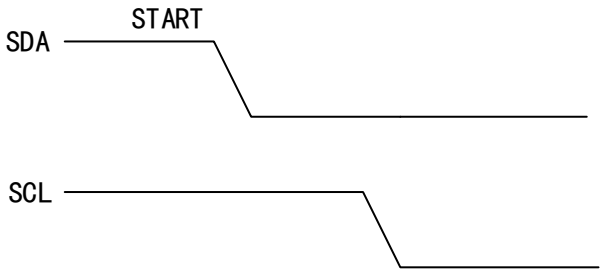
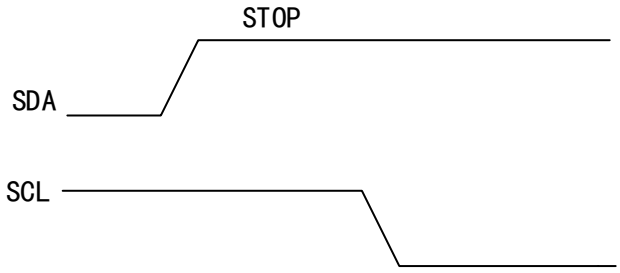


Figure 78 STOP signal is defined as: when SCL is at high level, SDA will convert from low level to high level



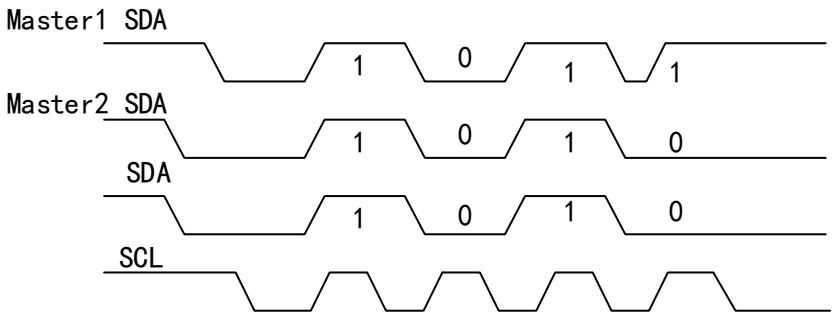
20.5.5 Arbitration

Arbitration is also used to solve the bus control conflict in case of multiple masters. The arbitration process takes place on the master and has nothing to do with the slave.

The master can start transmission only when the bus is idle. Two masters may generate an effective START signal on the bus within the shortest hold time of the START signal. In this situation, it is required that arbitration should decide which master completes the transmission.

Arbitration is conducted by bit. During each arbitration, when SCL is high, each master will check whether the SDA level is the same as that transmitted by itself. The arbitration process needs to last for many bits. Theoretically, if two masters transmit exactly the same content, they can successfully transmit without arbitration failure. If one master transmits high level, but it is detected that SDA is at low level, an arbitration failure error will occur, the SDA output of the master will be disabled, and the other master will complete its own transmission.

Figure 79 SDA Timing Diagram



Note: Master1 arbitration failure

20.5.6 SMBus specific function

System management bus (SMBus) is a simple single-end double-wire bus, which can meet the requirements of lightweight communication.

SMBus is commonly used in computer motherboard, mainly for power transmission ON/OFF instructions. SMBus is the derivative bus of I2C. It is mainly used for communication of low-bandwidth devices on computer motherboard, and power-related chip.

Address resolution protocol

SMBus specification includes an address resolution protocol, which can realize dynamic address assignment. Dynamic recognition hardware and software enable the bus to support hot plugging, and the bus devices will be automatically identified and assigned with a unique address.

SMBus alarm

SMBus alarm is an optional signal with an interrupt line, which enhances the control capability of extended devices by occupying pins.

20.5.7 Error flag bit

Table 68 The following several error flag bits exist in I2C communication

Error flag bit	Description of error flag bit
Answer error flag bit (AEFLG)	No acknowledgment received
Bus error flag bit (BERRFLG)	An external stop or start condition is detected
Arbitration loss flag bit (ALFLG)	Arbitration loss is detected by the interface
Overrun/Underrun error flag bit (OVRURFLG)	In slave mode, the received data is not read out, the next data has arrived, and an overrun error occurs. The transmitting data clock has arrived, but the data has not been written to the DATA register, and an underrun error occurs.
Timeout or Tlow error flag bit (TTEFLG)	SCL is pulled down for more than a certain time
PEC comparison error flag bit (PECEFLG)	CRC values are not equal

20.5.8 Packet error check (PEC)

I2C module has a PEC module, which checks the message of I2C data by CRC-8 calculator. The CRC-8 polynomial used by the calculator is: $C(x)= X^8+X^2+X+1$.

When PECEN bit is set to 1 and PEC function is enabled, PEC module will calculate all data transmitted by I2C bus, including address data.

20.5.9 DMA mode

According to the software process of I2C, when the transmit register is empty or the receiver register is full, MCU needs to write or read bytes, then we can complete the operation more quickly through the DMA function of I2C.

DMA transmission

Set the DMAEN bit in I2C_CTRL2 register to enable the DMA mode. When the transmit register is empty (TXBEFLG is set to 1), the data will be directly loaded from the memory area to the DATA register through DMA.

DMA receiving

Set DMAEN bit in I2C_CTRL2 register to enable DMA mode. When the receive register is full (RXBNEFLG is set to 1), DMA will transmit DATA register data to the set storage area.

20.5.10 Programmable noise filter

In Fm mode, the I2C standard requires that the spikes on SDA and SCL lines should be suppressed to a length of 50ns.

The analog noise filter is implemented by SDA and SCL I/O. The analog noise filter is enabled by default and can be disabled by setting ANFEN bit of I2C_FILTER register.

The digital noise filter can be enabled and its filtering capability can be configured by DNFCFG bit of I2C_FILTER register.

20.5.11 I2C interrupt

Table 69 I2C Interrupt Request

Interrupt event	Event flag bit	Interrupt control bit
Start bit transmission completed	STARTFLG	EVIEN
Transmission completed/Address matching address signal	ADDRFLG	
10-bit address head segment transmission completed	ADDR10FLG	
Stop signal received	STOPFLG	
Data byte transmission completed	BTCFLG	
Receive buffer not empty	RXBNEFLG	EVIEN and BUFIEN
Transmit buffer empty	TXBEFLG	
Bus error	BERRFLG	ERRIEN
Arbitration loss	ALFLG	
Answer failed	AEFLG	
Overrun/Underrun	OVRURFLG	
PEC error	PECEFLG	
Timeout or Tlow error	TTEFLG	
SMBus reminder	ALERTEN	

20.6 Register address mapping

Table 70 I2C Register Address Mapping

Register name	Description	Offset Address
I2C_CTRL1	Control register 1	0x00
I2C_CTRL2	Control register 2	0x04
I2C_SADDR1	Slave mode address register 1	0x08
I2C_SADDR2	Slave mode address register 2	0x0C
I2C_DATA	Data register	0x10
I2C_STS1	Status register 1	0x14
I2C_STS2	Status register 2	0x18
I2C_CLKCTRL	Slave clock control register	0x1C
I2C_RISETMAX	Maximum rising time register	0x20
I2C_FILTER	Filter control register	0x24

20.7 Register functional description

20.7.1 Control register 1 (I2C_CTRL1)

Offset address: 0x00

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15	SWRST	R/W	Software Configure I2C under Reset State 0: Not reset 1: Reset; before I2C is reset, ensure that I2C pin is released and the bus is in idle state.
14	Reserved		
13	ALERTEN	R/W	SMBus Alert Enable This bit can be set to 1 or cleared to 0 by software; when I2CEN=0, it is cleared to 0 by hardware. 0: Release the SMBAlert pin to make it higher, and transmit the response address header immediately after prompt is given to transmit the NACK signal 1: Drive SMBAlert pin to make it lower, and transmit the response address header immediately after prompt is given to transmit the ACKEN signal
12	PEC	R/W	Packet Error Check Transfer Enable This bit can be set to 1 or cleared to 0 by software; after PEC, start bit or stop bit is transmitted, or when I2CEN=0, it is cleared to 0 by hardware.

Field	Name	R/W	Description
			0: Disable 1: Enable
11	ACKPOS	R/W	Acknowledge /PEC Position Configure This bit can be set to 1 or cleared to 0 by software; when I2CEN=0, it is cleared by hardware. 0: When receiving current byte, whether transmitting NACK/ACK, whether PEC is in shift register 1: When receiving next byte, whether transmitting NACK/ACK and whether PEC is in the next byte of shift register
10	ACKEN	R/W	Acknowledge Transfer Enable This bit can be set to 1 or cleared to 0 by software; when I2CEN=0, it is cleared by hardware. 0: Not transmit 1: Transmit
9	STOP	R/W	Stop Bit Transfer This bit can be set to 1 or cleared to 0 by software; when transmitting the stop bit, it is cleared to 0 by hardware; when timeout error is detected, it is set to 1 by hardware. 0: Not transmit 1: Transmit
8	START	R/W	Start Bit Transfer This bit can be set to 1 and cleared to 0 by software; when transmitting the start bit or 2CEN=0, it is cleared to 0 by hardware. 0: Not transmit 1: Transmit
7	CLKSTRETCHD	R/W	Slave Mode Clock Stretching Disable 0: Enable 1: Disable In slave mode, enabling extending the low-level time of the clock can avoid overrun and underrun errors.
6	SRBEN	R/W	Slave Responds Broadcast Enable 0: Disable 1: Enable Note: The broadcast address is 0X00
5	PECEN	R/W	PEC Enable 0: Disable 1: Enable
4	ARPEN	R/W	ARP Enable 0: Disable 1: Enable If SMBTCFG=0, use the default address of SMBus device If SMBTCFG=1, use the master address of SMBus
3	SMBTCFG	R/W	SMBus Type Configure 0: SMBus device 1: SMBus master

Field	Name	R/W	Description
2	Reserved		
1	SMBEN	R/W	SMBus Mode Enable 0: I2C mode 1: SMBus mode
0	I2CEN	R/W	I2C Enable 0: Disable 1: Enable

20.7.2 Control register 2 (I2C_CTRL2)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:13	Reserved		
12	LTCFG	R/W	DMA Last Transfer Configure Configure whether the EOT of the next DMA is the last transmission received, and only used for the master receiving mode. 0: No 1: Yes
11	DMAEN	R/W	DMA Requests Enable 0: Disable 1: When TXBEFLG=1 or RXBNEFLG=1, enable DMA request
10	BUFIEN	R/W	Buffer Interrupt Enable 0: Disable 1: Enable; when the bit of any of the following state registers is set to 1, the interrupt will be generated: TXBEFLG and RXBNEFLG
9	EVIEN	R/W	Event Interrupt Enable 0: Disable 1: Enable; when the bit of any of the following state registers is set to 1, the interrupt will be generated: STARTFLG, ADDRFLG, ADDR10FLG, STOPFLG, BTCFLG, TXBEFLG is set to 1 and BUFIEN is set to 1, RXBNEFLG is set to 1 and BUFIEN is set to 1.
8	ERRIEN	R/W	Error Interrupt Enable 0: Disable 1: Enable; when the bit of any of the following state registers is set to 1, this interrupt will be generated: SMBALTFLG, TTEFLG, PECEFLG, OVRURFLG, AEFLG, ALFLG, and STS1_BERRFLG

Field	Name	R/W	Description
7:0	CLKFCFG	R/W	I2C Clock Frequency Configure The clock frequency is frequency of the clock of I2C module, namely, the clock input from APB bus. 0: Disable 1: Disable 2：2MHz ... 50：50MHz 128:128MHz Greater than 10000001: Disable. Minimum clock frequency of I2C bus: 1MHz in standard mode, and 4MHz in fast mode.

20.7.3 Slave mode address register 1 (I2C_SADDR1)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15	ADDRLEN	R/W	Slave Address Length Configure 0: 7-bit address mode 1: 10-bit address mode
14:10	Reserved		
9:8	ADDR[9:8]	R/W	Slave Address Setup When the address mode is 7 bits, the bit is invalid; when the address mode is 10 bits, this bit is the 9:8 bit of the address.
7:1	ADDR[7:1]	R/W	Slave Address Setup The 7:1 bit of slave address
0	ADDR[0]	R/W	Slave Address Setup When the address mode is 7 bits, the bit is invalid; when the address mode is 10 bits, this bit is the 0 bit of the address.

20.7.4 Slave mode address register 2 (I2C_SADDR2)

Offset address: 0x0C

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:8	Reserved		
7:1	ADDR2[7:1]	R/W	Slave Dual Address Mode Address Setup The 7:1 bit of the address in double-address mode

Field	Name	R/W	Description
0	ADDRNUM	R/W	Slave Address Number Configure In 7-bit address mode, the slave can be configured to identify the single-address mode and double-address mode; only ADDR1 is identified in single-address mode; both ADDR1 and ADDR2 can be identified in double-address mode Single or double-address registers can be identified in 7-bit address mode, specifically as follows: 0: Identify 1 address (ADDR1) 1: Identify 2 addresses (ADDR1 and ADDR2)

20.7.5 Data register (I2C_DATA)

Offset address: 0x10

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:8	Reserved		
7:0	DATA	R/W	Data Register In I2C transmission mode, write the data to be transmitted to this register; in I2C receiving mode, read the received data from this register.

20.7.6 Status register 1 (I2C_STS1)

Offset address: 0x14

Reset value: 0x0000 0080

Field	Name	R/W	Description
31:16	Reserved		
15	SMBALTFLG	RC_W0	SMBus Alert Occur Flag 0: SMBus master mode, without alarm; SMBus slave mode, without alarm, SMBAlert pin level unchanged 1: SMBus master mode, with an alarm generated on the pin; SMBus slave mode, receiving an alarm, causing SMBAlert pin level to become low This bit is set to 1 by hardware; this bit can be cleared by writing 0 by software; when I2CEN=0, it can be cleared to 0 by hardware.
14	TTEFLG	RC_W0	Timeout or Tlow Error Flag 0: No timeout error 1: When a timeout error occurs, in slave mode, the slave is reset and the bus is released; in master mode, the hardware transmits the stop bit. This bit can be set to 1 by hardware when timeout error occurs in any of the following situations: (1) SCL maintains low level for more than 25ms; (2) SCL low-level extension time of the master device is more than 10ms; (3) SCL low-level extension time of the slave device is more than 25ms. This bit can be cleared by writing 0 by software; and be cleared to 0 by hardware when I2CEN=0.

Field	Name	R/W	Description
13	Reserved		
12	PECEFLG	RC_W0	PEC Error in Reception Flag 0: No PEC error: when ACKEN=1, after PEC is received, the receiver will return ACKEN 1: There is PEC error: regardless of the value of ACKEN, as long as PEC is received, the receiver will return NACK This bit can be cleared by writing 0 by software; and be cleared to 0 by hardware when I2CEN=0.
11	OVRURFLG	RC_W0	Overrun/Underrun Flag 0: Not occur 1: Occurred This bit can be set to 1 by hardware when CLKSTRETCHD=1 and any of the following conditions is met: (1) In the slave receiving mode, when the data in the DATA register is not read out, but a new data is received (this data will be lost), overrun occurs; (2) In the slave transmission mode, no data is written in the data register but data still needs to be transmitted (the same data is transmitted twice), and then underrun occurs. This bit can be cleared by writing 0 by software; and be cleared to 0 by hardware when I2CEN=0.
10	AEFLG	RC_W0	Acknowledge Error Flag 0: No acknowledgment error 1: Acknowledgment error occurred This bit is set to 1 by hardware; this bit can be cleared by writing 0 by software; when I2CEN=0, it can be cleared to 0 by hardware.
9	ALFLG	RC_W0	Master Mode Arbitration Lost Flag 0: No arbitration loss 1: In case of arbitration loss, I2C interface will automatically switch back to slave mode "Arbitration loss in master mode" means the master loses the control of buses; this bit can be set to 1 by hardware; this bit can be cleared by writing 0 by software; when I2CEN=0, it can be cleared to 0 by hardware.
8	BERRFLG	RC_W0	Bus Error Flag 0: No bus error 1: Bus error occurred Bus error means exception of start bit or stop bit; when an error is detected, this bit can be set to 1 by hardware; this bit can be cleared by writing 0 by software; when I2CEN=0, it can be cleared to 0 by hardware.
7	TXBEFLG	R	Transmit Buffer Empty Flag 0: The transmit buffer is not empty 1: The transmit buffer is empty This bit can be set to 1 by hardware when the content of DATA register is empty;

Field	Name	R/W	Description
			When the software writes the first data to the DATA register, it will immediately move the data to the shift register, then the data in the DATA register is empty and this bit cannot be cleared; This bit can be cleared after the software writes data to DATA register; after transmitting the start bit and stop bit, or when I2CEN=0, it can be cleared to 0 by hardware.
6	RXBNEFLG	R	Receive Buffer Not Empty Flag 0: The receive buffer is empty 1: The receive buffer is not empty This bit can be set to 1 by hardware when there is data in DATA register; When BTCFLG is set to 1, since the data register is still full, the RXBNEFLG bit cannot be cleared by reading DATA register; This bit can be cleared after the software reads and writes DATA register; when I2CEN=0, it can be cleared to 0 by hardware.
5	Reserved		
4	STOPFLG	R	Stop Bit Detection Flag 0: Not detected 1: Detected If ACKEN=1, after one answer, when the slave detects the stop bit on the bus, it will be set to 1 by hardware; This bit can be cleared after the software first reads STS1 register and then writes the CTRL1 register; when I2CEN=0, it can be cleared to 0 by hardware.
3	ADDR10FLG	R	10-Bit Address Header Transmit Flag 0: Not transmit 1: Transmitted The bit can be set to 1 by hardware; this bit can be cleared after the software first reads STS1 register and then writes the DATA register; when I2CEN=0, it can be cleared to 0 by hardware.
2	BTCFLG	R	Byte Transfer Complete Flag 0: Not completed 1: Completed When receiving data, if the data received in DATA register fails to be read, and a new data is received then, it will be set to 1 by hardware; When transmitting data, if the DATA register is empty, it will be set to 1 by hardware to transmit the data in the shift register. This bit can be cleared after the software first reads STS1 register, and then reads or writes the DATA register; this bit can be cleared to 0 by hardware by transmitting a start bit and stop bit during the transmission, or when I2CEN=0.
1	ADDRFLG	R	Address Transfer Complete /Receive Match Flag Whether the matching address is received in slave mode: 0: Not received 1: Received Whether master mode address transmission is completed: 0: Not completed

Field	Name	R/W	Description
			1: Completed The bit can be set to 1 by hardware; this bit can be cleared after the software first reads STS1 register and then reads STS2 register; when I2CEN=0, it can be cleared to 0 by hardware.
0	STARTFLG	R	Start Bit Sent Finished Flag 0: Not transmit 1: Transmitted When the start bit is transmitted, this bit can be set to 1 by hardware; this bit can be cleared after the software first reads STS1 register and then writes the DATA register; when I2CEN=0, it can be cleared to 0 by hardware.

20.7.7 Status register 2 (I2C_STS2)

Offset address: 0x18

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15:8	PECVALUE[7:0]	R	Save Packet Error Checking Value When PECEN=1, the internal PEC value is saved in PECVALUE.
7	DUALADDRFLG	R	Slave Mode Received Dual Address Match Flag 0: The received address matches the content of ADDR1 register 1: The received address matches the content of ADDR2 register This bit can be set to 1 by hardware; and cleared to 0 by hardware when any of the following conditions is met: (1) Stop bit is generated (2) Repeated start bit is generated (3) I2CEN=0
6	SMMHADDR	R	SMBus Received Master Header Flag in Slave Mode 0: Failed to receive the master head address 1: Received the master head address when SMBTCFG =1 and ARPEN=1 This bit can be set to 1 by hardware; and cleared to 0 by hardware when any of the following conditions is met: (1) Stop bit is generated (2) Repeated start bit is generated (3) I2CEN=0
5	SMBDADDRFLG	R	SMBus Received Default Address Flag in Slave Mode Device 0: Failed to receive the default address 1: Received the default address when ARPEN=1 This bit can be set to 1 by hardware; and cleared to 0 by hardware when any of the following conditions is met: (1) Stop bit is generated (2) Repeated start bit is generated (3) I2CEN=0
4	GENCALLFLG	R	Slave Mode Received General Call Address Flag

Field	Name	R/W	Description
			0: Failed to receive the broadcast address 1: Received the Broadcast address This bit can be set to 1 by hardware; and cleared to 0 by hardware when any of the following conditions is met: (1) Stop bit is generated (2) Repeated start bit is generated (3) I2CEN=0
3	Reserved		
2	TRFLG	R	Transmitter / Receiver Mode Flag 0: The device is in receiver mode (read) 1: The device is in transmitter mode (write) Decide the bit value according to R/W bit; This bit can be cleared to 0 by hardware when any of the following conditions is met: (1) Stop bit is generated (2) Repeated start bit is generated (3) Bus arbitration is lost (4) I2CEN=0
1	BUSBSYFLG	R	Bus Busy Flag 0: The bus is idle (no communication) 1: The bus is busy (in the progress of communication) This bit can be set to 1 by hardware when SDA or SCL is at low level; and cleared to 0 by hardware after the stop bit is generated.
0	MSFLG	R	Master Slave Mode Flag 0: Slave mode 1: Master mode This bit can be set to 1 by hardware when I2C is configured as master mode; This bit can be cleared to 0 by hardware when any of the following conditions is met: (1) Stop bit is generated (2) Bus arbitration is lost (3) I2CEN=0

20.7.8 Slave clock control register (I2C_CLKCTRL)

Offset address: 0x1C

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15	SPEEDCFG	R/W	Master Mode Speed Configure 0: Standard mode 1: Fast mode

Field	Name	R/W	Description
14	FDUTYCFG	R/W	Fast Mode Duty Cycle Configure Duty cycle = t_{low}/t_{high} 0: SCLK duty cycle is 2 1: SCLK duty cycle is 16/9
13:12	Reserved		
11:0	CLKS [11:0]	R/W	Clock Setup in Fast/Standard Master Mode In I2C standard mode or SMBus mode: T_{high}=CLKS × T_{PCLK1} T_{low}=CLKS × T_{PCLK1} In I2C fast mode: When FDUTYCFG=0: T_{high}=CLKS×T_{PCLK1} T_{low}=2×CLKS× T_{PCLK1} When FDUTYCFG=1: T_{high}=9 × CLKS × T_{PCLK1} T_{low}=16 × CLKS × T_{PCLK1}

20.7.9 Maximum rising time register (I2C_RISETMAX)

Offset address: 0x20

Reset value: 0x0000 0000

Field	Name	R/W	Description
15:8	Reserved		
7:0	RISETMAX	R/W	Master Mode Maximum Rise Time in Fast/Standard Mode The time is in T_{PCLK1}, and RISETMAX is the maximum rising time unit of SCL plus 1.

20.7.10 Filter control register (I2C_FILTER)

Offset address: 0x24

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:5	Reserved		
4	ANFDIS	R/W	Analog Noise Filter Disable 0: Enable 1: Disable
3:0	DNFCFG	R/W	Digital Noise Filter Filtering Capability Configure 0000: Disable 0001：1 × T_{PCLK1} …… 1111：15 × T_{PCLK1} Note: These bits can be configured only when I2CEN=0.

21 Serial peripheral interface (SPI)

21.1 Full Name and Abbreviation of Terms

Table 71 Full Name and Abbreviation Description of Terms

Full name in English	English abbreviation
Most Significant Bit	MSB
Least Significant Bit	LSB
Master Out Slave In	MOSI
Master In Slave Out	MISO
Serial Clock	SCK
Serial Data	SD
Master Clock	MCK
Word Select	WS
Pulse-code Modulation	PCM
Transmit	TX
Receive	RX
Busy	BSY

21.2 Introduction

Serial Peripheral Interface (SPI) provides data transmission and reception capabilities based on the SPI protocol, allowing a chip to communicate with external devices in half-duplex or full-duplex, synchronous and serial modes, and can operate in either master or slave mode.

21.3 Main characteristics

(1) Master and slave operation with 3-wire full duplex synchronous transmission and receiving

(2) Simplex synchronous transmission can be realized by two wires (the third bidirectional data line can be included/not included)

(3) Select 8-bit or 16-bit transmission frame format

(4) Support multi-master device mode

(5) Support special transmission and receiving mark and can trigger interrupts

(6) Have SPI bus busy state flag

(7) The communication rate of SPI is up to 50Mbit/s

(8) Programmable clock polarity and phase

(9) Data sequence is programmable; select MSB or LSB in front

(10) An interrupt can be triggered by master mode fault, overrun and CRC error flag

(11) Have DMA transmit and receive buffers

(12) SPI TI mode

(13) Compatible with BiSS-C Slave interface

(14) Calculate, transmit and verify by hardware CRC

21.3.1 Main characteristics of I2S

(1) Have master/slave mode of simplex communication (transmit/receive only)

(2) Four audio standards
- I2S Philips standard
- MSB alignment standard
- LSB alignment standard
- PCM standard

(3) 16/24/32-bit data length can be selected

(4) 16-bit or 32-bit channel length

(5) Clock polarity is programmable

(6) 16-bit data register is used for transmitting and receiving

(7) MSB is always the first in the data direction

(8) Transmitting and receiving supports DMA function

21.4 SPI functional description

21.4.1 Description of SPI signal line

Table 72 SPI Signal Line Description

Pin Name	Description
SCK	Master device: SPI clock outputs Slave device: SPI clock inputs
MISO	Master device: Input the pin and receive data Slave device: Output the pin and transmit data Data direction: From slave device to master device

Pin Name	Description
MOSI	Master device: Output the pin and transmit data Slave device: Input the pin and receive data Data direction: From master device to slave device
NSS	Software NSS mode: NSS pin can be used for other purposes. NSS mode of master device hardware: NSS output, single master mode. Disable NSS output: Operation of multiple master environments is allowed. NSS mode of slave device hardware: NSS signal is set to low level as chip selection signal of slave.

21.4.2 Phase and polarity of clock signal

The clock polarity and clock phase are CPOL and CPHA bits of SPI_CTRL1 register.

Clock polarity CPOL means the level signal of SCK signal line when SPI is in idle state.

- When CPOL=0, SCK signal line is low in idle state
- When CPOL=1, SCK signal line is high in idle state

Clock phase CPHA means the sampling moment of data:

- When CPHA=0, the signal on MOSI or MISO data line will be sampled by the "odd edge" on SCK clock line.
- When CPHA=1, the signal on MOSI or MISO data line will be sampled by the "even edge" on SCK clock line.

SPI can be divided into four modes according to the states of clock phase CPHA and clock polarity CPOL.

Table 73 Four Modes of SPI

SPI mode	CPHA	CPOL	Sampling moment	Idle SCK clock
0	0	0	Odd edge	Low level
1	0	1	Odd edge	High Level
2	1	0	Even edge	Low level
3	1	1	Even edge	High Level

21.4.3 Data frame format

Set MSB or LSB to be first by configuring LSBSEL bit of SPI_CTRL1 register. Select to transmit/receive in 8/16-bit data frame format by configuring DFLSEL bit of SPI_CTRL1 register.

21.4.4 NSS mode

Software NSS mode: Select to enable or disable this mode by configuring SSEN bit of SPI_CTRL1 register, and the internal NSS signal level is driven by ISSEL bit of SPI_CTRL1 register.

Hardware NSS mode:

- Enable NSS output: When SPI is in master mode, enable SSOEN bit, NSS pin will be pulled to low and SPI will automatically enter the slave mode.
- Disable NSS output: Operation is allowed in multi-master environments.

21.4.5 SPI mode

21.4.5.1 **SPI master mode**

In master mode, generate serial clock on SCK pin.

Master mode configuration:

- Configure MSMSEL=1 in SPI_CTRL1 register
- Select the polarity and phase by configuring CPOL and CPHA bits of SPI_CTRL1 register.
- Select 8/16-bit data frame format by configuring DFLSEL bit of SPI_CTRL1 register
- Select LSB or MSB first by configuring LSBSEL in SPI_CTRL1 register
- NSS configuration:
  - NSS pin works in input mode: in hardware mode, it is required to connect NSS pin to high level during the entire data frame transmission; in software mode, it is required to set SSEN bit and ISSEL bit in SPI_CTRL1 register
  - NSS works in output mode and it is required to configure SSOEN bit of SPI_CTRL2 register
- Configure FRFCFG bit of SPI_CTRL2 register to select TI mode protocol for serial communication
- Enable SPI by configuring SPIEN bit in SPI_CTRL1 register

In master mode: MOSI pin is data output, while MISO is data input

TI protocol

In slave mode, SPI interface supports TI protocol. It is controlled by FRFCFG bit of SPI_CTRL2 register. Both clock polarity and phase conform to TI protocol. NSS management is specific to TI protocol, not needing to configure SPI_CTRL1 and SPI_CTRL2 registers.

21.4.5.2 **SPI slave mode**

In slave mode, SCK pin receives the serial clock transmitted from the master device.

Configuration of slave mode:

- Configure MSMSEL=0 in SPI_CTRL1 register

- Select the polarity and phase by configuring CPOL and CPHA bits of SPI_CTRL1 register.
- Select 8/16-bit data frame format by configuring DFLSEL bit of SPI_CTRL1 register
- Select LSB or MSB first by configuring LSBSEL in SPI_CTRL1 register
- NSS configuration:
  - In hardware mode: NSS pin must be low in the whole data frame transmission process
  - In software mode: Set SSEN bit in SPI_CTRL1 register and clear ISSEL bit (this step is not required for TI mode)
- Configure FRFCFG bit of SPI_CTRL2 register to select TI mode protocol for serial communication
- Enable SPI by configuring SPIEN bit in SPI_CTRL1 register

In slave mode: MOSI pin is data input, while MISO is data output

TI protocol

In slave mode, SPI interface supports TI protocol. It is controlled by FRFCFG bit of SPI_CTRL2 register. Both clock polarity and phase conform to TI protocol. NSS management is specific to TI protocol, not needing to configure SPI_CTRL1 and SPI_CTRL2 registers.

In slave mode, SPI baud rate prescaler can use any baud rate to control the moment of switching MISO pin state to high-impedance state, so it can determine this moment very flexibly. The baud rate is generally the baud rate of external master clock. The baud rate value set by BRSEL[2:0] of SPI_CTRL1 register and the internal circuit of the chip synchronously determine the time when the MISO pin state changes to high-impedance state.

21.4.5.3 Half-duplex communication of SPI

One clock line and one bidirectional data line

- Enable this mode by setting BMEN of SPI_CTRL1 register
- Control the data line to be input or output by setting BMOEN bit of SPI_CTRL1 register
- SCK pin is used as clock, MOSI pin is used in master device to transmit data, and MISO pin is used in slave device to transmit data

21.4.6 Data transmission and receiving process in different modes of SPI

Table 74 Run Mode of SPI

Mode	Configure	Data pin
Full-duplex mode of master device	BMEN=0，RXOMEN=0	MOSI transmits; MISO receives
Unidirectional receiving mode of master device	BMEN=0，RXOMEN=1	MOSI is not used; MISO receives
Bidirectional transmitting mode of master device	BMEN=1，BMOEN=1	MOSI transmits; MISO is not used
Bidirectional receiving mode of master device	BMEN=1，BMOEN=0	MOSI is not used; MISO receives
Full-duplex mode of slave device	BMEN=0，RXOMEN=0	MOSI receives, and MISO transmits
Unidirectional receiving mode of slave device	BMEN=0，RXOMEN=1	MOSI receives, and MISO is not used
Bidirectional transmitting mode of slave device	BMEN=1，BMOEN=1	MOSI is not used, and MISO transmits
Bidirectional receiving mode of slave device	BMEN=1，BMOEN=0	MOSI receives, and MISO is not used

Figure 80 Connection in Full-duplex Mode

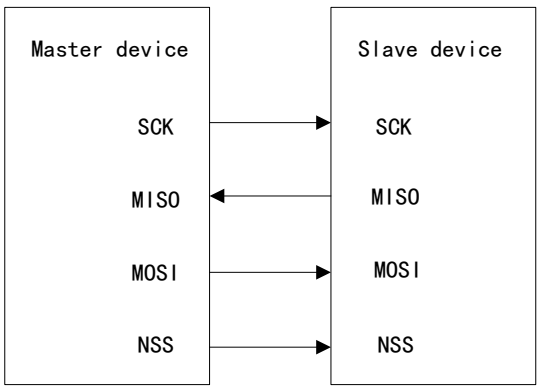


Figure 81 Connection in Half-duplex Mode (the master receives, while the slave transmits)

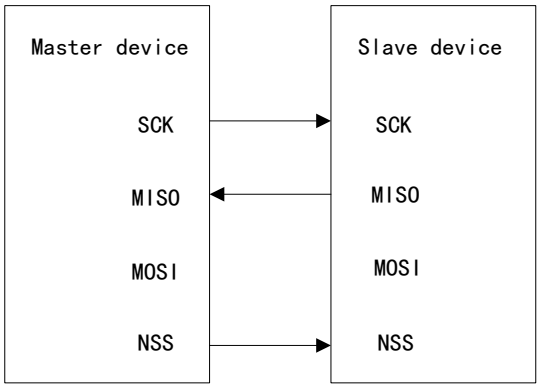


Figure 82 Connection in Half-duplex Mode (the master only transmits, while the slave receives)

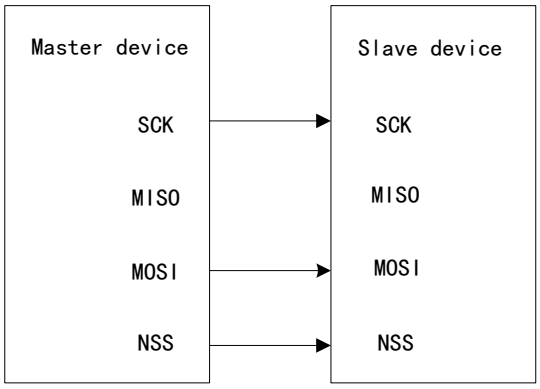
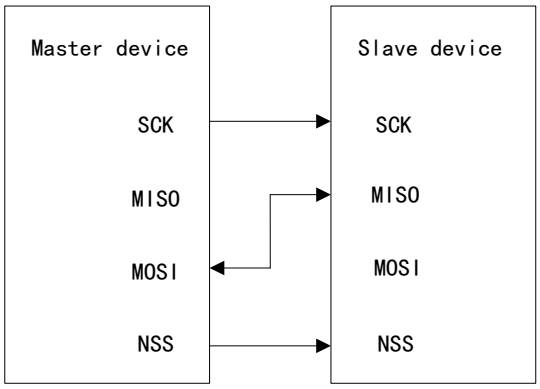


Figure 83 Bidirectional Line Connection



21.4.6.1 **Transmitting and receiving of processed data**

Data transmission

After the mode configuration is completed, the SPI module is enabled to remain idle.

Master mode: The software writes a data frame to the transmit buffer, and the transmission process starts

Slave mode: The SCK signal on the SCK pin starts to jump, while the NSS pin level is low, and the transmission process starts (before starting data transmission, make sure that the data has been written to the transmit buffer in advance).

When SPI is transmitting a data frame, it will load the data frame from the data buffer to the shift register, and then start to transmit data. After one bit of data frame is transmitted, TXBEFLG in SPI_STS register is set to 1. To continue to transmit data, the software needs to wait and when TXBEFLG=1 data will be written to the SPI_DATA register. (TXBEFLG flag is set to 1 by hardware and cleared to 0 by software).

Data receiving

BSYFLG flag is always set to 1 in the data receiving process.

At the last edge of the sampling clock, the received data is transmitted from the shift register to the receive buffer; set the RXBNEFLG flag in SPI_STS register, and the software reads the data in data register (SPI_DATA) to obtain the content of the receive buffer; if RXBNEIEN bit of SPI_CTRL2 register is set, an interrupt will be generated, and after data is read, the BSYFLG flag will be automatically cleared.

21.4.6.2 Full duplex transmitting and receiving mode under master/slave device

Full duplex mode under master device

- After writing data to SPI_ DATA register (transmit buffer), data transmission starts.
- When SPI transmits the first bit of data, the data is transmitted from the transmit buffer to the shift register and then transmitted to the MOSI pin serially according to the sequence.
- The data received on MISO pin is serially transmitted to SPI_DATA register (receive buffer) according to the sequence.

Transmitting and receiving are synchronous.

Full duplex mode under slave device

- When the slave device receives the clock signal and the first data bit appears on the MOSI pin, data transmission starts, and the subsequent data bits will be transmitted to the shift register in turn.
- When SPI transmits the first bit of data, the data is transmitted from the transmit buffer to the shift register and then transmitted to the MISO pin serially according to the sequence.

- The software must ensure that the data to be transmitted has been written before the SPI master device starts to transmit data.

Transmitting and receiving are synchronous.

Full duplex transmitting and receiving process under master/slave device

(1) Enable SPI module: Configure SPIEN=1 for SPI_CTRL1 register.

(2) Write the first data to be transmitted to SPI_DATA register, and the TXBEFLG flag will be cleared.

(3) Wait until TXBEFLG flag bit is set to 1 (controlled by hardware), and write the second data bit to be transmitted.

(4) Wait until RXBNEFLG flag bit is set to 1 (controlled by hardware), read the first received data in the SPI_DATA register, at the same time, clear the RXBNEFLG flag (cleared to 0 by software). Repeat the operation, and transmit and receive data at the same time.

(5) Wait until RXBNEFLG=1 and receive the last data.

(6) Wait until TXBEFLG=1 and disable SPI module after BSYFLG=0.

21.4.6.3 **Bidirectional transmission mode under master/slave device**

Bidirectional transmission under master device

- After the data is written to SPI_DATA register, start transmission
- The data in the transmit buffer is transmitted to the shift register in parallel, and then transmitted to the MOSI pin serially according to the sequence.

Bidirectional transmission under slave device

- When the slave device receives the clock signal and the first data bit appears on the MISO pin, data transmission starts.
- At the same time, the data to be transmitted by the transmit buffer is transmitted to the shift register in parallel, and then transmitted to the MISO pin serially (before data transmission, make sure that the data has been written to the transmit buffer in advance).

Bidirectional transmission process under master/slave device

(1) Enable SPI module: Configure SPIEN=1 for SPI_CTRL1 register.

(2) Write the first data to be transmitted to SPI_DATA register, and the TXBEFLG flag will be cleared.

(3) Wait until TXBEFLG=1, write the second data, repeat the operation and transmit the subsequent data

(4) After the last data is written, wait until TXBEFLG=1 and BSYFLG=0 and transmission is completed

21.4.6.4 Unidirectional/Bidirectional receiving mode under master/slave device

(1) Enable SPI module: Configure SPIEN=1 for SPI_CTRL1 register.

(2) In the master device: Generate SCK clock immediately, and continuously receive data before SPI is disabled.

(3) Slave device: When SPI master device pulls down NSS and generates a clock, receive data.

(4) Wait until the RXBNEFLG flag is set to 1, read data through SPI_DATA, and repeat the operation to receive data.

21.4.7 CRC functions

SPI module contains two CRC computing units, which are used for data receiving and data transmission respectively.

CRC computing unit is used to define polynomials in SPI_CRCPOLY register.

Enable CRC computing by configuring CRCEN bit in SPI_CTRL1 register; at the same time, reset the CRC register (SPI_RXCRC and SPI_TXCRC).

To obtain the CRC value of transmission calculation, after the last data is written to the transmit buffer, it is required to set CRCNXT bit of SPI_CTRL1; indicate that the hardware transmits the CRC value after the last data is transmitted, and the CRCNXT bit will be cleared; at the same time, compare the values of CRC and SPI_RXCRC, and if they do not match, it is required to set CRCEFLG bit of SPI_STS register, and after ERRIEN bit of SPI_CTRL2 register is set, an interrupt will occur.

Note:

(1) If SPI is under slave device and CRC function is used, CRC computing will continue when NSS pin is at high level. For example, when the master device communicates with multiple slave devices alternately, the above situation will occur, so it is necessary to avoid faulty operation of CRC.

(2) In the process of a slave device from being unselected (NSS is at high level) to being selected (NSS is at low level 0), it is required to clear the CRC value at both ends of the master and slave devices to keep the next CRC computing results of the master and slave devices synchronized.

(3) When SPI is in slave mode, CRC computing can be enabled after the clock becomes stable.

(4) When the SPI clock frequency is too high, the CPU operation will affect the SPI bandwidth. It is recommended to use DMA mode to avoid reduction of SPI speed.

(5) When the SPI clock frequency is too high, during the CRC transmission period, the CPU utilization frequency will be reduced, and the function call is disabled in the CRC transmission process to avoid errors when receiving the last data and CRC.

(6) When NSS hardware mode is used in slave mode, NSS pin should be kept low during data transmission and CRC transmission period.

Sequence of clearing CRC values

(1) Disable SPI (SPIEN=0)

(2) Clear CRCEN bit

(3) Set CRCEN bit to 1

(4) Enable SPI (SPIEN=1)

21.4.8 DMA function

The request/response DMA mechanism in SPI facilitates high-speed data transmission, improves the system efficiency and enable to transfer data to SPI transmit buffer promptly, and the receive buffer can read the data in time to prevent overrun.

When SPI only transmits data, it is only necessary to enable DMA transmission channel; when SPI only receives data, it is only necessary to enable DMA receiving channel.

DMA function of SPI mode can be enabled by configuring TXDEN and RXDEN bits of SPI_CTRL2 register.

- During transmitting: When TXBEFLG flag bit is set to 1, issue the DMA request, DMA controller writes data to SPI_DATA register, and then the TXBEFLG flag bit will be cleared.
- When receiving: When setting RXBNEFLG flag bit to 1, issue the DMA request, DMA controller reads data from SPI_DATA register, and then RXBNEFLG flag bit is cleared.

By monitoring BSYFLG flag bit, confirm whether SPI communication is over after DMA has transferred all data to be transmitted in transmitting mode, which can avoid damaging the transmission of last data.

DMA function with CRC

By the end of communication, if SPI enables both CRC operation and DMA function, transmitting and receiving of CRC bytes will be completed automatically.

At the end of data and CRC transmission, if CRCEFLG flag bit of SPI_STS register is set to 1, it indicates that an error occurred during transmission.

21.4.9 Disable SPI

After data transmission is over, end the communication by disabling SPI module. In some configurations, if SPI is disabled before data transmission is completed, a data transmission error may be caused. Different methods are required in different operation modes to disable SPI

Maser mode/Full-duplex slave mode

(1) Wait until RXBNEFLG flag bit is set to 1, and receive the last data

(2) Wait until TXBEFLG flag bit is set to 1

(3) Wait for clearing BSYFLG flag bit to 0

(4) Disable SPI (set SPIEN=0 for SPI_CTRL1 register)

Unidirectional transmit-only/Bidirectional transmitting mode of master mode/slave mode

After the last data is written into SPI_DATA register:

(1) Wait until TXBEFLG flag bit is set to 1

(2) Wait for clearing BSYFLG flag bit to 0

(3) Disable SPI (set SPIEN=0 for SPI_CTRL1 register)

Unidirectional receive-only/bidirectional receiving mode of master mode/slave mode

(1) Wait until No. n-1 RXBNEFLG flag bit is set to 1

(2) Wait for one SPI clock cycle before SPI is disabled (set SPIEN=0 for SPI_CTRL1 register)

(3) Before entering the stop mode, wait until the last RXBNEFLG flag bit is set to 1

Receive-only/bidirectional receiving mode in slave mode

SPI can be disabled at any time (set SPIEN=0 for SPI_CTRL1 register) and it will be disabled when the transmission is over. To enter the stop mode, wait until BSYFLG flag bit is cleared to 0.

21.4.10 SPI interrupt

21.4.10.1 **Status flag bit**

Transmit buffer idle flag TXBEFLG

TXBEFLG=1 indicates that the transmit buffer bit is empty, and the next data to

be transmitted can be written. When the data is written to SPI_DATA register, clear the TXBEFLG flag bit.

Receive buffer non-empty flag RXBNEFLG

RXBNEFLG=1 indicates that the receive buffer contains valid data and the data can be read through SPI_DATA register; and the RXBNEFLG flag can be cleared.

Busy flag BSYFLG

BSYFLG flag is set and cleared by hardware, which can indicate the state of SPI communication layer. BSYFLG=1 indicates SPI is communicating, but in the two-line receiving mode under the master device, BSYFLG=0 during the period of receiving data.

BSYFLG flag can be used to detect whether transmission is over to avoid destroying the last transmitted data.

BSYFLG flag bit can be used to avoid conflict when writing data in multi-master mode.

BSYFLG flag will be cleared to 0 when the transmission ends (except for continuous communication in master mode), SPI is disabled and the master mode fails.

BSYFLG=0 between data item and data item when communication is discontinuous.

When communication is continuous:

- In master mode: BSYFLG=1 in the whole transmission process
- In save mode: BSYFLG is kept low within one SCK clock cycle between transmission of data

Note: It is better to use TXBEFLG and RXBNEFLG flags to process the transmitting and receiving of each data item.

21.4.10.2 **Error flag bit**

Master mode error MEFLG

MEFLG is an error flag bit. The master mode error occurs when: in hardware NSS mode, the NSS pin of the master device is pulled down; in software NSS mode, ISSEL bit is cleared to 0; MEFLG bit is set automatically.

Influence of master mode failure: MEFLG is set to 1, and SPI interrupt is generated when ERRIEN is set; SPIEN is cleared to 0 (output stops, and SPI interface is disabled); MSMSEL is cleared to 0 and the device is forced to enter the slave mode.

