User Manual

APM32F035x8

Arm® Cortex® -M0+ based 32-bit MCU

Version: V 0.3

Contents

1 Introduction and document description rules

1.1 Introduction

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

For information about Arm® Cortex® -M0+ core, please refer to Arm® Cortex®-M0+ 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 data manual 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 of terms

Table 1 R/W Abbreviation and Description

R/W	Description	Abbreviations
read/write	The software can read and write this bit.	R/W
read-only	The software can only read this bit.	R
write-only	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

R/W	Description	Abbreviations
read/clear	The software can read this bit and clear it by writing 0. Writing 1 has no effect on this bit.	RC_W0
read/clear by read	The software can read this bit, 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 flip this bit only by writing 1, and writing 0 has no effect on this bit.	T

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
Software	SW
Hardware	HW
Source	SRC
System	SYS
Peripheral	PER

Full name in English	English abbreviation
Address	ADDR
Direction	DIR
Clock	CLK
Input	I
Output	O
Interrupt	INT
Data	DATA
Size	SIZE
Divider	DIV
Prescaler	PSC
Multiplier	MUL
Period	PRD
Cipher	CIPH

Table 3 Full Name and Abbreviation of Modules

Full name in English	English abbreviation
Reset and Clock Management	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
Real-Time Clock	RTC
Universal Synchronous Asynchronous Receiver Transmitter	USART
Inter-integrated Circuit Interface	I2C
Serial Peripheral Interface	SPI
Inter-IC Sound Interface	I2S

Full name in English	English abbreviation
Analog-to-Digital Converter	ADC
Cyclic Redundancy Check Calculation Unit	CRC

2 System architecture

2.1 Full name and abbreviation of terms

Table 4 Full Name and Abbreviation of Terms

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

2.2 System architecture block diagram

The main system mainly consists of two master modules and four slave modules. The main modules are Arm® Cortex® -M0+ core and general-purpose DMA. The slave modules are internal SRAM, internal flash memory FLASH, AHB2 bus connecting all GPIO ports, and AHB1/APB bridges on AHB1 bus, among which, AHB1/APB bridges connect all peripherals.

These are connected through a multi-level AHB bus architecture, as shown in the figure below:

Figure 1 APM32F035x8T7 System Architecture Block Diagram

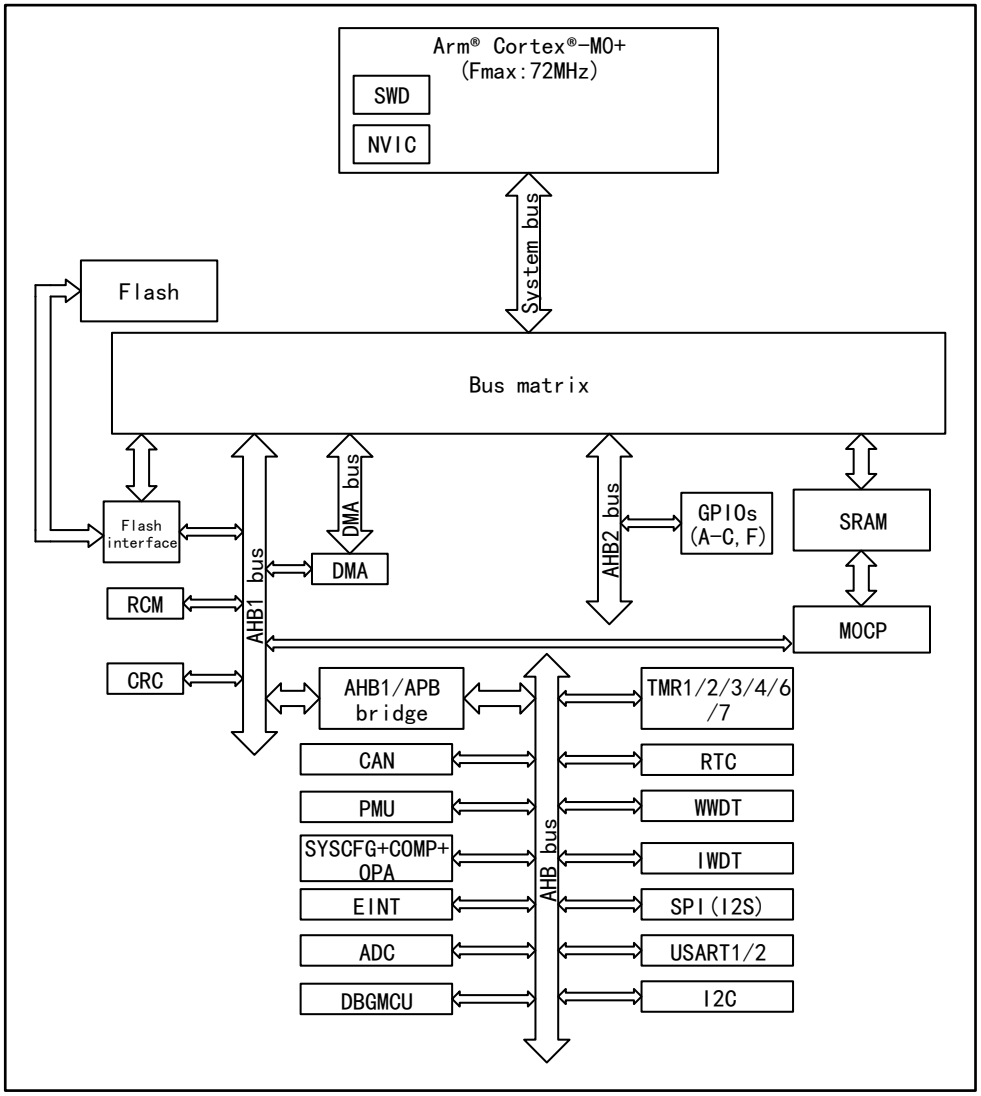


Table 5 Bus Name

Name	Description
System bus	Connect the system bus (peripheral bus) of Arm® Cortex®-M0 core and the bus matrix.
DMA bus	Connect AHB master control interface of DMA and the bus matrix.
Bus matrix	Coordinate the access of the core and DMA; consist of CPU AHB, system bus, DMA bus and FMC, SRAM, AHB2 and AHB1/APB bridges. AHB peripheral is connected with the system bus through the bus matrix and is allowed to access DMA.
AHB/APB bridge	The bridge provides synchronous connection between AHB and APB buses. The non-32-bit access to APB register will be converted into 32 bits automatically.

2.3 Memory mapping

The memory mapping address is totally 4GB address. The assigned addresses include the core (including core peripherals), on-chip Flash (including main

memory area, system memory area and option bytes), on-chip SRAM, and bus peripherals (including AHB and APB peripherals). Please refer to the data manual of the corresponding model for specific information of various addresses.

2.3.1 Embedded SRAM

Built-in static SRAM. It allows access by byte, half word (16 bits) or full word (32 bits). The start address of SRAM is 0x2000 0000.

SRAM does not support parity check.

2.4 Startup configuration

APM32F MCU series realizes a special mechanism. By configuring the BOOT pin parameter and the nBOOT1 bit in FMC_OBCS, there are three different startup modes, namely, the system can not only start from Flash memory or system memory, but also start from the built-in SRAM. The memory selected as the start zone is determined by the selected startup mode.

Table 6 Startup Mode Configuration and Access Mode

Startup mode selection pin		**Startup mode**	**Access methods**
BOOT1	BOOT0		
X	0	Main flash memory（Flash）	The main flash memory is mapped to the boot space, but it can still be accessed at its original address, that is, the contents of the flash memory can be accessed in two address areas.
0	1	System memory	The system memory is mapped to the boot space (0x0000 0000), but it can still be accessed at its original address.
1	1	Built-in SRAM	SRAM can be accessed only at the starting address.

Note:

（1） The boot space address is 0x0000 0000

（2） The original address of Flash is 0x0800 0000

（3） The original address of the system memory is 0x1FFF EC00

（4） The start address of SRAM is 0x2000 0000

（5） The value of BOOT1 is negation of nBOOT1 option bit

Users can select the startup mode after reset by setting the status of BOOT1 (configuring nBOOT1) and BOOT0 pins.

BOOT pin should keep the startup configuration required by user in standby mode. When exiting the standby mode, the value of boot pin will be latched.

If you choose to start from built-in SRAM, you must use NVIC's exception table and offset register to remap the vector table to SRAM when writing the application code.

Embedded startup program

The embedded startup program is written on the production line by Geehy and stored in the system memory area.

3 M0CP

3.1 Full name and abbreviation of terms

Table 7 Full Name and Abbreviation Description of Terms

Full name in English	term
Software/Hardware/Firmware	SW/HW/FW
Interface	I/F
Coprocessor for the ARM Cortex-M0+	M0CP
Program memory or instruction memory	PM
Data memory	DM
Master bus I/F for instruction or program memory accessing	IBUS
Master bus I/F for data memory accessing	DBUS
AHB slave interface	ahbs
Hardware-oriented, pre-synthesis user-defined	configurable
Software-oriented, post-synthesis user-programmed through SW I/F	programmable

3.2 Introduction

The coprocessor includes hardware division, hardware extraction of square root, trigonometric function generation, and generation of five-segment and seven-segment SVPWM.

3.3 Main characteristics

（1） 1 configurable RV32ECM core. PM is up to 8 KB.

（1） One 32-bit AHB slave interface is used for register configuration. It only supports 32-bit data aligned access.

（2） One 32-bit AHB master interface is used to access PM and DM. The response to the IBUS master interface must be zero wait.

（3） A tightly coupled interface is used to access the PWM module corresponding to the SVPWM algorithm.

（4） 1 TX interrupt (an interrupt is generated at the end of calculation).

（5） Two sets of parameter/result registers; the switching mode of the register set is programmable and can be realized by hardware or software.

（6） Support Arm® Cortex®-M0+ and TXEV interface communication.

（7） Programmable data format, Q0 to Q15.

（8） Support and optimize the CORDIC algorithm and realize acceleration of some trigonometric functions. Support rotation mode and vector mode.

（9） Support and optimize SVPWM algorithm, and support five-segment and seven-segment algorithms.

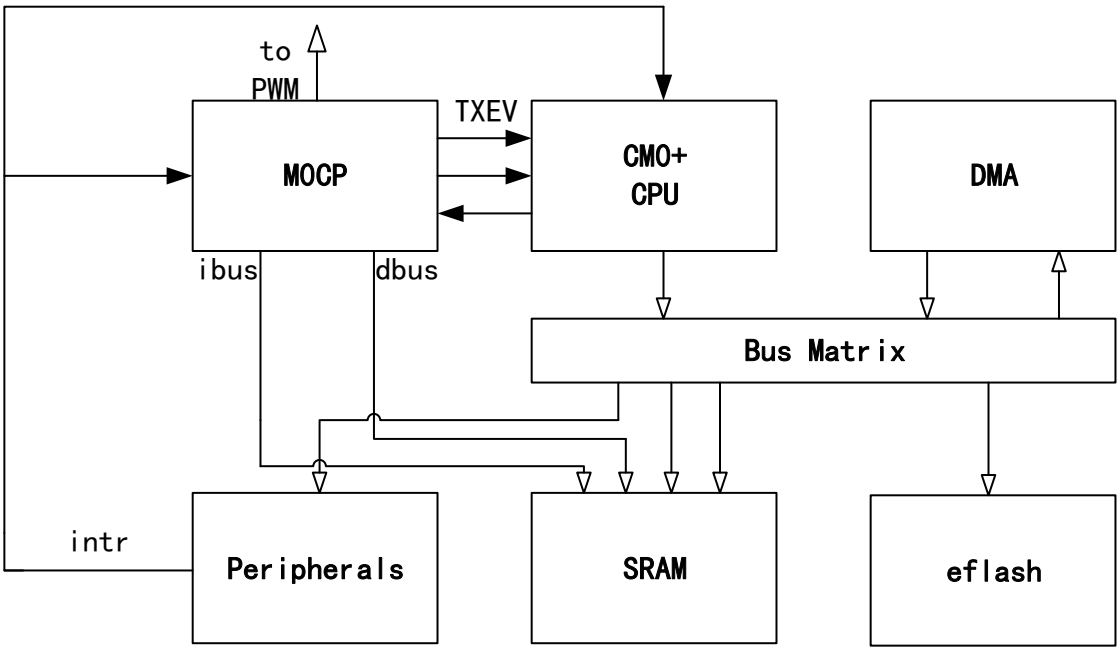
（10） Support and optimize fixed-point square root and division (32/32 bits).

（11） Support atan2 operation.

3.4 Structure block diagram

The coprocessor (M0CP) is designed to accelerate some applications related to FOC (Field Oriented Control) running on Arm Cortex-M0+. The typical integration of M0CP and MCU is shown in the figure below.

Figure 2 Structure Block Diagram



M0CP uses two 32-bit AHB lite master interfaces to access PM and DM respectively. The M0CP can be configured through a 32-bit AHB slave interface, communicate with the processor through TXEV, and directly update the data required by the PWM module through the tightly coupled handshake protocol interface.

3.5 Functional description

3.5.1 CORDIC algorithm description

CORDIC (coordinate rotation digital computer) is a hardware-efficient iterative method. Through basic addition and shift operations, vector rotation and orientation calculation can be achieved without complex functions. It uses elementary functions with a wide range of rotation calculation and can support both rotation mode and vector mode, as shown in the following table.

Table 8 Rotation Mode and Vector Mode

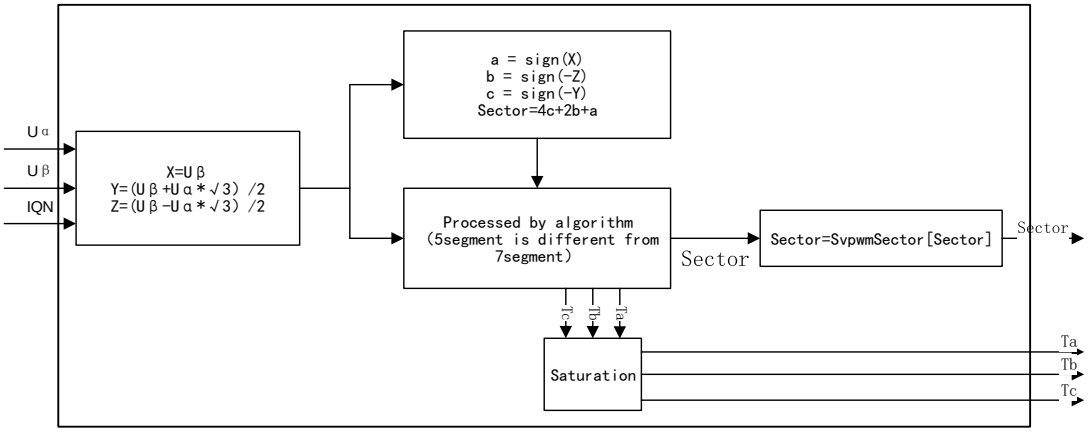
Circular function:$e_i = arctan(2^{-i})$			
Rotation mode	$d_i = sign(z_i), z_i \to 0$	$X_{result} = X\cos(Z) - Y\sin(Z)$ $Y_{result} = X\sin(Z) + Y\cos(Z)$ $Z_{result} = 0$	Set X=1, Y=0 Then $X_{result} = \cos(Z), Y_{result} = \sin(Z)$ Since pre-processing logic is used in CORDIC, X, Y and Z can take values in a full range
Vector mode	$d_i = sign(z_i), z_i \to 0$	$X_{result} = \sqrt{X^2 + Y^2}$ $Y_{result} = 0$ $Z_{result} = Z + \arctan(\frac{Y}{X})$	Set X=X, Y=Y Then the vector amplitude$X_{result} = \sqrt{X^2 + Y^2}$ Set Z=0, then $Z_{result} = Z + \arctan(\frac{Y}{X})$ Since pre-processing and post-processing logic are used in CORDIC, X and Y can take values in a full range ($[-2^{15}, (2^{15} - 1)]$)

Note: The data of operand/result X, Y, Z are stored in X_REG/Y_REG/Z_REG register.

3.5.2 SVPWM algorithm description

SVPWM (Space Vector Pulse Width Modulation) is used to modulate the three-phase voltage required by the motor. The five-segment and seven-segment algorithm flow is shown in the figure below.

Figure 3 Five-segment and Seven-segment Algorithm Flow




```
5-Segment algorithm
Sector=0,7:Ta=Tb=Tc=IQN(0.5);

Sector=1:t1=Z;t2=Y;
         Tc=0;
         Ta=Tc+t2;
         Tb=Ta+t1;

Sector=2:t1=Y;t2=-X;
         Tb=0;
         Tc=Tb+t2;
         Ta=Tc+t1;

Sector=3:t1=-Z;t2=X;
         Tb=0;
         Tc=Tb+t2;
         Ta=Tb+t1;

Sector=4:t1=-X;t2=Z;
         Ta=0;
         Tb=Ta+t2;
         Tc=Tb+t1;

Sector=5:t1=X;t2=-Y;
         Ta=0;
         Tc=Ta+t2;
         Tb=Tc+t1;

Sector=6:t1=-Y;t2=-Z;
         Tb=0;
         Ta=Tb+t2;
         Tc=Ta+t1;
```

```
7-Segment algorithm
Sector=0,7:Ta=Tb=Tc=IQN(0.5);

Sector=1:t1=Z;t2=Y;
         Tc=(IQN(1)-t1-t2)>>1;
         Ta=Tc+t2;
         Tb=Ta+t1;

Sector=2:t1=Y;t2=-X;
         Tb=(IQN(1)-t1-t2)>>1;
         Tc=Tb+t2;
         Ta=Tc+t1;

Sector=3:t1=-Z;t2=X;
         Tb=(IQN(1)-t1-t2)>>1;
         Tc=Tb+t2;
         Ta=Tb+t1;

Sector=4:t1=-X;t2=Z;
         Ta=(IQN(1)-t1-t2)>>1;
         Tb=Ta+t2;
         Tc=Tb+t1;

Sector=5:t1=X;t2=-Y;
         Ta=(IQN(1)-t1-t2)>>1;
         Tc=Ta+t2;
         Tb=Tc+t1;

Sector=6:t1=-Y;t2=-Z;
         Tb=(IQN(1)-t1-t2)>>1;
         Ta=Tb+t2;
         Tc=Ta+t1;
```

```
Note:
IQN: the value of n of Qn, 8-bit
unsigned digit

Uα、Uβ、t1、t2、Ta、Tb、Tc:
16-bit signed number

Multiplication:
Qn form multiplication

SvpwmSector[]:
Array {0,1,5,0,3,2,4,0}

Saturation: x=a, b, c
When Tx<0, Tx=0;
When Tx>IQN(1), Tx=IQN(1)

Sign function: x=sign(y)
When y>0, x=1;
When y<=0, x=0
```

Note: The COUNT mode of SVPWM needs to multiply the output duty cycle by the cycle value of TMR1, so M0CP will always be in the busy state without enabling TMR1. Therefore, TMR1 should be enabled in COUNT mode.

Note: The data of operand/result X, Y, Z , W(Sector) are stored in X_REG/Y_REG/Z_REG/W_REG register.

3.5.3 Division algorithm description

When performing 32bit/32bit division, the register stores data as follows:

- Divisor is stored in X_REG register
- Dividend is stored in Y_REG register
- Quotient is stored in X_REG register（after operation）
- Remainder is stored in Y_REG register（after operation）

CTRL_REG[ALG] is configured to select signed or unsigned division.

For the signed division operation, the remainder rule is described as follows: the sign of the remainder is the same as that of the dividend, as follow:

Remainder= Dividend-Quotient x Divisor

Example1: 20÷6, Quotient is-3, Remainder is-2, instead of Quotient is -4, Remainder is +4

Example2: 20÷(-6), Quotient is-7, Remainder is-1

The configured input data type and the read result data type correspond to the ALG value; That is, when calculating signed division, the input and output are of int32 type. When calculating unsigned division, the input and output are of uint32 type.

There is no order requirement for the configuration of the divisor and the dividend before the division operation is started. After starting division, if the hardware operation is completed, it will automatically set the BUSY bit of the STAT_REG register. At this time, the corresponding registers can be read the operation result, namely quotient and remainder.

3.5.4 I/O register allocation and cycle count

Table 9 I/O Register Allocation and Cycle Count

Algorithm	Input				Output				Period	Execution time (72MHz)
	X_REG	**Y_REG**	**Z_REG**	**Qn**	**X_REG**	**Y_REG**	**Z_REG**	**W_REG**		
CORDIC rotate	X input (int16)	Y input (int16)	Z input (int16)	15	X output (int16)	Y output (int16)	-	-	23	0.32us
CORDIC vector	X input (int16)	Y input (int16)	Z input (int16)	n （1~15）	-	-	Z output (int16)	-	n+3	0..28us
SVPWM duty	Uβ （int16）	（0xddb3》（15-Qn））*Uα （int32）	-		Ta （uint16）	Tb （uint16）	Tc （uint16）	sector （uint3）	6	0.08us
SVPWM counter	Uβ （int16）	（0xddb3》（15-Qn））*Uα （int32）	-		Ca（uint16）	Cb（uint16）	Cc （uint16）	sector （uint3）	≥12	≥0.17us
Square Root	Input (uint32)	-	-	-	Output (uint16)	-	-	-	12	0.17us
DIV	Divisor (uint/int32)	Dividend （uint/int32）	-	-	Quotient (uint32/int32)	Remainder (uint32/int32)	-	-	10	0.14us
ATAN2	X input (int16)	Y input (int16)	Z input (int16) (Should be 0)	15	-	-	Z output (int16)	-	26	0.36us

The number of cycles starts when the calculation starts (CTRL_REG.RUN=1) and ends when the result is written to the register (STAT_REG.BUSY=0). In the algorithm SVPWM counter, M0CP needs to update PWM through an effective ready protocol interface. Therefore, the number of cycles varies according to the response time of PWM.

3.6 Register address mapping

It only supports 32-bit data access.

Table 10 Coprocessor Register Address Mapping

Register name	Description	Address offset
CTRL_REG	Control register	0x00
STAT_REG	Status register	0x04
TXIS_REG	External interrupt status register	0x08
TXIR_REG	Original operation status register	0x0C
TXIE_REG	External interrupt enable register	0x10
X_REG	X variable register	0x14
Y_REG	Y variable register	0x18
Z_REG	Z variable register	0x1C
W_REG	W parameter register, only for SVPWM sector mode	0x20
BANK_REG	Register group selection control register	0x34
REV_REG	Revision or IP tag register	0x3C
INIT_REG	Initialization register	0x40

3.7 Register functional description

3.7.1 Control register (CTRL_REG)

Offset address: 0x00

Reset value: 0x0000 0078

Field	Name	R/W	Description
0	RUN	W	Run Operation starts, cleared by hardware, and set by software. Write 1 to start the operation. The next cycle is automatically cleared by hardware.
1	BANKAUTO	R/W	Bank Auto Conversion Whether the register bank conversion is automatic. 0: Bank conversion by software 1: Bank conversion by hardware
2	BANKHSEL	R/W	Bank Select 0: Use register bank 0 1: Use register bank 1 When BANKAUTO is configured as 1, this field is read-only, indicating that the bit is managed by hardware. When BANKAUTO is configured as 0, this field is readable and writable.
6:3	QN	R/W	Fixed-point information setting

Field	Name	R/W	Description
			15: Q15 format 14: Q14 format …… 0: Fixed-point operation not performed.
7	CORDICOUT	R/W	CORDIC Output XY output mode for CORDIC. 0: For final output, X and Y are the original result. 1: For final output, X and Y are the result of original result divided by 2.
8	SVPWMSEG	R/W	SVPWM Segmentation 0: SVPWM 7-segment 1: SVPWM 5-segment
9	SVPWMOUT	R/W	SVPWM Output 0: SVPWM output duty cycle result. 1: SVPWM output PWM numerical result (cycle*duty cycle).
10	TXEVEN	R/W	TX Interrupt Enable 0: No TX interrupt will be transmitted. 1: Transmit a TX interrupt after the operation is completed.
15:11	Reserved		
19:16	ALG	R/W	Algebra 0x1: SVPWM. 0x2: CORDIC rotation mode 0x3: CORDIC vector mode 0x4: Square root. 0x5: Division, unsigned 0x6: Division, signed. 0x7: ATAN2 (multiplexing CORDIC vector mode hardware) Others: Reserved
31:20	Reserved		

3.7.2 Status register (STAT_REG)

Offset address: 0x04

Reset value: 0x00000000

Field	Name	R/W	Description
0	BUSY	R	Busy Whether the core is busy (busy during calculation or initialization).
1	DIVERR	R	Division Whether the divisor is 0, which will be cleared when the next operation starts.
2	OVF	R	Overflow Whether data overflow occurs, which will be cleared when the next operation starts.
31:3	Reserved		

3.7.3 External interrupt status register (TXIS_REG)

Offset address: 0x08

Reset value: 0x00000000

Field	Name	R/W	Description
0	DONE	R	Done Flag TX interrupt status bit given to main CPU. When TX interrupt is enabled and the operation is completed, this bit will be set. 0: TX interrupt is disabled or operation is not completed (or no operation is running on m0cp). 1: TX interrupt is enabled and the operation is completed.
31:1	Reserved		

3.7.4 Original interrupt status register (TXIR_REG)

Offset address: 0x0C

Reset value: 0x00000000

Field	Name	R/W	Description
0	DONE	RC_W1	Done Flag Original interrupt signal. It will be set when the operation is completed, no matter whether TX interrupt is enabled. Write 1 to this register to clear 0. 0: No operation or operation not completed. 1: Operation completed.
31:1	Reserved		

3.7.5 External interrupt enable register (TXIE_REG)

Offset address: 0x10

Reset value: 0x00000000

Field	Name	R/W	Description
0	TXIEN	R/W	TX Interrupt Enable 0: Disable TX interrupt. When the operation is completed, no interrupt signal will be pulled up. 1: Enable TX interrupt. When the operation is completed, the interrupt signal to the main CPU will be pulled up.
31:1	Reserved		

Note: Before TXIE_REG is enabled, attention shall be paid to *TXIR_ REG* by software to avoid unexpected interrupt.

3.7.6 Operand/result X register (X_REG)

Offset address: 0x14

Reset value: 0x00000000

Field	Name	R/W	Description
31:0	DATA	R/W	Data of parameter X and result X (Data X)

3.7.7 Operand/result Y register (Y_REG)

Offset address: 0x18

Reset value: 0x00000000

Field	Name	R/W	Description
31:0	DATA	R/W	Data of parameter Y and result Y (Data Y)

3.7.8 Operand/result Z register (Z_REG)

Offset address: 0x1C

Reset value: 0x00000000

Field	Name	R/W	Description
31:0	DATA	R/W	Data of parameter Z and result Z (Data Z)

3.7.9 Result W register (W_REG)

Offset address: 0x20

Reset value: 0x00000000

Field	Name	R/W	Description
31:0	DATA	R	Only used to store data of result W (Data W)

Note: Since the X/Y/Z/W registers all use *32* bits, when the parameter bit width is less than 32 bits, it should be configured after correct sign extension.

For example, in CORDIC, the type of parameter X is int16. When the value of X is -1 (decimal), the value configured to X_REG should be 0xFFFF_FFFF instead of *0x0000_FFFF*. This process can be completed by the compiler by defining the input as *int32* data type.

When used as a result register, all values in the *X/Y/Z/W* register will be expanded to 32 bits according to the sign value. This operation will not affect the correctness of data less than 32 bits.

3.7.10 Register bank register (BANK_REG)

Offset address: 0x34

Reset value: 0x00000000

When CTRL_REG.BANKAUTO is configured as 1, this register is read-only, indicating that these bits are managed by hardware. When CTRL_REG.BANKAUTO is configured as 0, this field is readable and writable.

Field	Name	R/W	Description
0	BANKWSEL	R/W	Writing Bank Select 0: Select bank 0 1: Select bank 1
1	BANKRSEL	R/W	Reading Bank Select 0: Select bank 0 1: Select bank 1
31:2	Reserved		

3.7.11 Version register (REV_REG)

Offset address: 0x3C

Reset value: 0x00050100

Field	Name	R/W	Description
7:0	MIN	R	Minimum Reversion Indicate non-RTL change (document modification, etc.)

15:8	MID	R	Medium Reversion Indicate RTL change invisible to firmware
23:16	MAJ	R	Major Reversion Indicate RTL change visible to firmware
31:24	Reserved		

3.7.12 Initialization register (INIT_REG)

Offset address: 0x40

Reset value: 0x2000 0000

Field	Name	R/W	Description
28:0	Reserved		
29	INIT	R/W	M0CP Initialize Used to initialize M0CP, write 1 first and then 0
31:30	Reserved		

4 Flash memory

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

4.1 Full name and abbreviation of terms

Table 11 Full Name and Abbreviation Description of Terms

Full name in English	English abbreviation
Flash Memory Controller	FMC

4.2 Main characteristics

（1） Flash memory structure

- Contain main memory area and information block
- The capacity of main memory area is up to 64KB
- The information block is divided into system memory, OTP area and option byte area
- The system memory area stores the BootLoader program with a capacity of 2944 Bytes
- The OTP area stores the single-configuration option words with a capacity of 28 Bytes. Users of this area can only write once
- The chip configuration area contains 96-bit unique UID, and capacity information of main memory area, with a capacity of 100 Bytes
- The capacity of the option byte area is 64 Bytes

（2） Functional description

- Read Flash
- Page/Mass erase Flash
- Write Flash
- Read/White protection Flash
- Configure option bytes
- Flash memory content encryption

4.3 Flash memory structure

Table 12 Flash Memory Structure of APM32F035x8T7 Series Products

Block	Name	Address area	Size (bytes)	Sector
Main memory area	Page 0	0x0800 0000–0x0800 03FF	1K	Sector 0
Main memory area	Page 1	0x0800 0400–0x0800 07FF	1K	

Block	Name	Address area	Size (bytes)	Sector
Main memory area	Page 2	0x0800 0800–0x0800 0BFF	1K	
Main memory area	Page 3	0x0800 0C00–0x0800 0FFF	1K	
Main memory area	…	…	…	…
Main memory area	Page 28	0x0800 7000–0x0800 73FF	1K	Sector 7
Main memory area	Page 29	0x0800 7400–0x0800 77FF	1K	
Main memory area	Page 30	0x0800 7800–0x0800 7BFF	1K	
Main memory area	Page 31	0x0800 7C00–0x0800 7FFF	1K	
Main memory area	…	…	…	…
Main memory area	Page 60	0x0800 F000–0x0800 F3FF	1K	Sector 15
Main memory area	Page 61	0x0800 4000–0x0800 F7FF	1K	
Main memory area	Page 62	0x0800 8000–0x0800 FBFF	1K	
Main memory area	Page 63	0x0800 C000–0x0800 FFFF	1K	
Information block	System memory area	0x1FFF EC00–0x1FFF F77F	2944	-
Information block	OTP area	0x1FFF F780–0x1FFF F79B	28	-
Information block	Chip configuration area	0x1FFF F79C–0x1FFF F7FF	100	-
Information block	Option byte	0x1FFF F800–0x1FFF F83F	64	-

Notes:

（1）The OTP area cannot be erased after being configured.

4.4 Functional description

Describe the operation of main memory and information block (including system memory area and option byte), including read, write, erase and read/write protection.

Reading Flash includes main memory block and information block, while the

erase, write, read/write is introduced separately; the system memory area has been written before the product leaves the factory and cannot be modified by users. The erase, write, and read/write protection of the module will not be introduced.

4.4.1 Read Flash

Flash memory can be directly addressed, and reading Flash is affected by the following configuration:

Wait cycle

Different wait cycles should be configured for different system clocks:

- 0 wait cycle: 0<system clock≤24MHz
- 1 wait cycle: 24MHz<system clock≤48MHz
- 2 wait cycles: 48MHz<system clock≤72MHz

Prefetch buffer

It can improve the reading speed and every time it is reset, the prefetch buffer will be automatically opened; the read interface with prefetch buffer. It can be configured only when the system clock is consistent with AHB clock and is less than 24MHz, and can be used only when the system clock is consistent with AHB clock.

4.4.2 Main memory block

4.4.2.1 Erase main memory block

FMC supports page erase and mass erase (erase all) to initialize the contents of the main memory area to high level (the data is represented as 0xFFFF). Before writing to Flash, users are advised to erase the write address page. If the data of write address is not 0xFFFF, a programming error will be triggered.

Main memory page erase

Page erase is an independent erase according to the main memory area page selected by the program, which will not have any impact on the page not selected for erasure.

After the correct page erase (or flash write operation) is completed, OCF of FMC_STS register will be set. If OCIE interrupt is enabled, an operation completion interrupt will be triggered. Users need to pay attention that the page selected for erase must be a valid page (the valid address of the main memory area and the address not write-protected).

The page erasing process is:

（1） Check that BUSYF of the FMC_STS register is 0, indicating that no other operation is in progress at this time.

（2） Set PAGEERA of FMC_CTRL2 register.

（3） Write the address of the page to be erased to FMC_ ADDR register.

（4） Set STA of FMC_CTRL2 register.

（5） Wait until the BUSYF of FMC_STS register becomes 0.

（6） Check OCF flag of FMC_STS register.

（7） Clear OCF flag of FMC_STS register.

Main memory mass erase

The mass erase operation will erase all the contents in the main memory area of Flash, and erase all the data in the main memory area, so users need to pay special attention when using it to avoid the loss of important data caused by misoperation.

The mass erase process is:

（1） Check that BUSYF of the FMC_STS register is 0, indicating that no other operation is in progress at this time.

（2） Set MASSERA of FMC_CTRL2 register.

（3） Set STA of FMC_CTRL2 register.

（4） Wait until the BUSYF of FMC_STS register becomes 0.

（5） Check OCF flag of FMC_STS register.

（6） Clear OCF flag of FMC_STS register.

4.4.2.2 Write main memory block

FMC supports the writing of 16-bit (half word) data in the main memory area. You can judge whether the erasing is successful by Debug, BootLoader, program running in SRAM, or directly reading the erased page.

In order to ensure correct writing, it is necessary to check whether the destination address has been erased before writing; if it is not erased, the written data will be invalid and PEF bit of FMC_STS register will be set to "1". If the destination address has write protection, the written data will be invalid and a write protection error will be triggered (WPEF bit of FMC_STS is set to "1").

4.4.2.3 Main memory block of read/write protection

Read/Write protection of the flash is used to prevent invalid reading/modification of the main memory area code or data, and it is controlled by the read/write protection configuration byte of option byte. For APM32F035x8T7 series products, the basic unit of write protection is 4 pages (i.e. 4KB).

Read protection

The read protection has three levels, namely, Level 0, Level 1 and Level 2, which are specifically described as follows:

Table 13 Difference among Read Protection Levels

Category	READPROT	Description
Level 0	0xAA	The main memory area and option byte are erasable, writable and readable.
Level 1	Other values except 0xAA and 0xCC	User mode: Allowed to erase, write and read the main memory area and option byte. Debug, SRAM running, system memory area running: Access to the main memory area is disabled; the option byte is erasable, writable and readable, but when the level is modified to 0, mass erase of main memory area will be performed first.
Level 2	0xCC	User mode: It is allowed to erase, write and read the main memory area, and read and write the option bytes (except for READPROT, which cannot be written), and option bytes cannot be erased Unable to debug. The system cannot start from SRAM and System memory

Write protection

Write protection control can be conducted for the corresponding pages of the main memory block by configuring the value of write protection option byte WRP0/1/2/3. After the write protection is enabled, the content on the corresponding page of the main memory area cannot be modified in any way.

4.4.2.4 Unlock/Lock main memory block

FMC_CTRL2 of the reset FMC will be locked by hardware, and then FMC_CTRL2 cannot be directly written, and the corresponding value must be written to FMC_KEY in the correct sequence to unlock FMC. The KEY value is as follows:

- KEY1=0x45670123
- KEY2=0xCDEF89AB

The wrong writing sequence or wrong value will cause the program to enter the hardware wrongly. At this time, FMC will be locked, and all FMC operations will be invalid until it is reset next time. Users can also lock FMC by software by writing "1" to LOCK bit of the control register 2 (FMC_CTRL2).

In each Flash programming operation, users must follow the steps of "Flash unlock - program by user - Flash lock", so as to avoid the risk that user code/data is accidentally modified due to the Flash unlocking after the Flash programming operation.

4.4.3 Option byte

4.4.3.1 Erase option byte

Support erase function. After the correct option byte erase (or option byte write operation) is completed, OCF of FMC_STS register will be set. If OCIE interrupt is enabled, an operation completion interrupt will be triggered.

The option byte erasing process is:

（1） Check that BUSYF of the FMC_STS register is 0, indicating that no other operation is in progress at this time.

（2） Unlock OBWEN bit of FMC_CTRL2 register.

（3） Set OBE of FMC_CTRL2 register.

（4） Set STA of FMC_CTRL2 register.

（5） Wait until the BUSYF of FMC_STS register becomes 0.

（6） Check OCF flag of FMC_STS register.

（7） Clear OCF flag of FMC_STS register.

4.4.3.2 Write option byte

The option bytes support the write function.

The option byte writing process is:

（1） Check that BUSYF of the FMC_STS register is 0, indicating that no other operation is in progress at this time.

（2） Unlock OBWEN bit of FMC_CTRL2 register.

（3） Set OBP of FMC_CTRL2 register.

（4） Write 16-bit (half-word) data to the option byte area address (pay attention to complementation).

（5） Wait until the BUSYF of FMC_STS register becomes 0.

（6） Check OCF flag of FMC_STS register.

（7） Clear OCF flag of FMC_STS register.

4.4.3.3 Write-protect option byte

By default, the option byte is always readable and write protected. To perform write operation (program/erase) for the option byte block, first write the correct key sequence (the same as that of locking) in FMC_OBKEY, and then allow the write operation of option byte block; the OBWEN bit of FLASH_CTRL2 register indicates write enabled; clear this bit and write operation will be disabled.

4.4.3.4 Unlock/Lock option byte

After the system reset, the option byte is locked by default. Only when the option byte is unlocked correctly, can it be modified. The difference between option byte unlocking and flash unlocking is that the KEY value is written to FMC_OBKEY register rather than FMC_KEY register. The option byte does not support "software lock". The user should pay special attention to that every time the value of the option byte is modified, the system must be reset to make it take effect.

4.4.4 Functional description of option byte

The option byte provides some optional functions for users, and it mainly consists of multiple configurable bytes and corresponding complementary codes. Every time the system is reset, the option byte area will be reloaded to the FMC_ OBCS and FMC_WRTPROT register (the option byte will only take effect each time they are reloaded to FMC). In the process of reloading, if a certain configurable byte does not match its inverse code, an option byte error (OBE bit of FMC_register is set to "1") will be triggered, and this byte will be set to "0xFF".

APM32F035x8T7 encryption can be divided into data segment and instruction segment encryption:

The data segment encryption is configured by the security configuration word dcode_cipher_page[63:0]. Each bit corresponds to a page, and dcode_cipher_page[0]~ dcode_cipher_page[63] correspond to page0~page63 respectively.

The instruction segment encryption is configured by the security configuration word icode_cipher_page[63:0]. Each bit corresponds to a page, and icode_cipher_page[0]~ icode_cipher_page[63] correspond to page0~page63 respectively.

The encrypted key can be configured by opt_enc_same_key, icode_cipher_key_sel and dcode_cipher_key_sel. The key can be from UID or register configuration.

The encryption module can be enabled by the register in SYSCFG.

The information of the option byte area is shown in the table below.

Table 14 Option Bytes

Address	Option byte	Initial value	R/W	Functional description
0x1FFF F800	READPROT	0xA5	R/W	Read protection configuration Bit [7:0]: READPROT 0xAA: Level 0 0xCC: Level 2 Others: Level 1

Address	Option byte	Initial value	R/W	Functional description
0x1FFF F801	nREADPROT	0x5A	R	READPROT complementary code
0x1FFF F802	UOB	0xFF	R/W	User option byte Bit 0: WDTSEL 0: Hardware watchdog 1: Software watchdog Bit 1: nRSTSTOP 0: Reset occurs when entering the Stop mode 1: Reset does not occur when entering the Stop mode Bit 2: nRSTSTB 0: Reset occurs when entering the Standby mode 1: Reset does not occur when entering the Standby mode Bit 3: Reserved Bit 4: nBOOT1 Select BOOT mode Bit 5: VDDAMONI 0: V_{DDA} power supply detector is disabled 1: V_{DDA} power supply detector is enabled Bit 6: Reserved Bit 7: Reserved
0x1FFF F803	nUOB	0x00	R	UOB complementary code
0x1FFF F804	Data0	0xFF	R/W	User data byte 0
0x1FFF F805	nData0	0x00	R	Data0 complementary code
0x1FFF F806	Data1	0xFF	R/W	User data byte 1
0x1FFF F807	nData1	0x00	R	Data1 complementary code
0x1FFF F808	WRP0	0xFF	R/W	Write protection configuration 0
0x1FFF F809	nWRP0	0x00	R	WRP0 complementary code
0x1FFF F80A	WRP1	0xFF	R/W	Write protection configuration 1
0x1FFF F80B	nWRP1	0x00	R	WRP1 complementary code
0x1FFF F80C	WRP2	0xFF	R/W	Write protection configuration 2
0x1FFF F80D	nWRP2	0x00	R	WRP2 complementary code
0x1FFF F80E	WRP3	0xFF	R/W	Write protection configuration 3
0x1FFF F80F	nWRP3	0x00	R	WRP3 complementary code

Address	Option byte	Initial value	R/W	Functional description
0x1FFF F810	icode_cipher_page[7:0]	0xFF	W	icode_cipher_page[63:0] is the program segment scrambling sector number enable, and each bit is the scrambling enable of the corresponding page. Each bit scrambles one page; icode_cipher_page[0] corresponds to Page0, and icode_cipher_page[63] corresponds to Page63. 1'b0: Scrambling enabled 1'b1: Scrambling not enabled
0x1FFF F811	nicode_cipher_page[7:0]	0xFF	W	Complementary code of corresponding bit
0x1FFF F812	icode_cipher_page[15:8]	0xFF	W	-
0x1FFF F813	nicode_cipher_page[15:8]	0xFF	W	Complementary code of corresponding bit
0x1FFF F814	icode_cipher_page[23:16]	0xFF	W	-
0x1FFF F815	nicode_cipher_page[23:16]	0xFF	W	Complementary code of corresponding bit
0x1FFF F816	icode_cipher_page[31:24]	0xFF	W	-
0x1FFF F817	nicode_cipher_page[31:24]	0xFF	W	Complementary code of corresponding bit
0x1FFF F818	icode_cipher_page[39:32]	0xFF	W	-
0x1FFF F819	nicode_cipher_page[39:32]	0xFF	W	Complementary code of corresponding bit
0x1FFF F81A	icode_cipher_page[47:40]	0xFF	W	-
0x1FFF F81B	nicode_cipher_page[47:40]	0xFF	W	Complementary code of corresponding bit
0x1FFF F81C	icode_cipher_page[55:48]	0xFF	W	-
0x1FFF F81D	nicode_cipher_page[55:48]	0xFF	W	Complementary code of corresponding bit
0x1FFF F81E	icode_cipher_page[63:56]	0xFF	W	-
0x1FFF F81F	nicode_cipher_page[63:56]	0xFF	W	Complementary code of corresponding bit

Address	Option byte	Initial value	R/W	Functional description
0x1FFF F820	dcode_cipher_page[7:0]	0xFF	W	dcode_cipher_page[63:0] is the data segment scrambling sector number enable, and each bit is the scrambling enable of the corresponding page. Each bit scrambles one page; dcode_cipher_page[0] corresponds to Page0, and dcode_cipher_page[63] corresponds to Page63. 1'b0: Scrambling enabled 1'b1: Scrambling not enabled
0x1FFF F821	ndcode_cipher_page[7:0]	0xFF	W	Complementary code of corresponding bit
0x1FFF F822	dcode_cipher_page[15:8]	0xFF	W	-
0x1FFF F823	ndcode_cipher_page[15:8]	0xFF	W	Complementary code of corresponding bit
0x1FFF F824	dcode_cipher_page[23:16]	0xFF	W	-
0x1FFF F825	ndcode_cipher_page[23:16]	0xFF	W	Complementary code of corresponding bit
0x1FFF F826	dcode_cipher_page[31:24]	0xFF	W	-
0x1FFF F827	ndcode_cipher_page[31:24]	0xFF	W	Complementary code of corresponding bit
0x1FFF F828	dcode_cipher_page[39:32]	0xFF	W	-
0x1FFF F82A	dcode_cipher_page[47:40]	0xFF	W	-
0x1FFF F82B	ndcode_cipher_page[47:40]	0xFF	W	Complementary code of corresponding bit
0x1FFF F82C	dcode_cipher_page[55:48]	0xFF	W	-
0x1FFF F82D	ndcode_cipher_page[55:48]	0xFF	W	Complementary code of corresponding bit
0x1FFF F82E	dcode_cipher_page[63:56]	0xFF	W	-
0x1FFF F82F	ndcode_cipher_page[63:56]	0xFF	W	Complementary code of corresponding bit

Address	Option byte	Initial value	R/W	Functional description
0x1FFF F830	icode_cipher_key_sel	0xFF	W	Bit 7-2: Reserved Bit 1-0: icode_cipher_key_sel 2'b11: 2'b10: icode_cipher_key from UID 2'b0x: icode_cipher_key from SYSCFG register
0x1FFF F831	nicode_cipher_key_sel	0xFF	W	Bit 7-3: Complementary to 0x1FFF F830 bit 7-3 Bit 2: nopt_enc_same_key Bit 1-0: nicode_cipher_key_sel
0x1FFF F832	dcode_cipher_key_sel	0xFF	W	Bit 7-3: Reserved Bit 2: opt_enc_same_key 1'b1: Data encryption and instruction encryption use the same key (the encryption of the two encryption modules is generated by icode_cipher_key and dcode_cipher_key through operation, so it is necessary to ensure that icode_cipher_key and dcode_cipher_key are unchanged during the program operation). At this time, the data encryption space and the instruction encryption space page can coincide 1'b0: The data encryption and instruction encryption keys are independent. At this time, the data encryption space and the instruction encryption space page cannot coincide Bit 1-0: dcode_cipher_key_sel 2'b11: 2'b10: dcode_cipher_key from UID 2'b0x: dcode_cipher_key from SYSCFG register
0x1FFF F833	ndcode_cipher_key_sel	0xFF	W	Bit 7-2: Complementary to 0x1FFF F832 bit 7-2 Bit 1-0: ndcode_cipher_key_sel

Note: When the configurable byte and its reverse code value are "0xFF", the match will not be verified in the reloading process. Flash read/write protection configuration is independent of each other. Removing the write protection will not force the loss of the contents of the main memory area, but keep them as they are.

4.4.5 OTP option byte

4.4.5.1 Erase OTP option byte

Erase function is not supported.

4.4.5.2 Write OTP option byte

The OTP option byte only supports the write function in the full-F state.

4.4.5.3 OTP option byte of write protection

By default, the OTP option byte is always write protected. To perform write operation (program/erase) for the option byte block, first write the correct key sequence (the same as that of locking) in FMC_OBKEY, and then allow the write operation of option byte block; the OBWEN bit of FLASH_CTRL2 register indicates write enabled; clear this bit and write operation will be disabled.

4.4.5.4 Unlock/Lock OTP option byte

After the system reset, the option byte is locked by default. Only when the option byte is unlocked correctly, can it be modified. The difference between OTP option byte unlocking and flash unlocking is that the KEY value is written to FMC_OBKEY register rather than FMC_KEY register. The OTP option byte does not support "software lock". The user should pay special attention that every time the value of the OTP option byte is modified, the system must be reset to make it take effect.

4.4.6 Functional description of OTP option byte

APM32F035x8T7 can encrypt Flash memory content. max_security_page[63:0] is the sector number of the highest security zone. Each bit protects one page. max_security_page[0] corresponds to page0, and max_security_page[63] corresponds to page63

The highest security zone means that this zone can only be used to execute user code, and only the CPU is allowed to access this zone by fetch instructions; any other form of read operation for this area will cause a mass erase to the flash main area; generally, it is recommended that the highest security zone cover the program scrambling space.

opt_valid_key[15:0] configures enable for this zone, configures other option byte of this zone, and then configures to 16’h0914 to make the configuration take effect.

Table 15 The Functional description of Option byte

Address	Option byte	Initial value	R/W	Functional description
0x1FFF F780	opt_valid_key[7:0]	0xFF	W	The option byte of this zone will take effect only when

Address	Option byte	Initial value	R/W	Functional description
				opt_valid_key[15:0]=16'h0914. Therefore, it is necessary to configure the option byte of this zone and then configure this key
0x1FFF F781	nopt_valid_key[7:0]	0xFF	W	Complementary code of corresponding bit
0x1FFF F782	opt_valid_key[15:8]	0xFF	W	-
0x1FFF F783	nopt_valid_key[15:8]	0xFF	W	Complementary code of corresponding bit
0x1FFF F788	user_swd_key[7:0]	0xFF	W	When user_swd_key[15:0]=16'h2021, the SWD debugging interface will be disabled
0x1FFF F789	nuser_swd_key[7:0]	0xFF	W	Complementary code of corresponding bit
0x1FFF F78A	user_swd_key[15:8]	0xFF	W	-
0x1FFF F78B	nuser_swd_key[15:8]	0xFF	W	Complementary code of corresponding bit
0x1FFF F78C	max_security_page[7:0]	0xFF	W	max_security_page[63:0]: The sector number of the highest security zone. Each bit protects one page. max_security_page[0] corresponds to page0, and max_security_page[63] corresponds to page63 The highest security zone means that this zone can only be used to execute user code, namely, program segment; any other form of read operation for this area will cause a mass erase to the flash main area; generally, it is recommended that the highest security zone cover the program scrambling space; 1'b0: The highest security zone enabled

Address	Option byte	Initial value	R/W	Functional description
				1'b1: The highest security zone not enabled
0x1FFF F78D	nmax_security_page[7:0]	0xFF	W	Complementary code of corresponding bit
0x1FFF F78E	max_security_page[7:0]	0xFF	W	-
0x1FFF F78F	nmax_security_page[7:0]	0xFF	W	Complementary code of corresponding bit
0x1FFF F790	max_security_page[15:8]	0xFF	W	-
0x1FFF F791	nmax_security_page[15:8]	0xFF	W	Complementary code of corresponding bit
0x1FFF F792	max_security_page[23:16]	0xFF	W	-
0x1FFF F793	nmax_security_page[23:16]	0xFF	W	Complementary code of corresponding bit
0x1FFF F794	max_security_page[31:24]	0xFF	W	-
0x1FFF F795	nmax_security_page[31:24]	0xFF	W	Complementary code of corresponding bit
0x1FFF F796	max_security_page[39:32]	0xFF	W	-
0x1FFF F797	nmax_security_page[39:32]	0xFF	W	Complementary code of corresponding bit
0x1FFF F798	max_security_page[47:40]	0xFF	W	-
0x1FFF F799	nmax_security_page[47:40]	0xFF	W	Complementary code of corresponding bit
0x1FFF F79A	max_security_page[55:48]	0xFF	W	-
0x1FFF F79B	nmax_security_page[55:48]	0xFF	W	Complementary code of corresponding bit

4.5 Register address mapping

Base address: 0x40022000

Table 16 FMC Register Address Mapping

Register name	Description	Offset address
FMC_CTRL1	Control register 1	0x00
FMC_KEY	Key register	0x04
FMC_OBKEY	Option byte register	0x08
FMC_STS	Status register	0x0C
FMC_CTRL2	Control register 2	0x10
FMC_ADDR	Flash address register	0x14

Register name	Description	Offset address
FMC_OBCS	Option byte control/status register	0x1C
FMC_WRTPROT	Write protection register	0x20

4.6 Register functional description

4.6.1 Control register 1 (FMC_CTRL1)

Offset address: 0x00

Reset value: 0x0000 0000

Field	Name	R/W	Description
2:0	WS	R/W	Wait State Configure 000: 0 wait cycle, 0<system clock≤24MHz 001: 1 wait cycle, 24MHz<system clock≤48MHz 010: 2 wait cycles, 48MHz<system clock≤72MHz Others: Reserved
3	Reserved		
4	PBEN	R/W	Prefetch Buffer Enable 0: Disable 1: Enable
5	PBSF	R	Prefetch Buffer Status Flag 0: In disabled state 1: In enabled state
31:6	Reserved		

4.6.2 Key register 1 (FMC_KEY)

Offset address: 0x04

Reset value: xxxx xxxx

Field	Name	R/W	Description
31:0	KEY	W	FMC Key Writing the keys represented by these bits can unlock FMC. These bits can only perform write operation, and 0 is returned when read operation is performed.

4.6.3 Option byte key register (FMC_OBKEY)

Offset address: 0x08

Reset value: xxxx xxxx

Field	Name	R/W	Description
31:0	OBKEY	W	Option Byte Key Writing the keys represented by these bits can unlock the option byte write operation. These bits can only perform write operation and 0 is returned when read operation is performed.

4.6.4 Status register (FMC_STS)

Offset address: 0x0C

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	BUSYF	R	Busy Flag This bit indicates that a flash operation is in progress. These bits can only perform write operation, and 0 is returned when read operation is performed.
1	Reserved		
2	PEF	R/W	Programming Error Flag This bit will be set by software when the value before the address is edited is not "0xFFFF".
3	Reserved		
4	WPEF	R/W	Write Protection Error Flag This bit will be set by hardware when programming the write protection address in FLASH.
5	OCF	R/W	Operation Complete Flag This bit will be set by hardware when read/write operation in FLASH is completed.
31:6	Reserved		

4.6.5 Control register 2 (FMC_CTRL2)

Offset address: 0x10

Reset value: 0x0000 0080

Field	Name	R/W	Description
0	PG	R/W	Program Set this bit to 1 to program Flash.
1	PAGEERA	R/W	Page Erase Set this bit to 1 to erase the page.
2	MASSERA	R/W	Mass Erase Set this bit to 1 to erase the mass.
3	Reserved		
4	OBP	R/W	Option Byte Program Set this bit to 1 to program the option byte.
5	OBE	R/W	Option Byte Erase Set this bit to 1 to erase the option byte.
6	STA	R/W	Start Erase This bit can be only set to 1 by software, and can be reset by clearing STS_BUSYF bit.
7	LOCK	R/W	Lock This bit can be only written to 1, and when this bit is set to 1, it means that FMC and CTRL2 registers are locked.
8	Reserved		
9	OBWEN	R/W	Option Byte Write Enable When this bit is set to 1, the option byte can be programmed.

Field	Name	R/W	Description
10	ERRIE	R/W	Error interrupt Enable 0: Disable interrupt 1: Enable interrupt When STS_PEF=1 or STS_WPEF=1, set this bit to generate an interrupt.
11	Reserved		
12	OCIE	R/W	Operation Complete Interrupt Enable 0: Disable operation completion interrupt 1: Enable operation completion interrupt When STS_OCF=1, set this bit to generate an interrupt.
13	OBLOAD	R/W	Force Option Byte Load When this bit is set to 1, force to reload the option byte to generate system reset. 0: Idle 1: Force to load
31:14	Reserved		

4.6.6 Address register (FMC_ADDR)

Offset address: 0x14

Reset value: 0x0000 0000

The register is changed to currently/finally used address by hardware; in page erase, the register needs to be configured by software.

Field	Name	R/W	Description
31:0	ADDR	W	Flash Address In programming operation, the bit is written to the address to be programmed; in page erase, this bit is written to the page to be erased.

4.6.7 Option bye control/status register (FMC_OBCS)

Offset address: 0x1C

Reset value: 0xXXXX XX0X

The reset value of the register is related to the value written in the option byte; the reset value of OBE bit is related to the result whether the value of the loaded option byte is consistent with its inverse code.

Field	Name	R/W	Description
0	OBE	R	Option Byte Error 1: The loaded option byte does not match its complementary code. The option byte and its complementary code are forced to write to 0xFF
2:1	READPROT	R	Read Protect Display which level of read protection is enabled. The level is 1 when bit1 is set, and 2 when bit2 is set. 00: Level 0 01: Level 1 1X: Level 2

Field	Name	R/W	Description
7:3	Reserved		
8	WDTSEL	R	Watchdog Select 0: Hardware watchdog 1: Software watchdog
9	RSTSTOP	R	nReset in STOP Mode 0: Generate 1: Not generate
10	RSTSTDB	R	nReset in STANDBY Mode 0: Generate 1: Not generate
11	Reserved		
12	nBOOT1	R	nBoot1 Mode Configure
13	VDDAMONI	R	V_{DDA} Monitor
14	SRAMPARITY	R	SRAM Parity Check
15	Reserved		
23:16	DATA0	R	Data0
31:24	DATA1	R	Data1

4.6.8 Write protection register (FMC_WRTPROT)

Offset address: 0x20

Reset value: 0xXXXX XXXX (the reset value depends on the programming value in option bye)

Field	Name	R/W	Description
31:0	WRTPROT	R	Write Protect 0: Valid 1: Invalid

5 System configuration controller (SYSCFG)

5.1 Full name and abbreviation of terms

Table 17 Full Name and Abbreviation Description of Terms

Full name in English	English abbreviation
Fast Mode Plus	FM+
System Configuration Controller	SYSCFG

SYSCFG is mainly used to manage address mapping and control interrupts, specifically: controlling the fast mode plus of I2C on some IO ports; configuring DMA trigger source remapping; remapping from memory to code start area; and managing the external interrupts connected to GPIO.

For details of related configuration, see SYSCFG register configuration.

5.2 Register address mapping

Table 18 SYSCFG Register Address Mapping

Register Name	Description	Offset address
SYSCFG_CFG1	Configuration register 1	0x00
SYSCFG_EINTCFG1	External interrupt register 1	0x08
SYSCFG_EINTCFG2	External interrupt register 2	0x0C
SYSCFG_EINTCFG3	External interrupt register 3	0x10
SYSCFG_EINTCFG4	External interrupt register 4	0x14
SYSCFG_CFG2	Configuration register 2	0x18
SYSCFG_DCODE	Data scrambling key register	0x20
SYSCFG_ICODE	Instruction scrambling key register	0x24
SYSCFG_CIPHEN	Scrambling key register	0x28
SYSCFG_ACCESSEN	Processor access IP permission register	0x2C

5.3 Register functional description

5.3.1 Configuration register 1 (SYSCFG_CFG1)

Offset address: 0x00

Reset value: 0x0000 000X (X means memory mode, controlled by BOOT. After reset, these bits select mode configuration parameters through BOOT pin.)

This register is used to configure the memory and DMA requested remapping and control the specific I/O pins.

These two bits are used to configure the storage type with the address of 0x0000 0000.

All of these bits can skip the hardware to have the software to select the physical mapping, and can be controlled and reset by software.

Field	Name	R/W	Description
1:0	MMSEL	R/W	Memory Mapping Select Control the memory mapping address 0x0000 0000. After reset, the parameters of these bits are determined by actual BOOT. X0: Main flash mapping address: 0x0000 0000 01: System flash mapping address: 0x0000 0000 11: Embedded SRAM mapping address: 0x0000 0000
5:2	Reserved		
6	IRSEL	R/W	IR Modulation Envelope Signal Select This bit controls whether the modulation signal comes from TMR3 or USRAT1. 0: The modulation signal comes from TMR3 1: The modulation signal comes from USRAT1
7	Reserved		
8	ADCDMARMP	R/W	ADC DMA Request Remap Control remapping request of ADC DMA. 0: No remapping ADC—DMA_CH1 1: Remapping ADC—DMA_CH2
9	USART1TXRMP	R/W	USART1_TX DMA Request Remap This bit controls remapping request of USART1_TX DMA. 0: No remapping USART1_RX—DMA_CH2 1: Remapping USART1_RX—DMA_CH4
10	USART1RXRMP	R/W	USART1_RX DMA Request Remap This bit controls remapping request of USART1_RX DMA. 0: No remapping USART1_TX—DMA_CH3 1: Remapping USART1_TX—DMA_CH5
15:11	Reserved		
16	I2CPB6FMP	R/W	Fast Mode Plus Driving Capability Activate for PB6 This bit enables PB6 interface to enable I2C fast mode plus. 0: PB6 pin is set as standard mode. 1: PB6 pin is set as I2C fast mode plus and I2C speed control is bypassed (ignored).

Field	Name	R/W	Description
17	I2CPB7FMP	R/W	Fast Mode Plus Driving Capability Activate for PB7 This bit enables PB7 interface to enable I2C fast mode plus. 0: PB7 pin is set as standard mode. 1: PB7 pin is set as I2C fast mode plus and I2C speed control is bypassed (ignored).
18	I2CPB8FMP	R/W	Activate PB8 pin fast mode plus (FM+) driving capability (Fast Mode Plus Driving Capability Activate for PB8) This bit enables PB8 interface to enable I2C fast mode plus. 0: PB8 pin is set as standard mode. 1: PB8 pin is set as I2C fast mode lus and I2C speed control is bypassed (ignored).
19	I2CPB9FMP	R/W	Activate PB9 pin fast mode plus (FM+) driving capability (Fast Mode Plus Driving Capability Activate for PB9) This bit enables PB9 interface to enable I2C fast mode plus. 0: PB9 pin is set as standard mode. 1: PB9 pin is set as I2C fast mode plus and I2C speed control is bypassed (ignored).
20	I2CPC4FMP	R/W	Fast Mode Plus Driving Capability Activate for PB7 This bit enables PB9 interface to enable I2C fast mode plus. 0: PC4 pin is set as standard mode 1: PB9 pin is set as I2C fast mode plus and I2C speed control is bypassed (ignored).
21	I2CPC5FMP	R/W	Fast Mode Plus Driving Capability Activate for PB7 This bit enables PB9 interface to enable I2C fast mode plus. 0: PC5 pin is set as standard mode 1: PB9 pin is set as I2C fast mode plus and I2C speed control is bypassed (ignored).
22	SPIDMARMP	R/W	SPI DMA Request Remap This bit controls DMA remapping request of SPI. 0: No remapping SPI_RX—DMA_CH2，SPI_TX—DMA_CH3 1: Remapping SPI_RX—DMA_CH4，SPI_TX—DMA_CH5
26:23	Reserved		
27	I2CDMARMP	R/W	I2C DMA Request Remap This bit controls DMA remapping request of I2C. 0: No remapping I2C_RX—DMA_CH3，I2C_TX—DMA_CH2 1: Remapping I2C_RX—DMA_CH7，I2C_TX—DMA_CH6

Field	Name	R/W	Description
28	TMR1DMARMP	R/W	TMR1 DMA Request Remap This bit controls DMA remapping request of TMR1. 0: No remapping TMR1_CH1—DMA_CH2，TMR1_CH2—DMA_CH3，TMR1_CH3—DMA_CH5 1: Remapping TMR1_CH1—DMA_CH6 ， TMR1_CH2—DMA_CH6 ，TMR1_CH3—DMA_CH6
29	TMR2DMARMP	R/W	TMR2 DMA Request Remap This bit controls DMA remapping request of TMR2. 0: No remapping TMR2_CH1—DMA_CH3，TMR2_CH2—DMA_CH4 1: Remapping TMR2_CH1—DMA_CH7，TMR2_CH2—DMA_CH7
30	TMR3DMARMP	R/W	TMR3 DMA Request Remap This bit controls DMA remapping request of TMR3. 0: No remapping TMR3_CH1—DMA_CH4，TMR3_CH2—DMA_CH4 1: Remapping TMR3_CH1—DMA_CH6，TMR3_CH2—DMA_CH6
31	Reserved		

5.3.2 External interrupt register 1 (SYSCFG_EINTCFG1)

These bits are controlled by software to be rewritten to select the external interrupt source of EINTx (x=0…15). The selected external interrupt sources represented by values of the EINTx [3:0] (note: EINT3/7/11/15 is [2:0], and it has only 3 bits) are shown in the table below.

Please refer to the "Pin definitions" chapter of the datasheet for the specific number of pins.

Table 19 External Interrupt Sources Selected for Different Values

EINTx [3:0]	External interrupt source
x000	PA[x] pin
x001	PB[x] pin
x010	PC[x] pin
x101	PF[x] pin
Others	Reserved

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
3:0	EINT0	R/W	EINT0 Configure These bits are controlled by software to be rewritten to select the external interrupt source of EINT0. The selected external interrupt sources represented by values of the bit are shown in Table 19.
7:4	EINT1	R/W	EINT1 Configure These bits are controlled by software to be rewritten to select the external interrupt source of EINT1. The selected external interrupt sources represented by values of the bit are shown in Table 19.
11:8	EINT2	R/W	EINT2 Configure These bits are controlled by software to be rewritten to select the external interrupt source of EINT2. The selected external interrupt sources represented by values of the bit are shown in Table 19.
14:12	EINT3	R/W	EINT3 Configure These bits are controlled by software to be rewritten to select the external interrupt source of EINT3. The selected external interrupt sources represented by values of the bit are shown in Table 19.
31:15	Reserved		

5.3.3 External interrupt register 2 (SYSCFG_EINTCFG2)

These bits are controlled by software to be rewritten to select the external interrupt source of EINTx(x=4…7). The selected external interrupt sources represented by values of the EINTx [3:0] are shown in Table 19.

Offset address: 0x0C

Reset value: 0x0000 0000

Field	Name	R/W	Description
3:0	EINT4	R/W	EINT4 Configure These bits are controlled by software to be rewritten to select the external interrupt source of EINT4. The selected external interrupt sources represented by values of the bit are shown in Table 19.
7:4	EINT5	R/W	EINT5 Configure These bits are controlled by software to be rewritten to select the external interrupt source of EINT5. The selected external interrupt sources represented by values of the bit are shown in Table 19.
11:8	EINT6	R/W	EINT6 Configure These bits are controlled by software to be rewritten to select the external interrupt source of EINT6. The selected external interrupt sources represented by values of the bit are shown in Table 19.

Field	Name	R/W	Description
14:12	EINT7	R/W	EINT7 Configure These bits are controlled by software to be rewritten to select the external interrupt source of EINT7. The selected external interrupt sources represented by values of the bit are shown in Table 19.
31:15	Reserved		

5.3.4 External interrupt register 3 (SYSCFG_EINTCFG3)

These bits are controlled by software to be rewritten to select the external interrupt source of EINTx(x=8…11). The selected external interrupt sources represented by values of the EINTx [3:0] are shown in Table 19.