Operation of clearing the MEFLG flag bit: When MEFLG bit is set to 1, it is required to read or write SPI_STS register, and then write to SPI_CTRL1 register.

When MEFLG flag bit is 1, it is not allowed to set SPIEN and MSMSEL bits.

Overrun error OVRFLG

Overrun error: After the master device transmits the data, the RXBNEFLG flag bit is still 1, which indicates that an overrun error occurred. Then OVRFLG bit in SPI_STS register is set to 1, and if the ERRIEN bit is also set, an interrupt will be generated.

After an overrun error occurs, the data in the receive buffer is not the data transmitted by the master device, then the read data in SPI_DATA register is the data not read before, while the data transmitted later will not be read.

OVRFLG flag can be cleared by reading SPI_DATA register and SPI_STS register according to the sequence.

CRC error flag CRCEFLG

By setting CRCEN bit of SPI_CTRL1 register, enable CRC computing, and CRC error flag can be used to check whether the received data is valid.

When the value transmitted by SPI_TXCRC register does not match the value in SPI_RXCRC register, a CRC error will be generated, and CRCEFLG flag bit in SPI_STS register will be set to 1.

CRCEFLG can be cleared by writing 0 to CRCEFLG bit of SPI_STS register.

TI frame format error flag FREFLG

If SPI supports TI protocol in slave mode, TI frame format error will be detected when NSS pulse occurs during communication. When this error appears, SPI_STS[FREFLG]=1, SPI will not be disabled, but NSS pulse will be ignored, and SPI will start new transmission when next NSS pulse arrives. As the error detection may cause the loss of two data bytes, the data may be damaged.

FREFLG flag will be cleared to 0 when reading SPI_STS register. If ERRIEN=1, and a frame format error is detected, an interrupt will be generated. The continuity of data cannot be guaranteed at this time, the SPI shall be disabled and after the slave SPI is enabled again, the master will restart the communication.

Table 75 SPI Interrupt Request

Interrupt flag	Interrupt event	Enable control bit	Clearing method
TXBEFLG	Transmit buffer empty flag	TXBEIEN	Write SPI_DATA register

Interrupt flag	Interrupt event	Enable control bit	Clearing method
RXBNEFLG	Receive buffer non-empty flag	RXBNEIEN	Read SPI_DATA register
MEFLG	Master mode failure event	ERRIEN	Read/Write SPI_STS register, and then write SPI_CTRL1 register
OVRFLG	Overrun error		Read SPI_DATA register, and then read SPI_STS register
CRCEFLG	CRC error flag		Write 0 to CRCEFLG bit
FREFLG	TI frame format error		Read SPI_STS register

21.5 BISS-C functional description

21.5.1 BISS-C data connection

In BISS-C mode, data from spi_dr begins transmission at the second rising edge. When spi_dr is empty and stop_en is set to 1, the rising edge after transmitting the current frame latches the CDM, pulls the slo output low, and simultaneously starts the timeout timer. When the timeout expires, slo is pulled high.

In SSI mode, data from spi_dr begins transmission at the first rising edge. When spi_dr is empty and stop_en is set to 1, after transmitting the current frame, the slo output is pulled low and the timeout timer starts. When the timeout expires, slo is pulled high.

21.6 Register address mapping

Table 76 SPI Register Address Mapping

Register name	Description	Offset Address
SPI_CTRL1	SPI control register 1	0x00
SPI_CTRL2	SPI control register 2	0x04
SPI_STS	SPI status register	0x08
SPI_DATA	SPI data register	0x0C
SPI_CRCPOLY	SPI CRC polynomial register	0x10
SPI_RXCRC	SPI receive CRC register	0x14
SPI_TXCRC	SPI transmit CRC register	0x18
SPI_BCR	BISS-C control Register	0x1C
SPI_BCFG	BISS-C configuration Register	0x20
SPI_CDMDATA	CDM data Register	0x24

21.7 Register functional description

These peripheral registers can be operated by half word (16 bits) or word (32 bits).

21.7.1 SPI control register 1 (SPI_CTRL1)

Offset address: 0x00

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15	BMEN	R/W	Bidirectional Mode Enable 0: Double-line unidirectional mode 1: Single-line bidirectional mode Single-line bidirectional transmission: the transmission between MOSI pin of data master and MISO pin of slave
14	BMOEN	R/W	Bidirectional Mode Output Enable 0: Disable, namely, receive-only mode 1: Enable, namely, transmit-only mode When BMEN=1, namely, in single-line/ bidirection mode, this bit decides the transmission direction of transmission line.
13	CRCEN	R/W	CRC Calculate Enable 0: Disable 1: Enable Note: The CRC check function applies only to full-duplex mode; this bit can only be changed when SPIEN=0.
12	CRCNXT	R/W	CRC Transfer Next Enable 0: The next transmitted data is from transmit buffer 1: The next transmitted data is from CRC register Note: After the last data is written to SPI_DATA register, set CRCNXT bit immediately.
11	DFLSEL	R/W	Data Frame Length Format Select 0: 8-bit data frame format 1: 16-bit data frame format Note: This bit can only be written when SPIEN=0 to change the data frame length.
10	RXOMEN	R/W	Receive Only Mode Enable 0: Transmit and receive at the same time 1: Receive-only mode RXOMEN bit and BMEN bit together determine the transmission direction in the two-line and two-way mode. In the configuration of multiple slave devices, in order to avoid data transmission collision, it is necessary to set RXOMEN bit to 1 on the slave devices that are not accessed.
9	SSEN	R/W	Software Slave Device Enable 0: Software NSS mode is disabled, and the internal NSS level is determined by external NSS pin 1: Software NSS mode is enabled, and the internal NSS level is determined by the ISSEL bit of SPI_CTRL1

Field	Name	R/W	Description
8	ISSEL	R/W	Internal Slave Device Select When CTRL1_SSEN=1 (software NSS mode), configure this bit to select internal NSS level 0: Internal NSS is low 1: Internal NSS is high
7	LSBSEL	R/W	LSB First Transfer Select 0: First transmit the most significant bit (MSB) 1: First transmit the least significant bit (LSB)
6	SPIEN	R/W	SPI Device Enable 0: Disable 1: Enable Note: When SPI device is disabled, operate according to the process of disabling SPI.
5:3	BRSEL	R/W	Baud Rate Divider Factor Select 000：DIV=2 001：DIV=4 010：DIV=8 011：DIV=16 100：DIV=32 101：DIV=64 110：DIV=128 111：DIV=256 Baud rate=F_{PCLK}/DIV Note: This bit cannot be modified during communication
2	MSMCFG	R/W	Master/Salve Mode Configure 0: Configure as slave mode 1: Configure as master mode Note: This bit cannot be modified during communication
1	CPOL	R/W	Clock Polarity Configure The level state maintained by SCK when SPI is in idle state. 0: Low level 1: High level Note: This bit cannot be modified during communication
0	CPHA	R/W	Clock Phase Configure This bit indicates on the edge of which clock to start sampling 0: On the edge of the first clock 1: On the edge of the second clock Note: This bit cannot be modified during communication

21.7.2 SPI control register 2 (SPI_CTRL2)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:8	Reserved		

Field	Name	R/W	Description
7	TXBEIEN	R/W	Transmit Buffer Empty Interrupt Enable 0: Disable 1: Enable When TXBEFLG fag bit is set to 1, an interrupt request will be generated
6	RXBNEIEN	R/W	Receive Buffer Not Empty Interrupt Enable 0: Disable 1: Enable When RXBNEFLG flag bit is set to 1, an interrupt request will be generated.
5	ERRIEN	R/W	Error interrupt Enable 0: Disable 1: Enable When an error occurs, ERRIEN bit controls whether to generate the interrupt.
4	FRFCFG	R/W	Frame Format Configure 0: SPI Motorola mode 1: SPI TI mode
3	Reserved		
2	SSOEN	R/W	SS Output Enable SS output in master mode 0: Disable SS output, and it can work in multi-master mode. 1: Enable SS output, and it cannot work in multi-master mode.
1	TXDEN	R/W	Transmit Buffer DMA Enable When this bit is set, once TXBEFLG flag is set, DMA request will be issued. 0: Disable 1: Enable
0	RXDEN	R/W	Receive Buffer DMA Enable When RXDEN=1, once RXBNEFLG flag is set, DMA request will be issued. 0: Disable 1: Enable

21.7.3 SPI status register (SPI_STS)

Offset address: 0x08

Reset value: 0x0000 0002

Field	Name	R/W	Description
31:11	Reserved		
10	BSBUSY	R	Busy in BISS-C/SSI mode
9	CDM_NE	R	CDM non-empty status
8	FREFLG	R	Frame Format Error Flag 0: Not occur 1: Occurred

Field	Name	R/W	Description
			Note: This flag is used when working in TI slave mode. This bit is set to 1 by hardware and can be cleared to 0 when reading SPI_STS register.
7	BSYFLG	R	SPI Busy Flag 0: SPI is idle 1: SPI is communicating It is set or cleared by hardware.
6	OVRFLG	R	Overrun Occur Flag 0: Not occur 1: Occurred This bit is set by hardware, and reading the SPI_DATA register and the SPI_STS register in sequence can clear this bit.
5	MEFLG	R	Mode Error Occur Flag 0: Not occur 1: Occurred This bit is set by hardware. To clear it via software, a read or write operation to this register must be performed first, followed by a write operation to SPI_CTRL1.
4	CRCEFLG	RC_W0	CRC Error Occur Flag This bit indicates whether the received CRC value matches the value of RXCRC register 0: Match 1: Not match This bit is set by hardware, can be cleared by writing 0 to this bit by software.
3:2	Reserved		
1	TXBEFLG	R	Transmit Buffer Empty Flag 0: Not empty 1: Empty
0	RXBNEFLG	R	Receive Buffer Not Empty Flag 0: Empty 1: Not empty

21.7.4 SPI data register (SPI_DATA)

Offset address: 0x0C

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15:0	DATA	R/W	Transmit Receive Data register When writing this register, the data will be written to the transmit buffer; when reading this register, the data in receive buffer will be read. The size of the buffer is consistent with the length of the data frame, that is, for 8-bit data, only DATA[7:0] will be used when transmitting and receiving data, and DATA[15:8] is invalid; for 16-bit data, DATA[15:0] will be used when transmitting and receiving data. In BISS-C/SSI mode, only DATA[7:0] is used for transmitting data.

21.7.5 SPI CRC polynomial register (SPI_CRCPOLY)

Offset address: 0x10

Reset value: 0x0000 0007

Field	Name	R/W	Description
31:16	Reserved		
15:0	CRCPOLY	R/W	CRC Polynomial Value Setup This register contains CRC polynomial of CRC computing, which can be modified, and the reset value is 0x0007.

21.7.6 SPI receive CRC register (SPI_RXCRC)

Offset address: 0x14

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15:0	RXCRC	R	Receive Data CRC Value The CRC data of the data received and calculated by hardware are stored in this register; the bits and the length of data frames are consistent, that is, if the received data are 8 bits, the CRC computing is made based on CRC8; if the received data are 16 bits, the CRC computing is made based on CRC16. When CRCEN is set, the hardware clears the register. Note: When BSYFLG bit is set to 1, the value of reading RXCRC register may be wrong.

21.7.7 SPI transmit CRC register (SPI_TXCRC)

Offset address: 0x18

Reset value: 0x0000

Field	Name	R/W	Description
31:16	Reserved		
15:0	TXCRC	R	Transmit Data CRC Value The CRC data of transmitted data calculated by hardware is stored in this register; the bits and the length of data frames are consistent, that is, if the transmitted data are 8 bits, the CRC computing is made based on CRC8; if the transmitted data is are 16 bits, the CRC computing is made based on CRC16. When CRCEN is set, the hardware clears the register. Note: When BSYFLG bit is set to 1, the value of reading RXCRC register may be wrong.

21.7.8 BISS-C control register (SPI_BCR)

Offset address: 0x1C

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:5	Reserved		
4	CDM_DMAEN	R/W	DMA Request Enable When CDM Received in BISS Mode 0: Disable

Field	Name	R/W	Description
			1: Enable
3	CDM_IE	R/W	Interrupt Enable When CDM Received in BISS Mode 0: Disable 1: Enable
2	STOP_EN	R/W	Stop_EN Enable 0: Disable 1: Enable
1	SSI_EN	R/W	SSI Mode Enable 0: Disable 1: Enable
0	BISS_EN	R/W	BISS-C Mode Enable 0: Disable 1: Enable

21.7.9 BISS-C configuration register（SPI_BCFG）

Offset address: 0x20

Reset value: 0x0000 2850

Field	Name	R/W	Description
31:14	Reserved		
13:8	FREQ_NUM[5:0]	R/W	Timeout Period When the timeout time exceeds the value, a timeout occurs with the base time unit in microseconds. The timeout duration is configured according to the actual situation; for example, a 40 μs timeout is configured as 6'h28.
7:0	FREQ_NUM[7:0]	R/W	Current SPI Module System Clock It is used to generate baseline clock for timeout. The clock is configured according to the actual situation; for example, if the clock is 80MHz, configure as 8'h50.

21.7.10 CDM data register（SPI_ CDMDATA）

Offset address: 0x24

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:1	Reserved		
0	DATA	R	CDM Data Received in BISS-C Mode

22 16-bit Analog-to-Digital Converter (16-bit ADC)

22.1 Full Name and Abbreviation Description of Terms

Table 77 Full Name and Abbreviation Description of ADC Terms

Full name in English	English abbreviation
Analog watchdog	AWD
Conversion	C
Injected	INJ
Regular	REG
Start	S
Scan	SCAN
Single	SINGLE
Automatic	A
Group	G
Discontinuous	DISC
Count	CNT
Dual	DUAL
Continuous	C
Calibration	CAL
Reset	RST
Alignment	ALIGN
External	EXT
Event	E
Trigger	TRG
Temperature	T
Sensor	S
Time	TIM
Sample	SMP
Offset	OF
High	H
Low	L
Threshold	T
Sequence	SEQ

Full name in English	English abbreviation
Length	LEN
Regular Channels	REG
Injected Channel	INJ
Injected Group	INJG
Automatic	A
Conversion	C
Analog Watchdog	AWD
Discontinuous Mode	DISC
Scan Mode	SCAN
Continuous Conversion	CONTC
Single Conversion	SINGLEC
External	EXT
External Trigger	EXTTRG
Sample Time	SMPTIM
Sequence	SEQ
Number	NUM

22.2 Introduction

The series product features two 16-bit precision successive approximation register (SAR) ADC. Each ADC has up to six external channels. The A/D conversion modes for each channel include single, continuous, scan, or discontinuous modes. The conversion results in dual-ADC synchronous mode can be stored in the 32-bit data register of master ADC (ADC1).

22.3 Main characteristics

(1) 16-bit resolution, ENOB 13.5 bits

(2) Built-in high-precision reference source: 1.65V±2mV (at 25°C)

(3) Maximum sampling rate of 1 Msps at 16-bit resolution

(4) Supports single-ended and differential mode input. Number of sampling channels (differential/single-ended): Up to 6 pairs of differential/ 12 single-ended channels.

(5) Supports regular sequence, injected sequence, single, and continuous sampling modes

(6) The two ADC support master-slave mode, and sampling synchronization is maintained when it is configured as master-slave mode.

(7) Supports up to 16x oversampling rate

(8) Trigger modes: Dual ADC synchronous sampling can be triggered by software or hardware. Supports multiple internal or external trigger inputs. Each ADC module has trigger mode selection:
- On-chip timer signal trigger
- External pin signal trigger
- Software trigger

(9) Supports external reference voltage pins, allowing the input voltage range to be independent of the power supply.

(10) Highly universal configuration
- Supports scan mode for single or continuous/discontinuous sequences; each ADC can convert multiple channels or scan a sequence of channels
- The ADC conversion results can be stored in 16/32-bit result registers or transferred directly to RAM via DMA
- Data pre-processing: Offset compensation per channel
- Programmable sampling time for channels, providing flexible sampling time control
- Analog watchdog, used for automatic voltage monitoring, generation of interrupts and triggers for selected timers

22.4 Functional description

22.4.1 Internal block diagram

The figure below shows the internal block diagram of the ADC module, which illustrates the interconnections of various functional modules.

Figure 84 Internal Block Diagram of 16-bit ADC Module

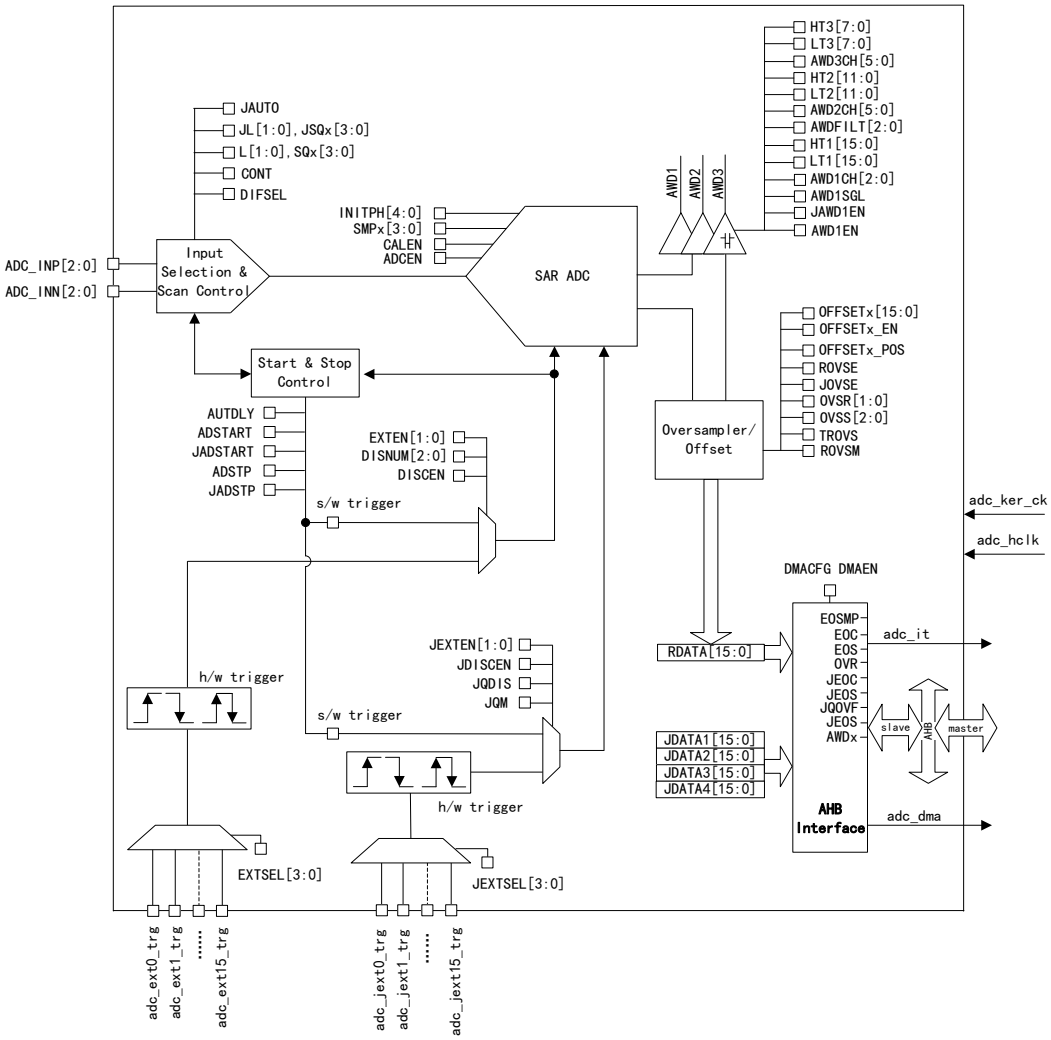
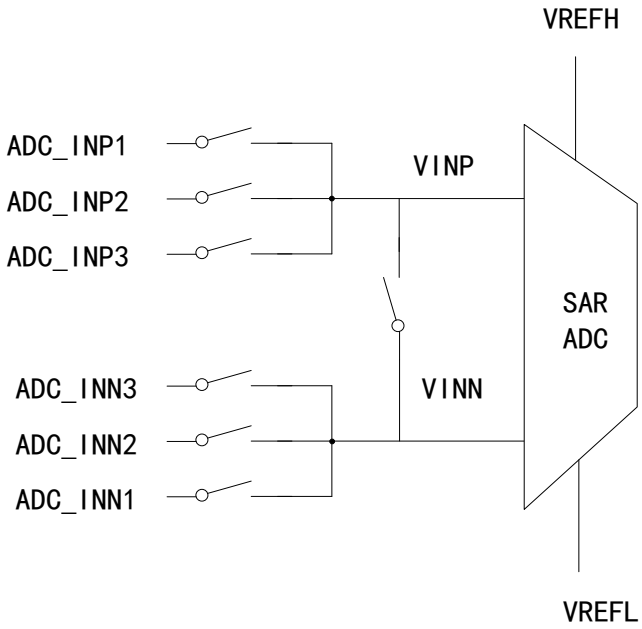


Figure 85 ADC Connection Function



22.4.2 Signals and pins

Table 78 ADC Internal Input/Output Signals

Internal signal name	Signal type	Description
EXT[15:0]	Input	There are up to 16 external trigger inputs used for conventional conversion (Can be connected to on-chip timers)
JEXT[15:0]	Input	There are up to 16 external trigger inputs used for injected conversion (Can be connected to on-chip timers)
adc_ker_ck	Input	ADC core clock
adc_hclk	Input	ADC peripheral clock
adc_it	Output	ADC interrupt
adc_dma	Output	ADC DMA request
VREFINT	Input	Output voltage of internal reference voltage

Table 79 ADC Input/Output Pins

Pin name	Signal type	Description
VREF+	Positive analog reference voltage input	ADC high/positive reference voltage, 1.625V ≤ VREF+ ≤ VDDA
VDDA	Analog power input	The analog power equals to VDDA: 1.62V ≤ VDDA ≤ 3.6V
VREF−	Negative analog reference voltage input	ADC low/negative reference voltage
VSSA	Analog power ground input	Analog power ground pin

Pin name	Signal type	Description
VINP[2:0]	Positive input analog channel	Connected to external channels: ADC_INPi or internal channel
VINN[2:0]	Negative input analog channel	Connected to VREF− or external channel: ADC1_INNi
ADC1_INN[3:1]	External analog input signal	Up to 3 analog input channels
ADC1_INP[3:1]	External analog input signal	Up to 3 analog input channels

22.4.3 Buses and clocks

The ADC uses the AHB slave port for control/status register access and data access. The AHB interface supports word (32-bit) access.

The AHB slave interface does not support split/retry requests and will not generate AHB errors. The dual-clock domain architecture means that the ADC clock is independent of the AHB bus clock.

32-bit register read and write access is implemented via the AHB bus; ADC sampling, conversion process, and data processing use an independent ADC clock (6, 12, and 24 frequency division can be configured based on the system clock).

22.4.4 Single-ended channels and differential input channels.

The channel can be configured as single-ended input or differential input by writing to the DIFSEL bit in the ADCx_CFGR2 register.

In single-ended input mode, the conversion voltage is the difference between the current input channel voltage external VIN and 0. If VREFH = internal reference voltage (1.65V), the conversion range is 0 ~ 2*internal reference voltage, i.e., 0~3.3V; if VREFH = external reference voltage, the conversion range is 0 ~ external reference voltage.

In differential input mode, the conversion voltage is the difference between external VINP and VINN. If VREFH = internal reference voltage (1.65V), the conversion range is -2*internal reference voltage ~ +2*internal reference voltage, i.e., -3.3V ~ +3.3V; if VREFH = external reference voltage, the conversion range is -external reference voltage ~ +external reference voltage.

In differential mode, if VREFH = internal reference voltage (1.65V), the input signal common-mode voltage is 1.65V; if VREF+ = VDDA, the input signal common-mode is VDDA/2.

A 16-bit unsigned conversion result is output in single-ended mode, and a 16-bit signed conversion result is output in differential mode.

22.4.5 ADC switch control

Enable the ADC by setting the ADEN bit in the ADCx_CR register to 1.

Subsequently, regular conversion can be started by setting the ADSTART bit in the ADCx_CR register to 1, or by an external trigger event if the trigger is enabled.

Injected conversion can be started by setting the JADSTART bit in the ADCx_CR register to 1, or by an external injected trigger event if the injected trigger is enabled.

Procedure to disable the ADC by software:

(1) Check to confirm that both the ADSTART and JADSTART bits in the ADCx_CR register are 0 to ensure no conversion is being executed If needed, set the ADSTP bit to 1, and set the JADSTP bit to 1, then wait until both the ADSTP and JADSTP bits are cleared to 0 by hardware to stop any ongoing regular and injected conversions.

(2) If the application requires, clear the ADEN bit in the ADCx_CR register to 0 to disable the ADC.

22.4.6 Restrictions when writing to ADC control bits

Only when the ADC is disabled (the ADEN bit in the ADCx_CR register must be 0) can software configure and enable the ADC clock by setting the ADCADIV bit in the RCM_SCCR register and the ADCxEN bit in the RCM_AHBCG register (refer to the CMU section in the SYSCTRL chapter)

Software is allowed to write to the control bits ADSTART, JADSTART, ADSTP, and JADSTP in the ADCx_CR register only when the ADC is enabled.

For all other control bits in the ADCx_CFGR, ADCx_CFGR2, ADCx_SMPRx, ADCx_SQRy, ADCx_JDRy, ADCx_OFFSETy, ADCx_TRy, ADCx_AWD2CR, ADCx_AWD3CR and ADCx_IER registers:

For control bits related to regular conversion configuration, software is only allowed to write to these bits when the ADC is enabled (ADEN=1 in ADCx_CR) and no regular conversion is in progress (ADSTART shall equal 0).

For control bits related to injected conversion configuration, software is only allowed to write to these bits when the ADC is enabled (ADEN=1 in ADCx_CR) and no injected conversion is in progress (JADSTART shall equal 0).

If the ADC is enabled (ADEN=1 in ADCx_CR), software can write to the ADCx_JSQR register at any time.

22.4.7 Channel selection (SQRx, JSQRx)

Each ADC multiplexed channel has up to 6 channels, and conversions can be divided into two groups: regular conversion and injected conversion. Each group contains a conversion sequence that can be completed on any channel in any order.

A regular conversion group consists of a maximum of 12 conversions. The regular channels and their order in the conversion sequence shall be selected in the ADCx_SQRy registers. The total number of conversions in the regular conversion group shall be written to the RL[3:0] bits in the ADCx_SQR1 register.

A injected conversion group consists of a maximum of 4 conversions. The injected channels and their order in the conversion sequence shall be selected in the ADCx_JSQR registers. The total number of conversions in the injected conversion group shall be written to the JL[1:0] bits in the ADCx_JSQR register.

Modification of the ADCx_SQRy registers is not allowed when regular conversions might occur. Therefore, the ADSTP bit in the ADCx_CR register shall first be set to 1 to stop ADC regular conversion; during injected conversions, the ADCx_JSQR register can be modified in real time.

22.4.8 Sampling time of each channel can be set independently (SMPR1)

Before starting conversion, the ADC must establish a direct connection between the voltage source to be measured and the ADC's built-in sampling capacitor. The sampling time must be sufficient for the input voltage source to charge the embedded capacitor to the input voltage level.

Different sampling time can be used for sampling each channel. The sampling time can be programmed via the SMP[3:0] bits in the ADCx_SMPR1 register. The selectable sampling time values are from 1 to 16 ADC clock cycles.

The formula for total conversion time is as follows:

TCONV = Sampling time + 21 ADC clock cycles

The ADC indicates the end of the sampling phase by setting the EOSMP bit in the ADCx_ISR register to 1 by hardware (only for regular conversions).

22.4.9 Single conversion mode (CONT=0)

In single conversion mode, the ADC performs all conversions on the channel once. When the CONT bit in the ADCx_CFGR register is set to 0, this mode can be activated by:

- Set the ADSTART bit in the ADC_CR register to 1 (for regular channels)
- Set the JADSTART bit in the ADCx_CR register to 1 (for injected channels)
- External hardware trigger event (for regular or injected channels)

In a regular sequence, after each conversion is completed:

- The conversion data is stored in the ADCx_DR register
- The EOC (End of Regular Conversion) flag in the ADCx_ISR register is set to 1 by hardware

- An interrupt is generated upon completion of a conversion when the EOCIE bit in the ADCx_IER register is set to 1.

In an injected sequence, after each conversion is completed:

- Conversion data is stored in one of the four ADCx_JDRy registers.
- The JEOC (End of Injected Conversion) flag in the ADCx_ISR register is set to 1 by hardware.
- An interrupt is generated upon completion of a conversion when the JEOCIE bit in the ADCx_IER register is set to 1.

After the regular sequence is completed:

- The EOS (End of Regular Sequence) flag in the ADCx_ISR register is set to 1 by hardware
- An interrupt is generated upon completion of a conversion when the EOSIE bit in the ADCx_IER register is set to 1

After the injected sequence is completed:

- The JEOS (End of Injected Sequence) flag in the ADCx_ISR register is set to 1 by hardware
- An interrupt is generated upon completion of a conversion when the JEOSIE bit in the ADCx_IER register is set to 1

Subsequently, the ADC stops working until a new external regular or injected trigger occurs, or the ADSTART or JADSTART bit in the ADCx_CR register is set to 1 again.

Note: To convert a single channel, program the sequence length to 1.

22.4.10 Continuous conversion mode (CONT=1)

This mode is only applicable to regular channels.

In continuous conversion mode, if a software or hardware regular trigger event occurs, the ADC performs all regular conversions on the channel once, then automatically restarts and continues to perform each conversion in the sequence. When the CONT bit in the ADCx_CFGR register is 1, this mode can be enabled by an external trigger or by setting the ADSTART bit in the ADC_CR register to 1.

In a regular sequence, after each conversion is completed:

- The conversion data is stored in the ADCx_DR register
- The EOC (End of Conversion) flag in the ADCx_ISR register is set to 1 by hardware
- An interrupt is generated upon completion of a conversion when the EOCIE bit in the ADCx_IER register is set to 1.

After the conversion sequence is completed:

- The EOS (End of Sequence) flag in the ADCx_ISR register is set to 1 by hardware

- An interrupt is generated upon completion of a conversion when the EOSIE bit in the ADCx_IER register is set to 1. Immediately afterward, a new sequence restarts, and the ADC continues to repeat the conversion sequence.

Note: To convert a single channel, program the sequence length to 1.

Discontinuous mode and continuous mode cannot be enabled simultaneously; i.e., it is forbidden to set both the DISCEN bit and the CONT bit in the ADCx_CFGR register to 1 simultaneously.

Injected channels cannot be converted continuously. The only exception is that in continuous conversion mode (the JAUTO bit in the ADCx_CFGR register is set to 1), the injected channels can be configured for automatic conversion after regular channels.

22.4.11 Start of conversion (ADSTART, JADSTART)

Software starts ADC regular conversion by setting the ADSTART bit to 1 in the ADCx_CR register.

After ADSTART is set to 1, conversion starts:

- Conversion starts immediately: When the EXTEN[1:0] bits in the ADCx_CFGR register are 00 (software trigger).
- Conversion starts on the next valid edge of the selected regular hardware trigger: When the EXTEN bit in the ADCx_CFGR register is not 0, software starts ADC injected conversion by setting the JADSTART bit in the ADCx_CR register to 1.

After JADSTART is set to 1, conversion starts:

- Conversion starts immediately: When the JEXTEN bit in the ADCx_CFGR register is 0 (software trigger)
- Conversion starts on the next valid edge of the selected regular hardware trigger: When the JEXTEN[1:0] bits in the ADCx_CFGR register is not 00

Note: In automatic injection mode (the JAUTO bit in the ADCx_CFGR register is set to 1), use the ADSTART bit to start regular conversion, followed by automatic injected conversion (the JADSTART bit in the ADCx_CR register shall remain cleared to 0).

The ADSTART and JADSTART bits in the ADCx_CR register also provide information on whether ADC operation is being performed. The ADC can be reconfigured when ADSTART=0 and JADSTART=0 (indicating the ADC is idle).

ADSTART is cleared to 0 by hardware:

In single mode with software regular trigger (the CONT bit in ADCx_CFGR is set to 0, and the EXTEN bit is set to 0)

- If the DISCEN bit in ADCx_CFGR is set to 1, as long as the regular conversion sequence ends (EOS is set to 1) or subgroup

processing ends, it clears to 0 in all cases (the CONT bit in the ADCx_CFGR register is a non-0 value, and the EXTEN[1:0] bits are a non-00 value)

- Clear to 0 after executing the software setting of the ADSTP bit in the ADCx_CR register to 1.

Note: In continuous mode (the CONT bit in the ADCx_CFGR register is set to 1), since the sequence automatically restarts, the ADSTART bit in the ADCx_CR register is not cleared to 0 by hardware when the EOS bit in the ADCx_ISR register is set to 1 by hardware.

If hardware trigger is selected in single mode (the CONT bit in ADCx_CFGR register is set to 0, and the EXTEN bit is not 0), the ADSTART bit in the ADCx_CR register is not cleared to 0 by hardware when the EOS bit in the ADCx_ISR register is set to 1 by hardware, so the software does not need to reset ADSTART again for the next hardware trigger event. This ensures that no subsequent hardware triggers are missed.

JADSTART is cleared to 0 by hardware:

In single mode with software injected trigger (the JEXTEN[1:0] bits in ADCx_CFGR is set to 00).

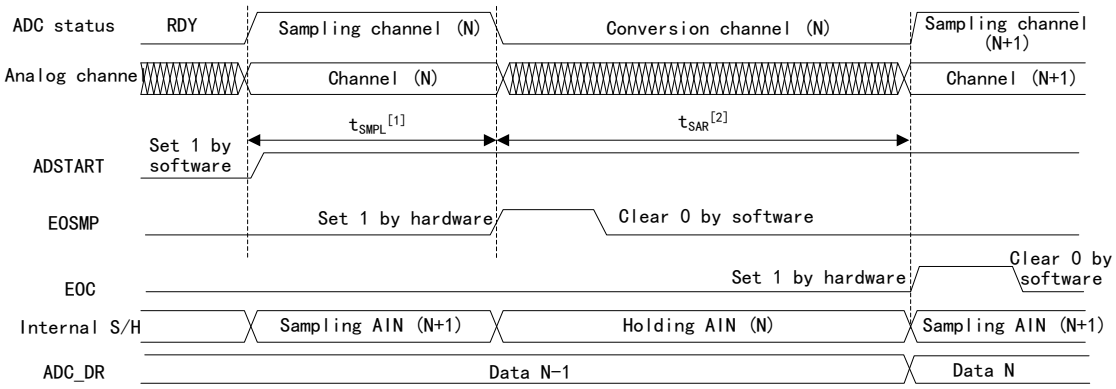
- If the DISCEN bit in ADCx_CFGR is set to 1, as long as the injected conversion sequence ends (JEOS is set to 1) or subgroup processing ends, it clears to 0 in all cases (the JEXTEN[1:0] bits in the ADCx_CFGR register is not 00)
- Clear to 0 after executing the software setting of the JADSTP bit in the ADCx_CR register to 1

Note: When software trigger is selected, the ADSTART bit shall not be set to 1 if the EOC flag of the ADCx_ISR register is still high.

22.4.12 ADC timing

The time elapsed from the start to the end of conversion is the sum of the configured sampling time and the successive approximation time.

Figure 86 Analog-to-Digital Conversion Time



Note:

(1) t_{SMPL} depends on SMPx[3:0].

(2) t_{SAR} is the conversion time: 21 ADC_CLK cycles .

22.4.13 Stop the ongoing conversion (ADSTP, JADSTP)

Software decides whether to stop conversion. To stop an ongoing regular conversion, set the ADSTP bit in the ADCx_CR register to 1; to stop an ongoing injected conversion, set the JADSTP bit to 1.

Stopping the conversion will reset the ongoing ADC operation. Subsequently, the ADC can be reconfigured (e.g., changing the channel selection or trigger) to prepare for new operation.

Note: Injected conversion can be stopped while a regular conversion is still executing, and vice versa. This allows reconfiguring the injected conversion sequence and trigger while a regular conversion is still in progress, and vice versa.

If the ADSTP bit in the ADCx_CR register is set to 1 by software, any ongoing regular conversion will be aborted, and partial conversion results will be discarded (the ADCx_DR register will not be updated with the current conversion result).

If the JADSTP bit in the ADCx_CR register is set to 1 by software, any ongoing injected conversion will be aborted, and partial conversion results will be discarded (the ADCx_JDRy register will not be updated with the current conversion result). The scan sequence will also be aborted and reset (this means restarting the ADC will begin a new sequence).

After the program completes execution, the ADSTP/ADSTART bits (for regular conversion) or JADSTP/JADSTART bits (for injected conversion) in the ADCx_CR register will be cleared to 0 by hardware. Software shall poll the ADSTART (or JADSTART) until it is reset, and then it can determine that the ADC has fully stopped.

Note: In automatic injection mode (the JAUTO bit in the ADCx_CFGR register is set to 1), setting the ADSTP bit in the ADCx_CR register to 1 will abort both regular and injected conversions (JADSTP shall not be used).

Figure 87 Stop Ongoing Regular Conversion

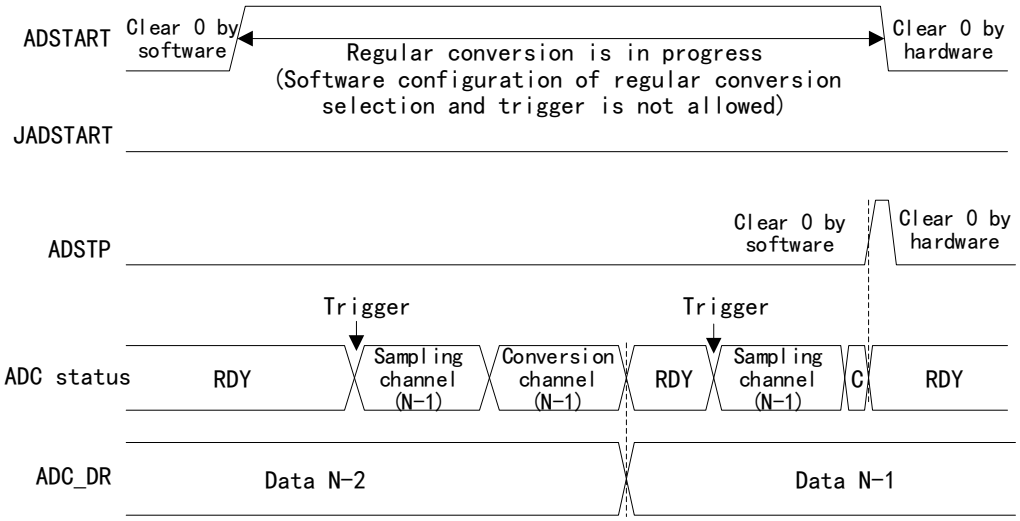
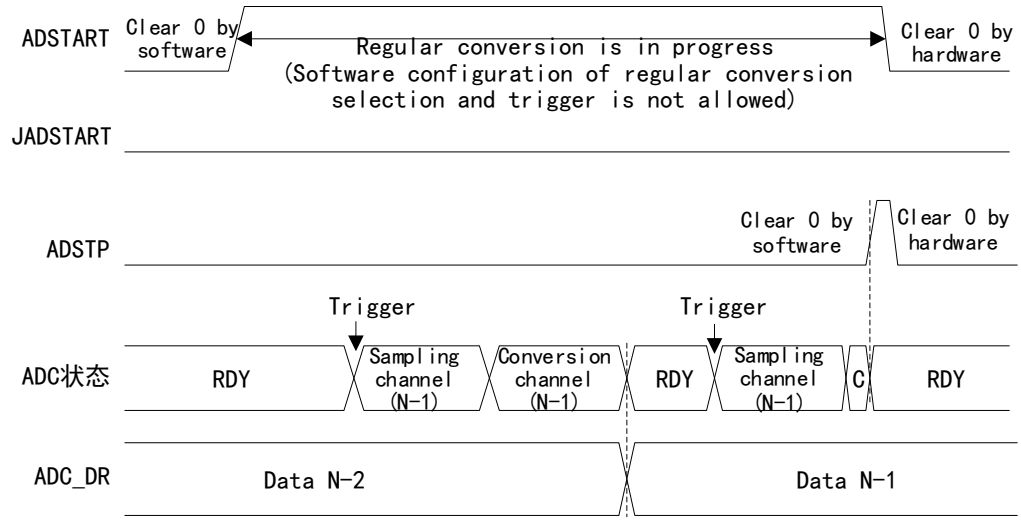


Figure 88 Stop Ongoing Regular Conversion and Injected Conversion



22.4.14 External trigger conversion and trigger polarity (EXTSEL, EXTEN, JEXTSEL, JEXTEN)

Conversions or conversion sequences can be triggered by software or external events (e.g., timer capture, input pin). If neither the EXTEN[1:0] control bits in the ADCx_CFGR register (for regular conversions) nor the JEXTEN[1:0] bits in the ADCx_JSQR register (for injected conversions) are "00", external events can trigger conversions with the selected polarity.

Injected software trigger cannot be implemented when the injected queue is enabled (the JQDIS bit in ADCx_CFGR register is set to 1).

Regular trigger selection becomes effective after software sets the ADSTART bit in the ADCx_CR register to 1; injected trigger selection becomes effective after software sets the JADSTART bit in the ADCx_CR register to 1.

Hardware triggers occurring during conversion will be ignored.

- If the ADSTART bit is set to 0, any regular hardware triggers that occur will be ignored.
- If the JADSTART bit is set to 0, any injected hardware triggers that occur will be ignored.

The table below provides the correspondence between the EXTEN[1:0] values in the ADCx_CFGR register and the JEXTEN[1:0] values in the ADCx_JSQR register and the trigger polarity.

Table 80 Configure Trigger Polarity for Regular External Triggers

EXTEN[1:0]	Source
00	Disable hardware trigger detection, enable software trigger detection
01	Hardware trigger detection on rising edge
10	Hardware trigger detection on falling edge
11	Perform hardware trigger detection on both rising and falling edges

Note: The regular trigger polarity cannot be changed in real time.

Table 81 Configure Trigger Polarity for Injected External Triggers

JEXTEN[1:0]	Source
00	If JQDIS=1 (queue disabled): Hardware trigger detection is disabled, and software trigger detection is enabled If JQDIS=0 (queue enabled): Both hardware and software trigger detections are disabled
01	Hardware trigger detection on rising edge
10	Hardware trigger detection on falling edge
11	Perform hardware trigger detection on both rising and falling edges

Note: If the queue is enabled (the JQDIS bit in ADCx_CFGR register is set to 0), the polarity of the injected trigger can be expected and changed in real time.

The EXTSEL[3:0] control bits in the ADCx_CFGR register and the JEXTSEL[3:0] control bits in the ADCx_JSQR register are used to select from 16 possible events the events that can trigger regular group conversions and injected group conversions. Injected trigger can interrupt regular group conversions.

Note: The regular trigger selection cannot be changed in real-time. The injected trigger selection can be expected and changed in real time.

The following table lists all possible external triggers of the ADC for regular and injected conversions.

Table 82 External Trigger of Regular Channel

Source	Type	EXTSEL[3:0]
TMR1_CC1	Internal signal from on-chip timer	0000
TMR1_CC2	Internal signal from on-chip timer	0001

Source	Type	EXTSEL[3:0]
TMR1_CC3	Internal signal from on-chip timer	0010
TMR2_CC2	Internal signal from on-chip timer	0011
TMR3_TRGO	Internal signal from on-chip timer	0100
TMR4_CC4	Internal signal from on-chip timer	0101
EINT line 11	External pin	0110
TMR1_TRGO	Internal signal from on-chip timer	0111
TMR2_TRGO	Internal signal from on-chip timer	1000
TMR4_TRGO	Internal signal from on-chip timer	1001
TMR3_CC4	Internal signal from on-chip timer	1010

Table 83 External Trigger of Injected Channel

Source	Type	JEXTSEL[3:0]
TMR1_TRGO	Internal signal from on-chip timer	0000
TMR1_CC4	Internal signal from on-chip timer	0001
TMR2_TRGO	Internal signal from on-chip timer	0010
TMR2_CC1	Internal signal from on-chip timer	0011
TMR3_CC4	Internal signal from on-chip timer	0100
TMR4_TRGO	Internal signal from on-chip timer	0101
EINT line 15	External pin	0110
TMR3_CC3	Internal signal from on-chip timer	0111
TMR3_TRGO	Internal signal from on-chip timer	1000
TMR3_CC1	Internal signal from on-chip timer	1001

22.4.15 Injected channel management

Trigger injection mode

To use trigger injection, the JAUTO bit in the ADCx_CFGR register shall be set to 0.

- Regular channel group conversion is started by an external trigger or by setting the ADSTART bit in the ADCx_CR register to 1.
- If an external injected trigger occurs during the regular channel group conversion, or if the JADSTART bit in the ADCx_CR register is set to 1, the current conversion will be reset, and the injected channel sequence switching will be started (all injected channels are converted once).
- The regular conversion of the regular channel group then resumes from the last interrupted regular conversion.
- If a regular event occurs during an injected conversion period, the injected conversion will not be interrupted, but the regular sequence will be executed after the injected sequence ends.

Note: When using triggered injection, ensure that the interval between trigger events is longer than the injected sequence.

Automatic injection mode

If the JAUTO bit in the ADCx_CFGR register is set to 1, the channels in the injected group are automatically converted after the regular group channels. This can be used to convert a sequence of up to 16 conversions, which are programmed in the ADCx_SQRy and ADCx_JSQR registers.