Offset address: 0x10

Reset value: 0x0000 0000

Field	Name	R/W	Description
3:0	EINT8	R/W	EINT8 Configure These bits are controlled by software to be rewritten to select the external interrupt source of EINT8. The selected external interrupt sources represented by values of the bit are shown in Table 19.
7:4	EINT9	R/W	EINT9 Configure These bits are controlled by software to be rewritten to select the external interrupt source of EINT9. The selected external interrupt sources represented by values of the bit are shown in Table 19.
11:8	EINT10	R/W	EINT10 Configure These bits are controlled by software to be rewritten to select the external interrupt source of EINT10. The selected external interrupt sources represented by values of the bit are shown in Table 19.
14:12	EINT11	R/W	EINT11 Configure These bits are controlled by software to be rewritten to select the external interrupt source of EINT11. The selected external interrupt sources represented by values of the bit are shown in Table 19.
31:15	Reserved		

5.3.5 External interrupt register 4 (SYSCFG_EINTCFG4)

These bits are controlled by software to be rewritten to select the external interrupt source of EINTx(x=12 to 15). The selected external interrupt sources represented by values of the EINTx [3:0] are shown in Table 19.

Offset address: 0x14

Reset value: 0x0000 0000

Field	Name	R/W	Description
3:0	EINT12	R/W	EINT12 Configure These bits are controlled by software to be rewritten to select the external interrupt source of EINT12. The selected external interrupt sources represented by values of the bit are shown in Table 19.
7:4	EINT13	R/W	EINT13 Configure These bits are controlled by software to be rewritten to select the external interrupt source of EINT13. The selected external interrupt sources represented by values of the bit are shown in Table 19.
11:8	EINT14	R/W	EINT14 Configure These bits are controlled by software to be rewritten to select the external interrupt source of EINT14. The selected external interrupt sources represented by values of the bit are shown in Table 19.
14:12	EINT15	R/W	EINT15 Configure These bits are controlled by software to be rewritten to select the external interrupt source of EINT15. The selected external interrupt sources represented by values of the bit are shown in Table 19.
31:15	Reserved		

5.3.6 Configuration register 2 (SYSCFG_CFG2)

Offset address: 0x18

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	LOCK	R/W	Core LOCKUP Enable This bit is set by software and cleared by system reset. It can enable and lock the connection between Arm® Crotex®-M0+ LOCKUP Hardfault (hardware error) output and TMR1 brake input.
1	Reserved		
2	PVDLOCK	R/W	PVD Lock Enable It can enable and lock the connection between PVD interrupt and TMR1 brake input and lock the state of PVDEN bit and PLSEL bit in PMU_CTRL. 0: Connection locked; PVDEN bit and PLSEL bit are editable 1: Connection enabled; PVDEN bit and PLSEL bit are read-only
31:3	Reserved		

5.3.7 Data scrambling key register (SYSCFG_DCODE)

Offset address: 0x20

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	DKEY	W	Data Code Cipher Key

5.3.8 Instruction scrambling key register (SYSCFG_ICODE)

Offset address: 0x24

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	IKEY	W	Instruction Code Cipher Key

5.3.9 Scrambling enable register (SYSCFG_CIPHEN)

Offset address: 0x28

Reset value: 0x0000 0004

Field	Name	R/W	Description
0	DCIPHEN	R/W	Instruction Code Cipher Enable 0: Disable 1: Enable
1	ICIPHEN	R/W	Instruction Code Cipher Enable 0: Disable 1: Enable
31:2	Reserved		

5.3.10 Processor and DMA access IP permission register (SYSCFG_ACCESSEN)

Offset address: 0x2C

Reset value: 0x17FF EEC9

Field	Name	R/W	Description
0	DMAEN	R/W	CPU Access DMA Enable 0: Disable 1: Enable
2:1	Reserved		
3	CRCEN	R/W	CPU and DMA Access CRC Enable 0: Disable 1: Enable
5:4	Reserved		
6	ADCEN	R/W	CPU and DMA Access ADC Enable 0: Disable 1: Enable
7	CANEN	R/W	CPU and DMA Access CAN Enable 0: Disable 1: Enable
8	Reserved		
9	EINTEN	R/W	CPU and DMA Access EINT Enable 0: Disable 1: Enable
10	I2CEN	R/W	CPU and DMA Access I2C Enable 0: Disable 1: Enable

Field	Name	R/W	Description
11	IWDTEN	R/W	CPU and DMA Access IWDT Enable 0: Disable 1: Enable
12	Reserved		
13	RTCEN	R/W	CPU and DMA Access RTC Enable 0: Disable 1: Enable
14	SPIEN	R/W	CPU and DMA Access SPI Enable 0: Disable 1: Enable
15	TMR1EN	R/W	CPU and DMA Access TMR1 Enable 0: Disable 1: Enable
16	TMR2EN	R/W	CPU and DMA Access TMR2 Enable 0: Disable 1: Enable
17	TMR3EN	R/W	CPU and DMA Access TMR3 Enable 0: Disable 1: Enable
18	TMR4EN	R/W	CPU and DMA Access TMR4 Enable 0: Disable 1: Enable
19	TMR6EN	R/W	CPU and DMA Access TMR6 Enable 0: Disable 1: Enable
20	TMR7EN	R/W	CPU and DMA Access TMR7 Enable 0: Disable 1: Enable
21	USART1EN	R/W	CPU and DMA Access USART1 Enable 0: Disable 1: Enable
22	USART2EN	R/W	CPU and DMA Access USART2 Enable 0: Disable 1: Enable
23	WWDTEN	R/W	CPU and DMA Access WWDT Enable 0: Disable 1: Enable
24	GPIOAEN	R/W	CPU and DMA Access GPIOA Enable 0: Disable 1: Enable
25	GPIOBEN	R/W	CPU and DMA Access GPIOB Enable 0: Disable 1: Enable

Field	Name	R/W	Description
26	GPIOCEN	R/W	CPU and DMA Access GPIOC Enable 0: Disable 1: Enable
27	Reserved		
28	GPIOFEN	R/W	CPU and DMA Access GPIOF Enable 0: Disable 1: Enable
31:29	Reserved		

6 Reset and clock management (RCM)

6.1 Full name and abbreviation of terms

Table 20 Full Name and Abbreviation 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

6.2 Reset functional description

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

6.2.1 System reset

6.2.1.1 "System reset" reset source

The reset source can be 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)

- Low-power management reset
- Load option byte reset
- Power reset

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_CSTS (control/status register).

Generally speaking, when the system is reset, the values of all registers except the reset flag bit of RCM_CSTS and the register in RTC domain will be reset to the reset value.

Software Reset

Software can be reset by setting SYSRESETREQ in Arm® Cortex®-M0+ interrupt application and reset control register to "1".

Low-power management reset

Low-power management may reset in two cases, one is when entering the standby mode, and the other is when entering the stop mode. In these two cases, if RSTSTDB bit (in standby mode) or RSTSTOP bit (in stop mode) in user-selected byte is cleared, the system will be reset and not enter the standby or stop mode.

For more information about user-selected bytes, refer to the chapter of "Flash memory".

Load option byte reset

The load byte reset is triggered by OBLOAD bit in FMC_CTRL2 register which is controlled by software.

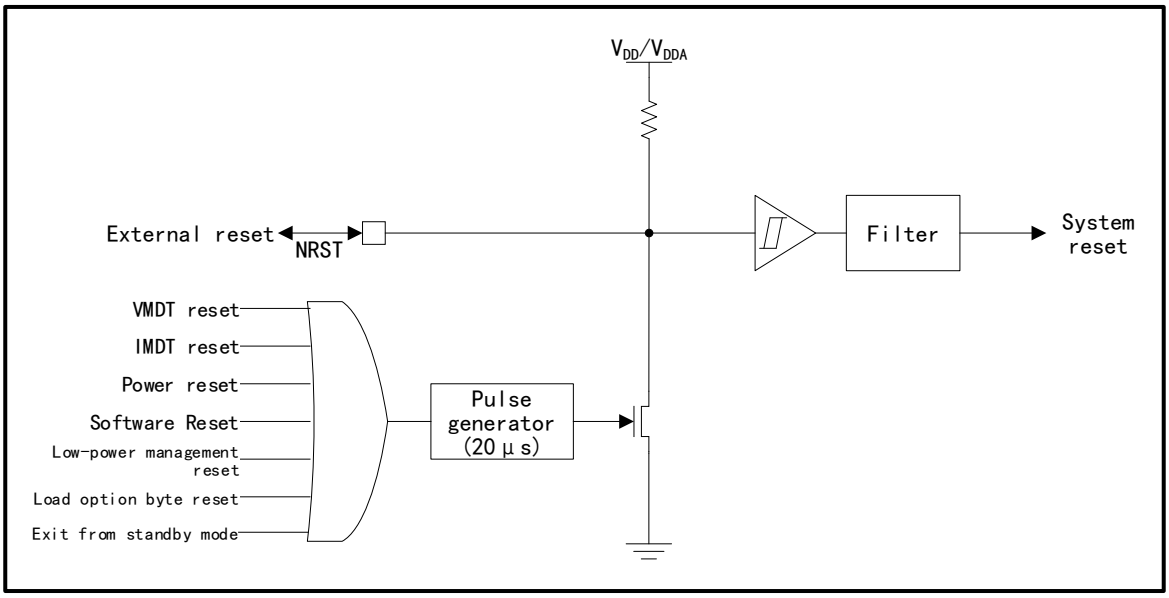
6.2.1.2 "System reset" reset circuit

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 20μs 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 4 "System Reset" Reset Circuit



6.2.2 Power reset

"Power reset" reset source is as follows:

- Power-on (POR reset)
- Power-down reset (PDR reset)
- Wake up from standby mode

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

Power reset will reset all registers except RTC.

6.2.3 RTC domain reset

"RTC domain reset" reset source is as follows:

- Software reset triggered by resetting RTCRST bit in RCM_RTCCTRL
- V_{DD} power-down

A RTC domain reset will occur when any of the above events occurs. Note that the RTC domain is also powered by V_{DD}.

RTC area reset only affects RTC area.

The RTC domain register can also be reset by any of the following events:

- RTC modification detected
- Read protection level changed from Level 1 to Level 0

6.3 Functional description of clock management

The clock sources of the whole system are: HSECLK, LSECLK, HSICLK, HSICLK14, LSICLK and PLL. For the characteristics of the clock source, please refer to the relevant chapter of "Electrical Characteristics" in the datasheet.

6.3.1 External clock source

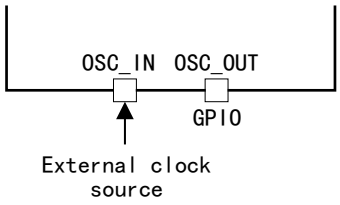
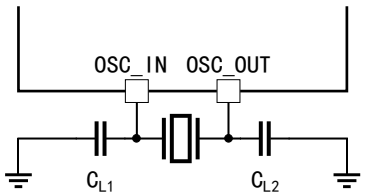
The external clock signal includes HSECLK (high-speed external clock signal) and LSECLK (low-speed external clock signal).

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 5 HSECLK/LSECLK Clock Source Hardware Configuration

Clock Source	Hardware configuration
External clock	
Crystal/ceramic resonator	

Note:

（1） Adjust the value of load capacitance (CL1, CI2) according to the selected oscillator.

（2） 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.

6.3.1.1 HSECLK high-speed external clock signal

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

Table 21 Clock Source Generating HSECLK

Name	Description
External clock source (HSECLK bypass)	Provide clock to the MCU through 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 40%-60% duty cycle, and the maximum frequency is up to 32MHz. In hardware connection, it should be connected to OSC_IN pin, ensuring OSC_OUT pin is suspended; in MCU configuration, the user can select this mode by setting HSEBCFG and HSEEN bits in RCM_CTRL1 (clock control register 1).

Name	Description
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 4-32MHz. When needing to connect OSC_IN and OSC_OUT to the resonator, it can be enabled and disabled by setting the HSEEN bit in clock control register RCM_CTRL1 (clock control register). HSERDYFLG bit in the clock control register RCM_CTRL1 (clock control register 1) 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 interrupt is allowed in RCM_INT (clock interrupt register), corresponding interrupt will be generated.

6.3.1.2 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 22 Clock Source 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 OSC32_IN pin, ensuring OSC32_OUT pin is suspended; for MCU configuration, the user can select this mode by setting LSEBCFG and LSEEN bits in RCM_RTCCTRL.
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. OSC32_IN and OSC32_OUT needs to be connected to the oscillator, which can be enabled and disabled by LSEEN bit in RCM_RTCCTRL. LSERDYFLG in RCM_RTCCTRL indicates whether LSECLK crystal oscillator is stable. At startup stage, LSECLK clock signal is not released until this bit is set to "1" by hardware. If it is allowed in the clock interrupt register, an interrupt request can be generated.

6.3.2 Internal clock source

The internal clock includes HSICLK (high-speed internal clock signal) and LSICLK (low-speed internal clock signal). HSICLK has 8MHz and 14MHz two specifications, and HSICLK14 is mainly used to provide clock signals to ADC.

6.3.2.1 HSICLK high-speed internal clock signal

HSICLK clock signal is generated by internal 8MHz RC oscillator.

The RC oscillator frequency of different chips is different, and that of the same chip may 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_CTRL1 (clock control register); in addition, the users can further adjust the frequency by setting HSITRM in RCM_CTRL1 according to the application environment (temperature and voltage) of the site.

HSIRDYFLG bit can be used to indicate whether HSICLK RC oscillator is stable. In the clock startup process, HSICLK RC output clock is not released until the HSIRDYFLG bit is set to "1" by hardware. HSICLK RC oscillator can be enabled or disabled by HSIEN bit in RCM_CTRL1.

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.

6.3.2.2 HSICLK14 high-speed internal clock signal

HSICLK14 clock signal is generated by the internal 14MHz RC oscillator and is mainly used to provide the clock signals to ADC. HSI14RDYFLG bit can be used to indicate whether HSICLK14 RC oscillator is stable. In the clock startup process, HSICLK14 RC output clock is not released until the HSI14RDYFLG bit is set to "1" by hardware. HSICLK14 RC oscillator can be enabled or disabled by HSI14EN bit in RCM_CTRL2.

Users can further adjust the frequency by setting HSI14TRM in RCM_CTRL2 according to the application environment (temperature and voltage) of the site.

6.3.2.3 LSICLK low-speed internal clock signal

Main characteristics of LSICLK

LSICLK is generated by RC oscillator, within the range of 40kHz (between 30kHz and 60kHz). The frequency may change along with the change of temperature and voltage. It can keep running in stop and standby mode and provide clock for IWDT (independent watchdog) and RTC (real-time clock).

LSICLK can be enabled or disabled by LSIEN bit of RCM_CSTS (control/status register). LSIRDYFLG bit in RCM_CSTS indicates whether the low-speed internal oscillator is stable. At startup stage, the clock is not released until this bit is set to "1" by hardware. If it is allowed in RCM_INT (clock interrupt register), LSICLK interrupt request will be generated.

6.3.3 PLL (phase locked loop)

The internal PLL can be used to double the frequency of HSICLK output clock or HSECLK crystal output clock.

To configure PLL parameters, first clear PLLEN bit, and after PLLRDYFLG is cleared (PLL is in the disabled state), change the parameters, then set PLLEN to 1, enable PLL, and when PLLRDYFLG is set to 1, the configuration is

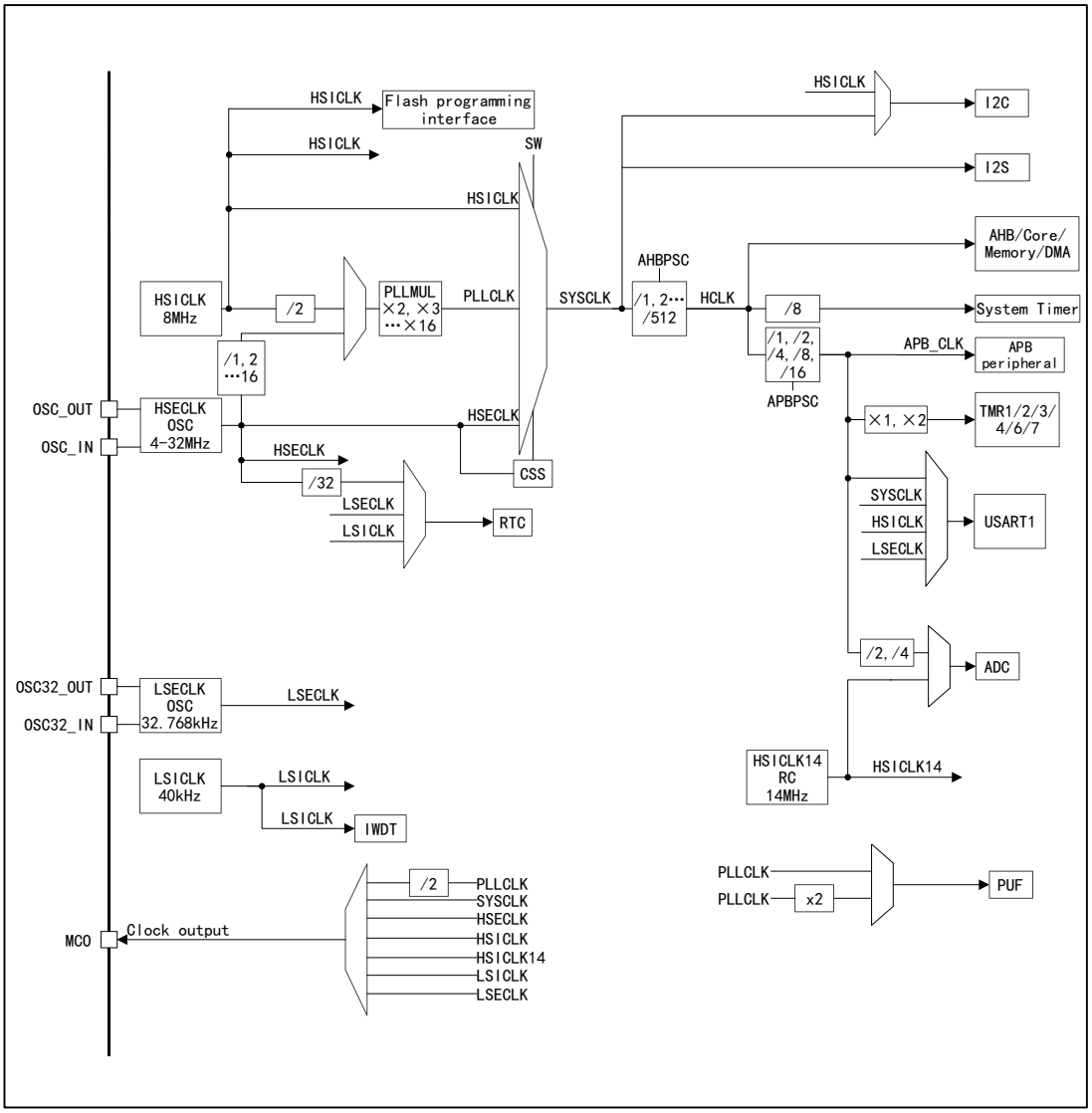
completed.

The clock source and multiplication factor should be selected before activated. Once PLL is activated, the selection cannot be changed.

When PLL is ready and PLL interrupt in RCM_INT is allowed, PLL can transmit interrupt request.

6.3.4 Clock tree

Figure 6 APM32F035x8T7 Clock Tree



Note:

（1） HCLK means AHB clock.

（2） PCLK is clock signal of the peripheral connected to APB.

（3） FCLK is running clock of Arm® Cortex®-M0+.

（4） The frequency of AHB and APB domains can be configured by multiple prescalers

（5） When needing to run the peripheral connected to AHB and APB, it is required to enable the corresponding enable end to make the peripheral get the clock signal.

（6） Frequency assignment of all TMRxCLK (timer clocks) is automatically set by hardware according to the following two situations:

- If the corresponding APB prescaler factor is 1, the clock frequency of the timer is the same as the frequency of the APB bus.
- Otherwise, the clock frequency of the timer will be set to twice the frequency of the APB bus connected to it.

（7） The frequency of TMRx (x=1, 2, 3, 4, 6, 7) clock signals is divided by APB.

（8） HSICLK14 RC oscillator is used to provide the ADC with a clock (see the chapter of "ADC Clock Source Selection" for details).

6.3.5 Clock source selection of RTC

HSECLK/32, LSECLK or LSICLK can be selected as RTCCLK clock source by setting RTCSRCSEL bit of RCM_RTCCTRL. The selection of clock source can be changed only when the RTC domain is reset. Only when PCLK is greater than or equal to RTCCLK, can the system operate RTC normally.

6.3.6 Clock source selection of IWDT

When IWDT (independent watchdog) is enabled, LSICLK oscillator will be enabled by force, and when it is stable, it will provide the clock signal to IWDT. After LSICLK is enabled by force, it will always be enabled and cannot be disabled.

6.3.7 Clock source selection of MCO

When the corresponding GPIO port register is configured with corresponding function, the clock signal can be selected to be output to MCO pin by configuring MCOSEL in RCM_CFG1 (clock configuration register). See the instructions for clock tree or MCOSEL bit of RCM_CFG1 register for specific clock signals.

6.3.8 Clock source selection of SYSCLK

SYSCLK clock source can be HSECLK, PLLCLK or HSICLK.

The status bit of RCM_CFG1 can indicate the ready clock and selected SYSCLK clock source.

When the system is reset, HSICLK oscillator is selected as the system clock, and the clock source cannot be stopped when PLL is directly or indirectly used as the system clock. If you want to switch the SYSCLK clock source, you must wait until the target clock source is ready (i.e. the target clock source is stable).

6.3.9 CSS clock security system

In order to prevent MCU from failing to to run normally due to short circuit of external crystal oscillator, MCU can activate CSS clock security system by software. After the security system is activated, if the HSECLK oscillator is used as the system clock directly or indirectly (used as the PLL input clock and PLL is

the system clock), the external HSECLK oscillator will be disabled when the HSECLK clock fails, and the system clock will automatically switch to HSICLK. At this time, the PLL which selects HSECLK as the clock input and as the system clock input source will also be disabled.

CSS can be activated by software. When HSECLK clock fails, CSS interrupt will be generated, and NMI will be generated automatically. NMI will be executed continuously until the CSS interrupt pending bit is cleared. Therefore, CSS interrupt must be cleared by setting CSSCLR bit of RCM_INT (clock interrupt register) in NMI processing program.

6.3.10 Clock source selection of ADC

The clock source of ADC is controlled by ADC_CFG2. It can select HSICLK14 or PCLK with the frequency divided by 2/4 as the clock source. When PCLK is used as the clock source of ADC, HSICLK14 cannot be changed over to ADC interface.

6.3.11 Low-power mode

PCLK and DMACLK can be disabled by software.

Sleep mode:

- Stop CPU clock
- Flash and RAM interface clocks can be stopped by software
- When all peripheral clocks connected to APB bus are disabled, the AHb1/APB bridge clocks can be stopped by hardware

Stop mode and standby mode:

- All 1.5V power domains are disabled
- PLLCLK, HSICLK, HSICLK14 and HSECLK are disabled

Deep sleep mode:

- The system can be debugged by setting STOP_CLK_STS bit and STANDBY_CLK_STS bit in DBGMCU_CFG.
- The system selects HSICLK as SYSCLK by interrupt (in stop mode) or reset (standby mode)
- If Flash programming is in progress, the system will enter the deep sleep mode only after all programming operations are completed
- If APB domain is being used, the system will enter the deep sleep mode only after all operations are completed

6.4 Register address mapping

Table 23 RCM Register Address Mapping

Register Name	Description	Offset address
RCM_CTRL1	Clock control register 1	0x00

Register Name	Description	Offset address
RCM_CFG1	Clock configuration register 1	0x04
RCM_INT	Clock interrupt register	0x08
RCM_APBRST2	APB2 peripheral reset register	0x0C
RCM_APBRST1	APB1 peripheral reset register	0x10
RCM_AHBCLKEN	AHB peripheral clock enable register	0x14
RCM_APBCLKEN2	APB2 peripheral clock enable register	0x18
RCM_APBCLKEN1	APB1 peripheral clock enable register	0x1C
RCM_RTCCTRL	RTC domain control register	0x20
RCM_CSTS	Control/Status register	0x24
RCM_AHBRST	AHB peripheral reset register	0x28
RCM_CFG2	Clock configuration register 2	0x2C
RCM_CFG3	Clock configuration register 3	0x30
RCM_CTRL2	Clock control register 2	0x34

6.5 Register functional description

6.5.1 Clock control register 1 (RCM_CTRL1)

Offset address: 0x00

Reset value: 0x0000 XX83; X means undefined

Access: Access in the form of word, half word and byte, without wait cycle

Field	Name	R/W	Description
0	HSIEN	R/W	High Speed Internal Clock Enable Set 1 or clear 0 by software. HSICLK is an RC oscillator. When one of the following conditions occurs, it will be set to 1 by hardware: power-on start, software reset, wake-up from standby mode, wake-up from stop mode, failure of external high-speed clock source (as system clock or providing system clock through PLL). When HSICLK is used as system clock or provides system clock through PLL, this bit cannot be cleared to 0. 0: HSICLK RC oscillator is disabled 1: HSICLK RC oscillator is enabled
1	HSIRDYFLG	R	High Speed Internal Clock Ready Flag 0: HSICLK RC oscillator is not stable 1: HSICLK RC oscillator is stable
2	Reserved		
7:3	HSITRM	R/W	High Speed Internal Clock Trim The product has been calibrated to 8MHz±1% when leaving the factory. However, it changes as the temperature and voltage changes, but the frequency of HSICLK RC oscillator can be adjusted by HSITRM[4:0].

Field	Name	R/W	Description
15:8	HSICAL	R	High Speed Internal Clock Calibrate It will be calibrated to 8MHz±1% before leaving the factory. When the system is started up, the calibration parameters will be automatically written to the register.
16	HSEEN	R/W	High Speed External Clock Enable When entering the standby or stop mode, this bit is cleared to 0 by hardware and HSECLK is disabled; when HSECLK is used as system clock source or the system clock is provided through PLL, this bit cannot be cleared to 0. 0: HSECLK is disabled 1: HSECLK is enabled
17	HSERDYFLG	R	High Speed External Clock Ready Flag When HSECLK is stable, this bit is set to 1 by hardware and cleared to 0 by software. 0: HSECLK is not stable 1: HSECLK is stable
18	HSEBCFG	R/W	High Speed External Clock Bypass Configure Bypass mode refers to the mode in which external clock is used as the HSECLK clock source; otherwise, the resonator is used as the HSECLK clock source. 0: Non-bypass mode 1: Bypass mode
19	CSSEN	R/W	Clock Security System Enable 0: Disable 1: Enable
23:20	Reserved		
24	PLLEN	R/W	PLL Enable When entering the standby and stop mode, this bit is cleared to0 by hardware; when PLLCLK has been configured (or in the process of configuration) as the clock source of the system clock, this bit cannot be cleared to 0; in other cases, it can be set to 1 or cleared to 0 by software. 0: PLL is disabled 1: PLL is enabled
25	PLLRDYFLG	R	PLL Clock Ready Flag PLL is set to 1 by hardware after it is locked. 0: PLL is unlocked 1: PLL is locked
31:	Reserved		

6.5.2 Clock configuration register (RCM_CFG1)

Offset address: 0x04

Reset value: 0x0000 0000

All bits of this register are set or cleared to 0 by software.

Access: Access in the form of word, half word and byte, with 0 to 2 wait cycles. 1 or 2 wait cycles are inserted only when the access occurs during clock switching.

Field	Name	R/W	Description
1:0	SCLKSEL	R/W	System Clock Source Select When returning from stop or standby mode or the HSECLK directly or indirectly used as system clock fails, the hardware selects HSICLK as system clock by force (if the clock security system has been started) 00: HSICLK is used as system clock 01: HSECLK is used as system clock 10: PLLCLK is used as system clock 11: Reserved
3:2	SCLKSELSTS	R	System Clock Selection Status Indicate which clock source is used as system clock; set to 1 or clear to 0 by hardware. 00: HSICLK is used as system clock 01: HSECLK is used as system clock 10: PLLCLK output is used as system clock 11: Unavailable
7:4	AHBPSC	R/W	AHB Clock Prescaler Factor Configure Control the prescaler factor of AHB clock. 0xxx: No frequency division for SYSCLK 1000: SYSCLK two-divided frequency 1001: SYSCLK four-divided frequency 1010: SYSCLK eight-divided frequency 1011: SYSCLK 16-divided frequency 1100: SYSCLK 64-divided frequency 1101: SYSCLK 128-divided frequency 1110: SYSCLK 256-divided frequency 1111: SYSCLK 512-divided frequency
10:8	APB1PSC	R/W	APB1 Clock Prescaler Factor Configure Control the prescaler factor of low-speed APB1 clock (PCLK1) 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
13:11	Reserved		
14	ADCPSC	R/W	ADCCLK Prescaler Factor Configure It is determined by the corresponding bit of ADC configuration register.
15	Reserved		
16	PLLSRCSEL	R/W	PLL Clock Source Select This bit can be changed only when PLL is disabled. 0: HSICLK 2 is used as PLL clock source after frequency division 1: HSECLK is used as PLL clock source after frequency division
17	PLLHSEPSC	R/W	HSECLK Prescaler Factor for PLL Clock Source Refer to the 0 bit of RCM_CFG2.

Field	Name	R/W	Description
21:18	PLLMULCFG	R/W	PLL Multiplication Factor Configure Determine PLL multiplication factor. This bit can be written only when PLL is disabled. 0000: PLLCLK 2-multiple frequency output 0001: PLL 3-multiple frequency output 0010: PLL 4-multiple frequency output 0011: PLL 5-multiple frequency output 0100: PLL 6-multiple frequency output 0101: PLL 7-multiple frequency output 0110: PLL 8-multiple frequency output 0111: PLL 9-multiple frequency output 1000: PLL 10-multiple frequency output 1001: PLL 11-multiple frequency output 1010: PLL 12-multiple frequency output 1011: PLL 13-multiple frequency output 1100: PLL 14-multiple frequency output 1101: PLL 15-multiple frequency output 1110: PLL 16-multiple frequency output 1111: PLL 16-multiple frequency output Note: The output frequency of PLL cannot be greater than 48MHz.
23:22	Reserved		
26:24	MCOSEL	R/W	Main Clock Output Select Set or clear 0 by software. 000: No clock output 0001: HSICLK14 is output as a clock 0010: LSICLK is output as a clock 01: LSECLK is output as a clock 100: SYSCLK is output as a clock 101: HSICLK is output as a clock 110: HSECLK is output as a clock 111: PLLCLK is output as a clock after two divided frequency
31:27	Reserved		

6.5.3 Clock interrupt register (RCM_INT)

Offset address: 0x08

Reset value: 0x0000 0000

Access: Access in the form of word, half word and byte, without wait cycle.

Field	Name	R/W	Description
0	LSIRDYFLG	R	LSICLK Ready Interrupt Flag When LSICLK is stable and LSIRDYEN bit is set to 1, set 1 by hardware; set 1 by software and clear 0 by LSIRDYCLR. 0: No LSICLK ready interrupt 1: LSICLK ready interrupt occurred

Field	Name	R/W	Description
1	LSERDYFLG	R	LSECLK Ready Interrupt Flag When LSECLK is stable and LSERDYEN bit is set to 1, set 1 by hardware; set 1 by software and clear 0 by LSERDYCLR. 0: No LSECLK ready interrupt 1: LSECLK ready interrupt occurred
2	HSIRDYFLG	R	HSICLK Ready Interrupt Flag When HSICLK is stable and HSIRDYEN bit is set to 1, set 1 by hardware; set 1 by software and clear 0 by HSIRDYCLR. 0: No HSICLK ready interrupt 1: HSICLK ready interrupt occurred
3	HSERDYFLG	R	HSECLK Ready Interrupt Flag When HSECLK is stable and HSERDYEN bit is set to 1, set 1 by hardware; set 1 by software and clear 0 by HSERDYCLR. 0: No HSECLK ready interrupt 1: HSECLK ready interrupt occurred
4	PLLRDYFLG	R	PLL Ready Interrupt Flag When PLL is stable and PLLRDYEN bit is set to 1, set 1 by hardware; set 1 by software and clear 0 by PLLRDYCLR. 0: No clock ready interrupt caused by PLL locked 1: Clock ready interrupt caused by PLL locked
5	HSI14RDYFLG	R	HSICLK14 Ready Interrupt Flag When the internal high-speed clock is ready and HSI14RDYEN bit is set to 1, set 1 by hardware. Set 1 by software and clear 0 by HSI14RDYCLR. 0: No security system interrupt caused by HSECLK failure 1: Security system interrupt is caused by HSECLK failure
6	Reserved		
7	CSSFLG	R	Clock Security System Interrupt Flag When the external 4-16MHz oscillator clock fails, set 1 by hardware. Set 1 by software, and clear 0 by 1CSSCLR. 0: No security system interrupt caused by HSE clock failure 1: Clock security system interrupt is caused by HSE clock failure
8	LSIRDYEN	R/W	LSICLK Ready Interrupt Enable Enable or disable internal 40kHz RC oscillator ready interrupt. 0: Disable 1: Enable
9	LSERDYEN	R/W	LSECLK Ready Interrupt Enable Enable external 32kHz RC oscillator ready interrupt. 0: Disable 1: Enable
10	HSIRDYEN	R/W	HSICLK Ready Interrupt Enable Enable internal 8MHz RC oscillator ready interrupt. 0: Disable 1: Enable.

Field	Name	R/W	Description
11	HSERDYEN	R/W	HSCLKE Ready Interrupt Enable Enable external 4-16MHz oscillator ready interrupt. 0: Disable 1: Enable
12	PLLRDYEN	R/W	PLL Ready Interrupt Enable Enable PLL ready interrupt. 0: Disable 1: Enable
13	HSI14RDYEN	R/W	HSICLK14 Ready Interrupt Enable Enable internal 14MHz RC oscillator ready interrupt. 0: Disable 1: Enable.
15:14	Reserved		
16	LSIRDYCLR	W	LSICLK Ready Interrupt Clear Clear LSI ready interrupt flag bit LSIRDYFLG. 0: No effect 1: Clear
17	LSERDYCLR	W	Ready Interrupt Clear Clear LSE ready interrupt flag bit LSERDYFLG. 0: No effect 1: Clear
18	HSIRDYCLR	W	HSICLK Ready Interrupt Clear Clear HSI ready interrupt flag bit HSIRDYFLG. 0: No effect 1: Clear
19	HSERDYCLR	W	HSECLK Ready Interrupt Clear Clear HSE ready interrupt flag bit HSERDYFLG. 0: No effect 1: Clear
20	PLLRDYCLR	W	PLL Ready Interrupt Clear Clear PLL ready interrupt flag bit PLLRDYFLG. 0: No effect 1: Clear
21	HSI14RDYCLR	W	HSICLK14 Ready Interrupt Clear Clear HSICLK14 ready interrupt flag bit HSI14RDYFLG. 0: No effect 1: Clear
22	Reserved		
23	CSSCLR	W	Clock Security System Interrupt Clear Clear the security system interrupt flag bit CSSFLG. 0: No effect 1: Clear
31:24	Reserved		

6.5.4 APB peripheral reset register 2 (RCM_APBRST2)

Offset address: 0x0C

Reset value: 0x0000 0000

Access: Access in the form of word, half word and byte, without wait cycle.

All bits can be set or cleared to 0 by software.

Field	Name	R/W	Description
0	SYSCFGRST	R/W	SYSCFG Reset 0: No effect 1: Reset
8:1	Reserved		
9	ADCRST	R/W	ADC Reset 0: No effect 1: Reset ADC
10	Reserved		
11	TMR1RST	R/W	TMR1 Timer Reset 0: No effect 1: Reset
12	SPIRST	R/W	SPI Reset 0: No effect 1: Reset
13	Reserved		
14	USART1RST	R/W	USART1 Reset 0: No effect 1: Reset
15	Reserved		
16	TMR7RST	R/W	TMR7 Reset 0: No effect 1: Reset
21:17	Reserved		
22	DBGRST	R/W	Debug Reset 0: No effect 1: Reset
31:23	Reserved		

6.5.5 APB peripheral reset register 1 (RCM_APBRST1)

Offset address: 0x10

Reset value: 0x0000 0000

Access: Access in the form of word, half word and byte, without wait cycle

Field	Name	R/W	Description
0	TMR2RST	R/W	Timer 2 Reset 0: No effect 1: Reset

Field	Name	R/W	Description
1	TMR3RST	R/W	Timer 3 Reset 0: No effect 1: Reset
3:2	Reserved		
4	TMR6RST	R/W	Timer 6 Reset 0: No effect 1: Reset
7:5	Reserved		
8	TMR14RST	R/W	Timer 14 Reset 0: No effect 1: Reset
10:9	Reserved		
11	WWDTRST	R/W	Window Watchdog Reset 0: No effect 1: Reset
16:12	Reserved		
17	USART2RST	R/W	USART2 Reset 0: No effect 1: Reset
20:18	Reserved		
21	I2CRST	R/W	I2C Reset 0: No effect 1: Reset
24:22	Reserved		
25	CANRST	R/W	CAN Reset 0: No effect 1: Reset
27:26	Reserved		
28	PMURST	R/W	Power Interface Reset 0: No effect 1: Reset
31:29	Reserved		

6.5.6 AHB peripheral clock enable register (RCM_AHBCLKEN)

Offset address: 0x14

Reset value: 0x0000 0014

Access: Access in the form of word, half word and byte, without wait cycle

All bits can be set or cleared to 0 by software.

Note: When the peripheral clock is not enabled, the software cannot read the value of the peripheral register, and the value returned is always 0x0.

Field	Name	R/W	Description
0	DMAEN	R/W	DMA Clock Enable 0: Disable 1: Enable
1	Reserved		
2	SRAMEN	R/W	SRAM Interface Clock Enable Enable SRAM clock in sleep mode. 0: Disable 1: Enable
3	Reserved		
4	FMCEN	R/W	FMC Clock Enable Enable the flash interface circuit clock in sleep mode. 0: Disable 1: Enable
5	Reserved		
6	CRCEN	R/W	CRC Clock Enable 0: Disable 1: Enable
16:7	Reserved		
17	PAEN	R/W	I/O PortA Clock Enable 0: Disable 1: Enable
18	PBEN	R/W	I/O PortB Clock Enable 0: Disable 1: Enable
19	PCEN	R/W	I/O PortC Clock Enable 0: Disable 1: Enable
21:20	Reserved		
22	PFEN	R/W	I/O PortF Clock Enable 0: Disable 1: Enable
23	Reserved		
24	M0CPEN	R/W	M0CP Clock Enable 0: Disable 1: Enable
31:25	Reserved		

6.5.7 APB peripheral clock enable register 2 (RCM_APBCLKEN2)

Offset address: 0x18

Reset value: 0x0000 0000

Access: Access in the form of word, half word and byte

All bits can be reset or cleared to 0 by software.

Note: When the peripheral clock is not enabled, the software cannot read the value of the peripheral register, and the value returned is always 0x0.

Field	Name	R/W	Description
0	SCFGCOMPEN	R/W	SYSCFG Clock Enable 0: Disable 1: Enable
8:1	Reserved		
9	ADCEN	R/W	ADC Interface Clock Enable 0: Disable 1: Enable
10	Reserved		
11	TMR1EN	R/W	TMR1 Timer Clock Enable 0: Disable 1: Enable
12	SPIEN	R/W	SPI Clock Enable 0: Disable 1: Enable
13	Reserved		
14	USART1EN	R/W	USART1 Clock Enable 0: Disable 1: Enable
15	Reserved		
16	TMR7EN	R/W	TMR7 Timer Clock Enable 0: Disable 1: Enable
21:17	Reserved		
22	DBGEN	R/W	Debug Clock Enable 0: Disable 1: Enable
31:23	Reserved		

6.5.8 APB peripheral clock enable register 1 (RCM_APBCLKEN1)

Offset address: 0x1C

Reset value: 0x0000 0000

Access: Access in the form of word, half word and byte

Usually there is no wait cycle. However, when the peripheral on the APB bus is accessed, the waiting state will be inserted until the APB peripheral access ends.

All bits can be reset or cleared to 0 by software.

Note: When the peripheral clock is not enabled, the software cannot read the value of the peripheral register, and the value returned is always 0x0.

Field	Name	R/W	Description
0	TMR2EN	R/W	TMR2 Timer Clock Enable 0: Disable 1: Enable
1	TMR3EN	R/W	Timer 3 Clock Enable 0: Disable 1: Enable
3:2	Reserved		
4	TMR6EN	R/W	Timer 6 Clock Enable 0: Disable 1: Enable
7:5	Reserved		
8	TMR4EN	R/W	Timer4 Clock Enable 0: Disable 1: Enable
10:9	Reserved		
11	WWDTEN	R/W	Window Watchdog Clock Enable 0: Disable 1: Enable
16:12	Reserved		
17	USART2EN	R/W	USART 2 Clock Enable 0: Disable 1: Enable
20:18	Reserved		
21	I2CEN	R/W	I2C1 Clock Enable 0: Disable 1: Enable
24:22	Reserved		
25	CANEN	R/W	CAN Clock Enable 0: Disable 1: Enable
27:26	Reserved		
28	PMUEN	R/W	Power Interface Clock Enable 0: Disable 1: Enable
31:29	Reserved		

6.5.9 RTC domain control register (RCM_RTCCTRL)

Offset address: 0x20

Reset value: 0x0000 0018, which can be reset effectively only by RTC domain

Access: Access in the form of word, half word and byte, with 0 to 3 wait cycles

When the register is accessed continuously, the waiting state will be inserted.

Note: Only when BPWEN bit in PMU_CTRL is set to 1, can LSEEN, LSEBCFG, RTCSRCSEL and RTCCLKEN be changed.

Field	Name	R/W	Description
0	LSEEN	R/W	Low-Speed External Oscillator Enable 0: Disable 1: Enable
1	LSERDYFLG	R	Low-Speed External Clock Ready Flag Set 1 by hardware when LSECLK is stable, and clear 0 by hardware when it is unstable. 0: Not ready 1. Ready
2	LSEBCFG	R/W	Low-Speed External Clock Bypass Mode Configure Bypass mode refers to the mode in which external clock is used as the LSECLK clock source; otherwise, the resonator is used as the LSECLK clock source. 0: Non-bypass mode 1: Bypass mode
4:3	LSEDRVCFG	R/W	LSE Oscillator Drive Capability Configure Set or clear 0 by software; set the driving capability of LSECLK oscillator (crystal mode is not bypassed). When the RTC domain is reset, this bit is restored to the default value. 00: Weak 01: Medium-low 10: Medium-high 11: Strong
7:5	Reserved		
9:8	RTCSRCSEL	R/W	RTC Clock Source Select First set the RTCRST bit to reset the RTC domain, and then select the RTC clock source. It is impossible to directly configure the register to modify. 00: No clock 01: LSECLK is used as RTC clock 10: LSICLK is used as RTC clock 11: HSECLK is used as RTC clock after 32 divided frequency
14:10	Reserved		
15	RTCCLKEN	R/W	RTC Clock Enable 0: Disable 1: Enable
16	RTCRST	R/W	RTC Domain Software Reset Set 1 or clear o by software 0: Reset is not activated 1: Reset RTC domain (only affecting LSECLK oscillator, RTC clock and register RCM_RTCCTRL)
31:17	Reserved		

6.5.10 Control/Status register (RCM_CSTS)

Offset address: 0x24

Reset value: 0xXXX0 0000, except reset flag, all are cleared by system reset, and reset flag can only be cleared by power reset.

Access: Access in the form of word, half word and byte, with 0 to 3 wait cycles. When the register is accessed continuously, the waiting state will be inserted.

Field	Name	R/W	Description
0	LSIEN	R/W	Low-Speed Internal Oscillator Enable Set 1 or clear 0 by software. 0: Disable 1: Enable
1	LSIRDYFLG	R	Low-Speed Internal Oscillator Ready Flag Set 1 by hardware when LSICLK is stable, and clear 0 by hardware when it is unstable. 0: Not ready 1. Ready
22:2	Reserved		
23	PWRRSTFLG	R	Reset Flag of The 1.5V Domain Set by software and clear by setting RSTFLGCLR.
24	RSTFLGCLR	RT_W	Reset Flag Clear Set or clear reset flag by software, including RSTFLGCLR. 0: No effect 1: Clear reset flag
25	OBRSTFLG	R	Option Byte Loader Reset Flag When the option byte load reset occurs, set by hardware; otherwise, clear by setting RSTFLGCLR. 0: Reset did not occur 1: Reset occurred
26	PINRSTFLG	R	PIN Reset Flag Set by hardware when pin reset occurs; otherwise, clear by setting RSTFLGCLR. 0: Reset did not occur 1: Reset occurred
27	PODRSTFLG	R	POR/PDR Reset Occur Flag Set 1 by hardware; and clear by software by writing RSTFLGCLR bit. 0: No power-on/power-down reset occurs 1: Power-on/power-down reset occurs
28	SWRSTFLG	R	Software Reset Occur Flag Set 1 by hardware; and clear by software by writing RSTFLGCLR bit. 0: Not occur 1: Occurred
29	IWDTRSTFLG	R	Independent Watchdog Reset Occur Flag Set 1 by hardware when independent watchdog reset occurs in V_{DD} area; clear by software by writing RSTFLGCLR bit. 0: Not occur 1: Occurred

Field	Name	R/W	Description
30	WWDTRSTFLG	R	Window Watchdog Reset Occur Flag Set 1 by hardware when window watchdog is reset; clear by software by writing RSTFLGCLR bit. 0: Not occur 1: Occurred
31	LPWRRSTFLG	R	Low Power Reset Occur Flag Set 1 by hardware when low-power management is reset; clear by software by writing RSTFLGCLR bit. 0: Not occur 1: Occurred

6.5.11 AHB peripheral reset register (RCM_AHBRST)

Offset address: 0x28

Reset value: 0x0000 0000

Access: Access in the form of word, half word and byte, without wait cycle

Set 1 or clear 0 by software.

Field	Name	R/W	Description
16:	Reserved		
17	PARST	R/W	I/O Port A Reset 0: Invalid 1: Reset
18	PBRST	R/W	I/O Port B Reset 0: Invalid 1: Reset
19	PCRST	R/W	I/O Port C Reset 0: Invalid 1: Reset
21:20	Reserved		
22	PFRST	R/W	I/O Port F Reset 0: Invalid 1: Reset
23	Reserved		
24	M0CPRST	R/W	M0CP Reset 0: Invalid 1: Reset
31:23	Reserved		

6.5.12 Clock configuration register 2 (RCM_CFG2)

Offset address: 0x2C

Reset value: 0x0000 0000

Access: Access in the form of word, half word and byte, without wait cycle

Field	Name	R/W	Description
3:0	PLLDIVCFG	R/W	PLLCLK Input Division Factor Configure Configure the input clock signal division factor of PLLCLK. 0000: No frequency of division 0001: 2 divided frequency 0010: 3 divided frequency .... 1111: 16 divided frequency
31:4	Reserved		

6.5.13 Clock configuration register 3 (RCM_CFG3)

Offset address: 0x30

Reset value: 0x0000 0000

Access: Access in the form of word, half word and byte, without wait cycle

Field	Name	R/W	Description
1:0	USART1SEL	R/W	USRAT1 Clock Source Select Set or clear 0 by software. The default value is 00. 00: PCLK is used as USART1CLK 01: SYSCLK is used as USART1CLK 10: LSECLK is used as USART1CLK 11: HSICLK is used as USART1CLK
3:2	Reserved		
4	I2CSEL	R/W	I2S Clock Source Select Set or clear 0 by software. The default value is 0. 0: HSICLK is used as I2CCLK 1: SYSCLK is used as I2CCLK
7:5	Reserved		
8	ADCSEL	R/W	ADC Clock Source Select Maintain the reset value, HSICLK14 is used as asynchronous clock input of ADCCLK, and the clock source of ADCCLK is determined by ADC_CFG2.
31:9	Reserved		

6.5.14 Clock control register 2 (RCM_CTRL2)

Offset address: 0x34

Reset value: 0xXX00 XX80; X means undefined

Access: Access in the form of word, half word and byte, without wait cycle

Field	Name	R/W	Description
0	HSI14EN	R/W	HSICLK14 Enable Set 1 or clear 0 by software. 0: Internal 14MHz oscillator disabled 1: Internal 14MHz oscillator enabled

Field	Name	R/W	Description
1	HSI14RDFLG	R	HSICLK14 Ready Flag Set by hardware, indicating the state of HSICLK14 oscillator. 0: Not ready 1: Ready
2	HSI14TO	R/W	ADC Interface Turn On HSICLK14 ADC interface can enable HSICLK14 oscillator; set or clear 0 by hardware. 0: Can enable 1: Cannot enable
7:3	HSI14TRM	R/W	HSICLK14 Trim The product has been calibrated to 14MHz±1% when leaving the factory. However, it changes as the temperature and voltage changes, but the frequency of HSICLK14 RC oscillator can be adjusted by HSI14TRM.
15:8	HSI14CAL	R	HSICLK14 Calibrate It will be calibrated to 14MHz±1% before leaving the factory. When the system is started up, the calibration parameters will be automatically written to the register.
31:16	Reserved		

7 Power management unit (PMU)

7.1 Full name and abbreviation of terms

Table 24 Full Name and Abbreviation Description of Terms

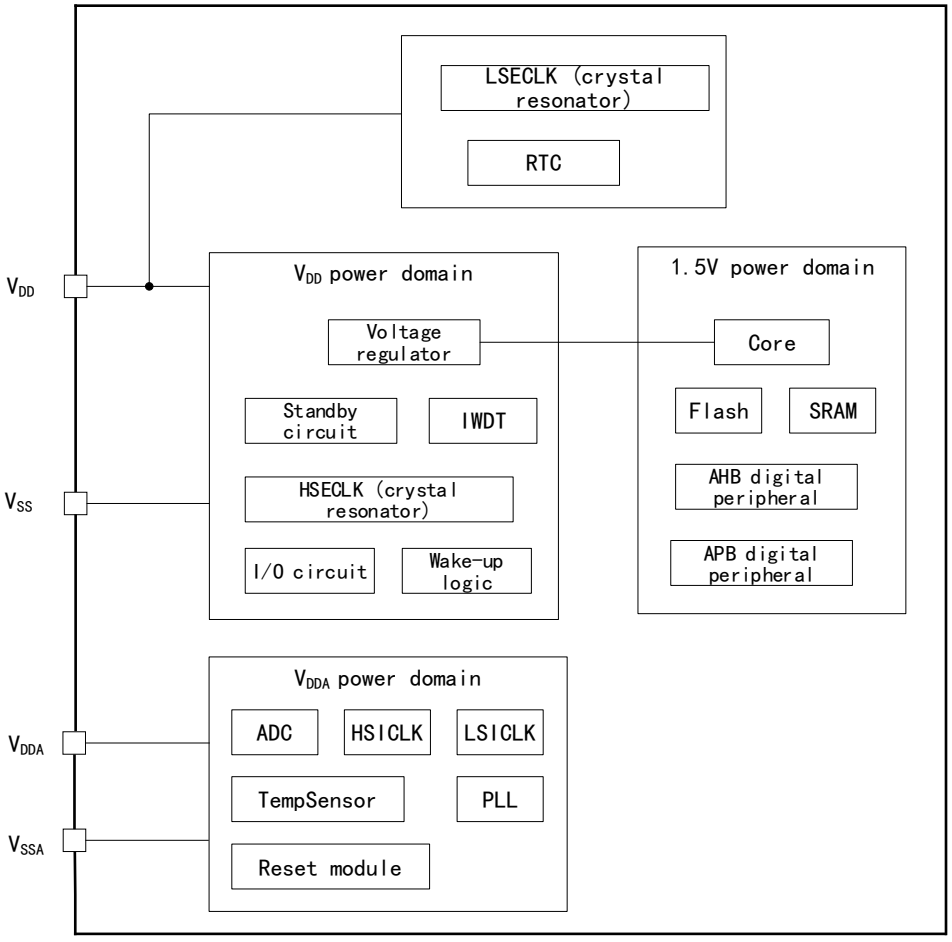
Full name in English	English abbreviation
Power Management Unit	PMU
Power On Reset	POR
Power Down Reset	PDR

7.2 Introduction

The power supply is the foundation for stable operation of a system, with working voltage of 2.0 ~ 3.6V, and 1.5V power supply can be provided by the built-in voltage regulator.

7.3 Structure block diagram

Figure 7 Power Control Structure Block Diagram



7.4 Functional description

7.4.1 Power domain

The power domain of the product includes: V_{DD} power domain, V_{DDA} power domain, and 1.5V power domain.

7.4.1.1 V_{DD} power domain

Power supply is provided through V_{DD}/V_{SS} pins to power the voltage regulator, standby circuit, IWDT, HSECLK, I/O (except PC13, PC14, PC15 pins) and wake-up logic.

Voltage regulator

Power can be supplied to 1.5V power domain in the following operating modes:

- Normal mode: In this mode, 1.5V power supply area runs at full power
- Stop mode: In this mode, 1.5V power supply area works in low-power state, all clocks are off, and peripherals stop work
- Standby mode: In this mode, 1.5V power supply area stops power supply, and except for the standby circuit, the content of register and SRAM will be lost

7.4.1.2 V_{DDA} power domain

Power the ADC, HSICLK, LSICLK, TempSensor, PLL and reset module through V_{DDA}/V_{SSA} pins.

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

7.4.1.3 1.5V power domain

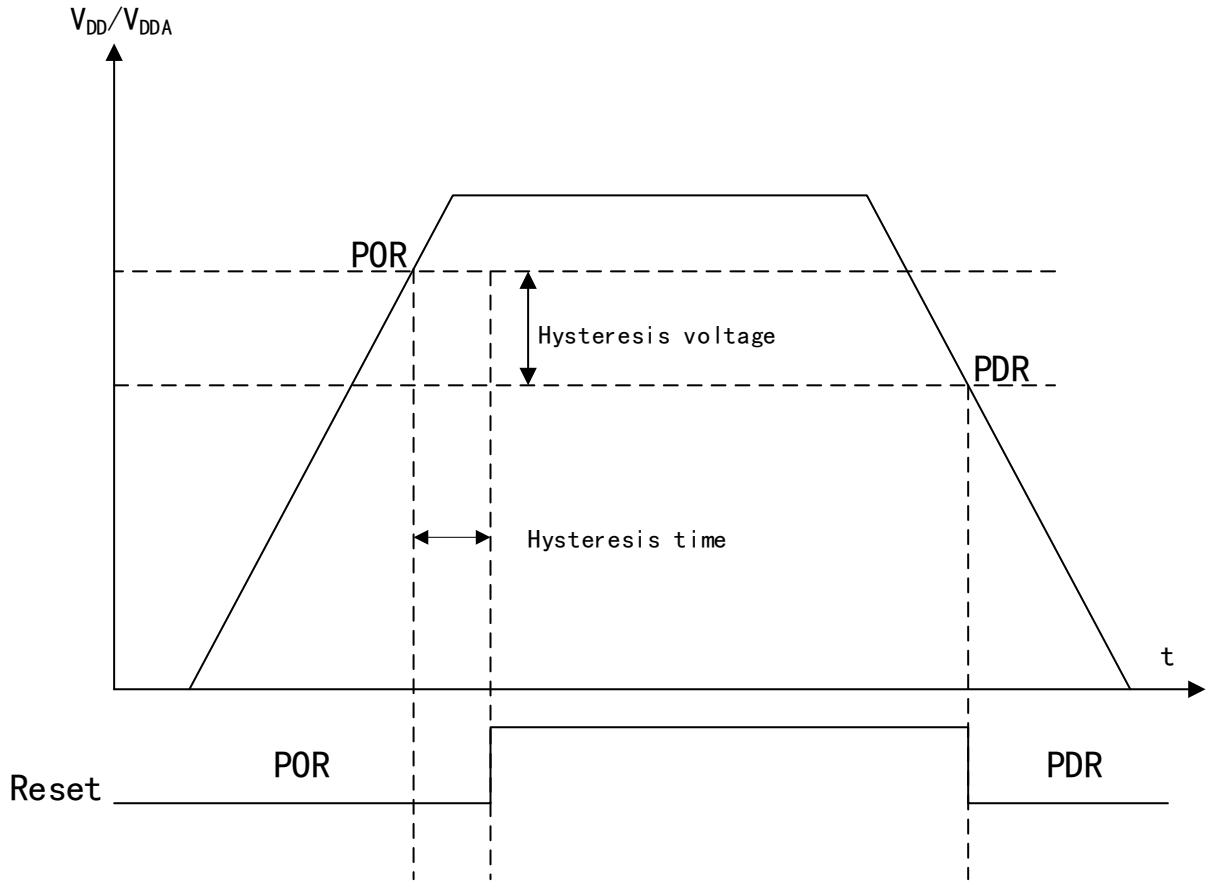
The core, Flash, SRAM and digital peripherals are powered by voltage regulator.

7.4.2 Power Management

7.4.2.1 Power-on/power-down reset (POR and PDR)

When the V_{DD}/V_{DDA} is detected to be lower than the threshold voltage V_{POR} and V_{PDR}, 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 8 Power-on Reset and Power-down Reset Oscillogram



7.4.3 Power control

7.4.3.1 Reduce the power in low-power mode

There are three low-power modes: sleep mode, stop mode and standby mode. The power is reduced by disabling the core and clock source and setting the voltage regulator.

The power consumption, wake-up start time, wake-up mode and data storage after wake-up of each low-power mode are different; the lower the power consumption is, the longer the wake-up time is, the less the wake-up mode is, the less the data saved are after wake-up; users can choose the most appropriate low-power mode according to their needs. The following table shows the difference among three low-power modes.

Table 25 Difference among "Sleep Mode, Stop Mode and Standby Mode"

Mode	Description	Entry method	Wake-up mode	Voltage regulator	Effect on 1.5V area clock	Effect on V_{DD} area clock
Sleep	Arm® Cortex®-M0+ core stops,	Call WFI instruction	Any interrupt	On	Only the core clock is	N/A

Mode	Description	Entry method	Wake-up mode	Voltage regulator	Effect on 1.5V area clock	Effect on V_{DD} area clock
	and all peripherals including the core peripheral are still working	Call WFE instruction	Wake-up event	On	disabled and it has no effect on other clocks and ADC clocks	N/A
Stop	All clocks have stopped	PDDSCFG and LPDSCFG bits +SLEEPDEEP bit +WFI or WFE	Any external interrupt	Enable or be in low-power mode	Disable all clocks of 1.5V area	HSICLK and HSECLK oscillators are disabled
Standby	1.5V power off	PDDSCFG bit +SLEEPDEEP bit +WFI or WFE	Rising edge of WKUP pin, RTC alarm event, external reset on NRST pin, IWDT reset	OFF		

Sleep mode

The characteristics of sleep mode are shown in the table below

Table 26 Characteristics of Sleep Mode

Characteristics	Description
Enter	Enter the sleep mode immediately by executing WFI or WFE instructions; When SLEEPONEXIT is set to 0 and WFI or WFE instruction is executed, the system will enter the sleep mode immediately; when SLEEPONEXIT is set to 1, the system will exit the interrupt program and then enter the sleep mode immediately.
Wake-up	If WFI instruction is executed to enter the sleep mode, wake up by any interrupt; if WFE instruction is executed to enter the sleep mode, wake up by an event.
When entering sleep mode	The core stops working, all peripherals are still running, and the data in the core registers and memory before sleep are saved.
Wakeup delay	N/A
After wake-up	If waking up by interrupt, first enter the interrupt, exit the interrupt, and then execute the program after WFI instruction. If waking up by event, directly execute the program after WFE instruction.

Stop mode

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

Table 27 Characteristics of Stop Mode

Characteristics	Description
Enter	SLEEPDEEP bit of the core register is set to 1, PDDSCFG bit of the register PMU_CTRL is set to 0, and when executing WFI or WFE instruction, enter the stop mode immediately; When LPDSCFG bit of the register PMU_CTRL is set to 0, the voltage regulator is working in normal mode; when LPDSCFG bit of the register PMU_CTRL is set to 1, the voltage regulator is working in low-power mode.
Wake-up	If WFI instruction is executed to enter the sleep mode, wake up by any interrupt; if WFE instruction is executed to enter the sleep mode, wake up by an event.
When stopping	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 waking up by interrupt, first enter the interrupt, exit the interrupt, and then execute the program after WFI instruction. If waking up by event, directly execute the program after WFE instruction.

Standby mode

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

Table 28 Standby Mode

Characteristics	Description
Enter	SLEEPDEEP bit of the core register is set to 1, PDDSCFG bit of the register PMU_CTRL is set to 1, WUEFLG bit is set to 0 and when WFI or WFE instruction is executed, it will enter the standby mode immediately.
Wake-up	Wake up by rising edge of WKUP pin, RTC alarm, wake-up, tamper, timestamp event or NRST pin external reset and IWDT reset.
In standby state	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.

7.4.3.2 Reduce the power in run mode

In the run mode, the power in run mode can be reduced by reducing the system clock, enabling or disabling the peripheral clock on the APB/AHB bus.

7.5 Register address mapping

Table 29 PMU Register Address Mapping Table

Register name	Description	Offset address
PMU_CTRL	Power control register	0x00
PMU_CSTS	Power control/status register	0x04

7.6 Register functional description

7.6.1 Power control register (PMU_CTRL)

Offset address: 0x00

Reset value: 0x0000 0000 (cleared when waking up from standby mode)

Field	Name	R/W	Description
0	LPDSCFG	R/W	Low Power Deepsleep Configure Configure the working state of the voltage regulator in stop mode. 0: Enable 1: Low-power mode
1	PDDSCFG	R/W	Power Down Deep Sleep Configure When the CPU enters deep sleep, configure the voltage regulator state in standby or stop mode. 0: The voltage regulator is controlled by LPDSCFG bit when entering the stop mode 1: Enter standby mode
2	WUFLGCLR	RC_W1	Wakeup Flag Clear 0: Invalid 1: Clear the wake-up flag after 2 system clock cycles by writing 1
3	SBFLGCLR	RC_W1	Standby Flag Clear 0: Invalid 1: Write 1 to clear the standby flag
7:4	Reserved		
8	BPWEN	R/W	RTC Domain Write Access Enable RTC area refers to RTC and RTC register; write access is disabled after reset, and is enabled after writing 1. 0: Write is disabled 1: Write is enabled
31:9	Reserved		

7.6.2 Power control/status register (PMU_CSTS)

Offset address: 0x04

Reset value: 0x0000 000X (not cleared when waking up from standby mode)

Compared with the standard APB read, it requires extra APB cycle to read this register

Field	Name	R/W	Description
0	WUEFLG	R	Wakeup Event Flag This bit is set by hardware, indicating whether wake-up event or RTC alarm wake-up event occurs on WKUP pin. 0: Not occur 1: Occurred Note: Enable the WKUP pin, and an event will be detected when the WKUP pin is at high level.

Field	Name	R/W	Description
1	SBFLG	R	Standby Flag This bit is set to 1 by hardware, and can only be cleared by POR/PDR (power-on/power-down reset) or by setting the SBFLGCLR bit of the power control register (PMU_CTRL). 0: Not enter the standby mode 1: Have entered the standby mode
7:2	Reserved		
9:8	WKUPCFGx	R/W	WKUPxPin Configure When WKUPx is used as a normal I/O, the event on WKUPx pin cannot wake up the CPU in standby mode; it can wake up CPU only when it is not used as a normal I/O. 0: Configure normal I/O 1: Can wake up MCU Note: Clear this bit in system reset.
31:10	Reserved		

8 Nested Vector Interrupt Controller (NVIC)

8.1 Full name and abbreviation of terms

Table 30 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 Cortex-M0+ 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 and with low delay. Please see *Cortex-M0+ Technical Reference Manual* for more instructions about NVIC.

8.3 Main characteristics

（1） 32 maskable interrupt channels (excluding 16 Cortex-M0+ interrupt lines)

（2） 4 programmable priority levels (use 2-bit interrupt priority level)

（3） Low-delay exception and interrupt processing

（4） Power management control

（5） Realization of system control register

8.4 Interrupt and exception vector table

Table 31 Interrupt and Exception Vector Table

Name	Vector No.	Priority	Vector address	Description
-	-	-	0x0000_0000	Reserved
RST	-	-3	0x0000_0004	Reset
NMI	-	-2	0x0000_0008	Non-maskable interrupt
Hard fault	-	-1	0x0000_000C	Various hardware faults
SVCall	-	Can set	0x0000_002C	System service called by general SWI instruction
PendSV	-	Can set	0x0000_0038	Pending system service
SysTick	-	Can set	0x0000_003C	System tick timer
WWDT	0	Can set	0x0000_0040	Window watchdog interrupt

Name	Vector No.	Priority	Vector address	Description
PVD	1	Can set	0x0000_0044	PVD interrupt
RTC	2	Can set	0x0000_0048	RTC interrupt
FLASH	3	Can set	0x0000_004C	FLASH interrupt
RCM	4	Can set	0x0000_0050	RCM interrupt
EINT0_1	5	Can set	0x0000_0054	EINT line [1:0] interrupt
EINT2_3	6	Can set	0x0000_0058	EINT line [3:2] interrupt
EINT4_15	7	Can set	0x0000_005C	EINT line [15:4] interrupt
M0CP	8	Can set	0x0000_0060	M0CP interrupt
DMA_CH1	9	Can set	0x0000_0064	DMA channel 1 interrupt
DMA_CH2_3	10	Can set	0x0000_0068	DMA Channel 2 and 3 interrupt
DMA_CH4_7	11	Can set	0x0000_006C	DMA Channel 4/5/6/7 interrupt
ADC_COMP	12	Can set	0x0000_0070	ADC and COMP interrupt
TMR1_BRK_UP_TRG_COM	13	Can set	0x0000_0074	TMR1, BRK, UP, TRG and COM interrupt
TMR1_CC	14	Can set	0x0000_0078	TMR1 capture/compare interrupt
TMR2	15	Can set	0x0000_007C	TMR2 interrupt
TMR3	16	Can set	0x0000_0080	TMR3 interrupt
TMR6	17	Can set	0x0000_0084	TMR6 interrupt
-	-	-	0x0000_0088	保留
TMR4	19	Can set	0x0000_008C	TMR4 interrupt
TMR7	20	Can set	0x0000_0090	TMR7 interrupt
-	-	-	0x0000_0094	Reserved
-	-	-	0x0000_0098	Reserved
I2C1	23	Can set	0x0000_009C	I2C1 interrupt
-	-	-	0x0000_00A0	Reserved
SPI1	25	Can set	0x0000_00A4	SPI1 interrupt
-	-	-	0x0000_00A8	Reserved
USART1	27	Can set	0x0000_00AC	USART1 interrupt
USART2	28	Can set	0x0000_00B0	USART2 interrupt
-	-	-	0x0000_00B4	Reserved
CAN	30	Can set	0x0000_00B8	CAN interrupt
-	-	-	0x0000_00BC	Reserved

9 External interrupt and event controller (EINT)

9.1 Introduction

The interrupts/events contain internal interrupts/events and external interrupts/events. In this manual, external interrupt refers to the interrupt/event caused by I/O pin input signal, which is EINTx in interrupt vector table; other interrupts are 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 can output pulse by events such as GPIO, while the internal events trigger another TMR to work, for example, by an update event of a TMR.

9.2 Main characteristics

（1） Support 28 event/interrupt requests

（2） Can be configured independently as the line of external/internal event request

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

（4） The internal line is automatically disabled when the system is not in the stop mode

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

（6） Each external interrupt line has dedicated status bit

（7） Simulate all external event interrupts

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 32 Classification and Differences of "External Interrupt and Event"

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 waken up 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

9.3.3 External interrupt and event line mapping

Table 33 External Interrupt and Event Line Mapping

External Interrupt and Event Channel Name	External Interrupt and Event Line No.
PA0/PB0/PC0/PF0	EINT 0
PA1/PB1/PC1/PF1	EINT 1
…	…
PA15/PB15/PC15	EINT 15
PVD output	EINT 16
RTC alarm event	EINT 17
Reserved	EINT 18
RTC tampering and timestamp event	EINT 19
Reserved	EINT 20
COMP1 output	EINT 21
COMP2 output	EINT 22
Internal I2C1 wake-up event	EINT 23
Reserved	EINT 24
Internal USART1 wake-up event	EINT 25
Reserved	EINT 26
Reserved	EINT 27

9.4 Register address mapping

Table 34 External Interrupt/Event Controller Register Address Mapping

Register name	Description	Offset address
EINT_IMASK	Interrupt mask register	0x00
EINT_EMASK	Event mask register	0x04
EINT_RTEN	Rising edge trigger selection register	0x08
EINT_FTEN	Falling edge trigger selection register	0x0C
EINT_SWINTE	Software interrupt event register	0x10

Register name	Description	Offset address
EINT_IPEND	Pending register	0x14

9.5 Register functional description

9.5.1 Interrupt mask register (EINT_IMASK)

Offset address: 0x00

Reset value: 0x0F94 0000

Field	Name	R/W	Description
27:0	IMASKx	R/W	Interrupt Request Mask on Line x 0: Mask 1: Open
31:28	Reserved		

9.5.2 Event mask register (EINT_EMASK)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
27:0	EMASKx	R/W	Event Request Mask on Line x 0: Mask 1: Open
31:28	Reserved		

9.5.3 Enable rising edge trigger selection register (EINT_RTEN)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
17:0	RTENx	R/W	Rising Trigger Event Enable and Interrupt of Line x 0: Disable 1: Enable
18	Reserved		
19	RTEN19	R/W	Rising Trigger Event Enable and Interrupt of Line 19 0: Disable 1: Enable
20	Reserved		
22:21	RTENx	R/W	Rising Trigger Event Enable and Interrupt of Line x 0: Disable 1: Enable
31:23	Reserved		

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 falling edge trigger selection register (EINT_FTEN)

Offset address: 0x0C

Reset value: 0x0000 0000

Field	Name	R/W	Description
17:0	FTENx	R/W	Falling Trigger Event Enable and Interrupt of Line x 0: Disable 1: Enable
18	Reserved		
19	FTEN19	R/W	Falling Trigger Event Enable and Interrupt of Line 19 0: Disable 1: Enable
20	Reserved		
22:21	FTENx	R/W	Falling Trigger Event Enable and Interrupt of Line x 0: Disable 1: Enable
31:23	Reserved		

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
17:0	SWINTEx	R/W	Software Interrupt Event on Line x 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)
18	Reserved		
19	SWINTE19	R/W	Software Interrupt Event on Line 19 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)

Field	Name	R/W	Description
20	Reserved		
22:21	SWINTEx	R/W	Software Interrupt Event on Line x 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)
31:23	Reserved		

9.5.6 Interrupt pending register (EINT_IPEND)

Offset address: 0x14

Reset value: 0xXXXX XXXX

Field	Name	R/W	Description
17:0	IPENDx	RC_W1	Interrupt Pending Occur of Line x Flag Whether the selected trigger request occurs 0: No 1: Occurred When a request is triggered by the corresponding edge of EINT_RTEN/EINT_FTEN on the external interrupt line, set 1 by hardware; clear 0 by changing the polarity of the edge detection or clear 0 by writing 1 to this bit.
18	Reserved		
19	IPEND19	RC_W1	Interrupt Pending Occur of Line 19 Flag Whether the selected trigger request occurs 0: No 1: Occurred When a request is triggered by the corresponding edge of EINT_RTEN/EINT_FTEN on the external interrupt line, set 1 by hardware; clear 0 by changing the polarity of the edge detection or clear 0 by writing 1 to this bit.
20	Reserved		
22:21	IPENDx	RC_W1	Interrupt Pending Occur of Line x Flag Whether the selected trigger request occurs 0: No 1: Occurred When a request is triggered by the corresponding edge of EINT_RTEN/EINT_FTEN on the external interrupt line, set 1 by hardware; clear 0 by changing the polarity of the edge detection or clear 0 by writing 1 to this bit.
31:23	Reserved		

10 Direct memory access (DMA)

10.1 Full name and abbreviation of terms

Table 35 Full Name and Abbreviation Description of Terms

Full name in English	English abbreviation
Global	G
Transfer	T
Half	H
Complete	C
Error	E
Channel	CH
Circular	CIR
Peripheral	PER
Increment	I
Memory	M
Priority	PRI
Number	N
Address	ADDR

10.2 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 a DMA controller with seven channels. Each channel can manage multiple DMA requests, but each channel can only respond to one DMA request at the same time. Each channel can set priority, and the arbiter can coordinate the priority of corresponding DMA requests of each DMA channel according to the priority of the channels.

10.3 Main characteristics

（1） DMA has seven channels

（2） There are three data transmission modes: peripheral to memory, memory to peripheral, and memory to memory

（3） Each channel has a special hardware DMA request for connection

（4） Support software priority and hardware priority when multiple requests occur at the same time

（5） Each channel has three event flags and independent interrupts

（6） Support circular transmission mode

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

10.4 Functional description

10.4.1 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 seven channels. Each channel is connected with different peripherals, and each channel has three event flags (DMA half transmission, DMA transmission completion and DMA transmission error). The logic of the three event flags may become a separate interrupt request, and they all support software trigger.

When multiple peripherals request the same channel, it is required to configure the corresponding register to enable or disable the request of each peripheral, so as to ensure that one channel can only enable one peripheral request.

Table 36 DMA Request Mapping Table

Peripheral Clock	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7
TMR1	—	TMR1_CH1	TMR1_CH2	TMR1_CH4 TMR1_TRIG TMR1_COM	TMR1_CH3(1) TMR1_UP	TMR1_CH1(2) TMR1_CH2(2) TMR1_CH3(2)	—
TMR2	TMR2_CH3	TMR2_UP	TMR2_CH2(1)	TMR2_CH4(1)	TMR2_CH1	—	TMR2_CH2(2) TMR2_CH4(2)
TMR3	—	TMR3_CH3	TMR3_CH4 TMR3_UP	TMR3_CH1(1) TMR3_TRIG(1)	—	TMR3_CH1(2) TMR3_TRIG(2)	—
TMR4	—	—	—	—	TMR4_CH3	TMR4_CH4 TMR4_UP	TMR4_CH1 TMR4_TRIG
TMR7	—	—	—	—	TMR7_UP	—	—
TMR6	—	—	TMR6_UP	—	—	—	—
ADC	ADC(1)	ADC(2)	—	—	—	—	—
SPI	—	SPI1_RX(1)	SPI1_TX(1)	SPI1_RX(2)	SPI1_TX(2)	—	—
USART	—	USART1_TX(1)	USART1_RX(1)	USART1_TX(2) USART2_TX	USART1_RX(2) USART2_RX	—	—
I2C	—	I2C1_TX(1)	I2C1_RX(1)	—	—	I2C1_TX(2)	I2C1_RX(2)

Notes:

（1）This DMA request is mapped to the DAM channel only when the corresponding bit of SYSCFG_CFG1 register is cleared to 0.

（2）This DMA request is mapped to the DAM channel only when the corresponding remapping bit of SYSCFG_CFG1 register is set.

10.4.2 DMA channel

10.4.2.1 Transmission data are programmable

The data transmitted by DMA are programmable, up to 65535, and the transmission data bit width of peripherals and memory can be set by configuring PERSIZE bit and MEMSIZE bit of DMA_CHCFGx register.

10.4.2.2 Transmission width and alignment method are programmable

DMA transmission operation of programmable data transmission width:

Figure 9 Transmission Width with Source of 8bits and Target of 8bits

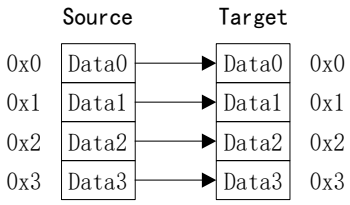


Figure 10 Transmission Width with Source of 8bits and Target of 16bits

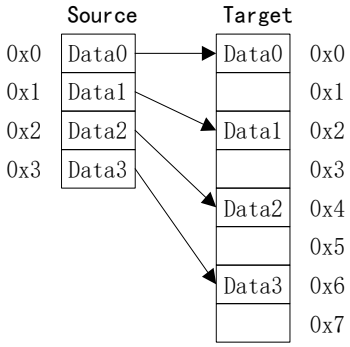


Figure 11 Transmission Width with Source of 8bits and Target of 32bits

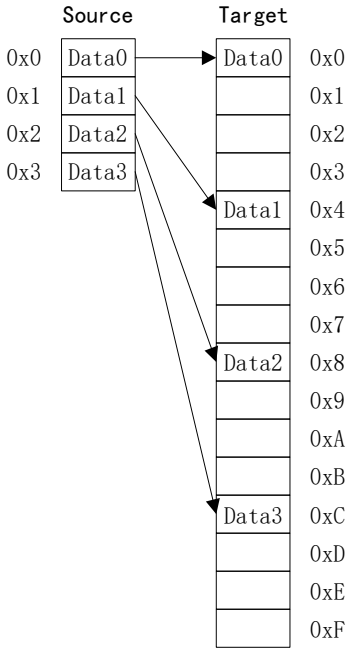


Figure 12 Transmission Width with Source of 32bits and Target of 8bits

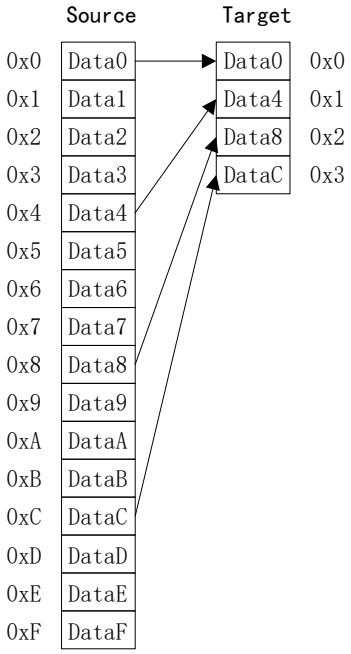


Figure 13 Transmission Width with Source of 16bits and Target of 16bits

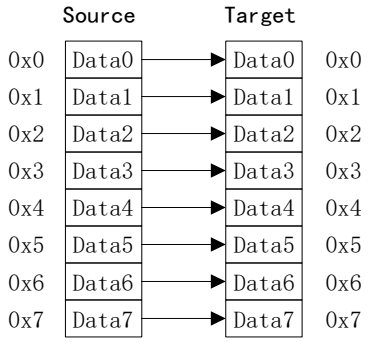


Figure 14 Transmission Width with Source of 16bits and Target of 32bits

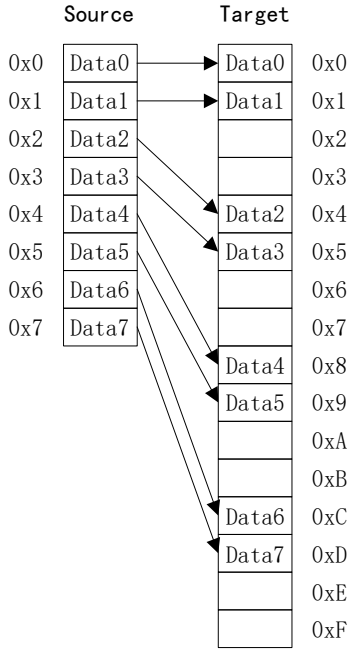
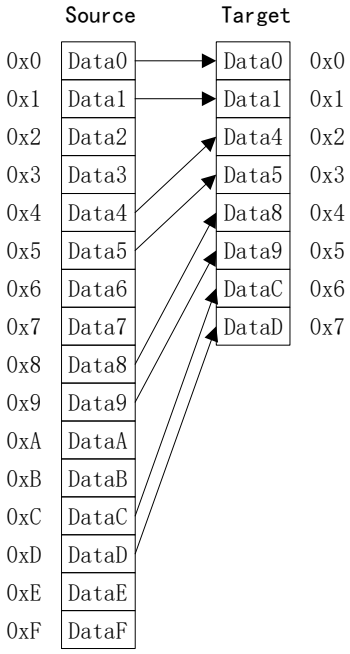


Figure 15 Transmission Width with Source of 32bits and Target of 16bits



10.4.2.3 Address setting

The transmission address supports two modes: fixed mode and pointer increment mode.

Transmission address pointer increment mode

The automatic pointer increment of peripheral and memory is completed by PERIMODE bit and MIMODE bit of configuration register DMA_CHCFGx. The next address to be transmitted is the one by adding the increment to the previous address. The increment depends on the selected data width.

10.4.2.4 Transmission mode

There are two channel configuration modes: non-circular mode and circular mode.

Non-circular mode

When the data transmission ends, the DMA operation will not be performed any more, and a new DMA transmission will be started. When the DMA channel is not working, the register DMA_CHNDATAx will rewrite the transmission value.

Circular mode

After the data transmission ends, the content of the register DMA_CHNDATAx will be automatically reloaded to the previously configured value, and the peripheral address register DMA_CHPADDRx and the memory address register DMA_CHMADDRx will also be reloaded as the initial base address.

The configuration method is as follows:

- Set the CIRMODE bit of the configuration register DMA_CHCFGx to 1 to turn on the circular mode;
- This mode is used to process continuous peripheral requests. When the number of data for transmission becomes 0, it will automatically return to the initial value and continue the DMA operation and when the CIRMODE bit is cleared to 0, it will exit the circular mode.

10.4.2.5 DMA request priority setting

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 four priorities; the second stage is hardware stage, and under the condition of the same software priority, the lower the channel number is, the higher the priority is.

10.4.2.6 Transmission direction

Support three directions: from memory to memory, from memory to peripheral, and from peripheral to memory.

If the write operation (target address) is performed on the memory, the memory includes internal SRAM, and external RAM supported by EMMC (such as external SRAM); if the read operation (source address) is performed on the memory, the address includes internal Flash and internal SRAM.

Examples of "from memory to memory" configuration are as follows:

- M2MMODE bit of configuration register DMA_CHCFGx can enable the memory-to-memory mode;
- The DMA operation in this mode is performed under the condition of no peripheral request. Set CHEN bit of configuration register DMA_CHCFGx to 1, and after the channel is enabled, the data transmission will start and when the transmission quantity register DMA_CHNDATAx becomes 0, the transmission ends.

10.4.3 Interrupt

Each DMA channel has three types of interrupt events, which are half transmission (HT), transmission completion (TC) and transmission error (TE).

（1） The interrupt event flag bit for half transmission is HTFLG, and the interrupt enable control bit is HTINTEN

（2） The interrupt event flag bit for transmission completion is TCFLG, and the interrupt enable control bit is TCINTEN

（3） The interrupt event flag bit for transmission error is TERRFLG, and the interrupt enable control bit is TERRINTEN

10.5 Register address mapping

Table 37 Register Address Mapping

Register Name	Description	Offset address
DMA_INTSTS	DMA interrupt status register	0x00
DMA_INTFCLR	DMA interrupt flag reset register	0x04
DMA_CHCFGx	DMA Channel x configuration register	0x08+20 x
DMA_CHNDATAx	DMA Channel x transmission quantity register	0x0C+20 x
DMA_CHPADDRx	DMA Channel x peripheral address register	0x10+20 x
DMA_CHMADDRx	DMA Channel x memory address register	0x14+20 x

10.6 Register functional description

10.6.1 DMA interrupt status register (DMA_INTSTS)

Offset address: 0x00

Reset value: 0x0000 0000

Field	Name	R/W	Description
24,20,16, 12,8,4,0	GINTFLGx	R	Channel x Global Interrupt Occur Flag (x=1…7) Indicate whether TC, HT or TE interrupt is generated on the channel; these bits are set to 1 by hardware; write 1 and clear 0 on the corresponding bit of DMA_INTFCLR. 0: Not generate 1: Generate
25,21,17, 13,9,5,1	TCFLGx	R	Channel x All Transfer Complete Flag (x=1..7) Indicate whether the transmission completion interrupt (TC) is generated on the channel; these bits are set to 1 by hardware; write 1 and clear 0 on the corresponding bit of DMA_INTFCLR. 0: Not completed 1: Completed
26,22,18, 14,10,6,2	HTFLGx	R	Channel x Half Transfer Complete Flag (x=1..7) Indicate whether the half transmission interrupt (HT) is generated on the channel; these bits are set to 1 by hardware; write 1 and clear 0 on the corresponding bit of DMA_INTFCLR. 0: Not generate 1: Generate
27,23,19, 15,11,7,3	TERRFLGx	R	Channel x Transfer Error Occur Flag (x=1..7) Indicate whether the transmission error interrupt (TE) is generated on the channel; these bits are set to 1 by hardware; write 1 and clear 0 on the corresponding bit of DMA_INTFCLR. 0: Not generate 1: Generate

Field	Name	R/W	Description
31:28	Reserved		

10.6.2 DMA interrupt flag clear register (DMA_INTFCLR)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
24,20,16,12, 8,4,0	GINTCLRx	R/W	Channel x Global Interrupt Occur Flag Clear (x=1..7) Clear the corresponding GINTFLG, TCFLG, HTFLG and TERRFLG flags in the interrupt status register. 0: Invalid 1: Clear the GINTFLG flag
25,21, 17,13, 9,5,1	TCCLRx	R/W	Channel x Transfer Complete Clear (x=1..7) Clear the corresponding TCFLG flag in interrupt status register. 0: Invalid 1: Clear the TCFLG flag
26,22 18,14, 10,6,2	HTCLRx	R/W	Channel x Half Transfer Complete Clear (x=1..7) Clear the corresponding HTFLG flag in interrupt status register. 0: Invalid 1: Clear the HTFLG flag
27,23, 19,15, 11,7,3	TERRCLRx	R/W	Channel x Transfer Error Occur Clear (x=1..7) Clear the corresponding TERRFLG flag in interrupt status register. 0: Invalid 1: Clear the TERRFLG flag
31:28	Reserved		

10.6.3 DMA Channel x configuration register (DMA_CHCFGx) (x=1…7)

Offset address: 0x08+20 x (channel number–1)

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	CHEN	R/W	DMA Channel Enable 0: Disable 1: Enable
1	TCINTEN	R/W	All Transfer Complete Interrupt Enable 0: Disable 1: Enable
2	HTINTEN	R/W	Half Transfer Complete Interrupt Enable 0: Disable 1: Enable
3	TERRINTEN	R/W	Transfer Error Occur Interrupt Enable 0: Disable 1: Enable

Field	Name	R/W	Description
4	DIRCFG	R/W	Data Transfer Direction Configure 0: Read from peripheral to memory 1: Read from memory to peripheral
5	CIRMODE	R/W	Circular Mode Enable 0: Disable 1: Enable
6	PERIMODE	R/W	Peripheral Address Increment Mode Enable 0: Disable 1: Enable
7	MIMODE	R/W	Memory Address Increment Mode Enable 0: Disable 1: Enable
9:8	PERSIZE	R/W	Peripheral Data Size Configure 00: 8 bits 01: 16 bits 10: 32 bits 11: Reserved
11:10	MEMSIZE	R/W	Memory Data Size Configure 00: 8 bits 01: 16 bits 10: 32 bits 11: Reserved
13:12	CHPL	R/W	Channel Priority Level Configure 00: Low 01: Medium 10: High 11: Highest
14	M2MMODE	R/W	Memory to Memory Mode Enable 0: Disable 1: Enable
31:15	Reserved		

10.6.4 Transmission quantity register of DMA Channel x (DMA_CHNDATAx) (x=1…7)

Offset address: 0x0C+20 x (channel number–1)

Reset value: 0x0000 0000

Field	Name	R/W	Description
15:0	NDATAT	R/W	Number of Data to Transfer Setup This register indicates the number of bytes to be transmitted. The number of data transmission ranges from 0 to 65535. This register can only be written when the channel is not working; once the channel is enabled, the register will become read-only, indicating the number of remaining bytes to be transmitted. The register will decrease every time DMA is transmitted; when the data transmission is completed, the register will change to 0, or when the channel is configured to auto reload mode, it will be automatically reloaded to the previously configured value; if the register is 0, data transmission will not occur regardless of whether the channel is enabled or not.
31:16	Reserved		

10.6.5 DMA Channel x peripheral address register (DMA_CHPADDRx) (x=1…7)

Offset address: 0x10+20 x (channel number–1)

Reset value: 0x0000 0000

This register cannot be written when the channel is enabled (CHEN=1 for DMA_CHCFGx).

Field	Name	R/W	Description
31:0	PERADDR	R/W	Peripheral Basic Address Setup When PERSIZE= '01' (16 bits) and PERADDR[0] bit is not used, it will be aligned with 16-bit address automatically during transmission. When PERSIZE= '10' (32 bits) and PERADDR[1:0] bit is not used, it will be aligned with 32-bit address automatically during transmission.

10.6.6 DMA Channel x memory address register (DMA_CHMADDRx) (x=1…7)

Offset address: 0x14+20 x (channel number–1)

Reset value: 0x0000 0000

This register cannot be written when the channel is enabled (CHEN=1 for DMA_CHCFGx).

Field	Name	R/W	Description
31:0	MEMADDR	R/W	Memory Basic Address Setup When MEMSIZE= '01' (16 bits) and MEMADDR[0] bit is not used, it will be aligned with 16-bit address automatically during transmission. When MEMSIZE= '10' (32 bits) and MEMADDR[1:0] bit is not used, it will be aligned with 32-bit address automatically during transmission.

11 Debug MCU (DBGMCU)

11.1 Full name and abbreviation of terms

Table 38 Full Name and Abbreviation of Terms

Full name in English	English abbreviation
Frame Clock	FCLK
Data Watchpoint Trigger	DWT
Break Point Unit	BPU

11.2 Introduction

APM32F0xx MCU series uses Arm® Cortex®-M0+ core, and Arm® Cortex®-M0+ core includes hardware debug module and supports complex debug 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.

Supported debugging interface: Serial interface

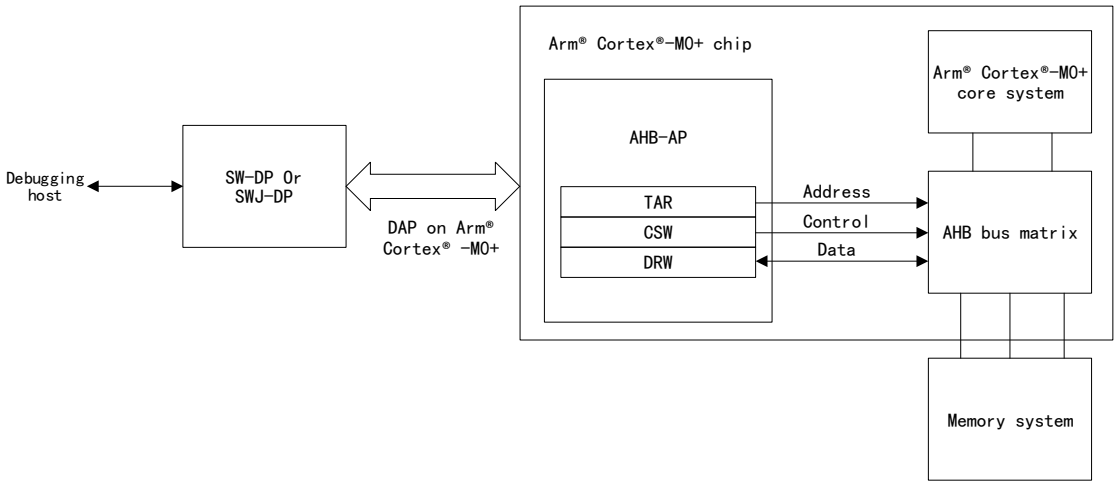
Note: The hardware debug interface included in Arm® Cortex®-M0 core is a subset of Arm CoreSight development tool set. Please refer to Cortex®-M0+ (Version r1p1) technical reference manual (TRM) and CoreSight development tool set (Version r1p0) TRM for more information about debug function of Arm® Cortex®-M0+ core.

11.3 Main characteristics

（1） Flexible debug pin assignment

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

Figure 16 APM32F0xx Level and Arm® Cortex®-M0+ Level Debugging Block Diagram



11.4 Functional description

（1） Realize 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 production of bus-off programmer

11.5 Register address mapping

Table 39 DBGMCU Register Address Mapping

Register name	Description	Offset address
DBGMCU_IDCODE	Debug MCU device ID register	0x00
DBGMCU_CFG	Debug MCU configuration register	0x04
DBGMCU_APB1F	Debug MCU APB1 freeze register	0x08
DBGMCU_APB2F	Debug MCU APB2 freeze register	0x0C

11.6 Register functional description

11.6.1 Debug MCU device ID register (DBGMCU_IDCODE)

Address: 0x00

Only support 32-bit access

Reset value: 0x0001 4001

Field	Name	R/W	Description
15:0	EQR	R	Equipment Recognition This field indicates device ID: 0x001
31:16	WVR	R	Wafer Version Recognition This field indicates the device version: 0x0013

11.6.2 Device ID register (DBGMCU_CFG)

This register allows configuring MCU during debugging and supports low-power mode.

It is reset asynchronously by POR (not reset by system), and can be written by debugger through system reset.

If the debugging master does not support these characteristics, the user software can write to these registers.

Only support 32-bit access

Address: 0x04

Reset value: 0x0000 (unaffected by system reset)

Field	Name	R/W	Description
0	Reserved		
1	STOP_CLK_STS	R/W	Debug Stop Mode Configure 0: In the stop mode when both FCLK and HCLK are disabled, all clocks will be disabled by clock controller. When exiting the stop mode, the clock configuration is the same as that after reset (the clock is provided by the 8MHz internal RC oscillator HSICLK), so the software needs to reconfigure the clock controller to enable PLL, crystal oscillator, etc. 1: In the stop mode when both FCLK and HCLK are enabled, both FCLK and HCLK are provided by internal RC oscillator. The internal RC oscillator remains or is active in the stop mode. When it exits the stop mode, the software must reconfigure the clock controller to enable PLL, crystal oscillator, etc.
2	STANDBY_CLK_STS	R/W	Debug Standby Mode 0: When both FCLK and HCLK are disabled, the digital part is not powered on. From the software level, it indicates that when the MCU just exits the standby mode, others exit the debug mode, which is the same as reset 1: When both FCLK and HCLK are enabled, the digital part is powered on, and the internal RC oscillator provides FCLK and HCLK clocks. Besides, the MCU exits the standby mode by system reset, which is the same as reset.
31:3	Reserved		

11.6.3 Debug MCU APB1 freeze register (DBGMCU_APB1F)

This register is used to configure MCU during debugging.

Involve some APB peripherals:

- Freeze the timer counter
- Freeze I2C SMBus timeout
- Freeze supporting system window regulators and independent watchdog counter

This register is reset asynchronously by POR (not reset by system) and can be written by the debugger through system reset.
Only support 32-bit access
Address: 0x08
Reset value: 0x0000 (unaffected by system reset)

Field	Name	R/W	Description
0	TMR2_STS	R/W	Configure TMR2 Work Status When Core is in Halted Whether TMR2 counter continues to work when the core stops work 0; Continue to work 1: Stop working
1	TMR3_STS	R/W	Configure Timer3 Work Status When Core is in Halted Whether TMR3 counter continues to work when the core stops work 0; Continue to work 1: Stop working
3:2	Reserved		
4	TMR6_STS	R/W	Configure Timer6 Work Status When Core is in Halted Whether TMR6 counter continues to work when the core stops work 0; Continue to work 1: Stop working
7:5	Reserved		
8	TMR4_STS	R/W	ConfigureTimer4 Work Status When Core is in Halted Whether TMR4 counter continues to work when the core is halted 0; Continue to work 1: Stop working
9	Reserved		
10	RTC_STS	R/W	Configure RTC Work Status When Core Is in Halted Whether RTC counter continues to work when the core stops work 0; Continue to work 1: Stop working
11	WWDT_STS	R/W	Configure Window Watchdog Work Status When Core Is in Halted Whether WWDT continues to work when the core stops 0; Continue to work 1: Stop working
12	IWDT_STS	R/W	Configure Independent Watchdog Work Status When Core Is in Halted Whether IWDT continues to work when the core stops 0; Continue to work 1: Stop working
20:13	Reserved		

Field	Name	R/W	Description
21	I2C1_SMBUS_TIMEOUT_STS	R/W	Configure I2C1_SMBUS_TIMEOUT Work Status When Core Is in Halted 0: Work normally 1: Freeze the timeout mode of SMBUS
24:22	Reserved		
25	CAN_STS	R/W	Configure CAN Work Status When Core is in Halted Whether CAN continues to work when the core stops 0; Continue to work 1: Stop working
31:22	Reserved		

11.6.4 Debug MCU APB2 freeze register (DBGMCU_APB2F)

This register is used to configure MCU during debugging.

Involve some APB peripherals:

- Freeze the timer counter

This register is reset asynchronously by POR (not reset by system) and can be written by the debugger through system reset.

Only support 32-bit access

Address: 0x0C

Reset value: 0x0000 (unaffected by system reset)

Field	Name	R/W	Description
10:0	Reserved		
11	TMR1_STS	R/W	Configure Timer1 Work Status When Core is in Halted Whether TMR1 counter continues to work when the core stops 0; Continue to work 1: Stop working
15:12	Reserved		
16	TMR7_STS	R/W	Configure Timer7 Work Status When Core is in Halted Whether TMR7 counter continues to work when the core stops 0; Continue to work 1: Stop working
31:17	Reserved		

12 General-purpose/Alternate function input/output pin (GPIO/AFIO)

12.1 Full name and abbreviation of terms

Table 40 Full Name and Abbreviation Description of Terms

Full name in English	English abbreviation
P-channel Metal Oxide Semiconductor	P-MOS
N-channel Metal Oxide Semiconductor	N-MOS

12.2 Main characteristics

（1） Input mode
- Floating input
- Pull-up input
- Pull-down input

（2） Output mode
- Push-pull output
- Open-drain output
- Configurable maximum output rate

（3） Multiplexing mode
- Push-pull multiplexing function
- Open-drain multiplexing function

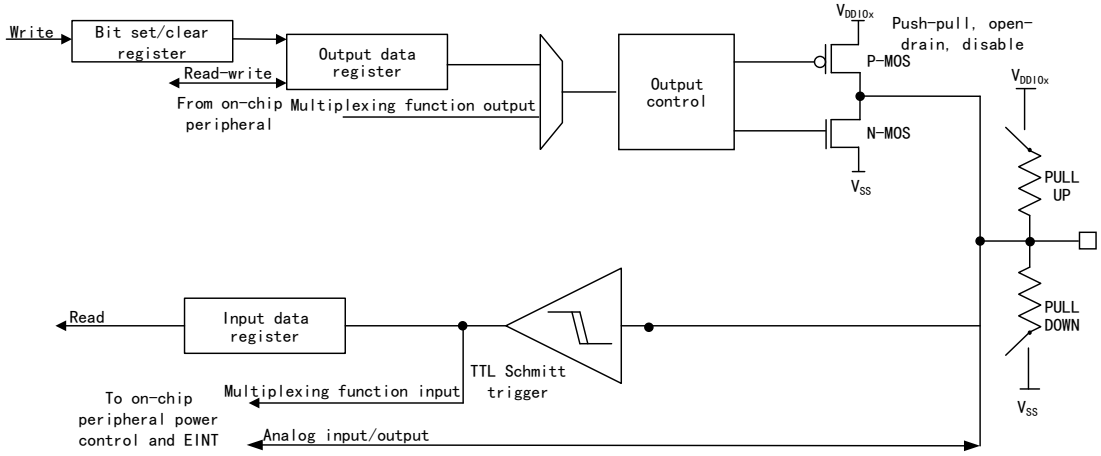
（4） Analog mode

（5） GPIO can be used as external interrupt/wakeup line

（6） Support locking I/O configuration function

12.3 Structure block diagram

Figure 17 GPIO Structure Block Diagram



12.4 Functional description

Each pin of GPIO can be configured as pull-up, pull-down, floating and analog input, or push-pull/open-drain output and input mode and multiplexing function by software. All GPIO interfaces have external interrupt capability.

12.4.1 IO status during reset and just after reset

During and just after GPIO reset, if the multiplexing function is not enabled, the I/O port will be configured as floating input mode.

After reset, the debug pin is in AF pull-up or pull-down state:

- PA14: SWCLK in pull-down mode
- PA13: SWDIO in pull-up mode

12.4.2 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

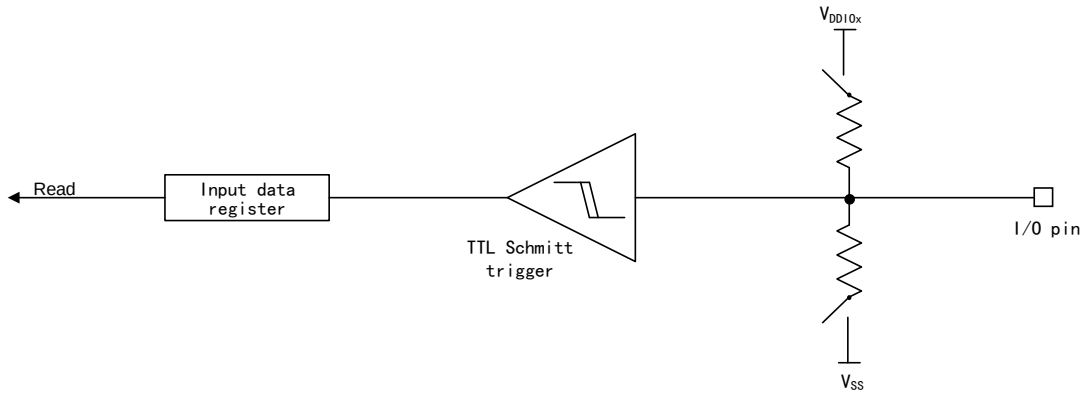
- Schmitt trigger is enabled
- Disable output buffer
- By configuring the pull-up/pull-down register GPIOx_PUPD, select whether to use pull-up/pull-down resistor

- The input data register GPIOx_IDATA captures the data on I/O pin in each AHB clock cycle.
- Read I/O state by the input data register GPIOx_IDATA

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.

Figure 18 Input Mode I/O Structure



12.4.3 Output mode

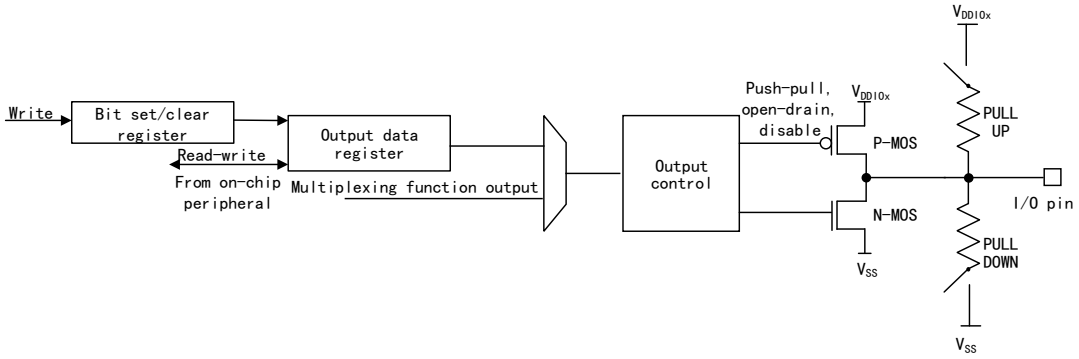
In the output mode, it can be set as push-pull output and open-drain output.

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:

- Schmitt trigger is enabled
- Activate output buffer
- By configuring the pull-up/pull-down register GPIOx_PUPD, select whether to use pull-up/pull-down resistor
- Push-pull mode:
  - Double MOS transistor works by turns and the output data register can control the high and low level of I/O output
  - Read the finally written value through the output data register GPIOx_ODATA
- Open-drain mode:
  - Only N-MOS works, and the output data register can control I/O output high-resistance state or low level
  - The input data register GPIOx_IDATA captures the data on I/O pin in each AHB clock cycle
  - Read the actual I/O state through the input data register GPIOx_IDATA

Figure 19 Output Mode I/O Structure



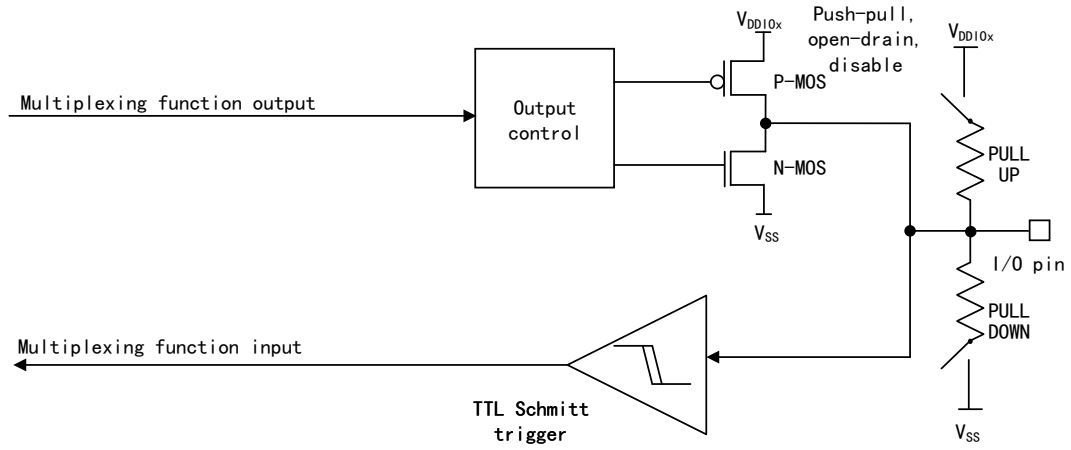
12.4.4 Multiplexing mode

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
- By configuring the pull-up/pull-down register GPIOx_PUPD, select whether to use pull-up/pull-down resistor
- The data on the I/O pin is sampled in each AHB clock cycle and stored in the port input status register
- Read the actual I/O state through the input data register GPIOx_IDATA

Figure 20 Multiplexing Mode I/O Structure



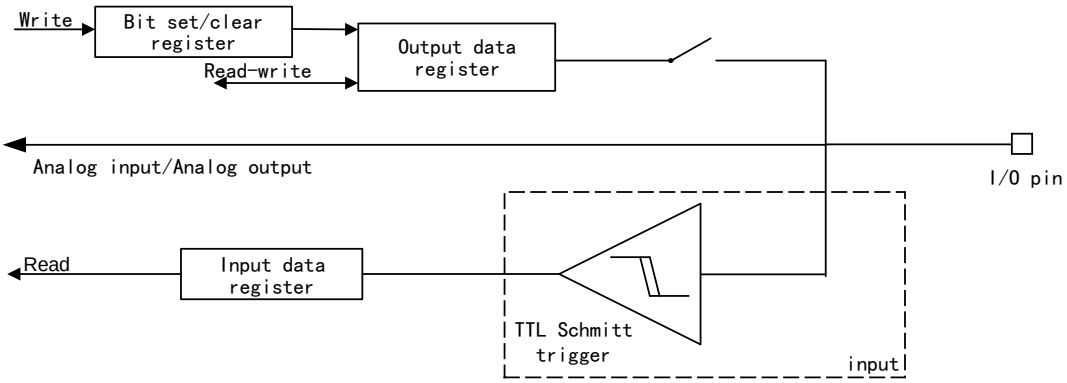
12.4.5 Analog mode

In analog function 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
- Read the value of the input status register to be 0

Figure 21 Analog Function I/O Structure



12.4.6 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.

12.4.7 I/O data bit processing

GPIO port set/reset register (GPIOx_BSC) allows set/reset operation for each bit of the output data register (GPIOx_ODATA). The valid data width of the set/reset register is double the valid data width of GPIOx_ODATA.

Writing 0 to any bit in GPIOx_BSC will not affect the value of the GPIOx_ODATA register. If BS and BC bits of GPIOx_BSC are set to 1 at the same time, BS bit has the priority. GPIOx_ BSC register can change the corresponding bit of the GPIOx_ODATA register, and GPIOx_ODATA bit can be accessed directly from GPIOx_BSC register.

When the access mechanism is set or reset by GPIOx_ODATA through GOIOx_BSC register, it is not necessary to disable the interrupt by software to access GPIOx_ODATA.

12.4.8 Multiplexing function and remapping

Multiplexer

The multiplexer is used to connect the I/O port line of the device to the embedded peripheral module, and it can only be one-to-one at the same time.

Each I/O pin is equipped with a multiplexer. The multiplexer has up to 16 multiplexing function inputs, but in fact it uses up to 8 (AF0-AF7), which are configured by GPIOx_ALFL and GPIOx_ALFH registers. When I/O pin is reset, all pin ports are connected to AF0.

Remapping

Each peripheral has multiple multiplexing functions, but only one multiplexing function input can be selected for a pin, so the multiplexing function of the peripheral can be mapped to other I/O pins, that is, the multiplexing function signal can be reassigned to a pin address.

The multiplexing function and remapping address table of pins are shown in the datasheet.

I/O multiplexing configuration

When I/O port is connected to the peripheral multiplexing function, the following debugging needs to be done:

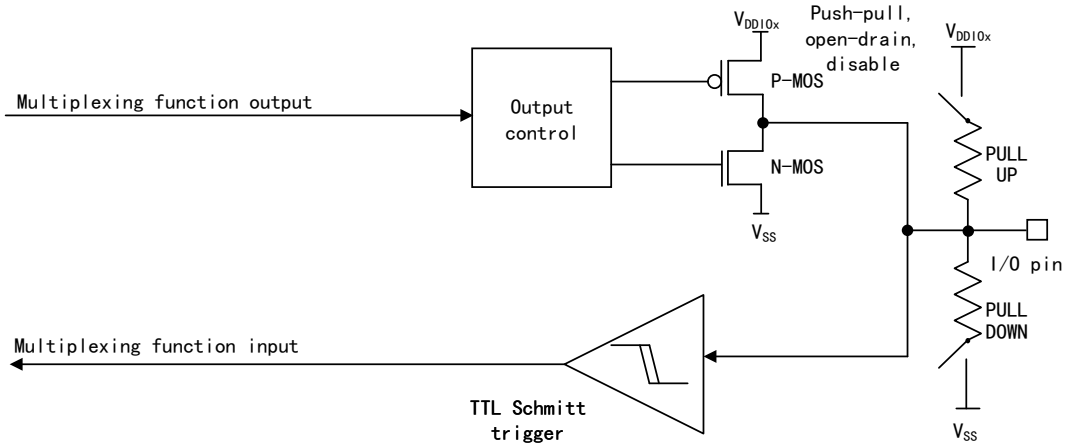
- After reset, the pin is configured with multiplexing function
- I/O port is configured as input, output or analog input
- The I/O port is connected to the defined AFx
- Configure pin pull-up/pull-down and output speed
- Configure I/O as multiplexing function in GPIOx_MODE

When the I/O port is configured with multiplexing function, its input and output mode is as follows:

- Enable the output buffer
- Output buffer is driven by peripheral
- Activate Schmitt trigger input
- By configuring the pull-up/pull-down register GPIOx_PUPD, select whether to use pull-up/pull-down resistor
- The data on the I/O pin is sampled in each AHB clock cycle and stored in the port input status register
- Read the actual I/O state through the input data register GPIOx_IDATA

The multiplexing mode I/O structure is shown in the figure below:

Figure 22 Multiplexing Mode I/O Structure



12.4.9 GPIO locking function

The locking mechanism of GPIO can protect the configuration of I/O port.

Write sequence (specific) to GPIOx_LOCK register so as to freeze the control register of Port A and Port B. If you want to write GPIOx_LOCK register, a specific write/read sequence should be transmitted.

I/O configuration can be locked by configuring the lock register (GPIOx_LOCK). When a port bit executes the locking program, the configuration of port bit cannot be modified before the next reset.

12.4.10 HSECLK or LSECLK pin is used as GPIO

By configuring HSEEN/LSEEN in RCM_CTRL1 and RCM_RTCCTRL registers, set whether to enable HSECLK/LSECLK RC oscillator.

When HSECLK/LSECLK RC oscillator is enabled, the oscillator controls the related pins, and the related pins are unrelated to GPIO configuration; when HSECLK/LSECLK RC oscillator is disabled, the related oscillators can be used as general GPIO interface.

12.4.11 GPIO is used in RTC power supply field

When the core power supply domain is powered off, it will lose PC13/PC14/PC15 GPIO function, and at this time, if the GPIO is not configured by RTC, PC13/14/PC15 pin will be set as analog input mode.

For detailed information about RTC controlled I/O pins, please refer to I/O pin controlled by RTC.

12.5 Register address mapping

Table 41 GPIO Register Address Mapping

Register name	Description	Offset address
GPIOx_MODE	Port mode register	0x00
GPIOx_OMODE	Port output mode register	0x04
GPIOx_OSSEL	Port output speed register	0x08
GPIOx_PUPD	Port pull-up/pull-down register	0x0C
GPIOx_IDATA	Port bit input data register	0x10
GPIOx_ODATA	Port bit output clear register	0x14
GPIOx_BSC	Port set/reset register	0x18
GPIOx_LOCK	Port lock register	0x1C
GPIOx_ALFL	Port multiplexing function low-8-bit register	0x20
GPIOx_ALFH	Port multiplexing function high-8-bit register	0x24
GPIOx_BR	Port reset register	0x28

12.6 Register functional description

12.6.1 Port mode register (GPIOx_MODE) (x=A…C, F)

Offset address: 0x00

Reset value: 0x2800 000 for Port A

0x0000 000 for other ports

Field	Name	R/W	Description
31:0	MODEy[1:0]	R/W	PortxPin y Mode Configure (y=0…15) 00: Input mode (state after reset) 01: General output mode 10: Multiplexing function mode 11: Analog mode

12.6.2 Port output mode register (GPIOx_OMODE) (x=A…C, F)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
15:0	OMODEy	R/W	PortxPin y Output Mode Configure (y=0…15) 0: Push-pull output (reset state) 1: Open-drain output
31:16	Reserved		

12.6.3 Port output speed register (GPIOx_OSSEL) (x=A…C, F)

Offset address: 0x08

Reset value: 0x0C00 0000 for Port A

0x0000 0000 for other ports

Field	Name	R/W	Description
31:0	OSSELy[1:0]	R/W	PortxPin y Output Speed Select (y=0…15) x0: Low speed 01: Medium speed 11: High speed The speed of I/O port is written by software

12.6.4 GPIO port pull-up/pull-down register (GPIOx_PUPD) (x=A…C, F)

Offset address: 0x0C

Reset value: 0x2400 0000 for Port A

0x0000 0000 for other ports

Field	Name	R/W	Description
31:0	PUPDy[1:0]	R/W	PortxPin y Pull-up/Pull-down Configure (y=0…15) These bits are written by software to configure pull-up/pull-down of the port bit 00: Pull-up/Pull-down is disabled 01: Pull up 10: Pull down 11: Reset

12.6.5 GPIO port input data register (GPIOx_IDATA) (x=A..C, F)

Offset address: 0x10

Reset value: 0x0000 XXXX

Field	Name	R/W	Description
15:0	IDATAy	R	PortxPin y Input Data (y=0…15) These bits can only be read to store the input values of the corresponding I/O ports
31:16	Reserved		

12.6.6 GPIO port output data register (GPIOx_ODATA) (x=A..C, F)

Offset address: 0x14

Reset value: 0x0000 0000

Field	Name	R/W	Description
15:0	ODATAy	R/W	PortxPin y Output Data (y=0…15) Read and write operation can be performed by software For atomic bit setting/setting, the ODATAy bit can be set separately by writing to GPIOx_BSC or GPIOx_BR register
31:16	Reserved		

12.6.7 GPIO port set/reset register (GPIOx_BSC) (x=A..C, F)

Offset address: 0x18

Reset value: 0x0000 0000

Field	Name	R/W	Description
15:0	BSy	W	PortxPin y Set Bit (y=0…15) These bits can only be written, and the value of 0x0000 is returned when reading these bits. These bits are used to affect the corresponding ODATAy bits 0: No effect 1: Set the corresponding ODATAy bit
31:16	BCy	W	PortxPin y Reset Bit (y=0…15) These bits can only be written, and the value of 0x0000 is returned when reading these bits. These bits are used to affect the corresponding ODATAy bits 0: No effect 1: Clear corresponding ODATAy bit to 0 If BSy bit and BCy bit are set at the same time, BSy has the priority

12.6.8 GPIO port lock register (GPIOx_LOCK) (x=A…B)

This register protects the configuration of GPIO from being modified by mistake during the running of the program. If the GPIO configuration needs to be modified again, it can be modified only after the system is reset. When configuring GPIO locking function, it is necessary to execute the specified sequence to the register to enable the GPIO locking function.

Offset address: 0x1C

Reset value: 0x0000 0000

Field	Name	R/W	Description
15:0	LOCKy	R/W	PortxLock bit y Configure (y=0…15) 0: The configuration of Port x Pin y is not locked 1: The configuration of Port x Pin y is locked These bits can be read and written, but can only be written when LOCKKEY=0.
16	LOCKKEY	R/W	LOCK key This bit determines whether the port configuration lock key bit is activated 0: Not activated 1: Activate; GPIOx_LOCK register is locked until the next MUC reset is generated. Lock key write sequence: Write LOCK[16]=1+LOCK[15:0] Write LOCK[16]=0+LOCK[15:0] Write LOCK[16]=1+LOCK[15:0] Read LOCK Read LOCK[16]=1 (this read operation can be selected to confirm whether to activate the lock key) Note: (1) The value of LOCKy cannot be changed in the write sequence of operation lock key. (2) Any error in the write sequence of operation lock key will abort the lock. (3) After the first lock sequence on any bit of the port, any read access on the LOCKKEY bit will return "1" until the next MCU is reset or the peripheral is reset.
31:17	Reserved		

12.6.9 GPIO multiplexing function low 8-bit register (GPIOx_ALFL) (x=A…C, F)

Offset address: 0x20

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	ALFSELy	R/W	PortxPin y Alternate Function Select (y=0….7) These bits can be read by software to configure the multiplexing function of the port. ALFSELy selection: 0000: AF0 0001: AF1 0010: AF2 0011: AF3 0100: AF4 0101: AF5 0110: AF6 0111: AF7 1000: Reserved 1001: Reserved 1010: Reserved 1011: Reserved 1100: Reserved 1101: Reserved 1111: Reserved 1110: Reserved

12.6.10 GPIO multiplexing function high 8-bit register (GPIOx_ALFH) (x=A…C, F)

Offset address: 0x24

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	ALFSELy	R/W	PortxPin y Alernate Function Select (y=8….15) These bits can be read by software to configure the multiplexing function of the port. ALFSELy selection: 0000: AF0 0001: AF1 0010: AF2 0011: AF3 0100: AF4 0101: AF5 0110: AF6 0111: AF7 1000: Reserved 1001: Reserved 1010: Reserved 1011: Reserved 1100: Reserved 1101: Reserved 1111: Reserved 1110: Reserved

12.6.11 GPIO port reset register (GPIOx_BR) (x=A…C, F)

Offset address: 0x28

Reset value: 0x0000 0000

Field	Name	R/W	Description
15:0	BRy	W	PortxPin y Reset Configure (y=0…15) These bits can only be written, and the returned value is 0x0000 when reading these bits. These bits are used to affect the corresponding ODATA 0: No effect 1: Corresponding ODATA bit clears 0
31:16	Reserved		

13 Timer overview

13.1 Full name and abbreviation of terms

Table 42 Full Name and Abbreviation Description of Terms

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

13.2 Timer category and main difference

This series of products contains three types of timers: advanced timer, general-purpose timer and basic timer (watchdog timer is described in other chapters).

The advanced timer includes the functions of general-purpose timer and basic timer. The advanced timer has four capture/compare channels, supports timing function, input capture and output compare function, braking and complementary output function, and is a 16-bit timer that can count up/down.

The function of general-purpose timer is simpler than that of advanced timer. The main differences are the total number of channels, the number of complementary output channel groups and the braking function.

The basic timer is a timer that can only realize timing function without external interface.

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

Table 43 Main Differences among Timers Included in the Products

Item	Specific content/Category	Advanced timer	General-purpose timer		Basic timer	
Name	—	TMR1	TMR2	TMR3/4	TMR6	TMR7
Timebase unit	Counter	16 bits	32 bits	16 bits	16 bits	
	Prescaler	16 bits	16 bits	16 bits	16 bits	

Item	Specific content/Category	Advanced timer	General-purpose timer		Basic timer
	Counting mode	UP Down Center-aligned	UP Down Center-aligned	UP Down Center-aligned	UP
Channel	Input channel	4	4	4	0
	Capture/Compare channel	4	4	4	0
	Output channel	8	4	4	0
	Complementary output channel	3 sets	0	0	0
Function	Generate DMA request	Can	Can	Can	Can
	PWM mode	Yes	Yes	Yes	N/A
	Single-pulse mode	Yes	Yes	Yes	N/A
	Forced output mode	Yes	Yes	Yes	N/A
	Dead zone insertion	Yes	N/A	N/A	N/A

Timer term

Table 44 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 45 Definitions and Terms of Internal Signals

Name	Description
ETR	TMRx_ETR external trigger signal
ETRF	External trigger filter
ETRP	External trigger prescaler
-	
ITR，ITR0，ITR1	Internal trigger
TRGI	Clock/Trigger/Slave mode controller trigger input
TIF_ED	Timer input filter edge detection
-	

Name	Description
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 timer (TMR1)

14.1 Introduction

The advanced timer TMR1 takes the time base unit as the core, with the functions of input capture, output compare and braking input, and has a 16-bit autoload counter. The advanced timer supports complementary output, repeat count and programmable dead zone insertion functions, and is more suitable for motor control.

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

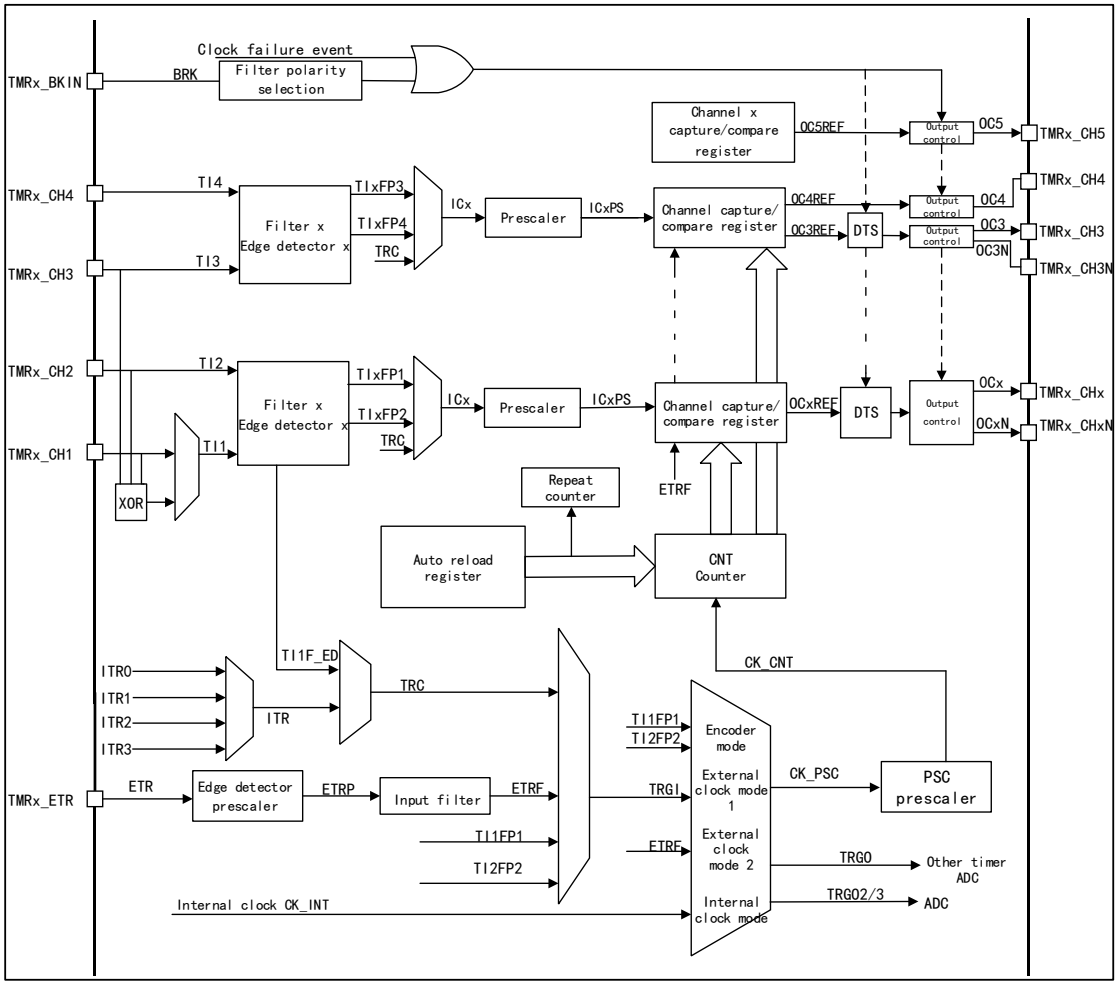
（7） Master/Slave mode controller of timer
- Timers can be synchronized and cascaded
- Support multiple slave modes and synchronization signals

（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

14.3 Structure block diagram

Figure 23 TMR1 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

- Counter register (CNT) 16 bits
- Autoreload register (AUTORLD) 16 bits
- Prescaler (PSC) 16 bits
- Repetition count register (REPCNT) 8 bits

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).

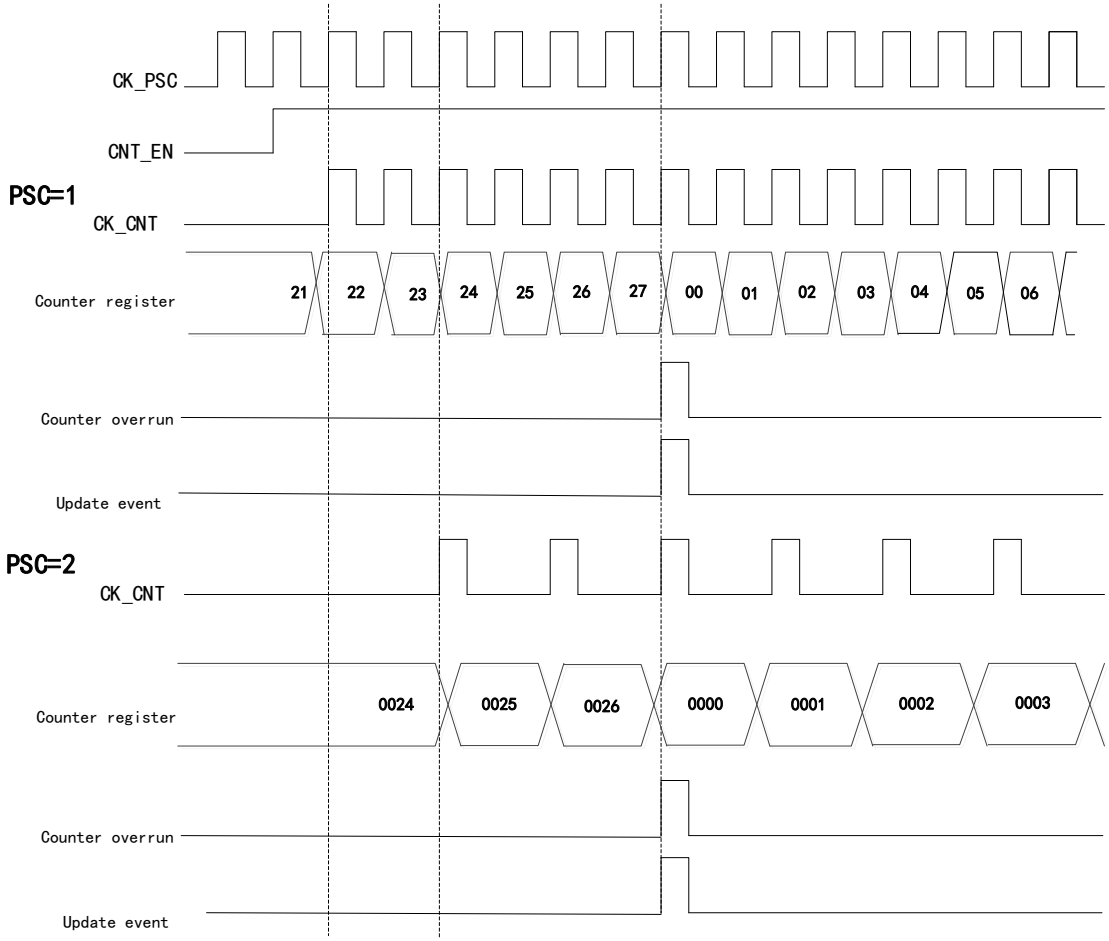
When the counter is in count-up mode, the counter will count up from 0; every time a pulse is generated, the counter will increase by 1 and when the value of the counter (TMRx_CNT) is equal to the value of the auto reload (TMRx_AUTORLD), the counter will start to count from 0 again, a count-up overrun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

When the counter overruns, an update event will be generated. 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 24 Timing Diagram of Count-up Mode when Division Factor is 1 or 2



Count-down mode

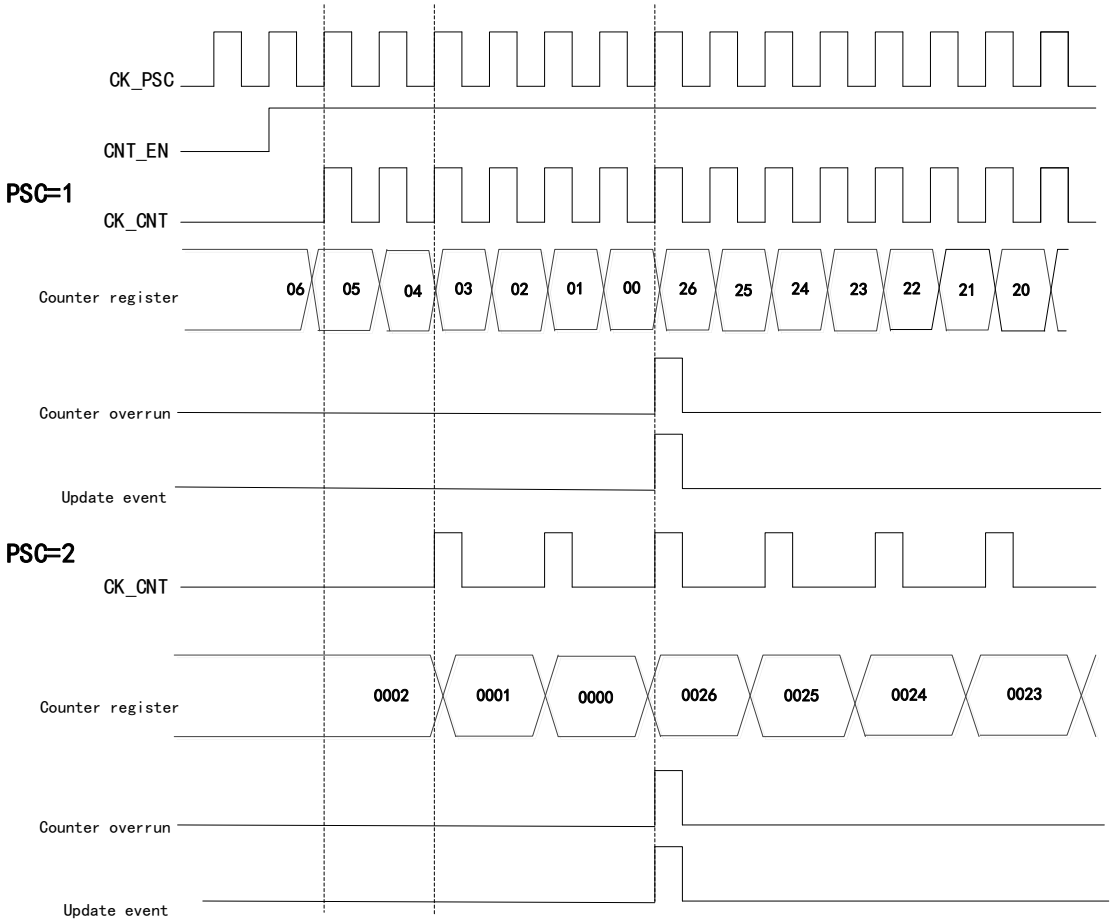
Set to the count-down mode by configuring CNTDIR bit of control register (TMRx_CTRL1).