In this mode, the ADSTART bit in the ADCx_CR register shall be set to 1 to start the regular conversion, followed by the injected conversion (JADSTART shall remain 0). Setting the ADSTP bit to 1 will abort both regular and injected conversions (the JADSTP bit shall not be used). In this mode, external triggers on injected channels shall be disabled.

If both the CONT bit and the JAUTO bit in the ADCx_CFGR register are set to 1, the regular channels and subsequent injected channels will be continuously converted.

Note: Automatic injection and discontinuous sampling modes cannot be used simultaneously.

When using DMA to export data from the regular sequencer in automatic injection group mode, it shall be set to circular mode (setting the CIRCMEN bit in the DMA_SCFG register to 1). If the CIRCMEN bit is set to 0 (circular mode is disabled), the automatic injection group conversion sequence will stop upon a DMA transmission completion event.

22.4.16 Discontinuous modes (DISCEN, DISCNUM, JDISCEN)

Regular group mode

This mode can be enabled by setting the DISCEN bit in the ADCx_CFGR register to 1.

This mode is used to convert short sequences (subgroups) containing n (n ≤ 6) conversions, which are part of the conversion sequence selected in the ADCx_SQRy register. The value of n can be specified by writing to the DISCNUM[2:0] bits in the ADCx_CFGR register.

When an external trigger occurs, the next n conversions selected in the ADCx_SQRy register will start until all conversions in the sequence are completed. The total sequence length is defined by the RL[3:0] bits in the ADCx_SQR1 register.

Examples:

DISCEN=1, n=3, channels to be converted = 1, 2, 3, 6, 7, 8, 9, 10, 11

- First trigger: The converted channels are 1, 2, 3 (an EOC event is generated in each conversion).
- Second trigger: The converted channels are 6, 7, 8 (an EOC event is generated in each conversion).

- Third trigger: The converted channels are 9, 10, 11 (an EOC event is generated in each conversion), and an EOS event will be generated upon conversion completion of Channel 11.
- Fourth trigger: The converted channels are 1, 2, 3 (an EOC event is generated in each conversion).
- ...

DISCEN=0, channels to be converted = 1, 2, 3, 6, 7, 8, 9, 10, 11

- First trigger: The entire sequence is converted: Channel 1, then Channels 2, 3, 6, 7, 8, 9, 10, and 11. Each conversion generates an EOC event, and the last conversion also generates an EOS event.
- All subsequent trigger events will restart the entire sequence.

Note: When converting a regular group in discontinuous mode, there is no reverse (the number of conversions in the last subgroup of the sequence is less than n).

After converting all subgroups, the next trigger signal will start the conversion of the first subgroup. In the above example, the fourth trigger re-converts Channels 1, 2, and 3 from the first subgroup.

Discontinuous mode and continuous mode cannot be enabled simultaneously. If both modes are enabled simultaneously (i.e., DISCEN=1, CONT=1), the ADC assumes continuous mode is disabled and continues related operation.

Injection group mode

This mode can be enabled by setting the JDISCEN bit in the ADCx_CFGR register to 1. After an external injection trigger event occurs, this mode will convert the sequence selected in the ADCx_JSQR register channel by channel, equivalent to the case where the regular channel "n" in discontinuous mode is fixed to 1.

When an external trigger occurs, the next channel conversions selected in the ADCx_JSQR register will start until all conversions in the sequence are completed. The total sequence length is defined by the JL[1:0] bits in the ADCx_JSQR register.

Examples:

JDISCEN=1, channels to be converted = 1, 2, 3

- First trigger: Convert Channel 1 (a JEOC event is generated)
- Second trigger: Convert Channel 2 (a JEOC event is generated)
- Third trigger: Convert Channel 3, and generate a JEOC event and JEOS event
- ...

Note: After all injected channels are converted, the next trigger signal will start the conversion of the first injected channel. In the above example, the fourth trigger re-converts the first injected channel 1.

Automatic injection mode and discontinuous mode cannot be used simultaneously: When JAUTO is set to 1 in the ADCx_CFGR register, the DISCEN and JDISCEN bits must be kept cleared to 0 by software.

Context queue for injected conversion

Implementing a context queue can prepare up to 2 contexts for the next injected conversion sequence. The JQDIS bit in the ADCx_CFGR register shall be set to 0 to enable this function. When the context queue is enabled, only hardware-triggered conversions can be performed.

This context includes:

- Configuration of injected trigger (JEXTEN[1:0] and JEXTSEL bits in the ADCx_JSQR register)
- Definition of injected sequence (JSQx[4:0] and JL[1:0] bits in the ADCx_JSQR register)

All parameters of the context are defined in the ADCx_JSQR register, which implements a double-buffer queue capable of buffering up to 2 sets of parameters:

- The ADCx_JSQR register can be written at any time, even during an injected conversion.
- Each data written to the ADCx_JSQR register will be stored in the context queue.
- The queue is empty at the beginning. The first write access to the ADCx_JSQR register immediately changes the context, and the ADC is then ready to receive the injected trigger.
- After the injected sequence is completed, the queue will be occupied, and the context will be changed according to the subsequent JSQR parameters stored in the queue. This new context is used for the next injected conversion sequence.
- If a write operation is performed to the ADCx_JSQR register when the queue is full, a queue overflow will occur. This overflow will be indicated by setting the JQOVF flag bit in the ADCx_ISR register to 1 by hardware. In case of overflow, the ADCx_JSQR register write access that causes the overflow will be ignored, and the context queue remains unchanged. If the JQOVFIE bit in the ADCx_IER register is set to 1, an interrupt can be generated.
- Two operations might be performed when the queue becomes empty, depending on the value of the control bit JQM in the ADCx_CFGR register.

  - If JQM=0 in the ADCx_CFGR register, the queue is empty right after enabling the ADC but will never become empty during subsequent operation: the queue always retains the last valid

context, and subsequent valid injection sequence will be processed based on the last valid context.

- If JQM=1 in the ADCx_CFGR register, the queue will become empty after an injected sequence ends or after the queue is emptied. In this case, there is no context in the queue, and hardware triggers will be disabled. Therefore, all subsequent hardware injected triggers will be ignored until software writes a new injection context to the JSQR register.

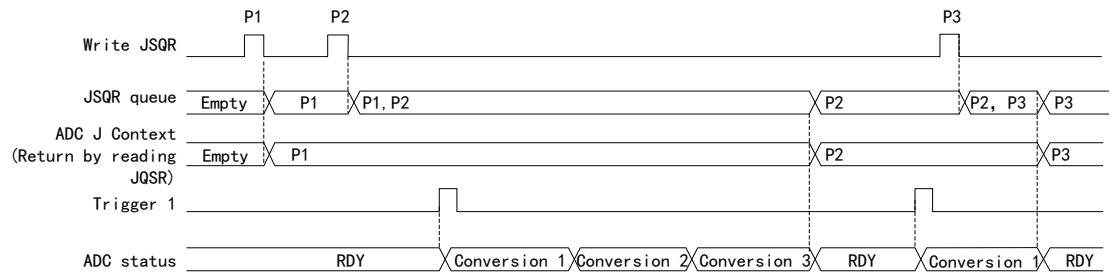
- Reading the ADCx_JSQR register will return the currently valid JSQR context. If the JSQR context is empty, the read value of JSQR is 0x0000.
- If an injected conversion is stopped by setting the JADSTP bit in the ADCx_CR register to 1, the queue will be emptied.
  - If JQM=0 in the ADCx_CFGR register, the queue will retain the last valid context.
  - If JQM=1 in the ADCx_CFGR register, the queue will become empty, and triggers will be ignored.

Note: If configured in discontinuous mode (the bit JDISCEN=1 in the ADCx_CFGR register), only the last trigger of the injected sequence can change the context and occupy the queue. The first trigger only occupies the queue, but other triggers remain valid, as shown in the discontinuous mode example of the contexts below (each of the two contexts has a length of 3):

- First trigger, discontinuous. Sequence 1: Context 1 occupied, first conversion executed.
- Second trigger, discontinuous. Sequence 1: Second conversion.
- Third trigger, discontinuous. Sequence 1: Third conversion.
- Fourth trigger, discontinuous. Sequence 2: Context 2 occupied, first conversion executed.
- Fifth trigger, discontinuous. Sequence 2: Second conversion.
- Sixth trigger, discontinuous. Sequence 2: Third conversion

Operations when changing trigger or sequence context

Figure 89 JSQR Context Queue Example (Queue Change)

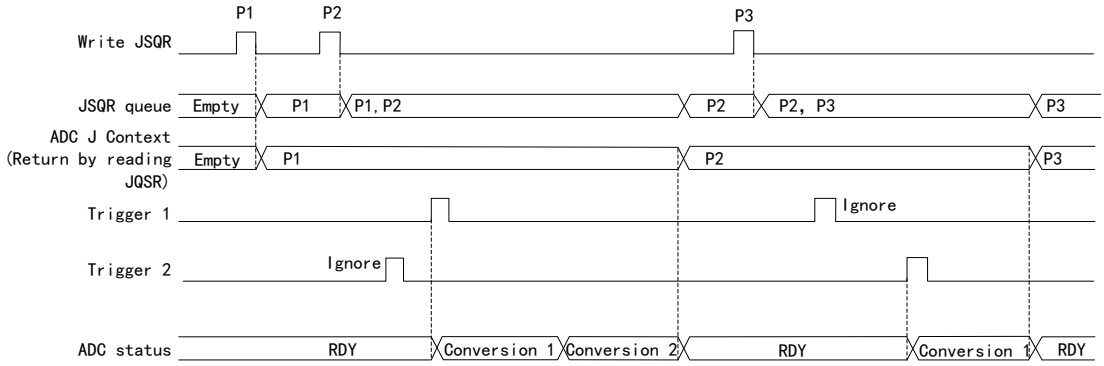


Parameters:

- P1: Sequence consisting of 3 conversion groups, hardware trigger 1

- P2: Sequence consisting of 1 conversion group, hardware trigger 1
- P3: Sequence consisting of 4 conversion groups, hardware trigger 1

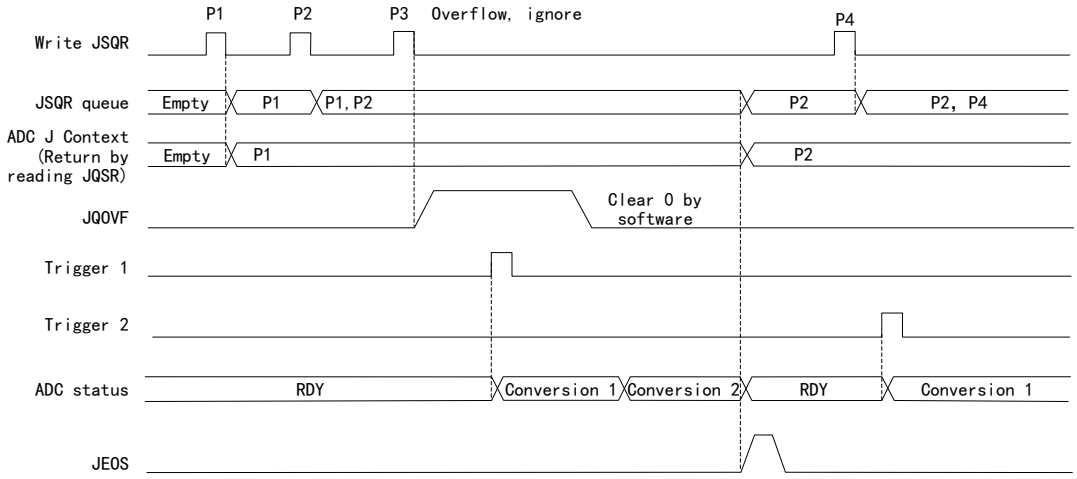
Figure 90 JSQR Context Queue Example (Trigger Change)



Parameters:

- P1: Sequence consisting of 2 conversion groups, hardware trigger 1
- P2: Sequence consisting of 1 conversion group, hardware trigger 2
- P3: Sequence consisting of 4 conversion groups, hardware trigger 1

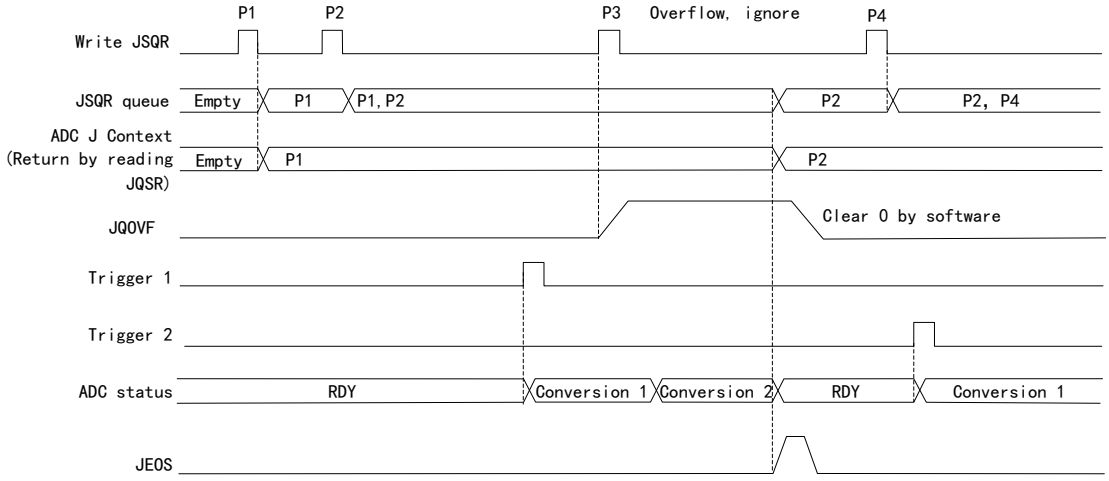
Figure 91 Example of JSQR Context Queue with Overflow Before Conversion



Parameters:

- P1: Sequence consisting of 2 conversion groups, hardware trigger 1
- P2: Sequence consisting of 1 conversion group, hardware trigger 2
- P3: Sequence consisting of 3 conversion groups, hardware trigger 1
- P4: Sequence consisting of 4 conversion groups, hardware trigger 1

Figure 92 Example of JSQR Context Queue with Overflow During Conversion



Parameters:

- P1: Sequence consisting of 2 conversion groups, hardware trigger 1
- P2: Sequence consisting of 1 conversion group, hardware trigger 2
- P3: Sequence consisting of 3 conversion groups, hardware trigger 1
- P4: Sequence consisting of 4 conversion groups, hardware trigger 1

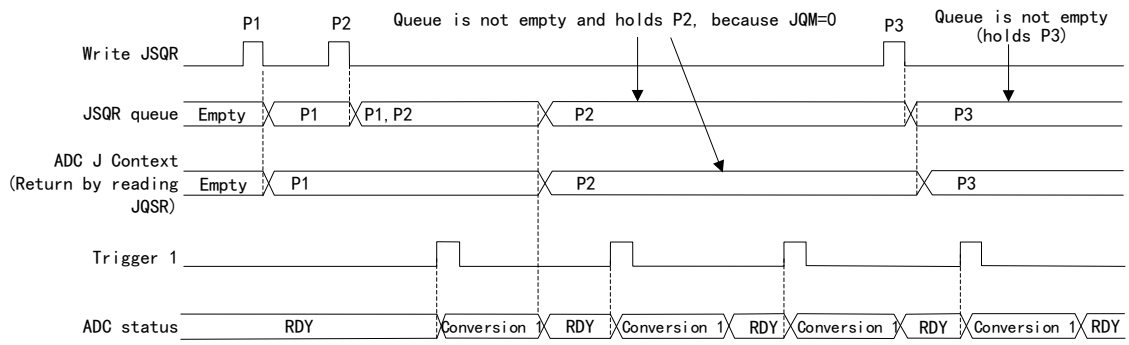
It is recommended to manage queue overflow according o the method below:

After writing each P context to the ADCx_JSQR register, the JQOVF bit in the ADCx_ISR register indicates whether the write operation is ignored (an interrupt can be generated).

To avoid queue overflow, write the third context (P3) only after the JEOS flag in the ADCx_ISR register for the previous context P2 is set to 1 by hardware. This ensures that the previous context has been occupied, and the queue is not full.

Context queue: Operations when the queue becomes empty

Figure 93 JSQR Context Queue Example When the Queue is Empty (JQM=0 in ADCx_CFGR Register)



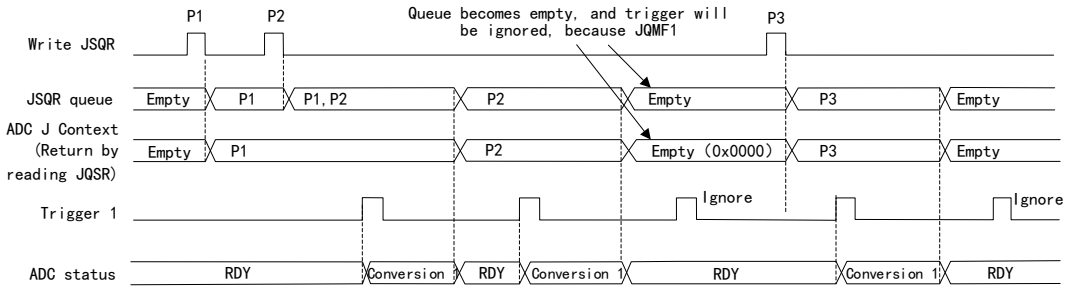
Parameters:

- P1: Sequence consisting of 1 conversion group, hardware trigger 1

- P2: Sequence consisting of 1 conversion group, hardware trigger 1
- P3: Sequence consisting of 1 conversion group, hardware trigger 1

Note: When P3 is written, the context will change immediately. However, due to internal resynchronization, there is a certain delay. If a trigger occurs just before or after P3 is written, the started conversion might be considered under context P2. To avoid this situation, the user shall ensure that no ADC trigger occurs when writing a new context that will be applied immediately.

Figure 94 JSQR Context Queue Example When the Queue is Empty (JQM=1 in ADCx_CFGR Register)



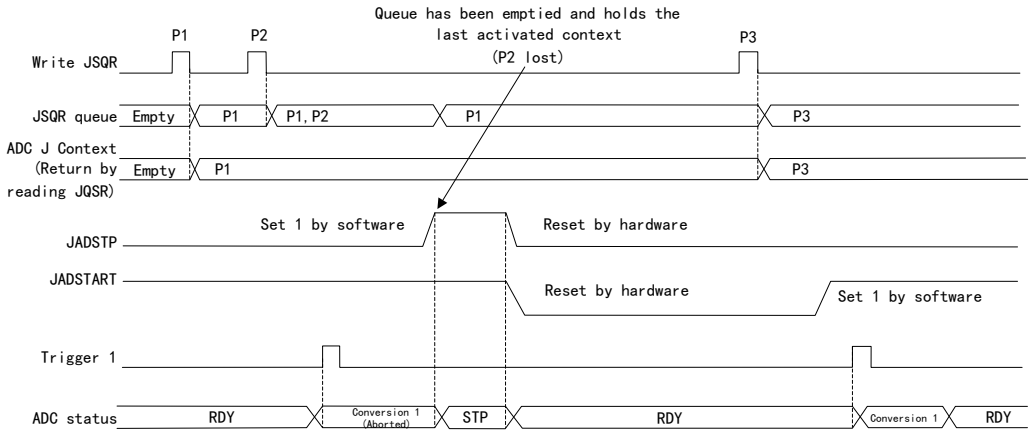
Parameters:

- P1: Sequence consisting of 1 conversion group, hardware trigger 1
- P2: Sequence consisting of 1 conversion group, hardware trigger 1
- P3: Sequence consisting of 1 conversion group, hardware trigger 1

Empty the context queue

The following figure shows the operation of the context queue when the queue is emptied in various situations.

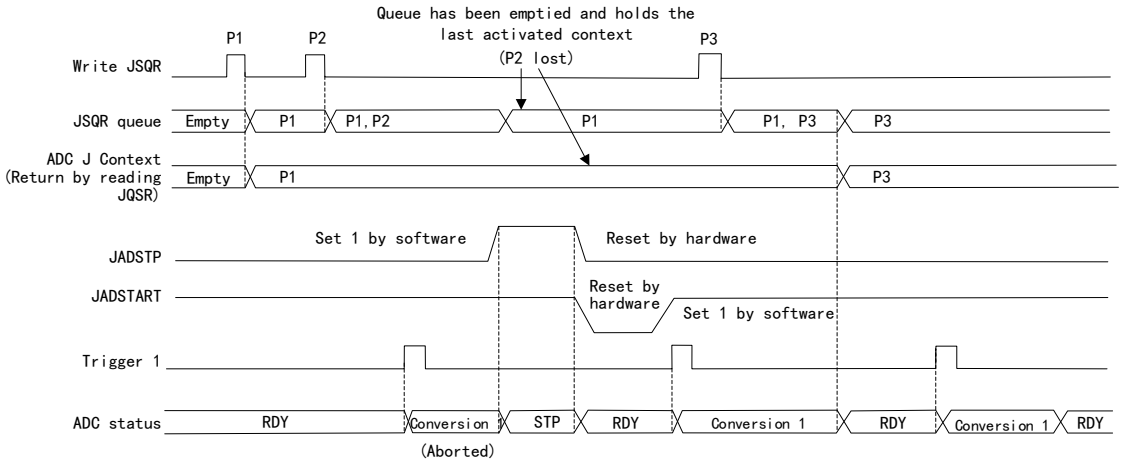
Figure 95 Emptying of JSQR Context Queue by Setting JADSTP Bit in ADCx_CR Register to 1 (JQM=0 in ADCx_CFGR Register)



Parameters:

- P1: Sequence consisting of 1 conversion group, hardware trigger 1
- P2: Sequence consisting of 1 conversion group, hardware trigger 1
- P3: Sequence consisting of 1 conversion group, hardware trigger 1

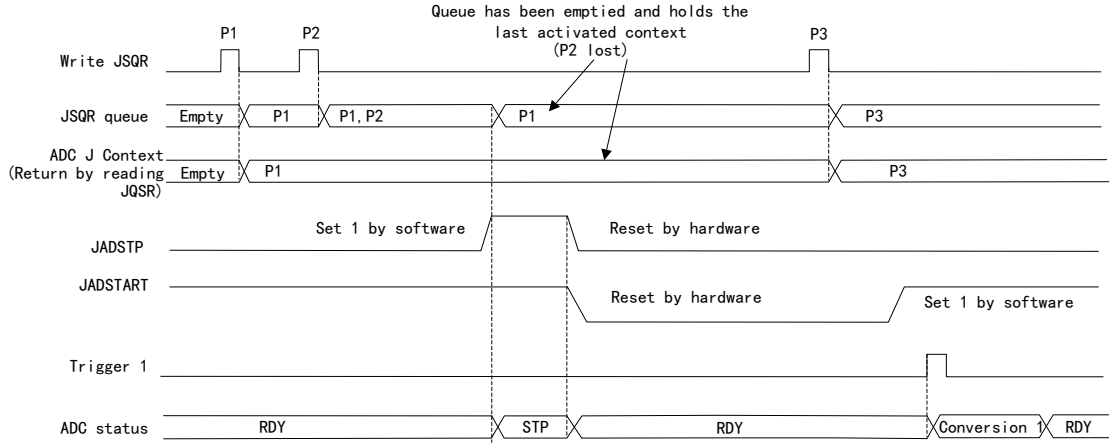
Figure 96 Emptying of JSQR Context Queue by Setting JADSTP Bit in ADCx_CR Register to 1 (JQM=0 in ADCx_CFGR Register)



Parameters:

- P1: Sequence consisting of 1 conversion group, hardware trigger 1
- P2: Sequence consisting of 1 conversion group, hardware trigger 1
- P3: Sequence consisting of 1 conversion group, hardware trigger 1

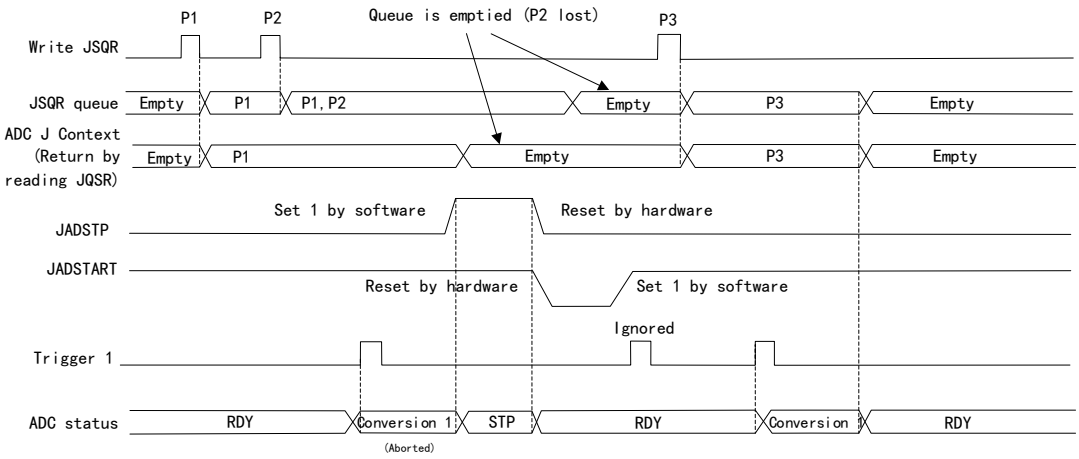
Figure 97 Emptying of JSQR Context Queue by Setting JADSTP Bit in ADCx_CR Register to 1 (JQM=0 in ADCx_CFGR Register)



Parameters:

- P1: Sequence consisting of 1 conversion group, hardware trigger 1
- P2: Sequence consisting of 1 conversion group, hardware trigger 1
- P3: Sequence consisting of 1 conversion group, hardware trigger 1

Figure 98 Emptying of JSQR Context Queue by Setting JADSTP Bit in ADCx_CR Register to 1 (JQM=1 in ADCx_CFGR Register)



Parameters:

- P1: Sequence consisting of 1 conversion group, hardware trigger 1
- P2: Sequence consisting of 1 conversion group, hardware trigger 1
- P3: Sequence consisting of 1 conversion group, hardware trigger 1

Context queue: Start ADC when queue is empty

To start ADC operation when the queue is empty, the following procedure shall be followed to avoid failure to obtain the first context during ADC initialization. This procedure is only applicable when the JQM bit in the ADCx_CFGR register is 0:

(1) Write an empty JSQR, ensuring that the JEXTEN[1:0] bits in the ADCx_JSQR register do not equal 0 (otherwise, the software conversion will be triggered).

(2) Set the JADSTART bit in the ADCx_CR register to 1

(3) Set the JADSTP bit in the ADCx_CR register to 1

(4) Wait until the JADSTART bit in the ADCx_CR register is reset

(5) Set the JADSTART bit in the ADCx_CR register to 1

The queue can be disabled by setting the JQDIS bit in the ADCx_CFGR register to 1.

22.4.17 End of conversion, end of sampling phase (EOC, JEOC, EOSMP)

The ADC will notify the application each time an event of end of regular conversion (EOC) or end of injected conversion (JEOC) occurs.

When the new regular conversion data appears in the ADCx_DR register, the ADC will immediately set the EOC flag bit in the ADCx_ISR register to 1 by

hardware. If the EOCIE bit in the ADCx_IER register is set to 1, an interrupt can be generated. The EOC flag can be cleared to 0 by writing 1 to it via software or reading the ADC_DR register.

When the new injected conversion data appears in the ADCx_JDRy register, the ADC will immediately set the JEOC flag bit in the ADCx_ISR register to 1 by hardware. If the JEOCIE bit in the ADCx_IER register is set to 1, an interrupt can be generated. The JEOC flag can be cleared to 0 by writing 1 to it via software or reading the corresponding ADCx_JDRy register.

The ADC indicates the end of the sampling phase by setting the EOSMP status bit in the ADCx_ISR register to 1 by hardware (only for regular conversions). The EOSMP flag can be cleared to 0 by writing 1 to it via software. If the EOSMPIE bit in the ADCx_IER register is set to 1, an interrupt can be generated.

22.4.18 End of conversion sequence (EOS, JEOS)

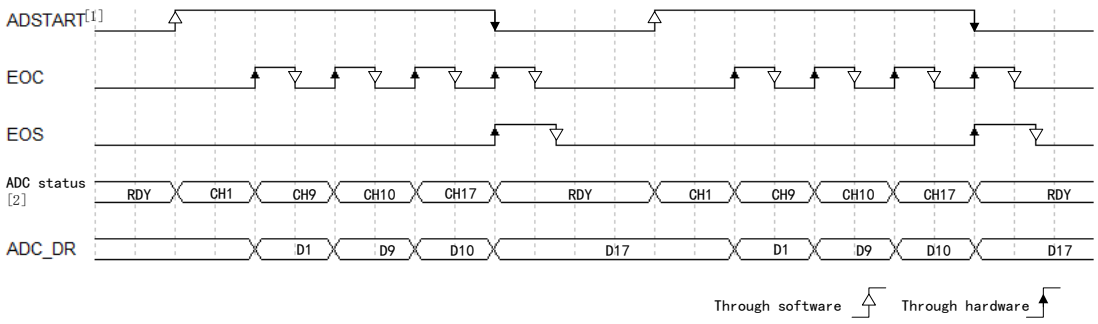
The ADC will notify the application each time an event of end of regular sequence (EOS) or end of injected sequence (JEOS) occurs.

When the last data of the regular conversion sequence appears in the ADCx_DR register, the ADC will immediately set the EOS flag bit in the ADCx_ISR register to 1 by hardware. If the EOSIE bit in the ADCx_IER register is set to 1, an interrupt can be generated. The EOS flag can be cleared by software writing 1 to it.

When the last data of the injected conversion sequence appears is completed, the ADC will immediately set the JEOS flag bit in the ADCx_ISR register to 1 by hardware. If the JEOSIE bit in the ADCx_IER register is set to 1, an interrupt can be generated. The JEOS flag can be cleared to 0 by writing 1 to it via software.

22.4.19 Example timing diagrams (single mode/continuous mode, hardware/software trigger)

Figure 99 Single sequence conversion, software trigger

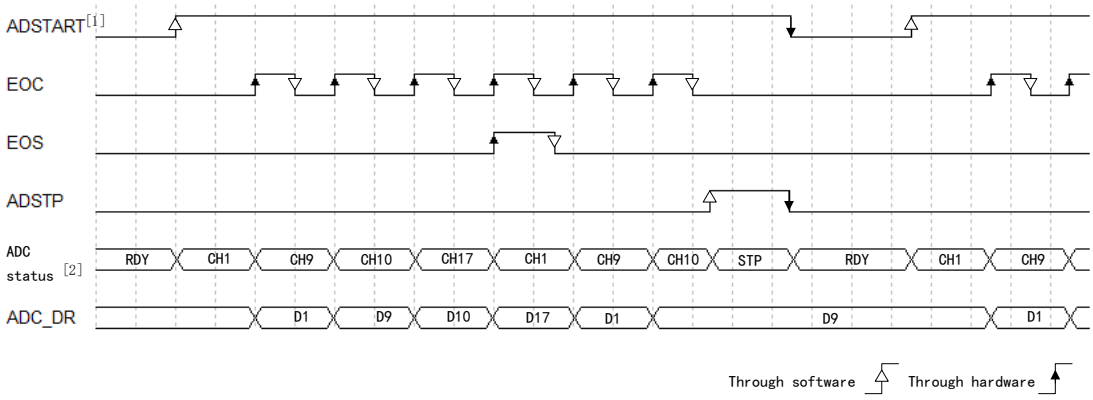


Note:

(1) In the ADCx_CFGR register, EXTEN[1:0]=00, CONT=0

(2) Selected channels = 1, 9, 10, 17; AUTDLY=0

Figure 100 Continuous sequence conversion, software trigger

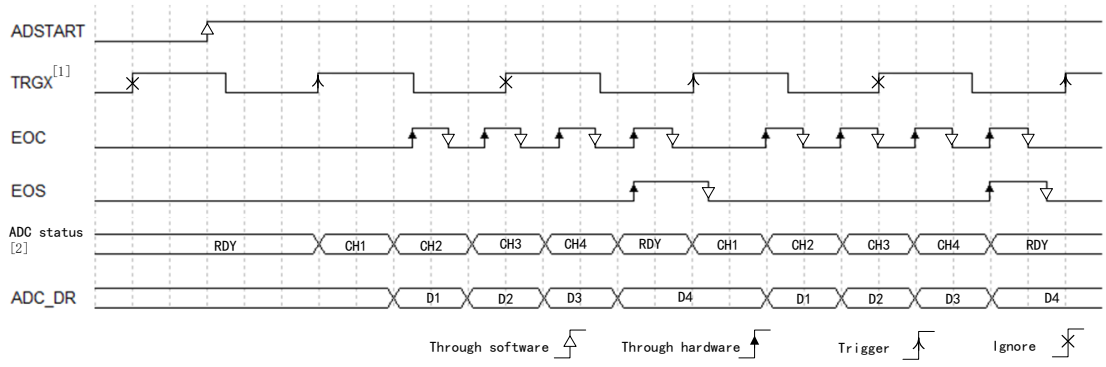


Note:

(1) In the ADCx_CFGR register, EXTEN[1:0]=00, CONT=0

(2) Selected channels = 1, 9, 10, 17; AUTDLY=0 in the ADCx_CFGR register

Figure 101 Single sequence conversion, hardware trigger

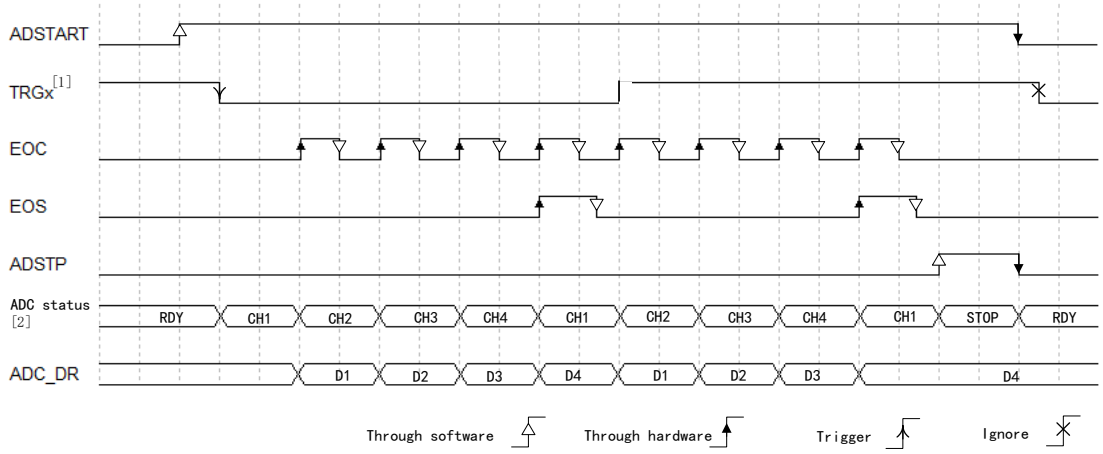


Note:

(1) Select TRGx (over-frequency) as trigger source; in the ADCx_CFGR register, EXTEN[1:0]=01, CONT=0.

(2) Selected channels = 1, 2, 3, 4; AUTDLY=0 in the ADCx_CFGR register.

Figure 102 Continuous sequence conversion, hardware trigger



Note:

(1) Select TRGx (over-frequency) as trigger source; in the ADCx_CFGR register, EXTEN[1:0]=01, CONT=1.

(2) Selected channels = 1, 2, 3, 4; AUTDLY=0 in the ADCx_CFGR register.

22.4.20 Data management

DMA transfer data size only supports words (32 bits).

Data registers, data offset (ADCx_DR, OFFSETy)

At the end of each regular conversion for a channel (when an EOC event occurs), the results of the data after conversion will be stored in the 16-bit-wide ADCx_DR data register.

At the end of each injected conversion for a channel (when an JEOC event occurs), the results of the data after conversion will be stored in the 16-bit-wide ADCx_JDRy data register.

An offset y (y=0~5) can be applied to a channel by setting the OFFSETy_EN bit in the ADCx_OFFSETy register to 1. The channel to which the offset is applied will be programmed in the OFFSETy[15:0] bits of the ADCx_OFFSETy register. In this case, the converted value will be subtracted by the user-defined offset written into the OFFSETy[15:0] bits.

- A 16-bit unsigned conversion result needs to be output in ADC single-ended mode, and a 16-bit signed conversion result needs to be output in differential mode.
- In synchronous mode, the 16-bit results of two ADC are placed into the same 32-bit register, namely in the ADC1_DR data register of the master ADC (ADC1).

Based on the 16-bit unsigned conversion result output in ADC single-ended

mode and the 16-bit signed conversion result in differential mode, the functions for the 16-bit ADC part are designed as follows:

The offset value register is set to 16 bits, without sign bits.

The conversion results of single-ended mode do not need adding sign bits after offset correction. If overflow/underflow occurs after correction, it can saturate at 0xFFFF/0x0000. In differential mode, overflow/underflow saturates at 0x7FFF and 0x8000.

ADC overflow (OVR, OVRMOD)

If the data after a regular conversion is not read (by the CPU or DMA) before the new conversion data is available, a buffer overflow event will be indicated by the overflow flag (OVR bit in the ADCx_ISR register).

If the EOC flag bit in the ADCx_ISR register is still 1 when a new conversion is completed, the OVR flag bit will be set to 1 by hardware. If the OVRIE bit in the ADCx_IER register is set to 1, an interrupt can be generated.

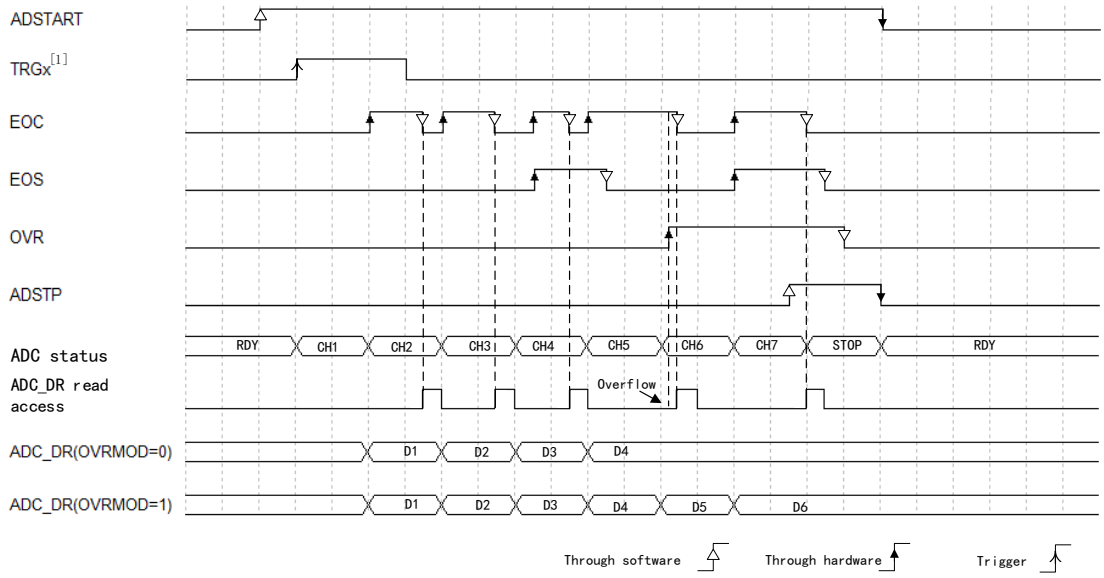
If an overflow occurs, the ADC will remain operating and can continue conversion unless the sequence is stopped and reset by setting the ADSTP bit in the ADCx_CR register to 1 via software.

The OVR flag bit in the ADCx_ISR register can be cleared to 0 by writing 1 to it via software.

The OVRMOD control bit in the ADCx_CFGR register can be programmed to 0 or 1 to configure whether to retain or overwrite the data in case of an overflow event:

- OVRMOD=0: In the event of an overflow event, the data register's data will be retained and be prevented from being overwritten: the original data will be retained, and the new conversion result will be discarded. If the OVR bit remains 1, the conversion can continue, but the result data will also be discarded.
- OVRMOD=1: The data register will be overwritten by the last conversion result, and previously unread data will be lost. If the OVR bit remains 1, the conversion can proceed normally, and the ADCx_DR register will always contain the latest conversion data.

Figure 103 Overflow Example (OVR)



Note: Since the four injected channels have dedicated data registers, overflow detection is not performed for the injected channels.

Management of conversion sequence without using DMA

If the conversion process is slow enough, software can be used to handle the conversion sequence. In this case, the software shall use the EOC flag bit in the ADCx_ISR register, and related interrupts to handle each data. Whenever a conversion ends, the EOC bit will be set to 1 by hardware, and the ADCx_DR register can be read. The OVRMOD bit in the ADCx_CFGR register shall be configured to 0 to manage overflow events as errors.

Management of conversion sequence without using DMA and without overflow

This may be useful when the ADC does not read data every time it converts one or more channels (e.g., when there is an analog watchdog). In this case, the OVRMOD bit in the ADCx_CFGR register shall be configured to 1, and the OVR flag bit in the ADCx_ISR register shall be ignored by software. Overflow events will not prevent the ADC from continuing conversion, and the ADCx_DR register always contains the latest conversion result.

Management of conversions using DMA

Since the converted channel values are stored in the unique data register, using DMA is very helpful for conversion of multiple channels. This can avoid losing data already stored in the ADCx_DR register.

If DMA mode is enabled (the DMAEN bit in the ADCx_CFGR register is set to 1), a DMA request will be generated after conversion of each channel. Then the

converted data can be transmitted from the ADCx_DR register to a target location selected by software.

However, if an overflow occurs (OVR=1) because the DMA cannot handle the DMA transmission request in time, the ADC will stop generating DMA requests, and the corresponding data of new conversion will not be transmitted via DMA. That is, DMA transmission requests will be disabled until the OVR bit is cleared to 0 via software. This means that all data transmitted to RAM can be considered valid.

According to the configuration of the OVRMOD bit in the ADCx_CFGR register, data can be retained or overwritten (see the ADC Overflow). Depending on the application, two different DMA modes are recommended, and the corresponding mode can be configured using the DMACFG bit of the ADCx_CFGR register:

If DMA is programmed to transmit a fixed number of data, DMA single mode shall be selected (DMACFG=0).

If DMA is programmed in circular mode, DMA circular mode shall be selected (DMACFG=1).

DMA single mode (DMACFG=0)

In this mode, the ADC will generate a DMA transmission request each time new conversion data appears. When the DMA reaches the last DMA transmission operation (DMA will generate a TXCIFLG interrupt, as seen in the DMA), and the ADC will stop generating DMA requests even if the conversion has started again.

After DMA transmission is completed (after all transmission operations configured in the DMA controller are completed):

- The content of the ADC data register is frozen
- Any ongoing conversion will be aborted, and partial results will be discarded.
- No new DMA requests will be transmitted to the DMA controller. This prevents overflow errors if there are still conversions that have already started.
- The scan sequence will stop and be reset
- DMA stops

DMA circular mode (DMACFG=1)

In this mode, each time new conversion data appears in the data register, the ADC will generate a DMA transmission request, even if the DMA has reached the last DMA transmission operation. Then the DMA in circular mode can be configured to handle continuous analog input data stream.

22.4.21 Dynamic low-power characteristics

Automatic Delayed Conversion Mode (AUTDLY)

The ADC operates in the automatic delayed conversion mode controlled by the AUTDLY configuration bit in the ADCx_CFGR register. Automatic delay conversion can be used to simplify software and optimize the performance of applications using low-frequency clocks (which may be at risk of ADC overflow).

When AUTDLY=1, a new conversion can only start when all previous data in the same group has been processed:

- For regular conversions: When the ADCx_DR register has been read or the EOC bit in the ADCx_ISR register has been cleared to 0.
- For injected conversions: When the JEOS bit in the ADCx_ISR register has been cleared to 0.

In this way, the speed of the ADC can be automatically adjusted to match the speed at which the system reads data.

The delay is inserted after each regular conversion (no matter whether the DISCEN bit in the ADCx_CFGR register is set to 0 or 1) and after each injected conversion sequence (no matter whether the JDISCEN bit in the ADCx_CFGR register is set to 0 or 1).

Note:

(1) A delay is not inserted between injected conversion sequences, but is inserted after the last sequence. During conversion, any hardware trigger event (for the same group conversion) that occurs during this delay period will be ignored.

(2) If a software trigger occurs during the delay, it will not be ignored, and conversion can be restarted within this delay time by setting the ADSTART or JADSTART bit in the ADCx_CR register to 1. The software will read the data before a new conversion is started.

No delay will be inserted between different groups of conversions (regular conversion is followed immediately by injected conversion, and vice versa):

- If an injected trigger occurs during the automatic delay of a regular conversion, the injected conversion will start immediately.
- After completion of an injected sequence, the ADC will wait for the delay of the previous regular conversion (if not finished) before starting a new regular conversion.

The operation in automatic injected mode (JAUTO bit in ADCx_CFGR register is set to 1) is slightly different. A new regular conversion can start only after the automatic delay of the previous injected conversion sequence has ended (when the JEOS bit in the ADCx_ISR register has been cleared to 0 by hardware). This ensures that software can read all data of a given sequence before starting a new sequence.