When the counter is in count-down mode, the counter 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 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 25 Timing Diagram of Count-down mode when Division Factor is 1 or 2

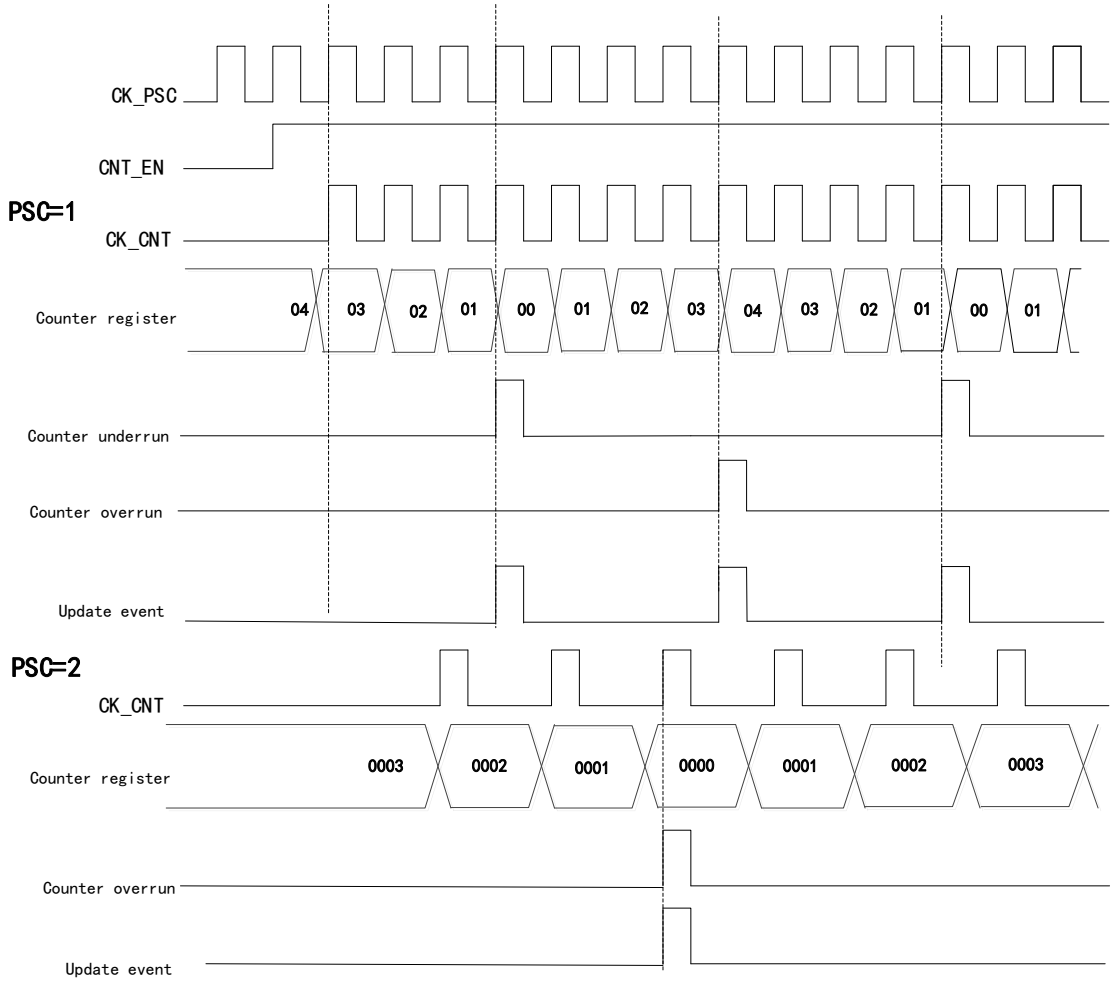


Center-aligned mode

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



Repeat counter REPCNT

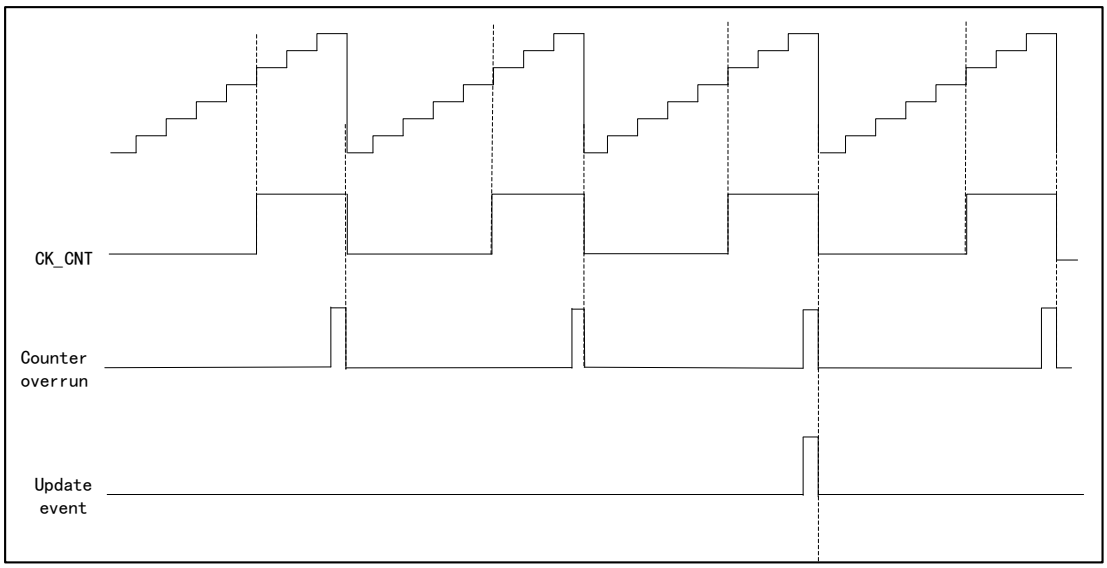
There is no repeat counter REPCNT in the basic/general-purpose timer, which means that when an overrun event or underrun event occurs in the basic/general-purpose timer, an update event will be generated directly; while in the advanced timer, because of the existence of the repeat counter, when an overrun/unerrrun 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 27 Timing Diagram of Count-up Mode when Setting REPCNT=2



Prescaler PSC

The 16-bit programmable prescaler can divide the clock frequency of the counter by any value from 1 to 65536 (controlled by TMRx_PSC register). The clock after frequency division 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 T1/2/3/4 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 when matching, channel x is invalid when matching, flip, force is invalid, forced to be valid, PWM1 and PWM2 modes, 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 edge 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 when CCx=5, AUTORLD=7

Figure 28 Timing Diagram of PWM1 Count-up Mode

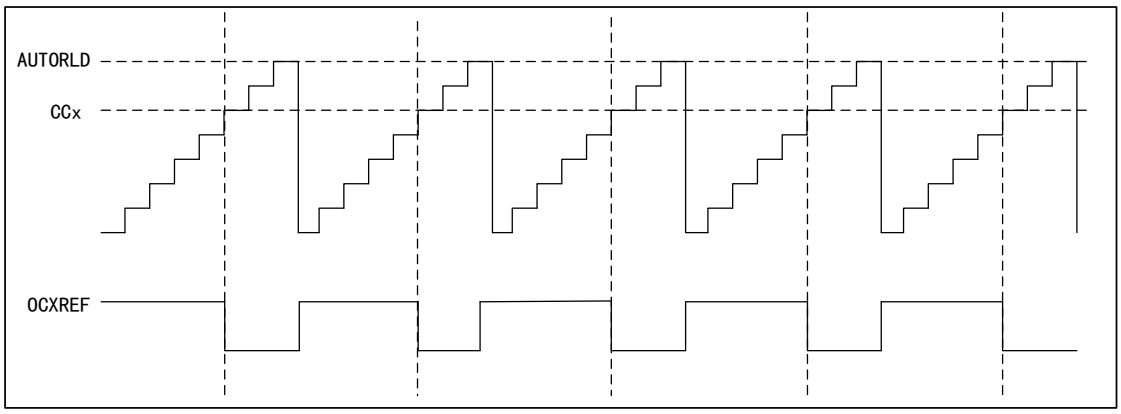


Figure 29 Timing Diagram of PWM1 Count-down Mode

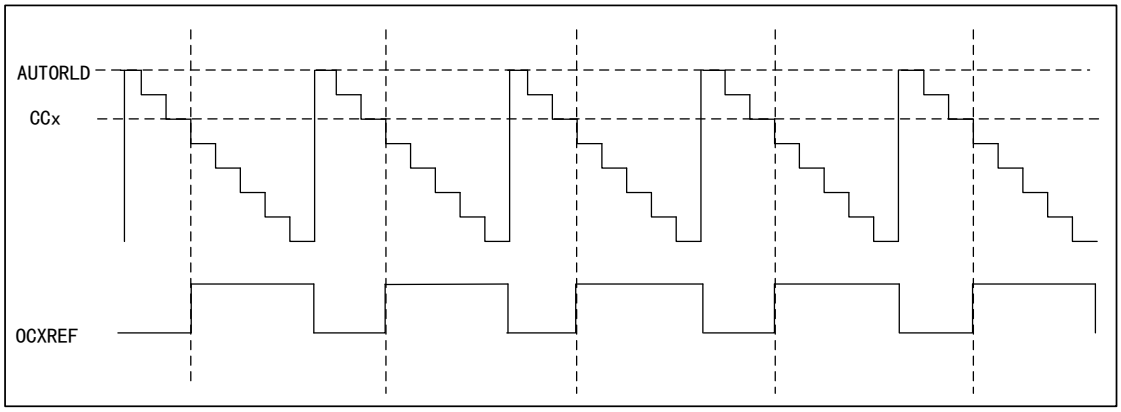
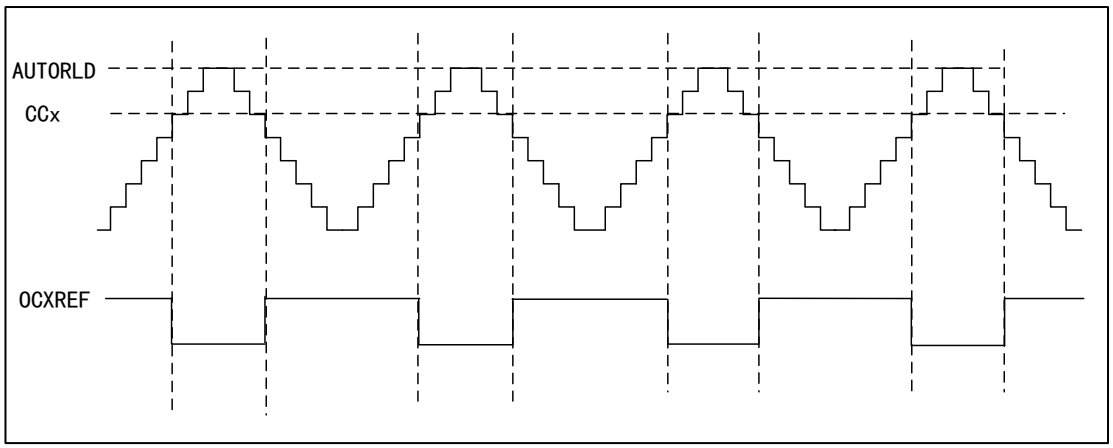


Figure 30 Timing Diagram of PWM1 Center-aligned Mode



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 31 Timing Diagram of PWM2 Count-up Mode

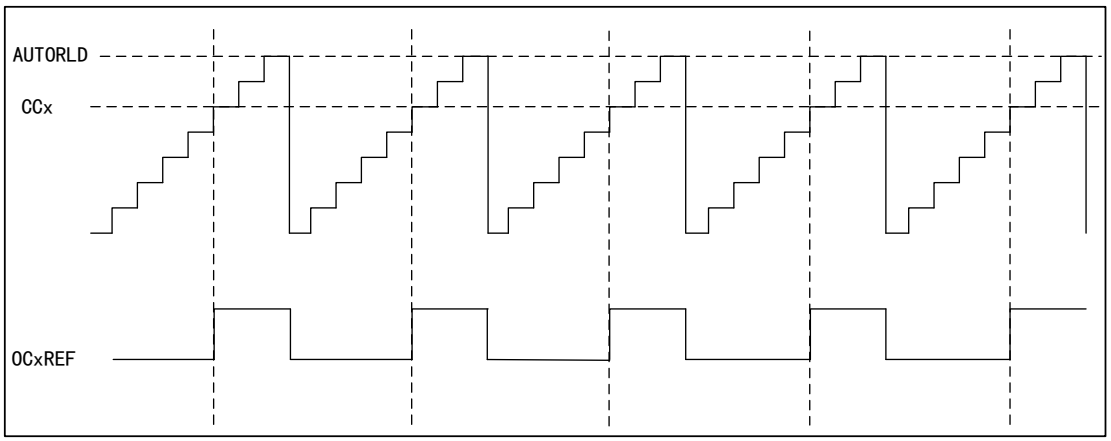


Figure 32 Timing Diagram of PWM2 Count-down Mode

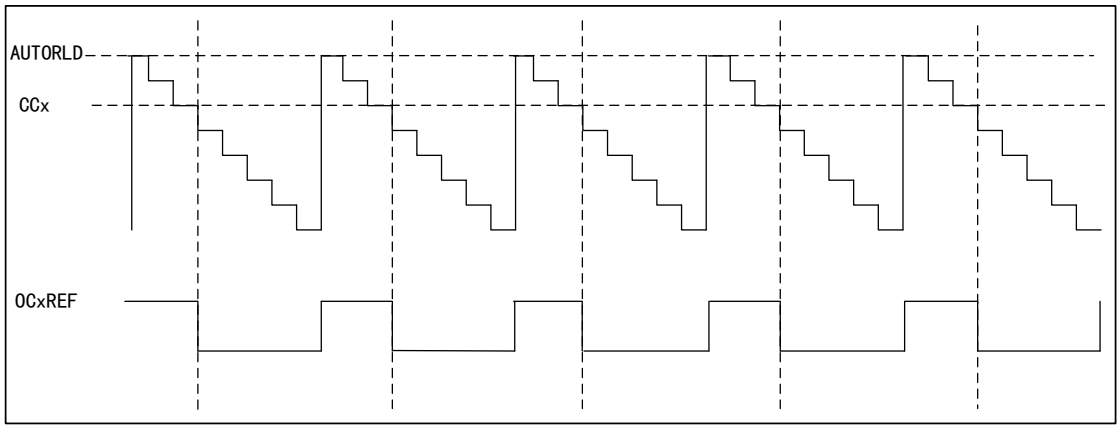
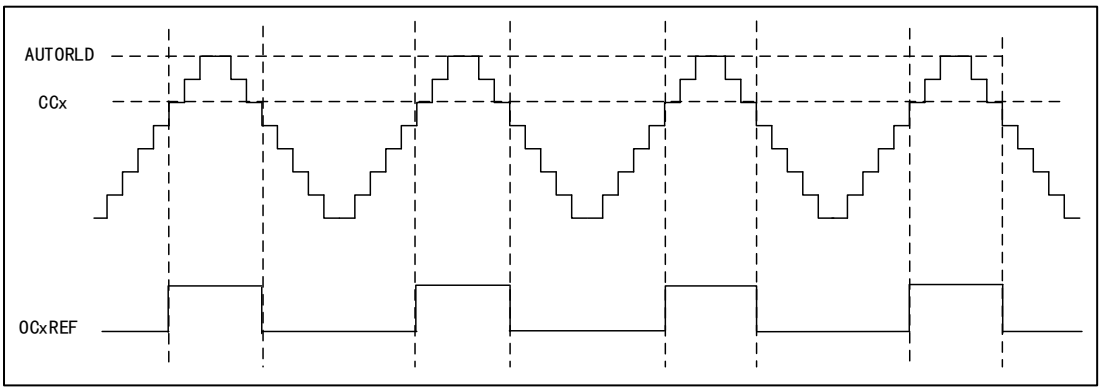


Figure 33 Timing Diagram of PWM2 Center-aligned Mode



14.4.6 PWM input mode

PWM input mode is a particular case of input capture.

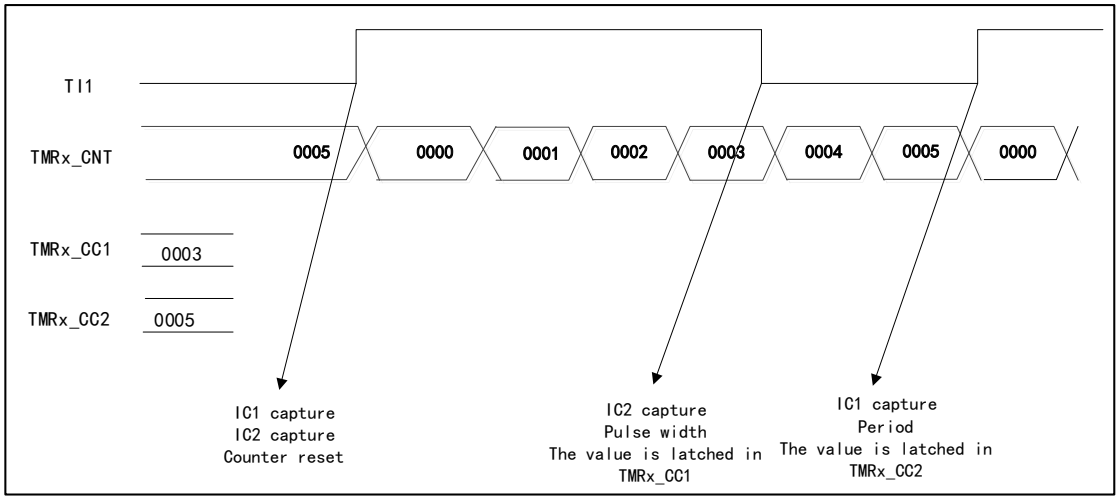
In PWM input mode, as only TI1FP1 and TI1FP2 are connected to the slave mode controller, input can be performed only through the channels TMRx_CH1

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 34 Timing Diagram of PWM Input Mode



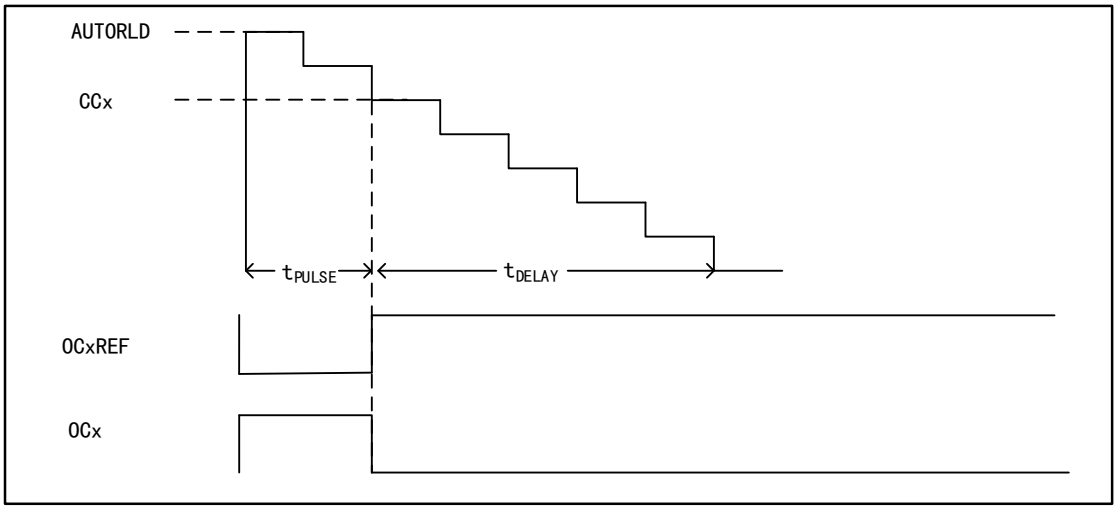
14.4.7 Single-pulse mode

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 35 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 corresponding complementary channel and timer run mode (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

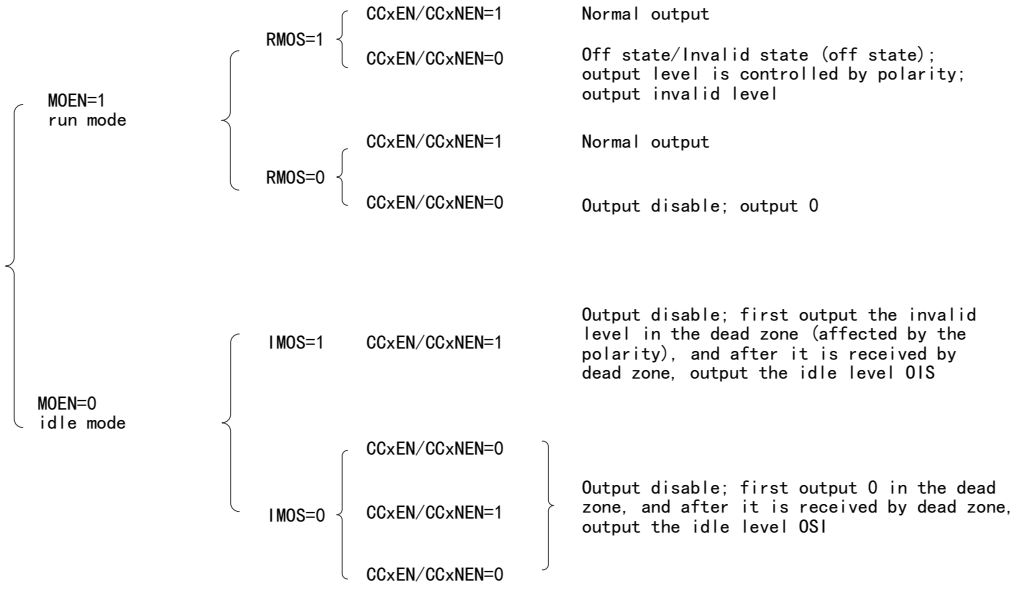
- Application environment of IMOS: In corresponding complementary channel and timer in idle mode (MOEN=0), the timer is not working (CCxEN=0, CCxNEN=0) or is working (CCxEN=1, CCxNEN=1)

（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 36 Register Structural Relationship Affecting Output Waveform



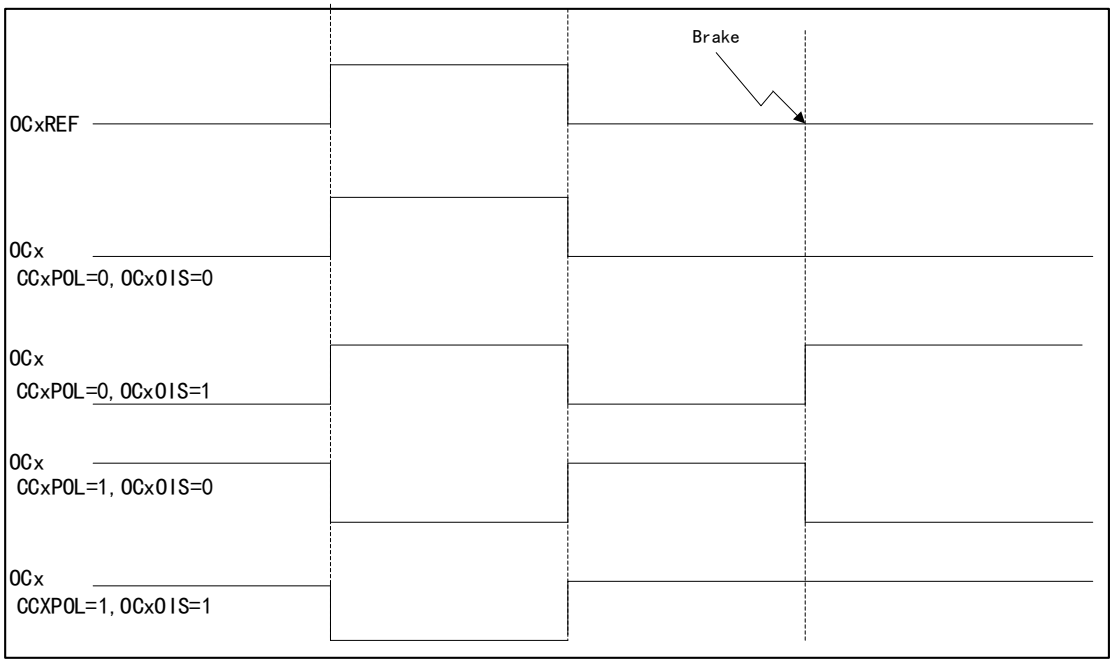
14.4.9 Braking function

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. TMRx_BKFT register can be configured with brake input signal filter.

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

Figure 37 Braking Event Timing Diagram

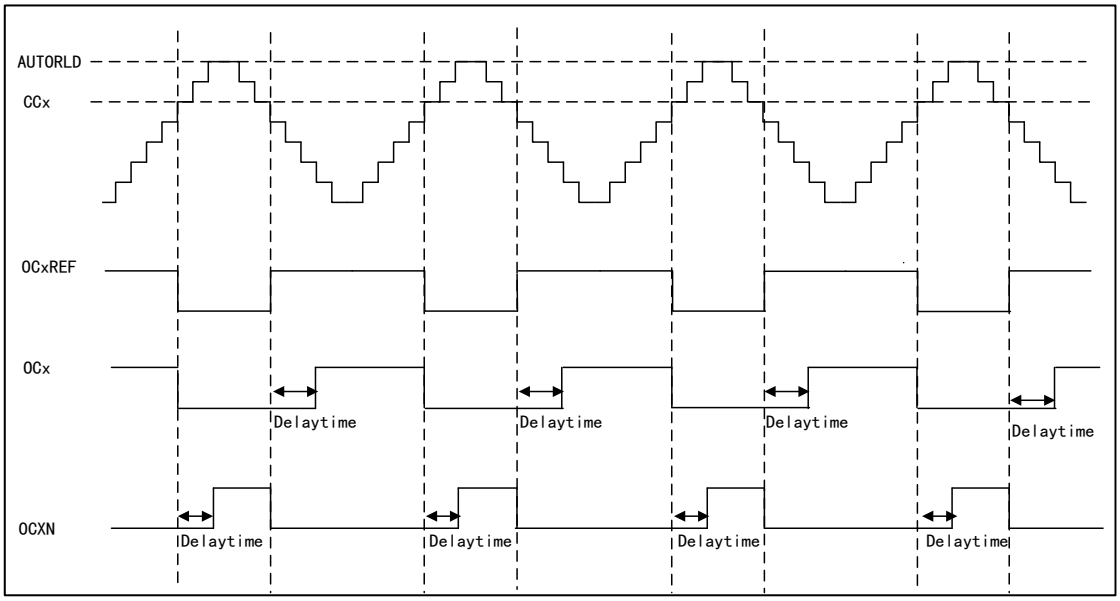


14.4.10 Complementary output and dead zone insertion

TMR1 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 38 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 46 Relationship between Count Direction and Encoder

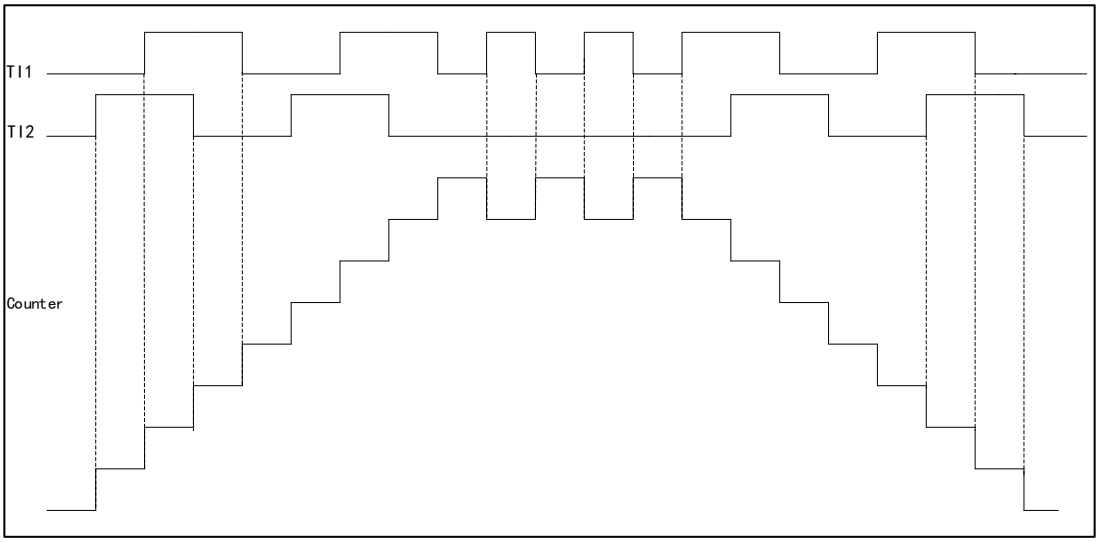
Effective edge		**Count only in TI1**		**Count only in TI2**		**Count in both TI1 and TI2**	
Level of relative signal		High	Low	High	Low	High	Low
TI1FP1	Rising edge	—		Count down	Count up	Count down	Count up
	Falling edge			Count up	Count down	Count up	Count down
TI2FP2	Rising edge	Count up	Count down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

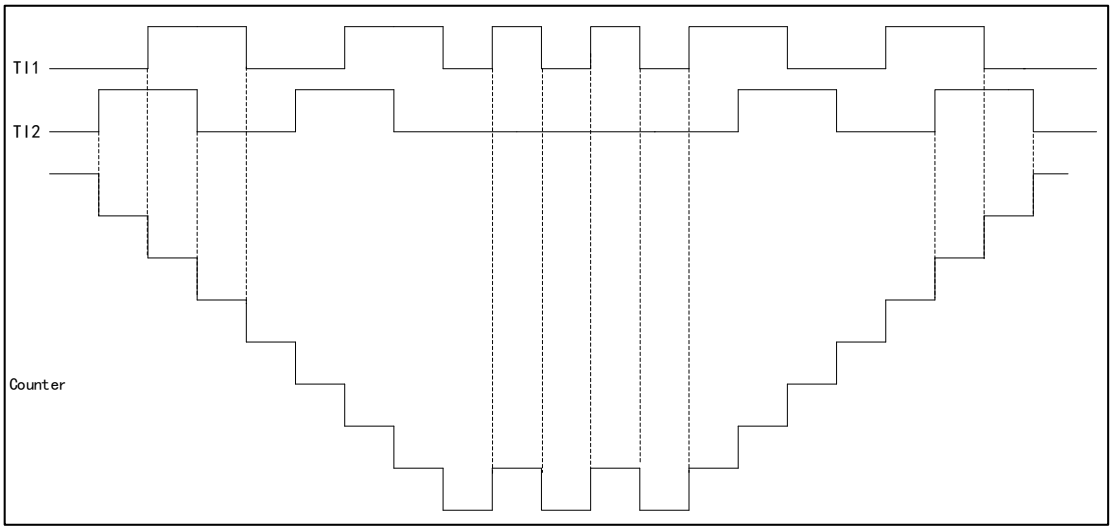
- TI1FP1 is mapped to TI1
- TI2FP2 is mapped to TI2
- Neither TI1FP1 nor TI2FP2 is phase-inverting
- The input signal is valid at the rising edge and falling edge
- Enable the counter

Figure 39 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 40 Example of Encoder Interface Mode of IC1FP1 Phase Inverting



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

TMR1 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.

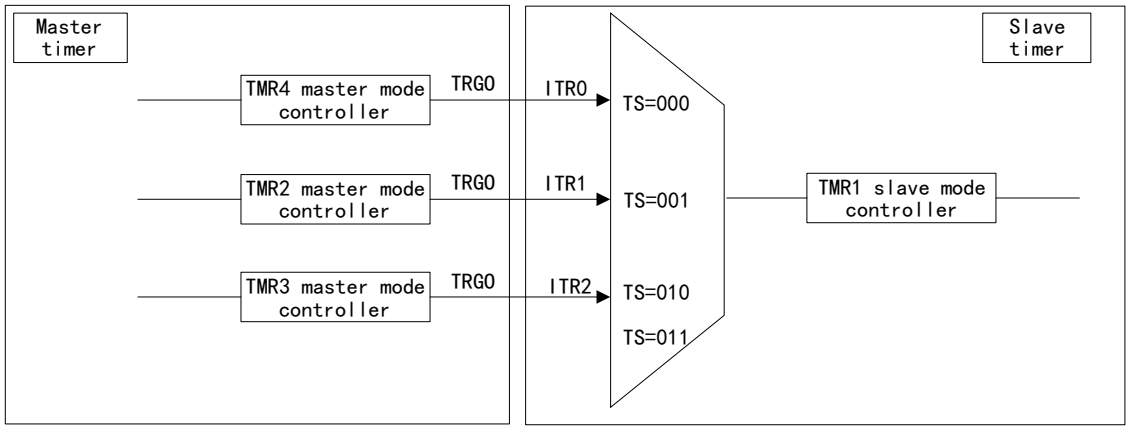
In the trigger mode, the enable of the counter depends on the event on the selected input, the counter will be enabled at the rising edge of the trigger input (but not be reset), and only the start of the counter is controlled.

14.4.14 Timer interconnection

Each timer of TMR1 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 41 Interconnection between TMR1 and Other Timer



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.

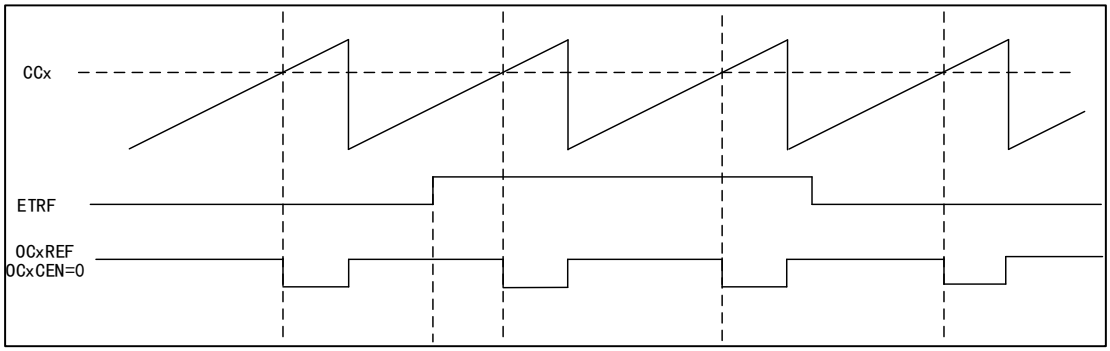
14.4.16 Clear OCxREF signal when an external event occurs

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.

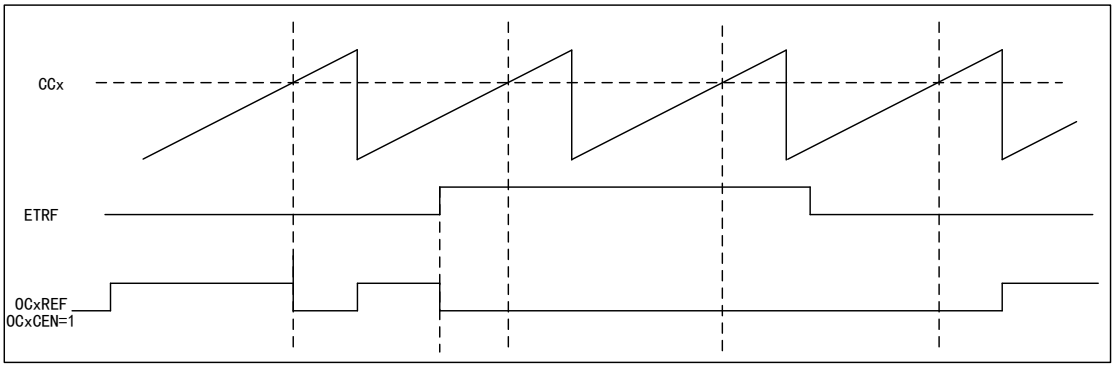
Set TMR1 to PWM mode, disable the external trigger prescaler, and disable the external trigger mode 2; when ETRF input is high, set OCxCEN=0, and the output OCxREF signal is shown in the figure below.

Figure 42 OCxREF Timing Diagram



Set TMR1 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 43 OCxREF Timing Diagram



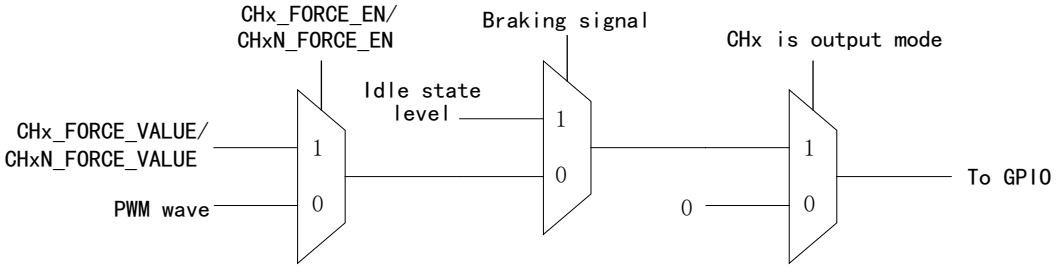
14.4.17 Manual control PWM output

When CCxEN/CCxNEN bit is 1 and MOEN bit is 1, if CHx_FORCE_EN/CHxN_FORCE_EN bit of TMRx_OUTPUTCTRL1 register is

1, the corresponding channel will output the value of CHx_FORCE_VALUE/CHxN_FORCE_VALUE bit of TMRx_OUTPUTCTRL2 register; if CHx_FORCE_EN /CHxN_FORCE_EN bit of TMRx_OUTPUTCTRL1 register is 0, the corresponding channel will output PWM waveform. Note that in the manual control PWM output mode, its output will be controlled by the PWM brake signal, while in the I/O output mode, it will not be controlled by such signal. Details are shown below:

When changing TMRx_OUTPUTCTRL1 and TMRx_OUTPUTCTRL2, if OUTPUTCTRL_BUF bit of TMRx_OUTPUTCTRL1 is 0, the change of the output of the manual control pin is synchronized with the system clock, and the change of the output status takes effect immediately; if OUTPUTCTRL_BUF bit of TMRx_OUTPUTCTRL1 is 1, the change of the output of the manual control pin is synchronized with the update event, and the output status takes effect at the next update event.

Figure 44 Logic Diagram of Output Control Module



Note: OUTPUTCTRL_BUF bit cannot be written with other bits of TMRx_OUTPUTCTRL1 register at the same time. If you need to modify OUTPUTCTRL_BUF bit and other control bits of TMRx_OUTPUTCTRL1 register, it is necessary to perform write operation twice. In any case, modification of OUTPUTCTRL_BUF bit takes effect synchronously with the clock.

14.4.18 Interaction of TMR1 with M0CP

In most motor control applications, the Ta, Tb and Tc output by SVPWM are actually the three-phase duty cycle of TMR1, which is a positive number less than 1. The software will multiply the three values by the PWM cycle and put the product into the PWM duty cycle register. To improve the efficiency, APM32F035x8T7 provides optional hardware scheme, configures M0CP as SVPWM calculation mode (see the coprocessor (M0CP) for details), and sets RUN to start the operation, the hardware will automatically multiply Ta, Tb and Tc by the PWM cycle in Qn format, and then automatically put the product into CCx and CCxC (x=0-2).

Note: The effective time of the calculation results after M0CP puts the calculation results into CCx/CCxC can be determined by CCx_NO_BUFFER bit of TMRx_M0CP register: when this bit is set to 0, the value written by

CCx/CCxC will take effect at the next update event; when this bit is set to 1, the value written by CCx/CCxC will take effect immediately.

It should be noted that when the PDCON bit in the TMR1 module is 0, Ta, Tb, and Tc are multiplied by the PWM cycle PWM_PERIOD; when PDCON is 1, Ta, Tb and Tc will be subtracted by 1 and then multiplied by the PWM cycle PWM_PERIOD, as shown in the following table:

Table 47 Relationship between PWMx Cycle and PDCON, Ta, Tb and Tc

CCx	Cycle (PDCON = 0)	Cycle (PDCON = 1)
CC1/CC1C	Ta * PWM_PERIOD	(1 – Ta)* PWM_PERIOD
CC2/CC2C	Tb * PWM_PERIOD	(1 – Tb)* PWM_PERIOD
CC3/CC3C	Tc * PWM_PERIOD	(1 – Tc)* PWM_PERIOD

Note: PWM_PERIOD can be selected by PWM_PERIOD_SEL bit of TMRx_M0CP from TMRx_AUTORLD register or TMRx_AUTORLD shadow register. The value of PWM_PDCON is controlled by PWM_PDCON_SEL bit of TMRx_M0CP.

In practical application, TMRx_M0CP register must be configured first. Then Ta, Tb and Tc shall be put into the operand registers X, Y, Z of M0CP (the same as the placement sequence of the results after SVPWM operation is completed), and then CTRL_REG of M0CP shall be configured to start the operation.

In the operation process, BUSY bit of STAT_REG of M0CP is always 1, and will be cleared by hardware after the operation is completed.

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 48 TMR1 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

Register name	Description	Offset address
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	Braking and dead zone register	0x44
TMRx_DCTRL	DMA control register	0x48
TMRx_DMADDR	DMA address register of continuous mode	0x4C
TMRx_CC1C	Channel 1 capture/compare register complementary register	0x50
TMRx_CC2C	Channel 2 capture/compare register complementary register	0x54
TMRx_CC3C	Channel 3 capture/compare register complementary register	0x58
TMRx_CC5	Channel 1 capture/compare register complementary register	0x5C
TMRx_CCM3	Capture/Compare mode register	0x60
TMRx_CTRL3	Control register 3	0x64
TMRx_BKFT	Brake filter register	0x68
TMRx_M0CP	Coprocessor square register	0x6C
TMRx_OUTPUTCTRL1	Output control register 1	0x70
TMRx_OUTPUTCTRL2	Output control register 2	0x74
TMRx_CTRL4	Control register 4	0x78

14.6 Register functional description

14.6.1 Control register 1 (TMRx_CTRL1)

Offset address: 0x00

Reset value: 0x0000

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.
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
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
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 be stopped, and the subsequent output level of the channel will no long be changed. 0: Disable 1: Enable
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
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)

Field	Name	R/W	Description
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
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
15:10	Reserved		

14.6.2 Control register 2 (TMRx_CTRL2)

Offset address: 0x04

Reset value: 0x0000

Field	Name	R/W	Description
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
1	Reserved		
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
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
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
7	TI1SEL	R/W	Timer Input 1 Select 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
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.
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 LOCKCFG bit in TMRx_BDT register is at the Level 1, 2 or 3, this bit cannot be modified.
10	OC2OIS	R/W	Configure OC2 output idle state. Refer to OC1OIS bit
11	OC2NOIS	R/W	Configure OC2N output idle state. Refer to OC1NOIS bit
12	OC3OIS	R/W	Configure OC3 output idle state. Refer to OC1OIS bit
13	OC3NOIS	R/W	Configure OC3N output idle state. Refer to OC1NOIS bit
14	OC4OIS	R/W	Configure OC4 output idle state. Refer to OC1OIS bit
15	Reserved		

14.6.3 Slave mode control register (TMRx_SMCTRL)

Offset address: 0x08

Reset value: 0x0000

Field	Name	R/W	Description
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.

Field	Name	R/W	Description
			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; the slave mode timer starts the counter to work after receiving the TRGI high level signal; it stops the counter when receiving TRGI low level signal; when receiving 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. 111: External clock mode 1; select the rising edge signal of TRGI as the clock source to drive the counter to work.
3	OCCSEL	R/W	OCREF Clear Source Select This bit is used to select OCREF clear source 0：OCREF_CLR 1：ETRF
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)
7	MSMEN	R/W	Master/slave Mode Enable 0: Invalid 1: Enable the master/slave mode
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

Field	Name	R/W	Description
			Sampling frequency=timer clock frequency/DIV; the filter length=N, and a jump is generated by every N events.
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 TMR1CLK 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
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 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.
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

Table 49 TMR1 Internal Trigger Connection

Slave timer	ITR0（TS=000）	ITR1（TS=001）	ITR2（TS=010）	ITR3（TS=011）
TMR1	TMR4	TMR2	TMR3	-

14.6.4 DMA/Interrupt enable register (TMRx_DIEN)

Offset address: 0x0C

Reset value: 0x0000

Field	Name	R/W	Description
0	UIEN	R/W	Update Interrupt Enable 0: Disable 1: Enable
1	CC1IEN	R/W	Capture/Compare Channel1 Interrupt Enable 0: Disable 1: Enable
2	CC2IEN	R/W	Capture/Compare Channel2 Interrupt Enable 0: Disable 1: Enable
3	CC3IEN	R/W	Capture/Compare Channel3 Interrupt Enable 0: Disable 1: Enable

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

14.6.5 Status register (TMRx_STS)

Offset address: 0x10

Reset value: 0x0000

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

Field	Name	R/W	Description
			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.
1	CC1IFLG	RC_W0	capture/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, set 1 by hardware; clear 0 by software or clear 0 when reading TMRx_CC1 register.
2	CC2IFLG	RC_W0	capture/Compare Channel2 Interrupt Flag Refer to STS_CC1IFLG
3	CC3IFLG	RC_W0	capture/Compare Channel3 Interrupt Flag Refer to STS_CC1IFLG
4	CC4IFLG	RC_W0	capture/Compare Channel4 Interrupt Flag Refer to STS_CC1IFLG
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.
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.
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.
8	Reserved		
9	CC1RCFLG	RC_W0	capture/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 is set to 1 by hardware and cleared to 0 by software only when the channel is configured as input capture.

Field	Name	R/W	Description
10	CC2RCFLG	RC_W0	capture/Compare Channel2 Repetition Capture Flag Refer to STS_CC1RCFLG
11	CC3RCFLG	RC_W0	capture/compare Channel3 Repetition Capture Flag Refer to STS_CC1RCFLG
12	CC4RCFLG	RC_W0	capture/Compare Channel4 Repetition Capture Flag Refer to STS_CC1RCFLG
15:13	Reserved		

14.6.6 Control event generation register (TMRx_CEG)

Offset address: 0x14

Reset value: 0x0000

Field	Name	R/W	Description
0	UEG	W	Update Event Generate 0: Invalid 1: Initialize the counter and generate the 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 reads the value of TMRx_AUTORLD; in center-aligned mode or count-up mode, the counter will be cleared to 0.
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.
2	CC2EG	W	Capture/Compare Channel2 Event Generation Refer to CC1EG description
3	CC3EG	W	Capture/Compare Channel3 Event Generation Refer to CC1EG description
4	CC4EG	W	Capture/Compare Channel4 Event Generation Refer to CC1EG description
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.
6	TEG	W	Trigger Event Generate

Field	Name	R/W	Description
			0: Invalid 1: Generate trigger event This bit is set to 1 by software and cleared to 0 automatically by hardware.
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.
15:8	Reserved		

14.6.7 Capture/Compare mode register 1 (TMRx_CCM1)

Offset address: 0x18

Reset value: 0x0000

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 OCxx in the register describes the function of the channel in the output mode, and the ICxx in the register describes the function of the channel in the input mode.

Output compare mode:

Field	Name	R/W	Description
1:0	CC1SEL	R/W	Capture/Compare Channel 1 Select 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).
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.
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.

Field	Name	R/W	Description
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 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.
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
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).
10	OC2FEN	R/W	Output Compare Channel2 Preload Enable
11	OC2PEN	R/W	Output Compare Channel2 Buffer Enable
14:12	OC2MOD	R/W	Output Compare Channel1 Mode
15	OC2CEN	R/W	Output Compare Channel2 Clear Enable

Input capture mode:

Field	Name	R/W	Description
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).

Field	Name	R/W	Description
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.
7:4	IC1F	R/W	Input Capture Channel1 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, indicating that a jump is generated by every N events.
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 TI1 10: CC2 channel is input, and IC2 is mapped on TI2 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).
11:10	IC2PSC	R/W	Input Capture Channel 2 Perscaler Configure
15:12	IC2F	R/W	Input Capture Channel2 Filter Configure

14.6.8 Capture/Compare mode register 2 （TMRx_CCM2）

Offset address: 0x1C

Reset value: 0x0000

Refer to the description of the above CCM1 register.

Output compare mode:

Field	Name	R/W	Description
1:0	CC3SEL	R/W	Capture/Compare Channel 1 Select 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

Field	Name	R/W	Description
			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).
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.
3	OC3PEN	R/W	Output Compare Channel3 Preload Enable
6:4	OC3MOD	R/W	Output Compare Channel3 Mode Configure)
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
9:8	CC4SEL	R/W	Capture/Compare Channel 4 Select 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).
10	OC4FEN	R/W	Output Compare Channel4 Preload Enable
11	OC4PEN	R/W	Output Compare Channel4 Buffer Enable
14:12	OC4MOD	R/W	Output Compare Channel4 Mode Configure
15	OC4CEN	R/W	Output Compare Channel4 Clear Enable

Input capture mode:

Field	Name	R/W	Description
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).
3:2	IC3PSC	R/W	Input Capture Channel3 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.

Field	Name	R/W	Description
7:4	IC3F	R/W	Input Capture Channel3 Perscaler Configure
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).
11:10	IC4PSC	R/W	Input Capture Channel 4 Perscaler Configure
15:12	IC4F	R/W	Input Capture Channel4 Filter Configure

14.6.9 Capture/Compare enable register (TMRx_CCEN)

Offset address: 0x20

Reset value: 0x0000

Field	Name	R/W	Description
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
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: CC1POL and CC1NPOL control the polarity of the triggered or captured signals TI1FP1 and TI2FP1 at the same time 00: Non-phase-inverting/rising edge: TIxFP1 is not phase-inverting (triggered in gated and encoder mode), and is captured at the rising edge of TIxFP1 (reset trigger, capture, external clock and trigger mode). 01: Phase inverting/falling edge: TIxFP1 is phase-inverting (triggered in gated and encoder mode), and is captured at the rising edge of TIxFP1 (reset trigger, capture, external clock and trigger mode). 10: Reserved 11: Non-phase-inverting/Rising and falling edges: TIxFP1 is not phase-inverting (triggered in gated mode, cannot be used in encoder mode), and is captured at the rising edge of TIxFP1 (reset trigger, capture, external clock and trigger mode).

Field	Name	R/W	Description
2	CC1NEN	R/W	Capture/Compare Channel1 Complementary Output Enable 0: Disable 1: Enable
3	CC1NPOL	R/W	Capture/Compare Channel1 Complementary Output Polarity When CC1 channel is configured as output 0: OC1N is active high 1: OC1N is active low When CC1 channel is configured as input This bit, together with CC1POL, defines the polarity of TI1FP1 and TI2FP1 Notes: On the complementary output channel, if this bit is preloaded, and CCPEN=1 for TMRx_CTRL2, CC1NPOL can obtain new value from the preload bit only when a reversing event is generated. When the protection level is 2 or 3, this bit cannot be modified
4	CC2EN	R/W	Capture/Compare Channel2 Output Enable Refer to CCEN_CC1EN
5	CC2POL	R/W	Capture/Compare Channel2 Output Polarity Configure Refer to CCEN_CC1POL
6	CC2NEN	R/W	Capture/Compare Channel1 Complementary Output Enable Refer to CCEN_CC1NEN
7	CC2NPOL	R/W	Capture/Compare Channel2 Complementary Output Polarity Configure Refer to CCEN_CC1NPOL
8	CC3EN	R/W	Capture/Compare Channel3 Output Enable Refer to CCEN_CC1EN
9	CC3POL	R/W	Capture/Compare Channel3 Output Polarity Configure Refer to CCEN_CC1POL
10	CC3NEN	R/W	Capture/Compare Channel3 Complementary Output Enable Refer to CCEN_CC1NEN
11	CC3NPOL	R/W	Capture/Compare Channel3 Complementary Output Polarity Configure Refer to CCEN_CC1NPOL
12	CC4EN	R/W	Capture/Compare Channel4 Output Enable Refer to CCEN_CC1EN
13	CC4POL	R/W	Capture/Compare Channel4 Output Polarity Refer to CCEN_CC1POL
15:14	Reserved		

14.6.10 Counter register (TMRx_CNT)

Offset address: 0x24

Reset value: 0x0000

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

14.6.11 Prescaler register (TMRx_PSC)

Offset address: 0x28

Reset value: 0x0000

Field	Name	R/W	Description
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)

Offset address: 0x2C

Reset value: 0xFFFF

Field	Name	R/W	Description
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 counter register (TMRx_REPCNT)

Offset address: 0x30

Reset value: 0x0000

Field	Name	R/W	Description
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.
15:8	Reserved		

14.6.14 Channel 1 capture/compare register (TMRx_CC1)

Offset address: 0x34

Reset value: 0x0000

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 comparison result when an update event is generated.

14.6.15 Channel 2 capture/compare register (TMRx_CC2)

Offset address: 0x38

Reset value: 0x0000

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

14.6.16 Channel 3 capture/compare register (TMRx_CC3)

Offset address: 0x3C

Reset value: 0x0000

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

14.6.17 Channel 4 capture/compare register (TMRx_CC4)

Offset address: 0x40

Reset value: 0x0000

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

14.6.18 Brake and dead zone register (TMRx_BDT)

Offset address: 0x44

Reset value: 0x0000

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
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 16μs to 31750ns; If the step time is 1μs, the dead time can be set from 32μs to 63μs; If the step time is 2μs, the dead time can be set from 64μs to 126μs. Note: Once LOCK level (LOCKCFG bit in TMRx_BDT register) is set to 1, 2 or 3, these bits cannot be modified.
9:8	LOCKCFG	R/W	Lock Write Protection Mode Configure 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.

Field	Name	R/W	Description
			10: Lock write protection level 2 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.
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
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 outptus invalid level (the specific level value is affected by the polarity configuration)
12	BRKEN	R/W	Brake Function Enable 0: Disable 1: Enable 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.
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.
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, clear 0 by hardware asynchronously. Note: Setting 1 by software or setting 1 automatically depends on AOEN bit of the TMRx_BDT register.

14.6.19 DMA control register (TMRx_DCTRL)

Offset address: 0x48

Reset value: 0x0000

Field	Name	R/W	Description
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 …….
7:5	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=TMR1_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.
15:13	Reserved		

14.6.20 DMA address register of continuous mode (TMRx_DMADDR)

Offset address: 0x4C

Reset value: 0x0000

Field	Name	R/W	Description
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;

Field	Name	R/W	Description
			"DMA index" is the offset automatically controlled by DMA, and it depends on DBLEN defined in TMRx_DCTRL register.

14.6.21 Channel 1 capture/compare register complementary register (TMRx_CC1C)

Offset address: 0x50

Reset value: 0x0000

Field	Name	R/W	Description
15:0	CC1C	R/W	Channel 1 Capture/Compare Register Complementary Register When the capture/compare channel 1 is configured as PWM output mode, the counter is in the center-aligned mode, and the asymmetric PWM output mode is enabled (OC1AEN of TMRx_CTRL3 register is 1), CC1C contains the value currently loaded in the capture/compare register complementary register. When the counter is a count-up counter, compare the value CC1 of the capture/compare channel 1 with the value CNT of the counter, and the output signal is generated on OC1. When the counter counts down, compare the value of CC1C 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 comparison result when an update event is generated.

14.6.22 Channel 2 capture/compare register complementary register (TMRx_CC2C)

Offset address: 0x54

Reset value: 0x0000

Field	Name	R/W	Description
15:0	CC2C	R/W	Channel 2 Capture/Compare Register Complementary Register Refer to TMRx_CC1C

14.6.23 Channel 3 capture/compare register complementary register (TMRx_CC3C)

Offset address: 0x58

Reset value: 0x0000

Field	Name	R/W	Description
15:0	CC3C	R/W	Channel 3 Capture/Compare Register Complementary Register Refer to TMRx_CC1C

14.6.24 Channel 5 capture/compare register (TMRx_CC5)

Offset address: 0x5C

Reset value: 0x0000

Field	Name	R/W	Description
15:0	CC5	R/W	Capture/Compare Channel5 Value Refer to TMRx_CC1

14.6.25 Capture/Compare mode register 3 (TMRx_CCM3)

Offset address: 0x60

Reset value: 0x0000

Channel 5 only has output compare function:

Field	Name	R/W	Description
0	CC5EN	R/W	Capture/Compare Channel5 Output Enable 0: Disable output 1: Enable output
1	CC5POL	R/W	Capture/Compare Channel5 Output Polarity Configure） 0: OC1N is active high 1: OC1N is active low Note: Once LOCK level (LOCKCFG bit in TMRx_BDT register) is set to 1, 2 or 3, these bits cannot be modified.
2	OC5FEN	R/W	Output Compare Channel5 Fast Enable 0: Disable 1: Enable This bit is used to improve the response of the capture/compare output to the trigger input event.
3	OC5PEN	R/W	Output Compare Channel5 Preload Enable Refer to OC1PEN bit. Note: Once LOCK level (LOCKCFG bit in TMRx_BDT register) is set to 3, these bits cannot be modified.
6:4	OC5MOD	R/W	Output Compare Channel5 Mode Configure） Refer to OC1MOD bit. Note: Once LOCK level (LOCKCFG bit in TMRx_BDT register) is set to 3, these bits cannot be modified.
7	OC5CEN	R/W	Output Compare Channel5 Clear Enable 0: OC5REF is unaffected by ETRF input. 1: When ETRF input is high, OC5REF=0
8	OC5OIS	R/W	OC5 Output Idel State Configure Note: Once LOCK level (LOCKCFG bit in TMRx_BDT register) is set to 1, 2 or 3, these bits cannot be modified.
15:9	Reserved		

14.6.26 Control register 3 (TMRx_CTRL3)

Offset address: 0x64

Reset value: 0x0000

Field	Name	R/W	Description
0	MMSZE	R/W	When the counter returns to 0, TRGO signal is generated The generated TRGO is superimposed with TRGO signal selected by MMSEL and MMSEL2. It is valid only when the counter is in the center-aligned mode, and valid only when the TRGO selected by MMSEL or MMSEL2 is a pulse signal (invalid in enable mode or compare mode 1-5). 1: TRGO is generated when the counter returns to 0 0: No TRGO is generated when the counter returns to 0
1	MMSPE	R/W	TRGO Signal is Generated when the Counter Matches the Autoreload Register

Field	Name	R/W	Description
			The generated TRGO is superimposed with TRGO signal selected by MMSEL and MMSEL2. It is valid only when the counter is in the center-aligned mode, and valid only when the TRGO selected by MMSEL or MMSEL2 is a pulse signal (invalid in enable mode or compare mode 1-5). 1: TRGO is generated when the counter matches the autoreload register 0: No TRGO is generated when the counter matches the autoreload register
3:2	Reserved		
7:4	MMSEL2	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. 0xxx: TRGO is selected by MMSEL [2:0] 1000: TRGO is not generated (TRGO will be generated only according to MMSZE and MMSPE) 1001: Compare mode 5; OC5REF is used to trigger TRGO 1010: OC4REF rising edge and falling edge generate TRGO 1011: OC5REF rising edge and falling edge generate TRGO 1100: OC4REF rising edge and OC5REF rising edge generate TRGO 1101: OC4REF falling edge and OC5REF falling edge generate TRGO 1110: OC4REF rising edge and OC5REF falling edge generate TRGO 1111: OC4REF falling edge and OC5REF rising edge generate TRGO
8	OC1AEN	R/W	Channel 1 Asymmetric PWM Output Mode Enable Valid only when the counter is in the center symmetry mode and channel 1 is configured as PWM output mode. In the asymmetric PWM mode, when the counter counts up, OC1REF is controlled by CC1, when the counter counts down, OC1REF is controlled by CC1C. 1: Enable asymmetric PWM output mode 0: CC1C does not affect OC1REF output Note: Once LOCK level (LOCKCFG bit in TMRx_BDT register) is set to 3, these bits cannot be modified.
9	OC2AEN	R/W	Output Compare Channel2 Asynchronous Mode Configure Refer to OC1AEN. Note: Once LOCK level (LOCKCFG bit in TMRx_BDT register) is set to 3, these bits cannot be modified.
10	OC3AEN	R/W	Output Compare Channel3 Asynchronous Mode Configure Refer to OC1AEN. Note: Once LOCK level (LOCKCFG bit in TMRx_BDT register) is set to 3, these bits cannot be modified.
15:11	Reserved		

14.6.27 Brake filter register (TMRx_BKFT)

Offset address: 0x68

Reset value: 0x0000

Field	Name	R/W	Description
3:0	BKF	R/W	External Brake Input Filter 0000: Disable filter, sampled by f_{DTS} 0001：DIV=1，N=2

Field	Name	R/W	Description
			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. Note: When the protection level is 1, this bit cannot be modified.
15:4	Reserved		

14.6.28 Coprocessor register (TMRx_M0CP)

Offset address: 0x6c

Reset value: 0x4824

Field	Name	R/W	Description
1:0	CH1_WDATA_SEL	R/W	Channel 1 Data Selection Register 00: CCR1 register writes 15-0 bits of M0CP outgoing data 01: CCR1 register writes 31-16 bits of M0CP outgoing data 10: CCR1 register writes 47-32 bits of M0CP outgoing data 11: Reserved
3:2	CH2_WDATA_SEL	R/W	Channel 2 Data Selection Register 00: CCR2 register writes 15-0 bits of M0CP outgoing data 01: CCR2 register writes 31-16 bits of M0CP outgoing data 10: CCR2 register writes 47-32 bits of M0CP outgoing data 11: Reserved
5:4	CH3_WDATA_SEL	R/W	Channel 3 Data Selection Register 00: CCR3 register writes 15-0 bits of M0CP outgoing data 01: CCR3 register writes 31-16 bits of M0CP outgoing data 10: CCR3 register writes 47-32 bits of M0CP outgoing data 11: Reserved
6	PWM_PERIOD_SEL	R/W	PWM Cycle Selection Register 0: The cycle is the value of TMRx_AUTORLD shadow register 1: The cycle is the value of TMRx_AUTORLD register
7	CCR_NO_BUF	R/W	CCR Buffer Enable Register 0: Update of CCx/CCxN register by M0CP takes effect at the next update event 1: Update of CCx/CCxN register by M0CP will take effect immediately
8	PWM_PDCON_SEL	R/W	PDCON Select Register 0: PDCON is 0

Field	Name	R/W	Description
			1: PDCON is 1
10:9	CH1N_WDATA_SEL	R/W	Channel 1N Data Selection Register 00: CCR1 register writes 31-16 bits of M0CP outgoing data 01: CCR1 register writes 63-48 bits of M0CP outgoing data 10: CCR1 register writes 95-80 bits of M0CP outgoing data 11: Reserved
12:11	CH2N_WDATA_SEL	R/W	Channel 2N Data Selection Register 00: CCR2 register writes 31-16 bits of M0CP outgoing data 01: CCR2 register writes 63-48 bits of M0CP outgoing data 10: CCR2 register writes 95-80 bits of M0CP outgoing data 11: Reserved
14:13	CH3N_WDATA_SEL	R/W	Channel 3N Data Selection Register 00: CCR3 register writes 31-16 bits of M0CP outgoing data 01: CCR3 register writes 63-48 bits of M0CP outgoing data 10: CCR3 register writes 95-80 bits of M0CP outgoing data 11: Reserved
15	SVPWM_EN	R/W	SVPMW Function Enable Register 0: M0CP will not update the value in CCx/CCxN 1: M0CP will update the value in CCx/CCxN

14.6.29 Output control register 1 (TMRx_OUTPUTTCTRL1)

Offset address: 0x70

Reset value: 0x0000

Field	Name	R/W	Description
0	CH1_FORCE_EN	R/W	Channel 1 Outputs the Control Enable Register 0: Channel 1 outputs PWM waveform 1: The output of channel 1 is controlled by the corresponding bit of output control register 2
1	CH1N_FORCE_EN	R/W	Complementary Channel Output Control Enable Register for Channel 1 0: Complementary channel of channel 1 outputs PWM waveform 1: The output of complementary channel of channel 1 is controlled by the corresponding bit of output control register 2
2	CH2_FORCE_EN	R/W	Channel 2 Outputs the Control Enable Register 0: Channel 2 outputs PWM waveform 1: The output of channel 2 is controlled by the corresponding bit of output control register 2
3	CH2N_FORCE_EN	R/W	Complementary Channel Output Control Enable Register for Channel 2 0: Complementary channel of channel 2 outputs PWM waveform 1: The output of complementary channel of channel 2 is controlled by the corresponding bit of output control register 2
4	CH3_FORCE_EN	R/W	Channel 3 Outputs the Control Enable Register 0: Channel 3 outputs PWM waveform 1: The output of channel 3 is controlled by the corresponding bit of output control register 2

Field	Name	R/W	Description
5	CH3N_FORCE_EN	R/W	Complementary Channel Output Control Enable Register for Channel 3 0: Complementary channel of channel 3 outputs PWM waveform 1: The output of complementary channel of channel 1 is controlled by the corresponding bit of output control register 3
6	CH4_FORCE_EN	R/W	Channel 4 Outputs the Control Enable Register 1: The output of channel 1 is controlled by the corresponding bit of output control register 4 0: Channel 4 outputs PWM waveform
7	CH5_FORCE_EN	R/W	Channel 5 Outputs the Control Enable Register 0: Channel 5 outputs PWM waveform 1: The output of channel 5 is controlled by the corresponding bit of output control register 2
14:8	Reserved		
15	OUTPUTCTRL_BUF	R/W	Output Control Buffering is Enable 0: The modification of output control register takes effect immediately 1: The modification of output control register takes effect at the next update event

Note: Once LOCK level (LOCKCFG bit in TMRx_BDT register) is set to 1, 2 or 3, this register cannot be modified.

14.6.30 Output control register 2 (TMRx_OUTPUTCTRL2)

Offset address: 0x04

Reset value: 0x0000

Field	Name	R/W	Description
0	CH1_FORCE_VALUE	R/W	Channel 1 Output Level Register 0: Channel 1 outputs low level 1: Channel 1 outputs high level
1	CH1N_FORCE_VALUE	R/W	Complementary Channel Output Level Register for Channel 1 0: Complementary channel of channel 1 outputs low level 1: Complementary channel of channel 1 outputs high level
2	CH2_FORCE_VALUE	R/W	Channel 2 Output Level Register 0: Channel 2 outputs low level 1: Channel 2 outputs high level
3	CH2N_FORCE_VALUE	R/W	Complementary Channel Output Level Register for Channel 2 0: Complementary channel of channel 2 outputs low level 1: Complementary channel of channel 2 outputs high level
4	CH3_FORCE_VALUE	R/W	Channel 3 Output Level Register 0: Channel 3 outputs low level 1: Channel 3 outputs high level

Field	Name	R/W	Description
5	CH3N_FORCE_VALUE	R/W	Complementary Channel Output Level Register for Channel 3 0: Complementary channel of channel 3 outputs low level 1: Complementary channel of channel 3 outputs high level
6	CH4_FORCE_VALUE	R/W	Channel 4 Output Level Register 0: Channel 4 outputs low level 1: Channel 4 outputs high level
7	CH5_FORCE_VALUE	R/W	Channel 5 Output Level Register 0: Channel 5 outputs low level 1: Channel 5 outputs high level
15:8	Reserved		

Note: First configure TMRx_OUTPUTCTRL2 and then TMRx_OUTPUTCTRL1. If it is necessary to change the output control buffer function and modify the values of other control bits in TMRx_OUTPUTCTRLx: first write OUTPUTCTRL_BUF of TMRx_OUTPUTCTRL, and then write TMRx_OUTPUTCTRL2 register and TMRx_OUTPUTCTRL1 register.

14.6.31 Control register 4 (TMRx_CTRL4)

Offset address: 0x78

Reset value: 0x0000

Field	Name	R/W	Description
0	MM2SZE	R/W	When the counter returns to 0, TRGO2 signal is generated The generated TRGO2 is superimposed with the TRGO2 signal selected by MM2SEL. It is valid only when the counter is in the center-aligned mode, and valid only when the TRGO selected by MM2SEL is a pulse signal (invalid in enable mode or compare mode 1-5). 0: TRGO2 is not generated when the counter returns to 0 1: TRGO2 is generated when the counter returns to 0
1	MM2SPE	R/W	TRGO2 Signal is Generated when the Counter Matches the Autoreload Register The generated TRGO2 is superimposed with the TRGO2 signal selected by MM2SEL. It is valid only when the counter is in the center-aligned mode, and valid only when the TRGO selected by MM2SEL is a pulse signal (invalid in enable mode or compare mode 1-5). 0: TRGO2 is not generated when the counter matches the autoreload register 1: TRGO2 is generated when the counter matches the autoreload register
3:2	Reserved		
7:4	MM2SEL	R/W	Master Mode Signal Select The signals of timers working in master mode can be used for TRGO2, which affects 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. 0000: Reset; the reset signal of master mode timer is used for TRGO2 0001: Enable; the counter enable signal of master mode timer is used for TRGO2

Field	Name	R/W	Description
			0010: Update; the update event of master mode timer is used for TRGO2 0011: Compare pulses; when the master mode timer captures/compares successfully (CCxIFLG=1), a pulse signal is output for TRGO2 0100: Compare mode 1; OC1REF is used to trigger TRGO2 0101: Compare mode 2; OC2REF is used to trigger TRGO2 0110: Compare mode 3; OC3REF is used to trigger TRGO2 0111: Compare mode 4; OC4REF is used to trigger TRGO2 1000: TRGO2 is not generated (TRGO2 will be generated only according to MM2SZE and MM2SPE) 1001: Compare mode 5; OC5REF is used to trigger TRGO2 1010: OC4REF rising edge and falling edge generate TRGO2 1011: OC5REF rising edge and falling edge generate TRGO2 1100: OC4REF rising edge and OC5REF rising edge generate TRGO2 1101: OC4REF falling edge and OC5REF falling edge generate TRGO2 1110: OC4REF rising edge and OC5REF falling edge generate TRGO2 1111: OC4REF falling edge and OC5REF rising edge generate TRGO2
8	MM3SZE	R/W	When the counter returns to 0, TRGO3 signal is generated Refer to MM2SZE bit. 0: TRGO3 is not generated when the counter returns to 0 1: TRGO3 is generated when the counter returns to 0
9	MM3SPE	R/W	TRGO3 Signal is Generated when the Counter Matches the Autoreload Register Refer to MM2SPE bit. 0: TRGO3 is not generated when the counter matches the autoreload register 1: TRGO3 is generated when the counter matches the autoreload register
11:10	Reserved		
15:12	MM3SEL	R/W	Master Mode Signal Select Refer to MM2SEL bit.

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. Include a 16-bit or 32-bit auto reload counter (realize up, down and center-aligned counting).

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

15.2 Main characteristics

（1） Timebase unit
- Counter: 16-bit (TMR3/4) or 32-bit (TMR2) counter, which can realize up, down and center-aligned counting
- Prescaler: 16-bit programmable prescaler
- Autoreload function

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

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

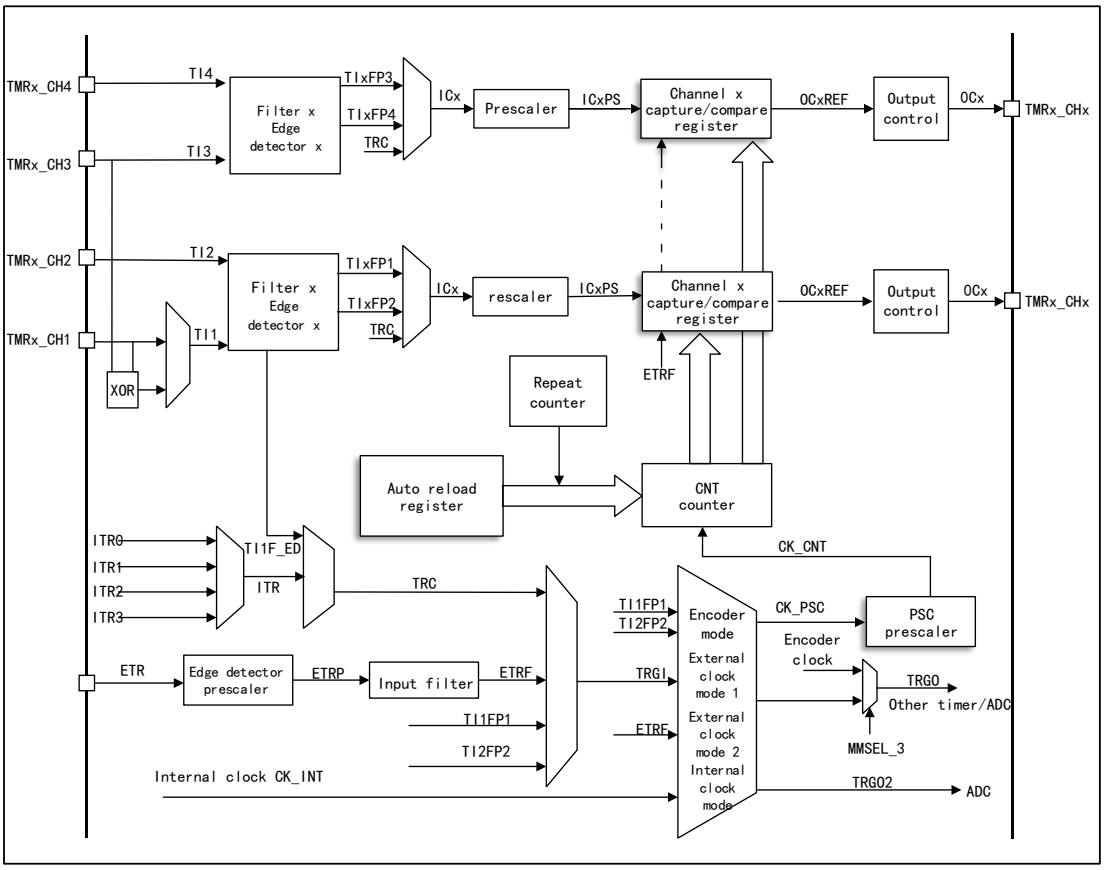
（4） Output function
- PWM output mode
- Forced output mode
- Single-pulse mode

（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

15.3 Structure block diagram

Figure 45 General-purpose Timer TMR2/3/4 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. Specially 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

- Counter register (CNT) 16 bits or 32 bits
- Auto reload register (AUTORLD) 16/32 bits
- Prescaler (PSC) 16/32 bits

Counter CNT

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

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

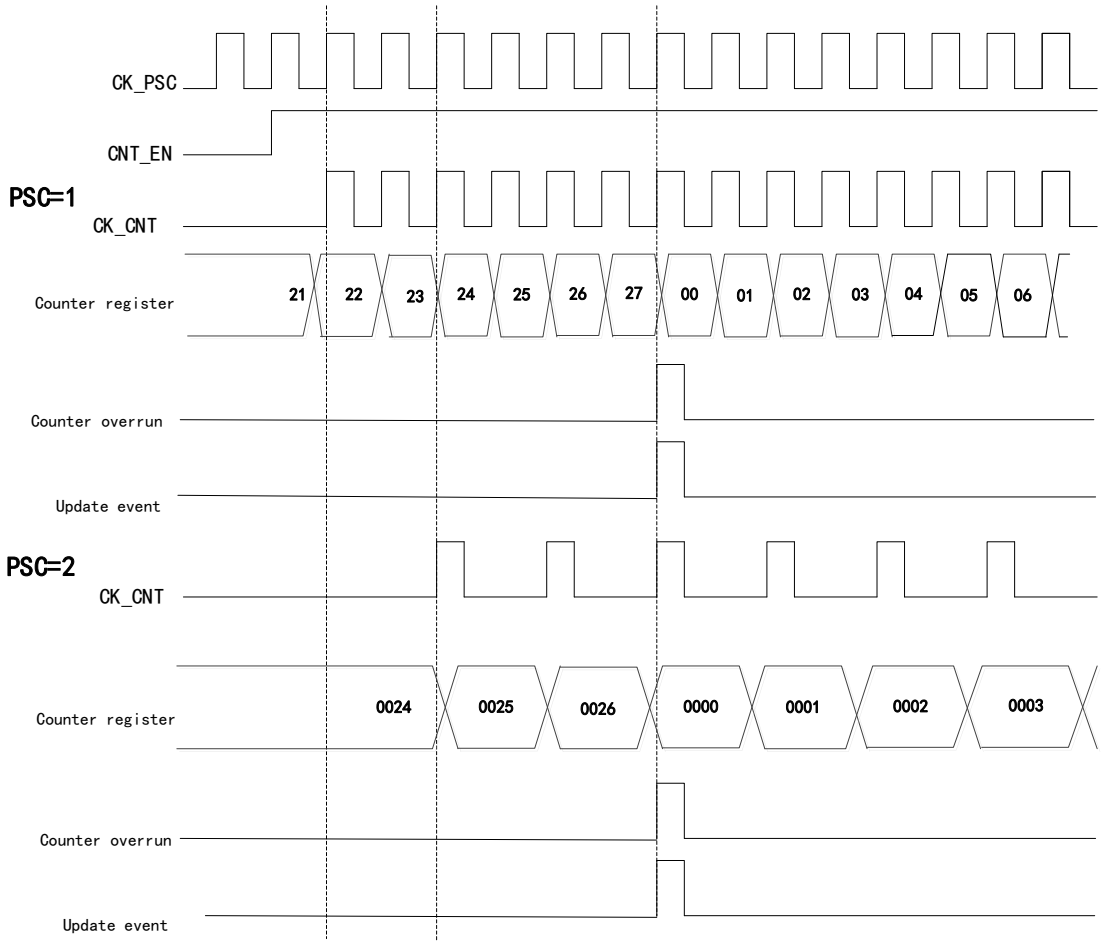
Count-up mode

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.

Figure 46 Timing Diagram of Count-up Mode when Division Factor is 1 or 2



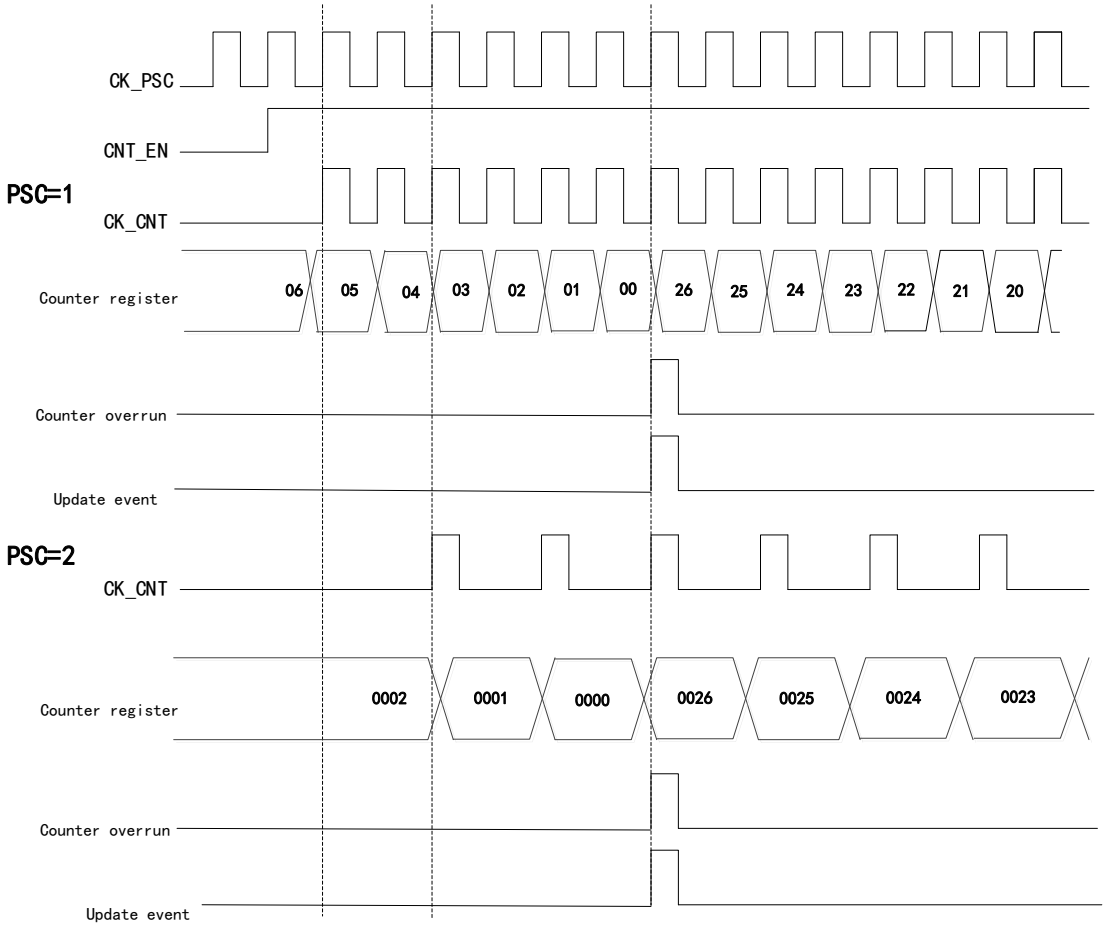
Count-down mode

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 47Timing Diagram of Count-down Mode when Division Factor is 1 or 2

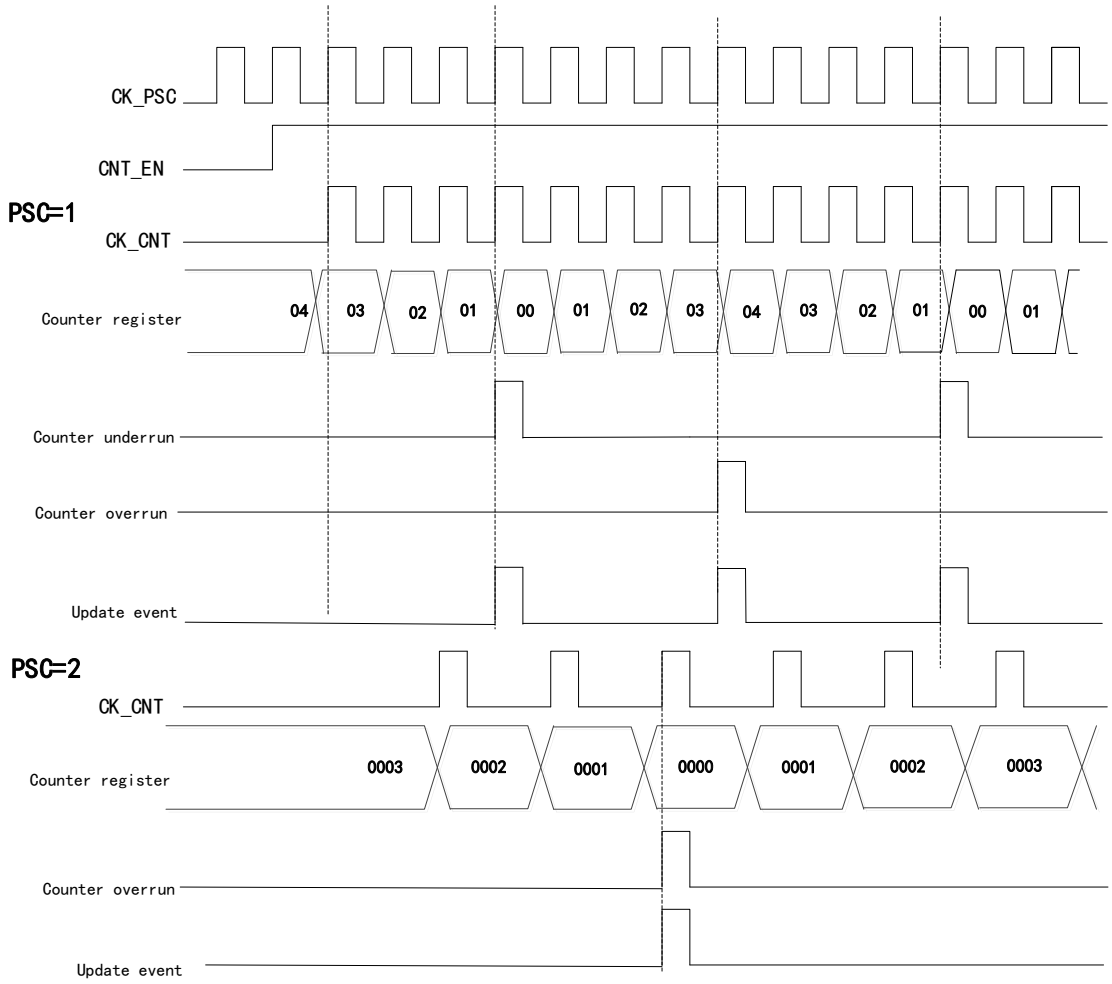


Center-aligned mode

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 48Timing 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 T1/2/3/4 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 when matching, channel x is invalid when matching, flip, force to invalid, force to valid, PWM mode 1 and PWM mode 2, which are configured by OCxMOD bit in TMR_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 edge 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 49Timing Diagram of PWM1 Count-up Mode

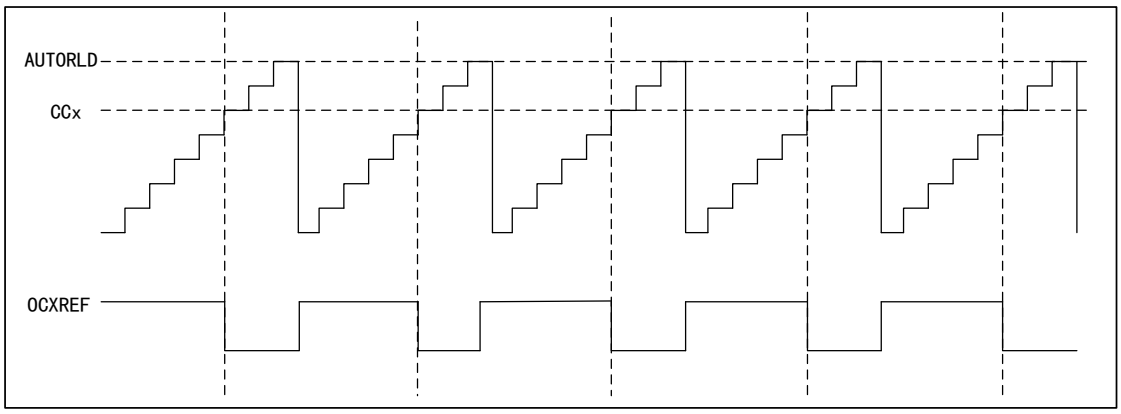


Figure 50Timing Diagram of PWM1 Count-down Mode

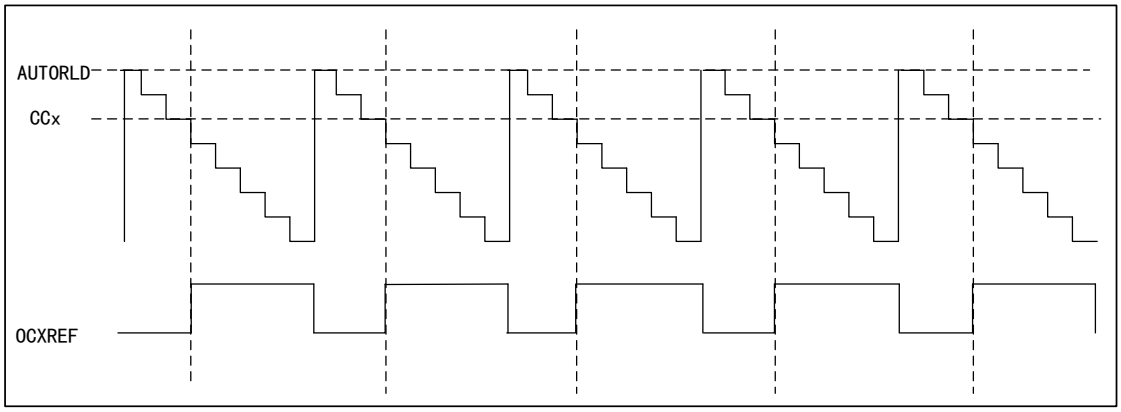
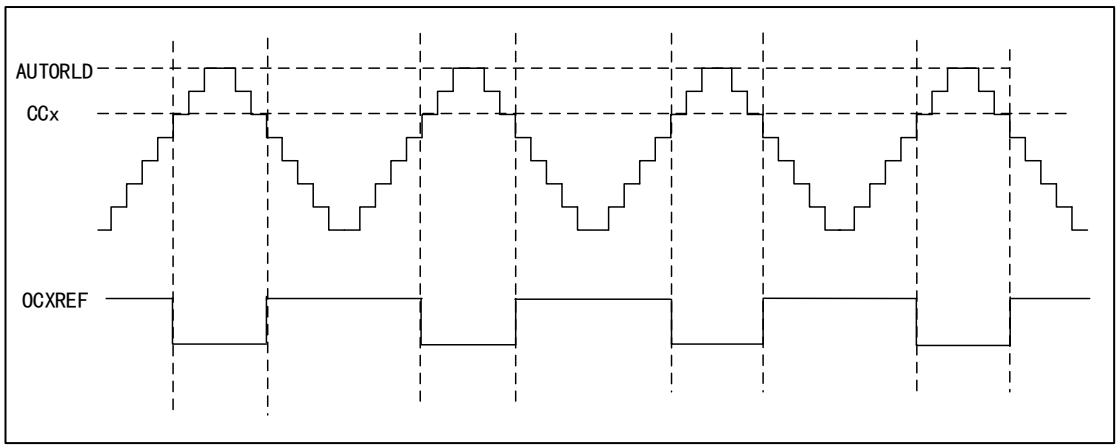


Figure 51Timing Diagram of PWM1 Center-aligned Mode



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 52 Timing Diagram of PWM2 Count-up Mode

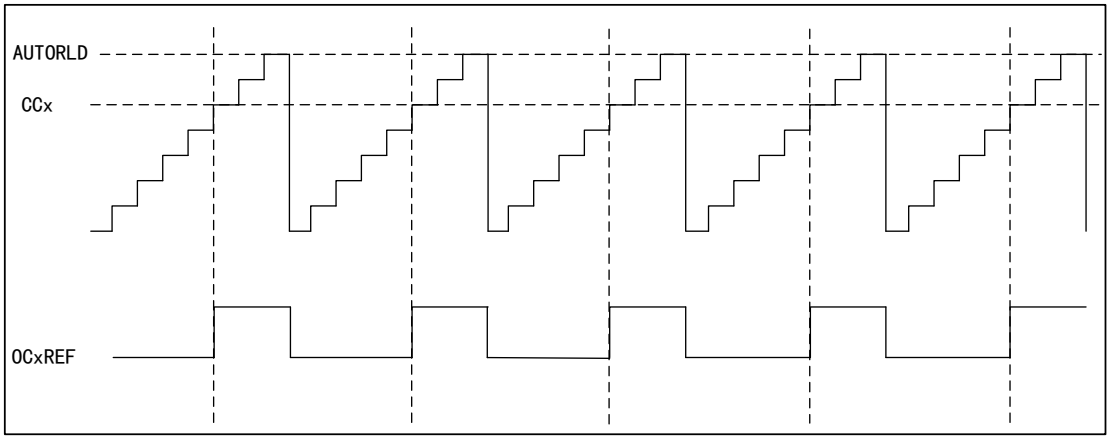


Figure 53 Timing Diagram of PWM2 Count-down Mode

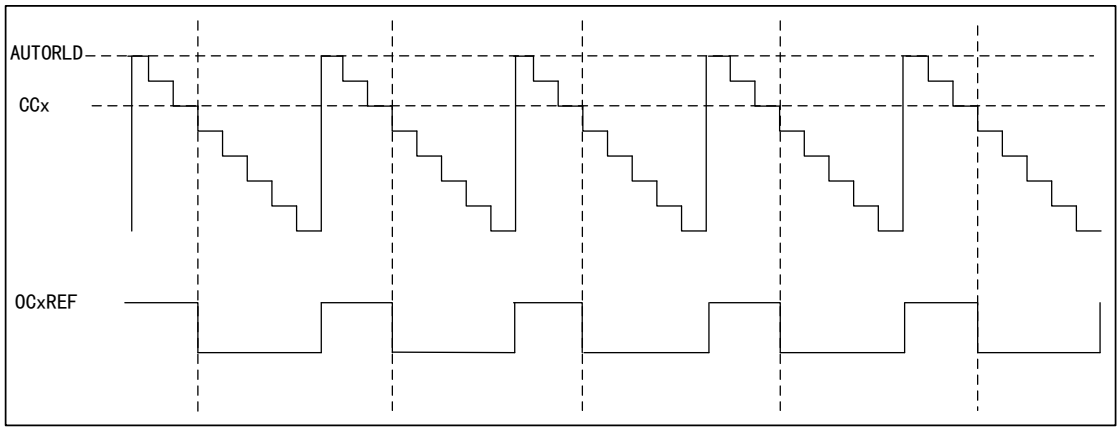
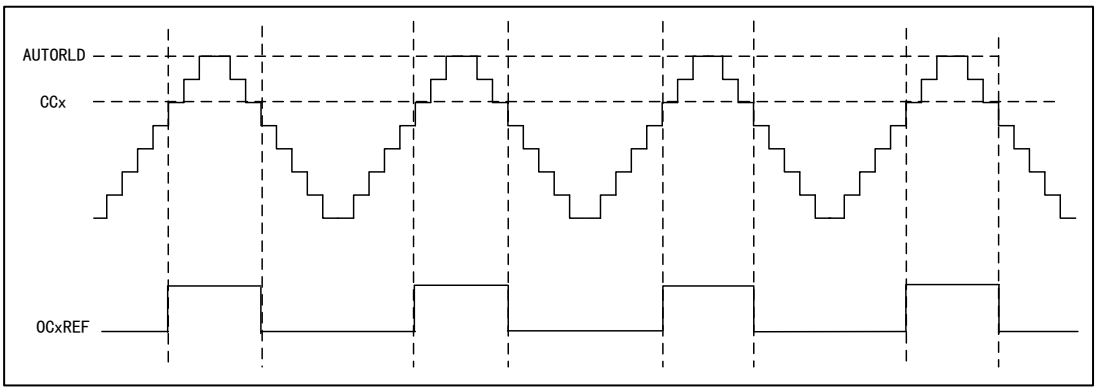


Figure 54 Timing Diagram of PWM2 Center-aligned Mode



15.4.6 PWM input mode

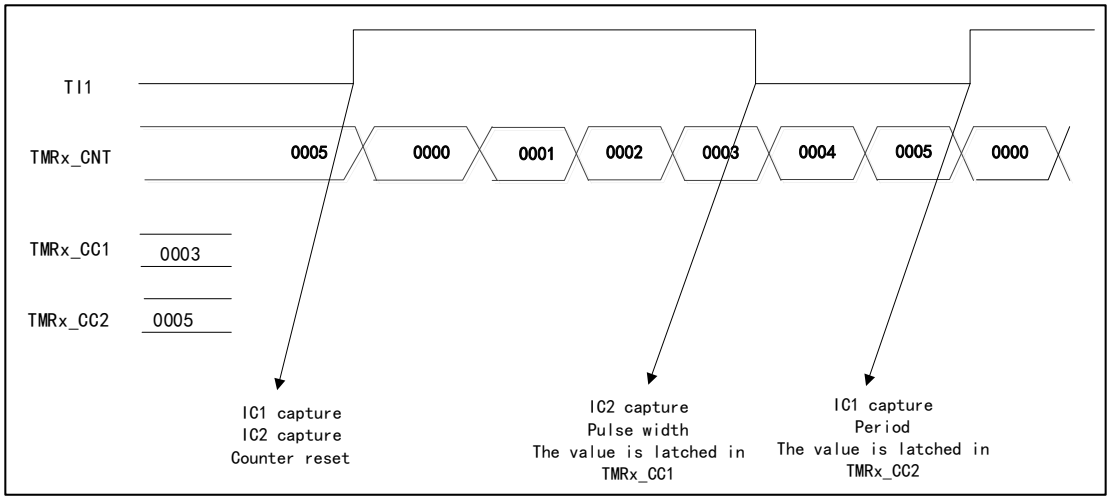
PWM input mode is a particular case of input capture.

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 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 55 Timing Diagram in PWM Input Mode



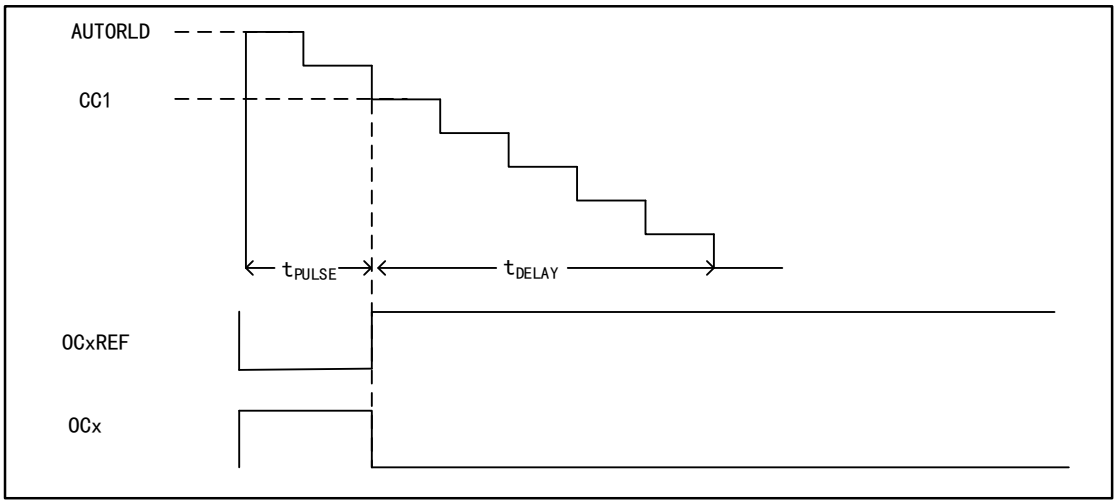
15.4.7 Single-pulse mode

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 56 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

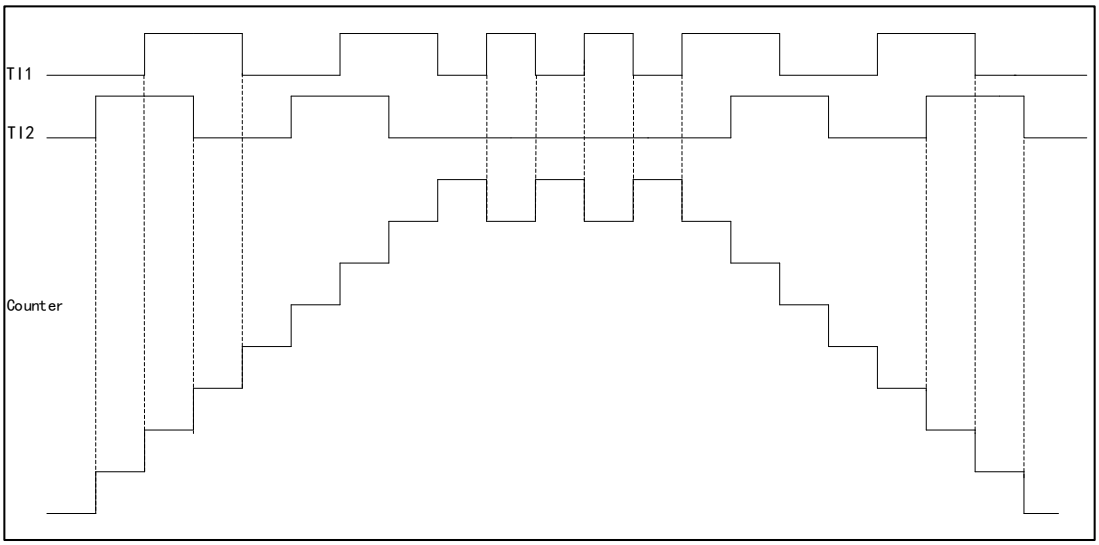
Effective edge		**Count only in TI1**		**Count only in TI2**		**Count in both TI1 and TI2**	
Level of relative signal		High	Low	High	Low	High	Low
TI1FP1	Rising edge	—		Count down	Count up	Count down	Count up
	Falling edge			Count up	Count down	Count up	Count down
TI2FP2	Rising edge	Count up	Count down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

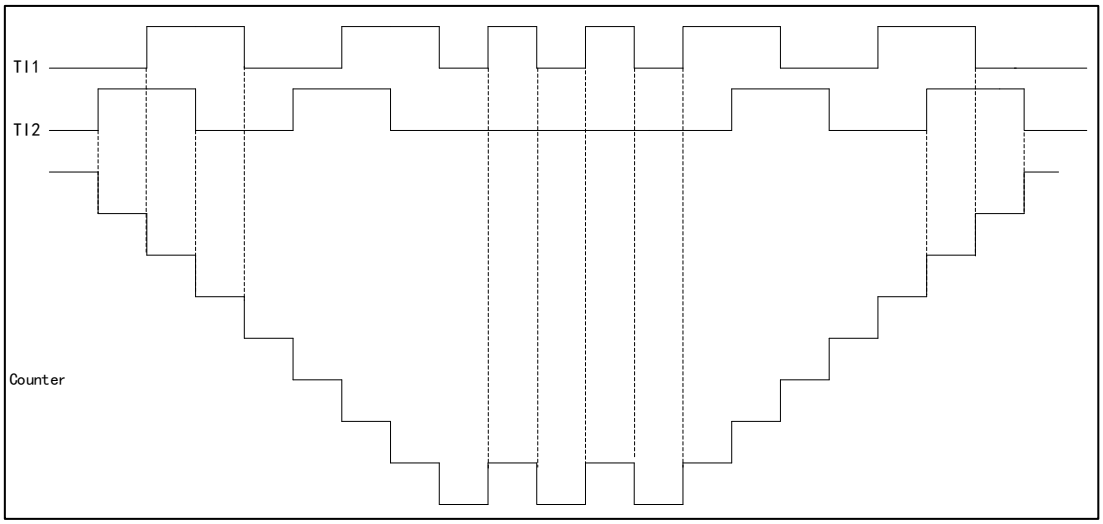
- TI1FP1 is mapped to TI1
- TI2FP2 is mapped to TI2
- Neither TI1FP1 nor TI2FP2 is phase-inverting
- The input signal is valid at the rising edge and falling edge
- Enable the counter

Figure 57 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 58 Example of Encoder Interface Mode of IC1FP1 Phase Inverting



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 after CNTEN bit of TMRx_CTRL1 register is set. 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 TMR1 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.

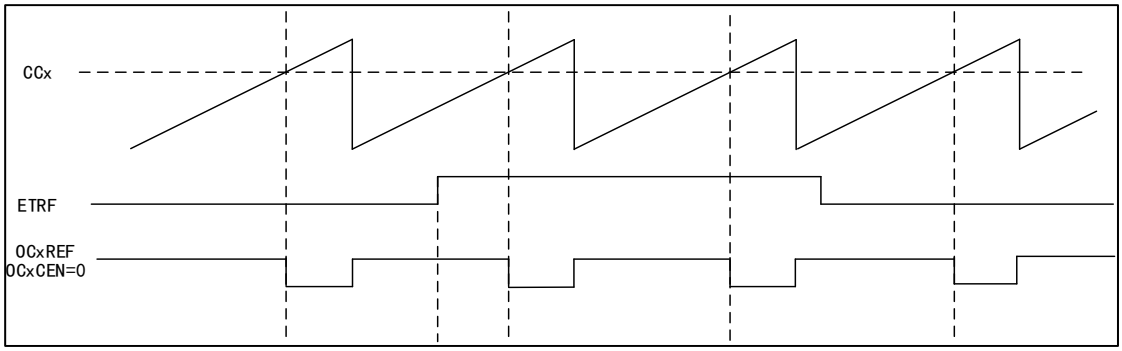
15.4.13 Clear OCxREF signal when an external event occurs

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.

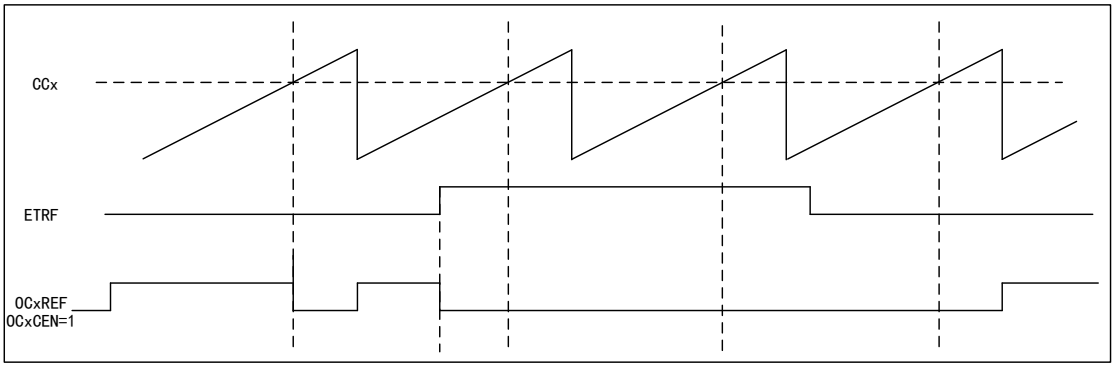
Set TMRx to PWM mode, disable the external trigger prescaler, and disable the external trigger mode 2; when ETRF input is high, set OCxCEN=0, and the output OCxREF signal is shown in the figure below.

Figure 59 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 60 OCxREF Timing Diagram



15.5 Register address mapping

In the following table, all registers of TMR2/3/4 are mapped to a 16-bit addressable (address) space.

Table 51 TMR2/3/4 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

Register name	Description	Offset address
TMRx_CCEN	Capture/Compare enable 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

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.
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
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

Field	Name	R/W	Description
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 be stopped, and the subsequent output level of the channel will no long be changed. 0: Disable 1: Enable
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
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)
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
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
15:10	Reserved		

15.6.2 Control register 2 (TMRx_CTRL2)

Offset address: 0x04

Reset value: 0x0000

Field	Name	R/W	Description
0	Reserved		

Field	Name	R/W	Description
1	MMSEL_3	R/W	Supplementary signal of MMSEL It is used to extend the signals used for TRGO in master mode of timer. 0: TRGO signal is determined by MMSEL<2:0> 1: TRGO signal is the encoder clock signal of the master mode timer, that is, when the counter jumps, TRGO outputs a cycle of high pulse. It is recommended to use it in encoder interface mode
2	Reserved		
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
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
7	TI1SEL	R/W	Timer Input 1 Select 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
10:8	MM2SEL	R/W	Master Mode 2 Signal Select） The signals of timers working in master mode can be used for TRGO2, which affects 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 TRGO2 001: Enable; the counter enable signal of master mode timer is used for TRGO2 010: Update; the update event of master mode timer is used for TRGO2 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 TRGO2

Field	Name	R/W	Description
			101: Compare mode 2; OC2REF is used to trigger TRGO2 110: Compare mode 3; OC3REF is used to trigger TRGO2 111: Compare mode 4; OC4REF is used to trigger TRGO2
15:11	Reserved		

15.6.3 Slave mode control register (TMRx_SMCTRL)

Offset address: 0x08

Reset value: 0x0000

Field	Name	R/W	Description
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; the slave mode timer starts the counter to work after receiving the TRGI high level signal; it stops the counter when receiving TRGI low level signal; when receiving 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. 111: External clock mode 1; select the rising edge signal of TRGI as the clock source to drive the counter to work.
3	OCCSEL	R/W	OCREF Clear Source Select This bit is used to select OCREF clear source 0:OCREF_CLR 1:ETRF
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)

Field	Name	R/W	Description
7	MSMEN	R/W	Master/slave Mode Enable 0: Invalid 1: Enable the master/slave mode
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.
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
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 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.
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

Table52 TMRx Internal Trigger Connection

Slave timer	ITR0（TS=000）	ITR1（TS=001）	ITR2（TS=010）	ITR3（TS=011）
TMR2	TMR1	TMR4	TMR3	-
TMR3	TMR1	TMR2	TMR4	-
TMR4	TMR1	TMR2	TMR3	-

15.6.4 DMA/Interrupt enable register (TMRx_DIEN)

Offset address: 0x0C

Reset value: 0x0000

Field	Name	R/W	Description
0	UIEN	R/W	Update Interrupt Enable 0: Disable 1: Enable
1	CC1IEN	R/W	Capture/Compare Channel1 Interrupt Enable 0: Disable 1: Enable
2	CC2IEN	R/W	Capture/Compare Channel2 Interrupt Enable 0: Disable 1: Enable
3	CC3IEN	R/W	Capture/Compare Channel3 Interrupt Enable 0: Disable 1: Enable
4	CC4IEN	R/W	Capture/Compare Channel4 Interrupt Enable 0: Disable 1: Enable
5	Reserved		
6	TRGIEN	R/W	Trigger Interrupt Enable 0: Disable 1: Enable
7	Reserved		
8	UDIEN	R/W	Update DMA Request Enable 0: Disable 1: Enable
9	CC1DEN	R/W	Capture/Compare Channel1 DMA Request Enable 0: Disable 1: Enable
10	CC2DEN	R/W	Capture/Compare Channel2 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
12	CC4DEN	R/W	Capture/Compare Channel4 DMA Request Enable 0: Disable 1: Enable
13	Reserved		
14	TRGDEN	R/W	Trigger DMA Request Enable 0: Disable 1: Enable
15	Reserved		

15.6.5 Status register (TMRx_STS)

Offset address: 0x10

Reset value: 0x0000

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.
1	CC1IFLG	RC_W0	capture/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, set 1 by hardware; clear 0 by software or clear 0 when reading TMRx_CC1 register.
2	CC2IFLG	RC_W0	capture/Compare Channel2 Interrupt Flag Refer to STS_CC1IFLG
3	CC3IFLG	RC_W0	capture/Compare Channel3 Interrupt Flag Refer to STS_CC1IFLG
4	CC4IFLG	RC_W0	capture/Compare Channel4 Interrupt Flag Refer to STS_CC1IFLG
5	Reserved		

Field	Name	R/W	Description
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.
8:7	Reserved		
9	CC1RCFLG	RC_W0	capture/Compare Channel1 Repetition Capture Flag 0: Repeated capture does not occur 1: Repeated capture occurs The value of the counter is captured to TMRx_CCR1 register, and CC1IFLG=1; this bit is set to 1 by hardware and cleared to 0 by software only when the channel is configured as input capture.
10	CC2RCFLG	RC_W0	capture/compare Channel2 Repetition Capture Flag Refer to STS_CC1RCFLG
11	CC3RCFLG	RC_W0	capture/compare Channel3 Repetition Capture Flag Refer to STS_CC1RCFLG
12	CC4RCFLG	RC_W0	capture/compare Channel4 Repetition Capture Flag Refer to STS_CC1RCFLG
15:13	Reserved		

15.6.6 Control event generation register (TMRx_CEG)

Offset address: 0x14

Reset value: 0x0000

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 reads the value of TMRx_AUTORLD; in center-aligned mode or count-up mode, the counter will be cleared to 0.
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
2	CC2EG	W	Capture/Compare Channel2 Event Generation Refer to CC1EG description
3	CC3EG	W	Capture/Compare Channel3 Event Generation Refer to CC1EG description
4	CC4EG	W	Capture/Compare Channel4 Event Generation Refer to CC1EG description
5	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.
15:8	Reserved		

15.6.7 Capture/Compare mode register 1 (TMRx_CCM1)

Offset address: 0x18

Reset value: 0x0000

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 OCxx in the register describes the function of the channel in the output mode, and the ICxx in the register describes the function of the channel in the input mode.

Output compare mode:

Field	Name	R/W	Description
1:0	CC1SEL	R/W	Capture/Compare Channel 1 Select 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).
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.
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

Field	Name	R/W	Description
			uncertain, PWM mode can be used only in single-pulse mode (SPMEN=1); otherwise, the following output compare result is uncertain.
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 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.
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
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).
10	OC2FEN	R/W	Output Compare Channel2 Preload Enable
11	OC2PEN	R/W	Output Compare Channel2 Buffer Enable
14:12	OC2MOD	R/W	Output Compare Channel1 Mode
15	OC2CEN	R/W	Output Compare Channel2 Clear Enable

Input capture mode:

Field	Name	R/W	Description
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

Field	Name	R/W	Description
			Note: This bit can be written only when the channel is closed (TMRx_CCEN bit CC1EN=0).
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.
7:4	IC1F	R/W	Input Capture Channel1 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, indicating that a jump is generated by every N events.
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 TI1 10: CC2 channel is input, and IC2 is mapped on TI2 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).
11:10	IC2PSC	R/W	Input Capture Channel2 Perscaler Configure
15:12	IC2F	R/W	Input Capture Channel2 Filter Configure

15.6.8 Capture/Compare mode register 2 （TMRx_CCM2）

Offset address: 0x1C

Reset value: 0x0000

Refer to the description of the above CCM1 register.

Output compare mode:

Field	Name	R/W	Description
1:0	CC3SEL	R/W	Capture/Compare Channel 1 Select 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).
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.
3	OC3PEN	R/W	Output Compare Channel3 Preload Enable
6:4	OC3MOD	R/W	Output Compare Channel3 Mode Configure)
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
9:8	CC4SEL	R/W	Capture/Compare Channel 4 Select 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).
10	OC4FEN	R/W	Output Compare Channel4 Preload Enable
11	OC4PEN	R/W	Output Compare Channel4 Buffer Enable
14:12	OC4MOD	R/W	Output Compare Channel4 Mode Configure
15	OC4CEN	R/W	Output Compare Channel4 Clear Enable

Input capture mode:

Field	Name	R/W	Description
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).

Field	Name	R/W	Description
3:2	IC3PSC	R/W	Input Capture Channel3 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.
7:4	IC3F	R/W	Input Capture Channel3 Perscaler Configure
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).
11:10	IC4PSC	R/W	Input Capture Channel4 Perscaler Configure
15:12	IC4F	R/W	Input Capture Channel4 Filter Configure

15.6.9 Capture/Compare enable register (TMRx_CCEN)

Offset address: 0x20

Reset value: 0x0000

Field	Name	R/W	Description
0	CC1EN	R/W	Capture/Compare Channel1 Output Enable When CC1 is configured as output: 0: Disable output 1: Enable output When CC1 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
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: CC1POL and CC1NPOL control the polarity of the triggered or captured signals TI1FP1 and TI2FP1 at the same time 00: Non-phase-inverting/rising edge: TIxFP1 is not phase-inverting (triggered in gated and encoder mode), and is captured at the rising edge of TIxFP1 (reset trigger, capture, external clock and trigger mode). 01: Phase inverting/falling edge: TIxFP1 is phase-inverting (triggered in gated and encoder mode), and is captured at the rising edge of TIxFP1 (reset trigger, capture, external clock and trigger mode).

Field	Name	R/W	Description
			10: Reserved 11: Non-phase-inverting/Rising and falling edges: TIxFP1 is not phase-inverting (triggered in gated mode, cannot be used in encoder mode), and is captured at the rising edge of TIxFP1 (reset trigger, capture, external clock and trigger mode).
2	Reserved		
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.
4	CC2EN	R/W	Capture/Compare Channel2 Output Enable Refer to CCEN_CC1EN
5	CC2POL	R/W	Capture/Compare Channel2 Output Polarity Configure Refer to CCEN_CC1POL
6	Reserved		
7	CC2NPOL	R/W	Capture/Compare Channel2 Output Polarity Configure Refer to CCEN_CC1NPOL
8	CC3EN	R/W	Capture/Compare Channel3 Output Enable Refer to CCEN_CC1EN
9	CC3POL	R/W	Capture/Compare Channel3 Output Polarity Configure Refer to CCEN_CC1POL
10	Reserved		
11	CC3NPOL	R/W	Capture/Compare Channel3 Output Polarity Configure Refer to CCEN_CC1NPOL
12	CC4EN	R/W	Capture/Compare Channel4 Output Enable Refer to CCEN_CC1EN
13	CC4POL	R/W	Capture/Compare Channel4 Output Polarity Configure Refer to CCEN_CC1POL
14	Reserved		
15	CC4NPOL	R/W	Capture/Compare Channel4 Output Polarity Configure Refer to CCEN_CC1NPOL

Table53 Output Control Bit of Standard OCx Channel

CCxEN bit	OCx output state
0	Disable output ((OCx=0, OCx_EN=0)
1	OCx=OCxREF+ polarity, OCx_EN=1

Note: The state of external I/O pin connected to the standard OCx channel depends on the state of the OCx channel and the GPIO and AFIO registers.

15.6.10 Counter register (TMRx_CNT)

Offset address: 0x24

Reset value: 0x0000

Field	Name	R/W	Description
15:0	CNT	R/W	Counter Value
31:16	CNT	R/W	Counter Value (only TMR2)

15.6.11 Prescaler register (TMRx_PSC)

Offset address: 0x28

Reset value: 0x0000

Field	Name	R/W	Description
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: 0xFFFF FFFF

Field	Name	R/W	Description
15:0	AUTORLD	R/W	Auto Reload Value When the value of auto reload is empty, the counter will not count.
31:16	AUTORLD	R/W	Auto Reload Value (only TMR2)

15.6.13 Channel 1 capture/compare register (TMRx_CC1)

Offset address: 0x34

Reset value: 0x0000

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 comparison result when an update event is generated.
31:16	CC1	R/W	Capture/Compare Channel 1 Value

15.6.14 Channel 2 capture/compare register (TMRx_CC2)

TMR2 Channel 2 capture/compare register (TMRx_CC2)

Offset address: 0x38

Reset value: 0x0000

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

TMR3/4 Channel 2 capture/compare register (TMRx_CC2)

Offset address: 0x38

Reset value: 0x0000

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

15.6.15 Channel 3 capture/compare register (TMRx_CC3)

TMR2 Channel 3 capture/compare register (TMRx_CC3)

Offset address: 0x3C

Reset value: 0x0000

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

TMR3/4 Channel 3 capture/compare register (TMRx_CC3)

Offset address: 0x3C

Reset value: 0x0000

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

15.6.16 Channel 4 capture/compare register (TMRx_CC4)

TMR2 Channel 4 capture/compare register (TMRx_CC4)

Offset address: 0x40

Reset value: 0x0000

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

TMR3/4 Channel 4 capture/compare register (TMRx_CC4)

Offset address: 0x40

Reset value: 0x0000

Field	Name	R/W	Description
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

Field	Name	R/W	Description
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 …….
7:5	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.
15:13	Reserved		

15.6.18 DMA address register of continuous mode (TMRx_DMADDR)

Offset address: 0x4C

Reset value: 0x0000

Field	Name	R/W	Description
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;

Field	Name	R/W	Description
			"DMA index" is the offset automatically controlled by DMA, and it depends on DBLEN defined in TMRx_DCTRL register.

16 Basic timer (TMR6/7)

16.1 Introduction

The basic timers TMR6/TMR7 have an unsigned 16-bit counter, auto reload register, prescaler and trigger controller.

The basic timer provides time reference for general-purpose timer, and can generate DMA request by configuration.

16.2 Main characteristics

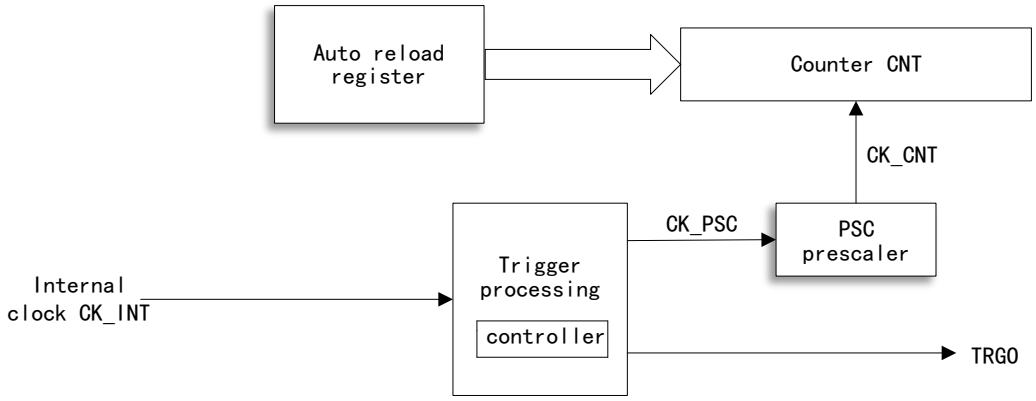
（1） Counter: 16-bit counter, which can only count up

（2） Prescaler: 16-bit programmable prescaler

（3） Clock source: There is only internal clock

16.3 Structure block diagram

Figure 61 Basic Timer Structure Block Diagram



16.4 Functional description

16.4.1 Clock source selection

The basic timer is driven by internal clock source TMRx_CLK

Configure the CNTEN bit of TMRx_CTRL1 register to enable the counter; when CNTEN bit is set, the internal clock CK_INT can generate CK_INT to drive the counter through the controller and prescaler.

16.4.2 Timebase unit

The time base unit in the basic timer contains three registers:

- Counter register (CNT) 16 bits
- Autoreload register (AUTORLD) 16 bits

- Prescaler (PSC) 16 bits

Counter CNT

The basic timer only has one count mode: count-up

Count-up mode

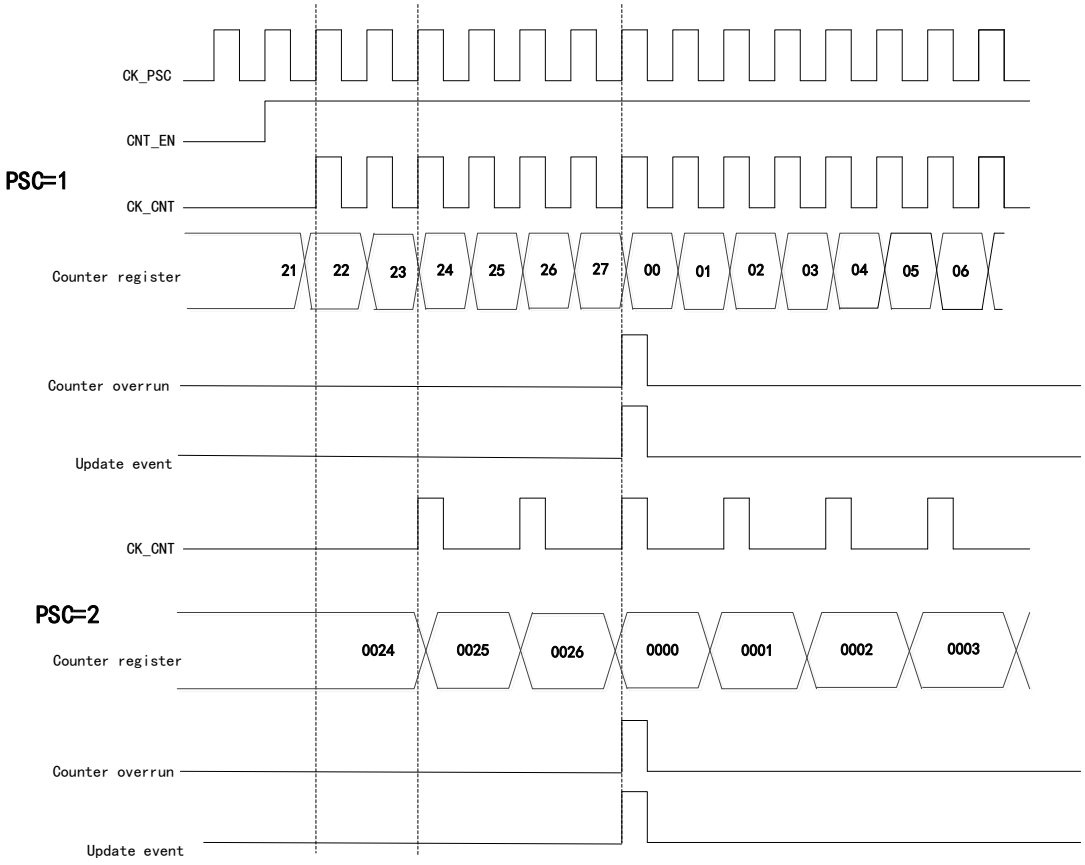
When the counter is in count-up mode, the counter will count up from 0; every time a pulse is generated, the counter will increase by 1 and when the value of the counter (TMRx_CNT) is equal to the value of the auto reload (TMRx_AUTORLD), then the counter will start to count again from 0, a count-up overrun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

Disable the update event and set UD bit of TMRx_CTRL1 register to 1.

Generate the update interrupt or DMA request and set URSSEL bit in TMRx_CTRL1 register.

When an update event occurs, both the auto reload register and the prescaler register will be updated.

Figure 62 Timer Timing Diagram with Internal Clock Division Factor of 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.

16.5 Register address mapping

In the following table, all registers of TMR6/TMR7 are mapped to a 16-bit addressable (address) space.

Table 54 TMR6 and TMR7 Register Address Mapping

Register name	Description	Offset address
TMRx_CTRL1	Control register 1	0x00
TMRx_CTRL2	Control register 2	0x04
TMRx_DIEN	DMA/Interrupt enable register	0x0C
TMRx_STS	Status register	0x10
TMRx_CEG	Control event generation register	0x14
TMRx_CNT	Counter register	0x24
TMRx_PSC	Prescaler register	0x28
TMRx_AUTORLD	Auto reload register	0x2C

16.6 Register functional description

16.6.1 Control register 1 (TMRx_CTRL1)

Offset address: 0x00

Reset value: 0x0000

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.
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
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
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 be stopped, and the subsequent output level of the channel will no long be changed. 0: Disable 1: Enable
6:4	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
15:8	Reserved		

16.6.2 Control register 2 (TMRx_CTRL2)

Offset address: 0x04

Reset value: 0x0000

Field	Name	R/W	Description
3:0	Reserved		
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
15:7	Reserved		

16.6.3 DMA/Interrupt enable register (TMRx_DIEN)

Offset address: 0x0C

Reset value: 0x0000

Field	Name	R/W	Description
0	UIEN	R/W	Update Interrupt Enable 0: Disable 1: Enable
7:1	Reserved		
8	UDIEN	R/W	Update DMA Request Enable 0: Disable 1: Enable
15:9	Reserved		

16.6.4 Status register (TMRx_STS)

Offset address: 0x10

Reset value: 0x0000

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.
15:1	Reserved		

16.6.5 Control event generation register (TMRx_CEG)

Offset address: 0x14

Reset value: 0x0000

Field	Name	R/W	Description
0	UEG	W	Update Event Generate 0: Invalid 1: Initialize the counter and generate the 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 reads the value of TMRx_AUTORLD; in center-aligned mode or count-up mode, the counter will be cleared to 0.
15:1	Reserved		

Note: The state of external I/O pin connected to the standard OCx channel depends on the state of the OCx channel and the GPIO and AFIO registers.

16.6.6 Counter register (TMRx_CNT)

Offset address: 0x24

Reset value: 0x0000

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

16.6.7 Prescaler register (TMRx_PSC)

Offset address: 0x28

Reset value: 0x0000

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

16.6.8 Auto reload register (TMRx_AUTORLD)

Offset address: 0x2C

Reset value: 0xFFFF

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

17 Infrared timer (IRTMR)

17.1 Introduction

IRTMR is an infrared interface for remote control, which can use an infrared LED to realize remote control function.

17.2 Functional description

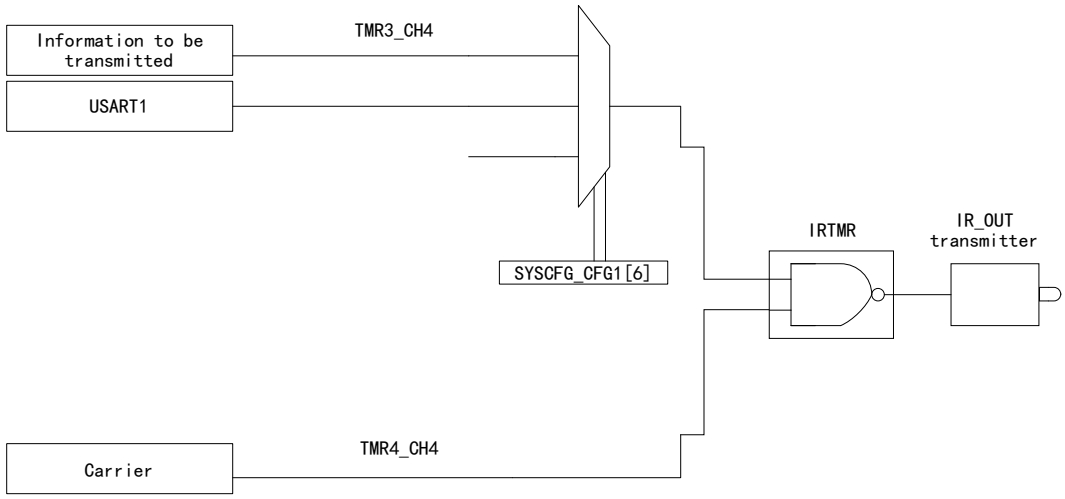
17.2.1 IRTMR receive

The infrared receiver can be connected to the GPIO of the controller or the input capture channel of the timer through the output of the external IR receiver module to realize data receiving.

17.2.2 IRTMR transmit

IRTMR is internally connected to TMR3 and TMR4, and the specific block diagram is as follows:

Figure 63 IRTMR Structure Block Diagram



In order to generate correct infrared remote control signal, TMR4_CH4 should be configured correctly to provide a high-frequency carrier signal, while TMR3_CH4 provides the information we transmit or by configuring IRSEL bit of SYSCFG_CFG1 register, selects USART1 to generate modulation envelope.

The final modulation signal is output through IR_OUT pin, and this function is activated by enabling the related multiplexing functions in GPIOx_ALFx register.

Configure I2CPB9FMP in SYSCFG_CFG1 register when controlling infrared LED driven by high current, configure I2CPB9FMP in SYSCFG_CFG1 register to enable the fast mode plus of PB9 to support high-current output. (This function is used on PB9 only)

18 Watchdog timer (WDT)

18.1 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 when the counter decreases to 0, and when the value of the counter is greater than the window value, it will be reset if it is reloaded.

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.

18.2 Hardware watchdog

18.2.1 Introduction

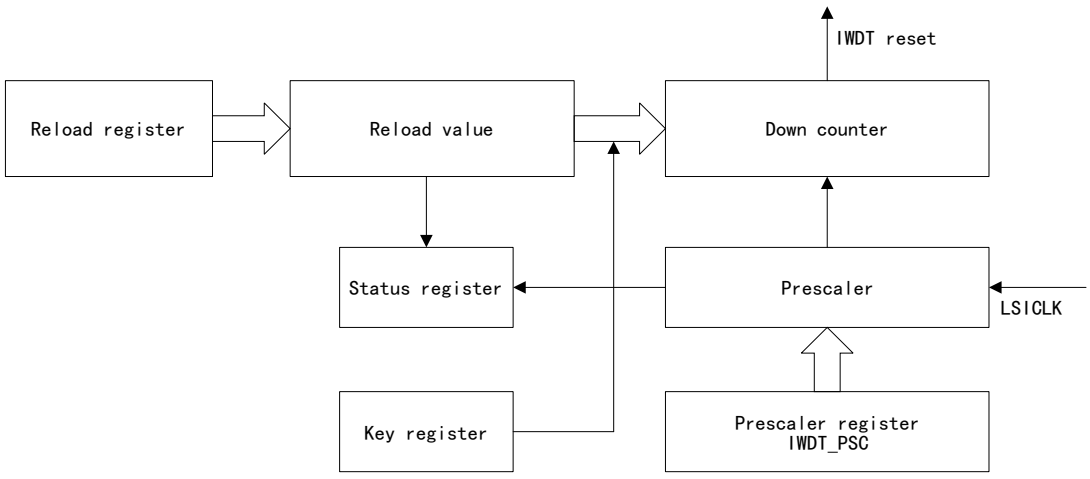
The independent watchdog consists of an 8-bit prescaler IWDT_PSC, 12-bit count-down counter, 12-bit reload register IWDT_CNTRLD, key register IWDT_KEY, status register IWDT_STS and window register IWDT_WIN.

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.

18.2.2 Structure block diagram

Figure 64Independent Watchdog Structure Block Diagram



Note: The prescaler, reload value and count-down counter are in V_{DD} power supply area; the prescaler register, status register, reload register and keyword register are in 1.5V power supply area. The watchdog function is in the V_{DD} power supply area and it can work normally in the stop or standby mode.

18.2.3 Functional description

18.2.3.1 Key register

Write 0xCCCC in the key register to enable the independent watchdog, then the counter starts to count down from the reset value 0xFFF 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 in the key register to rewrite the value of the prescaler register, reload register and window value register.

18.2.3.2 Window register

The default value of window register IWDT_WIN is 0xFFF. Without update, the window option is disabled. When the window value is changed, the reload operation will be performed, and the value of watchdog counter will be set to the value of IWDT_CNTRLD, which can delay the event cycle needed for reset.

The independent watchdog can work in the window watchdog mode, and the value of window register IWDT_WIN needs to be set appropriately.

18.2.3.3 Configuration IWDT

Configure IWDT when window register is used

- Enable IWDT (write 0xCCCC to the key register IWDT_KEY)
- Enable the register access permission (write 0x5555 to the key register IWDT_KEY)
- Configure IWDT_PSC prescaler register (write the value within 0~7 to IWDT_PSC)
- Wait until the value of status register IWDT_STS is updated to 0x00
- Configure the window register IWDT_WIN (the value of auto reload register IWDT_CNTRLD can be updated to the value of watchdog register)

Note: When the value of status register IWDT_STS is 0x00, the window value will be written to refresh the counter by the value of auto reload

Configure IWDT when window register is disabled

- Enable IWDT (write 0xCCCC to the key register IWDT_KEY)
- Enable the register access permission (write 0x5555 to the key register IWDT_KEY)
- Configure IWDT_PSC prescaler register (write the value within 0~7 to IWDT_PSC)

- Configure reload register IWDT_CNTRLD
- Wait until the value of status register IWDT_STS is updated to 0x00
- Referesh the watchdog counter using IWDT_CNTRLD register

18.2.3.4 Register access protection

The prescaler register IWDT_PSC, reload register IWDT_CNTRLD and window register IWDT_WIN have the function of write protection. If you want to rewrite these three registers, you need to write 0X5555 in the key register. If you write other value in the key register, the protection of the register will be started again.