To stop the conversion in continuous automatic injected and automatic delay combined mode (the JAUTO bit in ADCx_CFGR register is set to 1, the CONT bit is set to 1, and the AUTDLY bit is set to 1). Please operate according to the following procedure:

(1) Wait until the JEOS bit in the ADCx_ISR register is set to 1 by hardware (no conversion will be restarted)

(2) Clear the JEOS bit to 0 by writing 1 to it

(3) Set the ADSTP bit in the ADCx_CR register to 1

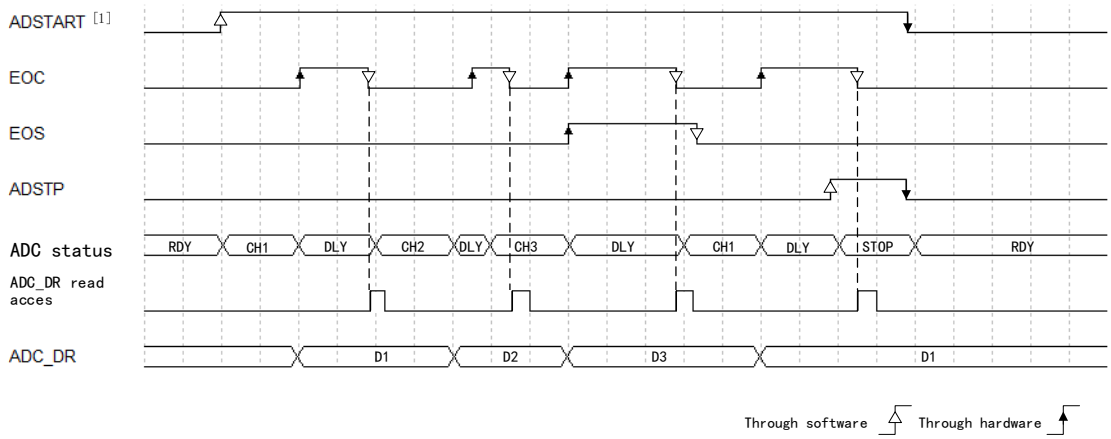
(4) Read the regular data

Otherwise, if the JEOS bit in the ADCx_ISR register is cleared to 0 after the ADSTP bit in the ADCx_CR register is set to 1, a new regular sequence might restart.

In AUTDLY mode, if a hardware regular trigger event occurs during an ongoing regular sequence or during the delay after the last regular conversion of a sequence, the trigger event will be ignored. However, if the trigger event occurs after this delay, even during the subsequent injected sequence's delay, it is considered pending. The conversion will then start at the end of the injected sequence's delay.

In AUTDLY mode, if a hardware injected trigger event occurs during an ongoing injected sequence or during the delay after the last injected conversion of a sequence, the trigger event will be ignored.

Figure 104 AUTODLY=1, Regular Conversion in Continuous Mode, Software Trigger



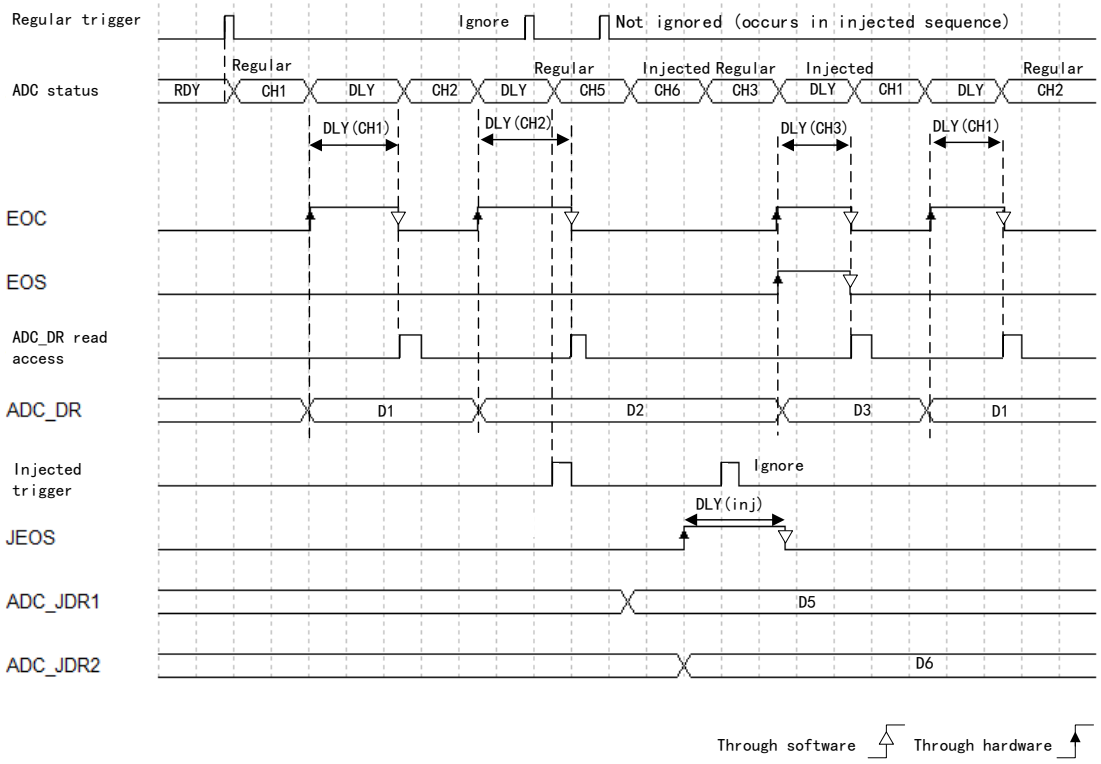
Note:

(1) The AUTDLY bit in the ADCx_CFGR register is set to 1.

(2) Regular configuration: The EXTEN[1:0] bits in the ADCx_CFGR register are set to 00 (software trigger), the CONT bit is set to 1, and channels are configured with CH1, CH2, CH3.

(3) Injected configuration is disabled.

Figure 105 AUTODLY=1, Regular Hardware Conversion Interrupted by Injected Conversion (DISCEN=0; JDISCEN=0)



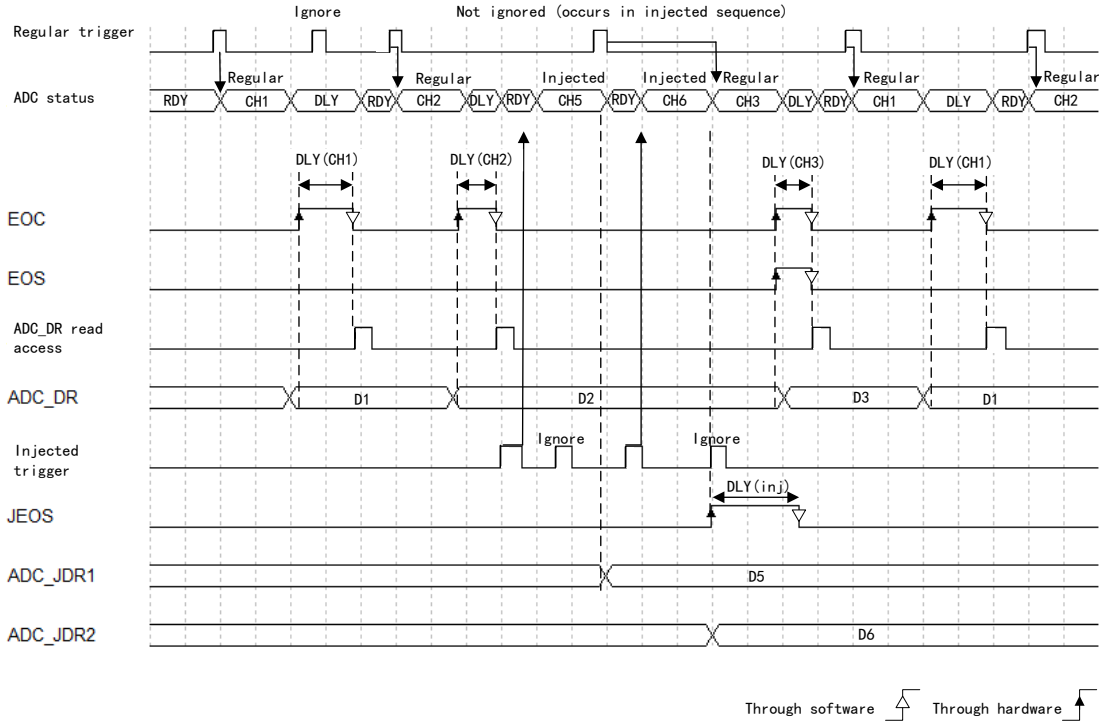
Note:

(1) The AUTDLY bit in the ADCx_CFGR register is set to 1

(2) Regular configuration: The EXTEN[1:0] bits in the ADCx_CFGR register are set to 01 (hardware trigger), the CONT bit is set to 0, the DISCEN bit is set to 0, and channels are configured with CH1, CH2, CH3.

(3) Injected configuration: The JEXTEN[1:0] bits in ADCx_JSQR register are set to 01 (hardware trigger), the JDISCEN bit in ADCx_CFGR register is set to 0, and channels are configured with CH5 and CH6.

Figure 106 AUTODLY=1, Regular Hardware Conversion Interrupted by Injected Conversion (DISCEN=1; JDISCEN=1)



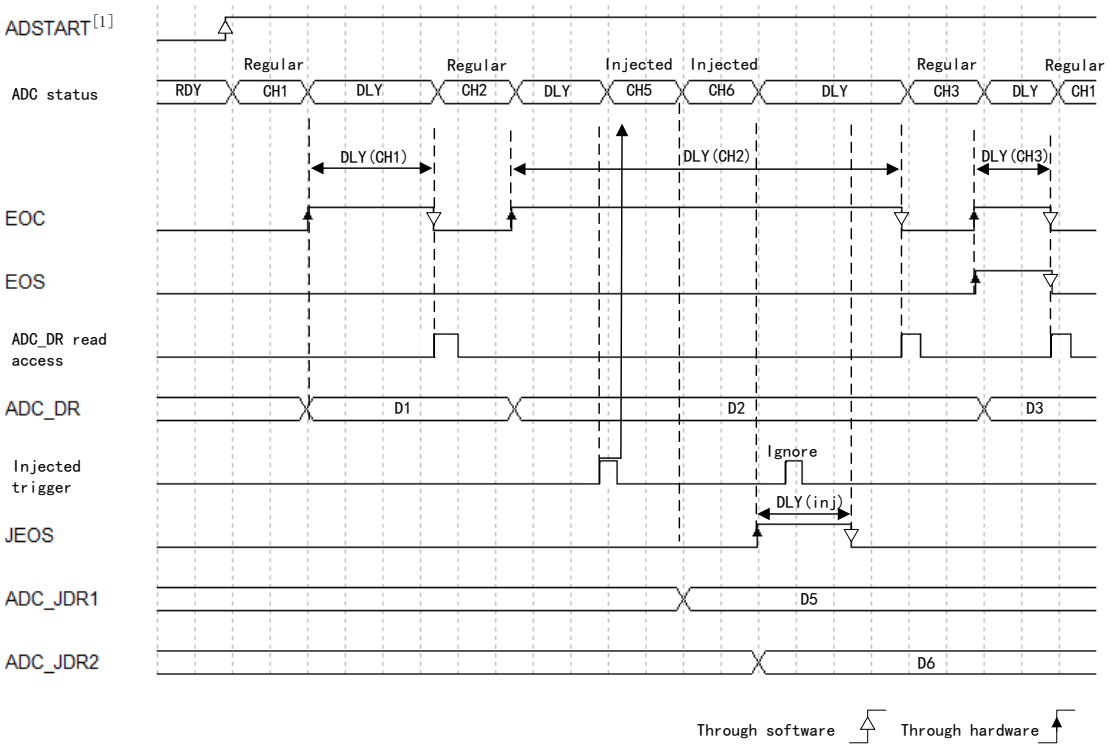
Note:

(1) The AUTDLY bit in the ADCx_CFGR register is set to 1.

(2) Regular configuration: The EXTEN[1:0] bits in the ADCx_CFGR register are set to 01 (hardware trigger), the CONT bit is set to 0, the DISCEN bit is set to 1, the DISCNUM bit is set to 1, and channels are configured with CH1, CH2, CH3.

(3) Injected configuration: The JEXTEN[1:0] bits in ADCx_JSQR register are set to 01 (hardware trigger), the JDISCEN bit in ADCx_CFGR register is set to 1, and channels are configured with CH5 and CH6.

Figure 107 AUTODLY=1, Regular Continuous Conversion Interrupted by Injected Conversion



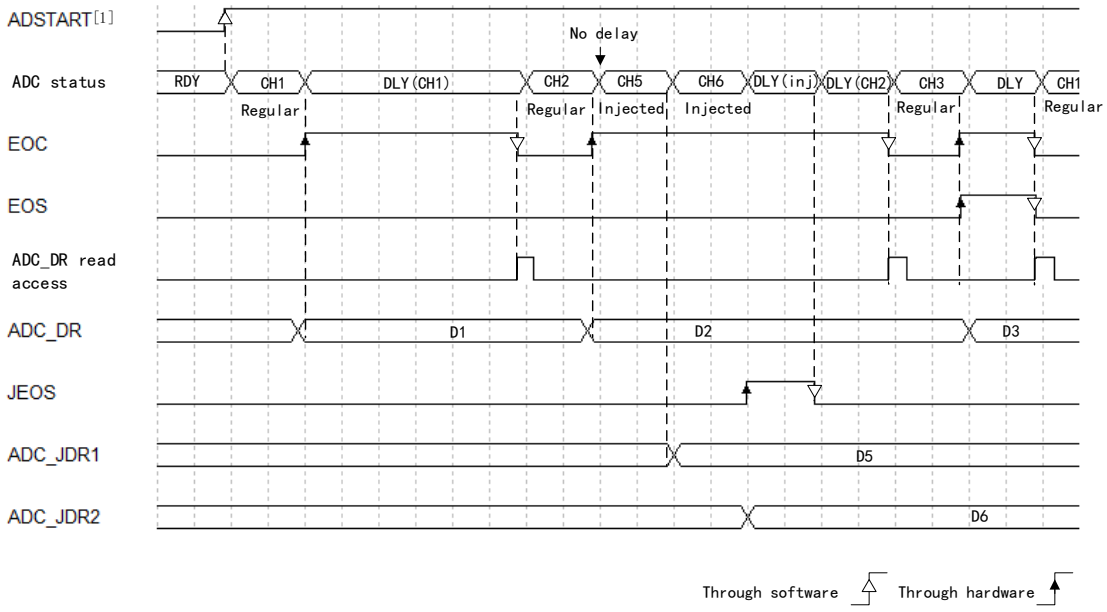
Note:

(1) The AUTDLY bit in the ADCx_CFGR register is set to 1.

(2) Regular configuration: The EXTEN[1:0] bits in the ADCx_CFGR register are set to 00 (software trigger), the CONT bit is set to 1, the DISCEN bit is set to 0, and channels are configured with CH1, CH2, CH3.

(3) Injected configuration: The JEXTEN[1:0] bits in ADCx_JSQR register are set to 01 (hardware trigger), the JDISCEN bit in ADCx_CFGR register is set to 0, and channels are configured with CH5 and CH6.

Figure 108 AUTODLY=1, Automatic Injected Mode (JAUTO=1)



Note:

(1) The AUTDLY bit in the ADCx_CFGR register is set to 1.

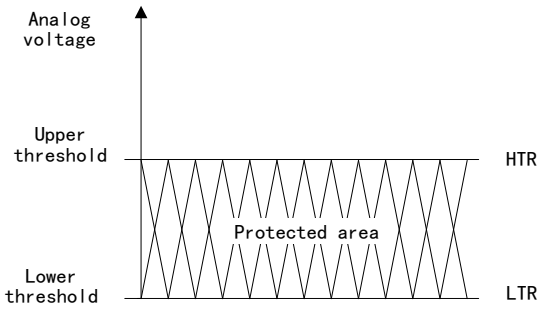
(2) Regular configuration: The EXTEN[1:0] bits in the ADCx_CFGR register are set to 00 (software trigger), the CONT bit is set to 1, the DISCEN bit is set to 0, and channels are configured with CH1, and CH2.

(3) Injected configuration: The JAUTO bit in ADCx_CFGR register is set to 1, and channels are configured with CH5 and CH6.

22.4.22 Analog window watchdogs (AWD1EN, JAWD1EN, AWD1SGL, AWD1CH, AWD2CH, AWD3CH, AWD_HTx, AWD_LTx, AWDx)

Three AWD analog watchdogs monitor whether certain channels remain within the configured voltage range (window).

Figure 109 Protected Area of Analog Watchdog



AWDx flags and interrupts

Interrupts can be enabled respectively for the three analog watchdogs by setting

the AWDyIE (y=1, 2, 3) bits in the ADCx_IER register to 1. The AWDy (y=1, 2, 3) flags in the ADCx_ISR register can be cleared to 0 by writing 1 to them.

Description for analog watchdog 1

Setting the AWD1EN bit to 1 in the ADCx_CFGR register can enable the AWD analog watchdog 1. This watchdog monitors a selected channel or all enabled channels to see if they remain within the configured voltage range (window).

The table below introduces how to configure the ADCx_CFGR register to enable the analog watchdog on one or multiple channels.

Table 84 Analog Watchdog Channel Selection

Channels protected by analog watchdog	AWD1SGL bit	AWD1EN bit	JAWD1EN bit
None	x	0	0
All injected channels	0	0	1
All regular channels	0	1	0
All regular and injected channels	0	1	1
Single injected channel	1	0	1
Single regular channel	1	1	0
Single regular or injected channel	1	1	1

Select via the AWD1CH[4:0] bits in ADCx_CFGR register. Additionally, the channel must be programmed for conversion in the appropriate regular or injected sequence.

If the analog voltage for the ADC conversion falls below the lower threshold or exceeds the upper threshold, the AWD1 analog watchdog status bit in the ADCx_ISR register will be set to 1. These thresholds are programmed into the HT1[15:0] and LT1[15:0] bits of the ADCx_TR1 register for analog watchdog 1.

Description for analog watchdogs 2 and 3

The second and third analog watchdogs are more flexible, and multiple selected channels can be protected by programming the corresponding bits in AWDCHy[5:0] (y=2, 3) in the ADC_AWD2CR register. Setting any bit of AWDCHy[5:0] (y=2, 3) to 1 can enable the analog watchdog of corresponding channel.

The value that can be set for the analog watchdog 2 is 12 bits, and it is only compared with the higher 12 bits of the conversion result. The value that can be set for the analog watchdog 3 is 8 bits, and it is only compared with the higher 8 bits of the conversion result.

22.4.23 Oversampler

The oversampling design supports oversampling ratios of 2/4/8/16 times, the

formula for ADC oversampling is as follows:

$$Result = \frac{1}{M} \times \sum_{n=0}^{n=N-1} Conversion\ (tn)$$

Where the oversampling ratio N is defined using the OVFS[2:0] bits in the ADC_CFGR2 register, ranging from 2x to 16x. The partition coefficient M is defined using the OVSS[3:0] bits in the ADC_CFGR2 register, ranging from No-shift to 4-bit shift.

The summation unit can produce a result of up to 20 bits (16x16-bit results), which is first right-shifted. It is then truncated to the 16 least significant bits, rounded to the nearest value using the least significant bits left by the shift, and finally transmitted to the ADC_DR data register.

The table below shows the data format for various combinations of N and M, using raw conversion data equal to 0xFFF. (The data highlighted in yellow indicates the occurrence of data overflow.)

Table 85 Oversampling Combination

Oversampling Ratio	**Maximum conversion data**	**No-shift**	**1-bit shift**	**2-bit shift**	**3-bit shift**	**4-bit shift**
		OVSS=0000	OVSS=0001	OVSS=0010	OVSS=0011	OVSS=0100
2x	0x1FFFE	0x1FFFE	0xFFFF	0x7FFF	0x3FFF	0x1FFF
4x	0x3FFFC	0x3FFFC	0x1FFFE	0xFFFF	0x7FFF	0x3FFF
8x	0x7FFF8	0x7FFF8	0x3FFFC	0x1FFFE	0xFFFF	0x7FFF
16x	0xFFFF0	0xFFFF0	0x7FFF8	0x3FFFC	0x1FFFE	0xFFFF

In oversampling mode, the conversion time does not change. That is, the sampling time remains unchanged throughout the oversampling sequence. A new data is provided every N conversions, with an equivalent delay of $NxT_{CONV}=Nx（t_{SMPL}+t_{SAR}$). The ADC flag bits are set as follows:

- The bit for the end of the sampling phase (EOSMP in the ADCx_ISR register) is set to 1 after each sampling phase
- The bit for the end of conversion (EOC in the ADCx_ISR register) is set to 1 once every N conversions when the oversampling result is available
- The bit for the end of sequence (EOS in the ADCx_ISR register) is set to 1 after the completion of the oversampling data sequence (i.e., after the conversion of Nx sequence length)

22.4.23.1 **ADC operating modes supported during oversampling (single ADC)**

In ADC oversampling mode, the following operating modes are still supported:

- Single or continuous mode conversion
- Start of ADC conversion by software or trigger

- ADC stops (or aborts) during conversion
- Read Data by CPU or DMA with overflow detection
- Low-power mode, i.e., delayed conversion mode (AUTDLY)

Note: Calibration mode is not available when processing oversampling data.

Oversampling mode does not support offset correction. That is, the OFFSETy_EN bit in the ADCx_OFFSETy register is ignored.

The ADC's oversampling mode is configured by the following four bits in the ADCx_CFGR2 register:

- Regular oversampling (ROVSE)
- Injected oversampling (JOVSE)
- Oversampler mode (continuous mode or resumed mode) (ROVSM)
- Regular mode with trigger (TROVS)

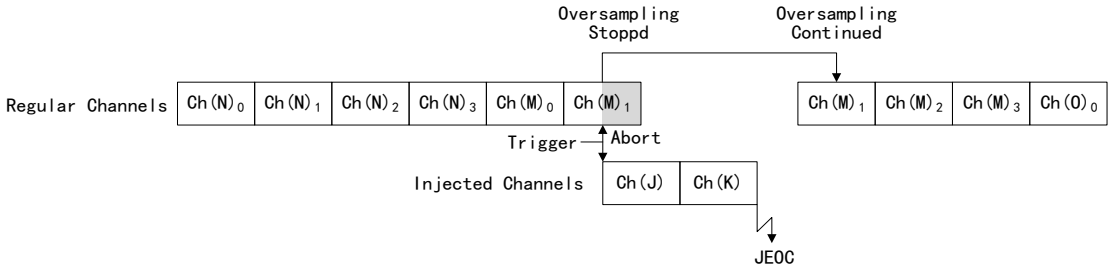
The table below lists the ADC oversampling operating modes under different configurations. X indicates it can be ignored.

Table 86 ADC Oversampling Operating Modes under Different Configurations.

ROVSE	JOVSE	ROVSM	TROVS	Comment
1	0	0	0	Regular channels only + continuous mode oversampling
1	0	0	1	Not supported
1	0	1	0	Regular channels only + resumed mode oversampling
1	0	1	1	Regular channels only with trigger + resumed mode oversampling
1	1	0	X	Not supported
1	1	1	0	Injected and regular channels + resumed mode oversampling
1	1	1	1	Not supported
0	1	X	X	Injected channel oversampling only

Regular channels + continuous mode oversampling

Figure 110 Regular Channels + Continuous Mode Oversampling

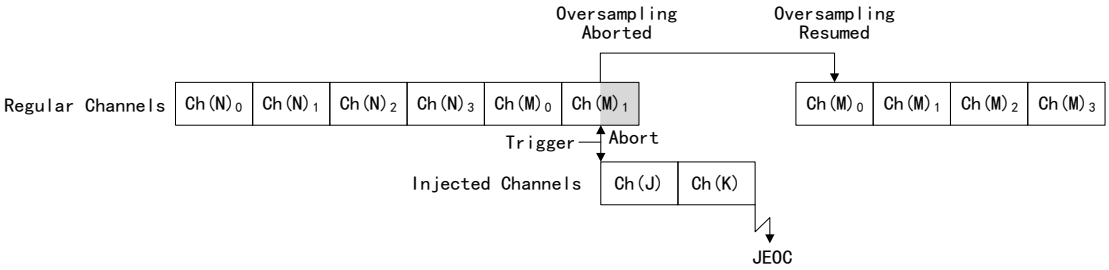


After the oversampling of regular channels is interrupted by injected channels, the resumed sampling will continue to sample from the last valid data, ensuring

the integrity of the oversampled data.

Regular channels only + resumed mode oversampling

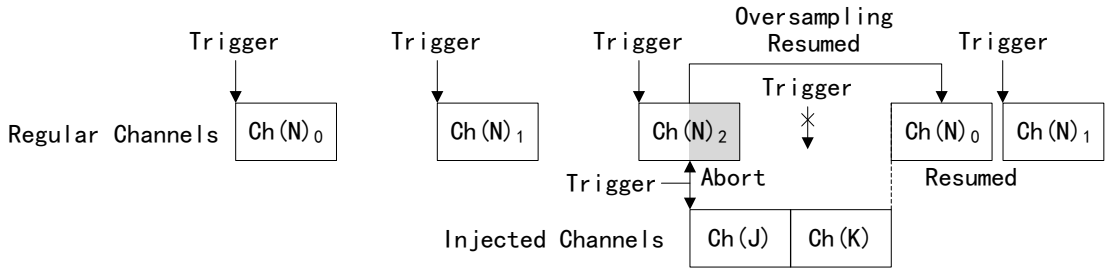
Figure 111 Regular Channels Only + Resumed Mode Oversampling



After the oversampling of regular channels is interrupted by injected channels, resumed sampling will start from the beginning of the interrupted data set, ensuring that all data used for oversampling undergoes back-to-back conversions within one time slot.

Regular channels only with trigger + resumed mode oversampling

Figure 112 Regular Channels Only With trigger + Resumed Mode Oversampling

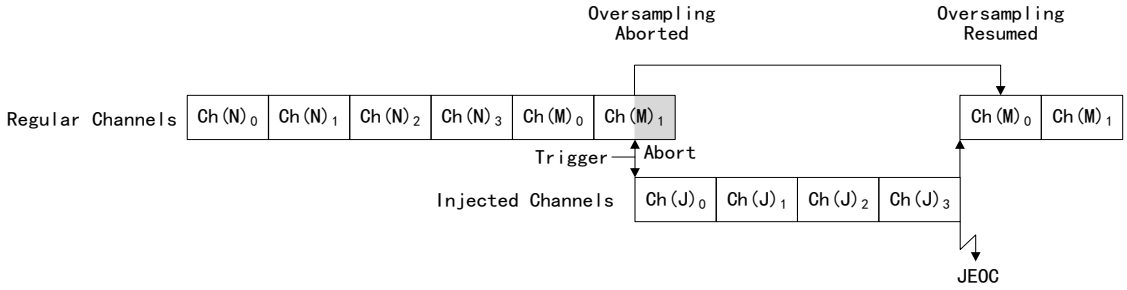


In a specific discontinuous mode, the regular channels start conversion in response to the trigger, allowing a user-defined oversampling frequency that is independent of the conversion time itself. The oversampler is forced into resumed mode. After being interrupted by injected channel sampling, the resumed sampling will restart from the beginning of the interrupted set of data.

Injected and regular channels + resumed mode oversampling

In this mode, the injected channels also enable the oversampling, and the regular channels are forced to resumed mode oversampling.

Figure 113 Injected and Regular channels + Resumed mode oversamplin



Injected channel oversampling only

In this mode, the regular channels sample normally, and only injected channels oversample.

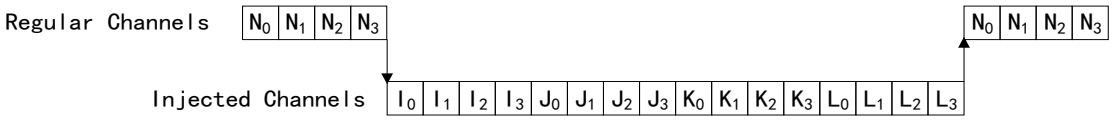
Automatic injected oversampling

In the mode of (simultaneous oversampling of injected and regular channels), oversampling can be performed on the automatic injected sequence by setting the JAUTO bit in the ADCx_CFGR register to 1, and all conversion results are stored in registers to save DMA resources.

Additionally, the automatic injected mode oversampling of trigger mode can be enabled by setting the TROVS bit in the ADCx_CFGR2 register to 1.

Note: The ROVSE and JOVSE bits in the ADCx_CFGR2 register shall be 1; other combinations are not supported.

Figure 114 Automatic Injected Oversampling



Oversampling in dual-ADC mode

When configuring dual ADC, oversampling can also be enabled for injected synchronous mode and regular synchronous mode. However, both ADC shall use identical settings (including oversampling).

When regular or injected oversampling is enabled on both ADC (ROVSE = 1 or JOVSE = 1 in the ADCx_CFGR2 register), other dual-ADC modes (alternate mode, alternate trigger mode, and modes of combining all modes) are no longer supported.

22.4.24 Dual-ADC mode

The dual-ADC mode can be used on the devices with dual ADC.

In dual-ADC mode, the start of conversion is triggered by the master ADC. It can

be triggered alternately or simultaneously by the slave ADC, depending on the mode selected by the DUALMOD[3:0] bits in the ADCx_CR register.

Four possible modes are implemented:

- Injected simultaneous mode
- Regular simultaneous mode
- Alternate mode
- Alternate trigger mode

These modes can also be combined as follows:

- Combined injected simultaneous mode + regular simultaneous mode
- Combined regular simultaneous mode + alternate trigger mode
- Combined injected simultaneous mode + alternate mode

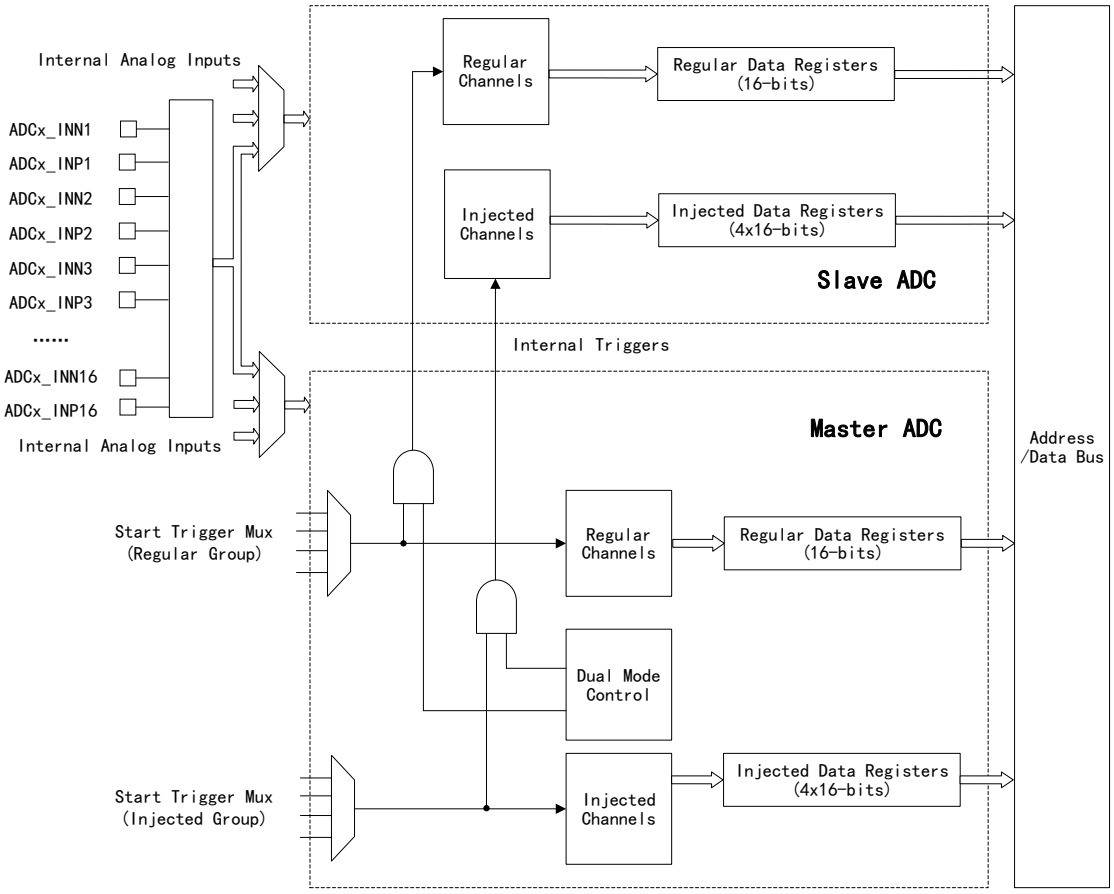
In dual ADC mode (when the DUALMOD[3:0] bits in the ADCx_CR register do not equal zero), the CONT, AUTDLY, DISCEN, DISCNUM[2:0], JDISCEN, JQM, and JAUTO bits in the ADCx_CFGR register are shared between the master and slave ADC: the bits in the slave ADC are always equal to the corresponding bits in the master ADC.

To start a conversion in dual mode, the user shall only program the EXTEN[1:0] and EXTSEL[3:0] bits in the ADCx_CFGR register of the master ADC, and the JEXTSEL[3:0] and JEXTEN[1:0] bits in the ADCx_JSQR register to configure software or hardware trigger and regular or injected trigger (the EXTEN[1:0] and JEXTEN[1:0] bits of the slave ADC do not need to be configured).

In regular simultaneous or alternate mode: Once the user sets the ADSTART or ADSTP bit in the ADCx_CR register of the master ADC, the corresponding bit in the slave ADC will be automatically set. However, the ADSTART or ADSTP bit of the slave ADC does not need to be cleared simultaneously with the master ADC bit.

In injected simultaneous or alternate trigger mode: Once the user sets the JADSTART or JADSTP bit in the ADCx_CR register of the master ADC, the corresponding bit in the slave ADC will be automatically set. However, the JADSTART or JADSTP bit of the slave ADC does not need to be cleared simultaneously with the master ADC bit. In dual-ADC mode, the data from both master and slave ADC after conversion can be read in parallel by reading the master ADC data register (ADC1_DR). The status bit cannot be read in parallel.

Figure 115 Dual ADC Block Diagram



Injected simultaneous mode

This mode is selected by setting the DUALMOD[3:0] bits in the ADC1_CR register of master ADC to 0101. It converts the injected channel group. The external trigger source comes from the master ADC's injected group multiplexer (selected by the JEXTSEL bit in the ADC1_JSQR register).

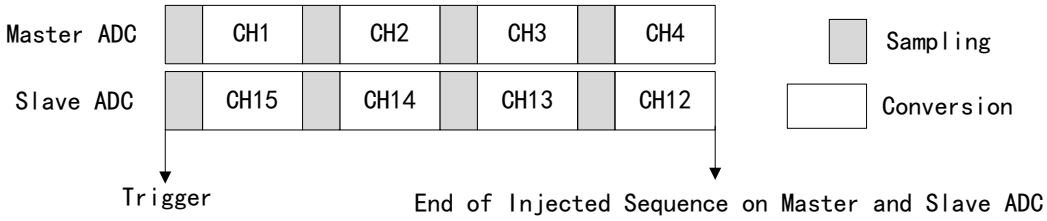
In synchronous mode, sequences of the same length must be converted, or it shall be ensured that the interval between triggers is greater than the length of the two sequences. Otherwise, the ADC with the shortest sequence might restart while the ADC with the longest sequence is completing the previous conversion.

Regular conversions can be executed on one or all ADC. In this case, they are independent of each other and are interrupted when an injected event occurs. They are resumed at the end of the injected conversion group.

- At the end of the conversion event injection sequence (JEOS) on the master ADC, the converted data is stored in the master ADC1_JDRy register, and a JEOS interrupt is generated (if enabled).

- At the end of the conversion event injection sequence (JEOS) on the slave ADC, the converted data is stored in the slave ADC2_JDRy register, and a JEOS interrupt is generated (if enabled).
- If the duration of the master injection sequence equals that of the slave injection sequence, the software might only enable one of the two JEOS interrupts and read the converted data (from both master ADC1_JDRy and slave ADC2_JDRy registers).

Figure 116 Injected Simultaneous Mode



If JDISCEN = 1 in the ADCx_CFGR register, an injection trigger event will occur for each simultaneous conversion of the injection sequence. This mode can be combined with AUTDLY mode:

- Once a simultaneous injected conversion sequence ends, a new injected trigger event is accepted only when the JEOS bits in the ADCx_ISR registers of both master and slave ADC are cleared (delay phase). Any new injected trigger event occurring during an ongoing injected sequence and related delay phase will be ignored.
- Once a regular conversion sequence on the master ADC ends, a new regular trigger event for the master ADC is accepted only when the master data register (ADC1_DR) is read. Any new regular trigger event occurring on the master ADC during an ongoing regular sequence, and the related delay phase will be ignored. The regular sequence occurring on the slave ADC exhibits the same behavior.

Regular simultaneous mode of independent injection

This mode is selected by setting the DUALMOD[3:0] bits in the ADC1_CR register of master ADC to 0110. This mode is executed on a set of regular channels. The external trigger source comes from the master ADC's regular group multiplexer (selected by the EXTSEL bit in the ADC1_CFGR register). A synchronous trigger is provided to the slave ADC. In this mode, independent injected conversions are supported. An injection request (regardless of on the master or slave) aborts the current simultaneous conversions. These conversions will restart once the injected conversion is completed.

In regular synchronous mode, sequences of the same length must be converted, or it shall be ensured that the interval between triggers is greater

than the longer conversion time of the two sequences. Otherwise, the ADC with the shortest sequence might restart while the ADC with the longest sequence is completing the previous conversion.

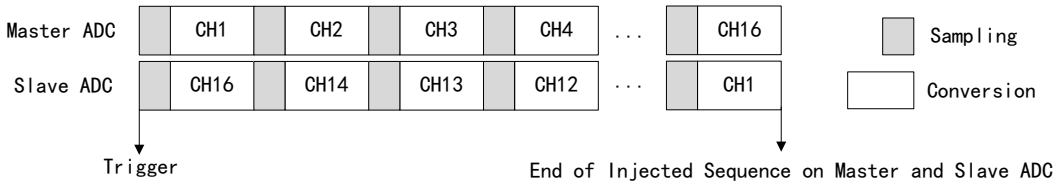
When software can read data, interrupts will notify it:

- At the end of each conversion event (EOC) on the master ADC, a master EOC interrupt will be generated (if the EOCIE bit in the ADC1_IER register is enabled), and the software can read the master ADC's ADC1_DR.
- At the end of each conversion event (EOC) on the slave ADC, a slave EOC interrupt will be generated (if the EOCIE bit in the ADC2_IER register is enabled), and the software can read the slave ADC's ADC2_DR.
- If the duration of the master regular sequence equals that of the slave regular sequence, the software may only enable the master EOC interrupt and read both converted data from the master ADC's data register (ADC1_DR).

DMA can also be used to read regular data. There are two ways:

- Use two DMA channels (one for master ADC, one for slave ADC).
  - Configure the DMA master ADC channel to read ADC1_DR from the master ADC; the DMA request will be generated in each EOC event of the master ADC.
  - Configure the DMA slave ADC channel to read ADC2_DR from the slave ADC; the DMA request will be generated in each EOC event of the slave ADC.

Figure 117 Regular Simultaneous Mode



If the DISCEN bit in the ADCx_CFGR register is configured to 1, a regular trigger event is required for every 'n' simultaneous conversions of a regular sequence (“n” is defined by the DISCNUM[2:0] bits in the ADCx_CFGR register).

This mode can be combined with AUTDLY mode:

- Once the simultaneous conversions of a sequence ends, the next conversion in the sequence will start only when the regular data register of the master ADC is read (delay phase).
- Once a simultaneous regular conversion sequence ends, a new regular trigger event is accepted only when the master ADC's

regular data register is read (delay phase). Any new regular trigger event occurring during an ongoing regular sequence and related delay phase will be ignored.

DMA can be used to process data in regular simultaneous mode combined with AUTDLY mode.

When the regular simultaneous mode is combined with AUTDLY mode, it shall be ensured that:

- The number of conversions in the master sequence equals that in the slave sequence.
- For each simultaneous conversion of the sequence, the conversion length of the slave ADC is lower than that of the master ADC. Note that the length of a sequence depends on the number of channels to be converted, sampling time, and resolution of each channel.

This combination of regular simultaneous mode with AUTDLY mode is limited to use cases programmed for regular channels: programming for injected channels in this combined mode is disabled.

Alternate mode of independent injection

This mode is selected by setting the DUALMOD[3:0] bits in the ADC1_CR register of master ADC to 0111.

This mode can only be started on a regular group (usually one channel). The external trigger source comes from the master ADC's regular channel multiplexer.

After an external trigger occurs:

- The master ADC starts immediately.
- After the sampling phase of the master ADC is over, the slave ADC starts after a delay of several ADC clock cycles.

The minimum delay is 1 ADCCLK. This delay begins counting half a cycle after the end of the sampling phase of the master conversion. In this way, the ADC cannot start a conversion if the complementary ADC is still sampling its input (only one ADC can sample the input signal at a given time).

If the CONT bit in the ADCx_CFGR register is set to 1 on both master and slave ADC, the selected regular channels on both ADC will convert continuously. Interrupts will notify the software when software can read data at the end of each conversion event (EOC) on the slave ADC. At this time, the master/slave ADC's EOC interrupt will be generated (if the EOCIE bit in the ADCx_IER register is enabled), and the software can read data from the master/slave ADC's ADCx_DR.

Note: It is allowed to enable only the slave's EOC interrupt and read the master's data register (ADC1_DR). However, in this case, the user shall ensure that the conversion duration is compatible to

ensure that in the sequence, the master conversion is always followed by the slave conversion before a new master conversion is started.

Figure 118 Alternate mode of Independent Injection (Continuous)

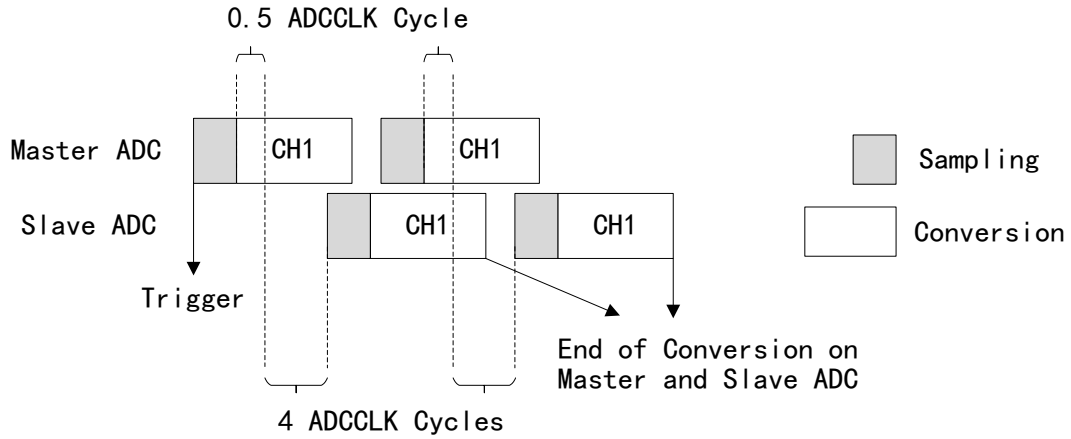
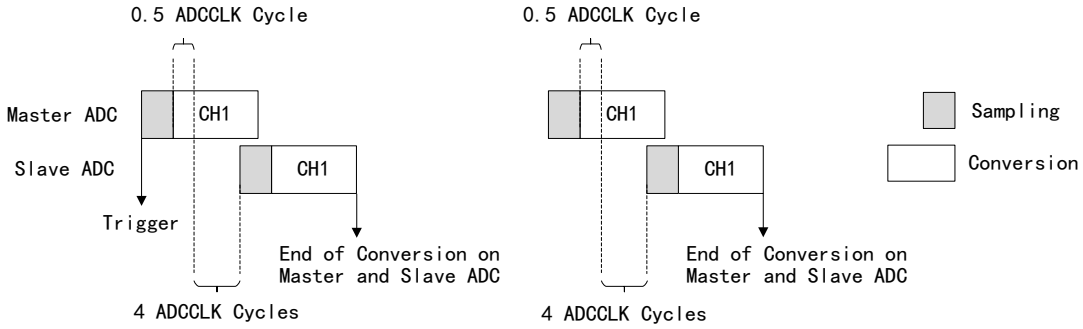


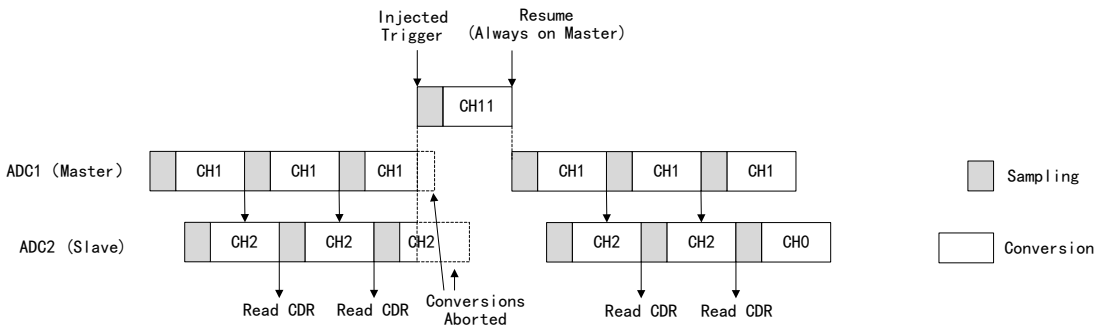
Figure 119 Alternate mode of Independent Injection (Discontinuous)



If DISCEN = 1 in the ADCx_CFGR register, a regular trigger event is required for every “n” simultaneous conversions of a regular sequence (“n” is defined by the DISCNUM[2:0] in the ADCx_CFGR register).