Write 0xAAAA to the key register and the write protection function will also be enabled.

The prescaler register, reload register and window register can be observed through the status register.

18.2.3.5 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.

18.2.3.6 Debug mode

The independent watchdog can be configured in debug mode and choose to stop or continue to work. It depends on the IWDT_STS bit of DBGMCU_APB1F register in DBGMCU module.

18.3 Window watchdog

18.3.1 Introduction

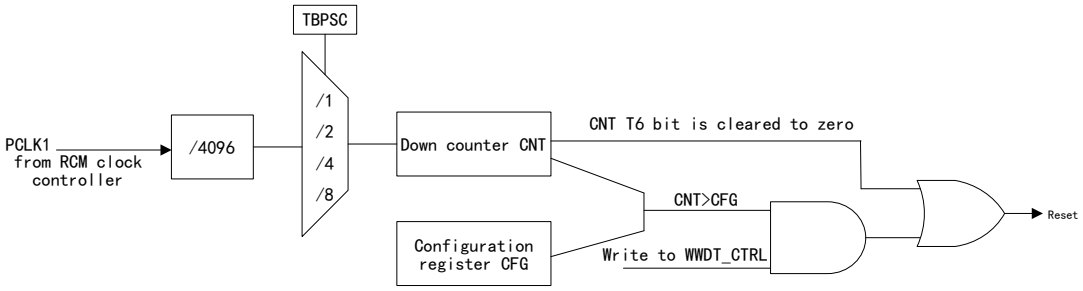
The window watchdog contains a 7-bit free-running down counter, prescaler and control register WWDT_CTRL, configuration register WWDT_CFG and status register WWDT_STS.

The window watchdog clock comes from PCLK, 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.

18.3.2 Structure block diagram

Figure 65 Window Watchdog Structure Block Diagram



18.3.3 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.

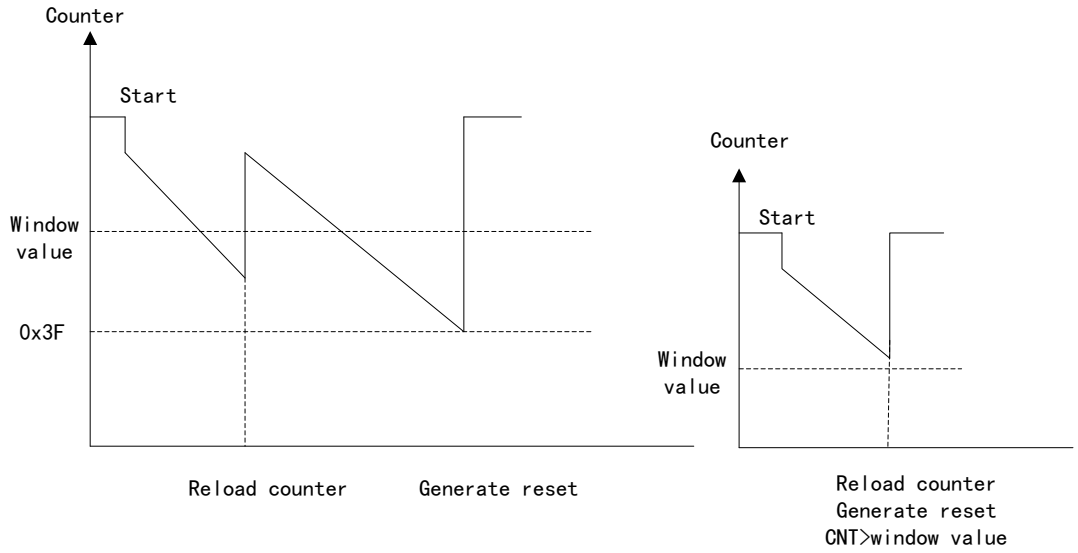
After reset, the watchdog is always closed and it can be enabled only by setting the WWDTEN bit of WWDT_CTRL register.

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 configuration register can enable the early wake-up interrupt. When the count reaches 0x40, an interrupt will be generated. Entering the interrupt service program (ISTS) can be used 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, and the value of the window register is 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 66 Window Watchdog Timing Diagram



The calculation formula of window watchdog timer timeout is as follows:

$$T_{WWDT}=T_{PCLK1}\times 2^{WTB}\times（T[5:0]+1）$$

Where:

- T_{WWDT}: WWDT timeout period
- T_{PCLK1}: Clock cycle of APB in ms

Table 55 Minimum/Maximum timeout value when PCLK1=36MHZ

WTB	Minimum timeout value	Maximum timeout value
0	113μs	7.28ms
1	227μs	14.56ms
2	455μs	29.12ms
3	910μs	58.25ms

18.3.3.1 Debug mode

The window watchdog can be configured in debug mode and choose to stop or continue to work. It depends on the WWDT_STS bit of DBGMCU_APB1F register in DBGMCU module.

18.4 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

Register name	Description	Offset address
IWDT_STS	Status register	0x0C
IWDT_WIN	Window register	0x10

18.5 IWDT register functional description

These peripheral registers can be operated by half word (16 bits) or word (32 bits).

18.5.1 Key register (IWDT_KEY)

Offset address: 0x00

Reset value: 0x0000 0000 (reset in standby mode)

Field	Name	R/W	Description
15:0	KEY	W	Allow Access IWDT Register Key Value Writing 0x5555 means enabled access to IWDT_PSC, IWDT_CNTRLD and IWDT_WIN 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 to enable the watchdog (the hardware watchdog is unrestricted by this command word). The read-out value is 0x0000.
31:16	Reserved		

18.5.2 Prescaler register (IWDT_PSC)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
2:0	PSC	R/W	Prescaler Factor Configure Support write protection function; when writing 0x5555 in the IWDT_KEY register, it is allowed to access the register; in the process of writing 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
31:3	Reserved		

18.5.3 Counter reload register (IWDT_CNTRLD)

Offset address: 0x08

Reset value: 0x0000 0FFF(reset in standby mode)

Field	Name	R/W	Description
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.
31:12	Reserved		

18.5.4 Status register (IWDT_STS)

Offset address: 0x0C

Reset value: 0x0000 0000 (not reset in standby mode)

Field	Name	R/W	Description
0	PSCUFLG	R	Watchdog Prescaler Value Update Flag When the prescaler factor is updated, set 1 by hardware; after the prescaler factor is updated, clear 0 by hardware; the prescaler factor is updated only when the PSCUFLG bit is cleared to 0.
1	CNTUFLG	R	Watchdog Counter Reload Value Update Flag When the counter reload value is updated, set 1 by hardware; after the counter reload value is updated, clear 0 by hardware; the counter reload value is updated only when the CNTUFLG bit is cleared to 0.
2	WINUFLG	R	Watchdog Counter Window Value Update Flag When the window value is updated, set 1 by hardware; after the window value of the counter is updated, clear 0 by hardware; the window value is valid only when the IWDT_WIN register is enabled.
31:3	Reserved		

18.5.5 Window register (IWDT_WIN)

Offset address: 0x10

Reset value: 0x0000 0FFF(reset in standby mode)

Field	Name	R/W	Description
11:0	WIN	R/W	Watchdog Counter Window Value These bits include the window value and the initial value of down counter These bits can be modified only when STS_WINUFLG=0 Reloading the counter between the counter value and the window value can prevent resetting Note: When reading this register, the value of V_{DD} power supply domain will be returned, so if you want to read data, you should ensure STS_WINUFLG=0.
31:12	Reserved		

Note: When the reload setting, prescaler setting and window value resetting are running, if you want to change the reload value, prescaler value and window value, you need to confirm that the relevant flag bits are 0. There is no need to wait after the update, unless you want to enter the low-power mode.

18.6 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

18.7 WWDT register functional description

These peripheral registers can be operated by half word (16 bits) or word (32 bits).

18.7.1 Control register (WWDT_CTRL)

Offset address: 0x00

Reset value: 0x0000 007F

Field	Name	R/W	Description
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.
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
31:8	Reserved		

18.7.2 Configuration register (WWDT_CFG)

Offset address: 0x04

Reset value: 0x0000 007F

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.
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
9	EWIEN	R/S	Early Wakeup Interrupt Enable 0: Meaningless 1: When the counter value reaches 0x40, an interrupt will be generated; this interrupt is cleared by hardware after reset.
31:10	Reserved		

18.7.3 Status register (WWDT_STS)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	EWIFLG	RC_W0	Early Wakeup Interrupt Occur Flag 0: Not occur 1: When the counter value reaches 0x40, set 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 Writing 1 to this bit is invalid.
31:1	Reserved		

19 Real-time clock (RTC)

19.1 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
Prescaler	PSC

19.2 Introduction

It has sub-second, time and date registers with BCD coding, as well as corresponding alarm registers, and can realize timestamp function together with external pins. It supports clock calibration function and time compensation.

19.3 Main characteristics

（1） Timebase unit

（2） Clock calibration

（3） Subsecond, time and date

（4） Time error compensation

（5） Alarm (subsecond, time and date mask)

（6） Timestamp

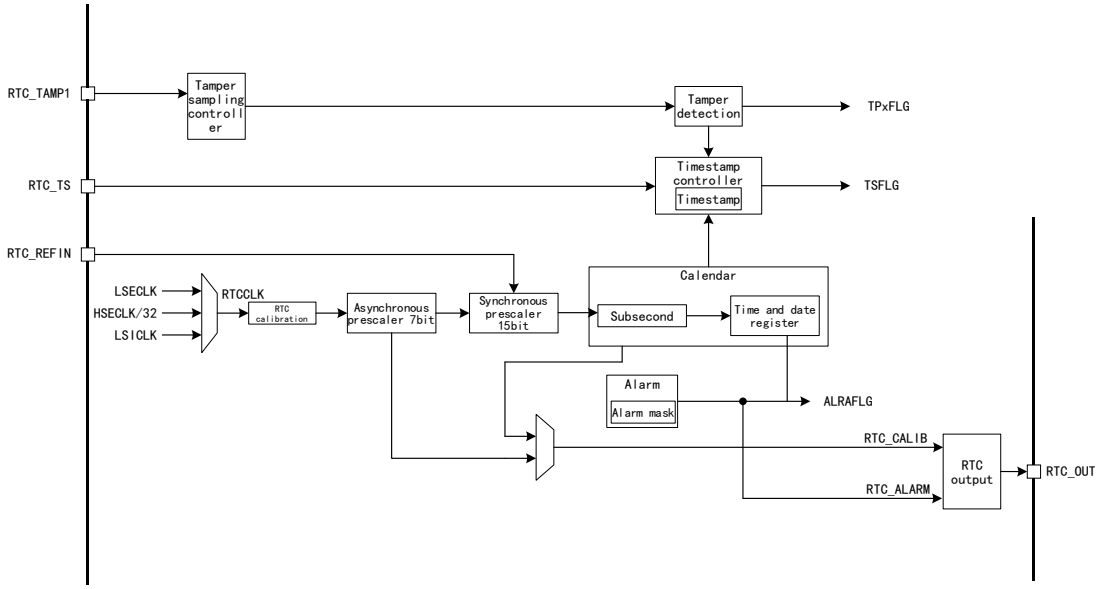
（7） Tamper detection

（8） 3 kinds of RTC outputs

（9） Multiple interrupt control

19.4 Structure block diagram

Figure 67 RTC Structure Block Diagram



Note:

1) Multiplexing function output: RTC_OUT is output in one of the following two forms
   - RTC_CALIB: This output is enabled through CALOEN bit of RTC_CTRL register, and when the frequency of LSECLK is 32.768kHz, the clock output is 512Hz or 1Hz.
   - RTC_ALARM: This output, alarm clock A, is enabled by OUTSEL bit of RTC_CTRL register.
2) Multiplexing function input:
   - RTC_TS: Timestamp event
   - RTC_TAMP1: Tamper event detection 1
   - RTC_REFIN: 50 or 60Hz reference clock input

19.5 Functional description

19.5.1 I/O pin controlled by RTC

RTC_OUT, RTC_TS and RTC_TAMP1 in RTC can be mapped to the same pin (PC13).

The output selection of RTC_ALARM is configured by RTC_TACFG, and PC13VAL bit of RTC_TACFG register is used to select RTC_ALARM to configure as push-pull output or open-drain output.

When PC13 pin does not use RTC multiplexing function, PC13 pin is forced to be push-pull output by setting PC13EN bit of RTC_TACFG register; PC13VAL bit is used to set the value of PC13 pin output data. Then the push-pull output and data value of PC13 pin can be maintained in standby mode.

The following table shows the priority sequence followed by the output

mechanism:

Table 59 PC13 Pin Controlled by RTC

Pin configuration and function	RTC_ALARM Output enable	RTC_CALIB Output enable	RTC_TAMP1 Input enable	RTC_TS Input enable	PC13EN	PC13VAL
RTC_ALARM Open-drain output	1	No impact	No impact	No impact	No impact	0
RTC_ALARM Push-pull output	1	No impact	No impact	No impact	No impact	1
RTC_CALIB Push-pull output	0	1	No impact	No impact	No impact	No impact
RTC_TAMP1 Floating input	0	0	1	0	No impact	No impact
RTC_TS and RTC_TAMP1 Floating input	0	0	0	1	No impact	No impact
RTC_TS Floating input	0	0	0	1	No impact	No impact
Forced to push-pull output	0	0	0	0	1	PC13 output data value
Wake-up pin or standard GPIO	0	0	0	0	0	No impact

When PC14 and PC15 do not use LSECLK oscillator, PC14/PC15 can be forced to be push-pull output by setting PC14EN and PC15EN bits of RTC_TACFG register; PC14VAL and PC15VAL bits set the output data, and the push-pull output and data value of PC14 and PC15 can be maintained in standby mode.

The following table shows the priority sequence followed by the output mechanism:

Table 60 PC14 Pin Controlled by LSECLK

Pin configuration and function	RCM_RTCCTRL LSEEN bit of register	RCM_RTCCTRL LSEBCFG bit of register	PC14EN	PC14VAL
LSECLK oscillator	1	0	No impact	No impact
LSECLK bypass	1	1	No impact	No impact
Forced to push-pull output	0	No impact	1	PC14 output data value
Standard GPIO	0	No impact	0	No impact

Table 61 PC15 Pin Controlled by LSECLK

Pin configuration and function	RCM_RTCCTRL LSEEN bit of register	RCM_RTCCTRL LSEBCFG bit of register	PC15EN	PC15VAL
LSECLK oscillator	1	0	No impact	No impact
Forced to push-pull output	1	1	1	PC15 output data value
	0	No impact		
Standard GPIO	0	No impact	0	No impact

19.5.2 Timebase unit

Clock Source

RTC has three clock sources RTC_CLK:

- External LSECLK crystal oscillator
- External HSECLK crystal oscillator
- Internal LSICLK

Different clock sources are configured through RCM peripheral of clock controller.

Prescaler

The power consumption of RTC peripheral should be minimized as far as possible. Considering the power consumption, dual prescalers, 7bit asynchronous prescaler APSC and 15bit synchronous prescaler SPSC, are used in RTC.

RTC_CLK first passes through the asynchronous prescaler, and the clock after frequency division reaches the synchronous prescaler. Two prescalers can be reasonably configured to generate a 1Hz clock to provide date.

When the prescaler is used, it is suggested that the asynchronous prescaler should be adjusted as high as possible to reduce power consumption.

The synchronous prescaled value can also be used as the reload value of the subsecond counter.

19.5.3 Clock calibration

Clock synchronization

RTC can realize clock synchronization according to external high-precision clcok and the register RTC_SHIFT. The deviation between RTC clock and external clock is detected mainly by acquiring the timestamps of subsecond time period twice. Since the synchronous prescaled value is used as the reload value of the subsecond counter, and the SFSEC bit of register RTC_SHIFT only

works in the subsecond counter, the SFSEC bit can be adjusted to finely tune the RTC clock and increase or decrease several cycles artificially.

Reference clock

RTC supports internal reference clock detection, which can be used to compensate the deviation of external LSECLK crystal oscillator. Set RCLKDEN bit to enable the reference clock detection, compare the external 50Hz or 60Hz reference clock with the internal 1Hz clock of RTC through RTC_REFIN pin, and through this mechanism, the 1Hz clock after LSECLK frequency division is automatically compensated.

After the reference clock detection is enabled, the synchronous and asynchronous prescalers of the clock unit must be configured as the default value.

The reference clock detection cannot be used simultaneously with the clock synchronization, and it should be disabled in standby mode.

RTC digital calibration

RTC uses 220 RTC_CLK as a calibration cycle by default. In addition, 219 or 218 RTC_CLK can be set as a calibration cycle through the registers CALW16 and CALW8. When LSECLK is used as RTC_CLK clock source, the calibration cycle of RTC can be 32s, 16s or 8s.

- 16s calibration cycle; set RECALF[0] to '0' by hardware
- 8s calibration cycle; set RECALF[1:0] to '00' by hardware

Take 32s calibration cycle as an example, the calibration mechanism is to add or reduce some RTC_CLK signals in the calibration cycle.

- When RECALF is used, RECALF RTC_CLK are reduced every 220 RTC_CLK
- When ICALFEN is used and ICALFEN=1, one RTC_CLK is added every 211 RTC_CLK
- When RECALF is used and ICALFEN is enabled, (512 * ICALFEN - RECALF) RTC_CLK are added every 220 RTC_CLK

19.5.4 RTC write protection

In order to prevent counting exception caused by accidental write, RTC register adopts write protection mechanism. Only when the write protection is removed, can the register with write protection function be operated.

After power-on, RTC register will enter the write protection state and the protection cannot be removed by system reset. The write protection can be removed by writing special keywords '0xCA' and '0x53' to the register RTC_WRPROT. If wrong keyword is written, RTC will immediately enable write protection.

19.5.5 Date register

RTC has subsecond, time and date shadow registers encoded by BCD, which are RTC_SUBSEC, RTC_TIME and RTC_DATE respectively. The current date can be obtained by accessing the shadow register or obtained directly from the date register. The time system of 24 hours and 12 hours can be selected by TIMEFCFG bit of configuration register RTC_CTRL.

RTC updates the shadow register and sets the flag bit RSFLG every two RTC_CLK cycles. When waking up from the stop or standby mode, generally the shadow register will not be updated, which requires waiting for up to two RTC_CLK cycles. The reset of shadow register is caused by system reset.

The shadow register is synchronized with f_{APB1}.

The way to read the date can be selected by RCMCFG bit of configuration register RTC_CTRL.

RCMCFG=0, read the date from the shadow register

In this mode, it is recommended that f_{APB1} should be greater than 7*f_{RTC_CLK}. If f_{APB1} is too small, to ensure the normal reading of date value, it is required to read the shadow register twice. If the date obtained twice is the same, the date is read successfully.

After the shadow register is updated, the flag bit RSFLG will be set. The software can read the date only after the bit RSFLG is set. Every time the date is read, the RSFLG flag should be cleared manually.

When waking up from stop or standby mode, since the shadow register is not updated, the RSFLG flag should be cleared immediately.

RCMCFG=1, read the date from the date register

When f_{APB1} is less than 7*f_{RTCCLK} or the system wakes up from low-power mode, it is recommended to read the date directly from the date register.

If RSFLG flag bit is not set to 1 when reading the date just at the stage of change of date register, it is required to read the date twice. Therefore, it is also advised to read the date register twice. When the read date value is the same twice, it means that the date is read successfully.

19.5.6 Time compensation

Due to seasonal changes, time compensation is sometimes needed to make it more suitable for daily needs. RTC integrates time compensation unit and its summer time flag. Users can choose whether to enable time compensation according to their own needs.

By setting STCCFG bit of the register RTC_CTRL, the summer time will

increase by 1 hour; by setting WTCCFG bit of the register RTC_CTRL, the winter time will decrease by 1 hour. BAKP flag is used to record whether the summer time is set.

19.5.7 Programmable alarm

As a real-time clock, RTC integrates alarm function, and it runs mainly through alarm configuration register and alarm mask in combination with date register.

Configure the alarm and alarm mask through the registers RTC_ALRMA and RTC_ALRMASS, and the alarm mask informs RTC to pay attention to the time period of the alarm. After the alarm function is enabled, the alarm will be triggered only when the concerned time period reaches the set value. At this time, the alarm flag is set. If the alarm interrupt is enabled, the interrupt processing will be triggered.

Select "seconds" as the time period of the alarm, and only when the synchronous prescaler value is greater than 3, can the alarm operate normally.

19.5.8 Timestamp

RTC supports timestamp function and the RTC_TS pin works in combination with the timestamp register.

The timestamp polarity is detected through TSETECFG bit of the register RTC_CTRL. When RTC_TS pin recognizes the external timestamp edge signal, RTC will automatically latch the current date in the subsecond, time and date timestamp registers, and the timestamp flag bit TSFLG will be set to 1. If the timestamp interrupt is enabled, the timestamp interrupt processing will be triggered.

When TSFLG flag bit is set to 1, and another timestamp event occurs, the timestamp will overrun, and the flag bit TSOVRFLG will be set to 1. If another timestamp event is detected once TSFLG flag is cleared, both TSFLG and TSOVRFLG flags will be set to 1.

19.5.9 Tamper detection

Tamper detection is a kind of data self-destruction protection device to prevent data leakage caused by tamper. Through the hardware circuit design, the tamper detection signal is transmitted to the tamper detection pin.

Tamper detection has multiple tamper detection pins, and each pin is enabled by a register bit separately. In order to detect real tamper events better, signal filtering can be configured, and tamper detection polarity can be configured for each pin.

Tamper detection polarity

The low level/rising edge and high level/falling edge can be selected as tamper

detection polarity through TPxALCFG bit in the register RTC_TACFG.

Tamper signal filter

TPSFSEL bit of the register RTC_TACFG is used to configure the sampling frequency of tamper detection, and TPFCSEL bit of RTC_TACFG is used to configure after how many valid tamper signals are detected continuously, a tamper event can be generated.

In particular, if a tamper signal has been generated on the tamper detection pin before the tamper detection pin is enabled, a tamper event will be immediately generated by enabling the tamper detection pin.

Tamper timestamp

At some times, in order to record the tamper detection events, RTC can latch the current tamper timestamp and this function can be enabled quickly through TPTSEN bit of the register RTC_TACFG, not needing to enable the timestamp function additionally.

19.5.10 RTC output

RTC output transmits the internal RTC calibration clock and alarm signal to the outside through PC13 pin.

RTC calibration clock

Calibration clock output is generally used to observe the accuracy of RTC clock source, and the observed value is used to calibrate the clock source. 512Hz and 1Hz signal output sources can be selected through CALOSEL bit of RTC_CTRL register, and CALOEN bit of RTC_CTRL register can enable the calibration output.

Alarm signal

When the alarm is running, the alarm event can be output as pulse signals. OUTSEL bit of RTC_CTRL register is used to select the signal output source, and POLCFG bit is used to configure the output polarity.

19.6 Register address mapping

Table 62 RTC Register Address Mapping

Register name	Description	Offset address
RTC_TIME	RTC time register	0x00
RTC_DATE	RTC date register	0x04
RTC_CTRL	RTC control register	0x08

Register name	Description	Offset address
RTC_STS	RTC status register	0x0C
RTC_PSC	RTC prescaler register	0x10
RTC_ALRMA	RTC alarm A register	0x1C
RTC_WRPROT	RTC write protection register	0x24
RTC_SUBSEC	RTC subsecond register	0x28
RTC_SHIFT	RTC shift register	0x2C
RTC_TSTIME	RTC timestamp time register	0x30
RTC_TSDATE	RTC timestamp date register	0x34
RTC_TSSUBSEC	RTC timestamp subsecond register	0x38
RTC_CAL	RTC calibration register	0x3C
RTC_TACFG	RTC tamper and multiplexing configuration register	0x40
RTC_ALRMASS	RTC alarm A subsecond register	0x44
RTC_BAKPx	RTC backup register	0x50-0x60

19.7 Register functional description

19.7.1 RTC time register (RTC_TIME)

RTC_TIME is date time shadow register, and this register can be written only in initialization mode to be put in write protection state.

Offset address: 0x00

Power-on reset value: 0x0000 0000

System reset: 0xXXXX XXXX

Field	Name	R/W	Description
3:0	SECU	R/W	Second Ones Unit in BCD Format Setup
6:4	SECT	R/W	Second Ten's Place Unit in BCD Format Setup
7	Reserved		
11:8	MINU	R/W	Minute Ones Unit in BCD Format Setup
14:12	MINT	R/W	Minute Ten's Place Unit in BCD Format Setup
15	Reserved		
19:16	HRU	R/W	Hour Ones Unit in BCD Format Setup
21:20	HRT	R/W	Hour Ten's Place Unit in BCD Format Setup
22	TIMEFCFG	R/W	Time Format Configure 0: AM or 24-hour system 1：PM
31:23	Reserved		

19.7.2 RTC date register (RTC_DATE)

RTC_DATE is date shadow register, and this register can be written only in initialization mode and is in write protection state.

Offset address: 0x04

Reset value: 0x0000 2101

Field	Name	R/W	Description
3:0	DAYU	R/W	Day Ones Unit in BCD Format Setup
5:4	DAYT	R/W	Day Ten's Place Unit in BCD Format Setup
7:6	Reserved		
11:8	MONU	R/W	Month Ones Unit in BCD Format Setup
12	MONT	R/W	Month Ten's Place Unit in BCD Format Setup
15:13	WEEKSEL	R/W	Week Day Units Select 000: Disable 001: Monday ... 111: Sunday
19:16	YRU	R/W	Year Ones Unit in BCD Format Setup
23:20	YRT	R/W	Year Ten's Place Unit in BCD Format Setup
31:24	Reserved		

19.7.3 RTC control register (RTC_CTRL)

（1） The bits 7, 6 and 4 of this register can be written only in initialization mode.

（2） It is not recommended to rewrite this register when the number of hours in the date increases, which is because the correct increment of hours may be masked.

（3） The written values of STCCFG and WTCCFG will take effect from next second.

（4） This register is under write protection.

Offset address: 0x08

Power-on reset value: 0x0000 0000

System reset: 0xXXXX XXXX

Field	Name	R/W	Description
2:0	Reserved		
3	TSETECFG	R/W	Time Stamp Event Trigger Edge Configure This bit indicates that RTC_TS generates a timestamp event on rising edge or falling edge. 0: Rising edge 1: Falling edge This bit will be changed when TSEN=0.

Field	Name	R/W	Description
4	RCLKDEN	R/W	RTC_REFIN reference clock detection enable 0: Disable 1: Enable SPSC must be 0x00FF
5	RCMCFG	R/W	Read Calendar Value Mode Configure 0: The date value is read from the shadow register, and the shadow register is updated every two RTCCLK cycles 1: Read the date value from the date register If the clock frequency of APB1 is lower than seven times of RTCCLK frequency, RCMCFG must be set to 1.
6	TIMEFCFG	R/W	Time Format Configure 0: 24-hour/day format 1: AM/PM time format
7	Reserved		
8	ALREN	R/W	Alarm A Function Enable 0: Disable 1: Enable
10:9	Reserved		
11	TSEN	R/W	Time Stamp Enable 0: Disable 1: Enable
12	ALRIEN	R/W	Alarm A Interrupt Enable 0: Disable 1: Enable
14:13	Reserved		
15	TSIEN	R/W	Time Stamp Interrupt Enable 0: Disable 1: Enable
16	STCCFG	R/W	Summer Time Change Configure The bit will always be 0 in the reading process; if this bit is set not in the initialization mode, the date time will increase by 1. 0: Invalid 1: The current time increases by 1 hour to calibrate the summer time change
17	WTCCFG	R/W	Winter Time Change Configure The bit will always be 0 in the reading process; if this bit is set not in the initialization mode, and HRx of RCT_TIME register is 0, this bit is invalid, and if HRx is not 0, the date time will decrease by 1. 0: Invalid 1: The current time increases by 1 hour to calibrate the winter time change
18	BAKP	R/W	Backup Value Setup This bit indicates whether the summer time has changed and is written by the user.

Field	Name	R/W	Description
19	CALOSEL	R/W	Calibration Output Value Select When CALOEN=1, this bit is used to select the output signal of RTC_CALIB. 0:512Hz 1:1Hz The above frequency is valid when RTCCLK is 32.768kHz and the prescaler is at the default value (APSC=127, SPSC=255).
20	POLCFG	R/W	Output Polarity Configure This bit indicates the level state of the pin when ALRAFLG bit is set to 1 (depending on OUTSEL bit). 0: High level 1: Low level
22:21	OUTSEL	R/W	Output Way Select This bit is used to select the flag bit associated with RTC_ALARM output 00: Output is disabled 01: Alarm A output is enabled 10: Reserved 11: Reserved
23	CALOEN	R/W	Calibration Output Enable This bit is used to enable RTC_CAL output 0: Disable 1: Enable
31:24	Reserved		

19.7.4 RTC status register (RTC_STS)

This register (except RTC_STS[13:8] bit) is in write protection state.

Offset address: 0x0C

Power-on reset value: 0x0000 0007

System reset: 0xXXXX XXXX

Field	Name	R/W	Description
0	ALRWFLG	R	Alarm A Write Occur Flag When ALREN=0 for RTC_CTRL, the value of alarm A will change and this bit will be set to 1 by hardware; this bit will be cleared by hardware in initialization mode. 0: The alarm A cannot be updated 1: The alarm A can be updated
2:1	Reserved		
3	SOPFLG	R	Shift Operation Pending Occur Flag 0: Not occur 1: Occurred When a shift operation is generated by writing to RTC_SHIFT register, this bit will be set to 1 by hardware immediately. After corresponding shift operation is performed, this bit will be cleared to 0 by software. It is invalid to write to SOPFLG.

Field	Name	R/W	Description
4	INITSFLG	R	Initialization State Occur Flag When the "year" field in the date is not "0", this bit will be set by hardware. 0: Not occur 1: Occurred
5	RSFLG	RC_W0	Registers Synchronization Occur Flag When the content in the date register is copied to the shadow registers (RTC_SUBSEC, RTC_TIME and RTC_DATE), this bit is set to 1 by hardware; when shifting operation is pending (SOPFLG=1) or is in the mode that the shadow register is ignored (RCMCFG=1), this bit is cleared to 0 by hardware in initialized mode; this bit can also be cleared by software. This bit is cleared by hardware/software in initialization mode. 0: Not synchronize 1: Synchronize
6	RINITFLG	R	Register Initialization Occur Flag This bit is set to "1", RTC is in initialization state, and the time, date and prescaler registers can be updated. 0: Cannot initialize 1: Initialize
7	INITEN	R/W	Initialization Mode Enable 0: Free running mode 1: Initialization mode; it can be used to program RTC_TIME, RTC_DATE and RTC_PSC. The counter stops counting, and after INITEN is reset, the counter will start counting from a new value.
8	ALRAFLG	RC_W0	Alarm A Match Occur Flag When RTC_TIME and RTC_DATE match the alarm A register RTC_ALRMA, this flag is set by hardware. This flag can be cleared by writing 0 by software.
10:9	Reserved		
11	TSFLG	RC_W0	Time Stamp Occur Flag When a timestamp event occurs, this flag is set to 1 by hardware; it can be cleared by writing 0 by software.
12	TSOVRFLG	RC_W0	Time Stamp Overflow Occur Flag When TSFLG=1 and a timestamp event is generated, this flag bit is set to 1 by hardware; it can be cleared by writing 0 by software. It is recommended to clear this bit after TSFLG flag bit is cleared.
13	TP1FLG	RC_W0	RTC_TP1FLG Detection Occur Flag When a tamper event is detected in RTC_TP1FLG input, this flag is set to 1 by hardware; it can be cleared by writing 0 by software.
15:14	Reserved		
16	RCALPFLG	R	Recalibration Pending Occur Flag When the software writes to RTC_CAL, this bit is set to 1 automatically, and the RTC_CAL register is locked. This bit will return 0 when other new calibration setting is performed.
31:17	Reserved		

19.7.5 RTC prescaler register (RTC_PSC)

The register can only be written in the initialization mode, and the initialization must be completed by two independent write accesses, and the register is in write protection state.

Offset address: 0x10

Power-on reset value: 0x007F 00FF

System reset: 0xXXXX XXXX

Field	Name	R/W	Description
14:0	SPSC	R/W	Synchronous Prescaler Coefficient ck_spre frequency=ck_apre frequency/(SPSC+1)
15	Reserved		
22:16	APSC	R/W	Asynchronous Prescaler Coefficient ck_apre frequency=RTCCLK frequency/(APSC+1)
31:23	Reserved		

19.7.6 RTC alarm A register (RTC_ALRMA)

This register can be written only when ALRWFLG of RTC_STS is set to 1 or in initialization mode, and it is in write protection state.

Offset address: 0x1C

Power-on reset value: 0x0000 0000

System reset: 0xXXXX XXXX

Field	Name	R/W	Description
3:0	SECU	R/W	Second Ones Unit in BCD Format Setup
6:4	SECT	R/W	Second Ten's Place Unit in BCD Format Setup
7	SECMEN	R/W	Alarm A Seconds Mask Enable 0: If the "second" matches, set Alarm A 1: Mask the effect of the "second" value on Alarm A
11:8	MINU	R/W	Minute Ones Unit in BCD Format Setup
14:12	MINT	R/W	Minute Ten's Place Unit in BCD Format Setup
15	MINMEN	R/W	Alarm A Minutes Mask Enable 0: If the "minute" matches, set Alarm A 1: Mask the effect of the "minute" value on Alarm A
19:16	HRU	R/W	Hour Ones Unit in BCD Format Setup
21:20	HRT	R/W	Hour Ten's Place Unit in BCD Format Setup
22	TIMEFCFG	R/W	Time Format Configure 0: AM or 24-hour system 1：PM
23	HRMEN	R/W	Alarm A Hours Mask Enable 0: If the "hour" matches, set Alarm A 1: Mask the effect of the "hour" value on Alarm A
27:24	DAYU	R/W	Day Ones Unit in BCD Format Setup

Field	Name	R/W	Description
29:28	DAYT	R/W	Day Ten's Place Unit in BCD Format Setup
30	WEEKSEL	R/W	Week Day Select 0: DAYU means date 1: DAYU means the number of weeks. DAYT has no effect.
31	DATEMEN	R/W	Alarm A Date Mask Enable 0: If the date/week matches, set Alarm A 1: Mask the effect of the date/week value on Alarm A

19.7.7 RTC write protection register (RTC_WRPROT)

Offset address: 0x24

Reset value: 0x0000 0000

Field	Name	R/W	Description
7:0	KEY	W	Write Protection Key Value Setup This byte is written by software; read this byte and it is always 0x00.
31:8	Reserved		

19.7.8 RTC subsecond register (RTC_SUBSEC)

Offset address: 0x28

Reset value: 0x0000 0000

Field	Name	R/W	Description
15:0	SUBSEC	R	Sub Second Value Setup SUBSEC is the value of synchronous prescaler counter. It is determined by the following formula: Subsecond value=(SPSC-SUBSEC)/(SPSC+1) After one shift operation is performed, SUBSEC may be greater than SPSC. The correct time/date is one second less than RTC_TIME/RTC_DATE.
31:16	Reserved		

19.7.9 RTC shift register (RTC_SHIFT)

This register is in write protection state.

Offset address: 0x2C

Reset value: 0x0000 0000

Field	Name	R/W	Description
14:0	SFSEC	W	Subtract a Fraction of a Second Setup This bit field can only be written; read this byte and it is always 0. Writing to this bit is invalid while an operation is being executed. The set SFSEC value will be added to the synchronous prescaler counter. If the counter counts down, the clock will be delayed, and the delay time is determined by the following formula: Delay (seconds)=SFSEC/(SPSC+1) When it takes effect at the same time with ADD1SECEN, advance the clock and a fraction of a second will be added; the specific added value is determined by the following formula: Advance(seconds)=(1-(SFSEC/(SPSC+1)))

Field	Name	R/W	Description
			Conduct write operation to this bit and RSFLG bit can be cleared. The software keeps running until RSFLG is set to 1 to ensure that the value of the shadow register is synchronized with the shift time.
30:15	Reserved		
31	ADD1SECEN	W	Add One Second Enable 0: Not add 1: The clock/date increases by one second This bit can only be written; read this byte and it is always 0. Writing to this bit is invalid while an operation is being executed. When it takes effect at the same time with SFSEC, it can increase the value of the clock by several tenths of a second.

19.7.10 RTC timestamp time register (RTC_TSTIME)

This register is valid only when TSFLG of RTC_STS is set to 1. When TSFLG bit is reset, the content of this register will be cleared.

Offset address: 0x30

Power-on reset value: 0x0000 0000

System reset: 0xXXXX XXXX

Field	Name	R/W	Description
3:0	SECU	R	Second Ones Unit in BCD Format Setup
6:4	SECT	R	Second Ten's Place Unit in BCD Format Setup
7	Reserved		
11:8	MINU	R	Minute Ones Unit in BCD Format Setup
14:12	MINT	R	Minute Ten's Place Unit in BCD Format Setup
15	Reserved		
19:16	HRU	R	Hour Ones Unit in BCD Format Setup
21:20	HRT	R	Hour Ten's Place Unit in BCD Format Setup
22	TIMEFCFG	R	Time Format Configure 0: AM or 24-hour system 1：PM
31:23	Reserved		

19.7.11 RTC timestamp date register (RTC_TSDATE)

This register is valid only when TSFLG bit of RTC_STS is set to 1. When TSFLG bit is reset, this register will be cleared.

Offset address: 0x34

Power-on reset value: 0x0000 0000

System reset: 0xXXXX XXXX

Field	Name	R/W	Description
3:0	DAYU	R	Day Ones Unit in BCD Format Setup
5:4	DAYT	R	Day Ten's Place Unit in BCD Format Setup

Field	Name	R/W	Description
7:6	Reserved		
11:8	MONU	R	Month Ones Unit in BCD Format Setup
12	MONT	R	Month Ten's Place Unit in BCD Format Setup
15:13	WEEKSEL	R	Week Day Units Select 000: Disable 001: Monday ... 111: Sunday
31:16	Reserved		

19.7.12 RTC timestamp subsecond register (RTC_TSSUBSEC)

This register is valid only when TSFLG bit of RTC_STS register is set to 1.

When TSFLG bit is reset, the content of this register will be cleared.

Offset address: 0x38

Power-on reset value: 0x0000 0000

System reset: 0xXXXX XXXX

Field	Name	R/W	Description
15:0	SUBSEC	R	Sub Second Value Setup When a timestamp event occurs, SUBSEC[15:0] is the value in synchronous prescaler counter.
31:16	Reserved		

19.7.13 RTC calibration register (RTC_CAL)

This register is in write protection state.

Offset address: 0x3C

Power-on reset value: 0x0000 0000

System reset: 0xXXXX XXXX

Field	Name	R/W	Description
8:0	RECALF	R/W	Reduced Calibration Frequency Reduce date frequency: Mask RECALF pulses within 220 RTCCLK pulses (32sec if the output frequency is 32768 Hz) and the date frequency will be reduced (the resolution is 0.9537 ppm). Increase date frequency: It takes effect at the same time with ICALFEN
12:9	Reserved		
13	CAL16CFG	R/W	16 Second Calibration Cycle Period Configure When CAL16CFG is set to 1, 16-second calibration cycle is used, and it cannot be set to 1 at the same time with CAL8CFG bit. When CAL16CFG=1, RECALF[0] is always 0.
14	CAL8CFG	R/W	8 Second Calibration Cycle Period Configure When CAL8CFG is set to 1, 8-second calibration cycle is used, and it cannot be set to 1 at the same time with CAL16CFG bit. When CAL8CFG=1, RECALF[1:0] is always 00.

Field	Name	R/W	Description
15	ICALFEN	R/W	Increase Calibration Frequency Enable 0: RTCCLK pulse is not increased 1: One RTCCLK pulse is increased (the frequency increases by 488.5 ppm) every 211 pulses It takes effect at the same time with RECALF, and when the resolution is high, the date frequency will be reduced. If the input frequency is 32768Hz, the number of RTCCLK pulses added in the 32-second window is determined by the following formula: (512*ICALFEN)–RECALF。
31:16	Reserved		

19.7.14 RTC tamper and multiplexing configuration register (RTC_TACFG)

Offset address: 0x40

Power-on reset value: 0x0000 0000

System reset: 0xXXXX XXXX

Field	Name	R/W	Description
0	TP1EN	R/W	RTC_TAMP1 Input Detection Enable 0: Disable 1: Enable
1	TP1ALCFG	R/W	RTC_TAMP1 Input Active Level Configure When TPFCSEL!=00, this bit determines that RTC_TAMP1 input will trigger a tamper detection event when it remains high/low. 0: Low level 1: High level When TPFCSEL=00, this bit determines that RTC_TAMP1 input will trigger a tamper detection event when it is on rising/falling edge. 0: Rising edge 1: Falling edge
2	TPIEN	R/W	Tamper Interrupt Enable 0: Disable 1: Enable
6:3	Reserved		
7	TPTSEN	R/W	Tamper Detection Event Timestamp Enable This bit determines whether the timestamp generated by the tamper detection event is saved 0: Not save 1: Save This bit is still valid when TSEN=0 for RTC_CTRL register.
10:8	TPSFSEL	R/W	Tamper Sampling Frequency Select These bits determine the sampling frequency of each RTC_TAMPx input. 0x0：RTCCLK/32768 0x1：RTCCLK/16384 0x2：RTCCLK/8192 0x3：RTCCLK/4096 0x4：RTCCLK/2048

Field	Name	R/W	Description
			0x5：RTCCLK/1024 0x6：RTCCLK/512 0x7：RTCCLK/256
12:11	TPFCSEL	R/W	RTC_TAMPx Filter Count Select These bits determine the number of sampling times after which a tamper event is activated at specific level (TAMP*TRG). TPFCSEL is valid for each RTC_TAMPx input. 0x0: Activate the tamper event on the edge where RTC_TAMPx input is converted into valid level 0x1: Continuous sampling twice 0x2: Continuous sampling for four times 0x3: Continuous sampling for eight times
14:13	TPPRDUSEL	R/W	RTC_TAMPx Precharge Duration Select These bits determine the number of RTCCLK cycles which are enabled by pull-up resistor before sampling; which is valid in each RTC_TAMPx input. 0x0：1 0x1：2 0x2：4 0x3：8
15	TPPUDIS	R/W	RTC_TAMPx Pull-up Function Disable This bit determines whether all RTC_TAMPx pins are precharged before sampling. 0: Enable (enable internal pull-up) 1: Disable
17:16	Reserved		
18	PC13VAL	R/W	RTC_ALARM Output Type/PC13 Value Configure When PC13 is used to output RTC_ALARM, this bit determines the output mode of RTC_ALARM: 0: Open-drain output 1: Push-pull output When all RTC multiplexing functions are disabled and PC13EN=1, this bit is used to set the output value of PC13.
19	PC13EN	R/W	PC13 Mode Enable 0: PC13 is controlled by GPIO configuration register, and is floating in standby mode. 1: When RTC multiplexing function is disabled, PC13 is forced to push-pull output mode.
20	PC14VAL	R/W	PC14 Output Value Setup Disable LSECLK and PC14EN=1, and this bit sets the output value of PC14.
21	PC14EN	R/W	PC14 Mode Enable 0: PC14 is controlled by GPIO configuration register, and is floating in standby mode. 1: When LSECLK is disabled, PC14 is forced to push-pull output mode

Field	Name	R/W	Description
22	PC15VAL	R/W	PC15 Output Value Setup Disable LSECLK and PC15EN=1, and this bit sets the output value of PC15.
23	PC15EN	R/W	PC15 Mode Enable 0: PC15 is controlled by GPIO configuration register, and is floating in standby mode. 1: When LSECLK is disabled, PC15 is forced to push-pull output mode.
31:24	Reserved		

19.7.15 RTC alarm clock A subsecond register (RTC_ALRMASS)

This register can be written only when ALREN of RTC_CTRL register is reset or is in initialization mode.

This register is in write protection state.

Offset address: 0x44

Power-on reset value: 0x0000 0000

System reset: 0xXXXX XXXX

Field	Name	R/W	Description
14:0	SUBSEC	R/W	Sub Second Value Setup The subsecond value is compared with the value in the synchronous prescaler counter to determine whether to activate the alarm A, and only the bits from 0 to MASKSEL-1 are compared.
23:15	Reserved		
27:24	MASKSEL	R/W	Mask the Most-significant Bits Starting at This Bit Select 0x0: Alarm A is not compared. The alarm is set when the second unit increases by 1 0x1: When comparing with alarm A, SUBSEC[14:1] is not involved, and only SUBSEC[0] is involved 0x2: When comparing with alarm A, SUBSEC[14:2] is not involved, and only SUBSEC[1:0] is involved 0x3: When comparing with alarm A, SUBSEC[14:3] is not involved, and only SUBSEC[2:0] is involved ... 0xC: When comparing with alarm A, SUBSEC[14:12] is not involved, and only SUBSEC[11:0] is involved 0xD: When comparing with alarm A, SUBSEC[14:13] is not involved, and only SUBSEC[12:0] is involved 0xE: When comparing with alarm A, SUBSEC[14] is not involved, and only SUBSEC[13:0] is involved 0xE: When comparing the alarm A, 15 SUBSEC bits all take part in, and the alarm can be activated only when all of them match. The synchronous counter overrun bit (Bit 15) is never compared. This bit is not 0 only after shift operation.
31:28	Reserved		

19.7.16 RTC backup register (RTC_BAKPx) (x=0-4)

Offset address: 0x50-0x60

Power-on reset value: 0x0000 0000

Reset value: 0xXXXX XXXX

Field	Name	R/W	Description
31:0	BAKP	R/W	Backup Value Setup If the VDD is powered off, the register will be powered down and cannot be maintained; this register willbe reset when a temper detection event occurs or flash read protection is disabled. The contents of this bit field are valid even if the device is running in low-power mode.

20 Controller area network (CAN)

20.1 Full name and abbreviation of terms

Table 63 Full Name and Abbreviation Description of Terms

Full name in English	English abbreviation
First Input First Output	FIFO
Request	REQ

20.2 Introduction

CAN is abbreviation of Controller Area Network, and is serial communication protocol of ISO international standardization and supports CAN Protocol 2.0A and 2.0B. In CAN protocol, the sender transmits the message to all receivers in the form of broadcast. When the node receives the message, it will go through the filter group and decide whether the message is needed according to the identifier. This design saves the CPU overhead.

20.3 Main characteristics

（1） Support CAN protocol 2.0A and 2.0B

（2） The maximum baud rate of communication is 1Mbit/s

（3） Transmitting function

- There are three transmit mailboxes
- The priority of transmitting message can be configured
- Record the transmission time

（4） Receiving function

- Have two receive FIFO with three depth levels
- Have 14 filter groups.
- Record the receiving time

20.4 Functional description

20.4.1 Characteristics of CAN physical layer

Multiple communication nodes can be designed on the CAN bus, and each node consists of a CAN controller and a transceiver. The controller and the transceiver are connected through CAN_TX and CAN_RX to transmit logic signals; the transceiver and the bus are connected through CAN_High and CAN_Low to transmit differential signals.

20.4.2 Message structure

Figure 68 Standard Data Frame

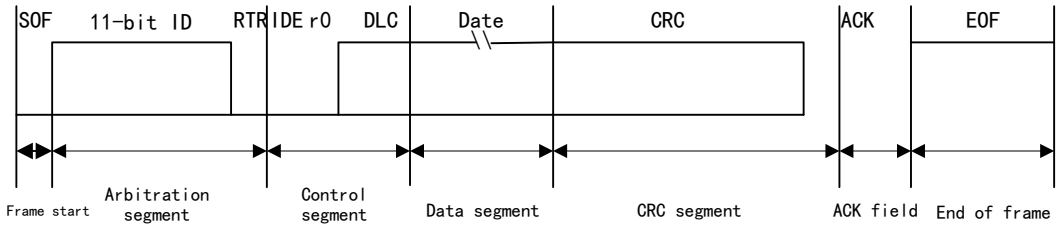
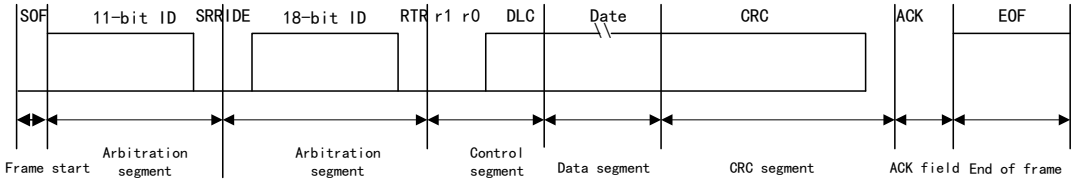


Figure 69 Extended Data Frame



Note:

（1）Frame start: It is used to inform each node that there will be data for transmission.

（2）Arbitration segment: It is used to decide which message can be transmitted when multiple messages are transmitted. Main content of this segment is ID information, the ID in standard format is 11 bits, and the ID in extended format is 29 bits.

（3）Control segment: The main content of this segment is data length code (DLC), which is used to indicate the number of bytes in the data segment of the message. The data segment has up to 8 bytes.

（4）Data segment: Include the data information to be transmitted by the node.

（5）CRC segment: CRC check code is used to ensure correct transmission of the messages.

（6）ACK segment: This segment includes ACK slot bit and ACK delimiter bit. The transmitting node in ACK slot transmits recessive bits, while the receiving node transmits the dominant bit in this bit to acknowledge.

（7）Frame end: Seven recessive bits transmitted by the transmitting nodes are used to indicate the end.

20.4.3 Working mode

CAN has three main working modes: initialization mode, normal mode and sleep mode.

20.4.3.1 Initialization mode

Set the INITREQ bit of the configuration register CAN_MCTRL to 1 to request to enter the initialization mode; clear the INITFLG bit to 0 to confirm entering the initialization mode.

Clear the INITREQ bit of the configuration register CAN_MCTRL to 0 to request

exiting the initialization mode; clear the INITFLG bit to 0 to confirm exiting the initialization mode.

Message receiving and transmitting is disabled in initialization mode.

20.4.3.2 Normal Mode

Clear the INITREQ bit of the configuration register CAN_MCTRL to 0 by software to request to enter the normal mode from the initialization mode; wait for the hardware to clear the INITFLG bit to 0 to confirm entering the normal mode.

Message receiving and transmitting is allowed in normal mode.

20.4.3.3 Sleep mode

Set the SLEEPREQ bit of the configuration register CAN_MCTRL to 1 to request to enter the sleep mode.

The clock of CAN stops work in sleep mode, the software can normally access the mailbox register, and the CAN is in low-power state.

20.4.4 Communication mode

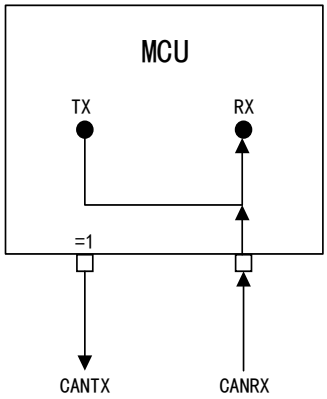
There are four communication modes: mute mode, loopback mode, mute loopback mode and normal mode. Different communication modes can be selected only in initialization mode.

20.4.4.1 Mute mode

Set the SILMEN bit of the configuration register CAN_BITTIM to 1 and select the mute mode.

In this mode, only the recessive bit (logic 1) can be transmitted to the bus, the dominant bit (logic 0) cannot be transmitted, and data can be received from the bus.

Figure 70 CAN Works in Mute Mode



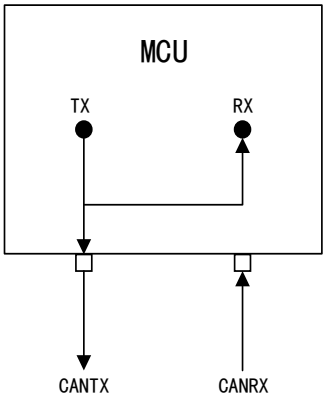
20.4.4.2 Loopback mode

Set the LBKMEN bit of the configuration register CAN_BITTIM to 1 and select

the loopback mode.

In this mode, the transmitted data are directly transmitted to the input end for receiving, the data are not received from the bus, and all data can be transmitted to the bus.

Figure 71 CAN Works in Loopback Mode

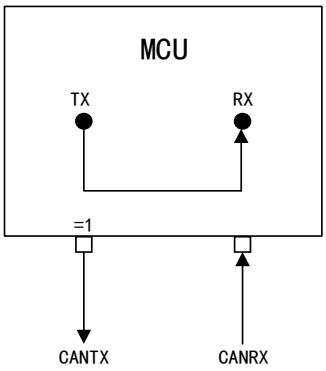


20.4.4.3 Loopback mute mode

Set the LBKMEN and SILMEN bits of the configuration register CAN_BITTIM to 1 and select the loopback mute mode.

In this mode, the transmitted data are directly transmitted to the input end for receiving, and the data are not received from the bus; only recessive bit (logic 1) can be transmitted to the bus, while the dominant bit (logic 0) cannot be transmitted.

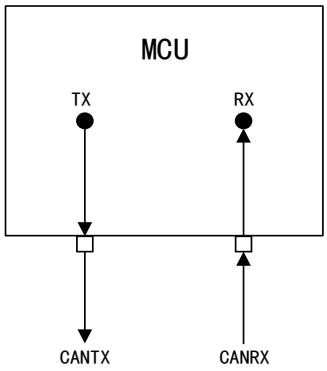
Figure 72 CAN Works in Mute Loopback Mode



20.4.4.4 Normal Mode

In this mode, data can be transmitted to the bus and be received from the bus.

Figure 73 CAN Works in Normal Mode



20.4.5 Data transmission

20.4.5.1 Conversion of transmit mailbox state

Conversion process of transmit mailbox state:

（1） First select an empty mailbox to set, submit the transmission request to the CAN bus controller by setting the TXMREQ bit of the configuration register CAN_TXMIDx to 1, and then the mailbox immediately enters the registration state.

（2） When multiple mailboxes are in the registered state, conduct priority scheduling. When a mailbox has the highest priority, it will enter the predetermined state.

（3） When the message in the transmit mailbox is transmitted to the bus, it will enter the transmitting state.

（4） After the message is transmitted successfully, the mailbox will become idle again.

20.4.5.2 Transmitting priority

When multiple messages are waiting for transmitting, determine the transmitting sequence by the TXFPCFG bit of the configuration register CAN_MCTRL:

- When the TXFPCFG bit is set to 0, the priority is determined by the message identifier, the identifier is the lowest, and the priority is the highest; if the identifier is equal, the message with small mailbox number will be transmitted first
- When the TXFPCFG bit is set to 1, the priority will be determined by the sequence of transmitting requests

20.4.5.3 Abort

Transmit the abort request by setting the ABREQFLG bit of the configuration register CAN_TXSTS to 1.

If the mailbox is in registered or predetermined state, stop transmitting the request immediately; if the mailbox is in the transmitting state, there are two situations: one is that the mailbox is successfully transmitted, and the mailbox

becomes empty, in such case, the TXSUSFLG bit of the CAN_TXSTS register is set to 1 by hardware; the other is that the mailbox fails to transmit, the mailbox becomes predetermined and the transmitting request is aborted.

20.4.5.4 Disable automatic retransmission

Generally, in time triggered communication mode, automatic retransmission should be disabled.

In the mode that the automatic retransmission is disabled, the message is transmitted only once, and no matter what the result is (success, error or arbitration loss), the hardware will not transmit the message again automatically.

When the transmission process is finished, set the REQCFLG bit of the CAN_TXSTS register to 1, and the transmission result will be on the TXSUSFLG, ARBLSTFLG and TXERRFLG bits

20.4.6 Data receiving

20.4.6.1 Receive FIFO

CAN has two receive FIFO, and each FIFO has three mailboxes. The FMNUM[1:0] bit of the register CAN_RXF reflects the number of messages currently stored; set the RFOM bit to 1 to release the output mailbox of receive FIFO; FFULLFLG is the full state flag bit; FOVRFLG is overrun state flag bit.

20.4.6.2 Receive FIFO state conversion

At the beginning FIFO is in empty state, and after receiving the message, it will become registered.

When FIFO is in registered state and three mailboxes are full, the next effective message received will enter overrun state, and there are two situations for loss of messages in overrun state:

- If FIFO lock function is disabled, the finally received message will be overwritten by new message
- If FIFO lock function is enabled, the newly received message will be discarded

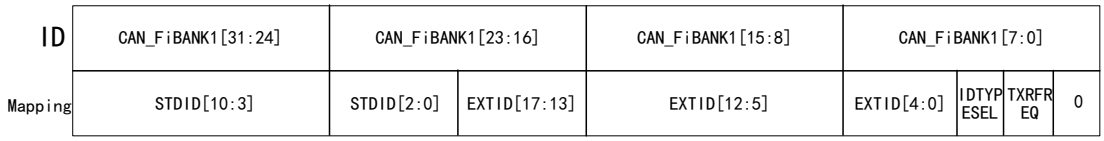
20.4.7 Filtering mechanism

Function of the filter: The receiving node decides whether the message is required according to the message identifier, and only the required message will be received after filtering. CAN controller has 14 filter groups.

20.4.7.1 Bit width

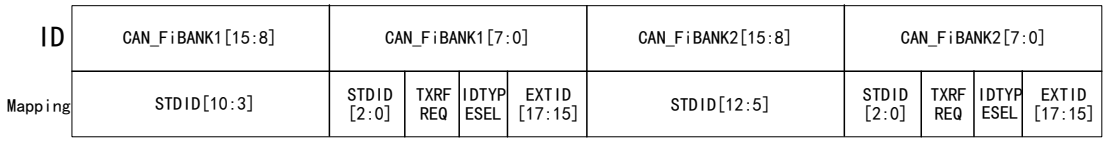
Each group of filters can configure two kinds of bit width.

Figure 74 One 32-bit Filter



ID	CAN_FiBANK1[31:24]	CAN_FiBANK1[23:16]		CAN_FiBANK1[15:8]	CAN_FiBANK1[7:0]			
Mapping	STDID[10:3]	STDID[2:0]	EXTID[17:13]	EXTID[12:5]	EXTID[4:0]	IDTYPESEL	TXRFREQ	0

Figure 75 Two 16-bit Filters



ID	CAN_FiBANK1[15:8]	CAN_FiBANK1[7:0]				CAN_FiBANK2[15:8]	CAN_FiBANK2[7:0]			
Mapping	STDID[10:3]	STDID [2:0]	TXRF REQ	IDTYP ESEL	EXTID [17:15]	STDID[12:5]	STDID [2:0]	TXRF REQ	IDTYP ESEL	EXTID [17:15]

20.4.7.2 Filtering mode

Mask bit mode

In this mode, it is only required to list some bits of the message identifier to form the mask, and the message ID should be the same as the mask, and then the message can be received

Table 64 Mask Bit Mode Example

ID	**1**	**0**	**1**	**1**	**0**	**0**	**1**	**0**	**.....**
Mask	1	0	1	1	1	0	0	1	.....
Screened ID	1	X	1	1	0	X	X	0	.....

Identifier list mode

In this mode, each bit of the message ID needs to be the same as the filter identifier, and then the message can be received.

Table65 Identifier List Mode Example

ID	**1**	**1**	**1**	**0**	**1**	**0**	**0**	**1**	**1**
ID	1	1	1	0	1	0	0	1	1
Screened ID	1	1	1	0	1	0	0	1	1

20.4.7.3 Filter priority

The priority rules are as follows:

- The priority of the filter with bit width of 32 bits is higher than that with bit width of 16 bits
- Under the condition of the same bit width, the priority of the identifier list mode is higher than that of mask bit mode
- Under the condition of the same bit width and mode, the priority of the small filtering number is high

20.4.8 Bit timing and baud rate

20.4.8.1 Bit timing

The CAN peripheral bit timing of APM32 contains three segments: synchronization segment (SYNC_SEG), time segment 1 (BS1) and time segment 2 (BS2), and the sampling points are at the junction of BS1 and BS2 segments.

- Synchronization segment (SYNC_SEG): This bit occupies one time cell
- Time segment 1 (BS1): This segment occupies 1 to 16 time cells, and it contains PROP_SEG and PHASE_SEG1 in CAN standard
- Time segment 2 (BS2): This segment occupies 1 to 8 time cells, and it represents PHASE_SEG2 in CAN standard

20.4.8.2 Calculation of baud rate

Time of BS1 segment: Ts1=Tq* (TIMSEG1[3:0]+1)

Time of BS2 segment:Ts2=Tq* (TIMSEG2[2:0]+1)

Time of one data bit: T1bit=1Tq+Ts1+Ts2

Baud rate=1/ T1bit

Tq =（BRPSC+1）* TPCLK

20.4.9 Error management

Transmit the error counter through the TXERRCNT bit of the configuration register CAN_ERRSTS and receive the error counter through the RXERRCNT bit of the register CAN_ERRSTS to reflect the error management of CAN bus.

Control the generation of interrupts in error state through the ERRIEN bit of the configuration register CAN_INTEN.

20.4.9.1 Bus-off recovery

When the TXERRCNT of the CAN error status register is greater than 255, the CAN bus controller will enter the bus-off state, then the BOFLG bit of the register CAN_ERRSTS will be set to 1, and in this state, the CAN bus controller cannot receive and transmit messages.

Decide the bus-off recovery mode through the ALBOFFM bit of the configuration register CAN_MCTRL:

- If the ALBOFFM bit is set to 1, once the hardware detects 11 continuous recessive bits for 128 times, it will exit the bus-off state automatically;
- If the ALBOFFM bit is set to 0, after the software requests entering and then exiting the initialization mode, it will exit the bus-off state.