In this mode, injected conversions are supported. When an injection is completed (no matter on the master or slave ADC), the regular conversions of both master and slave ADC will be terminated, and the sequence will restart from the master ADC.

Figure 120 Diagram of Injection Occurring during Alternate Conversion Process



Alternate trigger mode

This mode is selected by setting the DUALMOD[3:0] bits in the ADC1_CR register of master ADC to 1001. This mode can only be started on an injected group. The external trigger source comes from the master ADC's injected group multiplexer.

This mode is only available when the hardware trigger is selected: JEXTEN[1:0] in the ADCx_JSQR register cannot be 00.

When injected discontinuous mode is disabled (JDISCEN = 0 in the ADCx_CFGR register of both ADC)

(1) When the first trigger occurs, all injected channels of the master ADC in the group will be converted. If the JEOS interrupt is enabled, it will be generated after all injected channels of the master ADC in the group are converted.

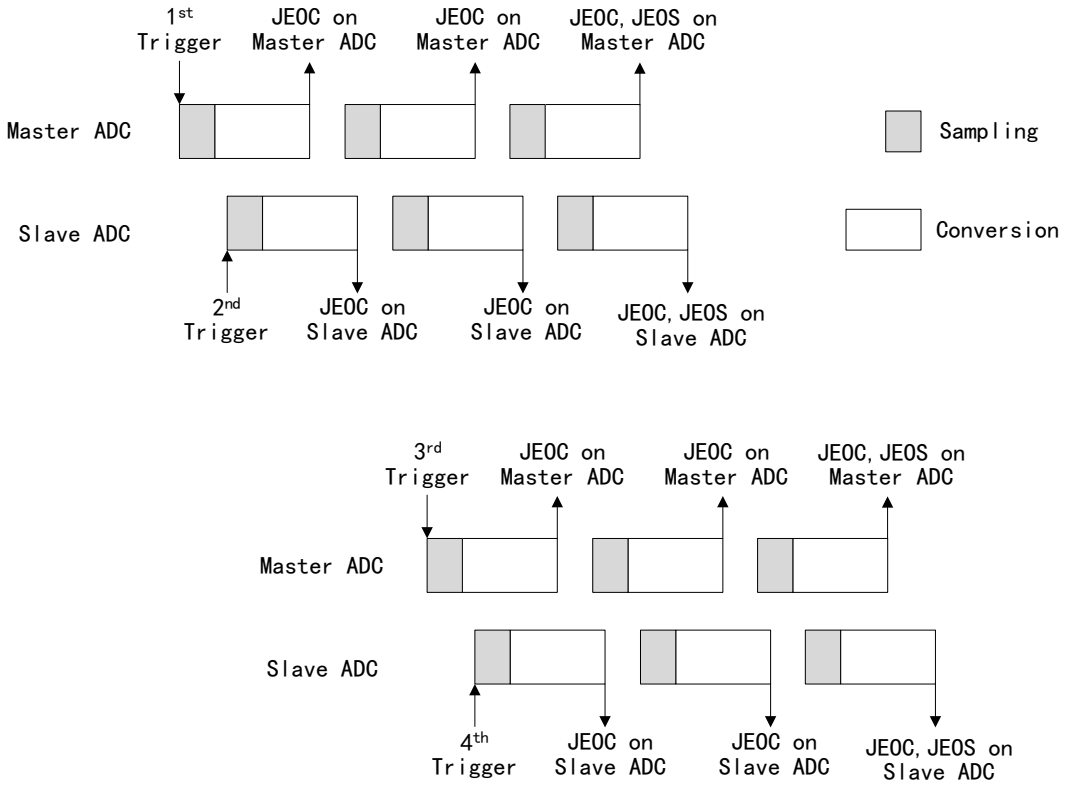
(2) When the second trigger occurs, all injected channels of the slave ADC in the group will be converted. If the JEOS interrupt is enabled, it will be generated after all injected channels of the slave ADC in the group are converted.

(3) And so on……

If the JEOC interrupt is enabled, it will be generated after each injected conversion.

If another external trigger occurs after all injected channels in the group are converted, the standby trigger process will restart by converting the injected channels of the master ADC in the group.

Figure 121 Alternate Trigger Mode



Note: Regular conversions can be enabled on one or all ADC. In this case, regular conversions are independent of each other. When ADC must execute injected conversion, the regular conversion will be interrupted. When the injected conversion is completed, the regular conversion will be restored. The time interval between two trigger events shall be greater than or equal to one ADC clock cycle. The minimum time interval between two trigger events that start conversion on the same ADC is the same as that in single-ADC mode.

When injected discontinuous mode is enabled (JDISCEN = 1 in the ADCx_CFGR register of both ADC)

If the injected discontinuous mode is enabled for both master and slave ADC:

(1) When the first trigger occurs, the first injected channel of the master ADC will be converted. If the JEOS interrupt is enabled, it will be generated after all injected channels of the master ADC in the group are converted.

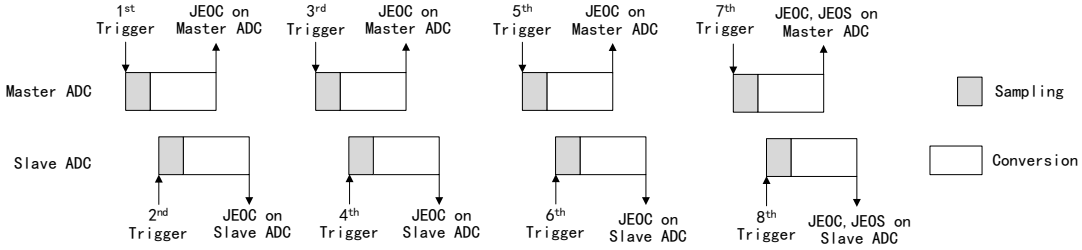
(2) When the second trigger occurs, the first injected channel of the slave ADC will be converted. If the JEOS interrupt is enabled, it will be generated after all injected channels of the slave ADC in the group are converted.

(3) And so on……

If the JEOC interrupt is enabled, it can also be generated after each injected conversion.

If another external trigger occurs after all injected channels in the group are converted, the alternate trigger process will restart.

Figure 122 Diagram of Alternate Trigger Mode in Discontinuous Mode



Combined regular/injected simultaneous mode

This mode is selected by setting the DUALMOD[3:0] bits in the ADC1_CR register of master ADC to 0001. Synchronous conversions of a regular group can be interrupted to start synchronous conversion of an injected group.

Note: In combined regular/injected synchronous mode, sequences of the same length must be converted, or it shall be ensured that the interval time between triggers is greater than the long conversion time of the two sequences. Otherwise, the ADC with the shortest sequence might restart while the ADC with the longest sequence is completing the previous conversion.

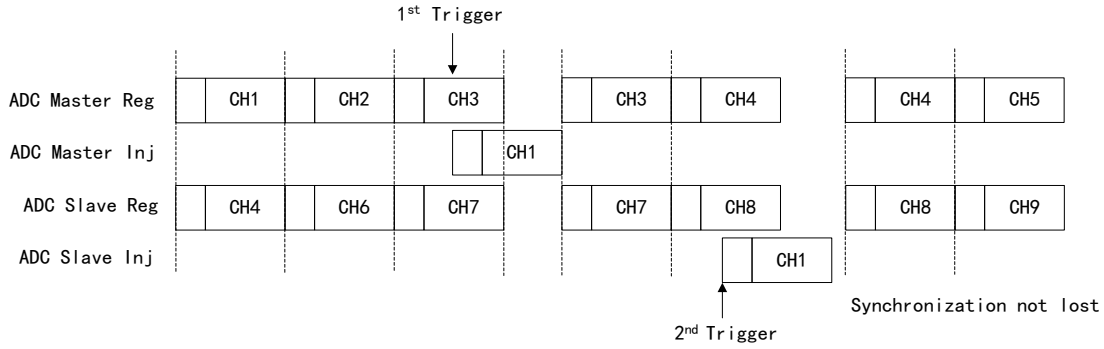
Combined regular simultaneous + alternate trigger mode

This mode is selected by setting the DUALMOD[3:0] bits in the ADC1_CR register of master ADC to 0010.

Simultaneous conversions of a regular group can be interrupted to start alternate trigger conversions of an injected group. The figure below illustrates the behavior of an alternate trigger interrupting the simultaneously ongoing regular conversion. The injected alternate conversion starts immediately after the injection event. If a regular conversion has already been running, to ensure the synchronization after the injected conversion, all (master/slave) ADC regular conversions will stop and be synchronously resumed at the end of the injected conversion.

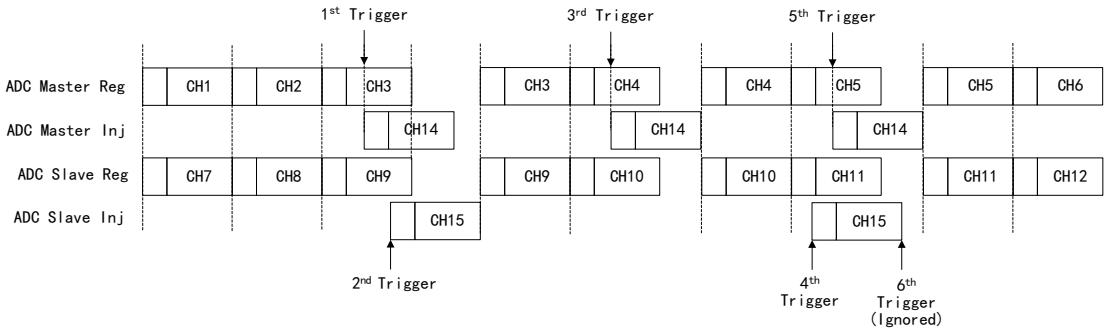
Note: In the regular synchronous + alternate combined trigger mode, sequences of the same length must be converted, or it shall be ensured that the interval time between triggers is greater than the long conversion time of the two sequences. Otherwise, the ADC with the shortest sequence might restart while the ADC with the longest sequence is completing the previous conversion.

Figure 123 Alternate + Regular Simultaneous Mode



If a trigger occurs during an injected conversion that interrupts a regular conversion, an alternate trigger will be provided. The figure below illustrates the behavior in this case (the sixth trigger is ignored because the related alternate conversion has not been completed).

Figure 124 Alternate (trigger occurs during injection conversion) + Regular Mode



Combined injected simultaneous + alternate mode

This mode is selected by setting the DUALMOD[3:0] bits in the ADC1_CR register of master ADC to 0011. Alternate conversion can be interrupted by a simultaneous injection event.

In this case, the alternate conversion is interrupted immediately, while an injected conversion starts. At the end of the injected sequence, the alternate conversion resumes. When the alternate regular conversion resumes, the first regular conversion performed is always the master regular conversion.

In this mode, the master ADC's data register shall be used for a read access to read the regular data. Otherwise, consistency between master and slave data cannot be guaranteed.

Figure 125 Single-channel CH0 Interleaved Mode + Injected Sequence CH11, CH12

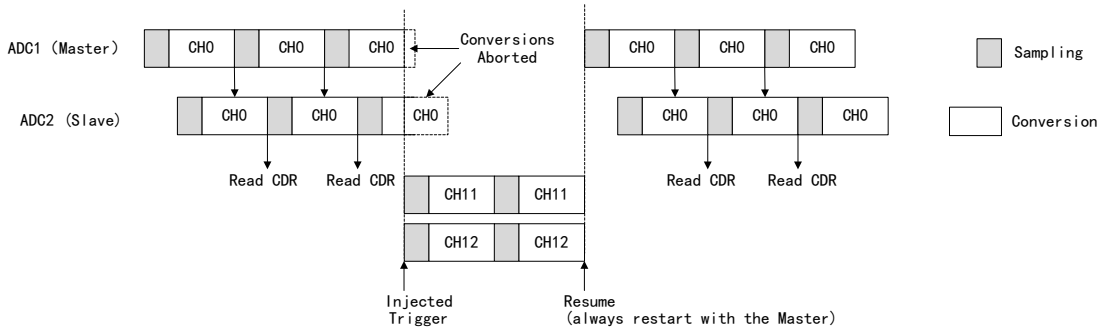


Figure 126 Double-channel interleaved Mode + Injected Sequence CH11, CH12 (Master Interrupt Priority)

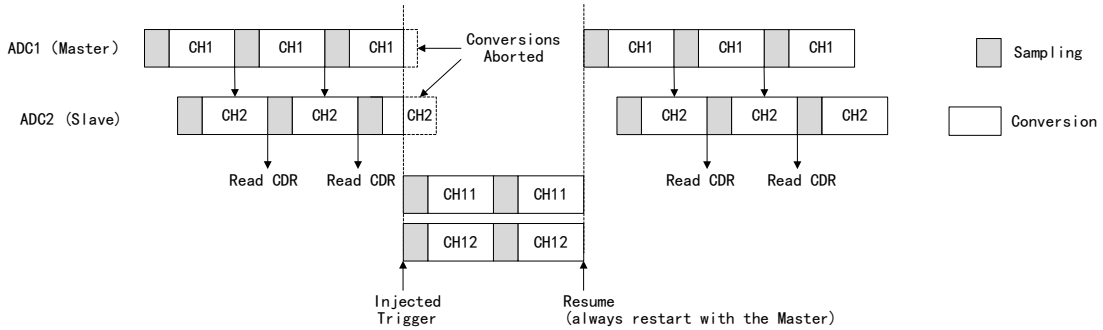
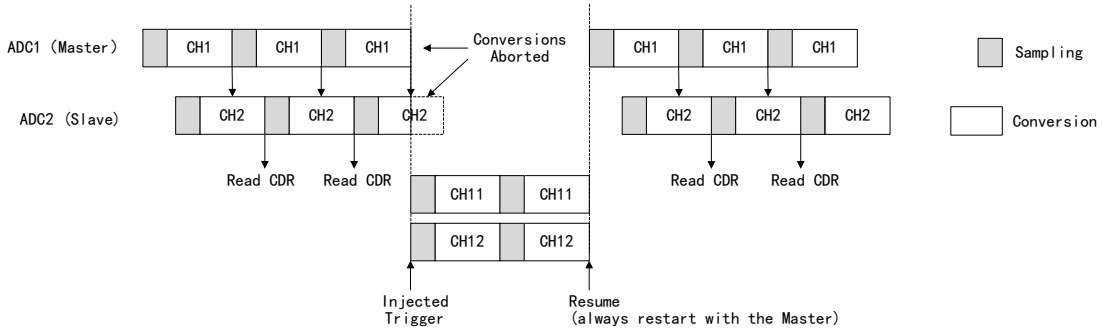


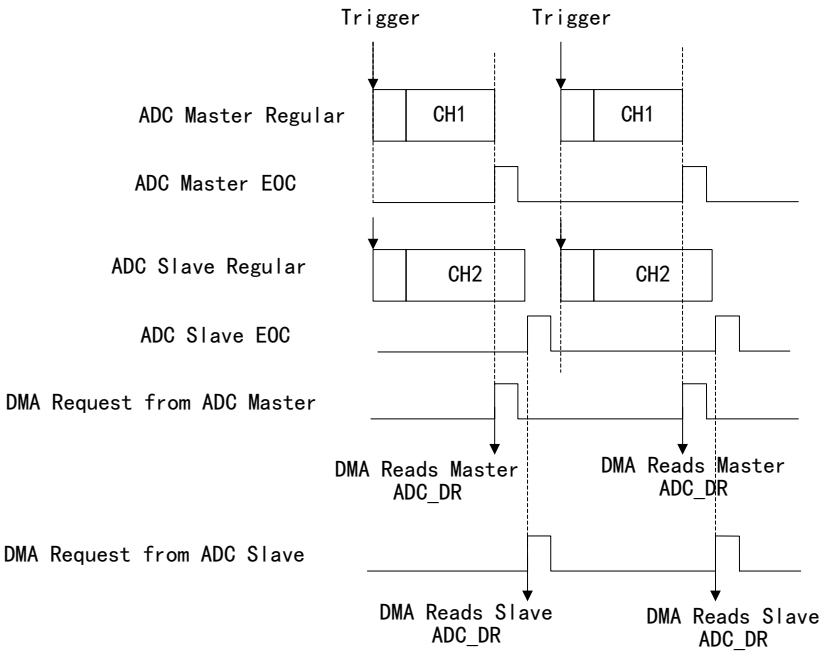
Figure 127 Double-channel interleaved Mode + Injected Sequence CH11, CH12 (Slave Interrupt Priority)



DMA request of dual-ADC mode

In all dual-ADC modes, two DMA channels (one for master ADC, one for slave ADC) can be used to transmit data, consistent with the single-ADC mode.

Figure 128 DMA Request in Regular Simultaneous Mode



In simultaneous regular mode and interleaved mode, a DMA channel can be saved and a single DMA channel can be used to transmit two data. A single DMA request will be generated each time both a master EOC event and a slave EOC event occur. That is: one DMA request will be generated each time 2 data items are available. At that time, the master ADC's DR register contains two half-words, representing the convertion data of both ADCs. The upper half-word is taken from the slave ADC data, while the lower half-word is taken from he master ADC. This mode is used for alternate mode and regular simultaneous mode. One DMA request will be generated each time 2 data items are available.

Figure 129 DMA Request in Regular Simultaneous Mode

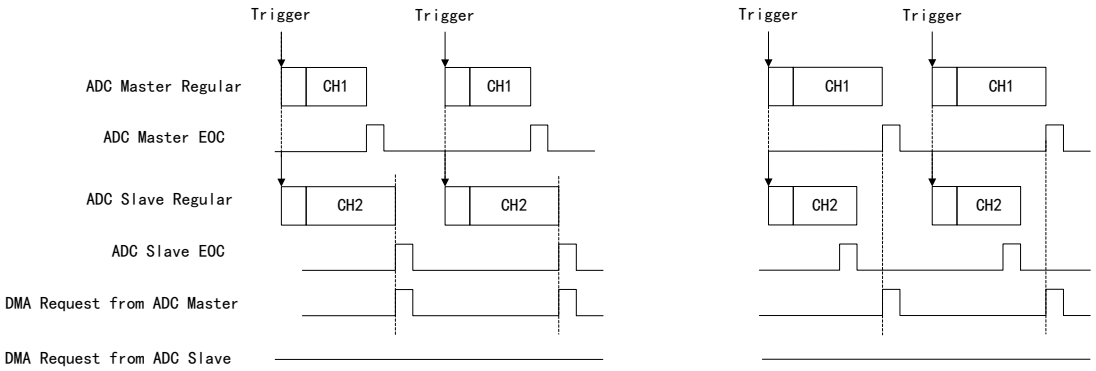
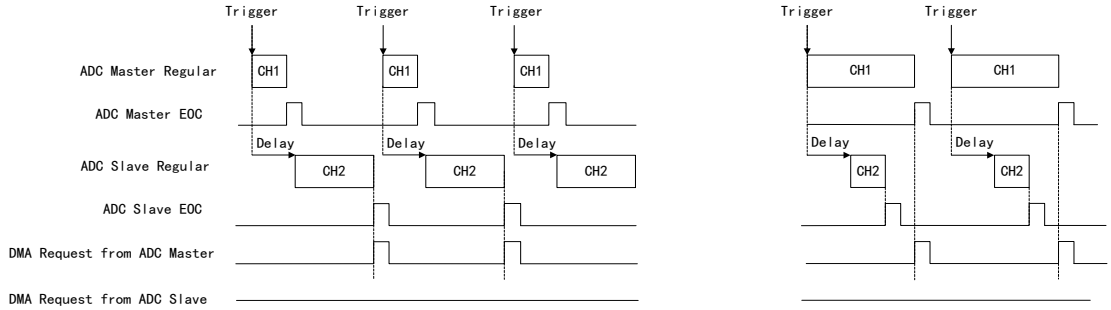


Figure 130 DMA Request in Interleaved Mode



The user shall correctly configure the duration of master and slave conversions so that the DMA request is generated and serviced for reading data (master + slave) before new conversions are available.

Overrun detection

In dual-ADC mode (when the DUALMOD[3:0] in the master ADC's ADC1_CR register is not equal to 0000), if an overrun is detected on one ADC, DMA requests are no longer issued to ensure that all data transmitted to RAM is valid. As the ADC's data register contains valid data, the EOC bit corresponding to one ADC might remain set.

Stop conversions in dual-ADC mode

The user shall set the control bits ADSTP/JADSTP in the master ADC's ADC1_CR register to stop the conversion of both ADC in dual-ADC mode. The control bits in the slave ADC's ADC2_CR register have no effect in dual-ADC mode.

Once both ADC effectively stop, the ADSTART/JADSTART bit in the ADCx_CR registers of both master and slave ADC are both cleared by hardware.

22.4.25 ADC (low-power mode)

Table 87 Impact of low-power Mode on ADC

Mode	Description
Low-power run mode	No impact
Stop mode	ADC stops operation.
Standby mode	ADC powers down, and it can start to work only after the system exits standby mode and becomes stable.

22.4.26 ADC interrupt

Interrupts are generated in the following cases:

- At the end of any conversion of the regular group (EOC flag)
- At the end of the conversion sequence of the regular group (EOS flag)

- At the end of any conversion of the injected group (EOC flag)
- At the end of the conversion sequence of the injected group (JEOS flag)
- When analog watchdog detection occurs (AWD1, AWD2 and AWD3 flags)
- At the end of the sampling phase (EOSMP flag)
- When a data overrun occurs (OVR flag)
- When the injected sequence context queue overflows (JQOVF flag)

Separate interrupt enable bits can be used for flexibility.

Table 88 ADC Interrupt

Interrupt event	Event flag	Enable control bit
End of conversion of regular group	EOC	EOCIE
End of conversion sequence of regular group	EOS	EOSIE
End of conversion of injected group	JEOC	JEOCIE
End of conversion sequence of injected group	JEOS	JEOSIE
Analog watchdog 1 status bit set to 1	AWD1	AWD1IE
Analog watchdog 2 status bit set to 1	AWD2	AWD2IE
Analog watchdog 3 status bit set to 1	AWD3	AWD3IE
End of sampling phase	EOSMP	EOSMPIE
Overrun	OVR	OVRIE
Injected context queue overflow	JQOVF	JQOVFIE

22.4.27 Reference voltage

The 16-bit ADC can be used to select the internal reference voltage or the input voltage of the VREFH pin as the reference voltage.

To use the internal reference voltage as the reference voltage, there is no need to input the voltage on the VREFH pin. Enabling the ADC16BUFEN and LNBGEN bits, and the internal reference voltage can be output on the VREFH pin.

To use the input voltage of VREFH pin as the reference voltage, the ADC16BUFEN and LNBGEN need to be disabled, and the voltage shall be input to the VREFH pin.

Enabling the CHOPPEREN bit can improve the accuracy, but increases the ADC's power-on stabilization time (up to 10 ms). After it becomes stable, the ADC's sampling conversion time will not be affected.

22.5 Register address mapping

Table 89 ADC16 Register Address Mapping

Register name	Description	Offset address
ADC16_ISR	Interrupt status register	0x00
ADC16_IER	Interrupt enable register	0x04
ADC16_CR	Control register	0x08
ADC16_CFGR	Configuration register	0x0C
ADC16_CFGR2	Configuration register 2	0x10
ADC16_SMPR1	Sampling time register 1	0x14
ADC16_TR1	Analog watchdog 1 threshold register	0x20
ADC16_TR2	Analog watchdog 2 threshold register	0x24
ADC16_TR3	Analog watchdog 3threshold register	0x28
ADC16_SQR1	Regular sequence register 1	0x30
ADC16_SQR2	Regular sequence register 2	0x34
ADC16_DR	Data register	0x38
ADC16_JSQR	Injected sequence register	0x3C
ADC16_JDR1	Injected channel data register 1	0x40
ADC16_JDR2	Injected channel data register 2	0x44
ADC16_JDR3	Injected channel data register 3	0x48
ADC16_JDR4	Injected channel data register 4	0x4C
ADC16_AWD2CR	Analog watchdog 2 control register	0x50
ADC16_AWD3CR	Analog watchdog 3 control register	0x54
ADC16_OFFSET0	Offset value register 0	0x58
ADC16_OFFSET1	Offset value regiser 1	0x5C
ADC16_OFFSET2	Offset value register 2	0x60
ADC16_OFFSET3	Offset value register 3	0x64
ADC16_OFFSET4	Offset value register 4	0x68
ADC16_OFFSET5	Offset value register 5	0x6C
ADC16_CAL	Standard register	0x80

22.6 Register functional description

22.6.1 Interrupt status register (ADC16_ISR)

Offset address: 0x00

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:11	Reserved		
10	JQOVF	R/W	Injected context queue overflow This bit will be set to 1 by hardware when an injected context queue overflow occurs. This bit can be cleared by software writing 1 to it. 0: No injected context queue overflow occurs (or the flag event has been acknowledged and cleared to 0 by software) 1: An injected context queue overflow occurs
9	AWD3	R/W	Analog watchdog 3 flag This bit will be set to 1 by hardware when the conversion voltage exceeds the value programmed in the LT3[7:0] and HT3[7:0] fields of the ADCx_TR3 register. This bit can be cleared by software writing 1 to it. 0: No analog watchdog 3 event occurs (or the flag event has been acknowledged and cleared to 0 by software) 1: Analog watchdog 3 event occurs
8	AWD2	R/W	Analog watchdog 2 flag This bit will be set to 1 by hardware when the conversion voltage exceeds the value programmed in the LT2[11:0] and HT2[11:0] fields of the ADCx_TR2 register. This bit can be cleared by software writing 1 to it. 0: No analog watchdog 2 occurs (or the flag event has been acknowledged and cleared to 0 by software) 1: Analog watchdog 2 event occurs
7	AWD1	R/W	Analog watchdog 1 flag This bit will be set to 1 by hardware when the conversion voltage exceeds the value programmed in the LT1[15:0] and HT1[15:0] fields of the ADCx_TR1 register. This bit can be cleared to 0 by writing 1 to it via software. 0: No analog watchdog 1 event occurs (or the flag event has been acknowledged and cleared to 0 by software) 1: Analog watchdog 1 event occurs
6	JEOS	R/W	Injected channel end of sequence flag This bit will be set to 1 by hardware at the end of the conversion for all the injected channels in the group. This bit can be cleared by software writing 1 to it. 0: The injected conversion sequence is not completed (or the flag event has been acknowledged and cleared to 0 by software) 1: The injected conversion is completed
5	JEOC	R/W	Injected channel end of conversion flag This bit will be set to 1 by hardware when each injected conversion of a channel is ended and new data appears in the corresponding ADCX_JDRY register. This bit can be cleared to 0 by writing 1 to it via software or reading the corresponding ADCX_JDRY register. 0: The injected channel conversion is not completed (or the flag event has been acknowledged and cleared to 0 by software) 1: The injected channel conversion is completed
4	OVR	R/W	ADC Overrun This bit will be set to 1 by hardware when an overrun event occurs on a regular channel, which means a new conversion has been completed

Field	Name	R/W	Description
			when the EOC flag is already set to 1. This bit can be cleared by software writing 1 to it. 0: No overrun event occurs (or the flag event has been confirmed and cleared to 0 by software) 1: Overrun occurs
3	EOS	R/W	End of regular sequence flag This bit will be set to 1 by hardware at the end of conversion of the regular channel sequence. This bit can be cleared by software writing 1 to it. 0: The regular conversion sequence is not completed (or the flag event has been acknowledged and cleared to 0 by software) 1: Regular conversion sequence is completed
2	EOC	R/W	End of conversion flag This bit will be set to 1 by hardware when each regular conversion of a channel is ended and new data appears in the ADCx_DR register. This bit can be cleared by software writing 1 to it or reading the ADCx_DR register. 0: The regular channel conversion is not completed (or the flag event has been confirmed and cleared by software) 1: The regular channel conversion is completed
1	EOSMP	R/W	End of sampling flag This bit will be set to 1 by hardware at the end of the sampling phase during the conversion of any channel (regular channels only). 0: The sampling phase is not ended (or the flag event has been confirmed and cleared to 0 by software) 1: The sampling phase is ended
0	Reserved		

22.6.2 Interrupt enable register (ADC16_IER)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:11	Reserved		
10	JQOVFIE	R/W	Injected context queue overflow interrupt enable This bit is set to 1 and cleared to 0 by software to enable/disable the injected context queue overflow interrupt. 0: Disable the injected context queue overflow interrupt 1: Enable the injected context queue overflow interrupt. An interrupt is generated when the JQOVF bit is set to 1. Note: Software is only allowed to write to this bit when JADSTART=0 (this ensures that no injected conversion is currently in progress).
9	AWD3IE	R/W	Analog watchdog 3 interrupt enable This bit is set to 1 and cleared to 0 by software to enable/disable the analog watchdog 3 interrupt. 0: Disable the analog watchdog 3 interrupt 1: Enable the analog watchdog 3 interrupt Note: Software is only allowed to write to this bit when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).
8	AWD2IE	R/W	Analog watchdog 2 interrupt enable

Field	Name	R/W	Description
			This bit is set to 1 and cleared to 0 by software to enable/disable the analog watchdog 2 interrupt. 0: Disable the analog watchdog 2 interrupt 1: Enable the analog watchdog 2 interrupt Note: Software is only allowed to write to this bit when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).
7	AWD1IE	R/W	Analog watchdog 1 interrupt enable This bit is set to 1 and cleared to 0 by software to enable/disable the analog watchdog 1 interrupt. 0: Disable the analog watchdog 1 interrupt 1: Enable the analog watchdog 1 interrupt Note: Software is only allowed to write to this bit when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).
6	JEOSIE	R/W	End of injected sequence of conversions interrupt enable This bit is set to 1 and cleared to 0 by software to enable/disable the injected conversion sequence end interrupt. 0: Disable JEOS interrupt. 1: Enable JEOS interrupt. An interrupt is generated when the JEOS bit is set to 1. Note: Software is only allowed to write to this bit when JADSTART=0 (this ensures that no injected conversion is currently in progress).
5	JEOCIE	R/W	End of injected conversion interrupt enable This bit is set to 1 and cleared to 0 by software to enable/disable the interrupt at the end of injected conversion. 0: Disable JEOC interrupt. 1: Enable JEOC interrupt. An interrupt is generated when the JEOC bit is set to 1. Note: Software is only allowed to write to this bit when JADSTART=0 (this ensures that no regular conversion is currently in progress).
4	OVRIE	R/W	Overrun interrupt enable This bit is set to 1 and cleared to 0 by software to enable/disable the regular conversion overflow interrupt. 0: Disable overflow interrupt 1: Enable overflow interrupt. An interrupt is generated when the OVR bit is set to 1. Note: Software is only allowed to write to this bit when ADSTART=0 (this ensures that no regular conversion is currently in progress).
3	EOSIE	R/W	End of regular sequence of conversions interrupt enable This bit is set to 1 and cleared to 0 by software to enable/disable the regular conversion sequence end interrupt. 0: Disable EOS interrupt 1: Enable EOS interrupt. An interrupt is generated when the EOS bit is set to 1. Note: Software is only allowed to write to this bit when ADSTART=0 (this ensures that no regular conversion is currently in progress).
2	EOCIE	R/W	End of regular conversion interrupt enable This bit is set to 1 and cleared to 0 by software to enable/disable the interrupt at the end of regular conversion.

Field	Name	R/W	Description
			0: Disable EOC interrupt. 1: Enable EOC interrupt. An interrupt is generated when the EOC bit is set to 1. Note: Software is only allowed to write to this bit when ADSTART=0 (this ensures that no regular conversion is currently in progress).
1	EOSMPIE	R/W	End of sampling flag interrupt enable for regular conversions This bit is set to 1 and cleared to 0 by software to enable/disable the interrupt at the end of regular conversion sampling phase. 0: Disable EOSMP interrupt 1: Enable EOSMP interrupt. An interrupt is generated when the EOSMP bit is set to 1. Note: Software is only allowed to write to this bit when ADSTART=0 (this ensures that no regular conversion is currently in progress).
0	Reserved		

22.6.3 Control register (ADC16_CR)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:10	Reserved		
9:6	DUALMOD[3:0]	R/W	Dual mode selection The software uses these bits to select the operating mode. 0000: Independent mode 0001: Mixed synchronous regular + injected synchronous mode 0010: Mixed synchronous regular + alternate trigger mode 0011: Mixed alternate mode + injected synchronous mode 0100: Reserved 0101: Injected synchronous mode 0110: Regular synchronous mode 0111: Only support alternate mode 1000: Reserved 1001: Alternate trigger mode Note: These bits are reserved in ADC2. In dual mode, changing the channel configuration will generate a restart condition, which may cause loss of synchronization. It is recommended to disable dual mode before making any configuration changes.
5	JADSTP	R/W	ADC stop of injected conversion command This bit is set to 1 by software to stop and discard an ongoing injected conversion (JADSTP command). The bit is cleared to 0 by hardware when the conversion has been effectively discarded and the ADC injected sequence and trigger can be reconfigured. Subsequently, the ADC will be ready to receive a new command to start injected conversion (JADSTART command). 0: The command that ADC stops injected conversion currently is not being executed. 1: Writing 1 can sop the ongoing injected conversion. A read value of 1 indicates that the ADSTP command is being executed.

Field	Name	R/W	Description
			Note: Software is only allowed to set JADSTP to 1 when JADSTART=1 and ADDIS=0 (the ADC is enabled, will eventually perform injected conversion, and there are no pending requests to disable the ADC). In automatic injection mode (JAUTO=1), setting the ADSTP bit to 1 will abort both regular and injected conversions (do not use JADSTP).
4	ADSTP	R/W	ADC stop of regular conversion command This bit is set to 1 by software to stop and discard an ongoing regular conversion (ADSTP command). The bit is cleared by hardware when the conversion has been effectively discarded and the ADC regular sequence and trigger can be reconfigured. Subsequently, the ADC will be ready to receive a new command to start regular conversion (ADSTART command). 0: No ADC stop regular conversion command is currently being executed. 1: Writing 1 stops the ongoing regular conversion. A read value of 1 indicates that the ADSTP command is being executed. Note: Software is only allowed to set ADSTP to 1 when ADSTART=1 and ADDIS=0 (the ADC is enabled, will eventually perform regular conversion, and there are no pending requests to disable the ADC). In automatic injection mode (JAUTO=1), setting the ADSTP bit to 1 will abort both regular and injected conversions (do not use JADSTP).
3	JADSTART	R/W	ADC start of injected conversion This bit is set to 1 by software to start the injected channel conversion of the ADC. Depending on the value of the EXTEN configuration bit, conversion can start immediately (software trigger configuration) or after an injected hardware trigger event occurs (hardware trigger configuration). This bit is cleared to 0 by hardware: In single-conversion mode, if software trigger is selected (JEXTSEL=0x0): It is cleared to 0 when the flag for the end of injected conversion sequence (JEOS) appears. In all cases: It is cleared to 0 by hardware when the JADSTP bit is cleared to 0 after the execution of the JADSTP command. 0: No ADC injected conversion is currently in progress. 1: Writing 1 can start injected conversion. A read value of 1 indicates that the ADC is running and will eventually convert injected channels. Note: Software is only allowed to set JADSTART to 1 when ADEN=1 and ADDIS=0 (the ADC is enabled, and there are no pending requests to disable the ADC). In automatic injected mode (JAUTO=1), start regular conversion and automatic injected conversion by setting the ADSTART bit to 1 (JADSTART must remain cleared to 0).
2	ADSTART	R/W	ADC start of regular conversion This bit is set to 1 by software to start the regular channel conversion of the ADC. Depending on the value of the EXTEN configuration bit, conversion can start immediately (software trigger

Field	Name	R/W	Description
			configuration) or after a regular hardware trigger event occurs (hardware trigger configuration). This bit is cleared to 0 by hardware: In single-conversion mode, if software trigger is selected (EXTSEL=0x0): It is cleared to 0 when the flag for the end of regular conversion sequence (EOS) appears. In all cases: It is cleared to 0 by hardware when the ADSTP bit is cleared to 0 after the execution of the ADSTP command. 0: No ADC regular conversion is currently in progress. 1: Writing 1 can start regular conversion. A read value of 1 indicates that the ADC is running and will eventually convert regular channels. Note: Software is only allowed to set ADSTART to 1 when ADEN=1 and ADDIS=0 (the ADC is enabled, and there are no pending requests to disable the ADC). In automatic injection mode (JAUTO=1), setting the ADSTART bit to 1 starts regular conversion and automatic injection conversion (JADSTART must remain cleared to 0).
1	ADDIS	R/W	ADC disable command This bit is set to 1 by software to disable the ADC (ADDIS command) and put it in power-down state (OFF state). It is cleared to 0 by hardware immediately after the ADC is effectively disabled (at this time, ADEN is also cleared to 0 by hardware). 0: No ADDIS command is currently being executed. 1: Writing 1 can disable the ADC. A read value of 1 indicates that the ADDIS command is being executed. Note: Software is only allowed to set ADDIS to 1 when ADEN=1, ADSTART=0, and JADSTART=0 (this ensures no conversion is currently in progress).
0	ADEN	R/W	ADC enable control This bit is set to 1 and cleared to 0 by software to enable the ADC. 0: Disable ADC (OFF state) 1: Write 1 to enable the ADC.

22.6.4 Configuration register (ADC16_CFGR)

Offset address: 0x0C

Reset value: 0x8000 0000

Field	Name	R/W	Description
31	JQDIS	R/W	Injected queue disable This bit is set to 1 and cleared to 0 by software to disable the injected queue mechanism. 0: Enable injected queue 1: Disable injected queue Note: Software is only allowed to write to this bit when ADSTART=0 and JADSTART=0 (this ensures no regular conversion or injected conversion is currently in progress). Setting the JQDIS bit to 1 or resetting it will cause the injected queue to be emptied, and the JSQR register will also be emptied.

Field	Name	R/W	Description
			Note: The application testing shows that JQDIS takes effect when it changes from 1 to 0 or from 0 to 1.
30:29	Reserved		
28:26	AWD1CH[2:0]	R/W	Analog Watchdog 1 Channel Selection These bits are set to 1 and cleared to 0 by software. They are used to select the input channel monitored by the analog watchdog. 000: Monitor ADC analog input channel single-ended mode INP<0> via AWD1 001: Monitor ADC analog input channel single-ended mode INP<1> via AWD1 010: Monitor ADC analog input channel single-ended mode INP<2> via AWD1 011: Monitor ADC analog input channel single-ended mode INN<0> via AWD1 100: Monitor ADC analog input channel single-ended mode INN<1> via AWD1 101: Monitor ADC analog input channel single-ended mode INN<2> via AWD1 Other: Reserved, cannot be used Note: The channel selected via AWD1CH must also be selected in the SQRI or JSQRI register. Software is only allowed to write to these bits when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).
25	JAUTO	R/W	Automatic injected group conversion This bit is set to 1 and cleared to 0 by software to enable/disable automatic injected group conversion after regular group conversion. 0: Disable automatic injected group conversion 1: Enable automatic injected group conversion Note: Software is only allowed to write to this bit when ADSTART=0 and JADSTART=0 (this ensures no regular conversion or injected conversion is currently in progress).
24	JAWD1EN	R/W	Analog watchdog 1 enable on injected channels This bit is set to 1 and cleared to 0 by software 0: Disable analog watchdog 1 on injected channel 1: Enable analog watchdog 1 on injected channel Note: Software is only allowed to write to this bit when JADSTART=0 (this ensures that no injected conversion is currently in progress).
23	AWD1EN	R/W	Analog watchdog 1 enable on regular channels This bit is set to 1 and cleared to 0 by software 0: Disable analog watchdog 1 on regular channel 1: Enable analog watchdog 1 on regular channel Note: Software is only allowed to write to this bit when ADSTART=0 (this ensures that no regular conversion is currently in progress).
22	AWD1SGL	R/W	Enable the watchdog 1 on a single channel or on all channels This bit is set to 1 and cleared to 0 by software to enable the analog watchdog on the channel determined by the AWD1CH[2:0] bits or on all channels.

Field	Name	R/W	Description
			0: Enable analog watchdog 1 on all channels 1: Enable analog watchdog 1 on a single channel Note: Software is only allowed to write to these bits when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).
21	JQM	R/W	JSQR queue mode This bit is set to 1 and cleared to 0 by software. This bit defines management mode of empty queues. 0: JSQR mode 0: The queue is never empty and retains the last configuration written to JSQR. 1: JSQR mode 1: The queue can be empty. When the queue is empty, software and hardware triggers for the injected sequence are internally disabled immediately after the previous valid injected sequence is completed. Note: Software is only allowed to write to this bit when JADSTART=0 (this ensures that no injected conversion is currently in progress).
20	JDISCEN	R/W	Discontinuous mode on injected channels This bit is set to 1 and cleared to 0 by software to enable/disable the discontinuous sampling mode for injected channels in the group. 0: Disable discontinuous sampling mode for injected channels 1: Enable discontinuous sampling mode for injected channels Note: Software is only allowed to write to this bit when JADSTART=0 (this ensures that no injected conversion is currently in progress). Automatic injection mode and discontinuous mode cannot be used simultaneously: When JAUTO=1, the DISCEN and JDISCEN bits must be kept cleared by software.
19:17	DISCNUM[2:0]	R/W	Discontinuous mode channel count Software writes to these bits to define the number of regular channels converted in discontinuous mode after receiving an external trigger. 000: 1 channel 001: 2 channels ... 101: 6 channels Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).
16	DISCEN	R/W	Discontinuous mode for regular channels This bit is set to 1 and cleared to 0 by software to enable/disable the discontinuous mode of regular channels. 0: Disable discontinuous mode for regular channels 1: Enable discontinuous mode for regular channels Note: Discontinuous mode and continuous mode cannot be enabled simultaneously: DISCEN and CONT must not be set to 1 at the same time.

Field	Name	R/W	Description
			Automatic injection mode and discontinuous mode cannot be used simultaneously: When JAUTO is set to 1, the DISCEN and JDISCEN bits must be kept cleared to 0 by software. Software is only allowed to write to this bit when ADSTART=0 (to ensure no regular conversion is currently in progress).
15	Reserved		
14	AUTDLY	R/W	Delayed conversion mode This bit is set to 1 and cleared to 0 by software to enable/disable the automatic delayed conversion mode. 0: Disable automatic delayed conversion mode 1: Enable automatic delayed conversion mode Note: Software is only allowed to write to this bit when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).
13	CONT	R/W	Single/continuous conversion mode for regular conversions This bit is set to 1 and cleared to 0 by software. When this bit is set to 1, regular conversions will continue until the bit is cleared. 0: Single conversion mode 1: Continuous conversion mode Note: Discontinuous mode and continuous mode cannot be enabled simultaneously: DISCEN and CONT must not be set to 1 at the same time. Software is only allowed to write to this bit when ADSTART=0 (to ensure no regular conversion is currently in progress).
12	OVRMOD	R/W	Overrun mode This bit is set to 1 and cleared to 0 by software to configure the management mode for data overrun. 0: If an overrun is detected, the ADCx_DR register retains the original data. 1: If an overrun is detected, the ADCx_DR register is overwritten by the previous conversion result. Note: Software is only allowed to write to this bit when ADSTART=0 (this ensures that no regular conversion is currently in progress).
11:10	EXTEN[1:0]	R/W	External trigger enable and polarity selection for regular channels These bits are set to 1 and cleared to 0 by software to select external trigger polarity and enable the trigger of regular group. 00: Disable hardware trigger detection (conversion can be started by software) 01: Execute hardware trigger detection on rising edge 10: Execute hardware trigger detection on falling edge 11: Execute hardware trigger detection on both rising and falling edges Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).
9:6	EXTSEL[3:0]	R/W	External trigger selection for regular group These bits can be used to select the external event used to trigger the regular group conversion. 0000: Event 0 0001: Event 1

Field	Name	R/W	Description
			0010: Event 2 0011: Event 3 0100: Event 4 0101: Event 5 0110: Event 6 0111: Event 7 ... 1111: Event 15 Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).
5:2	Reserved		
1	DMACFG	R/W	Direct memory access configuration This bit is set to 1 and cleared to 0 by software to select between two DMA operating modes. It is only valid when DMAEN=1. 0: Select DMA single mode 1: Select DMA circular mode Note: Software is only allowed to write to this bit when ADSTART=0 (this ensures that no regular conversion is currently in progress).
0	DMAEN	R/W	Direct memory access enable This bit is set to 1 and cleared to 0 by software to enable the generation of DMA requests. 0: Enable DMA 1: Enable DMA Note: Software is only allowed to write to this bit when ADSTART=0 (this ensures that no regular conversion is currently in progress).