20.4.10 Interrupt

Events generating transmission interrupt:

- The hardware sets REQCFLG0 bit of the register CAN_TXSTS to 1, and the transmit mailbox 0 becomes empty
- The hardware sets REQCFLG1 bit of the register CAN_TXSTS to 1, and the transmit mailbox 1 becomes empty
- The hardware sets REQCFLG2 bit of the register CAN_TXSTS to 1, and the transmit mailbox 2 becomes empty

Events generating FIFO0 interrupt:

- Set the FMNUM0[1:0] bit of the register CAN_RXF0 to a number rather than 0 by hardware, and FIFO0 will receive a new message
- Set the FFULLFLG0 bit of the register CAN_RXF0 to 1 by hardware, and FIFO0 will be full
- Set the FOVRFLG0 bit of the register CAN_RXF0 to 1 by hardware and FIFO0 will overrun

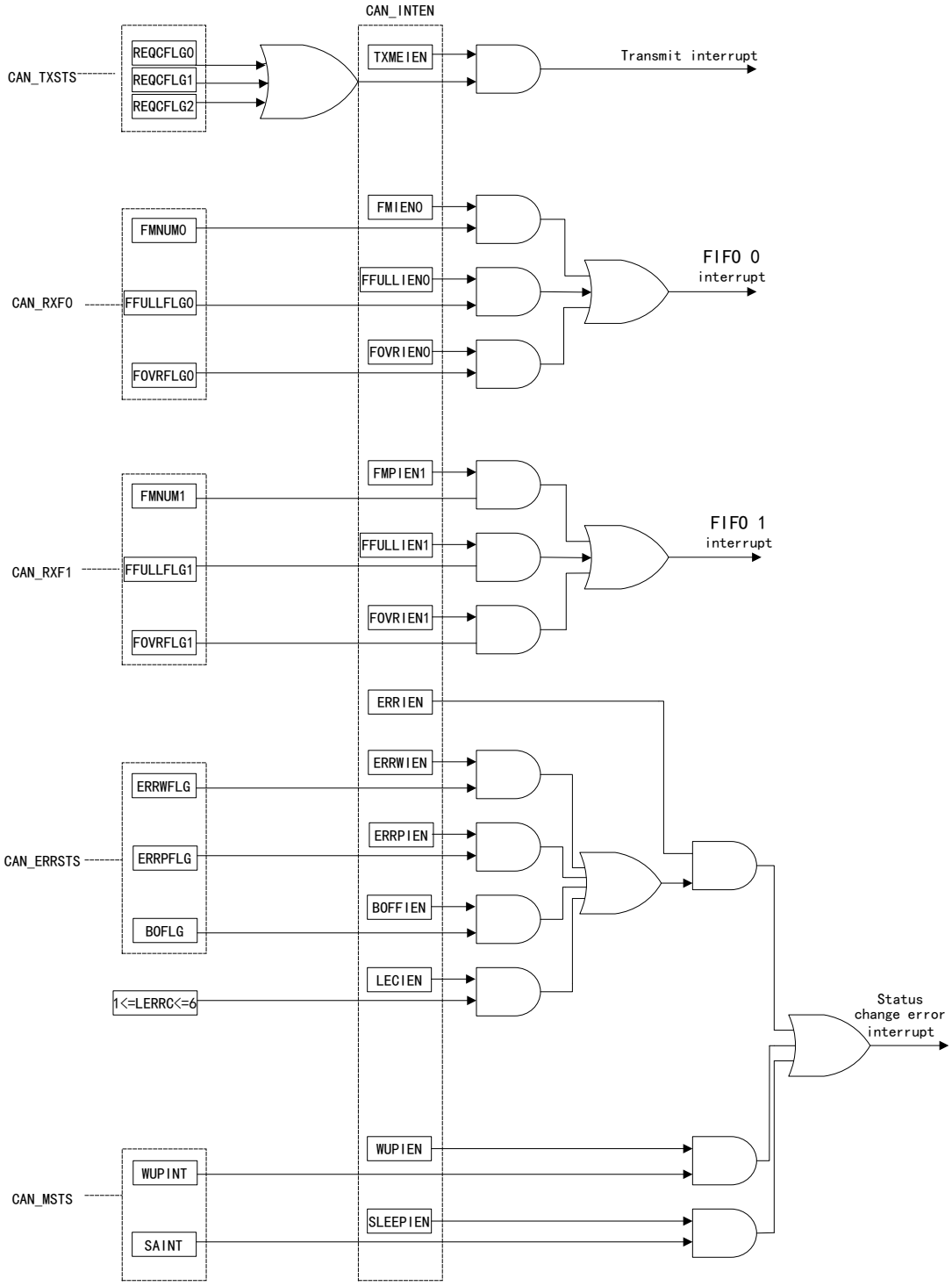
Events generating FIFO1 interrupt:

- Set the FMNUM1[1:0] bit of the register CAN_RXF1 to a number rather than 0 by hardware, and FIFO1 will receive a new message
- Set the FFULLFLG1 bit of the register CAN_RXF1 to 1 by hardware, and FIFO1 will be full
- Set the FOVRFLG1 bit of the register CAN_RXF1 to 1 by hardware and FIFO1 will overrun

Events generating state change and error interrupt:

- Set the SLEEPIEN bit of the register CAN_INTEN to 1 by hardware and it will enter the sleep mode
- Set the WUPIEN bit of the register CAN_INTEN to 1 by hardware and interrupt enable will wake up
- Set the ERRWFLG bit of the register CAN_ERRSTS to 1 by hardware, and it means that the number of errors has reached the threshold
- Set the ERRPFLG bit of the register CAN_ERRSTS to 1 by hardware, and it means that the number of errors has reached the threshold of passive error
- Set the LERRC[2:0] bit of the register CAN_ERRSTS by hardware, and it indicates the condition of last error

Figure 76 Event Flag and Interrupt Generation



20.5 Register address mapping

CAN1 base address: 0x4000_6400

Table 66 CAN Register Address Mapping

Register name	Description	Offset address
CAN_MCTRL	CAN main control register	0x00
CAN_MSTS	CAN main status register	0x04
CAN_TXSTS	CAN transmit status register	0x08
CAN_RXF0	CAN receive FIFO 0 register	0x0C
CAN_RXF1	CAN receive FIFO 1 register	0x10
CAN_INTEN	CAN interrupt enable register	0x14
CAN_ERRSTS	CAN error status register	0x18
CAN_BITTIM	CAN bit timing register	0x1C
CAN_TXMIDx	Each mailbox contains the transmit mailbox identifier register	0x180，0x190，0x1A0
CAN_TXDLENx	Transmit mailbox data length register	0x184，0x194，0x1A4
CAN_TXMDLx	Transmit mailbox low-byte data register	0x188，0x198，0x1A8
CAN_TXMDHx	Transmit mailbox high-byte data register	0x18C，0x19C，0x1AC
CAN_RXMIDx	Receive FIFO mailbox identifier register	0x1B0，0x1C0
CAN_RXDLENx	Receive FIFO mailbox data length register	0x1B4，0x1C4
CAN_RXMDLx	Receive FIFO mailbox low-byte data register	0x1B8，0x1C8
CAN_RXMDHx	Receive FIFO mailbox high-byte data register	0x1BC，0x1CC
CAN_FCTRL	CAN filter main control register	0x200
CAN_FMCFG	CAN filter mode register	0x204
CAN_FSCFG	CAN filter bit width register	0x20C
CAN_FFASS	CAN filter FIFO association register	0x214
CAN_FACT	CAN filter activation register	0x21C
CAN_FiBANKx	CAN filter group i register x	0x240..0x2AC

20.6 Register functional description

20.6.1 CAN control and status register

20.6.1.1 CAN main control register (CAN_MCTRL)

Offset address: 0x00

Reset value: 0x0001 0002

Field	Name	R/W	Description
0	INITREQ	R/W	Request to Enter Initialization Mode 0: Enter the normal work mode from the initialization mode 1: Enter the initialization mode from the normal work mode

Field	Name	R/W	Description
1	SLEEPREQ	R/W	Request to Enter Sleep Mode 0: Exit the sleep mode 1: Request to enter the sleep mode. If the AWUPCFG bit is set to 1, when the RX signal detects CAN message, this bit will be cleared to 0 by hardware; after reset, reset this bit to 1, and enter the sleep mode.
2	TXFPCFG	R/W	Transmit FIFO Priority Configure This bit is used to determine which parameters determine the transmission priority when multiple messages are waiting for transmission. 0: Determine by the message identifier 1: Determine by the sequence of transmission request
3	RXFLOCK	R/W	Receive FIFO Locked Mode Configure This bit is used to determine whether FIFO is locked when receiving overrun, and how to deal with the next received message when the message of the receive FIFO has not been read out. 0: Unlocked; if the message of the receive FIFO is not read out, the next received message will overwrite the original message 1: Locked; when the message of the receive FIFO is not read out, the next received message will be discarded
4	ARTXMD	R/W	Automatic Retransmission Message Disable 0: Enable automatic retransmission, and the message will be retransmitted automatically until it is transmitted successfully 1: Disable automatic retransmission and the message will be transmitted only once
5	AWUPCFG	R/W	Automatic Wakeup Mode Configure 0: Software wakes up the sleep mode by clearing the SMREQ bit of the CAN_MCTRL register 1: Hardware wakes up the sleep mode by detecting CAN message
6	ALBOFFM	R/W	Automatic Leaving Bus-Off Status Condition Management 0: After the software resets the INITREQ bit of the CAN_MCTRL register to 1 and then clears it, when the hardware detects 11 continuous recessive bits for 128 times, it will exit from the bus-off state 1: When the hardware detects 11 continuous recessive bits for 128 times, it will exit from the bus-off state automatically
14:7	Reserved		
15	SWRST	R/S	Software Reset CAN 0: Work normally 1: CAN is reset by force, and after reset, CAN enters the sleep mode; the hardware will clear this bit to 0 automatically
16	DBGFRZE	R/W	Debug Freeze 0: Invalid 1: During debugging, CAN cannot receive/transmit, but it still can read and write and control the receive FIFO normally
31:17	Reserved		

20.6.1.2 CAN main status register (CAN_MSTS)

Offset address: 0x04

Reset value: 0x0000 0C02

Field	Name	R/W	Description
0	INITFLG	R	Being Initialization Mode Flag This bit is set to 1 or cleared to 0 by hardware. 1: Exit the initialization mode 1: Being in the initialization mode; this bit is confirmation for initialization request bit of the CAN_MCTRL register.
1	SLEEPFLG	R	Being Sleep Mode Flag This bit is set to 1 or cleared to 0 by hardware 1: Exit the sleep mode 1: Being in the sleep mode; this bit is confirmation for sleep mode request bit of the CAN_MCTRL register.
2	ERRIFLG	RC_W1	Error Interrupt Occur Flag This bit is set to 1 by hardware and cleared to 0 by writing 1 by software. 0: Not occur 1: Occurred
3	WUPIFLG	RC_W1	Wakeup Interrupt Occur Flag When entering the sleep mode and detecting SOP wake-up, the bit is set to 1 by hardware; clear 0 by writing 1 by software. 0: Fail to wake up from the sleep mode 1: Wake up from the sleep mode
4	SLEEPIFLG	RC_W1	Being Sleep Mode Interrupt Flag When entering the sleep mode, this bit is set to 1 by hardware and corresponding interrupt will be triggered; when exiting the sleep mode, this bit is cleared to 0 by hardware and cleared to 0 by writing 1 by software. 0: Fail to enter the sleep mode 1: Enter the sleep mode
7:5	Reserved		
8	TXMFLG	R	Being Transmit Mode Flag 0: CAN is not in transmission mode 1: CAN is in transmission mode
9	RXMFLG	R	Being Receive Mode Flag 0: CAN is not in receiving mode 1: CAN is in receiving mode
10	LSAMVALUE	R	CAN Rx Pin Last Sample Value
11	RXSIGL	R	CAN Rx Pin Signal Level
31:12	Reserved		

20.6.1.3 CAN transmitting status register (CAN_TXSTS)

Offset address: 0x08

Reset value: 0x1C00 0000

Field	Name	R/W	Description
0	REQCFLG0	RC_W1	Mailbox 0 Request Completed Flag When the last transmission or abortion request of mailbox 0 is completed, this bit is set to 1 by hardware; when receiving the transmission request, this bit is cleared to 0 by hardware; it can be cleared to 0 by writing 1 by software. 0: Being transmitted 1: Transmission completed
1	TXSUSFLG0	RC_W1	Mailbox 0 Transmission Success Flag When mailbox 0 attempts to transmit successfully, this bit is set to 1 by hardware; it can be cleared to 0 by writing 1 by software. 0: Last transmission attempt failed 1: Last transmission attempt succeeded
2	ARBLSTFLG0	RC_W1	Mailbox 0 Arbitration Lost Flag When the mailbox 0 loses arbitration, this bit is set to 1 by hardware; this bit can be cleared to 0 by writing 1 by software. 0: Meaningless 1: Lost
3	TXERRFLG0	RC_W1	Mailbox 0 Transmission Error Flag When mailbox 0 fails to transmit, this bit is set to 1 by hardware; this bit can be cleared to 0 by writing 1 by software. 0: Meaningless 1: Failed to transmit
6:4	Reserved		
7	ABREQFLG0	R/S	Mailbox 0 Abort Request Flag If there is no message waiting for transmission in mailbox 0, this bit is invalid. 0: The transmitting message of mailbox 0 is cleared, and this bit is cleared to 0 by hardware 1: Set this bit to 1 to abort the transmission request of mailbox 0
8	REQCFLG1	RC_W1	Mailbox 1 Request Completed Flag When the last transmission or abortion request of mailbox 1 is completed, this bit is set to 1 by hardware; when receiving the transmission request, this bit is cleared to 0 by hardware; it can be cleared to 0 by writing 1 by software. 0: Being transmitted 1: Transmission completed
9	TXSUSFLG1	RC_W1	Mailbox 1 Transmission Success Flag When mailbox 1 attempts to transmit successfully, this bit is set to 1 by hardware; and cleared to 0 by writing 1 by software. 0: Last transmission attempt failed 1: Last transmission attempt succeeded
10	ARBLSTFLG1	RC_W1	Mailbox 1 Arbitration Lost Flag When the mailbox 1 loses arbitration, this bit is set to 1 by hardware; and cleared to 0 by writing 1 by software. 0: Meaningless 1: Lost

Field	Name	R/W	Description
11	TXERRFLG1	RC_W1	Mailbox 1 Transmission Error Flag When mailbox 1 fails to transmit, this bit is set to 1 by hardware; and cleared to 0 by writing 1 by software. 0: Meaningless 1: Failed to transmit
14:12	Reserved		
15	ABREQFLG1	R/S	Mailbox 1 Abort Request Flag If there is no message waiting for transmitting in mailbox 1, this bit is invalid. 0: The transmitting message of mailbox 1 is cleared, and this bit is cleared to 0 by hardware 1: Set this bit to 1 to abort the transmission request of mailbox 1
16	REQCFLG2	RC_W1	Mailbox 2 Request Completed Flag When the last transmission or abortion request of mailbox 2 is completed, this bit is set to 1 by hardware; when receiving the transmission request, this bit is cleared to 0 by hardware; it can be cleared to 0 by writing 1 by software. 0: Being transmitted 1: Transmission completed
17	TXSUSFLG2	RC_W1	Mailbox 2 Transmission Success Flag When mailbox 2 attempts to transmit successfully, this bit is set to 1 by hardware; it can be cleared to 0 by writing 1 by software. 0: Last transmission attempt failed 1: Last transmission attempt succeeded
18	ARBLSTFLG2	RC_W1	Mailbox 2 Arbitration Lost Flag When the mailbox 2 loses arbitration, this bit is set to 1 by hardware; it can be cleared to 0 by writing 1 by software. 0: Meaningless 1: Lost
19	TXERRFLG2	RC_W1	Mailbox 2 Transmission Error Flag When mailbox 2 fails to transmit, this bit is set to 1 by hardware; it can be cleared to 0 by writing 1 by software. 0: Meaningless 1: Failed to transmit
22:20	Reserved		
23	ABREQFLG2	R/S	Mailbox 2 Abort Request Flag If there is no message waiting for transmitting in mailbox 2, this bit is invalid. 0: The transmitting message of mailbox 2 is cleared, and this bit is cleared to 0 by hardware 1: Set this bit to 1 to abort the transmission request of mailbox 2
25:24	EMNUM[1:0]	R	Empty Mailbox Number This bit is applicable when there is empty mailbox. When all the transmit mailboxes are empty, it means the number of the transmit mailbox with the lowest priority; when the mailbox is not empty but not all empty, it means the number of next mailbox to be transmitted.

Field	Name	R/W	Description
26	TXMEFLG0	R	Transmit Mailbox 0 Empty Flag When the transmit mailbox 0 is empty, this bit is set to 1 by hardware. 0: There is message to be transmitted in mailbox 0 1: There is no message to be transmitted in mailbox 0
27	TXMEFLG1	R	Transmit Mailbox 1 Empty Flag When the transmit mailbox 1 is empty, this bit is set to 1 by hardware. 0: There is message to be transmitted in mailbox 1 1: There is no message to be transmitted in mailbox 1
28	TXMEFLG2	R	Transmit Mailbox 2 Empty Flag When the transmit mailbox 2 is empty, this bit is set to 1 by hardware. 0: There is message to be transmitted in mailbox 2 1: There is no message to be transmitted in mailbox 2
29	LOWESTP0	R	The Lowest Transmission Priority Flag For Mailbox 0 0: Meaningless 1: The priority of mailbox 0 is the lowest among those mailboxes waiting to transmit messages Note: If there is only one mailbox waiting, LOWESTP[2:0] is cleared to 0.
30	LOWESTP1	R	The Lowest Transmission Priority Flag For Mailbox 1 0: Meaningless 1: The priority of mailbox 1 is the lowest among those mailboxes waiting to transmit messages
31	LOWESTP2	R	The Lowest Transmission Priority Flag For Mailbox 2 0: Meaningless 1: The priority of mailbox 2 is the lowest among those mailboxes waiting to transmit messages

20.6.1.4 CAN receive FIFO 0 register (CAN_RXF0)

Offset address: 0x0C

Reset value: 0x00

Field	Name	R/W	Description
1:0	FMNUM0[1:0]	R	The number of Message in receive FIFO0 These bits are used to reflect the number of messages stored in current receive FIFO0. Every time a new message is received, add 1 to FMNUM0 bit; every time the mailbox message is released and output, subtract 1 from FMNUM0 bit.
2	Reserved		
3	FFULLFLG0	RC_W1	Receive FIFO0 Full Flag When there are three messages in FIFO0, it means the FIFO0 has been full; this bit is set to 1 by hardware and cleared to 0 by writing 1 by software. 0: Not full 1: Full

Field	Name	R/W	Description
4	FOVRFLG0	RC_W1	Receive FIFO 0 Overrun Flag When there are three messages in FIFO0 and then a new message is received, it means the FIFO0 overruns; this bit is set to 1 by hardware and cleared to 0 by writing 1 by software. 0: No overrun 1: Generate overrun
5	RFOM0	R/S	Release Receive FIFO0 Output Mailbox to Receive Massage This bit is set to 1 by hardware and cleared to 0 by software. If there is no message in FIFO, this bit is invalid. When FIFO contains more than two messages, the output mailbox must be first released to access the second message. 0: Meaningless 1: Release the output mailbox of receive FIFO0
31:6	Reserved		

20.6.1.5 CAN receive FIFO 1 register (CAN_RXF1)

Offset address: 0x10

Reset value: 0x00

Field	Name	R/W	Description
1:0	FMNUM1[1:0]	R	The number of Message in receive FIFO1 These bits are used to reflect the number of messages stored in current receive FIFO1. Every time a new message is received, add 1 to FMNUM1 bit; every time the mailbox message is released and output, subtract 1 from FMNUM1 bit.
2	Reserved		
3	FFULLFLG1	RC_W1	Receive FIFO0 Full Flag When there are three messages in FIFO1, it means the FIFO1 has been full; this bit is set to 1 by hardware and cleared to 0 by writing 1 by software. 0: Not full 1: Full
4	FOVRFLG1	RC_W1	Receive FIFO1 Overrun Flag When there are three messages in FIFO1 and then a new message is received, it means the FIFO1 overruns; this bit is set to 1 by hardware and cleared to 0 by writing 1 by software. 0: No overrun 1: Generate overrun
5	RFOM1	R/S	Release Receive FIFO1 Output Mailbox to Receive Massage This bit is set to 1 by hardware and cleared to 0 by software. If there is no message in FIFO, this bit is invalid. When FIFO contains more than two messages, the output mailbox must be first released to access the second message. 0: Meaningless 1: Release the output mailbox of receive FIFO1
31:6	Reserved		

20.6.1.6 CAN interrupt enable register (CAN_INTEN)

Offset address: 0x14

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	TXMEIEN	R/W	Transmit Mailbox Empty Interrupt Enable When REQCFLGx bit is set to 1, it means transmission has been completed, and the transmit mailbox is empty; if this bit is set to 1, an interrupt will be generated. 0: No interrupt is generated 1: Generate interrupt
1	FMIEN0	R/W	Generate an interrupt when enabling the number of messages in FIFO0 to be not 0 （ Interrupt Enable When The Number Of FIFO0 Message Is Not 0） When FMNUM0[1:0] bit of FIFO 0 is not zero, it means that the number of messages in FIFO0 is not 0; if this bit is set to 1, an interrupt will be generated. 0: No interrupt is generated 1: Generate interrupt
2	FFULLIEN0	R/W	FIFO0 Full Interrupt Enable When the FFULLFLG0 bit of FIFO0 is set to 1, it means that the message of FIFO0 is full; if this bit is set to 1, an interrupt will be generated. 0: No interrupt is generated 1: Generate interrupt
3	FOVRIEN0	R/W	FIFO0 Overrun Interrupt Enable When the FOVRFLG0 bit of FIFO0 is set to 1, it means that the FIFO0 has overrun; if this bit is set to 1, an interrupt will be generated. 0: No interrupt is generated 1: Generate interrupt
4	FMPIEN1	R/W	Generate an interrupt when enabling the number of messages in FIFO1 to be not 0 （I Interrupt Enable　wnhen the number of FIFO1 Message is not 0） When FMNUM1[1:0] bit of FIFO 1 is not zero, it means that the number of messages in FIFO1 is not 0; if this bit is set to 1, an interrupt will be generated. 0: No interrupt is generated 1: Generate interrupt
5	FFULLIEN1	R/W	FIFO1 Full Interrupt Enable When the FFULLFLG1 bit of FIFO1 is set to 1, it means that the message of FIFO1 is full; if this bit is set to 1, an interrupt will be generated. 0: No interrupt is generated 1: Generate interrupt

Field	Name	R/W	Description
6	FOVRIEN1	R/W	FIFO1 Overrun Interrupt Enable When the FOVRFLG1 bit of FIFO1 is set to 1, it means that the FIFO1 has overrun; if this bit is set to 1, an interrupt will be generated. 0: No interrupt is generated 1: Generate interrupt
7	Reserved		
8	ERRWIEN	R/W	Error Warning Interrupt Enable When ERRWFLG bit is set to 1, an error warning will occur; if this bit is set to 1, ERRIFLG shall be set and a warning error interrupt will be generated. 0: ERRIFLG bit is not set 1: ERRIFLG bit is set to 1
9	ERRPIEN	R/W	Error Passive Interrupt Enable When ERRPFLG bit is set to 1, a passive error will occur; if this bit is set to 1, ERRIFLG shall be set and a passive error interrupt will be generated. 0: ERRIFLG bit is not set 1: ERRIFLG bit is set to 1
10	BOFFIEN	R/W	Bus-Off Interrupt Enable When BOFFFLG bit is set to 1, bus-off will occur; if this bit is set to 1, ERRIFLG shall be set and a bus-off interrupt will be generated. 0: ERRIFLG bit is not set 1: ERRIFLG bit is set to 1
11	LECIEN	R/W	Last Error Code Interrupt Enable When an error is detected and the LERRC[2:0] is set by hardware, the last error code is recorded. If this bit set to 1, the ERRIFLG is set to generate the last error interrupt. 0: ERRIFLG bit is not set 1: ERRIFLG bit is set to 1
14:12	Reserved		
15	ERRIEN	R/W	Error interrupt Enable When the corresponding error status register is set to 1, if this bit is set to 1, an error interrupt will be generated. 0: No interrupt is generated 1: Generate interrupt
16	WUPIEN	R/W	Wakeup interrupt Enable When WUPINT bit is set to 1, if this bit is set to 1, a wake-up interrupt will be generated. 0: No interrupt is generated 1: Generate interrupt
17	SLEEPIEN	R/W	Sleep Interrupt Enable When SLEEPIFLG bit is set to 1, if this bit is set to 1, a sleep interrupt will be generated. 0: No interrupt is generated 1: Generate interrupt

Field	Name	R/W	Description
31:18	Reserved		

20.6.1.7 CAN error status register (CAN_ERRSTS)

Offset address: 0x18

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	ERRWFLG	R	Error Warning Occur Flag When the value of the receiving error counter or transmitting error counter ≥96, this bit is set to 1 by hardware. 0: No error warning occurred 1: Error warning occurred
1	ERRPFLG	R	Error Passive Occur Flag When the value of the receiving error counter or transmitting error counter ≥127, this bit is set to 1 by hardware. 0: No passive error occurred 1: Passive error occurred
2	BOFLG	R	Enter Bus-Off Flag When the value of the transmitting error counter TXERRCNT is greater than 255, CAN will enter the bus-off state and this bit is set to 1 by hardware. 0: CAN not in bus-off state 1: CAN in bus-off state
3	Reserved		
6:4	LERRC	R/W	Record Last Error Code When the error on CAN bus is detected, it is set by hardware according to the error category; when the message is transmitted or received correctly, this bit is cleared to 0 by hardware. 000: No error 001: Bit stuffing error 010: Form (Form) error 011: Acknowledgment (ACK) error 100: Recessive bit error 101: Dominant bit error 110: CRC error 111: Set by software
15:7	Reserved		
23:16	TXERRCNT	R	Least Significant Byte Of The 9-Bit Transmit Error Counter The counter is implemented according to the transmission part of fault definition mechanism of CAN protocol.
31:24	RXERRCNT	R	Receive Error Counter The receiving error counter is implemented according to the receiving part of fault definition mechanism of CAN protocol. When a receiving error occurs, according to the condition of the error, add 1 or 8 to the counter, and subtract 1 after receiving successfully. When the value of the counter is greater than 127, set the counter value to 120.

20.6.1.8 CAN bit timing register (CAN_BITTIM)

Offset address: 0x1C

Reset value: 0x0123 0000

Field	Name	R/W	Description
9:0	BRPSC	R/W	Baud Rate Prescaler Factor Setup Time unit t_q =(BRPSC+1)× t_{PCLK}
15:10	Reserved		
19:16	TIMSEG1	R/W	Set the time segment 1 (Time Segment 1 Setup) Time occupied by time period 1 t_{BS1}= t_{CAN} x (TIMSEG1+1).
22:20	TIMSEG2	R/W	Time Segment 2 Setup Time occupied by time period 2 t_{BS2} = t_{CAN} x (TIMSEG2+1).
23	Reserved		
25:24	RSYNJW	R/W	Resynchronization Jump Width Time that CAN hardware can extend or shorten in this bit: t_{RJW}=t_{CAN} x(RSYNJW+1).
29:26	Reserved		
30	LBKMEN	R/W	Loop Back Mode Enable 0: Disable 1: Enable
31	SILMEN	R/W	Silent Mode Enable 0: Normal state 1: Mute mode

Note: When CAN is in initialization mode, this register can be accessed only by software

20.6.2 CAN mailbox register

This section describes the transmit and receive mailbox registers.

The transmit and receive mailboxes are almost the same except the following examples:

- FMIDX domain of CAN_RXDLENx register;
- The receive mailbox is read-only;
- The transmit mailbox is writable only when it is empty, and if the corresponding TXMEFLG bit of CAN_TXSTS register is 1, it means the transmit mailbox is empty.

There are three transmit mailboxes and two receive mailboxes in total. Each receive mailbox is FIFO with level-3 depth, and can only access the message that is received first in FIFO.

20.6.2.1 Transmit mailbox identifier register (CAN_TXMIDx) (x=0..2)

Offset address: 0x180, 0x190, 0x1A0

Reset value: 0xXXXX XXXX, X=undefined bit (except Bit 0, TXMREQ=0 after reset)

Field	Name	R/W	Description
0	TXMREQ	R/W	Transmit Mailbox Data Request 0: When the data in the mailbox is transmitted, the mailbox is empty and this bit is cleared to 0 by hardware 1: Software writes 1, to enable request to transmit mailbox data
1	TXRFREQ	R/W	Transmit Remote Frame Request 0: Data frame 1: Remote frame
2	IDTYPESEL	R/W	Identifier Type Select 0: Standard identifier 1: Extended identifier
20:3	EXTID[17:0]	R/W	Extended Identifier Setup Low byte of extended identifier label.
31:21	STDID[10:0]/EXTID[28:18]	R/W	Standard Identifier Or Extended Identifier According to the content of IDTYPESEL bit, these bits are standard identifier STDID[10:0] and high byte EXTID[28:18] of extended identifier.

Note: 1. When its mailbox is in the state of waiting for transmission, this register is write-protected

2. This register realizes transmission request control function (No. 0 bit) - the reset value is 0

20.6.2.2 Transmit mailbox data length register (CAN_TXDLENx) (x=0..2)

When the mailbox is not idle, all bits of this register are write-protected.

Offset address: 0x184, 0x194, 0x1A4

Reset value: 0xXXXX XXXX

Field	Name	R/W	Description
3:0	DLCODE	R/W	Transmit Data Length Code Setup
31:4	Reserved		

20.6.2.3 Transmit mailbox low-byte data register (CAN_TXMDLx) (x=0..2)

When the mailbox is not idle, all bits of this register are write-protected, and the message contains 0 to 7-byte data and starts from the byte 0.

Offset address: 0x188, 0x198, 0x1A8

Reset value: 0xXXXX XXXX

Field	Name	R/W	Description
7:0	DATABYTE0	R/W	Data Byte 0 of the Message
15:8	DATABYTE1	R/W	Data Byte 1 of the Message
23:16	DATABYTE2	R/W	Data Byte 2 of the Message
31:24	DATABYTE3	R/W	Data Byte 3 of the Message

20.6.2.4 Transmit mailbox high-byte data register (CAN_TXMDHx) (x=0..2)

When the mailbox is not idle, all bits of this register are write-protected.

Offset address: 0x18C, 0x19C, 0x1AC

Reset value: 0xXXXX XXXX, X=undefined bit

Field	Name	R/W	Description
7:0	DATABYTE4	R/W	Data Byte 4 of the Message
15:8	DATABYTE5	R/W	Data Byte 5 of the Message
23:16	DATABYTE6	R/W	Data Byte 6 of the Message
31:24	DATABYTE7	R/W	Data Byte 7 of the Message

20.6.2.5 Receive FIFO mailbox identifier register (CAN_RXMIDx) (x=0..1)

Offset address: 0x1B0, 0x1C0

Reset value: 0xXXXX XXXX, X=undefined bit

Field	Name	R/W	Description
0	Reserved		
1	RFTXREQ	R	Remote Frame Transmission Request 0: Data frame 1: Remote frame
2	IDTYPESEL	R	Identifier Type Select 0: Standard identifier 1: Extended identifier
20:3	EXTID[17:0]	R	Extended Identifier Setup Low byte of extended identifier.
31:21	STDID[10:0]/EXTID[28:18]	R	Standard Identifier Or Extended Identifier According to the content of IDTYPESEL bit, these bits are standard identifier STDID[10:0] and high byte EXTID[28:18] of extended identifier.

Notes: All receive mailbox registers are read-only.

20.6.2.6 Receive FIFO mailbox data length register (CAN_RXDLENx) (x=0..1)

Offset address: 0x1B4, 0x1C4

Reset value: 0xXXXXX XXXX

Field	Name	R/W	Description
3:0	DLCODE	R	Receive Data Length Code Setup This bit represents the data length in the frame; for remote frame, DLCODE is constantly 0.
7:4	Reserved		
15:8	FMIDX	R	Filter Match Index Setup
31:16	Reserved		

Notes: All receive mailbox registers are read-only.

20.6.2.7 Receive FIFO mailbox low-byte data register (CAN_RXMDLx) (x=0..1)

Offset address: 0x1B8, 0x1C8; the message contains 0 to 8-byte data, which starts from the byte 0.

Reset value: 0xXXXXX XXXX

Field	Name	R/W	Description
7:0	DATABYTE0	R	Data Byte 0 of the Message
15:8	DATABYTE1	R	Data Byte 0 of the Message
23:16	DATABYTE2	R	Data Byte 0 of the Message
31:24	DATABYTE3	R	Data Byte 0 of the Message

Notes: All receive mailbox registers are read-only.

20.6.2.8 Receive FIFO mailbox high-byte data register (CAN_RXMDHx) (x=0..1)

Offset address: 0x1BC, 0x1CC

Reset value: 0xXXXX XXXX, X=undefined bit

Field	Name	R/W	Description
7:0	DATABYTE4	R	Data Byte 0 of the Message
15:8	DATABYTE5	R	Data Byte 0 of the Message
23:16	DATABYTE6	R	Data Byte 0 of the Message
31:24	DATABYTE7	R	Data Byte 0 of the Message

Notes: All receive mailbox registers are read-only.

20.6.3 CAN filter register

20.6.3.1 CAN filter control register (CAN_FCTRL)

Offset address: 0x200

Reset value: 0x2A1C 0E01

Field	Name	R/W	Description
0	FINITEN	R/W	Filter Init Mode Enable 0: Normal mode 1: Initialization mode
31:1	Reserved		

Notes: The non-reserved bit of this register is completely controlled by software.

20.6.3.2 CAN filter mode configuration register (CAN_FMCFG)

Offset addres: 0x204

Reset value: 0x0000 0000

Field	Name	R/W	Description
13:0	FMCFGx	R/W	Filter Mode Configure The value of x is within 0-13. 0: Identifier mask bit mode 1: Identifier list mode
31:14	Reserved		

Note: Only when CAN_FCTRL (FINITEN =1) is set to put the filter in initialization mode, can this register be written.

20.6.3.3 CAN filter bit width configuration register (CAN_FSCFG)

Offset address: 0x20C

Reset value: 0x0000 0000

Field	Name	R/W	Description
13:0	FSCFGx	R/W	Filterx Scale Configure The value of x is within 0-13. 0: 2 16 bits 1: Single 32 bits
31:14	Reserved		

Note: Only when CAN_FCTRL (FINITEN =1) is set to make the filter in initialization mode, can this register be written.

20.6.3.4 CAN filter FIFO association register (CAN_FFASS)

Offset address: 0x214

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	FFASS0	R/W	Configure Filter0 Associated with FIFO 0: The filter is associated with FIFO0 1: The filter is associated with FIFO1
1	FFASS1	R/W	Configure Filter1 Associated with FIFO Refer to FFASS0 for specific description.
2	FFASS2	R/W	Configure Filter2 Associated with FIFO Refer to FFASS0 for specific description.
3	FFASS3	R/W	Configure Filter3 Associated with FIFO Refer to FFASS0 for specific description.
4	FFASS4	R/W	Configure Filter4 Associated with FIFO Refer to FFASS0 for specific description.
5	FFASS5	R/W	Configure Filter5 Associated with FIFO Refer to FFASS0 for specific description.
6	FFASS6	R/W	Configure Filter6 Associated with FIFO Refer to FFASS0 for specific description.
7	FFASS7	R/W	Configure Filter7 Associated with FIFO Refer to FFASS0 for specific description.
8	FFASS8	R/W	Configure Filter8 Associated with FIFO Refer to FFASS0 for specific description.
9	FFASS9	R/W	Configure Filter9 Associated with FIFO Refer to FFASS0 for specific description.
10	FFASS10	R/W	Configure Filter10 Associated with FIFO Refer to FFASS0 for specific description.
11	FFASS11	R/W	Configure Filter11 Associated with FIFO Refer to FFASS0 for specific description.
12	FFASS12	R/W	Configure Filter12 Associated with FIFO Refer to FFASS0 for specific description.
13	FFASS13	R/W	Configure Filter13 Associated with FIFO Refer to FFASS0 for specific description.
31:14	Reserved		

Notes: Only when CAN_FCTRL (FINITEN =1) is set to put the filter in initialization mode, can this register be written.

20.6.3.5 CAN filter activation register (CAN_FACT)

Offset address: 0x21C

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	FACT0	R/W	Filter0 Active 0: Disable 1: Active
1	FACT1	R/W	Filter1 Active Refer to FACT0 for specific description
2	FACT2	R/W	Filter3 Active Refer to FACT0 for specific description
3	FACT3	R/W	Filte3 Active Refer to FACT0 for specific description
4	FACT4	R/W	Filter4 Active Refer to FACT0 for specific description
5	FACT5	R/W	Filter5 Active Refer to FACT0 for specific description
6	FACT6	R/W	Filte6 Active Refer to FACT0 for specific description
7	FACT7	R/W	Filter7 Active Refer to FACT0 for specific description
8	FACT8	R/W	Filter8 Active Refer to FACT0 for specific description
9	FACT9	R/W	Filter9 Active Refer to FACT0 for specific description
10	FACT10	R/W	Filter10 Active Refer to FACT0 for specific description
11	FACT11	R/W	Filter11 Active Refer to FACT0 for specific description
12	FACT12	R/W	Filter12 Active Refer to FACT0 for specific description
13	FACT13	R/W	Filter13 Active Refer to FACT0 for specific description
31:14	Reserved		

20.6.3.6 Register x of CAN filter group x (CAN_FiBANKx) (i＝0..13，x=1..2)

Offset address: 0x240..0x2AC

Reset value: 0xXXXX XXXX

Field	Name	R/W	Description
31:0	FBIT[31:0]	R/W	Filter Bits Setup Identifier list mode: 0: FBITx bit is dominant bit 1: FBITx bit is recessive bit Identifier mask bit mode: 0: FBITx is not used for comparison 1: FBITx must match Note: The value of x is 0~31, indicating the bit number of FBIT.

Note: There are 14 sets of filters in 错误!未提供文档变量。product: i=0..13. Each set of filters consists of two 32-bit registers and CAN_FiBANK[2:1]. The corresponding filter registers can be modified only when the corresponding FACTx bit of CAN_FACT register is cleared to 0 or the FINITEN bit of CAN_FCTRL register is 1.

21 Universal synchronous/asynchronous transceiver (USART)

21.1 Full name and abbreviation of terms

Table 67 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

21.2 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 receiving and transmitting mode, but also supports synchronous one-way communication and 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.

21.3 Main characteristics

（1） Full-duplex asynchronous communication

（2） Single-line half-duplex communication

（3） NRZ standard format

（4） Characteristics of programmable serial port:

- Data bit: 7 bits, 8 bits or 9 bits
- Check bits: Even parity check, odd parity check, no check
- Support 0.5, 1, 1.5 and 2 stop bits

（5） Check control

- Transmit the check bit
- Check the received data

（6） Select speed and clock tolerance with programmable 8 or 16-time oversampling rate

（7） Programmable high or low priority

（8） Independent transmitter and receiver enable bit

（9） Independent signal polarity control transmitter and receiver

（10） Switch TX/RX pin

（11） Support timeout detection

（12） Programmable baud rate generator, with the baud rate up to 6Mbits

（13） Automatic baud rate detection

（14） 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

（15） Double-clock drive

- Function of wake-up from the stop mode
- Baud rate selection independent of PCLK

（16） Synchronous transmission mode

（17） Generation and detection of LIN break frame

（18） Support smart card interface of ISO7816-3 standard

（19） Support IrDA protocol

（20） Support hardware flow control and RS485 drive enable

（21） DMA can be used for continuous communication

（22） Support ModBus communication

- Timeout detection
- CR/LF character recognition

（23） 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

（24） 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
- Address/Character match
- Wake up from the stop mode
- Failed to receive interrupt on time
- End of block interrupt

21.4 Functional description

Table 68 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 send in hardware flow control mode
USART_nCTS	Output	Clear to send in hardware flow control mode
USART_DE	Input	Drive enable activating external transmitter/receiver

21.4.1 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.

21.4.2 Frame format

The frame format of data frame is controlled by USART_CTRL1 register

- The character length is controlled by DBLCFG bit, and can be set to 7, 8 or 9 bits
- The PCEN bit controls whether to enable the check bit
- The PCFG bit controls the check bit to odd or even

Table 69 USART Frame Format

DBLCFG bit	PCEN bit	USART data frame
00	0	Start bit+8-bit data+stop bit
00	1	Start bit+7-bit data+parity check bit+stop bit
01	0	Start bit+9-bit data+stop bit
01	1	Start bit+8-bit data+ parity check bit+stop bit
10	0	Start bit+7-bit data+stop bit
10	1	Start bit+6-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: The default stop bit
- 0.5 stop bit: Used when receiving data in smart card mode.
- Two 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

Check 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 check: When the number of frame data and check bit '1' is even, the even check bit is 0; otherwise it is 1.
- Odd check: When the number of frame data and check bit '1' is even, the odd check bit is 1; otherwise it is 0.
- Check generation: When transmitting data, set PCEN bit of USART_CTRL1 register, and the check bit will replace the MSB bit of the data and be transmitted.
- Parity check:

- If the parity check fails, PEFLG flag bit of USART_STS register will be set.
- If the check control is enabled, corresponding interrupt will be triggered. Write 1 to PECLR bit of USART_INTFCLR register, and PEFLG flag bit can be cleared.

21.4.3 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.

21.4.3.1 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_TXDATA 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） Decide the word length by setting DBLCFG bit of USART_CTRL1 register

（2） Decide the number of stop bits by setting STOPCFG bit of USART_CTRL2 register

（3） If multi-buffer communication is selected, DMA should be enabled in USART_CTRL3 register

（4） Set the baud rate of communication in USART_BR register

（5） Set UEN bit of USART_CTRL1 register to enable USART. Wait for TXBEFLG bit of USART_STS register to be set to 1

（6） Enable TXEN bit in USART_CTRL1 register, and transmit an idle frame

（7） Write data to USART_TXDATA register (if DMA is not enabled, repeat step 7 for each byte to be transmitted)

（8） 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.

21.4.3.2 Single-byte communication

TXBEFLG bit can be cleared to 0 by writing to USART_TXDATA 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 cleared. 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 TXDATA 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_TXDATA register, before entering the low-power mode or before disabling the USART module, wait to set TXCFLG to 1.

21.4.3.3 Break frame

It is regarded that the break frames all receive '0' within one frame period. One break frame can be transmitted by setting TXBFQ bit of USART_REQUEST register, and the length of the break frame is determined by DBLCFG bit of USART_CTRL1 register. If the TXBFQ 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, this 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 TXBFQ bit is reset before transmission of the break frame, the break frame will not be transmitted. To transmit two consecutive break frames, the TXBFQ bit should be set after the stop bit of the previous break symbol.

21.4.3.4 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.

21.4.4 Receiver

21.4.4.1 Character receiving

During receiving period of USART, RX pin will first introduce the least significant bit of the data. In this mode, USART_RXDATA register has a buffer between the internal bus and the receive shift register. The data is transmitted to the buffer bit by bit. When fully receiving the data, the corresponding receive register is not empty, then the user can read USART_RXDATA.

Receiving configuration steps

（1） Program the oversampling rate to 8 or 16 times

（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） Set UEN bit of USART_CTRL1 register to enable USART

（5） If multi-buffer communication is selected, DMA should be enabled in USART_CTRL3 register

（6） Set the baud rate of communication in USART_BR register

（7） 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_RXDATA 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_RXDATA 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.

21.4.4.2 Break frame

When the receiver receives a break frame, USART will handle it as receiving a frame error.

21.4.4.3 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.

21.4.4.4 Select the clock source

The clock source must be selected by clock control system before USART is enabled

（1） The clock source shall be selected according to the transmission speed and the possibility of use of USART in low-power mode.

（2） The clock source frequency is f_{CK}.

- The range of communication speed is determined by the clock source. USART should be enabled before the clock source is selected.
- When USART adopts dual-clock domain or wakes up the stop mode, PCLK, LSECLK, HSICLK or SYSCLK can be used as the clock source; otherwise, the clock source is PCLK.
- If LSECLK and LSICLK are selected as the clock source, USART can receive data even in low-power mode. It can select according to the received data and wake-up mode, and wake up MCU when necessary, so that DMA can read the received data.
- The receiver realizes the data recovery of different oversampling technologies configured by users to distinguish valid incoming data and noises, which requires a trade-off between the maximum communication speed and noise/clock inaccuracy immunity.

21.4.4.5 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.

21.4.4.6 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 RXDATA 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

- OVREFLG bit of USART_STS is set to 1

- The data in RXDATA register will not be lost
- The data in the shift register previously received will be overwritten, but the data received later will not be saved
- If RXBNEIEN bit or ERRIEN bit of USART_CTRL1 is set, an interrupt will be generated
- When OVREFLG bit is set, it means there are data lost. There are two possibilities:
  - When RXBNEFLG=1, the previous valid data is still on RXDATA register, and can be read
  - When RXBNEFLG=0, there is no valid data in RXDATA register
- The OVREFLG bit can be reset by reading USART_STS and USART_RXDATA registers.

21.4.4.7 Noise error

When noise is detected in receiving process of the receiver:

- Set NEFLG flag on the rising edge of RXBNEFLG bit of USART_STS register
- Invalid data is transmitted from the shift register to USART_RXDATA register.
- In single- byte communication, no interrupt will be generated, but in multi-buffer communication, an interrupt will be generated by setting the ERRIEN bit of USART_CTRL3 register

Note: 8-time oversampling ratio cannot be used in LIN, smart card and IrDA modes.

21.4.4.8 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:

- Set the FEFLG bit of USART_STS register
- Invalid data is transmitted from the shift register to USART_RXDATA register.
- In single- byte communication, no interrupt will be generated, but in multi-buffer communication, an interrupt will be generated by setting the ERRIEN bit of USART_CTRL3 register

21.4.5 Tolerance of receiver to the change of clock

Only when the total clock system deviation is less than the tolerance of USART receiver, can the USART receiver work normally.

Deviation will occur in any of the following circumstances:

（1） DTRA: Deviation caused by transmitter error

（2） DQUANT: Deviation caused by receiver baud rate quantization

（3） DREC: Change of receiver oscillator

（4） DTCL: Deviation caused by transmission line

21.4.6 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=PCLK/16×(USARTDIV)

The system clock of USART2 is PCLK1, and that of USART1 is PCLK2. USART can be enabled only after the clock control unit enables the system clock.

21.4.7 Automatic baud rate detection

When a character is received, USART can detect and automatically set the value of the USART_BR register. Automatic baud rate detection functions when the communication speed of the system is unknown, the clock source with low precision is used, or the clock deviation is not measured to obtain the correct bit rate. The clock source must be compatible with the expected communication speed.

A non-zero baud rate must be written for initialization; confirm the character content, and then enable automatic baud rate detection. ABRDCFG bit of USART_CTRL2 register can be set to select the character content, and the possible character content is:

（1） For all characters starting with 1, in this case, measure the length of the start bit (the duration from the falling edge to the rising edge).

（2） For all characters starting with 10xx, in this case, measure the length of the start bit and the first data bit, the duration of the falling edge, to ensure better accuracy when the signal slew rate is small.

ABRDEN bit of USART_CTRL2 register determines whether to enable automatic baud rate detection. After the automatic baud rate detection is enabled, wait for the first character on RX line. After detection, ABRDFLG flag bit of USART_STS register will be set.

Note:

（1）If the line noise is too high, correct baud rate cannot be guaranteed. In this case, the BR value may be damaged and the ABRDEFLG flag bit will be set. This situation can also happen if the communication speed and automatic baud rate detection are not compatible.

（2）RXBNEFLG interrupt will be generated after detection.

（3）At any time, automatic baud rate detection may be restarted by resetting the ABRDFLG flag (writing a 0).

（4）USART cannot be disabled during automatic baud rate detection; otherwise, the BR value may be damaged.

21.4.8 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, the LINMEN bit of USART_CTRL2 register, and the SCEN bit, IREN bit and HDEN bit of USART_CTRL3 register are cleared to 0, any receive state bit will not be set, and all receive interrupts will be disabled.

When mute mode is enabled, there are two ways to exit the mute mode:

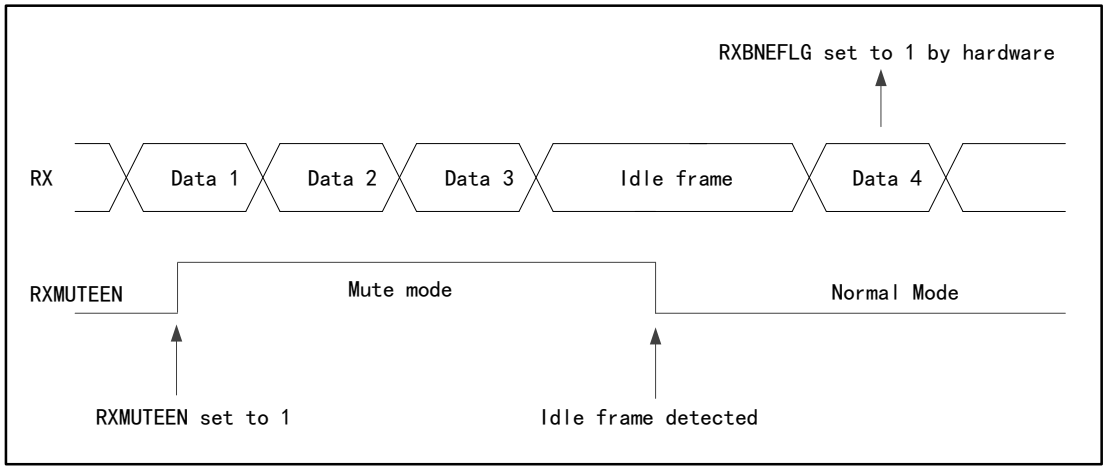
（1） Exit the mute mode when WUPMCFG bit is cleared and the bus is idle.

（2） Exit the mute mode when WUPMCFG bit is set and the address flag is received.

Idle bus detection (WUPMCFG=0)

When RXWFMUTE is set to 1, USART enters the mute mode, and it can wake up from the mute mode when an idle frame is detected, meanwhile, the RXWFMUTE bit will be cleared to 0 by hardware. RXWFMUTE can also be cleared to 0 by software.

Figure 77 Idle Bus Exit Mute Mode

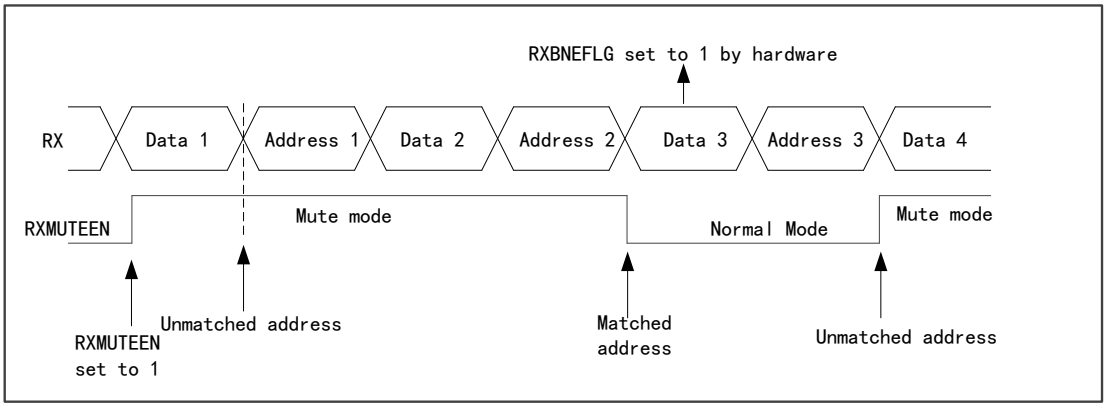


Address flag detection (WUPMCFG=1)

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 78 Address Flag Exit Mute Mode



21.4.9 Wake up from the stop mode

When USART uses HSICLK and LSECLK as clock source, USWMEN bit of USART_CTRL1 register decides whether to wake up from the stop mode. Before entering the stop mode, set USWMEN bit of USART_CTRL1 register, and when wake-up time is detected, set WSMFLG to 1, and at this time, an interrupt will be generated as long as WSMIEN is set.

Mute mode in stop mode

It is not allowed to exit from the mute mode during idle detection. If the system exits the mute mode by using the address matching, only the address matching event can be taken as its wake-up source. If the start bit is set to detect wake-up, WSMFLG will be set and RXBNEFLG flag bit will not.

21.4.10 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 the data frame have no clock output
- Whether the last data bit of the data frame generates USART_CK clock is determined by LBCPOEN bit of the register USART_CTRL2
- 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
- The external CK clock cannot be activated when the bus is idle or a break frame appears

Figure 79 USART Synchronous Transmission Example

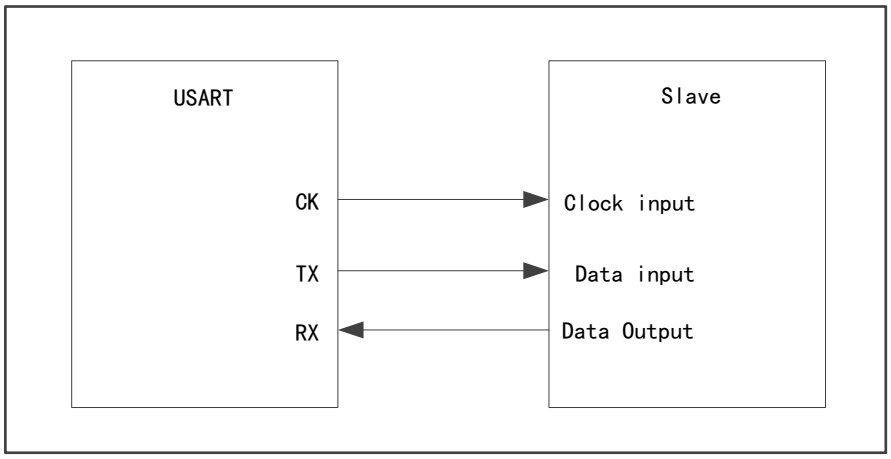


Figure 80 USART Synchronous Transmission Timing Diagram (DBLCFG=10)

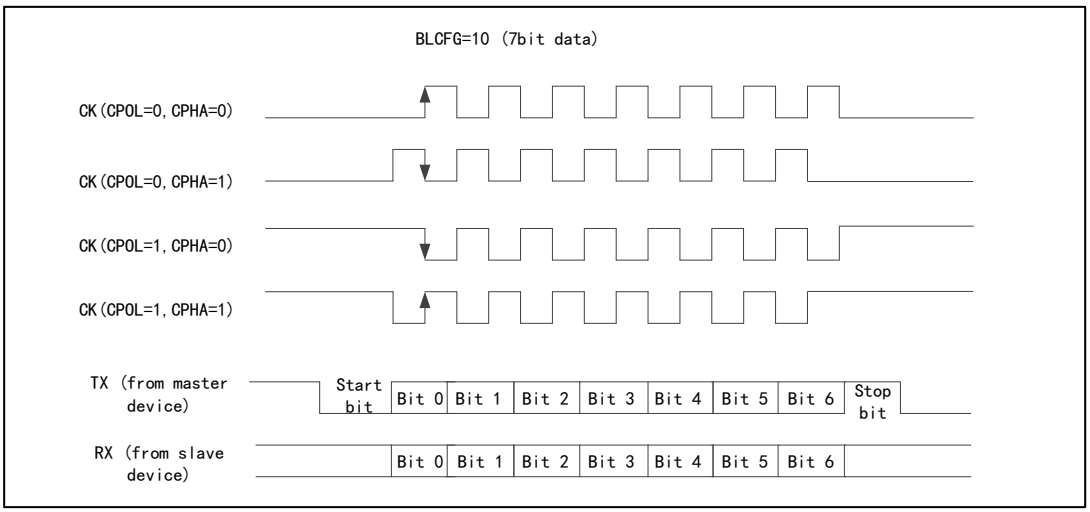


Figure 81 USART Synchronous Transmission Timing Diagram (DBLCFG=00)

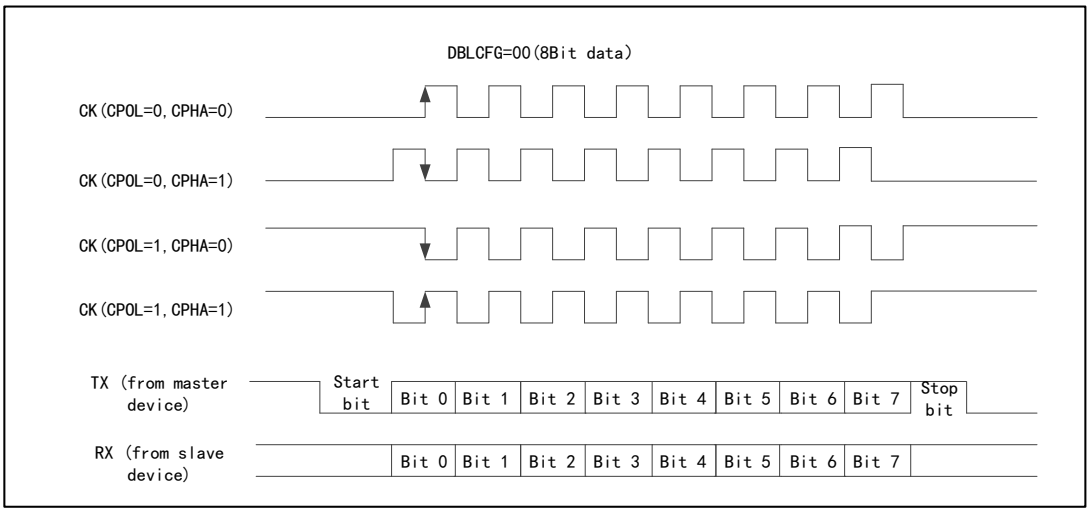
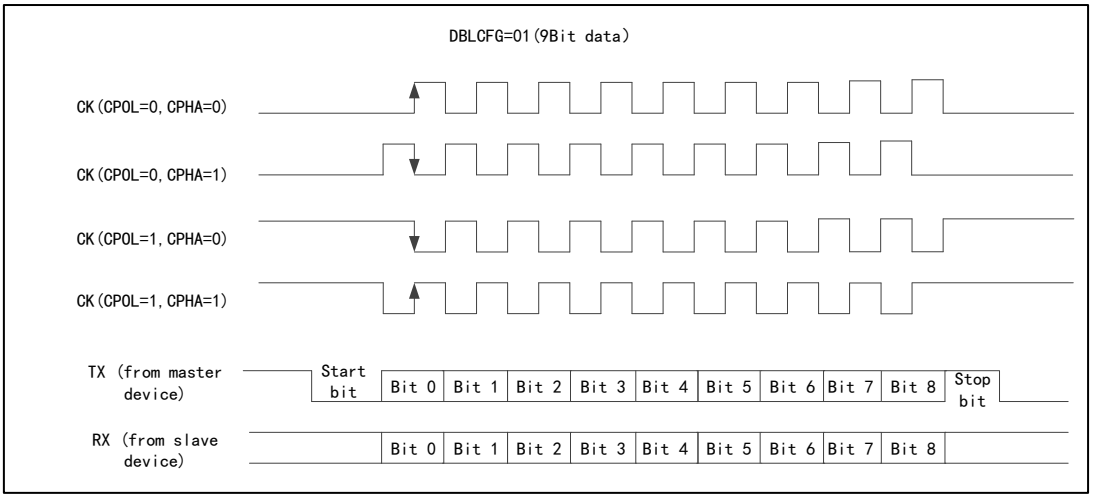


Figure 82 USART Synchronous Transmission Timing Diagram (DBLCFG=01)



21.4.11 LIN mode

LINMEN bit of USART_CTRL2 register decides whether to enter LIN mode.

When entering LIN mode:

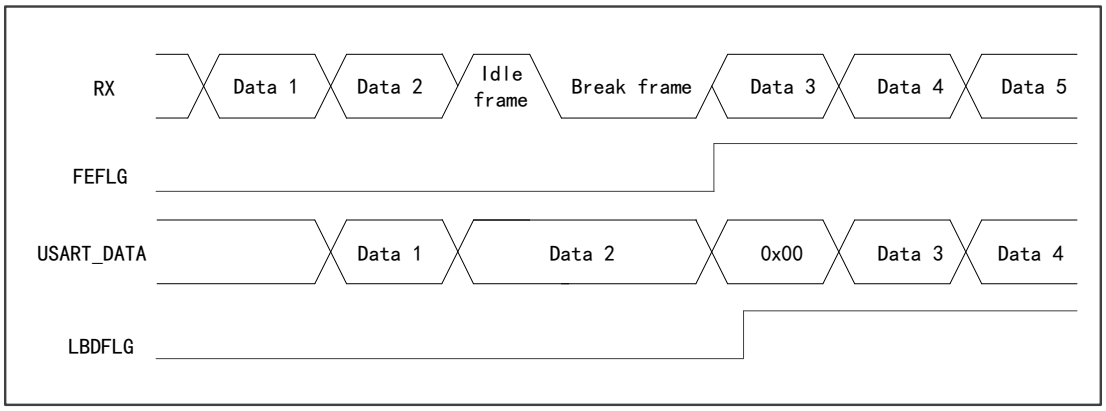
- Each data frame includes 8 data bits and 1 stop bit
- The CLKEN bit and STOPCF bit of USART_CTRL2 register and IREN bit, HDEN bit and SCEN bit of USART_CTRL3 register need to be cleared to 0.

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 error.

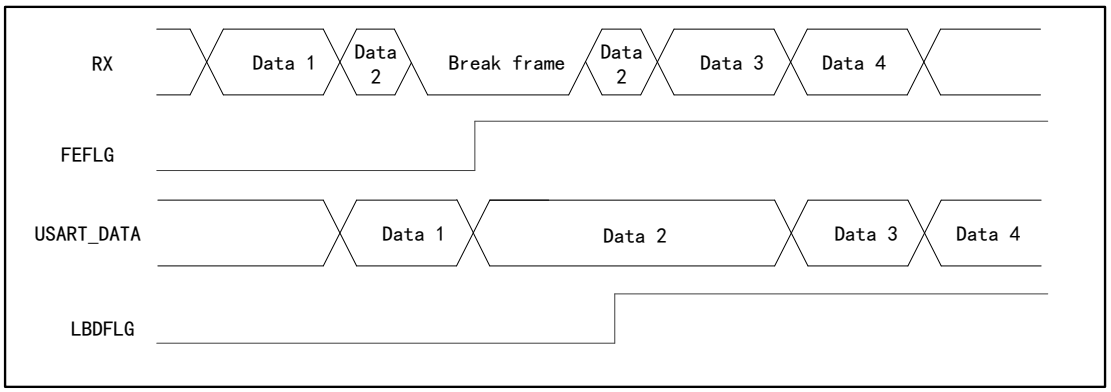
Figure 83 Break Frame Detection in Idle State



Detection of break frame 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 error.

Figure 84 Break Frame Detection in Data Transmission State



21.4.12 Smart card mode

Note: Only USART1 supports this function.

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.

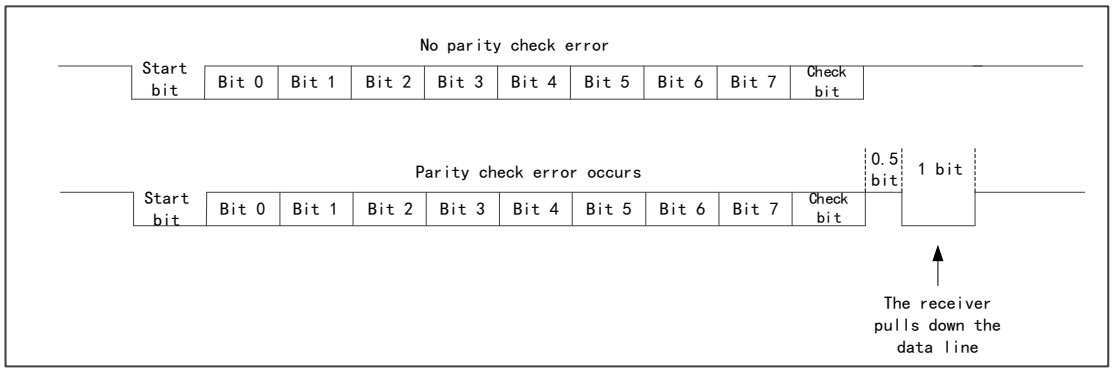
SCEN bit of USART_CTRL3 register decides whether to enter the smart card mode.

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 contains 8 data bits and 1 check bit, as well as 0.5 or 1.5 stop bits.
- CLKEN bit of USART_CTRL2 can be set to provide clocks for smart card
- During the communication, when the receiver detects a parity 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 85 ISO7816-3 Standard Protocol



21.4.13 Infrared (IrDA SIR) function mode

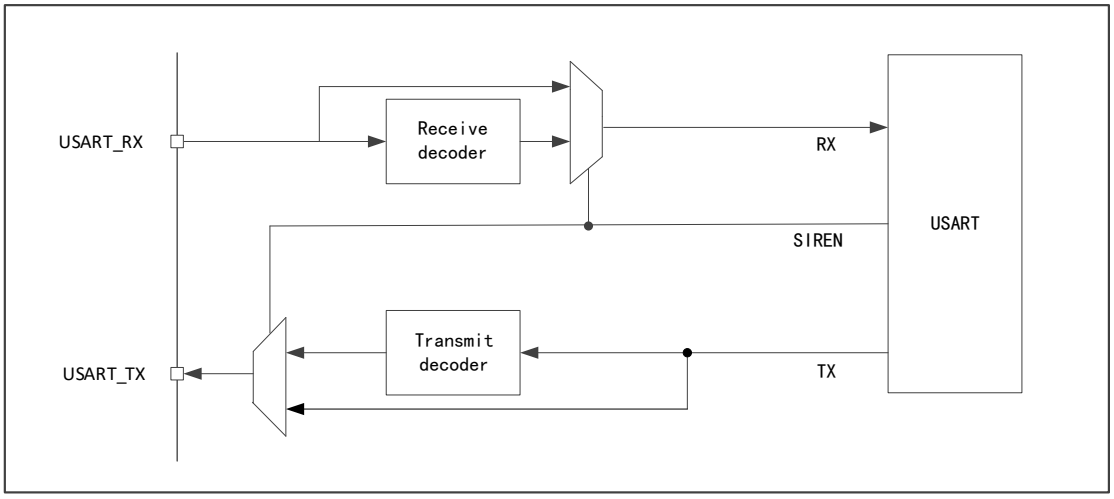
IrDA mode is a half-duplex protocol, transmitting and receiving data can not be carried out at the same time, and the delay between data transmitting and receiving should be more than 10ms.

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 cycles. 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.

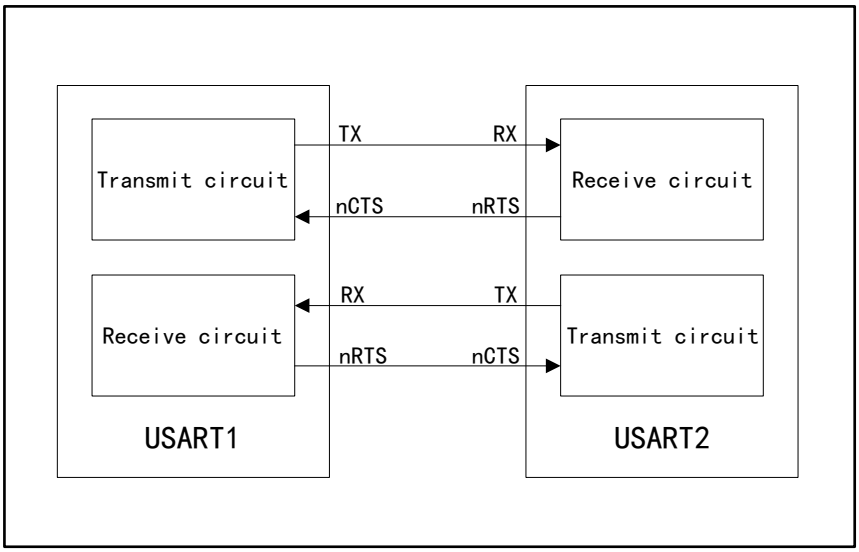
Figure 86 IrDA Mode Block Diagram



21.4.14 Hardware flow control and RS485 drive enable

The function of hardware flow control is to control the serial data stream between two devices through nCTS pin and nRTS pin.

Figure 87 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.

RS485 drive enable

DEN bit of USART_CTRL3 register determines whether to enable the driver enable function, and this function can enable DE signal to enable the control terminal of the external transceiver.

Lead time: The time interval between the drive enable signal and the start bit of the first byte. It is controlled by DLTEN[4:0] of USART_CTRL1 register.

Lag time: The time interval between the stop bit of the last byte and the release DE signal. It is controlled by DDLTEN[4:0] of USART_CTRL1 register.

21.4.15 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 SRAM will be transmitted to the buffer by DMA.

Configuration steps of transmission by DMA:

（1） Clear the TXCFLG flag bit of USART_STS register to 0

（2） Set the address of SRAM memory storing data as DMA source address

（3） Set the address of USART_TXDATA 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 receive by DMA. When receiving by DMA, every time one byte is received, the data in the

receive buffer will be transmitted to the designated SRAM area by DMA.

Configuration steps of receiving by DMA:

（1） Set the address of USART_RXDATA register as DMA source address

（2） Set the address of SRAM 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

21.4.16 ModBus communication

USART supports ModBus/RTU and ModBus/ASCII protocols, and ModBus/RTU is a half-duplex block transmission protocol. Control part of the protocol can be realized only in software. USART supports end of block detection, not requiring software or other condition.

21.4.16.1 ModBus/RTU

This function can be realized through programmable timeout function. In this mode, the end of one block is regarded as an idle line with the length greater than two characters. RXTOIEN bit of USART_CTRL1 register and RXTODEN bit of USART_CTRL2 register control the timeout function and corresponding interrupts. Write a timeout number to USART_RXTO register, and when the idle state of the receiving line reaches this length, an interrupt will be generated, indicating the completion of block receiving.

21.4.16.2 Modbus/ASCII

In this mode, the end of one block is identified by one specific (CR/LF) character sequence. USART uses character matching function to manage this mechanism. Program ASCII code of LF in ADDR[7:0] field and activate this character to match the interrupt (CMIEN=1). When a LF character is received, the software will be informed to check CR/LF in DMA buffer.