22.6.5 Configuration register 2 (ADC16_CFGR2)

Offset address: 0x10

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:15	Reserved		
14	DIFSEL	R/W	ADC single-ended/differential mode control signal 0 (Default): Single-ended mode 1: Differential mode Note: Software is only allowed to write to these bits when JADSTART=0 and ADSTART=0.
13:11	AWDFILT[2:0]	R/W	Analog watchdog filtering parameter 000: No filtering 001: Every two consecutive detections generate an AWD1 flag or interrupt … 111: Every eight consecutive detections generate an AWD1 flag or interrupt Note: Software is only allowed to write to this bit when ADSTART=0 and JADSTART=0 (this ensures that no conversion is currently in progress).
10	ROVSM	R/W	Regular oversampling mode

Field	Name	R/W	Description
			This bit is set to 1 and cleared to 0 by software to select regular oversampling mode. 0: Continuous mode: If an injected conversion is triggered, oversampling will be temporarily paused, and will continue after the injected sequence is completed (the oversampling buffer is reserved during the injected sequence process). 1: Resume mode: If an injected conversion is triggered, the current oversampling will be aborted, and restart the oversampling after the injected sequence is completed (the oversampling buffer will be cleared to 0 when the injected sequence starts). Note: Software is only allowed to write to this bit when ADSTART=0 (this ensures that no conversion is currently in progress). Note: Both ROVSE and JOVSE bits can be set to 1. In this case, the regular oversampling mode will be forced to enter the recovery mode (the ROVSM bit is ignored); the injected conversions are supported during triggered regular mode. In this case, the injected oversampling mode must be disabled, and the ROVSM bit will be ignored (forced to enter the recovery mode).
9	TROVS	R/W	Triggered regular oversampling This bit is set to 1 and cleared to 0 by software to enable the triggered oversampling. 0: All oversampling conversions for a certain channel will be continuously completed after trigger. 1: Each oversampling conversion on a certain channel requires a new trigger Note: Software is only allowed to write to this bit when ADSTART=0 (this ensures that no conversion is currently in progress).
8:6	OVSS[2:0]	R/W	Oversampling shift This bit is set to 1 and cleared to 0 by software to define the number of right shifts applied to the raw oversampling result. 000: No shift occurs 001: Shift 1 bit 010: Shift 2 bits 011: Shift 3 bits 100: Shift 4 bits Others: Reserved Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no conversion is currently in progress).
5:4	Reserved		
3:2	OVSR[1:0]	R/W	Oversampling ratio This bit is set to 1 and cleared to 0 by software to define the oversampling ratio. 00: 2 times 01: 4 times 10: 8 times 11: 16 times Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no conversion is currently in progress).
1	JOVSE	R/W	Injected oversampling enable

Field	Name	R/W	Description
			This bit is set to 1 and cleared to 0 by software to enable the injected oversampling. 0: Disable injected oversampling 1: Enable injected oversampling Note: Software is only allowed to write to this bit when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).
0	ROVSE	R/W	Regular oversampling enable This bit is set to 1 and cleared to 0 by software to enable the regular oversampling. 0: Disable regular oversampling 1: Enable regular oversampling Note: Software is only allowed to write to this bit when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).

22.6.6 Sampling time register 1 (ADC16_SMPR1)

Offset address: 0x14

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:24	Reserved		
23:20	SMP5[3:0]	R/W	Channel 5 (single-ended mode INN<2>) sampling time selection
19:16	SMP4[3:0]	R/W	Channel 4 (single-ended mode INN<1>) sampling time selection
15:12	SMP3[3:0]	R/W	Channel 3 (single-ended mode INN<0>) sampling time selection
11:8	SMP2[3:0]	R/W	Channel 2 (single-ended mode INP<2>/differential mode<2>) sampling time selection
7:4	SMP1[3:0]	R/W	Channel 1 (single-ended mode INP<1>/differential mode<1>) sampling time selection
3:0	SMP0[3:0]	R/W	Channel 0 (single-ended mode INP<0>/differential mode<0>) sampling time selection These bits are written by software to select the sampling time for each channel separately. During the sampling period, the channel selection bits must remain unchanged. 0000: 1 ADC clock cycle 0001: 2 ADC clock cycles 0010: 3 ADC clock cycles ... 1111: 16 ADC clock cycles Note: Software is only allowed to write to these bits when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).

22.6.7 Analog watchdog 1 threshold register (ADC16_TR1)

Offset address: 0x20

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	HT1[15:0]	R/W	Analog watchdog 1 higher threshold

Field	Name	R/W	Description
			These bits are written by software to define the higher threshold for analog watchdog 1. Note: Software is only allowed to write to these bits when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).
15:0	LT1[15:0]	R/W	Analog watchdog 1 lower threshold These bits are written by software to define the lower threshold for analog watchdog 1. Note: Software is only allowed to write to these bits when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).

22.6.8 Analog watchdog 2 threshold register (ADC16_TR2)

Offset address: 0x24

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:28	Reserved		
27:16	HT2[11:0]	R/W	Analog watchdog 2 higher threshold These bits are written by software to define the higher threshold for analog watchdog 1. Note: Software is only allowed to write to these bits when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).
15:12	Reserved		
11:0	LT2[11:0]	R/W	Analog watchdog 2 lower threshold These bits are written by software to define the lower threshold for analog watchdog 1. Note: Software is only allowed to write to these bits when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).

22.6.9 Analog watchdog 3 threshold register (ADC16_TR3)

Offset address: 0x28

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:24	Reserved		
23:16	HT3[7:0]	R/W	Analog watchdog 3 higher threshold These bits are written by software to define the higher threshold for analog watchdog 1. Note: Software is only allowed to write to these bits when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).
15:8	Reserved		
7:0	LT3[7:0]	R/W	Analog watchdog 3 lower threshold These bits are written by software to define the lower threshold for analog watchdog 1. Note: Software is only allowed to write to these bits when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).

22.6.10 Regular sequence register 1 (ADC16_SQR1)

Offset address: 0x30

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:28	SQ7[3:0]	R/W	7th conversion in regular sequence These bits are written by software to assign the channel number as the 7th conversion in regular conversion sequence. Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).
27:24	SQ6[3:0]	R/W	6th conversion in regular sequence These bits are written by software to assign the channel number as the 6th conversion in regular conversion sequence. Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).
23:20	SQ5[3:0]	R/W	5th conversion in regular sequence These bits are written by software to assign the channel number as the 5th conversion in regular conversion sequence. Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).
19:16	SQ4[3:0]	R/W	4th conversion in regular sequence These bits are written by software to assign the channel number as the 4th conversion in regular conversion sequence. Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).
15:12	SQ3[3:0]	R/W	3rd conversion in regular sequence These bits are written by software to assign the channel number as the 3rd conversion in regular conversion sequence. Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).
11:8	SQ2[3:0]	R/W	2nd conversion in regular sequence These bits are written by software to assign the channel number as the 2nd conversion in regular conversion sequence. Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).
7:4	SQ1[3:0]	R/W	1st conversion in regular sequence These bits are written by software to assign the channel number SQx as the 1st conversion in regular conversion sequence. SQx<3:0>=0000 (default) SQx<1:0>=00, INP<2:0> not selected SQx<1:0>=01, select input signal INP<0> SQx<1:0>=10, select input signal INP<1> SQx<1:0>=11, select input signal INP<2> SQx<3:2>=00, INN<2:0> not selected SQx<3:2>=01, select input signal INN<0> SQx<3:2>=10, select input signal INN<1> SQx<3:2>=11, select input signal INN<2> When configuring single--ended mode: If INP is selected, then SQx<3:0>=00XX If INN is selected, then SQx<3:0>=XX00 Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).
3:0	RL[3:0]	R/W	Regular channel sequence length

Field	Name	R/W	Description
			These bits are written by software to define the total number of conversions in the regular channel conversion sequence. 0000: 1 conversion 0001: 2 conversions ... 1011: 12 conversions Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).

22.6.11 Regular sequence register 2 (ADC16_SQR2)

Offset address: 0x34

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:20	Reserved		
19:16	SQ12[3:0]	R/W	12th conversion in regular sequence These bits are written by software to assign the channel number as the 12th conversion in regular conversion sequence. Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).
15:12	SQ11[3:0]	R/W	11th conversion in regular sequence These bits are written by software to assign the channel number as the 11th conversion in regular conversion sequence. Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).
11:8	SQ10[3:0]	R/W	10th conversion in regular sequence These bits are written by software to assign the channel number as the 10th conversion in regular conversion sequence. Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).
7:4	SQ9[3:0]	R/W	9th conversion in regular sequence These bits are written by software to assign the channel number as the 9th conversion in regular conversion sequence. Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).
3:0	SQ8[3:0]	R/W	8th conversion in regular sequence These bits are written by software to assign the channel number as the 8th conversion in regular conversion sequence. Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).

22.6.12 Data register (ADC16_DR)

Offset address: 0x38

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	ADC2RDATA[15:0]	R	ADC2 data In dual mode, these bits contain the regular channel data converted by ADC2.
15:0	RDATA[15:0]	R	Regular data converted

Field	Name	R/W	Description
			These bits are read-only. They contain the conversion result of the last converted regular channel.

22.6.13 Injected sequence register (ADC16_JSQR)

Offset address: 0x3C

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:24	Reserved		
23:20	JSQ4[3:0]	R/W	4th conversion in the injected sequence These bits are written by software to assign the channel number as the 4th conversion in injected conversion sequence. Note: After ADC is enabled (ADEN1), it is allowed to write these bits at any time by software.
19:16	JSQ3[3:0]	R/W	3rd conversion in the injected sequence These bits are written by software to assign the channel number as the 3rd conversion in injected conversion sequence. Note: After ADC is enabled (ADEN1), it is allowed to write these bits at any time by software.
15:12	JSQ2[3:0]	R/W	2nd conversion in the injected sequence These bits are written by software to assign the channel number as the 2nd conversion in injected conversion sequence. Note: After ADC is enabled (ADEN1), it is allowed to write these bits at any time by software.
11:8	JSQ1[3:0]	R/W	1st conversion in the injected sequence These bits are written by software to assign the channel number JSQX as the 1st conversion in injected conversion sequence. JSQx<3:0>=0000 (default) JSQX<1:0>=00: none of the INP<2:0> inputs selected JSQX<1:0>=01: select input signal INP<0> JSQX<1:0>=10: select input signal INP<1> JSQX<1:0>=11: select input signal INP<2> JSQX<3:2>=00: Inone of the INP<2:0> inputs selected JSQX<3:2>=01: select input signal INN<0> JSQX<3:2>=10: select input signal INN<1> JSQX<3:2>=11: select input signal INN<2> When configuring single-ended mode, if INP is selected, then JSQX<3:0>=00XX If INN is selected, then JSQX<3:0>=XX00 Note: After ADC is enabled (ADEN1), it is allowed to write these bits at any time by software.
7:6	JEXTEN[1:0]	R/W	External trigger enable and polarity selection for injected channels These bits are set to 1 and cleared to 0 by software to select external trigger polarity and enable the trigger of injected group. 00: If JQDIS=0 (queue enabled), both hardware and software trigger detections are disabled 00: If JQDIS=1 (queue disabled), hardware trigger detection is disabled (conversion can be started by software) 01: Execute hardware trigger detection on rising edge

Field	Name	R/W	Description
			10: Execute hardware trigger detection on falling edge 11: Execute hardware trigger detection on both rising and falling edges Note: After ADC is enabled (ADEN1), it is allowed to write these bits at any time by software. If JQM=1 and the context queue is empty, software and hardware trigger of injected sequence will be disabled internally.
5:2	JEXTSEL[3:0]	R/W	External trigger selection for injected group These bits can be used to select the external event used to trigger the injected group conversion: 0000: Event 0 0001: Event 1 0010: Event 2 0011: Event 3 0100: Event 4 0101: Event 5 0110: Event 6 0111: Event 7 ... 1111: Event 15 Note: After ADC is enabled (ADEN1), it is allowed to write these bits at any time by software.
1:0	JL[1:0]	R/W	Injected channel sequence length These bits are written by software to define the total number of conversions in the injected channel conversion sequence. 00: 1 conversion 01: 2 conversions 10: 3 conversions 11: 4 conversions Note: After ADC is enabled (ADEN1), it is allowed to write these bits at any time by software.

22.6.14 Injected channel data register 1 (ADC16_JDR1)

Offset address: 0x40

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15:0	JDATA1[15:0]	R	Injected data These bits are read-only. They include the conversion results from the injected channel Y.

22.6.15 Injected channel data register 2 (ADC16_JDR2)

Offset address: 0x44

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15:0	JDATA2[15:0]	R	Injected data

Field	Name	R/W	Description
			These bits are read-only. They include the conversion results from the injected channel Y.

22.6.16 Injected channel data register 3 (ADC16_JDR3)

Offset address: 0x48

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15:0	JDATA3[15:0]	R	Injected data These bits are read-only. They include the conversion results from the injected channel Y.

22.6.17 Injected channel data register 4 (ADC16_JDR4)

Offset address: 0x4c

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15:0	JDATA4[15:0]	R	Injected data These bits are read-only. They include the conversion results from the injected channel Y.

22.6.18 Analog watchdog 2 control register (ADC16_AWD2CR)

Offset address: 0x50

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:6	Reserved		
5:0	AWD2CH[5:0]	R/W	Analog watchdog 2 channel selection These bits are set to 1 and cleared to 0 by software. They are used to enable and select the input channel monitored by the analog watchdog 2. AWD2CH[I] = 0: Not monitor the ADC analog input channel 1 via AWD2 AWD2CH[I] = 1: Monitor the ADC analog input channel 1 via AWD2 In single-ended mode, when AWD2CH[5:0] = 000000, in differential mode, when AWD2CH[2:0] = 000, the analog watchdog 2 will be disabled. I=0: Single-ended mode INP<0> I=1: Single-ended mode INP<1> I=2: Single-ended mode INP<2> I=3: Single-ended mode INN<0> I=4: Single-ended mode INN<1> I=5: Single-ended mode INN<2> Note: The channel selected via AWD2CH must also be selected in the SQRI or JSQRI register. Software is only allowed to write to these bits when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).

22.6.19 Analog watchdog 3 control register (ADC16_AWD3CR)

Offset address: 0x54

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:6	Reserved		
5:0	AWD3CH[5:0]	R/W	Analog watchdog 3 channel selection These bits are set to 1 and cleared to 0 by software. They are used to enable and select the input channel monitored by the analog watchdog 3. AWD3CH[I] = 0: Not monitor the ADC analog input channel 1 via AWD3 AWD3CH[I] = 1: Monitor the ADC analog input channel 1 via AWD3 In single-ended mode, when AWD3CH[5:0] = 000000, in differential mode, when AWD3CH[2:0] = 000, the analog watchdog 3 will be disabled. I=0: Single-ended mode INP<0> I=1: Single-ended mode INP<1> I=2: Single-ended mode INP<2> I=3: Single-ended mode INN<0> I=4: Single-ended mode INN<1> I=5: Single-ended mode INN<2> Note: The channel selected via AWD3CH must also be selected in the SQRI or JSQRI register. Software is only allowed to write to these bits when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).

22.6.20 Offset value register 0 (ADC16_OFFSET0)

Offset address: 0x58

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:18	Reserved		
17	OFFSET0_POS	R/W	Positive offset 0: Negative offset 1: Positive offset Note: Software is allowed to write these bits only when ADSTART = 0 and JADSTART = 0 (ensuring no conversion is in progress).
16	OFFSET0_EN	R/W	Offset enable This bit is written by software to enable and disable the offset programmed in the OFFSET0 bit. Note: Software is only allowed to write to this bit when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).
15:0	OFFSET0[15:0]	R/W	Data offset compensation corresponding to the channel (single-ended mode INP<0>/differential mode<0>) These bits are written by software to define the offset subtracted from the raw conversion data when converting a channel (which can be a regular or injected channel).

Field	Name	R/W	Description
			Note: Software is only allowed to write to these bits when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).

22.6.21 Offset value register 1 (ADC16_OFFSET1)

Offset address: 0x5C

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:18	Reserved		
17	OFFSET1_POS	R/W	Positive offset 0: Negative offset 1: Positive offset Note: Software is allowed to write these bits only when ADSTART = 0 and JADSTART = 0 (ensuring no conversion is in progress).
16	OFFSET1_EN	R/W	Offset enable This bit is written by software to enable and disable the offset programmed in the OFFSET1 bit. Note: Software is only allowed to write to this bit when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).
15:0	OFFSET1[15:0]	R/W	Data offset compensation corresponding to the channel (single-ended mode INP<1>/differential mode<1>) These bits are written by software to define the offset subtracted from the raw conversion data when converting a channel (which can be a regular or injected channel). Note: Software is only allowed to write to these bits when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).

22.6.22 Offset value register 2 (ADC16_OFFSET2)

Offset address: 0x60

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:18	Reserved		
17	OFFSET2_POS	R/W	Positive offset 0: Negative offset 1: Positive offset Note: Software is allowed to write these bits only when ADSTART = 0 and JADSTART = 0 (ensuring no conversion is in progress).
16	OFFSET2_EN	R/W	Offset enable This bit is written by software to enable and disable the offset programmed in the OFFSET2 bit. Note: Software is only allowed to write to this bit when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).
15:0	OFFSET2[15:0]	R/W	Data offset compensation corresponding to the channel (single-ended mode INP<2>/differential mode<2>)

Field	Name	R/W	Description
			These bits are written by software to define the offset subtracted from the raw conversion data when converting a channel (which can be a regular or injected channel). Note: Software is only allowed to write to these bits when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).

22.6.23 Offset value register 3 (ADC16_OFFSET3)

Offset address: 0x64

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:18	Reserved		
17	OFFSET3_POS	R/W	Positive offset 0: Negative offset 1: Positive offset Note: Software is allowed to write these bits only when ADSTART = 0 and JADSTART = 0 (ensuring no conversion is in progress).
16	OFFSET3_EN	R/W	Offset enable This bit is written by software to enable and disable the offset programmed in the OFFSET3 bit. Note: Software is only allowed to write to this bit when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).
15:0	OFFSET3[15:0]	R/W	Data offset compensation corresponding to the channel (single-ended mode INP<0>) These bits are written by software to define the offset subtracted from the raw conversion data when converting a channel (which can be a regular or injected channel). Note: Software is only allowed to write to these bits when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).

22.6.24 Offset value register 4 (ADC16_OFFSET4)

Offset address: 0x68

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:18	Reserved		
17	OFFSET4_POS	R/W	Positive offset 0: Negative offset 1: Positive offset Note: Software is allowed to write these bits only when ADSTART = 0 and JADSTART = 0 (ensuring no conversion is in progress).
16	OFFSET4_EN	R/W	Offset enable This bit is written by software to enable and disable the offset programmed in the OFFSET4 bit. Note: Software is only allowed to write to this bit when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).

Field	Name	R/W	Description
15:0	OFFSET4[15:0]	R/W	Data offset compensation corresponding to the channel (single-ended mode INP<1>) These bits are written by software to define the offset subtracted from the raw conversion data when converting a channel (which can be a regular or injected channel). Note: Software is only allowed to write to these bits when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).

22.6.25 Offset value register 5 (ADC16_OFFSET5)

Offset address: 0x6C

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:18	Reserved		
17	OFFSET5_POS	R/W	Positive offset 0: Negative offset 1: Positive offset Note: Software is allowed to write these bits only when ADSTART = 0 and JADSTART = 0 (ensuring no conversion is in progress).
16	OFFSET5_EN	R/W	Offset enable This bit is written by software to enable and disable the offset programmed in the OFFSET5 bit. Note: Software is only allowed to write to this bit when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).
15:0	OFFSET5[15:0]	R/W	Data offset compensation corresponding to the channel (single-ended mode INP<2>) These bits are written by software to define the offset subtracted from the raw conversion data when converting a channel (which can be a regular or injected channel). Note: Software is only allowed to write to these bits when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).

22.6.26 Standard register (ADC16_CAL)

Offset address: 0x80

Reset value: 0x0000 0002

Field	Name	R/W	Description
31:4	Reserved		
3	REF_SEL	R/W	ADC reference voltage select 0 (default): VREFHI=AVDD (external) 1: VREFHI=1.65 (internal)
2:0	Reserved		

23 12-bit Analog-to-Digital Converter (12-bit ADC)

23.1 Main characteristics

Built-in one 12-bit ADC module, with the following features:

(1) Resolution: 12 bits

(2) Number of sampling channels: Up to 16

(3) Sampling intervals can be programmed per channel

(4) Data alignment with built-in data consistency

(5) Supports offset compensation

(6) Maximum conversion rate of 1 Msps at 12-bit resolution

(7) Trigger mode:
- On-chip timer signal trigger
- External pin signal trigger
- Software trigger

(8) Supports regular sequence, injected sequence, single, and continuous sampling modes

(9) Supports scan mode for single or continuous/discontinuous sequences; each ADC can convert multiple channels or scan a sequence of channels

(10) Significant digits: ADC ENOB = 10 bits

(11) DMA request supporting regular data conversion
- DMA request will be generated after the conversion of regular channels is completed; the converted data result can be transmitted to the memory from the ADC_DATA register

(12) The analog watchdog function can monitor the conversion voltage of multiple channels, and generate an interrupt when the monitored signal exceeds the preset value

(13) One internal LTCBG reference voltage input channel, connected to ADC_CH15.

(14) One analog loop signal AMUXBUS input channel, connected to ADC_CH14.

23.2 Functional description

The figure below shows the internal block diagram of the ADC module, which

illustrates the interconnections of various functional modules.

Figure 131 Internal Block Diagram of 12-bit ADC Module

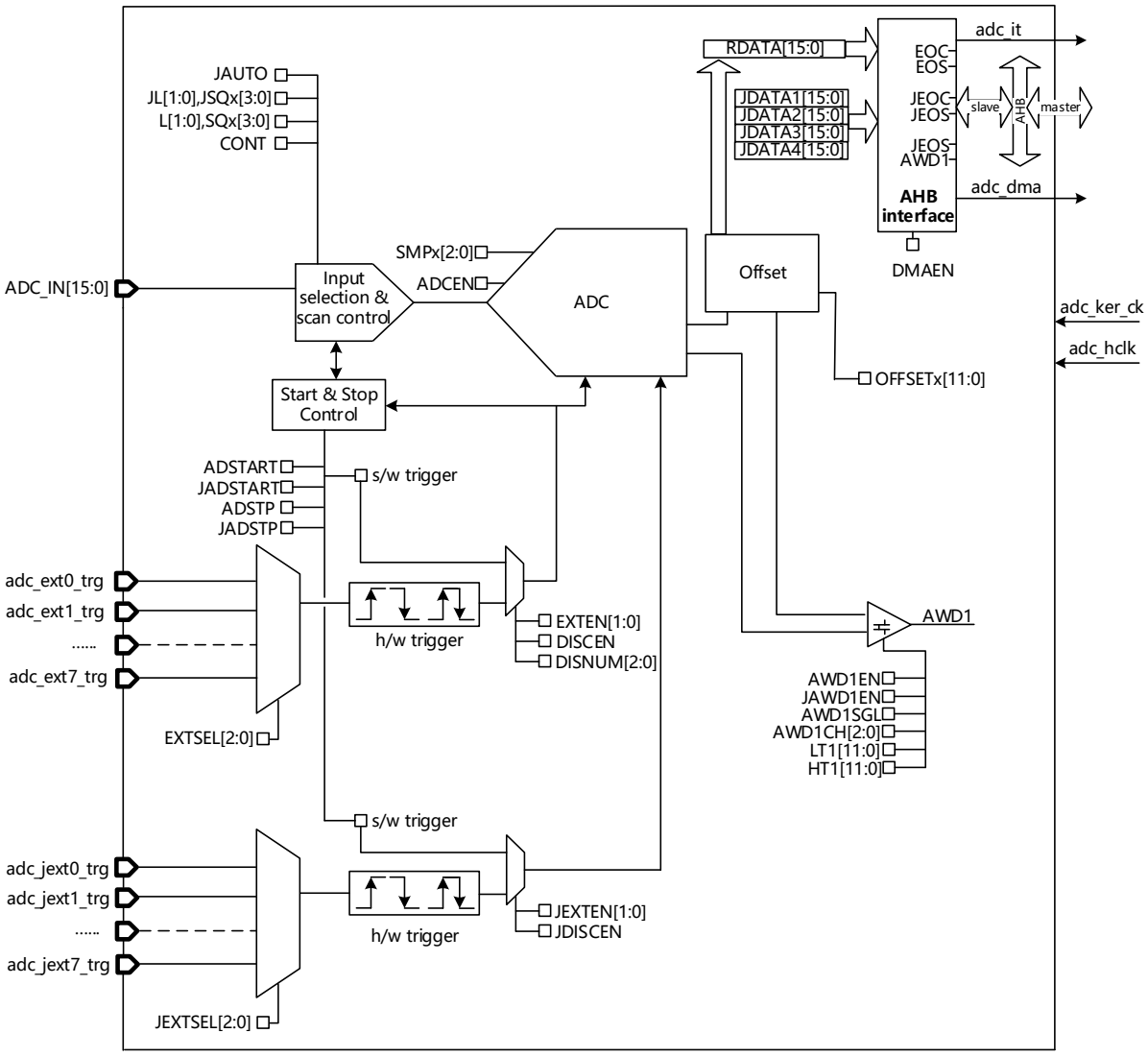


Table 90 ADC Input/Output Pins

Pin name	Signal type	Description
V_{REF+}	Input, analog reference positive electrode	ADC high/positive reference voltage, 2.7V≤VREF+≤3.6
VDDA	Analog power input	Analog power supply equal to VDDA: 2.7V≤VDDA≤3.6V
V_{REF-}	Input, negative analog reference	VREF-corresponding ground level
VSSA	Analog power ground input	Analog power ground pin
VIN[15:0]	Positive input analog channel	Connected to external channels: ADC_INi or internal channel

23.2.1 Clock

The dual-clock domain architecture means that the ADC clock is independent of the AHB bus clock.

32-bit register read and write access is implemented via the AHB bus; ADC sampling, conversion process, and data processing use an independent ADC clock (10MHz and 20MHz frequency clocks can be configured based on the system clock).

23.2.2 Slave device AHB interface

The ADC uses the AHB slave device port for control/status register access and data access. Features of the AHB interface include: word (32-bit) access. The AHB slave device interface does not support separation/retry requests and never generates AHB errors.

23.2.3 Single-ended channel

In single-ended input mode, the conversion voltage is the difference between the current input channel voltage and external V_{IN}-0.

The analog loop signal AMUXBUS channel is connected to ADC_IN14, and the internal reference voltage LTCBG is connected to ADC_IN15. After selecting the corresponding channel, the input analog voltage can be converted via ADC.

23.2.4 ADC switch control

Enable the ADC by setting the ADEN bit in the ADCx_CR register to 1.

Subsequently, regular conversion can be started by setting ADSTART to 1, or by an external trigger event if the trigger is enabled.

Injected conversion can be started by setting the JADSTART bit to 1, or by an external injected trigger event if the injected trigger is enabled.

Procedure to disable the ADC by software:

Check if ADSTART and JADSTART are 0 to ensure no conversion is currently being performed. If needed, set the ADSTP bit to 1, and set the JADSTP bit to 1, then wait until ADSTP=0 and JADSTP=0 to stop any ongoing regular and injected conversions.

If required by the application, clear ADEN to 0 to disable the ADC.

23.2.5 Restrictions when writing to ADC control bits

ADC clocks can be configured and enabled by writing to RCU control bit via software only when the ADC is disabled (ADEN must be 0) (see the RCU).

Software is allowed to write to the control bits ADSTART, JADSTART, ADSTP, and JADSTP in the ADCx_CR register only when the ADC is enabled.

For all other control bits in the ADCx_CFGR, ADCx_CFGR2, ADCx_SMPRx, ADCx_SQRy, ADCx_JDRy, ADCx_JOFFSETy, ADCx_TRy, and ADCx_IER registers:

For control bits related to regular conversion configurations, software is only

allowed to perform write operations on these bits when regular conversion has not been performed (ADSTART must be equal to 0).

For control bits related to injected conversion configurations, software is only allowed to perform write operations on these bits when injected conversion has not been performed (JADSTART must be equal to 0).

If the ADC is enabled (ADEN=1), software can write to the ADCx_JSQR register at any time.

23.2.6 Channel selection (SQRx, JSQRx)

Each ADC multiplexed channel has up to 16 channels, and conversions can be divided into two groups: regular conversion and injected conversion. Each group contains a conversion sequence that can be completed on any channel in any order.

A regular conversion group consists of a maximum of 16 conversions. The regular channels and their order in the conversion sequence shall be selected in the ADCx_SQRy registers. The total number of conversions in the regular conversion group must be written to the LT3[3:0] bits in the ADCx_SQR1 register.

A injected conversion group consists of a maximum of 4 conversions. The injected channels and their order in the conversion sequence shall be selected in the ADCx_JSQR registers. The total number of conversions in the injected conversion group shall be written to the JL[1:0] bits in the ADCx_JSQR register.

Modification of the ADCx_SQRy registers is not allowed when regular conversions might occur. Therefore, ADSTP=1 shall first be written to stop ADC regular conversion; during injected conversions, the ADCx_JSQR register can be modified in real time.

23.2.7 Sampling time of each channel can be set independently (SMPRx)

Before starting conversion, the ADC must establish a direct connection between the voltage source to be measured and the ADC's built-in sampling capacitor. The sampling time must be sufficient for the input voltage source to charge the embedded capacitor to the input voltage level.

Different sampling times can be used during sampling of each channel. The sampling time can be programmed via the SMPx[2:0] bits in the ADCx_SMPRx register, and independent sampling time can be set for each channel.

The formula for total conversion time is as follows:

$$TCONV = \text{Sampling time} + 13 \text{ ADC clock cycles}$$

23.2.8 Single conversion mode (CONT=0)

In single conversion mode, the ADC performs all conversions on the channel

once. When the CONT bit is 0, this mode can be started by setting the ADSTART bit in the ADCx_CR register to 1 (applicable to regular channels).

Setting the JADSTART bit in the ADCx_CR register can be used to start the injected channel conversion of ADC. Conversion can start immediately or after an injected hardware trigger event.

In regular sequence, each time a conversion is completed, the conversion data is stored in the ADCx_DR register, and the EOC (End of Conversion) flag is set to 1. An interrupt will be generated if the EOCIE bit is set to 1.

In injected sequence, each time a conversion is completed, the conversion data is stored in one of the four ADCx_JDRy registers, and the JEOC (end of injected conversion) flag is set to 1.

An interrupt will be generated when the JEOCIE bit is set to 1. After the regular sequence is completed: the EOS (end of regular sequence) flag will be set to 1, and an interrupt will be generated when the EOSIE bit is set to 1.

After the injected sequence is completed: the JEOS (end of injected sequence) flag is set to 1, and an interrupt will be generated when the JEOSIE bit is set to 1. Subsequently, the ADC stops working until a new external regular or injected trigger occurs, or the ADSTART or JADSTART bit is set to 1 again.

Note: To convert a single channel, program the sequence length to 1.

23.2.9 Continuous conversion mode (CONT=1)

This mode is only applicable to regular channels.

In continuous conversion mode, if a software or hardware regular trigger event occurs, the ADC performs all regular conversions on the channel once, then automatically restarts and continues to perform each conversion in the sequence. When the CONT bit is 1, this mode can be enabled by an external trigger or by setting the ADSTART bit in the ADC_CR register to 1.

In regular sequence, each time a conversion is completed, the conversion data is stored in the ADCx_DR register, and the EOC (end of conversion) flag is set to 1.

An interrupt will be generated when the EOCIE bit is set to 1. After the conversion sequence is completed: the EOS (end of sequence) flag will be set to 1, and an interrupt will be generated when the EOSIE bit is set to 1.

Subsequently, a new sequence is restarted immediately, and the ADC continues to repeat the conversion sequence.

Note: To convert a single channel, program the sequence length to 1.

Discontinuous mode and continuous mode cannot be enabled simultaneously: DISCEN and CONT must not be set to 1 at the same time.

Injected channels cannot be converted continuously. The only exception is that in continuous conversion mode (using the JAUTO bit), the injected channels are configured for automatic conversion after regular channels.

23.2.10 Start of conversion (ADSTART, JADSTART)

Software starts ADC regular conversion by setting ADSTART to 1. After ADSTART is set to 1, conversion will begin: when EXTEN = 0x0 (software trigger).

When EXTEN does not equal 0x0, software starts ADC injected conversion by setting JADSTART to 1, and conversion begins on the next valid edge of the selected regular hardware trigger. After ADSTART is set to 1, conversion will begin: when JEXTEN = 0x0 (software trigger).

When JEXTEN does not equal 0x0, conversion begins on the next valid edge of the selected injected hardware trigger.

Note: In automatic injected mode (JAUTO=1), start regular conversion using the ADSTART bit, and then conduct automatic injected conversion (JADSTART must remain cleared to 0).

The ADSTART and JADSTART bits also provide information on whether ADC operation is being performed. The ADC can be reconfigured when ADSTART=0 and JADSTART=0 (indicating the ADC is idle).

ADSTART is cleared to 0 by hardware:

In single mode using software regular trigger (CONT = 0, EXTEN = 0x0):

- If DISCEN = 1, it will be cleared to 0 in all cases as soon as the regular conversion sequence ends (EOS is set to 1) or the subgroup processing ends (CONT=x, EXTEN=x)
- Cleared after executing the ADSTP procedure called by software

Note: In continuous mode (CONT=1), since the sequence restarts automatically, the ADSTART bit will not be cleared to 0 by hardware when EOS is set to 1.

If hardware triggering is selected in single mode (CONT=0 and EXTEN is not equal to 0x0), the ADSTART bit will not be cleared by hardware when EOS is set to 1. This allows the software to avoid resetting ADSTART for the next hardware trigger event. This ensures that no subsequent hardware triggers are missed.

JADSTART is cleared to 0 by hardware:

In single mode using software injected trigger (JEXTEN = 0x0)

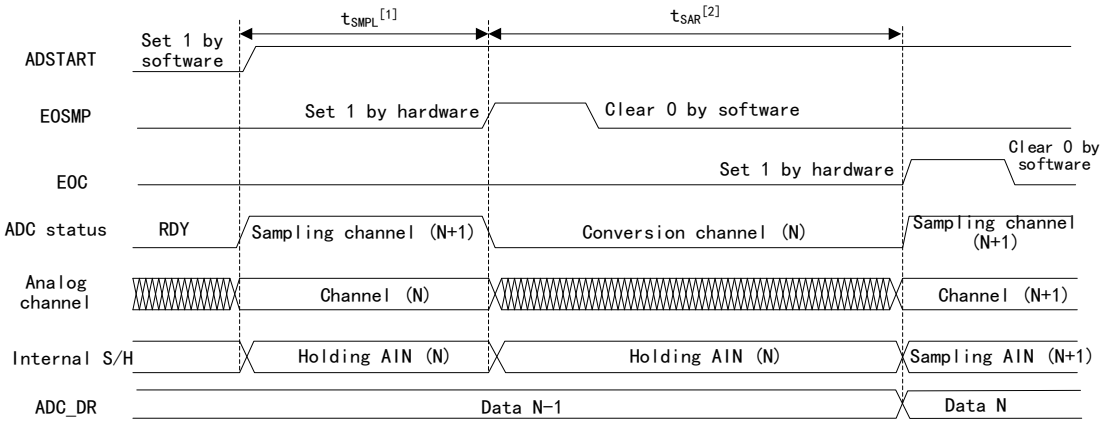
- If DISCEN = 1, it will be cleared to 0 in all cases as soon as the injected conversion sequence ends (JEOS is set to 1) or the subgroup processing ends (JEXTEN=x)
- Cleared after executing the JADSTP procedure called by software

Note: When software trigger is selected, the ADSTART bit shall not be set to 1 if the EOC flag is still high.

23.2.11 ADC timing

The time elapsed from the start to the end of conversion is the sum of the configured sampling time and conversion time.

Figure 132 Analog-to-Digital Conversion Time



Note:

(1) t_{SMPL} depends on SMPx[2:0].

(2) t_{SAR} is the conversion time: 13 ADC_CLK cycles .

23.2.12 Stop the ongoing conversion (ADSTP, JADSTP)

Software decides whether to stop conversion. Setting ADSTP to 1 can stop the ongoing regular conversion, and setting JADSTP to 1 can stop the ongoing injected conversion.

Stopping the conversion will reset the ongoing ADC operation. The ADC can then be reconfigured to prepare for new operations.

Note: Injected conversion can be stopped while a regular conversion is still executing, and vice versa. This allows reconfiguring the injected conversion sequence and trigger while a regular conversion is still in progress, and vice versa.

If the ADSTP bit is set to 1 by software, any ongoing regular conversion will be aborted, and partial conversion results will be discarded (the ADCx_DR register will not be updated with the current conversion result).

If the JADSTP bit is set to 1 by software, any ongoing injected conversion will be aborted, and partial conversion results will be discarded (the ADCx_JDRy register will not be updated with the current conversion result). The scan sequence will also be aborted and reset (this means restarting the ADC will begin a new sequence).

After the program completes execution, the ADSTP/ADSTART bits (for regular

conversion) or JADSTP/JADSTART bits (for injected conversion) will be cleared to 0 by hardware. Software shall poll the ADSTART (or JADSTART) until it is reset, and then it can determine that the ADC has fully stopped.

Note: In automatic injection mode (JAUTO=1), setting ADSTP to 1 will abort both regular and injected conversions (do not use JADSTP).

Figure 133 Stop Ongoing Regular Conversion

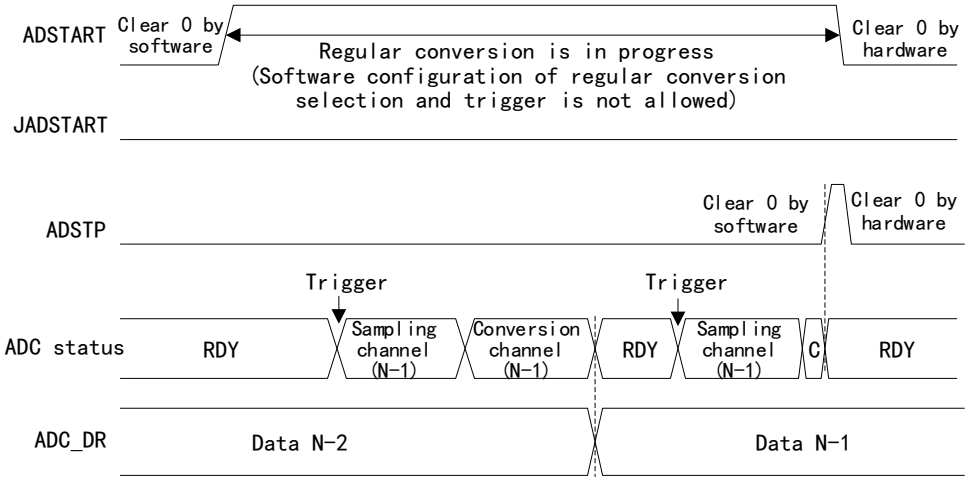
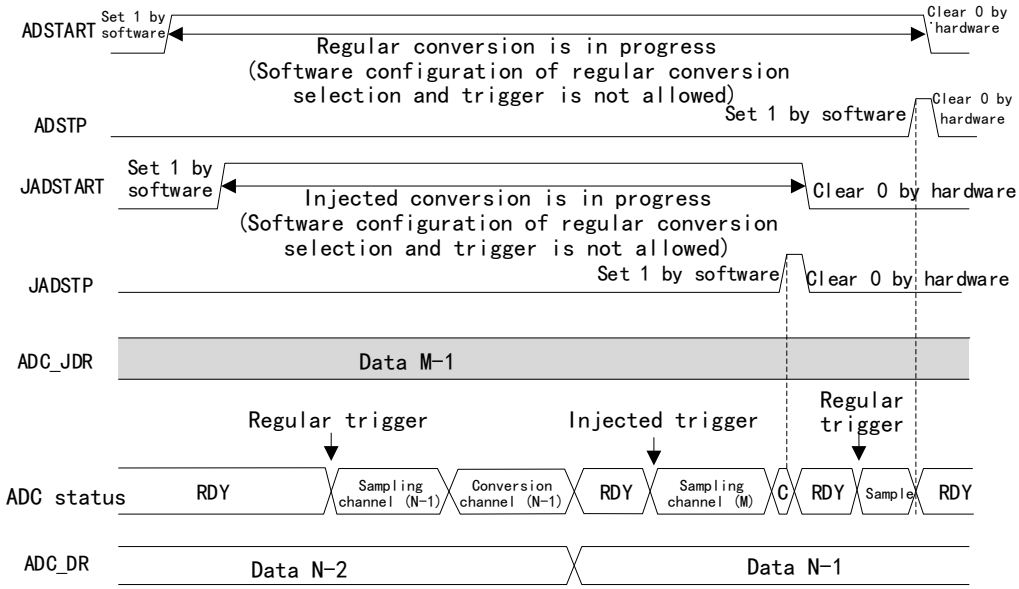


Figure 134 Stop Ongoing Regular Conversion and Injected Conversion



23.2.13 External trigger conversion and trigger polarity (EXTSEL, EXTEN, JEXTSEL, JEXTEN)

Conversions or conversion sequences can be triggered by software or external events (e.g., timer capture, input pin). If the EXTEN control bit (for regular conversion) or JEXTEN bit (for injected conversion) is set to 1, external events can trigger a conversion.

Regular trigger selection becomes effective after software sets ADSTART to 1; injected trigger selection becomes effective after software sets JADSTART to 1. Hardware triggers occurring during conversion will be ignored.

- If the ADSTART bit is 0, any regular hardware triggers that occur will be ignored.
- If the ADSTART bit is 0, any injected hardware triggers that occur will be ignored. The following table provides the correspondence between EXTEN and JEXTEN values and trigger polarity.

Table 91 Configure Trigger Polarity for Regular External Triggers

EXTEN	Source
0	Disable hardware trigger detection, enable software trigger detection
1	Hardware trigger detection on rising edge

Table 92 Configure Trigger Polarity for Injected External Triggers

JEXTEN	Source
0	Disable hardware trigger detection, enable software trigger detection
1	Hardware trigger detection on rising edge

The EXTSEL and JEXTSEL control bits are used to select events from 8 possible events that can trigger regular and injected group conversions. Injected trigger can interrupt regular group conversions.

Note: The regular trigger selection cannot be changed in real-time. The injected trigger selection can be expected and changed in real time.

The following table lists all possible external triggers of the ADC for regular and injected conversions.

Table 93 External Trigger of Regular Channel

Source	Type	EXTSEL[2:0]
TMR1_CC1	Internal signal from on-chip timer	000
TMR1_CC2	Internal signal from on-chip timer	001
TMR1_CC3	Internal signal from on-chip timer	010
TMR2_CC2	Internal signal from on-chip timer	011
TMR3_TRGO	Internal signal from on-chip timer	100
TMR4_CC4	Internal signal from on-chip timer	101
EINT line 11	External pin	110
-	Reserved	111

Table 94 External Trigger of Injected Channel

Source	Type	JEXTSEL[2:0]
TMR1_TRGO	Internal signal from on-chip timer	000
TMR1_CC4	Internal signal from on-chip timer	001
TMR2_TRGO	Internal signal from on-chip timer	010
TMR2_CC1	Internal signal from on-chip timer	011

Source	Type	JEXTSEL[2:0]
TMR3_CC4	Internal signal from on-chip timer	100
TMR4_TRGO	Internal signal from on-chip timer	101
EINT line 15	External pin	110
-	Reserved	111

23.2.14 Injected channel management

Trigger injection mode

To use trigger injection, the JAUTO bit in the ADCx_CFGR register shall be cleared to 0.

Regular channel group conversion is started by an external trigger or by setting the ADSTART bit in the ADCx_CR register to 1.

If an external injected trigger occurs during the regular channel group conversion, or if the JADSTART bit in the ADCx_CR register is set to 1, the current conversion will be reset, and the injected channel sequence switching will be started (all injected channels are converted once). The regular conversion of the regular channel group then resumes from the last interrupted regular conversion.

If a regular event occurs during an injected conversion period, the injected conversion will not be interrupted, but the regular sequence will be executed after the injected sequence ends.

Note: When using triggered injection, ensure that the interval between trigger events is longer than the injected sequence.

Automatic injection mode

If the JAUTO bit in the ADCx_CFGR register is set to 1, the channels in the injected group are automatically converted after the regular group channels. This can be used to convert a sequence of up to 16 conversions, which are programmed in the ADCx_SQRy and ADCx_JSQR registers.

In this mode, the ADSTART bit in the ADCx_CR register shall be set to 1 to start the regular conversion, followed by the injected conversion (JADSTART shall remain cleared to 0). Setting the ADSTP bit to 1 will abort both regular and injected conversions (the JADSTP bit shall not be used). In this mode, external triggers on injected channels shall be disabled.

If both the CONT and the JAUTO bits are set to 1, the regular channels and subsequent injected channels will be continuously converted.

Note: Automatic injection and discontinuous sampling modes cannot be used simultaneously.

23.2.15 Discontinuous modes (DISCEN, DISCNUM, JDISCEN)

Regular group mode

This mode can be enabled by setting the DISCEN bit in the ADCx_CFGR register to 1.

This mode is used to convert short sequences (subgroups) containing n (n ≤ 7) conversions, which are part of the conversion sequence selected in the ADCx_SQRy register. The value of n can be specified by writing to the DISCNUM[2:0] bits in the ADCx_CFGR register.

When an external trigger occurs, the next n conversions selected in the ADCx_SQRy register will start until all conversions in the sequence are completed. The total sequence length is defined by the RL[3:0] bits in the ADCx_SQR1 register.

Examples:

DISCEN=1, n=3, channels to be converted = 1, 2, 3, 6, 7, 8, 9, 10, 11

- First trigger: The converted channels are 1, 2, 3 (an EOC event is generated in each conversion).
- Second trigger: The converted channels are 6, 7, 8 (an EOC event is generated in each conversion).
- Third trigger: The converted channels are 9, 10, 11 (an EOC event is generated in each conversion), and an EOS event will be generated upon conversion completion of Channel 11.
- Fourth trigger: The converted channels are 1, 2, 3 (an EOC event is generated in each conversion).
- ...
- DISCEN=0, channels to be converted = 1, 2, 3, 6, 7, 8, 9, 10, 11
- First trigger: The entire sequence is converted: Channel 1, then Channels 2, 3, 6, 7, 8, 9, 10, and 11. Each conversion generates an EOC event, and the last conversion also generates an EOS event.
- All subsequent trigger events will restart the entire sequence.

Note: When converting a regular group in discontinuous mode, there is no reverse (the number of conversions in the last subgroup of the sequence is less than n).

After converting all subgroups, the next trigger signal will start the conversion of the first subgroup. In the above example, the fourth trigger re-converts Channels 1, 2, and 3 from the first subgroup.

Discontinuous mode and continuous mode cannot be enabled simultaneously. If both modes are enabled simultaneously (i.e., DISCEN=1, CONT=1), the ADC assumes continuous mode is disabled and continues related operation.

Injection group mode

This mode can be enabled by setting the JDISCEN bit in the ADCx_CFGR register to 1. After an external injection trigger event occurs, this mode will convert the sequence selected in the ADCx_JSQR register channel by channel, equivalent to the case where the regular channel "n" in discontinuous mode is fixed to 1.

When an external trigger occurs, the next channel conversions selected in the ADCx_JSQR register will start until all conversions in the sequence are completed. The total sequence length is defined by the JL[1:0] bits in the ADCx_JSQR register.

Examples:

JDISCEN=1, channels to be converted = 1, 2, 3

- First trigger: Convert Channel 1 (a JEOC event is generated)
- Second trigger: Convert Channel 2 (a JEOC event is generated)
- Third trigger: Convert Channel 3, and generate a JEOC event and JEOS event
- ...

Note: After all injected channels are converted, the next trigger signal will start the conversion of the first injected channel. In the above example, the fourth trigger re-converts the first injected channel.

Automatic injection mode and discontinuous mode cannot be used simultaneously: When JAUTO=1, the DISCEN and JDISCEN bits must be kept cleared by software.

23.2.16 End of conversion (EOC, JEOC)

The ADC will notify the application each time an event of end of regular conversion (EOC) or end of injected conversion (JEOC) occurs.

The ADC immediately sets the EOC flag to 1 when new regular conversion data appears in the ADC_DR register. An interrupt can be generated if the EOCIE bit is set to 1. The EOC flag can be cleared to 0 by writing 1 to it via software or reading the ADC_DR.

The ADC immediately sets the JEOC flag to 1 when new injected conversion data appears in the ADCx_JDRy register. An interrupt can be generated if the JEOCIE bit is set to 1. The JEOC flag can be cleared to 0 by writing 1 to it via software or reading the corresponding ADCx_JDRy register.

23.2.17 End of conversion sequence (EOS, JEOS)

The ADC will notify the application each time an event of end of regular sequence (EOS) or end of injected sequence (JEOS) occurs.

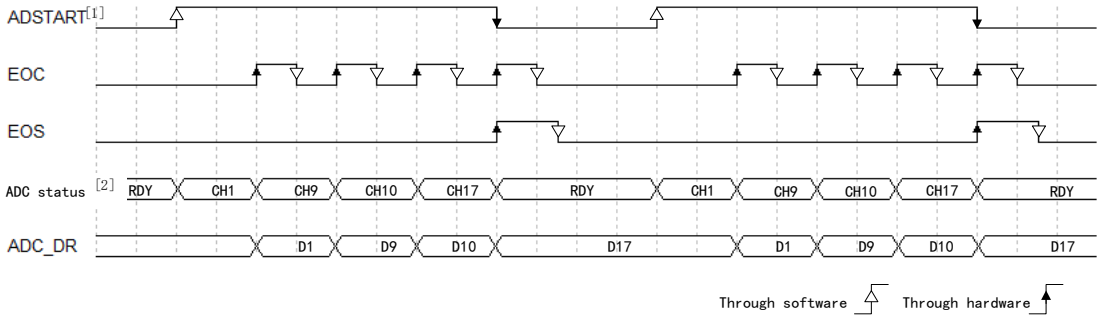
The ADC immediately sets the EOS flag to 1 when the last data of the regular conversion sequence appears in the ADCx_DR register. An interrupt can be

generated if the EOSIE bit is set to 1. The EOS flag can be cleared by software writing 1 to it.

The ADC will set JEOS flag to 1 immediately after the last data of the injected conversion sequence is completed. An interrupt can be generated if the JEOSIE bit is set to 1. The JEOS flag can be cleared to 0 by writing 1 to it via software.

23.2.18 Example timing diagrams (single mode/continuous mode, hardware/software trigger)

Figure 135 Single sequence conversion, software trigger

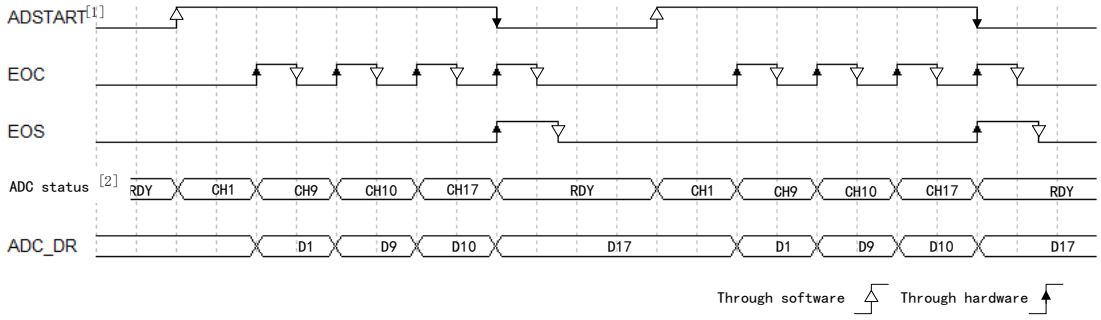


Note:

(3) EXTEN=0x0, CONT=0.

(4) Selected channel=1, 9, 10, 17.

Figure 136 Continuous sequence conversion, software trigger

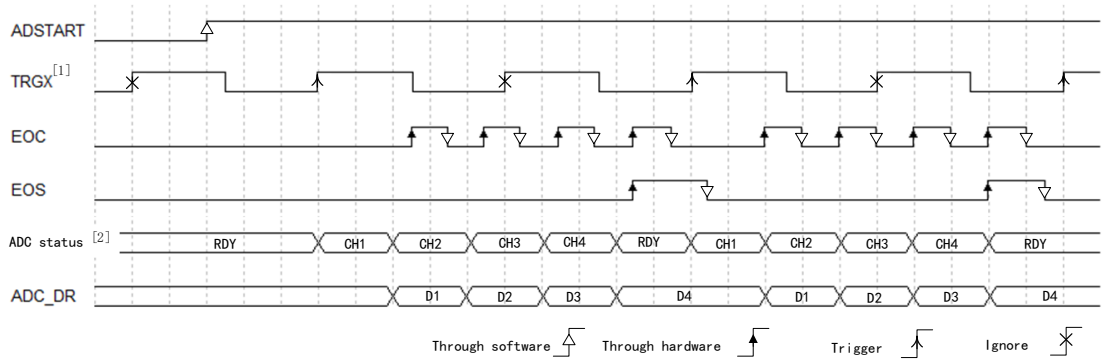


Note:

(1) EXTEN=0x0, CONT=1.

(2) Selected channel=1, 9, 10, 17.

Figure 137 Single sequence conversion, hardware trigger

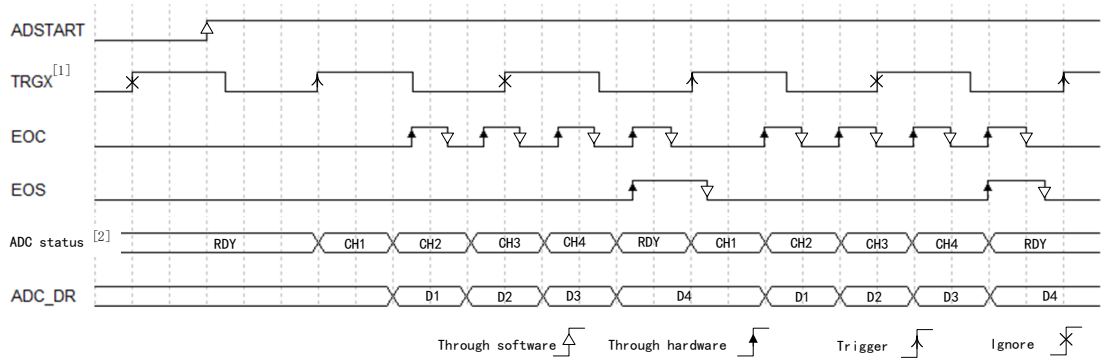


Note:

(1) Select TRGx (over-frequency) as trigger source, EXTEN=01, CONT=0.

(2) Selected channel=1, 2, 3, 4.

Figure 138 Continuous sequence conversion, hardware trigger



Note:

(1) Select TRGx (overfrequency) as the trigger source, EXTEN=01, CONT=1.

(2) Selected channel=1, 2, 3, 4.

23.2.19 Data alignment

The ALIGN bit in the ADC_CFGR register selects the alignment of the converted data storage. Data can be left-aligned or right-aligned, as shown in the figure below.

For injected group channels, the converted data value has been subtracted by the offset defined in the ADC_JOFFSETRx register, so the result can be a negative value. The SEXT bit is the extended sign value.

For regular group channels, there is no need to subtract an offset value, so only 12 bits are valid.

Figure 139 Data right alignment

Injected group	SEXT	SEXT	SEXT	SEXT	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Regular group	0	0	0	0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

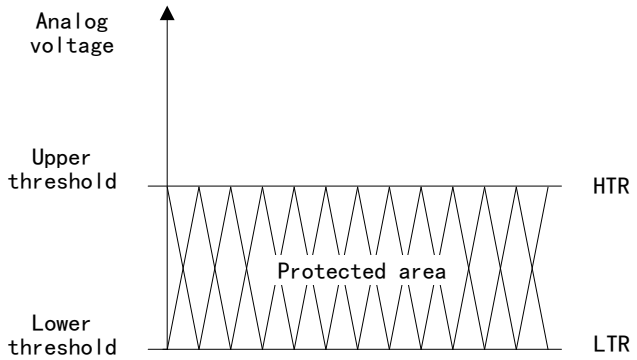
Figure 140 Data left alignment

Injected group	SEXT	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0
Regular group	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0

23.2.20 Analog window watchdogs (AWD1EN, JAWD1EN, AWD1SGL, AWD1CH, AWD_HTx, AWD_LTx, and AWDx)

Three AWD analog watchdogs monitor whether certain channels remain within the configured voltage range (window).

Figure 141 Protected Area of Analog Watchdog



AWDx flags and interrupts

Interrupts can be enabled for the analog watchdog by setting the AWDxIE bit in the ADCx_IER register to 1 (x = 1). The AWDx (x = 1) flag can be cleared to 0 by writing 1 to it via software.

Analog watchdog 1

Setting the AWD1EN bit in the ADCx_CFGR register to 1 can enable the AWD analog watchdog 1. This watchdog monitors a selected channel or all enabled channels to see if they remain within the configured voltage range (window).

The table below introduces how to configure the ADCx_CFGR register to enable the analog watchdog on one or multiple channels.

Table 95 Analog Watchdog Channel Selection

Channels protected by analog watchdog	AWD1SGL bit	AWD1EN bit	JAWD1EN bit
None	x	0	0
All injected channels	0	0	1
All regular channels	0	1	0
All regular and injected channels	0	1	1
Single injected channel	1	0	1
Single regular channel	1	1	0
Single regular or injected channel	1	1	1

Select through the AWD1CH[4:0] bits. Additionally, the channel must be programmed for conversion in the appropriate regular or injected sequence.

If the analog voltage for the ADC conversion falls below the lower threshold or exceeds the upper threshold, the AWD1 analog watchdog status bit will be set to 1. These thresholds are programmed into the HT1[11:0] and LT1[11:0] bits of the ADCx_TR1 register for analog watchdog 1.

23.2.21 DMA request

Since the converted values of regular channels are stored in a single data register, DMA needs to be used when converting multiple regular channels to avoid losing data already stored in the ADC_DR register.

A DMA request is generated only at the end of the conversion of a regular channel, and the converted data is transferred from the ADC_DR register to the user-specified destination address. DMA transfer data size only supports words (32 bits).

23.2.22 ADC (low-power mode)

Table 96 Impact of low-power Mode on ADC

Mode	Description
Low-power run mode	No impact
Stop mode	ADC stops operation
Standby mode	ADC powers down, and it can start to work only after the system exits standby mode and becomes stable.

23.2.23 ADC interrupt

Interrupts are generated in the following cases:

- At the end of any conversion of the regular group (EOC flag)
- At the end of the conversion sequence of the regular group (EOS flag)
- At the end of any conversion of the injected group (EOC flag)
- At the end of the conversion sequence of the injected group (JEOS flag)

- When analog watchdog detection occurs (AWD1 flag)

Separate interrupt enable bits can be used for flexibility.

Table 97 ADC Interrupt

Interrupt event	Event flag	Enable control bit
End of conversion of regular group	EOC	EOCIE
End of conversion sequence of regular group	EOS	EOSIE
End of conversion of injected group	JEOC	JEOCIE
End of conversion sequence of injected group	JEOS	JEOSIE
Analog watchdog 1 status bit set to 1	AWD1	AWD1IE

23.3 Register address mapping

Table 98 ADC12 Register Address Mapping

Register name	Description	Offset address
ADC12_ISR	Interrupt status register	0x00
ADC12_IER	Interrupt enable register	0x04
ADC12_CR	Control register	0x08
ADC12_CFGR	Configuration register	0x0C
ADC12_SMPR1	Sampling time register 1	0x14
ADC12_SMPR2	Sampling time register 2	0x18
ADC12_TR1	Analog watchdog register 1	0x20
ADC12_SQR1	Regular sequence register 1	0x2C
ADC12_SQR2	Regular sequence register 2	0x30
ADC12_SQR3	Regular sequence register 3	0x34
ADC12_DR	Data register	0x38
ADC12_JSQR	Injected sequence register	0x3C
ADC12_JDR1	Injected channel data register 1	0x40
ADC12_JDR2	Injected channel data register 2	0x44
ADC12_JDR3	Injected channel data register 3	0x48
ADC12_JDR4	Injected channel data register 4	0x4C
ADC12_JOFFSET1	Injected channel offset register 1	0x58
ADC12_JOFFSET2	Injected channel offset register 2	0x5C
ADC12_JOFFSET3	Injected channel offset register 3	0x60
ADC12_JOFFSET4	Injected channel offset register 4	0x64

23.4 Register functional description

23.4.1 Interrupt status register (ADC12_ISR)

Offset address: 0x00

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:8	Reserved		
7	AWD1	RC_W1	Analog Watchdog 1 Flag This bit will be set to 1 by hardware when the conversion voltage exceeds the value programmed in the LT1[11:0] and HT1[11:0] fields of the ADCx_TR1 register. This bit can be cleared by software writing 1 to it. 0: No analog watchdog 1 event occurs (or the flag event has been acknowledged and cleared to 0 by software) 1: Analog watchdog 1 event occurs
6	JEOS	RC_W1	Injected Channel End of Sequence Flag This bit will be set to 1 by hardware at the end of the conversion for all the injected channels in the group. This bit can be cleared by software writing 1 to it. 0: The injected conversion sequence is not completed (or the flag event has been acknowledged and cleared to 0 by software) 1: The injected conversion is completed
5	JEOC	RC_W1	Injected Channel End of Conversion Flag This bit will be set to 1 by hardware when each injected conversion of a channel is ended and new data appears in the corresponding ADCx_JDRY register. This bit can be cleared to 0 by writing 1 to it via software or reading the corresponding ADCx_JDRY register. 0: The injected channel conversion is not completed (or the flag event has been acknowledged and cleared to 0 by software) 1: The injected channel conversion is completed
4	Reserved		
3	EOS	RC_W1	End of Regular Sequence Flag This bit will be set to 1 by hardware at the end of conversion of the regular channel sequence. This bit can be cleared by software writing 1 to it. 0: The regular conversion sequence is not completed (or the flag event has been acknowledged and cleared to 0 by software) 1: Regular conversion sequence is completed
2	EOC	RC_W1	End Of Conversion Flag This bit will be set to 1 by hardware when each regular conversion of a channel is ended and new data appears in the ADCx_DR register. This bit can be cleared by software writing 1 to it or reading the ADCx_DR register. 0: The regular channel conversion is not completed (or the flag event has been confirmed and cleared by software) 1: The regular channel conversion is completed
1:0	Reserved		

23.4.2 Interrupt enable register (ADC12_IER)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:8	Reserved		
7	AWD1IE	R/W	Analog Watchdog 1 Interrupt Enable This bit is set to 1 and cleared to 0 by software to enable/disable the analog watchdog 1 interrupt. 0: Disable the analog watchdog 1 interrupt 1: Enable the analog watchdog 1 interrupt Note: Software is only allowed to write to this bit when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).
6	JEOSIE	R/W	End of Injected Sequence Of Conversions Interrupt Enable This bit is set to 1 and cleared to 0 by software to enable/disable the injected conversion sequence end interrupt. 0: Disable JEOS interrupt. 1: Enable JEOS interrupt. An interrupt is generated when the JEOS bit is set to 1. Note: Software is only allowed to write to this bit when JADSTART=0 (this ensures that no injected conversion is currently in progress).
5	JEOCIE	R/W	End of Injected Conversion Interrupt Enable This bit is set to 1 and cleared to 0 by software to enable/disable the injected conversion end interrupt. 0: Disable JEOC interrupt. 1: Enable JEOC interrupt. An interrupt is generated when the JEOC bit is set to 1. Note: Software is only allowed to write to this bit when JADSTART=0 (this ensures that no regular conversion is currently in progress).
4	Reserved		
3	EOSIE	R/W	End of Regular Sequence Of Conversions Interrupt Enable This bit is set to 1 and cleared to 0 by software to enable/disable the regular conversion sequence end interrupt. 0: Disable EOS interrupt 1: Enable EOS interrupt. An interrupt is generated when the EOS bit is set to 1. Note: Software is only allowed to write to this bit when ADSTART=0 (this ensures that no regular conversion is currently in progress).
2	EOCIE	R/W	End of Regular Conversion Interrupt Enable This bit is set to 1 and cleared to 0 by software to enable/disable the regular conversion end interrupt. 0: Disable EOC interrupt. 1: Enable EOC interrupt. An interrupt is generated when the EOC bit is set to 1. Note: Software is only allowed to write to this bit when ADSTART=0 (this ensures that no regular conversion is currently in progress).
1:0	Reserved		

23.4.3 Control register (ADC12_CR)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:6	Reserved		
5	JADSTP	R/W	ADC Stop of Injected Conversion Command This bit is set to 1 by software to stop and discard an ongoing injected conversion (JADSTP command). The bit is cleared to 0 by hardware when the conversion has been effectively discarded and the ADC injected sequence and trigger can be reconfigured. Subsequently, the ADC will be ready to receive a new command to start injected conversion (JADSTART command). 0: The command that ADC stops injected conversion currently is not being executed. 1: Writing 1 can sop the ongoing injected conversion. A read value of 1 indicates that the ADSTP command is being executed. Note: Software is only allowed to set JADSTP to 1 when JADSTART=1 and ADDIS=0 (the ADC is enabled, will eventually perform injected conversion, and there are no pending requests to disable the ADC). In automatic injection mode (JAUTO=1), setting the ADSTP bit to 1 will abort both regular and injected conversions (do not use JADSTP).
4	ADSTP	R/W	ADC Stop of Regular Conversion Command This bit is set to 1 by software to stop and discard an ongoing regular conversion (ADSTP command). The bit is cleared by hardware when the conversion has been effectively discarded and the ADC regular sequence and trigger can be reconfigured. Subsequently, the ADC will be ready to receive a new command to start regular conversion (ADSTART command). 0: The command that ADC stops regular conversion currently is not being executed. 1: Writing 1 stops the ongoing regular conversion. A read value of 1 indicates that the ADSTP command is being executed. Note: Software is only allowed to set ADSTP to 1 when ADSTART=1 and ADDIS=0 (the ADC is enabled, will eventually perform regular conversion, and there are no pending requests to disable the ADC). In automatic injection mode (JAUTO=1), setting the ADSTP bit to 1 will abort both regular and injected conversions (do not use JADSTP).
3	JADSTART	R/W	ADC Start of Injected Conversion This bit is set to 1 by software to start the injected channel conversion of the ADC. Depending on the value of the EXTEN configuration bit, conversion can start immediately (software trigger configuration) or after an injected hardware trigger event occurs (hardware trigger configuration). This bit is cleared to 0 by hardware: In single-conversion mode, if software trigger is selected (JEXTSEL=0x0): It is cleared to 0 when the flag for the end of injected conversion sequence (JEOS) appears.

Field	Name	R/W	Description
			In all cases: It is cleared to 0 by hardware when the JADSTP bit is cleared to 0 after the execution of the JADSTP command. 0: No ADC injected conversion is currently in progress 1: Writing 1 can start injected conversion. A read value of 1 indicates that the ADC is running and will eventually convert injected channels. Note: Software is only allowed to set JADSTART to 1 when ADEN=1 and ADDIS=0 (the ADC is enabled, and there are no pending requests to disable the ADC). In automatic injected mode (JAUTO=1), start regular conversion and automatic injected conversion by setting the ADSTART bit to 1 (JADSTART must remain cleared to 0).
2	ADSTART	R/W	ADC Start of Regular Conversion This bit is set to 1 by software to start the regular channel conversion of the ADC. Depending on the value of the EXTEN configuration bit, conversion can start immediately (software trigger configuration) or after a regular hardware trigger event occurs (hardware trigger configuration). This bit is cleared to 0 by hardware: In single-conversion mode, if software trigger is selected (EXTSEL=0X0): It is cleared to 0 when the flag for the end of regular conversion sequence (EOS) appears. In all cases: It is cleared to 0 by hardware when the ADSTP bit is cleared to 0 after the execution of the ADSTP command. 0: No ADC regular conversion is currently in progress 1: Writing 1 starts regular conversion. A read value of 1 indicates that the ADC is running and will eventually convert regular channels. Note: Software is only allowed to set ADSTART to 1 when ADEN=1 and ADDIS=0 (the ADC is enabled, and there are no pending requests to disable the ADC). In automatic injection mode (JAUTO=1), setting the ADSTART bit to 1 starts regular conversion and automatic injection conversion (JADSTART must remain cleared to 0).
1	Reserved		
0	ADEN	R/W	ADC Enable Control This bit is set to 1 and cleared to 0 by software to enable the ADC. 0: Disable ADC (OFF state) 1: Write 1 to enable the ADC

23.4.4 Configuration register (ADC12_CFGR)

Offset address: 0x0c

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:30	Reserved		
29:26	AWD1CH	R/W	Analog Watchdog 1 Channel Selection These bits are set to 1 and cleared to 0 by software. They are used to select the input channel monitored by the analog watchdog. 0000: Monitor the ADC analog input channel 0 via AWD1 0001: Monitor the ADC analog input channel 1 via AWD1

Field	Name	R/W	Description
			0010: Monitor the ADC analog input channel 2 via AWD1 ..... 1100: Monitor the ADC analog input channel 12 via AWD1 1101: Monitor the ADC analog input channel 13 via AWD1 Other: Reserved, cannot be used Note: The channel selected via AWD1CH must also be selected in the SQRI or JSQRI register. Software is only allowed to write to these bits when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).
25	JAUTO	R/W	Automatic Injected Group Conversion This bit is set to 1 and cleared to 0 by software to enable/disable automatic injected group conversion after regular group conversion. 0: Disable automatic injected group conversion 1: Enable automatic injected group conversion Note: Software is only allowed to write to this bit when ADSTART=0 and JADSTART=0 (this ensures no regular conversion or injected conversion is currently in progress).
24	JAWD1EN	R/W	Analog Watchdog 1 Enable on Injected Channels This bit is set to 1 and cleared to 0 by software. 0: Disable analog watchdog 1 on injected channel 1: Enable analog watchdog 1 on injected channel Note: Software is only allowed to write to this bit when JADSTART=0 (this ensures that no injected conversion is currently in progress).
23	AWD1EN	R/W	Analog Watchdog 1 Enable on Regular Channels This bit is set to 1 and cleared to 0 by software. 0: Disable analog watchdog 1 on regular channel 1: Enable analog watchdog 1 on regular channel Note: Software is only allowed to write to this bit when ADSTART=0 (this ensures that no regular conversion is currently in progress).
22	AWD1SGL	R/W	Enable The Watchdog 1 on A Single Channel or on All Channels This bit is set to 1 and cleared to 0 by software to enable the analog watchdog on the channel determined by the AWD1CH[3:0] bits or on all channels. 0: Enable analog watchdog 1 on all channels 1: Enable analog watchdog 1 on a single channel Note: Software is only allowed to write to these bits when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).
21	Reserved		
20	JDISCEN	R/W	Discontinuous Mode On Injected Channels This bit is set to 1 and cleared to 0 by software to enable/disable the discontinuous sampling mode for injected channels in the group. 0: Disable discontinuous sampling mode for injected channels 1: Enable discontinuous sampling mode for injected channels Note: Software is only allowed to write to this bit when JADSTART=0 (this ensures that no injected conversion is currently in progress). Automatic injection mode and discontinuous mode cannot be used simultaneously: When JAUTO=1, the DISCEN and JDISCEN bits must be kept cleared by software.

Field	Name	R/W	Description
19:17	DISCNUM	R/W	Discontinuous Mode Channel Count） These bits are written by software to define the number of regular channels converted in discontinuous sampling mode after receiving an external trigger. 000: 1 channel 001: 2 channels ... 110: 7 channels 111: 8 channels Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).
16	DISCEN	R/W	Discontinuous Mode For Regular Channels This bit is set to 1 and cleared to 0 by software to enable/disable the discontinuous mode of regular channels. 0: Disable discontinuous mode for regular channels 1: Enable discontinuous mode for regular channels Note: Discontinuous mode and continuous mode cannot be enabled simultaneously: DISCEN and CONT must not be set to 1 at the same time. Automatic injection mode and discontinuous mode cannot be used simultaneously: When JAUTO=1, the DISCEN and JDISCEN bits must be kept cleared by software. Software is only allowed to write to this bit when ADSTART=0 (to ensure no regular conversion is currently in progress).
15:14	Reserved		
13	CONT	R/W	Single/Continuous Conversion Mode For Regular Conversions This bit is set to 1 and cleared to 0 by software. When this bit is set to 1, regular conversions will continue until the bit is cleared. 0: Single conversion mode 1: Continuous conversion mode Note: Discontinuous mode and continuous mode cannot be enabled simultaneously: DISCEN and CONT must not be set to 1 at the same time. Software is only allowed to write to this bit when ADSTART=0 (to ensure no regular conversion is currently in progress).
12:11	Reserved		
10	EXTEN	R/W	External Trigger Enable And Polarity Selection For Regular Channels These bits are set to 1 and cleared to 0 by software to select external trigger polarity and enable the trigger of regular group. 0: Disable hardware trigger detection (conversion can be started by software) 1: Execute hardware trigger detection on rising edge Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).
9	Reserved		
8:6	EXTSEL	R/W	External Trigger Selection For Regular Group These bits select the external event used to trigger conversion of the regular group.

Field	Name	R/W	Description
			000: Event 0 001: Event 1 010: Event 2 011: Event 3 100: Event 4 101: Event 5 110: Event 6 111: Event 7 Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).
5	ALIGN	R/W	Data Alignment This bit is set and cleared by software. 0: Right-aligned 1: Left-aligned Software is only allowed to write to these bits when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).
4:1	Reserved		
0	DMAEN	R/W	Direct Memory Access Enable This bit is set to 1 and cleared to 0 by software to enable the generation of DMA requests. 0: Enable DMA 1: Enable DMA Note: Software is only allowed to write to this bit when ADSTART=0 (this ensures that no regular conversion is currently in progress).

23.4.5 Sampling time register 1 (ADC12_SMPR1)

Offset address: 0x14

Reset value: 0x0000 0000

Field	Name	R/W	Description
31	Reserved		
30:28	SMP7[2:0]	R/W	Channel 7 Sampling Time Selection
27	Reserved		
26:24	SMP6[2:0]	R/W	Channel 6 Sampling Time Selection
23	Reserved		
22:20	SMP5[2:0]	R/W	Channel 5 Sampling Time Selection
19	Reserved		
18:16	SMP4[2:0]	R/W	Channel 4 Sampling Time Selection
15	Reserved		
14:12	SMP3[2:0]	R/W	Channel 3 Sampling Time Selection
11	Reserved		
10:8	SMP2[2:0]	R/W	Channel 2 Sampling Time Selection
7	Reserved		
6:4	SMP1[2:0]	R/W	Channel 1 Sampling Time Selection
3	Reserved		

Field	Name	R/W	Description
2:0	SMP0[2:0]	R/W	Channel 0 Sampling Time Selection These bits are written by software to select the sampling time for each channel separately. During the sampling period, the channel selection bits must remain unchanged. 000: 7 ADC clock cycles 001: 16 ADC clock cycles 010: 27 ADC clock cycles 011: 32 ADC clock cycles 100: 67 ADC clock cycles 101: 96 ADC clock cycles 110: 128 ADC clock cycles 111: 256 ADC clock cycles Note: Software is only allowed to write to these bits when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).

23.4.6 Sampling time register 2 (ADC12_SMPR2)

Offset address: 0x18

Reset value: 0x0000 0000

Field	Name	R/W	Description
31	Reserved		
30:28	SMP15[2:0]	R/W	Channel 15 Sampling Time Selection
27	Reserved		
26:24	SMP14[2:0]	R/W	Channel 14 Sampling Time Selection
23	Reserved		
22:20	SMP13[2:0]	R/W	Channel 13 Sampling Time Selection
19	Reserved		
18:16	SMP12[2:0]	R/W	Channel 12 Sampling Time Selection
15	Reserved		
14:12	SMP11[2:0]	R/W	Channel 11 Sampling Time Selection
11	Reserved		
10:8	SMP10[2:0]	R/W	Channel 10 Sampling Time Selection
7	Reserved		
6:4	SMP9[2:0]	R/W	Channel 9 Sampling Time Selection
3	Reserved		
2:0	SMP8[2:0]	R/W	Channel 8 Sampling Time Selection These bits are written by software to select the sampling time for each channel separately. During the sampling period, the channel selection bits must remain unchanged. 000: 7 ADC clock cycles 001: 16 ADC clock cycles 010: 27 ADC clock cycles 011: 32 ADC clock cycles 100: 67 ADC clock cycles 101: 96 ADC clock cycles 110: 128 ADC clock cycles

Field	Name	R/W	Description
			111: 256 ADC clock cycles Note: Software is only allowed to write to these bits when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).

23.4.7 Analog watchdog register 1 (ADC12_TR1)

Offset address: 0x20

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:28	Reserved		
27:16	HT1[11:0]	R/W	Analog Watchdog 1 Higher Threshold These bits are written by software to define the higher threshold for analog watchdog 1. Note: Software is only allowed to write to these bits when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).
15:12	Reserved		
11:0	LT1[11:0]	R/W	Analog Watchdog 1 Lower Threshold These bits are written by software to define the lower threshold for analog watchdog 1. Note: Software is only allowed to write to these bits when ADSTART=0 and JADSTART=0 (this ensures no conversion is currently in progress).

23.4.8 Regular sequence register 1 (ADC12_SQR1)

Offset address: 0x2C

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:28	SQ7[3:0]	R/W	7th Conversion In Regular Sequence These bits are written by software to assign the channel number as the 7th conversion in regular conversion sequence. Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).
27:24	SQ6[3:0]	R/W	6th Conversion In Regular Sequence These bits are written by software to assign the channel number as the 6th conversion in regular conversion sequence. Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).
23:20	SQ5[3:0]	R/W	5th Conversion In Regular Sequence These bits are written by software to assign the channel number as the 5th conversion in regular conversion sequence. Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).
19:16	SQ4[3:0]	R/W	4th Conversion In Regular Sequence These bits are written by software to assign the channel number as the 4th conversion in regular conversion sequence. Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).

Field	Name	R/W	Description
15:12	SQ3[3:0]	R/W	3rd Conversion In Regular Sequence These bits are written by software to assign the channel number as the 3rd conversion in regular conversion sequence. Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).
11:8	SQ2[3:0]	R/W	2nd Conversion In Regular Sequence These bits are written by software to assign the channel number as the 2nd conversion in regular conversion sequence. Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).
7:4	SQ1[3:0]	R/W	1st Conversion In Regular Sequence These bits are written by software to assign the channel number as the 1st conversion in regular conversion sequence. Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).
3:0	LT3[3:0]	R/W	Regular Channel Sequence Length These bits are written by software to define the total number of conversions in the regular channel conversion sequence. 0000: 1 conversion 0001: 2 conversions ... 1111: 16 conversions Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).

23.4.9 Regular sequence register 2 (ADC12_SQR2)

Offset address: 0x30

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:28	Reserved		
27:24	SQ14[3:0]	R/W	14th Conversion In Regular Sequence These bits are written by software to assign the channel number as the 14th conversion in regular conversion sequence. Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).
23:20	SQ13[3:0]	R/W	13th Conversion In Regular Sequence These bits are written by software to assign the channel number as the 13th conversion in regular conversion sequence. Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).
19:16	SQ12[3:0]	R/W	12th Conversion In Regular Sequence These bits are written by software to assign the channel number as the 12th conversion in regular conversion sequence. Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).
15:12	SQ11[3:0]	R/W	11th Conversion In Regular Sequence

Field	Name	R/W	Description
			These bits are written by software to assign the channel number as the 11th conversion in regular conversion sequence. Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).
11:8	SQ10[3:0]	R/W	10th Conversion In Regular Sequence These bits are written by software to assign the channel number as the 10th conversion in regular conversion sequence. Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).
7:4	SQ9[3:0]	R/W	9th Conversion In Regular Sequence These bits are written by software to assign the channel number as the 9th conversion in regular conversion sequence. Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).
3:0	SQ8[3:0]	R/W	8th Conversion In Regular Sequence These bits are written by software to assign the channel number as the 8th conversion in regular conversion sequence. Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).

23.4.10 Regular sequence register 3 (ADC12_SQR3)

Offset address: 0x34

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:8	Reserved		
7:4	SQ16[3:0]	R/W	16th Conversion In Regular Sequence These bits are written by software to assign the channel number as the 16th conversion in regular conversion sequence. Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).
3:0	SQ15[3:0]	R/W	15th Conversion In Regular Sequence These bits are written by software to assign the channel number as the 15th conversion in regular conversion sequence. Note: Software is only allowed to write to these bits when ADSTART=0 (this ensures no regular conversion is currently in progress).

23.4.11 Data register (ADC12_DR)

Offset address: 0x38

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15:0	RDATA[15:0]	R	Regular Data Converted These bits are read-only. They contain the conversion result of the last converted regular channel.

23.4.12 Injected sequence register (ADC12_JSQR)

Offset address: 0x3C

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:24	Reserved		
23:20	JSQ4[3:0]	R/W	4th Conversion in the Injected Sequence These bits are written by software to assign the channel number as the 4th conversion in injected conversion sequence. Note: Software is only allowed to write to these bits when JADSTART=0 and JAUTO=0 (this ensures no injected conversion is currently in progress).
19:16	JSQ3[3:0]	R/W	3rd Conversion in the Injected Sequence These bits are written by software to assign the channel number as the 3rd conversion in injected conversion sequence. Note: Software is only allowed to write to these bits when JADSTART=0 and JAUTO=0 (this ensures no injected conversion is currently in progress).
15:12	JSQ2[3:0]	R/W	2nd Conversion in the Injected Sequence These bits are written by software to assign the channel number as the 2nd conversion in injected conversion sequence. Note: Software is only allowed to write to these bits when JADSTART=0 and JAUTO=0 (this ensures no injected conversion is currently in progress).
11:8	JSQ1[3:0]	R/W	1st Conversion in the Injected Sequence These bits are written by software to assign the channel number as the 1st conversion in injected conversion sequence. Note: Software is only allowed to write to these bits when JADSTART=0 and JAUTO=0 (this ensures no injected conversion is currently in progress).
7	Reserved		
6	JEXTEN	R/W	External Trigger Enable And Polarity Selection For Injected Channels These bits are set to 1 and cleared to 0 by software to select external trigger polarity and enable the trigger of injected group. 0: Disable hardware trigger detection (conversion can be started by software) 1: Execute hardware trigger detection on rising edge Note: Software is only allowed to write to these bits when JADSTART=0 and JAUTO=0 (this ensures no injected conversion is currently in progress).
5	Reserved		
4:2	JEXTSEL[2:0]	R/W	External Trigger Selection For Injected Group These bits can be used to select the external event used to trigger the injected group conversion: 000: Event 0 001: Event 1 010: Event 2 011: Event 3 100: Event 4 101: Event 5

Field	Name	R/W	Description
			110: Event 6 111: Event 7 Note: Software is only allowed to write to these bits when JADSTART=0 and JAUTO=0 (this ensures no injected conversion is currently in progress).
1:0	JL[1:0]	R/W	Injected Channel Sequence Length These bits are written by software to define the total number of conversions in the injected channel conversion sequence. 00: 1 conversion 01: 2 conversions 10: 3 conversions 11: 4 conversions Note: Software is only allowed to write to these bits when JADSTART=0 and JAUTO=0 (this ensures no injected conversion is currently in progress).

23.4.13 Injected channel data register 1 (ADC12_JDR1)

Offset address: 0x40

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15:0	JDATA1[15:0]	R	Injected Data These bits are read-only, including conversion results of injected channels. Data is left-aligned or right-aligned.

23.4.14 Injected channel data register 2 (ADC12_JDR2)

Offset address: 0x44

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15:0	JDATA2[15:0]	R	Injected Data These bits are read-only, including conversion results of injected channels. Data is left-aligned or right-aligned.

23.4.15 Injected channel data register 3 (ADC12_JDR3)

Offset address: 0x48

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15:0	JDATA3[15:0]	R	Injected Data These bits are read-only, including conversion results of injected channels. Data is left-aligned or right-aligned.

23.4.16 Injected channel data register 4 (ADC12_JDR4)

Offset address: 0x4c

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15:0	JDATA4[15:0]	R	Injected Data These bits are read-only, including conversion results of injected channels. Data is left-aligned or right-aligned.

23.4.17 Injected channel offset register 1 (ADC12_JOFFSET1)

Offset address: 0x58

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:12	Reserved		
11:0	JOFFSET1[11:0]	R/W	Inject channel offset register 1 Data offset compensation for the first conversion in the injected sequence. These bits are written by software to define the offset subtracted from the raw conversion data when converting a channel (which can be a regular or injected channel). Note: Software is only allowed to write to these bits when JAUTO=0 and JADSTART=0 (this ensures no injected conversion is currently in progress).

23.4.18 Injected channel offset register 2 (ADC12_JOFFSET2)

Offset address: 0x5C

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:12	Reserved		
11:0	JOFFSET2[11:0]	R/W	Inject channel offset register 2 Data offset compensation for the second conversion in the injected sequence. These bits are written by software to define the offset subtracted from the raw conversion data when converting a channel (which can be a regular or injected channel). Note: Software is only allowed to write to these bits when JAUTO=0 and JADSTART=0 (this ensures no injected conversion is currently in progress).

23.4.19 Injected channel offset register 3 (ADC12_JOFFSET3)

Offset address: 0x60

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:12	Reserved		
11:0	JOFFSET3[11:0]	R/W	Inject channel offset register 3 Data offset compensation for the third conversion in the injected sequence. These bits are written by software to define the offset subtracted from the raw conversion data when converting a channel (which can be a regular or injected channel). Note: Software is only allowed to write to these bits when JAUTO=0 and JADSTART=0 (this ensures no injected conversion is currently in progress).

23.4.20 Injected channel offset register 4 (ADC12_JOFFSET4)

Offset address: 0x64

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:12	Reserved		
11:0	JOFFSET4[11:0]	R/W	Inject channel offset register 4 Data offset compensation for the fourth conversion in the injected sequence. These bits are written by software to define the offset subtracted from the raw conversion data when converting a channel (which can be a regular or injected channel). Note: Software is only allowed to write to these bits when JAUTO=0 and JADSTART=0 (this ensures no injected conversion is currently in progress).

24 Digital-to-analog Converter (DAC)

24.1 Introduction

The Digital-to-Analog Converter (DAC) is a 10-bit digital-to-analog converter that converts input digital signals into analog voltage outputs. DAC data can be set to be left-aligned or right-aligned. The reference voltage of the DAC is VDDA. The DAC integrates an internal buffer to provide higher drive current.

24.2 Main characteristics

(1) Single output channel

(2) Data can be left-aligned or right-aligned

(3) Waveform generator

(4) Noise generator

(5) Triangular wave generator

(6) Sawtooth wave generator

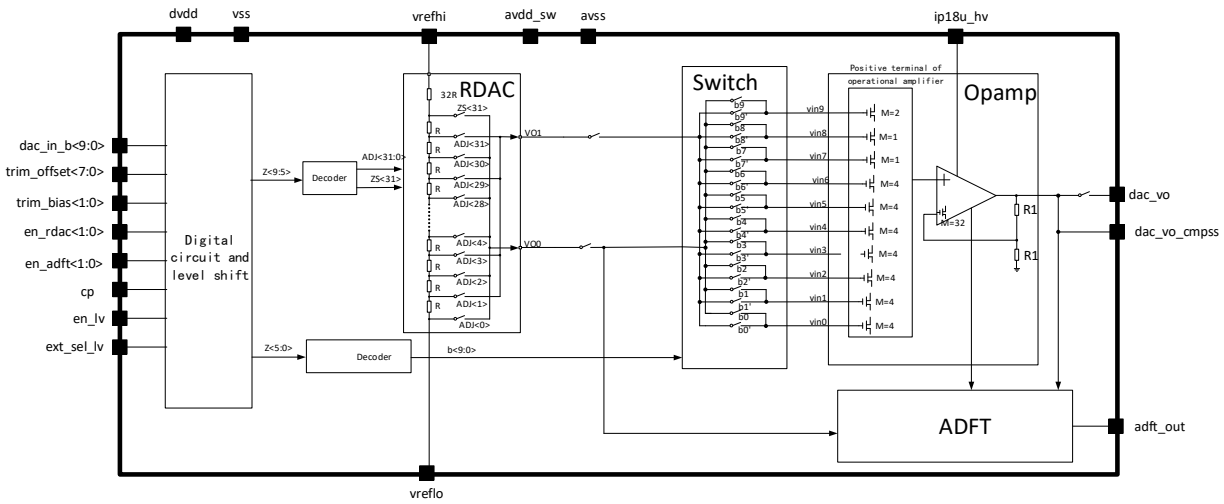
(7) Conversion supports internal, external, and software triggers

(8) Buffer offset calibration

(9) DAC can be configured to output to an external pin or connect to other internal peripherals.

24.3 Structure block diagram

Figure 142 DAC Structure Block Diagram



24.4 Functional description

24.4.1 DAC channel enable

The DAC channel can be enabled by setting the EN bit in the DACx_CR register. After enabled, the DAC channel requires a stabilization time.

24.4.2 DAC output buffer enable

The DAC integrates an output buffer internally. The driving capability can be increased by the output buffer. The output buffer of DAC is automatically enabled by hardware when the EN bit in DACx_CR is set to 1.

The EXT_SEL bit in the DACx_CR register selects whether the output buffer is connected to an external pin or to other internal peripherals. When EXT_SEL is 0, the output buffer is connected to other internal peripherals; when EXT_SEL is 1, it is connected to an external pin.

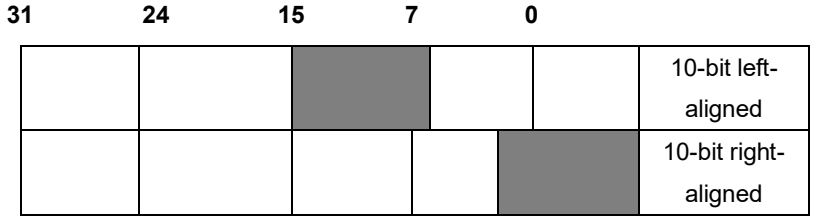
24.4.3 DAC data format

Single-DAC channel

- 10-bit left-aligned: Software must load data into the DACx_DHR10L[31:22] bits (stored in the DHR[9:0] bits).
- 10-bit right-aligned: Software must load data into the DACx_DHR10R[9:0] bits (stored in the DHR[9:0] bits).

According to the loaded DACx_DHR register, the user-written data will be shifted and stored in DHR (data holding register, internal non-memory-mapped register). The data in the DACx_DHR register is then loaded into the DOR register via software trigger or internal or external trigger.

Figure 143 DAC Data Alignment



24.4.4 DAC conversion

The DACx_DOR register cannot be written directly by software. Any data to be transmitted to a DAC channel must be transmitted by loading the DACx_DHR register (via write operation on the DACx_DHR10R or DACx_DHR10L).

24.4.5 DAC output voltage

The DAC output voltage is converted by a linear converter according to the input digital value to a voltage between 0 and VREF+. The voltage output from a DAC

channel is determined by the following formula:

$$DAC_{output} = VREF \times DOR / 1024$$

24.4.6 DAC trigger selection

When the EN bit in the DACx_CR register is 1, conversions can be triggered by software or by event. The TSEL[2:0] bits in the DACx_CR can be used to select one of eight trigger sources. When the EN bit is 0, the data in the DAC_DHR register will not be transmitted to the DACx_DOR register.

Internal signal names of DAC

Table 99 Internal Signal Names of DAC

Signal name	Trigger source
DACx_TRG1	TMR1_TRGO
DACx_TRG2	TMR2_TRGO
DACx_TRG3	TMR3_TRGO
DACx_TRG4	TMR4_TRGO
DACx_TRG5	EINT_Line0
DACx_TRG6	EINT_Line2
DACx_TRG7	EINT_Line11

DAC trigger source

Table 100 DAC Trigger Sources

TSEL[2:0]	Trigger source
000	SWTRIG
001	DACx_TRG1
010	DACx_TRG2
011	DACx_TRG3
100	DACx_TRG4
101	DACx_TRG5
110	DACx_TRG6
111	DACx_TRG7

Event trigger

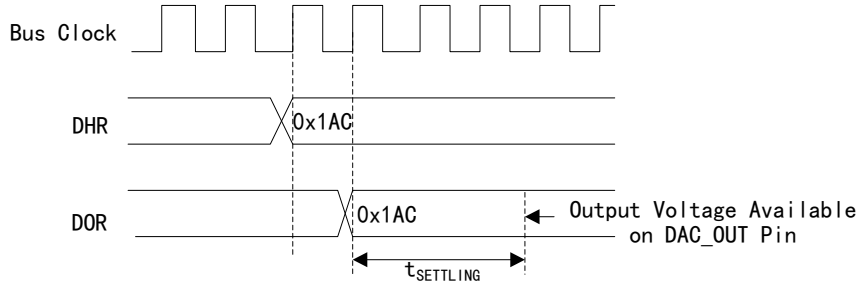
When the DAC interface detects a rising edge of the selected trigger source, the data finally stored in the DACx_DHRregister will be transmitted to the DACx_DOR register. The update time for the DACx_DOR register is one DAC_PCLK cycle after the trigger occurs.

Software trigger

If software trigger is selected, conversion will start as soon as software sets the SWTRIG bit in the DACx_SWTRGR register. The content of the DACx_DHR register will be loaded into the DACx_DOR register. Once it completes loading, the SWTRIG bit will be reset by hardware. The update time for the DACx_DOR register is one DAC_PCLK cycle after the trigger occurs.

Note: The TSEL[2:0] bits can only be modified when the EN bit is 0.

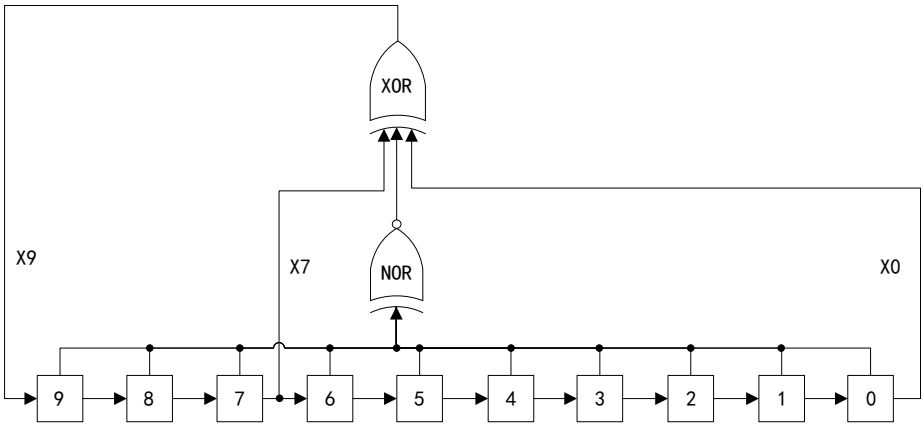
Figure 144 Triggered Conversion Timing Diagram (EN=1)



24.4.7 DAC noise generator

A linear feedback shift register (LFSR) can be used to generate pseudo-random noise with variable amplitude. DAC noise generation is selected by setting WAVE[1:0] to 01. The value preloaded in the LFSR is 0x2AA. This register will be updated one dac_pclk cycle after each trigger event, following a specific calculation algorithm.

Figure 145 DAC Noise Generator



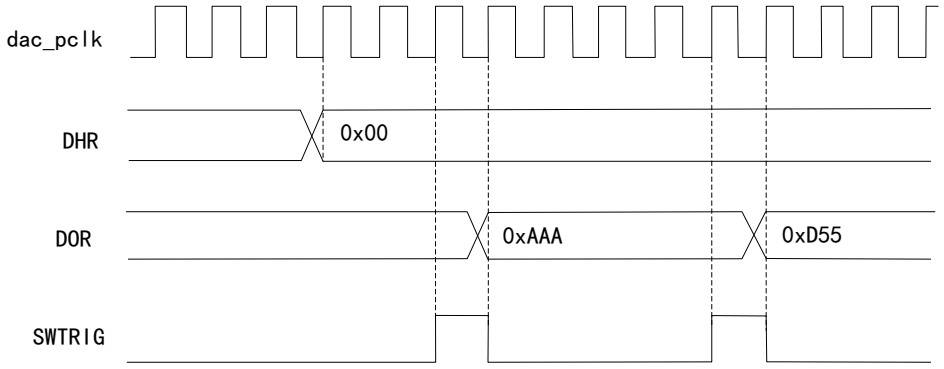
LFSR algorithm

The LFSR value can be partially or fully masked by the MAMP[3:0] bits in the DACx_CR register. This value is added to the value in the DACx_DHR register. Hardware ensures no overflow occurs, and the result will be transmitted to the DACx_DOR register.

If the LFSR is 0x0000, a '1' will be injected (anti-lock mechanism). Resetting the WAVE[1:0] bits in the DACx_CR register can reset the LFSR waveform

generation.

Figure 146 Software-Triggered LFSR Conversion Timing Diagram

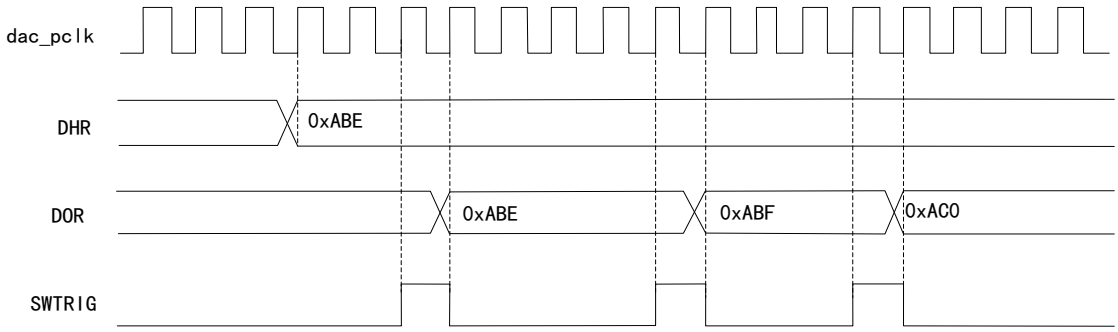


24.4.8 DAC triangular wave generator

A small-amplitude triangular waveform can be added to a DC or slowly varying signal. DAC triangular wave generation is selected by setting WAVE[1:0] in the DACx_CR register to 10. The amplitude is configured by the MAMP[3:0] bits in the DACx_CR register. The internal triangular counter will increment one DAC_PCLK cycle after each trigger event. Then the value of this counter will be added to the DACx_DHR register without overflow, and the result will be transmitted to the DACx_DOR register. It will continue to increment as long as the value of the triangular counter is less than the maximum amplitude defined by MAMP[3:0]. Once the configured amplitude is reached, the counter will decrement to 0 and then increments again, and so on.

Resetting the WAVE[1:0] bits in the DACx_CR register can reset the triangular wave generation.

Figure 147 Software-Triggered Triangular Wave Conversion Timing Diagram



24.4.9 DAC sawtooth wave generator

The DAC can generate sawtooth waveforms. Specific register settings are required to configure the initial value, increment value, and direction control: DAC sawtooth waveform generation is selected by setting WAVE[1:0] in the DACx_CR register to 11.

The initial value (reset value) of the sawtooth wave counter is configured by the

STRSTDATA[9:0] bits in the DACx_STR register. The increment value is defined by the STINCDATA[15:0] bits in the DACx_STR register.

The sawtooth wave direction is defined by the STDIR bit in the DACx_STR register.

The sawtooth wave counter starts from STRSTDATA[9:0] (bits 10 to 15 are set to 000000). Then each increment trigger adds (or subtracts) the value of STINCDATA[15:0].

The DAC output uses the 10 most significant bits (MSBs) of the counter value. The value will saturate when the counter reaches 0x0000 or 0xFFFF. The sawtooth wave reset trigger signal will initialize the counter value to the value of STRSTDATA[9:0] (bits 10 to 15 are set to 000000). The increment and reset trigger must be selected via the STINCTRIGSEL[2:0] and STRSTTRIGSEL[2:0] bits.

The STRSTTRIG signal has higher priority than the STINCTRIG signal. An error will occur if the trigger speed of STRSTTRIG and STINCTRIG exceeds the allowed DACx_DOR data update rate.

It only supports external mode, with data rate less than 200 kHz; in internal mode, the data rate is less than 1 MHz.

Figure 148 Software-Triggered Sawtooth Wave Conversion Timing Diagram (TDIR=0)

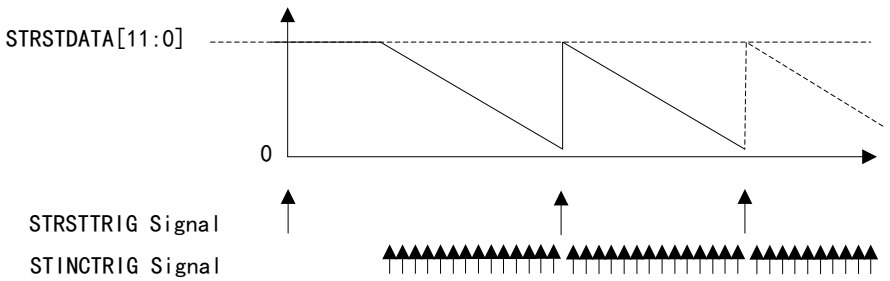


Table 101 Sawtooth Wave Increment Trigger Source

STINCTRIGSEL[2:0]	Trigger source
000	SWTRIG
001	DACx_TRG1
010	DACx_TRG2
011	DACx_TRG3
100	DACx_TRG4
101	DACx_TRG5
110	DACx_TRG6
111	DACx_TRG7

Table 102 Sawtooth Wave Reset Trigger Source

STRSTTRIGSEL[2:0]	Trigger source
000	SWTRIGB
001	DACx_TRG1
010	DACx_TRG2
011	DACx_TRG3
100	DACx_TRG4
101	DACx_TRG5
110	DACx_TRG6
111	DACx_TRG7

24.4.10 Calibration of DAC output buffer

Due to the characteristics of the output buffer, an offset error is introduced on the analog output. Calibration is required to compensate for Vos. The calibration steps are as follows:

Start DAC channel calibration by setting the EN bit in the DACx_CR register to 1. Write the code to the trim_offset[7:0] bits, starting from 0.

Test whether the output voltage is consistent with the expected value. If not, adjust trim_offset[7:0]. When the output voltage is consistent with the expected value, trim_offset[7:0] is the required calibration value.

24.5 Register address mapping

In the following table, all registers of DAC are mapped to a 16-bit addressable (address) space.

Table 103 DAC Register Address Mapping

Register name	Description	Offset address
DACx_CR	Control register	0x00
DACx_SWTRGR	Software trigger register	0x04
DACx_DHR10R	Right-aligned data register	0x08
DACx_DHR10L	Left-aligned data register	0x0C
DACx_DOR	Data output register	0x10
DACx_STR	Sawtooth waveform register	0x14
DACx_STMODR	Sawtooth mode register	0x18
DAC_SR	Status register	0x1C

24.6 Register functional description

24.6.1 Control register (DAC_CR)

Offset address: 0x00

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:15	Reserved		
14	EXT_SEL	R/W	Whether the BUFFER output is connected to the PAD 0: The BUFFER output is not connected to the PAD 1: The BUFFER output is connected to the PAD (external load 50PF) It can be modified only when EN=0.
13:12	EN_RDAC[1:0]	R/W	A switch for the DAC resistor string to output a signal to the operational amplifier input The switch for controlling the DAC resistor string to output a signal to the operational amplifier input is active low. It can be modified only when EN=0.
11:8	MAMP[3:0]	R/W	DAC channel shielding/amplitude select These bits are set by software to select the mask bits in noise generation mode, or select the waveform amplitude in triangular wave generation mode. 0100: Unmask LSFR bit 0 / triangle wave amplitude is 1 0001: Unmask LSFR bits [1:0] / triangle wave amplitude is 3 0010: Unmask LSFR bits [2:0] / triangle wave amplitude is 7 0011: Unmask LSFR bits [3:0] / triangle wave amplitude is 15 0100: Unmask LSFR bits [4:0] / triangle wave amplitude is 31 0101: Unmask LSFR bits [5:0] / triangle wave amplitude is 63 0110: Unmask LSFR bits [6:0] / triangle wave amplitude is 127 0111: Unmask LSFR bits [7:0] / triangle wave amplitude is 255 1000: Unmask LSFR bits [8:0] / triangle wave amplitude is 511 0110: Unmask LSFR bits [9:0] / triangle wave amplitude is 1023 It can be modified only when EN=0.
7:6	WAVE[1:0]	R/W	DAC channel noise/triangular wave generation enable 00: Waveform generator 01: Enable the noise waveform generator 10: Enable the triangular wave generator 11: Enable the sawtooth wave generator It can be modified only when EN=0.
5	Reserved		
4:2	TSEL[2:0]	R/W	DAC channel trigger select 000: SWTRIG 001: DACx_TRG1 010: DACx_TRG2 ... 111: DACx_TRG7 It can be modified only when EN=0.
1	Reserved		
0	EN	R/W	DAC enable 0: Disable 1: Enable

24.6.2 Software trigger register (DAC_SWTRGR)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:17	Reserved		
16	SWTRIGB	W	DAC channel software trigger B This bit is set by software to trigger the DAC in software trigger mode (sawtooth wave generation). It is cleared by hardware. 0: No trigger 1: Trigger the sawtooth wave increment
15:1	Reserved		
0	SWTRIG	W	DAC channel software triggers This bit is set by software to trigger the DAC in software trigger mode. 0: No trigger 1: Trigger Note: This bit will be cleared by hardware once the value of the DAC_DHR register is loaded into the DAC_DOR register (after one DAC_PCLK clock cycle).

24.6.3 Right-aligned data register (DAC_DHR10R)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:10	Reserved		
9:0	DHR10R[9:0]	R/W	DAC channel 10-bit right-aligned data These bits are written by software. They specify the 10-bit data for the DAC channel.

24.6.4 Left-aligned data register (DAC_DHR10L)

Offset address: 0x0C

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15:6	DHR10L[9:0]	R/W	DAC channel 10-bit left-aligned data These bits are written by software. They specify the 10-bit data for the DAC channel.
5:0	Reserved		

24.6.5 Data output register (DAC_DOR)

Offset address: 0x10

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:10	Reserved		
9:0	DOR[9:0]	R	DAC channel data output These bits are read-only. They contain the data output of the DAC channel.

Field	Name	R/W	Description
			External mode: Data rate < 200 KHZ Internal mode: Data rate < 1 MHZ

24.6.6 Sawtooth waveform register (DAC_STR)

Offset address: 0x14

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:22	STINCDATA[9:0]	R/W	DAC channel sawtooth wave increment value
21:13	Reserved		
12	STDIR	R/W	DAC channel sawtooth wave direction set This bit is written by software to select the sawtooth wave step direction. 0: Decrement 1: Increment
11:10	Reserved		
9:0	STRSTDATA[9:0]	R/W	DAC channel sawtooth wave reset value

24.6.7 Sawtooth mode register (DAC_STMODR)

Offset address: 0x18

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:11	Reserved		
10:8	STINCTRIGSEL[2:0]	R/W	DAC channel sawtooth wave incremental trigger select 000: SWTRIGB 001: DACx_TRG1 .... 111: DACx_TRG7 It can be modified only when EN=0.
7:3	Reserved		
2:0	STRSTTRIGSEL[2:0]	R/W	DAC channel sawtooth wave reset trigger select 000: SWTRIG 001: DACx_TRG1 .... 111: DACx_TRG7 The mapping is the same as TSEL[3:0]. It can be modified only when EN=0.

24.6.8 Status register (DAC_SR)

Offset address: 0x1C

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:2	Reserved		
1	UP_OW_FLAG	RC_W1	DOR data overflow flag 0: No data overflow occurs 1: Data overflow occurs
0	DOWN_OW_FLAG	RC_W1	DOR data underflow flag

Field	Name	R/W	Description
			0: No data underflow occurs 1: Data underflow occurs

25 Temperature Sensor (TS)

25.1 Introduction

The chip has a built-in temperature sensor module. Data generation can be configured at different rates. Once the data is generated, an interrupt can be triggered.

25.2 Main characteristics

(1) After it is enabled, the hardware automatically performs conversion

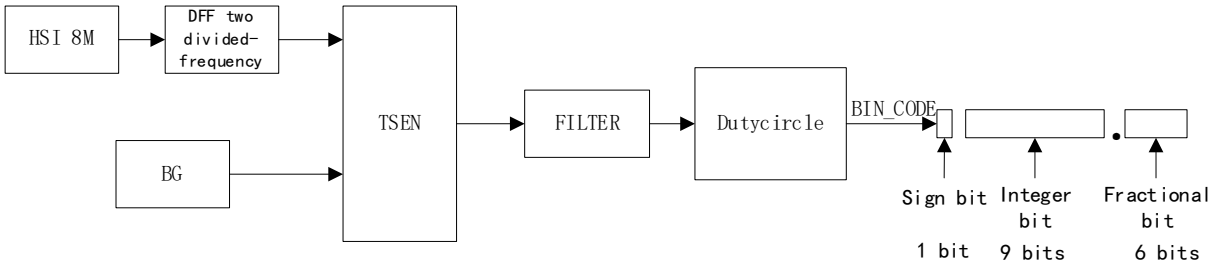
(2) The sampling rate is configurable

(3) 16-bit data
- The most significant bit is the sign bit
- 9 integer bits
- 6 fractional bits

(4) Supports interrupt generation after conversion is completed.

25.3 Structure block diagram

Figure 149 TS Structure Block Diagram



25.4 Functional description

After software sets the TS_EN bit in the TS_CR register to 1, the temperature sensor module is activated. After that, the hardware will automatically start to convert data. When conversion is completed, the data is stored in the TS_DR register, and the RDY bit in the TS_SR register is set to 1. If the TS_IE bit in the TS_CR register is 1, an interrupt will be triggered.

When the RDY bit in the TS_SR register is 1, after software reads the TS_DR register, the hardware will automatically clear the RDY bit to 0. If the RDY bit remains 1, the value in the TS_DR register will not be updated but will remain the value converted last time. It is recommended to enable data overflow update in the configuration to ensure the value saved in the TS_DR register is the latest

sampling value.

The value after temperature sensor conversion is stored in the DATA[15:0] bits of the TS_DR register. The most significant bit is the sign bit, DATA[14:6] bits are the integer part, and DATA[5:0] are the fractional part.

25.5 Register address mapping

Table 104 TS Register Address Mapping

Register name	Description	Offset address
TS_CR	Control register	0x00
TS_DR	Data register	0x04
TS_SR	Status register	0x08

25.6 Register functional description

25.6.1 Control register (TS_CR)

Offset address: 0x00

Reset value: 0x0000 0004

Field	Name	R/W	Description
31:18	Reserved		
17	OVRIE	R/W	TS module data overflow interrupt enable 0: Generate no interrupt when data overflow occurs 1: Generate interrupts when data overflow occurs
16	TS_IE	R/W	TS module interrupt enable 0: Disable 1: Enable
15:5	Reserved		
4	OVRMOD	R/W	Data overflow control 0: When data overflow occurs, the TS_DR register retains the original value 1: When data overflow occurs, the TS_DR register overwrites the original value with the new value.
3	Reserved		
2:1	OSR_SEL	R/W	Sampling rate configure 00: 128 01: 256 10: 512 11: 1024 Note: The value corresponding to the configuration is the sampling period value.
0	TS_EN	R/W	TS module enable 0: Disable 1: Enable

25.6.2 Data register (TS_DR)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:16	Reserved		
15:0	DATA	R	Sample data The most significant bit of sign bit + 9-bit integer part + 6-bit fractional part

25.6.3 Status register (TS_SR)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:2	Reserved		
1	OVR	RC_W1	Overflow flag When the RDY flag bit is 1 and new data is generated, this bit will be set to 1 by hardware. Writing 1 to this bit by software can clear it to 0. When this bit is 1. 0: No data overflow 1: Data overflow occurs
0	RDY	RC_W1	Sampling data valid flag It is set to 1 by hardware when the sampling data is ready. This bit will be automatically cleared to 0 by hardware after the TS_DR register is read.

26 Comparator (COMP)

26.1 Full Name and Abbreviation Description of Terms

Table 105 Full Name and Abbreviation Description of Terms

Full name in English	English abbreviation
Comparator	COMP
Invert	INV
Hysteresis	HYS
Input Plus	INP
Input Minus	INM

26.2 Introduction

Four independent general-purpose comparators (COMP1, COMP2, COMP3 and COMP4) are embedded in MCU, and they can be used in combination with the timers.

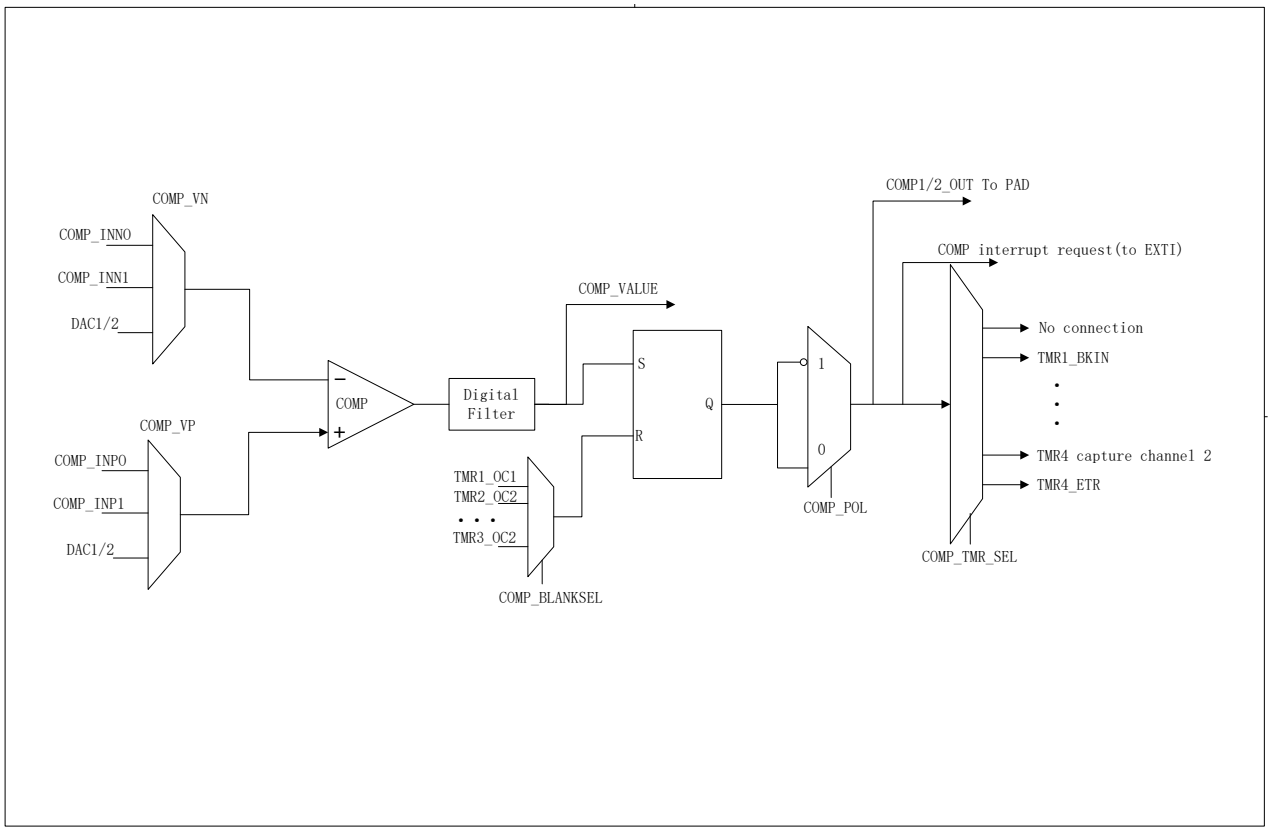
26.3 Main characteristics

The COMP module supports the following functions:

- Each comparator has configurable positive and negative inputs for flexible voltage selection;
  - Select the output voltage of GPIO
  - Internal input can select the output voltage of DAC1 or DAC2
- Output polarity is configurable
- The comparator outputs can be connected to I/O or to 16-bit internal timers for counting
- The comparator outputs are internally connected to the extended interrupt and event controller. Each comparator has its own EINT line that can generate interrupt events
- Output blanking to resist switching noise.

26.4 Structure block diagram

Figure 150 COMP Structure Block Diagram



26.5 Functional description

The G32R430 has four built-in analog comparators.

These comparators can be used for various functions, including analog signal conditioning, and cycle-by-cycle current control loops when used with PWM outputs of the timer.

26.5.1 COMP pins

The multiplexing input and output pins of the comparators are listed below:

Table 106 Comparator Pins

	COMP1	COMP2	COMP3	COMP4
Negative input	PD11/PD12	PD9/PD10	PB3/PB2	PB6/PB8
Positive input	PD12/PD11	PD10/PD9	PB2/PB3	PB7/PB9
Output	PB6	PB9	/	/

26.5.2 COMP clock and reset

The four COMP share a clock enable control bit. The clock is synchronized with

PCLK. Configuring the COMPCEN bit in the RCM_APBCG register can control the COMP clock enable or disable.

The COMP can be reset by a system reset or by configuring the COMPRST bit in the RCM_APBRST register.

26.5.3 COMP write protection

Comparators can be used for safety purposes, such as overcurrent or thermal protection. For applications with specific functional safety requirements, it shall be ensured that the comparator configuration cannot be changed in the event of dummy register access or damage to program counter. The comparator control and status registers can be write-protected.

Once programming is completed, the LOCK bit in the COMP_CxCSR register can be set. This will cause the entire register to become read-only, including the LOCK bit. Write protection can only be removed by reset.

26.5.4 COMP input

When serving as comparator input, GPIO is required to configure as analog mode.

COMP input consists of non-inverting input and inverting input. Input connection can select IO pins or the output voltage of internal DAC. The input voltage is selected by configuring the VP_SEL and VN_SEL bits in the COMP_CxCSR register. Internal connections can select the output voltage of DAC1 or DAC2.

26.5.5 COMP output

The comparator output can be connected to an external IO port, or connected to internal timer signals: the timer's input capture channel.

The output connection can be modified by programming the COMP_TMR_SEL[3:0] bits in the COMP_CxCSR register.

The output polarity can be modified by programming the POL bit of the COMP_CxCSR register

26.5.6 COMP hysteresis

The comparator includes a programmable hysteresis function to avoid false output transitions when the input signal has large noise. This hysteresis function is asymmetric and only acts on the falling edge of the comparator output. The internal hysteresis function can be disabled, allowing the hysteresis amount to be set by external components.

The output polarity can be modified by programming the POL bit of the COMP_CxCSR register

26.5.7 COMP output blanking

The purpose of blanking function is to prevent current regulation failure due to brief current spikes at the beginning of the PWM cycle (usually the recovery current in the anti-parallel diode of power switches). This is achieved by setting a blanking window defined by the timer output compare signal. The blanking source is selected for each comparator channel by software via the BLANKSEL[2:0] bit in the corresponding COMP_CxCSR register. Logical AND operation is performed on the inverted blanking signal and comparator-level output to generate the output of comparator channel x.

26.5.8 COMP interrupt

The comparator output is internally connected to the external interrupt/event controller (EINT). If the external interrupt is configured correctly, an EINT interrupt can be generated or MCU entering the STOP mode can be awakened.

26.6 Register address mapping

Table 107 COMP Register Address Mapping

Register name	Description	Offset address
COMP_C1CSR	COMP1 control status register	0x00
COMP_C2CSR	COMP2 control status register	0x04
COMP_C3CSR	COMP3 control status register	0x08
COMP_C4CSR	COMP4 control status register	0x0C

26.7 Register functional description

26.7.1 COMP1 control status register (COMP_C1CSR)

Offset address: 0x00

Reset value: 0x0000 0000

Field	Name	R/W	Description
31	LOCK	R/W	Lock comparator 1 control status register This bit is set by software and cleared by hardware system reset. It locks all content of the comparator 1 control status register COMP_C1CSR[31:0]. When locked, all control bits and flag bits are read-only. When unlocked, control bits can also be written by software. 0: Unlock 1: Locked
30	VALUE	R	Polarity selector and the level value before shielding This read-only flag reflects the level value of the comparator 1 output before the polarity selector and masking.
29:21	Reserved		
20:18	BLANKSEL	R/W	Comparator channel 1 mask signal select

Field	Name	R/W	Description
			This bit is controlled by software to select the mask signal for comparator channel 1. 000: TMR1_OC1 001: TMR2_OC2 010: TMR3_OC3 011: TMR4_OC4 100: TMR1_OC4 110: TMR2_OC1 111: TMR3_OC2
17:16	HYS	R/W	Comparator 1 hysteresis function enable 00: No hysteresis 01: Hysteresis voltage is 22 mV 10: Hysteresis voltage is 45 mV 11: Hysteresis voltage is 67 mV
15	POL	R/W	Comparator 1 output polarity select This bit is controlled by software. 0: Phase not reversed 1: Inverting
14:12	Reserved		
11:8	COMP_TMR_SEL	R/W	Output connection 0000: No connection 0001: TMR1_BKIN 0010: TMR1 capture channel 1 0011: TMR1 capture channel 2 0100: TMR1_ETR 0101: TMR2 capture channel 1 0110: TMR2 capture channel 2 0111: TMR2 capture channel 3 1000: TMR2 capture channel 4 1001: TMR2_ETR 1010: TMR3 capture channel 1 1011: TMR3 capture channel 2 1100: TMR3_ETR 1101: TMR4 capture channel 1 1110: TMR4 capture channel 2 1111: TMR4_ETR
7:6	VP_SEL	R/W	Comparator 1 positive input voltage select 00: Select the voltage of INP0 on COMP1 01: Select the voltage of INP1 on COMP1 10: Select the output voltage of DAC2 11: Reserved
5:4	VN_SEL	R/W	Comparator 1 negative terminal input voltage select 00: Select the voltage of INN0 on COMP1 01: Select the voltage of INN1 on COMP1 10: Select the output voltage of DAC2 11: Reserved
3:1	FILTER	R/W	Comparator 1 digital filter time select

Field	Name	R/W	Description
			000: No filtering 001: 2 x HSI CLK 010: 4 x HSI CLK 011: 6 x HSI CLK 100: 12 x HSI CLK 101: 18 x HSI CLK 110: 24 x HSI CLK 111: 30 x HSI CLK
0	EN	R/W	Comparator 1 enable This bit is controlled by software. 0: Disable 1: Enable

26.7.2 COMP2 control status register (COMP_C2CSR)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
31	LOCK	R/W	Lock comparator 2 control status register This bit is set by software and cleared by hardware system reset. It locks all content of the comparator 2 control status register COMP_C2CSR[31:0]. When locked, all control bits and flag bits are read-only. When unlocked, control bits can also be written by software. 0: Unlock 1: Locked
30	VALUE	R	Polarity selector and the level value before shielding This read-only flag reflects the level value of the comparator 2 output before the polarity selector and masking.
29:21	Reserved		
20:18	BLANKSEL	R/W	Comparator channel 2 mask signal select This bit is controlled by software to select the mask signal for comparator channel 2. 000: TMR1_OC1 001: TMR2_OC2 010: TMR3_OC3 011: TMR4_OC4 100: TMR1_OC4 110: TMR2_OC1 111: TMR3_OC2
17:16	HYS	R/W	Comparator 2 hysteresis function enable 00: No hysteresis 01: Hysteresis voltage is 22 mV 10: Hysteresis voltage is 45 mV 11: Hysteresis voltage is 67 mV
15	POL	R/W	Comparator 2 output polarity select This bit is controlled by software. 0: Phase not reversed

Field	Name	R/W	Description
			1: Inverting
14:12	Reserved		
11:8	COMP_TMR_SEL	R/W	Output connection 0000: No connection 0001: TMR1_BKIN 0010: TMR1 capture channel 1 0011: TMR1 capture channel 2 0100: TMR1_ETR 0101: TMR2 capture channel 1 0110: TMR2 capture channel 2 0111: TMR2 capture channel 3 1000: TMR2 capture channel 4 1001: TMR2_ETR 1010: TMR3 capture channel 1 1011: TMR3 capture channel 2 1100: TMR3_ETR 1101: TMR4 capture channel 1 1110: TMR4 capture channel 2 1111: TMR4_ETR
7:6	VP_SEL	R/W	Comparator 2 positive input voltage select 00: Select the voltage of INP0 on COMP2 01: Select the voltage of INP1 on COMP2 10: Select the output voltage of DAC2 11: Reserved
5:4	VN_SEL	R/W	Comparator 2 negative terminal input voltage select 00: Select the voltage of INN0 on COMP2 01: Select the voltage of INN1 on COMP2 10: Select the output voltage of DAC2 11: Reserved
3:1	FILTER	R/W	Comparator 2 digital filter time select 000: No filtering 001: 2 x HSI CLK 010: 4 x HSI CLK 011: 6 x HSI CLK 100: 12 x HSI CLK 101: 18 x HSI CLK 110: 24 x HSI CLK 111: 30 x HSI CLK
0	EN	R/W	Comparator 2 enable This bit is controlled by software. 0: Disable 1: Enable

26.7.3 COMP3 control status register (COMP_C3CSR)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
31	LOCK	R/W	Lock comparator 3 control status register This bit is set by software and cleared by hardware system reset. It locks all content of the comparator 3 control status register COMP_C3CSR[31:0]. When locked, all control bits and flag bits are read-only. When unlocked, control bits can also be written by software. 0: Unlock 1: Locked
30	VALUE	R	Polarity selector and the level value before shielding This read-only flag reflects the level value of the comparator 3 output before the polarity selector and masking.
29:21	Reserved		
20:18	BLANKSEL	R/W	Comparator channel 3 mask signal select This bit is controlled by software to select the mask signal for comparator channel 3. 000: TMR1_OC1 001: TMR2_OC2 010: TMR3_OC3 011: TMR4_OC4 100: TMR1_OC4 110: TMR2_OC1 111: TMR3_OC2
17:16	HYS	R/W	Comparator 3 hysteresis function enable 00: No hysteresis 01: Hysteresis voltage is 22 mV 10: Hysteresis voltage is 45 mV 11: Hysteresis voltage is 67 mV
15	POL	R/W	Comparator 3 output polarity select This bit is controlled by software. 0: Phase not reversed 1: Inverting
14:12	Reserved		
11:8	COMP_TMR_SEL	R/W	Output connection 0000: No connection 0001: TMR1_BKIN 0010: TMR1 capture channel 1 0011: TMR1 capture channel 2 0100: TMR1_ETR 0101: TMR2 capture channel 1 0110: TMR2 capture channel 2 0111: TMR2 capture channel 3 1000: TMR2 capture channel 4 1001: TMR2_ETR 1010: TMR3 capture channel 1 1011: TMR3 capture channel 2 1100: TMR3_ETR 1101: TMR4 capture channel 1 1110: TMR4 capture channel 2

Field	Name	R/W	Description
			1111: TMR4_ETR
7:6	VP_SEL	R/W	Comparator 3 positive input voltage select 00: Select the voltage of INP0 on COMP3 01: Select the voltage of INP1 on COMP3 10: Select the output voltage of DAC1 11: Reserved
5:4	VN_SEL	R/W	Comparator 3 negative terminal input voltage select 00: Select the voltage of INN0 on COMP3 01: Select the voltage of INN1 on COMP3 10: Select the output voltage of DAC1 11: Reserved
3:1	FILTER	R/W	Comparator 3 digital filter time select 000: No filtering 001: 2 x HSI CLK 010: 4 x HSI CLK 011: 6 x HSI CLK 100: 12 x HSI CLK 101: 18 x HSI CLK 110: 24 x HSI CLK 111: 30 x HSI CLK
0	EN	R/W	Comparator 3 enable This bit is controlled by software. 0: Disable 1: Enable

26.7.4 COMP4 control status register (COMP_C4CSR)

Offset address: 0x0C

Reset value: 0x0000 0000

Field	Name	R/W	Description
31	LOCK	R/W	Lock comparator 4 control status register This bit is set by software and cleared by hardware system reset. It locks all content of the comparator 4 control status register COMP_C4CSR[31:0]. When locked, all control bits and flag bits are read-only. When unlocked, control bits can also be written by software. 0: Unlock 1: Locked
30	VALUE	R	Polarity selector and the level value before shielding This read-only flag reflects the level value of the comparator 4 output before the polarity selector and masking.
29:21	Reserved		
20:18	BLANKSEL	R/W	Comparator channel 4 mask signal select This bit is controlled by software to select the mask signal for comparator channel 4. 000: TMR1_OC1 001: TMR2_OC2 010: TMR3_OC3

Field	Name	R/W	Description
			011: TMR4_OC4 100: TMR1_OC4 110: TMR2_OC1 111: TMR3_OC2
17:16	HYS	R/W	Comparator 4 hysteresis function enable 00: No hysteresis 01: Hysteresis voltage is 22 mV 10: Hysteresis voltage is 45 mV 11: Hysteresis voltage is 67 mV
15	POL	R/W	Comparator 4 output polarity select This bit is controlled by software. 0: Phase not reversed 1: Inverting
14:12	Reserved		
11:8	COMP_TMR_SEL	R/W	Output connection 0000: No connection 0001: TMR1_BKIN 0010: TMR1 capture channel 1 0011: TMR1 capture channel 2 0100: TMR1_ETR 0101: TMR2 capture channel 1 0110: TMR2 capture channel 2 0111: TMR2 capture channel 3 1000: TMR2 capture channel 4 1001: TMR2_ETR 1010: TMR3 capture channel 1 1011: TMR3 capture channel 2 1100: TMR3_ETR 1101: TMR4 capture channel 1 1110: TMR4 capture channel 2 1111: TMR4_ETR
7:6	VP_SEL	R/W	Comparator 4 positive input voltage select 00: Select the voltage of INP0 on COMP4 01: Select the voltage of INP1 on COMP4 10: Select the output voltage of DAC1 11: Reserved
5:4	VN_SEL	R/W	Comparator 4 negative terminal input voltage select 00: Select the voltage of INN0 on COMP4 01: Select the voltage of INN1 on COMP4 10: Select the output voltage of DAC1 11: Reserved
3:1	FILTER	R/W	Comparator 4 digital filter time select 000: No filtering 001: 2 x HSI CLK 010: 4 x HSI CLK 011: 6 x HSI CLK 100: 12 x HSI CLK

Field	Name	R/W	Description
			101: 18 x HSI CLK 110: 24 x HSI CLK 111: 30 x HSI CLK
0	EN	R/W	Comparator 4 enable This bit is controlled by software. 0: Disable 1: Enable

27 Chip Electronic Signature

27.1 Introduction

The chip's electronic signature is stored in the system memory area and can be read by the CPU or emulator. The electronic signature includes identification and calibration data written before leaving the factory. Such content can be used by user code or other external devices to identify the characteristics of the chip.

27.2 Functional description

27.2.1 96-bit unique ID

The 96-bit unique ID can be applied in the following situations:

- Used as a serial number, read or used by software
- Combined with the software encryption source text or protocol before programming the internal Flash, serving as part of a security key to enhance the security of the code in FLASH.
- Activate secure bootstrap process, etc.

27.3 Register functional description

The 96-bit unique ID provides a reference number that is unique to any chip and in any context. These bit fields are read-only for users and cannot be modified. The base address of this module is 0x0802 020C.

27.3.1 96-bit unique chip ID of unique device

Offset address: 0x00

Reset value: 0xXXXX XXXX

Field	Name	R/W	Description
31:0	U_ID[31:0]	R	Represents the X and Y coordinates on the wafer in BCD format

Offset address: 0x04

Reset value: 0xXXXX XXXX

Field	Name	R/W	Description
31:8	UID[63:40]	R	Represents the lower 24 bits of Lot number in ASCII code (LOT_NUM[23:0])
7:0	UID[39:32]	R	Represents Wafer number as an 8-bit unsigned number (WAFE_NUM[7:0])

Offset address: 0x08

Reset value: 0xXXXX XXXX

Field	Name	R/W	Description
31:0	U_ID[95:64]	R	Represents the higher 32 bits of Lot number in ASCII code (LOT_NUM[55:24])

27.3.2 Main Memory Area Capacity Register

This register is used to configure the flash memory capacity and the product ID.

Base address: 0x0802 0208

Offset address: 0x00

Reset value: 0xXXXX XXXX

Field	Name	R/W	Description
31:16	Reserved		
15:0	FLASH_SIZE [15:0]	R	FLASH capacity value, expressed in KB. For example, 0x0080 represents 128KB.

28 Revision History

Table 108 Document Revision History

Date	Version	Revision History
April 2026	1.0	• Initial version

Statement

This manual is formulated and published by Zhuhai Geehy Semiconductor Co., Ltd. (hereinafter referred to as "Geehy"). The contents in this manual are protected by laws and regulations of trademark, copyright and software copyright. Geehy reserves the right to correct and modify this manual at any time. Please read this manual carefully before using the product. Once you use the product, it means that you (hereinafter referred to as the "users") have known and accepted all the contents of this manual. Users shall use the product in accordance with relevant laws and regulations and the requirements of this manual.

1. Ownership of rights

This manual can only be used in combination with chip products and software products of corresponding models provided by Geehy. Without the prior permission of Geehy, no unit or individual may copy, transcribe, modify, edit or disseminate all or part of the contents of this manual for any reason or in any form.

The "Geehy" or "Geehy" words or graphics with "®" or "TM" in this manual are trademarks of Geehy. Other product or service names displayed on Geehy products are the property of their respective owners.

2. No intellectual property license

Geehy owns all rights, ownership and intellectual property rights involved in this manual.

Geehy shall not be deemed to grant the license or right of any intellectual property to users explicitly or implicitly due to the sale and distribution of Geehy products and this manual.

If any third party’s products, services or intellectual property are involved in this manual, Geehy shall not be deemed to authorize users to use the aforesaid third party’s products, services or intellectual property, nor shall it be deemed to provide any form of guarantee for third-party products, services, or intellectual property, including but not limited to any non-infringement guarantee for third-party intellectual property, unless otherwise agreed in sales order or sales contract of Geehy.

3. Version update

Users can obtain the latest manual of the corresponding products when ordering Geehy products.

If the contents in this manual are inconsistent with Geehy products, the agreement in Geehy sales order or sales contract shall prevail.

4. Information reliability

The relevant data in this manual are obtained from batch test by Geehy Laboratory or cooperative third-party testing organization. However, clerical errors in correction or errors caused by differences in testing environment are unavoidable. Therefore, users should understand that Geehy does not bear any responsibility for such errors that may occur in this manual. The relevant data in this manual are only used to guide users as performance parameter reference and do not constitute Geehy's guarantee for any product performance.

Users shall select appropriate Geehy products according to their own needs, and effectively verify and test the applicability of Geehy products to confirm that Geehy products meet their own needs, corresponding standards, safety or other reliability requirements. If losses are caused to users due to the user's failure to fully verify and test Geehy products, Geehy will not bear any responsibility.

5. Compliance requirements

Users shall abide by all applicable local laws and regulations when using this manual and the matching Geehy products. Users shall understand that the products may be restricted by the export, re-export or other laws of the countries of the product suppliers, Geehy, Geehy distributors and users. Users (on behalf of itself, subsidiaries and affiliated enterprises) shall agree and undertake to abide by all applicable laws and regulations on the export and re-export of Geehy products and/or technologies and direct products.

6. Disclaimer

This manual is provided by Geehy on an "as is" basis. To the extent permitted by applicable laws, Geehy does not provide any form of express or implied warranty, including without limitation the warranty of product merchantability and applicability of specific purposes.

Geehy products are not designed, authorized, or guaranteed to be suitable for use as critical components in military, life support, pollution control, or hazardous substance management systems, nor are they designed, authorized, or guaranteed to be suitable for applications that may cause injury, death, property, or environmental damage in case of product failure or malfunction.

If the product is not labeled as "Automotive grade", it means it is not suitable for automotive applications. If the user's application of the product is beyond the specifications, application fields, and standards provided by Geehy, Geehy will assume no responsibility.

Users shall ensure that their application of the product complies with relevant standards, and the requirements of functional safety, information security, and environmental standards. Users are fully responsible for their selection and use of Geehy products. Geehy will bear no responsibility for any disputes arising from the subsequent design and use of Geehy products by users.

7. Limitation of liability

In any case, unless required by applicable laws or agreed in writing, Geehy and/or any third party providing this manual and the products on an "as is" basis shall not be liable for damages, including any general or special direct, indirect or collateral damages arising from the use or no use of this manual and the products (including without limitation data loss or inaccuracy, or losses suffered by users or third parties), which cover damage to personal safety, property, or environment, for which Geehy will not be responsible.

8. Scope of application

The information in this manual replaces the information provided in all previous versions of the manual.