21.4.17 Interrupt request

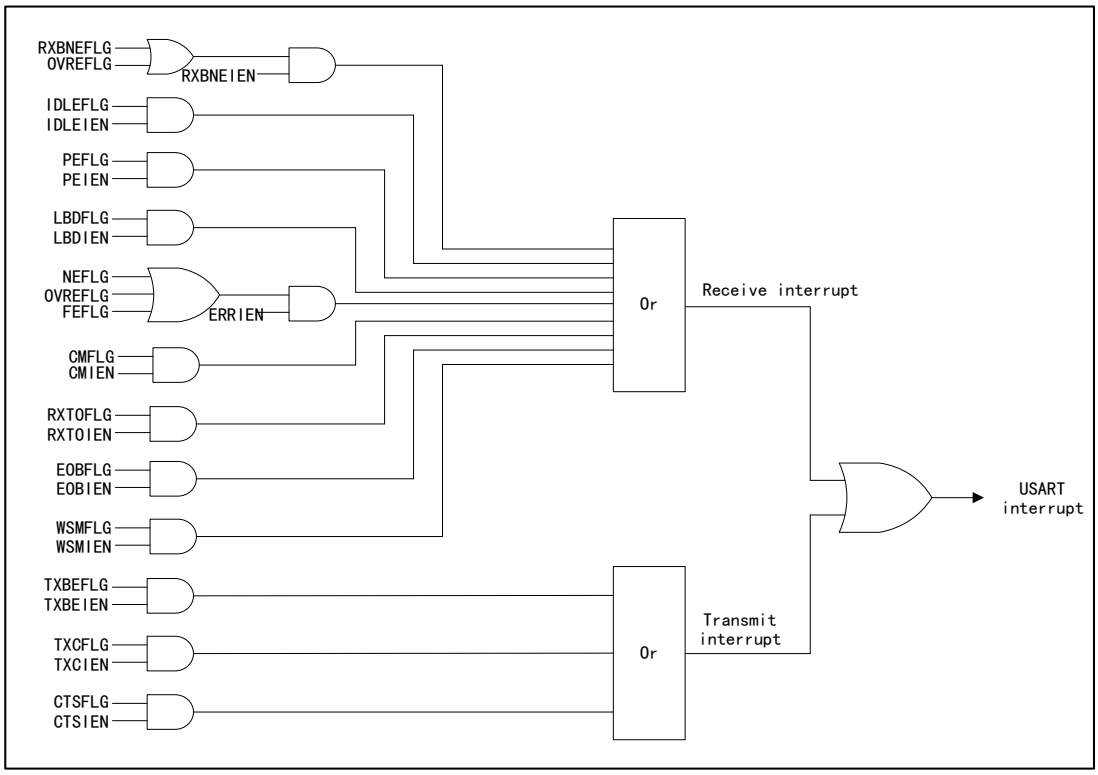
Table 70 USART Interrupt Request

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

Interrupt event		Event flag bit	Enable bit
LIN break error		LBDFLG	LBDIEN
Receiving error in DMA mode	Noise error	NEFLG	ERRIEN
	Overrun error	OVREFLG	
	Frame error	FEFLG	
Matching character		CMFLG	CMIEN
Error of failing to receive on time		RXTOFLG	RXTOIEN
End of block is detected		EOBFLG	EOBIEN
Stop mode	Wake up from the stop mode	WSMFLG	WSMIEN
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 88 USART Interrupt Mapping



21.4.18 Comparison of USART supporting functions

Table 71 Comparison of USART Supporting Functions

USART mode	USART1	USART2
Half duplex (single-line mode)	√	√
Automatic baud rate detection	√	—
Multiprocessor communication	√	√
Dual clock domain and wake-up from stop mode	√	—
Synchronize	√	—
LIN	√	—
Smart card	√	—
IrDA	√	—
Hardware flow control	√	√
RS485 drive enable	√	√
Multi-buffer communication (DMA)	√	√
Receiving timeout interrupt	√	—
ModBus communication	√	—

Note:

（1）"√" means this function is supported, while "—" means that this function is not supported.

21.5 Register address mapping

Table 72 USART Register Address Mapping

Register name	Description	Offset address
USART_CTRL1	Control register 1	0x00
USART_CTRL2	Control register 2	0x04
USART_CTRL3	Control register 3	0x08
USART_BR	Baud rate register	0x0C
USART_GTPSC	Protection time and prescaler register	0x10
USART_RXTO	Receive timeout register	0x14
USART_REQUEST	Request register	0x18
USART_STS	Interrupt and status register	0x1C

Register name	Description	Offset address
USART_INTFCLR	Interrupt flag clear register	0x20
USART_RXDATA	Data receive register	0x24
USART_TXDATA	Data transmit register	0x28

21.6 Register functional description

21.6.1 Control register 1 (USART_CTRL1)

Offset address: 0x00

Reset value: 0x0000

Field	Name	R/W	Description
0	UEN	R/W	USART Enable 0: Disable USART frequency divider and output 1: Enable USART module It is set to 1 or cleared to 0 by software; clearing this bit will cancel the current operation and the prescaler and output of USART will stop working immediately. The setting of USART will not be reset, but the status flag in USART_STS will be reset.
1	USWMEN	R/W	USART in Stop Mode Wake Up MCU Enable 0: Disable 1: Enable This bit can be set to 1 and cleared to 0 by software; to set this bit, it is required to select HSICLK or LSECLK as the clock source of USART (see chapter of RCM)
2	RXEN	R/W	Receive Enable 0: Disable 1: Enable, and start to detect the start bit on RX pin Set 1 or clear 0 by software.
3	TXEN	R/W	Transmit Enable 0: Disable 1: Enable Set 1 or clear 0 by software.
4	IDLEIEN	R/W	IDLE Interrupt Enable 0: Disable 1: Generate an interrupt when IDLEFLG is set Set 1 or clear 0 by software.
5	RXBNEIEN	R/W	Receive Buffer Not Empty Interrupt Enable 0: Disable 1: Generate an interrupt when OVREFLG or RXBNEFLG is set Set 1 or clear 0 by software.
6	TXCIEN	R/W	Transmit Complete Interrupt Enable 0: Disable 1: Generate an interrupt when TXCFLG is set Set 1 or clear 0 by software.

Field	Name	R/W	Description
7	TXBEIEN	R/W	Transmit Buffer Empty Interrupt Enable 0: Disable 1: Generate an interrupt when TXBEFLG is set Set 1 or clear 0 by software.
8	PEIEN	R/W	Parity Error interrupt Enable 0: Disable 1: Generate an interrupt when PEFLG is set Set 1 or clear 0 by software.
9	PCFG	R/W	Odd/Even Parity Configure 0: Even parity check 1: Odd parity check Set 1 or clear 0 by software. The selection will not take effect until the current transmission of bytes is completed. This bit can be set only when USART is not enabled.
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. This bit can be set only when USART is not enabled.
11	WUPMCFG	R/W	Wakeup Method Configure 0: Idle bus wakeup 1: Address tag wakeup Set 1 or clear 0 by software. This bit can be set only when USART is not enabled.
12	DBLCFG0	R/W	Data Bits Length Configure This bit and DBLCFG1 bit jointly decide the length of data bit. Set 1 or clear 0 by software. This bit cannot be modified during transmission of data.
13	RXMUTEEN	R/W	Receive Mute Mode Enable 0: Normal working mode 1: Can switch between normal mode and mute mode Set 1 or clear 0 by software.
14	CMIEN	R/W	Character Match Interrupt Enable 0: Disable 1: Generate an interrupt when CMFLG is set Set 1 or clear 0 by software.
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.

Field	Name	R/W	Description
20:16	DDLTEN[4:0]	R/W	Driver De-lead Time Enable This bit field is the time interval between the last stop bit and DE signal during transmission. Its unit is sampling time, determined by oversampling rate. If write operation is performed for USART_TXDATA within DDLTEN time, the just written data will be transmitted only after DDLTEN and DLTEN time. This bit field can be set only when USART is not enabled.
25:21	DLTEN[4:0]	R/W	Driver Lead Time Enable This bit field is the time interval between DE signal and the first start bit during transmission. Its unit is sampling time, determined by oversampling rate. This bit field can be set only when USART is not enabled.
26	RXTOIEN	R/W	Receiver Timeout Interrupt Enable 0: Disable 1: Generate an interrupt when RXTOFLG is set Set or clear 0 by software.
27	EOBIEN	R/W	End of Block Interrupt Enable Set 1 and clear 0 by software. 0: Disable 1: Generate an interrupt when EOBFLG is set Set or clear 0 by software.
28	DBLCFG1	R/W	Data Bits Length Configure This bit, together with DBLCFG0 bit, decides the length of data bit. DBLCFG[1:0]=00: 1 start bit, 8 data bits, n stop bits DBLCFG[1:0]=01: 1 start bit, 9 data bits, n stop bits DBLCFG[1:0]=10: 1 start bit, 7 data bits, n stop bits Set 1 or clear 0 by software. This bit cannot be modified during transmission of data.
31:29	Reserved		

21.6.2 Control register 2 (USART_CTRL2)

Offset address: 0x04

Reset value: 0x0000

Field	Name	R/W	Description
3:0	Reserved		
4	ADDRLEN	R/W	Slave Address Length Configure 0: 4-bit address 1: 7-bit address This bit field can be set only when USART is not enabled.
5	LBDLCFG	R/W	LIN Break Detection Length Configure 0: 10 bits 1: 11 bits This bit can be set only when USART is not enabled.
6	LBDIEN	R/W	LIN Break Detection Interrupt Enable 0: Disable 1: Generate an interrupt when LBDFLG bit is set.

Field	Name	R/W	Description
7	Reserved		
8	LBCPOEN	R/W	Last Bit Clock Pulse Output Enable 0: Not output from CK 1: Output from CK This bit is valid only in synchronous mode. This bit can be set only when USART is not enabled.
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 This bit is valid only in synchronous mode. This bit can be set only when USART is not enabled.
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. This bit can be set only when USART is not enabled.
11	CLKEN	R/W	Clock Enable (CK pin) 0: Disable 1: Enable This bit can be set only when USART is not enabled.
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 This bit can be set only when USART is not enabled.
14	LINMEN	R/W	LIN Mode Enable 0: Disable 1: Enable Set or clear 0 by software. In LIN mode, TXBFQ bit can be set to transmit and detect LIN synchronous break symbol. This bit can be set only when USART is not enabled.
15	SWAPEN	R/W	Swap TX/RX Pins Function Enable 0: Use according to standard allocation 1: The functions of TX and RX pins can be exchanged for use, and they will work when crossing and interconnecting with other USART. Set or clear 0 by software. This bit can be set only when USART is not enabled.
16	RXINVEN	R/W	RX Pin Active Level Inversion Enable 0: Standard logic level (VDD =1/IDLE, Gnd=0/mark) 1: Reverse direction (VDD =0/mark, Gnd=1/IDLE), which works when there is an external phase inverter on RX line. Set or clear 0 by software.

Field	Name	R/W	Description
			This bit can be set only when USART is not enabled.
17	TXINVEN	R/W	TX Pin Active Level Inversion Enable 0: Standard logic level (VDD =1/IDLE, Gnd=0/mark) 1: Reverse direction (VDD =0/mark, Gnd=1/IDLE), which works when there is an external phase inverter on TX line. Set or clear 0 by software. This bit can be set only when USART is not enabled.
18	BINVEN	R/W	Binary Data Inversion Enable 0: Positive/Direct logic (0=L, 1=H) 1: Negative/Reverse logic (0=H, 1=L) Set or clear 0 by software. This bit can be set only when USART is not enabled. The check bit will be inverted when this bit is set.
19	MSBFEN	R/W	Most Significant Bit First Enable 0: The data of No. 0 bit immediately follows the start bit 1: The data of the most significant bit immediately follows the start bit Set or clear 0 by software. This bit can be set only when USART is not enabled.
20	ABRDEN	R/W	Auto Baud Rate Detection Enable 0: Disable 1: Enable Set or clear 0 by software.
22:21	ABRDCFG	R/W	Auto Baud Rate Detection Mode Configure 00: Measure the start bit 01: Measure the falling edge 10: 0x7F frame detection 11: 0x55 frame detection Set or clear 0 by software.
23	RXTODEN	R/W	Receive Timeout Detection Function Enable 0: Disable 1: Enable Set or clear 0 by software. Set this bit, and when it is detected that the RX line is idle for the length of time configured by RXTO register, the RXTOFLG bit will be set by hardware.
27:24	ADDRL	R/W	USART Device Node Address Low Setup This bit field is used for wake-up detection of 7-bit address flag during multi-computer communication when entering the mute state or stop mode. This bit can be set only when the receiver is disabled or USAR is not enabled.
31:28	ADDRH	R/W	USART Device Node Address High Setup This bit field is not only used for wake-up detection of 7-bit address flag during multi-computer communication when entering the mute state or stop mode. (The most significant bit of the character of the transmitter should be 1) It is also used for character detection in normal receiving process. (Then the mute state is disabled) Then if the received 8-bit byte matches ADDRH, CMFLG bit will be set.

Field	Name	R/W	Description
			This bit can be set only when the receiver is disabled or USAR is not enabled.

21.6.3 Control register 3 (USART_CTRL3)

Offset address: 0x08

Reset value: 0x0000

Field	Name	R/W	Description
0	ERRIEN	R/W	Error interrupt Enable 0: Disable 1: Enable; when any bit among FEFLG, OVREFLG and NEFLG is set, an interrupt will be generated.
1	IREN	R/W	IrDA Function Enable 0: Disable 1: Enable Set or clear 0 by software. This bit can be set only when USART is not enabled.
2	IRLPEN	R/W	IrDA Low-power Mode Enable 0: Normal mode 1: Low-power mode This bit can be set only when USART is not enabled.
3	HDEN	R/W	Half-duplex Mode Enable 0: Disable 1: Enable This bit can be set only when USART is not enabled.
4	SCNACKEN	R/W	NACK Transmit Enable During Parity Error in Smartcard Function 0: Not transmit NACK 1: Transmit NACK This bit can be set only when USART is not enabled.
5	SCEN	R/W	Smartcard Function Enable 0: Disable 1: Enable This bit can be set only when USART is not enabled.
6	DMARXEN	R/W	DMA Receive Enable 0: Disable 1: Enable Set or clear 0 by software.
7	DMATXEN	R/W	DMA Transmit Enable 0: Disable 1: Enable Set or clear 0 by software.

Field	Name	R/W	Description
8	RTSEN	R/W	RTS 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. This bit can be set only when USART is not enabled.
9	CTSEN	R/W	CTS Function Enable 0: Disable 1: Enable CTS: Clear To Send, which is input signal When CTS input signal is low, the data can be transmitted; otherwise, the data cannot be transmitted; if CTS signal is pulled to 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. This bit can be set only when USART is not enabled.
10	CTSIEN	R/W	CTS Interrupt Enable 0: Disable 1: Generate an interrupt when CTSFLG is set
11	SAMCFG	R/W	Sample Method Configure 0: Sampling for three times 1: Single sample; flag of noise detectionf disabled This bit can be set only when USART is not enabled.
12	OVRDEDIS	R/W	Overrun Detection Disable 0: Enable. When RXBNEFLG bit is set and new data is received, OVREFLG bit will be set. 1: Disable. When new data are received, if RXBNEFLG is still set but OVREFLG is not set, the data not read will be overwritten by new data. This bit can be set only when USART is not enabled.
13	DDISRXEEN	R/W	DMA Disable on Receive Error Enable 0: Not disable DMA. The corresponding error flag bit will be set, but in order to avoid data from overrunning and being overwritten, RXBNEFLG will not be set. In smart card mode, as a result, no DMA request will be issued, so wrong data will not be transmitted, but the next correct data will be transmitted. 1: Disable DMA. If RXBNEFLG is set, the corresponding error flag bit will also be set. DMA request will be unmasked only when the corresponding error flag bit is cleared to 0. Therefore, it is required to first disable DMA request or first clear RXBNEFLG flag and then clear the error flag. This bit can be set only when USART is not enabled.

Field	Name	R/W	Description
14	DEN	R/W	Driver Enable Users are allowed to activate the control terminal of external transceiver through DE signal. 0: Disable DE function 1: Enable DE function, and output DE signal on RTS pin This bit can be set only when USART is not enabled.
15	DPCFG	R/W	Driver Polarity Configure 0: DE signal is active high 1: DE signal is active low This bit can be set only when USART is not enabled.
16	Reserved		
19:17	SCARCCFG	R/W	Smartcard Mode Auto-retry Count Configure 0x0: Disable the retransmission function and data will not be retransmitted in transmission mode. 0x1~0x7: The number of automatic retry times Transmitting mode: The number of times of automatic retransmission of data before transmission error is generated. Receiving mode: The number of times of automatic retry of receiving before receiving error is generated. This bit can be set only when USART is not enabled.
21:20	WSIFLGSEL	R/W	Wakeup From Stop Mode Interrupt Flag Select 00: Address matches 01: Reserved 10: When start bit is detected 11: When the receive data register is not empty This bit can be set only when USART is not enabled.
22	WSMIEN	R/W	Wakeup from Stop mode interrupt enable 0: Disable 1: Generate an interrupt when WSMFLG is set Set or clear 0 by software.
31:23	Reserved		

21.6.4 Baud rate register (USART_BR)

This register can be set only when USART is not enabled. This bit may be reset by hardware during automatic baud rate detection.

Offset address: 0x0C

Reset value: 0x0000

Field	Name	R/W	Description
3:0	FBR	R/W	Fraction of USART Baud Rate Divider factor The decimal part of USART baud rate division factor is determined by these four bits.
15:4	IBR	R/W	Integer of USART Baud Rate Divider factor The integral part of USART baud rate division factor is determined by these 12 bits.
31:16	Reserved		

21.6.5 Protection time and prescaler register (USART_GTPSC)

Offset address: 0x10

Reset value: 0x0000

Field	Name	R/W	Description
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. 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 00000: Reserved 00001: 2 divided frequency 00010: 4 divided frequency 00011: 6 divided frequency …… 11111: 62 divided frequency The bit [7:5] is not used in smart card mode. This bit can be set only when USART is not enabled.
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. This bit can be set only when USART is not enabled.
31:16	Reserved		

21.6.6 Receive timeout register (USART_RXTO)

Offset address: 0x14

Reset value: 0x0000

Field	Name	R/W	Description
23:0	RXTO	R/W	Receiver Timeout Value Setup This bit field specifies the receive timeout value in the unit of baud clock. In standard mode, after the last byte is received, if no new start bit is detected within the duration of RXTO value, RXTOFLG will be set by hardware. In smart card mode, this value is used to realize CWT and BWT. In this mode, start timeout measurement from the start bit of the last byte.
31:24	BLEN[7:0]	R/W	Block Length Setup

Field	Name	R/W	Description
			This bit field specifies the block length when receiving in the smart card mode T=1. This value is the number of characters in the information block + the end part (1-LEC/2-CRC)-1. For example: BLEN =0->0 information characters +LEC BLEN =1->0 information characters +CRC BLEN =255-> 254 information characters +CRC In smart card mode, when TXBEFLG=0, the block length counter will be cleared. The block length counter is cleared to 0 when RXEN=0 or EOBCFLG bit is set.

21.6.7 Request register (USART_REQUEST)

Offset address: 0x18

Reset value: 0x0000

Field	Name	R/W	Description
0	ABRDQ	W	Auto Baud Rate Detection Request Set this bit, the ABRDFLG flag will be cleared and an automatic baud rate detection will be conducted when receiving data next time.
1	TXBFQ	W	Transmit Break Frame Request Set this bit, TXBFFLG flag will be set and a break frame will be transmitted after the transmission state machine is enabled.
2	MUTEQ	W	Mute Mode Request Set this bit to enter the mute mode and RXWFMUTE flag will be cleared.
3	RXDFQ	W	Receive Data Flush Request Set this bit and RXBNEFLG flag will be cleared. The data that has not been read out in the receive register can be discarded to avoid overrun error
4	TXDFQ	W	Transmit Data Flush Request Set this bit and TX flag will be set. Data transmission can be canceled. This bit takes effect when data are not transmitted due to error interrupt and FEF flag will be set. This bit takes effect only in smart card mode
31:5	Reserved		

21.6.8 Interrupt and status register (USART_STS)

Offset address: 0x1C

Reset value: 0x0200 00C0

Field	Name	R/W	Description
0	PEFLG	R	Parity Error Occur Flag 0: No error 1: Parity error is detected In receiving mode, when a parity error occurs, it is set to 1 by hardware; it can be cleared by setting PECLR.
1	FEFLG	R	Frame Error Occur Flag 0: No frame error

Field	Name	R/W	Description
			1: Frame error or break symbol is detected When there is synchronous dislocation, too much noise or break symbol, it is set to 1 by hardware; this bit can be cleared by setting FECLR.
2	NEFLG	R	Noise Error Occur Flag 0: No noise 1: Noise is detected When there is noise error, this bit is set to 1 by hardware; this bit can be cleared by setting NFCLR.
3	OVREFLG	R	Overrun Error Occur Flag 0: No overrun error 1: Overrun error is detected When the RXBNEFLG bit is set and the data in the shift register is to be transmitted to the receive register, it is set to 1 by hardware; this bit can be cleared by setting OVRECLR.
4	IDLEFLG	R	IDLE Line Detected Flag 0: Idle bus is not detected 1: Idle bus is detected When idle bus is detected, this bit is set to 1 by hardware; this bit can be cleared by setting IDLECLR.
5	RXBNEFLG	R	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 is set to 1 by hardware; this bit can be cleared by reading the TXDATA register or setting RXDFQ.
6	TXCFLG	R	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 is set to 1 by hardware; this bit can be cleared by writing to TXDATA register or setting TXCCLR.
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, this bit is set to 1 by hardware; this bit can be cleared by writing to TXDATA register.
8	LBDFLG	R	LIN Break Detected Flag 0: LIN break is not detected 1: LIN break is detected When LIN break is detected, this bit is set to 1 by hardware; this bit can be cleared by setting LBDCLR. If LBDIEN in USART_CTRL2 is set, an interrupt will be generated.
9	CTSFLG	R	CTS Change Flag 0: No change on nCTS state line 1: There is change on nCTS state line

Field	Name	R/W	Description
			If the CTSEN bit is set, when switching to the nCTS input, this bit is set to 1 by hardware; this bit can be cleared by setting CTSCLR.
10	CTSCFG	R	CTS Status Configure 0: Set nCTS line 1: Reset nCTS line This bit is set to 1 or cleared to 0 by hardware. This bit sets reversed state of nCTS input pin.
11	RXTOFLG	R	Receiver Timeout Flag 0: Not time out 1: Timed out If the start bit is not detected within the duration set by RXTO bit, this bit is set to 1 by hardware; this bit can be cleared by setting RXTOCLR bit.
12	EOBFLG	R	End of Block Flag 0: Failed to reach the end of block 1: Reach the end of block When receiving a complete block, this bit is set to 1 by hardware; this bit can be cleared by setting EOBCLR bit. Detection is completed when the received bytes are BLEN+4. If EOBIEN bit is set, an interrupt will be generated.
13	Reserved		
14	ABRDEFLG	R	Auto Baud Rate Detection Error Flag This bit is set to 1 by hardware when baud rate detection fails; this bit can be cleared by setting ABRDQ bit.
15	ABRDFLG	R	Auto Baud Rate Detection Flag When the automatic baud rate function is enabled or the automatic baud rate operation is interrupted, it is set to 1 by hardware; this bit can be cleared when resuming the baud rate detection.
16	BSYFLG	R	Busy Flag 0: Idle state 1: In the process of receiving data This bit is set to 1 by hardware when the start bit is detected, and it will be cleared after receiving is over. This bit is set to 1 or cleared to 0 by hardware.
17	CMFLG	R	Character Match Flag 0: No character matches 1: There is matching character When the received character matches the value set by ADDR[7:0], this bit is set to 1 by hardware; this bit can be cleared by setting CMCLR bit.
18	TXBFFLG	R	Transmit Break Frame Flag 0: Not transmit 1: Will transmit If TXBFQ bit is set, this bit can be set to 1 by software; when transmitting the stop bit of the break frame, this bit is cleared to 0 by hardware.

Field	Name	R/W	Description
19	RXWFMUTE	R	Receiver Wakeup From Mute Mode 0: Normal mode 1: Mute mode When switching the wake-up mode and the mute mode, this bit is set to 1 and cleared to 0 by hardware; if it is waken up by idle signal, this bit is set to 1 by writing to USART_REQUEST register. WUPMCFG bit determines the control sequence of mute mode.
20	WSMFLG	R	Wakeup From Stop Mode Flag 0: Not detected 1: Detected This bit can be cleared by setting PECLR bit. If WSMFLG bit is set, an interrupt will be generated.
21	TXENACKFLG	R	Transmit Enable Acknowledge Flag Set to 1 by hardware when reading the transmit enable signal. The idle frame request will be generated when TXEN=0. To ensure minimum cycle of TXEN=0, TXEN will be set immediately.
22	RXENACKFLG	R	Receive Enable Acknowledge Flag Set to 1 by hardware when reading the receive enable signal. This bit is used to confirm whether USART has been ready to receive data before entering the stop mode.
31:23	Reserved		

21.6.9 Interrupt flag clear register (USART_INTFCLR)

Offset address: 0x20

Reset value: 0x0000

Field	Name	R/W	Description
0	PECLR	RC_W1	Parity Error Flag Clear Set this bit and PEFLG flag bit of USART_STS register can be cleared.
1	FECLR	RC_W1	Framing Error Flag Clear Set this bit and FEFLG flag bit of USART_STS register can be cleared.
2	NECLR	RC_W1	Noise Detected Flag Clear Set this bit and NEFLG flag bit of USART_STS register can be cleared.
3	OVRECLR	RC_W1	Overrun Error Flag Clear Set this bit and OVREFLG flag bit of USART_STS register can be cleared.
4	IDLECLR	RC_W1	IDLE Line Detected Clear Flag Set this bit and IDLEFLG flag bit of USART_STS register can be cleared.
5	Reserved		
6	TXCCLR	RC_W1	Transmission Data Complete Flag Clear

Field	Name	R/W	Description
			Set this bit and TXCFLG flag bit of USART_STS register can be cleared.
7	Reserved		
8	LBDCLR	RC_W1	LIN Break Detection Flag Clear LBDFLG flag bit of USART_STS register can be cleared by setting this bit.
9	CTSCLR	RC_W1	CTS Flag Clear Set this bit and CTSFLG flag bit of USART_STS register can be cleared.
10	Reserved		
11	RXTOCLR	RC_W1	Receiver Timeout Flag Clear Set this bit and RXTOFLG flag bit of USART_STS register can be cleared.
12	EOBCLR	RC_W1	End of Block Flag Clear EOBFLG flag bit of USART_STS register can be cleared by setting this bit.
16:13	Reserved		
17	CMCLR	RC_W1	Character Match Flag Clear Set this bit and CMFLG flag bit of USART_STS register can be cleared.
19:18	Reserved		
20	WSMCLR	RC_W1	Wakeup From Stop Mode Flag Clear WSMFLG flag bit of USART_STS register can be cleared by setting this bit.
31:21	Reserved		

21.6.10 Data receive register (USART_RXDATA)

Offset address: 0x24

Reset value: 0xXXXX

Field	Name	R/W	Description
8:0	RXDATA	R	Receive Data Value Setup Include the received data byte. Provide the parallel interface between input shift register and internal bus. If the check bit is turned on when receiving data, read this register and the most significant bit is the check bit.
31:9	Reserved		

21.6.11 Transmit data register (USART_TXDATA)

Offset address: 0x28

Reset value: 0xXXXX

Field	Name	R/W	Description
8:0	TXDATA	R/W	Transmit Data Value Setup Include the data byte to be transmitted.

Field	Name	R/W	Description
			Provide the parallel interface between transmit shift register and internal bus. If the check bit is turned on when transmitting data, it is invalid to write to the most significant bit, and it will be replaced by the check bit and transmitted.
31:9	Reserved		

22 Internal integrated circuit interface (I2C)

22.1 Full name and abbreviation of terms

Table 73 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
Acknowledgement	ACK
Packet Error Checking	PEC

22.2 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

22.3 Main characteristics

（1） Can select master or slave mode

（2） Multi-master function

（3） 7-bit and 10-bit addressing mode

（4） Response to broadcast

（5） Multiple 7-bit slave address

（6） Three modes

- Standard mode

- Fast mode
- Fast mode plus

（7） Programmable clock extension

（8） Software Reset

（9） Programmable start time and hold time

（10） DMA function

（11） Programmable noise filter

（12） SMBus specific function

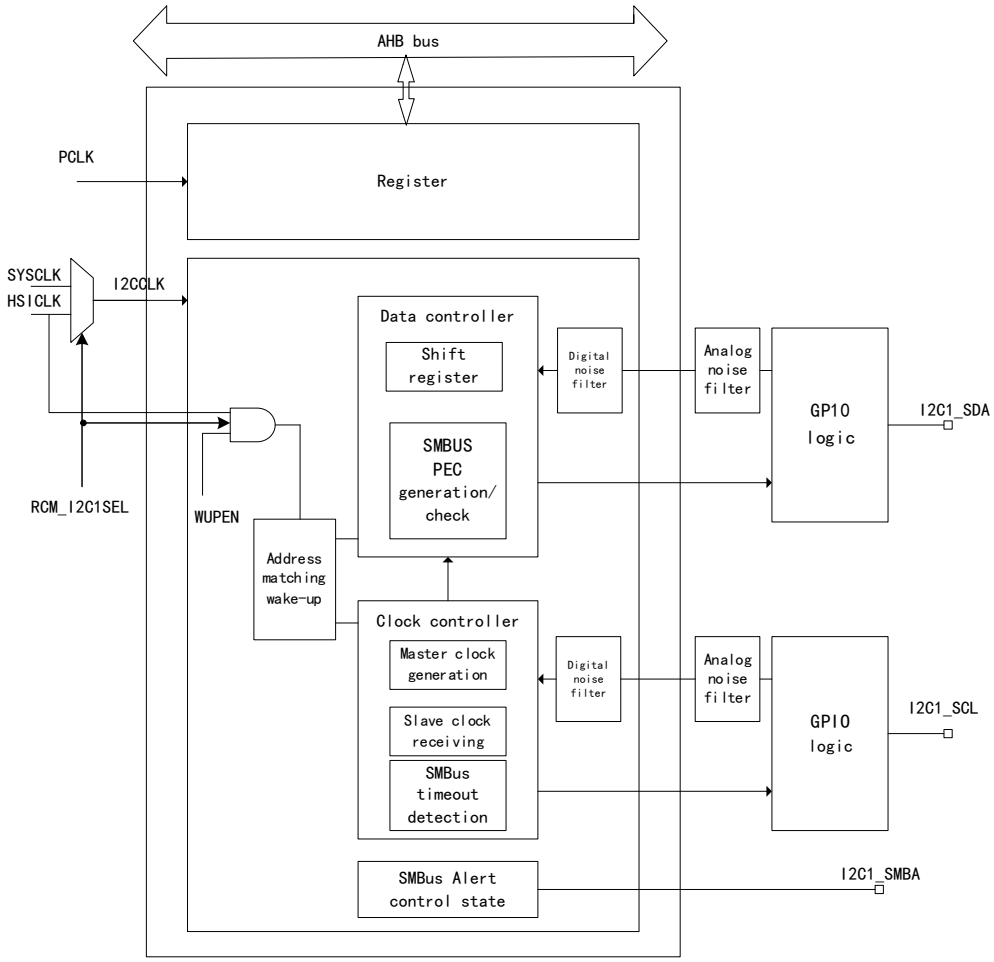
- Hardware PEC
- Command receiving and data acknowledgment control
- Address resolution protocol
- HOST notification protocol
- SMBus alarm
- SMBus timeout management

（13） Can select an independent clock source

（14） Wake up from the stop mode

22.4 Structure block diagram

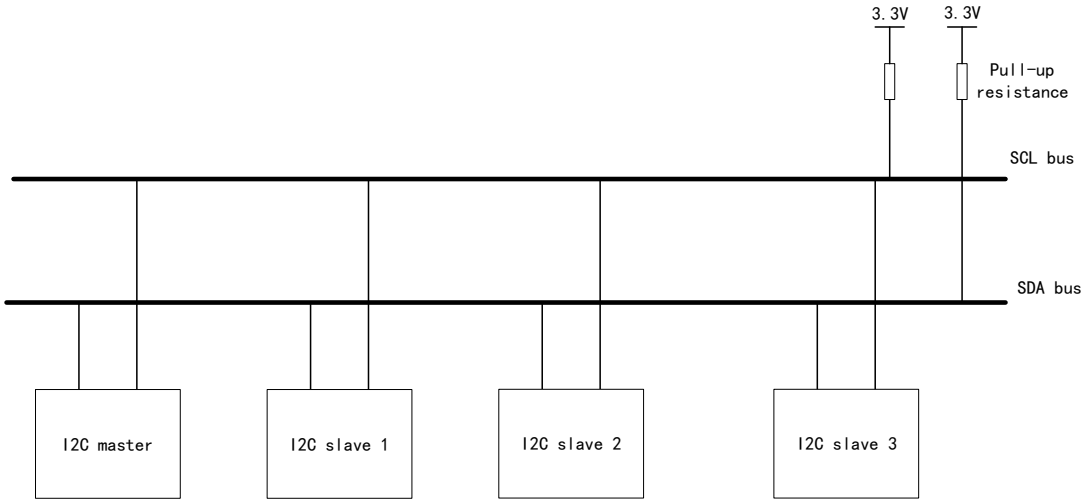
Figure 89 I2C1 Functional Structure Diagram



22.5 Functional description

22.5.1 I2C physical layer

Figure 90 Common I2C Communication Connection Diagram



Characteristics of physical layer

（1） It supports the buses of multiple devices (signal line shared by multiple devices), which, 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 (up to 100KHz), fast mode (up to 400KHz), and fast mode plus (up to 1MHz).

（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） Can program the setup and hold time, and program the high-level time and low-level time of SCL in I2C.

22.5.2 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 bit, stop bit and acknowledge bit.

- 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 period of the ninth clock pulse, the receiver will pull down SDA to respond to the received data.

I2C communication reading and writing process

Figure 91 Master Writes Data to Slave

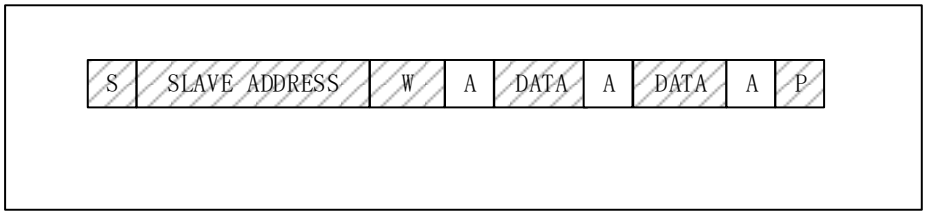
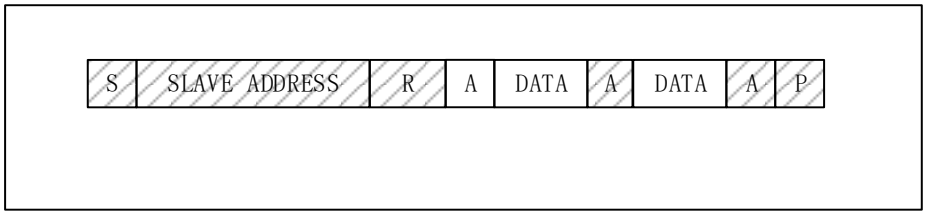


Figure 92 Master Reads Data from Slave



Notes:

（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 reading, while 0 means writing

（7）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 following data signal.

When the transmission direction is writing data

After broadcasting the address and receiving the acknowledge signal, the master will transmit data to the slave. The length of the data is one byte. Every time the master transmits one byte of data, it needs to wait for the acknowledge signal transmitted by the slave. When all the bytes are transmitted, the master will transmit a stop signal to the slave, indicating that the transmission is completed.

When the transmission direction is reading data

After broadcasting the address and receiving the acknowledgment 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 acknowledgment signal from the master. When the master wants to stop receiving data, it needs to return a non-acknowledge signal to the slave, then the slave will stop transmitting the data automatically.

22.5.3 Introduction to I2C clock

22.5.3.1 I2C clock source

I2C is driven by an independent clock source, and it can make I2C1 operate independent of PCLK frequency.

I2C clock source can be HSICLK or SYSCLK.

22.5.3.2 Requirements for I2C clock

（1） t_{I2C_CLK}< (tlow-tfilters)/4 and $t_{I2C\_CLK<tHIGH}$

（2） t_{low}: SCL low-level time

（3） t_{HIGH}: SCL high-level time

（4） $t_{filters}$: Total lag caused by analog filter and digital filter when I2C is enabled

I2C clock configuration

Before peripherals are enabled, it is required to configure SCLH and SCLL bits in I2C_TIMING register to configure the I2C clock.

It can realize clock synchronization mechanism and supports multi-master environment and slave clock extension.

tSCL=tSYNC1 +tSYNC2 +{（（SCLH+1）+（SCLL+1））*（TIMINGPSC+1） * tI2C_CLK}

t_{SYNC1} depends on:

- SCL descending slope
- Input delay of analog filter
- Input delay of digital filter
- Delay caused by synchronous I2C_CLK clock of SCL

t_{SYNC2} depends on:

- SCL rising slope:
- Input delay of analog filter
- Input delay of digital filter
- Delay caused by synchronous I2C_CLK clock of SCL

To make I2C compatible with SMBus mode, the requirements for clock timing are shown in the table below:

Table 74 Clock Timing Requirements

Symbol	Parameter	Standard mode		Fast mode		Fast mode plus		SMBus		Unit
		Minimum value	Maximum value	Minimum value	Maximum value	Minimum value	Maximum value	Minimum value	Maximum value	
fSCL	SCL clock frequency	-	100	-	400	-	1000	-	100	KHz
tHD:STA	START signal hold time	4	-	0.6	-	0.26	-	4.0	-	μs
tSU:STA	START signal setup time	5	-	0.6	-	0.26	-	4.7	-	μs
tSU:STO	STOP signal setup time	4	-	0.6	-	0.26	-	4.7	-	μs
tBUF	Idle time of bus between STOP and START signals	5	-	1.3	-	0.50	-	4.0	-	μs
tLOW	SCL clock low-level time	8	-	1.3	-	0.50	-	4.7	-	μs
tHIGH	SCL clock high-level time	4	-	0.6	-	0.26	-	4.0	50	μs

Symbol	Parameter	Standard mode		Fast mode		Fast mode plus		SMBus		Unit
		Minimum value	Maximum value	Minimum value	Maximum value	Minimum value	Maximum value	Minimum value	Maximum value	
tr	Rising edge time of SDA and SCL signals	-	1000	-	300	-	120	-	1000	ns
tf	Falling edge time of SDA and SCL signals	-	300	-	300	--	120	-	300	ns

22.5.3.3 I2C_TIMING register configuration

Table 75 I2C_TIMING Register Configuration

fI2C_CLK=48MHz				
Parameter	Standard mode		Fast mode	Fast mode plus
	10 KHz	100 KHz	400 KHz	1 MHz
TIMINGPSC	0xB	0xB	5	5
SCLL	0xC7	0x13	0x9	0x3
tSCLL	200x250ns=50μs	20x250ns= 5.0μs	10x125ns = 1250 ns	4x125 ns =500 ns
SCLH	0xC3	0xF	0x3	0x1
tSCLH	196x250 ns = 49μs	16x250 ns = 4.0μs	4x125 ns = 500 ns	2x125 ns = 250 ns
tSCL	100μs	10μs	2500ns	875ns
DATAHT	0x2	0x2	0x3	0
tDATAHT	2x250 ns = 500 ns	2x250 ns = 500 ns	3x125 ns = 375 ns	0ns
DATAT	0x4	0x4	0x3	0x1
tDATAT	5x250 ns = 1250 ns	5 x250 ns = 1250 ns	4x125 ns = 500 ns	2x125 ns = 250 ns

Note:

tI2C_CLK = 1/ fI2C_CLK

tTIMINGPSC =（TIMINGPSC+1）x tI2C_CLK

tDATAT =（DATAT+1）x tTIMINGPSC

tDATAHT = DATAHT x tTIMINGPSC

tSCLH =（SCLH+1）x tTIMINGPSC

tSCLL =（SCLL+1）x tTIMINGPSC

22.5.4 I2C function configuration mode

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.

22.5.4.1 Software Reset

Software reset is realized by setting SWRST bit of I2C_CTRL1 register. At this time, release the SCL and SDA lines, the internal state machine will be reset, all communication control bits and state bits will return to their reset values, and the configuration registers are unaffected.

22.5.4.2 Slave mode

Transmit in slave mode

After the master transmits the start signal and address, the addressing is successful, the ADDRMFLG bit is cleared, and the transmitter will transmit the data to be transmitted from I2C_TXDATA register to SDA line by internal shift register.

Every time the slave transmits a byte, it will wait for the master's acknowledge signal (ACK) and repeat this process until the master wants to stop receiving data and returns a non-acknowledge signal (NACK) to the slave. At this time, the slave will stop data transmission.

Receive in slave mode

After receiving the address of the master, ADDRMFLG bit will be cleared, and the data received by the slave from the SDA line through the internal shift register are stored in I2C_RXDATA register.

After the slave receives a byte, it will transmit an acknowledge signal (ACK) to the master and when the master transmits a stop signal, the transmission is over.

Extension of slave clock

In default mode, I2C slave will pull down SCL clock in the following situations:

- The received address matches the enabled slave address, and SCL clock is pulled down and will be released when ADDRMFLG flag is cleared to b y software. ADDRMFLG flag bit can be cleared by setting ADDRMCLR bit to 1.
- When transmitting, if the previous data have been transmitted and no new data are written to I2C_TXDATA register, or ADDRMFLG flag is cleared, and no byte is written to I2C_TXDATA register, the SCL clock will be pulled down and when data are written to I2C_TXDATA register, the SCL clock will be released.

- When receiving, if the content of I2C_RXDATA register is not read and new data are received, the SCL clock will be pulled down and when I2C_RXDATA register is read, the SCL clock will be released.

22.5.4.3 Master mode

Master transmitting

I2C interface transmits the start signal and transmits the address to the SDA line through the internal shift register. The transmission direction is write, waiting for the slave to respond. After the slave responds, the master will transmit bytes from I2C_TXDATA register to SDA line through the internal shift register and wait for the acknowledge signal (ACK) transmitted by slave, and so forth. When I2C_TXDATA register writes the last byte, the stop bit is set to generate a stop signal.

Master receiving

The I2C interface transmits the start signal and transmits the address to the SDA line through the internal shift register. The transmission direction is read. After the slave responds, the master enters the receiving mode, receives the data on the SDA line through the internal shift register and transmits them to I2C_RXDATA register. Every time the master receives a data, it will return an acknowledge signal (ACK). This process will be repeated and when the master needs to stop reading data, it will transmit a non-acknowledge signal (NACK) to stop reading data.

22.5.4.4 SMBus specific function

The system management bus (SMBus) is a two-wire interface, which is based on I2C bus principle.

The system management bus specification refers to three types of devices

Slave: Device of receiving or corresponding command.

Master: Device that issues commands, generates clocks and terminates transmission.

HOST: A special master, which provides interfaces to system CPU. The HOST must have dual functions of master and slave, and support SMBus HOST notification protocol, and one system has only one HOST.

Bus protocol

There are 11 possible command protocols for any given device, and one device can communicate with any or all of 11 protocols.

Address resolution protocol (ARP)

SMBus slave address collision can be solved by calibrating a new unique address for the slave device. In order to assign addresses, a mechanism is needed to distinguish each device, and each device has a unique device identifier. The 128-bit identifier is implemented by software.

This device supports address resolution protocol (ARP). Set DEADDREN bit in I2C_CTRL1 register to 1, and the default address of SMBus device (0b1100001) will be enabled. ARP command is implemented by user software.

The arbitration supported by ARP is also completed in slave mode.

Command receiving and data acknowledgment control

SMBus receiver will return NACK to each command and data received. Enable the ACK control in slave mode, and set SBCEN bit of I2C_CTRL1 register to 1 to enable the slave byte control mode.

HOST notification protocol

Set HADDREN bit of I2C_CTRL1 register to make this peripheral support HOST notification protocol. In such case, HOST will acknowledge SMBus host slave (0b0001000).

When this protocol is adopted, this device is used as the master, and HOST is the slave.

SMBus alarm

This peripheral can be supported by SMBus reminder signal. When a device that is used only as the slave wants to initiate communication, it can notify HOST through SMBALERT pin. HOST will handle the interrupt and then access all SMBALERTdevices through the reminder response address (0b0001100). Only the device with the SMBALERT pin pulled down will respond to the reminder response address.

SMBus timeout management

Table 76 SMBus Timeout Specification

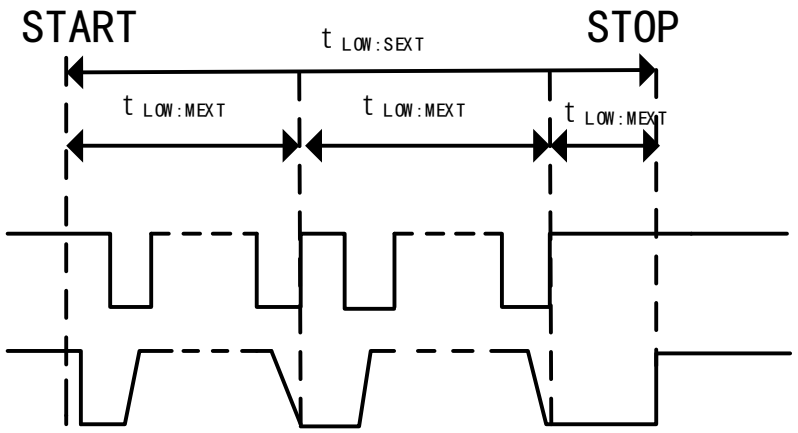
Symbol	Parameter	Range		Unit
		Minimum value	Maximum value	
t TIMEOUT	Low timeout of detection clock	25	35	ms
t LOW:SEXT	Low extension time of cumulative clock of slave		25	ms
t LOW:MEXT	Low extension time of cumulative clock of master		10	ms

$t_{LOW:SEXT}$ is the extensible clock cycle accumulation given by a slave device

from START to STOP. When a slave device or a master device occupies the clock, the total low clock time is greater than $t_{LOW:SEXT}$. Therefore, the test condition of this parameter is that the slave is the only communication target of a full-speed master.

$t_{LOW:MEXT}$ is the clock cycle accumulation allowed by a master device to transmit a byte from START to ACK, from ACK to ACK, from ACK to STOP. When another slave device or master occupies the clock, the total time occupies by the clock may also be greater than $t_{LOW:MEXT}$. Therefore, the measurement condition of this parameter is that only one full-speed slave is the only communication target.

Figure 93 $t_{LOW:SEXT}$ and $t_{LOW:MEXT}$ Time



Idle bus detection

If the master detects that the clock signal is high time greater than $t_{HIGH,MAX}$, the bus is in idle state.

22.5.4.5 Wake up from the stop mode

When WUPEN bit of I2C_CTRL1 register is set to 1, enable the function of wake-up from the stop mode.

It can wake up from the stop mode only when HSICLK oscillator is selected as the clock source of I2C_CLK.

In stop mode, HSICLK is disabled; after the start bit is detected, I2C interface will enable HSICLK and SCL will be pulled down until HSICLK is enabled. When the address matches, I2C will pull down SCL continuously during MCU wake-up period. SCL can be released only when ADDRMFLG flag is cleared by software, and then the transmission will enter normal state. If the address does not match, HSICLK will be disabled, and MCU will remain in stop mode.

Note:

（1）If I2C_CLK is selected as system clock or WUPEN=0, HSICLK oscillator will not be enabled even if it receives the start bit.

（2）MCU can be woken up only by ADDR interrupt, so when I2C is used as the master to transmit data or the slave is addressed, it is not allowed to enter the stop mode.

（3）The digital filter and wake-up in stop mode are not compatible. Therefore, if DNFCFG bit is not 0, setting WUPEN bit is invalid.

（4）Clock extension function needs to be enabled.

22.5.4.6 Error flag bit

I2C communication has the following error flag bits that may cause communication failure.

Bus error flag bit (BERRFLG)

When a START or STOP condition is detected outside 9 times the SCL clock pulse signal, a bus error will occur. When SCL is high and a rising edge or falling edge occurs on SDA, it will be detected as START or STOP signal.

Only when I2C is communicating and transmitting data, can bus error occur (after data have been transmitted as the master or the address has matched as the slave). This error will not occur in slave mode address matching phase.

When a bus error is detected, BERRFLG flag bit of I2C_STS register will be set to 1 by hardware; if ERRIEN bit of I2C_CTRL1 register is set to 1, an error interrupt will be generated.

Arbitration loss flag bit (ALFLG)

When a high level is transmitted on the SDA line, but the rising edge of SCL samples a low level from SDA, it will be detected as an arbitration loss error.

- In master mode, arbitration loss is detected in address phase, data phase and data validation phase. In such case, SDA and SCL lines will be released, the START control bit will be cleared by hardware, and the master mode will be automatically switched to slave mode.
- In slave mode, arbitration loss is detected in data phase and data validation phase. In this case, transmission is terminated and SCL and SDA lines are released.

When an arbitration loss error is detected, ALFLG flag bit of I2C_STS register will be set to 1 by hardware; if ERRIEN bit of I2C_CTRL1 register is set to 1, an error interrupt will be generated.

Overrun/Underrun error flag bit (OVRURFLG)

When clock extension is disabled, an underrun or overrun error will be detected under the following conditions in slave mode:

- When receiving, the RXDATA register has not been read, but the newly transmitted byte has been received.

- When transmitting, the first data byte should be transmitted, but STOPFLG=1. If TXBEFLG=0, the value of I2C_TXDATA register is transmitted; if it is not 0, 0xFF is transmitted.
- When transmitting, if a new byte should be written to I2C_TXDATA register, but it is not written, 0xFF will be transmitted.

When an overrun/underrun error is detected, OVRURFLG flag of I2C_STS register will be set to 1 by hardware; if ERRIEN bit of I2C_CTRL1 register is set to 1, an interrupt will be generated.

Packet error check error flag bit (PECEFLG)

This error condition is targeted only for SMBus function part. After receiving PEC byte not matching the content of I2C_PEC register, PEC error will be detected. After wrong PEC is received, a NACK will be returned automatically. When PEC error is detected, PECEFLG flag of I2C_STS register will be set to 1 by hardware; if ERRIEN bit of I2C_CTRL1 register is set to 1, an interrupt will be generated.

Timeout error flag bit (TTEFLG)

This error condition is targeted only for SMBus function part.

Timeout error will occur under the following conditions:

（1） SMBus timeout is detected

- IDLECLKTO=0 and the hold time of low SCL reaches the time defined by TIMEOUTA[11:0] bit field
- IDLECLKTO=1 and the high-level time of SDA and SCL exceeds the time defined by TIMEOUTA[11:0] bit field

（2） SMBus idle timeout is detected

- The accumulative time of low extension of master clock reaches the time ($t_{LOW:MEXT}$) defined by TIMEOUTB[11:0] bit field
- The accumulative time of low extension of slave clock reaches the time ($t_{LOW:SEXT}$) defined by TIMEOUTB[11:0] bit field

When a TIMEOUT error is detected, TTEFLG flag of I2C_STS register will be set to 1 by hardware; if ERRIEN bit of I2C_CTRL1 register is set to 1, an interrupt will be generated.

22.5.4.7 DMA request

DMA transmission can be enabled by setting DMATXEN bit of I2C_CTRL1 register. The data is put into the SRAM area set by DMA peripheral in advance and transmitted to I2C_TXDATA register (not needing to consider the state of TXINTFLG bit).

Only DMA is used to transmit bytes:

- Master mode: Initialization, slave address, direction, byte number and start bit are set by software (when the slave address has been transmitted, DMA cannot be used for transmission). When all data are transmitted by DMA, DMA must be initialized before START bit is set to 1.
- Slave mode: DMA must be initialized before the address matching event.

22.5.5 I2C interrupt

Table 77 Interrupt Request List

Interrupt event	Event flag bit	Method of clearing the event flag bit	Interrupt enable control bit
Received character is not empty	RXBNEFLG	Read I2C_RXDATA register	RXIEN
Transmit interrupt state	TXINTFLG	Write I2C_TXDATA register	TXIEN
Stop signal detection flag	STOPFLG	Write STOPCLR=1	STOPIEN
Transmission completion reload	TXCRFLG	Write I2C_CTRL2 and NUMBYT[7:0] is not 0	TXCIEN
Transmission completed	TXCFLG	Write START=1 or STOP=1	
Address match	ADDRMFLG	Write ADDRMCLR=1	SADDRMIEN
Receive NACK flag bit	NACKFLG	Write NACKCLR=1	NACKRXIEN
Bus error	BERRFLG	Write BERRCLR=1	ERRIEN
Arbitration loss	ALFLG	Write ALCLR=1	
Overrun/Underrun error	OVRURFLG	Write OVRURCLR=1	
PEC error	PECEFLG	Write PECECLR=1	
Clock timeout	TTEFLG	Write TTECLR=1	
SMBus reminder	SMBALTFLG	Write SMBALTCLR=1	

To enable I2C interrupt, it is required to:

- Configure and enable I2C channel in NVIC
- Configure I2C interrupt enable bit

22.6 Register address mapping

Table 78 I2C Register Address Mapping

Register name	Description	Offset address
I2C_CTRL1	Control register 1	0x00
I2C_CTRL2	Control register 2	0x04
I2C_ADDR1	Master address register 1	0x08
I2C_ADDR2	Master address register 2	0x0C
I2C_TIMING	Timing register	0x10

Register name	Description	Offset address
I2C_TIMEOUT	Timeout register	0x14
I2C_STS	Status register	0x18
I2C_INTFCLR	Interrupt flag clear register	0x1C
I2C_PEC	PEC register	0x20
I2C_RXDATA	Receive data register	0x24
I2C_TXDATA	Transmit data register	0x28

22.7 Register functional description

22.7.1 Control register 1 (I2C_CTRL1)

Offset address: 0x00

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	I2CEN	R/W	I2C Enable 0: Disable 1: Enable
1	TXIEN	R/W	Transmit Interrupt Enable 0: Disable 1: Enable
2	RXIEN	R/W	RX Interrupt Enable 0: Disable 1: Enable
3	SADDRMIEN	R/W	Slave Address Match Interrupt Enable 0: Disable 1: Enable
4	NACKRXIEN	R/W	NACK Received Interrupt Enable)\ 0: Disable 1: Enable
5	STOPIEN	R/W	STOP Detection Interrupt Enable 0: Disable 1: Enable
6	TXCIEN	R/W	Transmit Complete Interrupt Enable 0: Disable 1: Enable
7	ERRIEN	R/W	Error Interrupt Enable 0: Disable 1: When the bit of any of the following status register is enabled and set to 1, an interrupt will be generated: SMBALTFLG, TTEFLG, PECEFLG, OVRURFLG, ALFLG, and STS1_BERRFLG
11:8	DNFCFG	R/W	Digital Noise Filter Configure The digital noise filters of SDA and SCL are configured by this bit field. The length of digital filter is DNFCFG[3:0]*tI2C_CLK. 0000: Disable 0001: Enable; 1 tI2C_CLK ……

Field	Name	R/W	Description
			1111: Enable; 15 tI2C_CLK If the analog filter is enabled at the same time, the digital filter will be added to the analog filter; This bit can be set only when I2CEN is not set.
12	ANFD	R/W	Analog Noise Filter Disable 0: Enable 1: Disable This bit can be set only when I2CEN is not set.
13	SWRST	W	Software Reset When this bit is set to 1, the SCL and SDA lines will be released, the internal state machine, communication control bit and state bit will return to their reset values, and all other control bits will be reserved. This bit can be written as 1 only, and writing 0 is invalid. Read this bit and it will return 0.
14	DMATXEN	R/W	DMA Transmit Enable 0: Disable 1: Enable
15	DMARXEN	R/W	DMA Receive Enable 0: Disable 1: Enable
16	SBCEN	R/W	Slave Byte Control Enable 0: Disable 1: Enable
17	CLKSTRETCHD	R/W	Slave Mode Clock Stretching Disable 0: Enable 1: Disable This bit can be set only when I2CEN is not set, and it is applicable only to the slave mode.
18	WUPEN	R/W	Wakeup From Stop Mode Enable 0: Disable 1: Enable
19	RBEN	R/W	Responds Broadcast Enable The response broadcast address is 0x00. 0: Disable 1: Enable
20	HADDREN	R/W	SMBus Host Address Enable The HOST address is 0x10/0x11. 0: Disable 1: Enable If SMBus mode is not supported, this bit will be reserved and be forced to 0.
21	DEADDREN	R/W	SMBus Device Default Address Enable The default address is 0xC2/0xC3. 0: Disable 1: Enable

Field	Name	R/W	Description
			If SMBus mode is not supported, this bit will be reserved and be forced to 0.
22	ALTEN	R/W	SMBus Alert Function Enable Device mode (HADDREN=0): 0: Release SMBALERT pin and disable the notification response address header after NACK. 1: Pull down SMBALERT pin and enable the notification response address header after ACK. HOST mode (HADDREN=1): 0: Not support 1: Support If ALTEN=0, SMBALERT pin can be used as a GPIO; If SMBus mode is not supported, this bit will be reserved and be forced to 0.
23	PECEN	R/W	PEC Enable 0: Disable 1: Enable If SMBus mode is not supported, this bit will be reserved and be forced to 0.
31:24	Reserved		

22.7.2 Control register 2 (I2C_CTRL2)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	SADDR[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.
7:1	SADDR[7:1]	R/W	Slave Address Setup The 7:1 bit of slave address.
9:8	SADDR[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.
10	TXDIR	R/W	Master Mode Transfer Direction Setup 0: Write transmission 1: Read transmission
11	SADDRLEN	R/W	Slave Address Length Configure 0: 7-bit addressing mode 1: 10-bit addressing mode
12	ADDR10	R/W	Master Transmit 10-Bit Address Header Configure 0: Transmit 10-bit slave address read sequence: start bit + 2-byte 10-bit write direction address + restart + the first 7 bits of 10-bit read direction address. 1: Transmit the first 7 bits of 10-bit slave address read sequence + read direction.

Field	Name	R/W	Description
13	START	R/W	Start Bit Transmit This bit can be set to 1 and cleared by software; it can be cleared by hardware after the start bit and address sequence are transmitted, arbitration loss occurs, timeout error occurs or I2CEN bit is not set, or be cleared by setting ADDRMCLR bit of I2C_INTFCLR register. In master mode: 0: Not transmit 1: Transmit repeatedly In slave mode: 0: Not transmit 1: Transmit when the bus is idle It is meaningless to write 0 to this bit; Setting RELOAD bit and this bit does not work.
14	STOP	R/W	Stop Bit Transmit This bit can be set to 1 and cleared by software; it can be cleared by hardware when transmitting the stop bit or when I2CEN bit is not set. In master mode: 0: Not transmit 1: Transmit It is meaningless to write 0 to this bit.
15	NACKEN	R/W	Transmit NACK Enable) It can be set to 1 and cleared to 0 by software; when transmitting the stop bit, NACK, after receiving the address matching event or when I2CEN bit is not set, it can be cleared to 0 by hardware. 0: Transmit ACK 1: Transmit NACK It is meaningless to write 0 to this bit, and it is applicable only to the slave mode. In master receiving mode, it will be automatically transmitted after the last byte is transmitted and between transmitting the stop bit or RESTART bit. In slave receiving mode, NACK will be transmitted automatically when an overrun occurs. In this case, NACKEN bit does not work; After hardware PEC check is enabled, the confirmation value of PEC still does not depend on the value of NACK bit.
23:16	NUMBYT	R/W	Number of Bytes Setup This bit determines the number of bytes to be transmitted. This bit is meaningless when it is in slave mode and SBCEN=0. This bit can be set only when START bit is not set.
24	RELOADEN	R/W	NUMBYT Reload Mode Enable It can be set to 1 and cleared to 0 by software. 0: Transmission is over after transmission of NUMBYT bytes 1: Reload NUMBYT after transmission of NUMBYT bytes. After transmission of NUMBYT bytes, TXCFLG flag bit will be set and SCL will be pulled down.

Field	Name	R/W	Description
25	ENDCFG	R/W	End Mode Configure It can be set to 1 and cleared to 0 by software. 0: Software end mode: after transmission of NUMBYT data, TXCFLG flag bit will be set, and SCL will be pulled down. 1: Automatic end mode: after transmission of NUMBYT data, a stop bit will be transmitted automatically. This bit does not work when it is in slave mode or RELOADEN bit is set.
26	PEC	R/W	Transfer Packet Error Checking Byte Enable This bit can be set to 1 and cleared to 0 by software; it can be cleared to 0 by hardware after PEC transmission is completed, the stop bit is received, the address match event is received or when I2CEN bit is not set. 0: Disable 1: Enable It is meaningless to write 0 to this bit. Set RELOADEN bit or clear SBCEN bit in slave mode and this bit will not work; If SMBus mode is not supported, this bit will be reserved and be forced to 0.
31:27	Reserved		

22.7.3 Master address register 1 (I2C_ADDR1)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	ADDR1[0]	R/W	Master 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.
7:1	ADDR1[7:1]	R/W	Master Address Setup Master address No. 7:1 bit
9:8	ADDR1[9:8]	R/W	Master 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.
10	ADDR1LEN	R/W	Master Address Length Configure 0: 7-bit addressing mode 1: 10-bit addressing mode
14:11	Reserved		
15	ADDR1EN	R/W	Master Address 1 Enable 0: Disable. Transmit NACK after the slave address ADDR is received 1: Enable. Transmit ACK after the slave address ADDR is received
31:16	Reserved		

22.7.4 Master address register 2 (I2C_ADDR2)

Offset address: 0x0C

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	Reserved		
7:1	ADDR2[7:1]	R/W	Master Address Setup Master address No. 7:1 bit
10:8	ADDR2MSK	R/W	Masks Master Address 2 Select 000: No mask 001: Mask ADDR2[1], compared with ADDR2[7:2]. 010: Mask ADDR2[2:1], compared with ADDR2[7:3]. 011: Mask ADDR2[3:1], compared with ADDR2[7:4]. 100: Mask ADDR2[4:1], compared with ADDR2[7:5]. 101: Mask ADDR2[5:1], compared with ADDR2[7:6]. 110: Mask ADDR2[6:1], compared with ADDR2[7]. 111: Mask ADDR2[7:1], without comparison; all 7-bit addresses received will transmit ACK. This bit can be set only when ADDR2EN bit is not set; if ADDR2MSK is not 0, and the reserved I2C address does not response, matching is meaningless.
14:11	Reserved		
15	ADDR2EN	R/W	Master Address 2 Enable 0: Disable. Transmit NACK after the slave address ADDR2 is received. 1: Enable. Transmit ACK after receiving the slave address ADDR2.
31:16	Reserved		

22.7.5 Timing register (I2C_TIMING)

Offset address: 0x10

Reset value: 0x0000 0000

Field	Name	R/W	Description
7:0	SCLL	R/W	SCL Low Level Time Setup tSCLL =(SCLL+1) x tTIMINGPSC SCLL determines t_{BUF} and $t_{SU:STA}$ timing.
15:8	SCLH	R/W	SCL High Level Time Setup tSCLH =(SCLH+1) x tTIMINGPSC SCLH determines $t_{SU:STO}$ and $t_{HD:STA}$ timing.
19:16	DATAHT	R/W	Data Hold Time Setup This bit field determines the delay t_{DATAHT} between SCL falling edge and SDA edge in transmitting mode. tDATAHT=DATAHT x tTIMINGPSC DATAHT determines $t_{HD:DAT}$ timing.
23:20	DATAT	R/W	Data Time Setup This bit field determines the delay t_{DATAT} between SDA edge and SCL rising edge in transmitting mode. tDATAT=(DATAT+1) x tTIMINGPSC tDATAT determines $t_{SU:DAT}$ timing.
27:24	Reserved		

Field	Name	R/W	Description
31:28	TIMINGPSC	R/W	Timing Prescaler Setup This bit field divides the frequency of I2C_CLK, and provides clock cycle $t_{TIMINGPSC}$ for data setup, hold time counter and SCL high and low-level counter. tTIMINGPSC=(TIMINGPSC+1) x tI2C_CLK。

Note: This register can be set only when I2CEN bit is not set.

22.7.6 Timeout register (I2C_TIMEOUT)

Offset address: 0x14

Reset value: 0x0000 0000

Field	Name	R/W	Description
11:0	TIMEOUTA	R/W	Bus Timeout A Setup When IDLECLKTO=0, and SCL timeout is low: tTIMEOUT=(TIMEOUTA+1) x 2048 x tI2C_CLK When IDLECLKTO=1, and the bus is idle: tIDLE=(TIMEOUTA+1) x 4 x tI2C_CLK This bit can be set only when CLKTOEN bit is not set.
12	IDLECLKTO	R/W	Idle Clock Timeout Detection Configure 0: Detect SCL low-level timeout 1: Detect SCL and SDA high-level timeout (the bus is idle) This bit can be set only when CLKTOEN bit is not set.
14:13	Reserved		
15	CLKTOEN	R/W	Clock Timeout Enable 0: Disable 1: Enable. A timeout error is detected when the hold time of low SCL is more than $t_{TIMEOUT}$ or the hold time of high SCL is more than t_{IDLE}.
27:16	TIMEOUTB	R/W	Bus Timeout B Setup The accumulated master clock low extension time to be detected in master mode ($t_{LOW:MEXT}$). The accumulated slave clock low extension time to be detected in slave mode ($t_{LOW:SEXT}$). tTLOW:EXT=(TIMEOUTB+1)x2048xtI2C_CLK This bit field can be set only when EXCLKTOEN bit is not set.
30:28	Reserved		
31	EXCLKTOEN	R/W	Extended Clock Timeout Enable 0: Disable 1: Enable. A timeout error is detected when the hold time of low SC reaches $t_{TLOW:EXT}$.

22.7.7 Status register (I2C_STS)

Offset address: 0x18

Reset value: 0x0000 0001

Field	Name	R/W	Description
0	TXBEFLG	R/S	Transmit Data Buffer Empty Flag 0: The transmit buffer is not empty

Field	Name	R/W	Description
			1: The transmit buffer is empty This bit is set to 1 by hardware when the content of I2C_TXDATA register is empty; this bit is cleared when the data to be transmitted are written to I2C_TXDATA register. This bit can be set to 1 by software to clear I2C_TXDATA register; when I2CEN=0, this bit is cleared to 0 by hardware.
1	TXINTFLG	R/S	Transmit Interrupt State Flag 0: Not transmit 1: Transmit This bit is set to 1 by hardware when I2C_TXDATA register is empty; then write the data to be transmitted to I2C_TXDATA register. This bit can be cleared by writing the data to be transmitted to I2C_TXDATA register. This bit can be set to 1 by software when CLKSTRETCHD=1, so as to generate TXINTFLG flag bit; it can be cleared by to 0 hardware when I2CEN=0.
2	RXBNEFLG	R	Receive Data 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 are data in RXDATA register; this bit can be cleared by reading I2C_RXDATA; and be cleared to 0 by hardware when I2CEN=0.
3	ADDRMFLG	R	Slave Address Match Flag 0: The slave address does not match 1: The slave address matches When the received slave address matches any valid slave address, this bit is set to 1 by hardware. This bit can be cleared to 0 by software by setting ADDRMCLR bit to 1; or be cleared to 0 by hardware when I2CEN=0.
4	NACKFLG	R	Receive Not Acknowledge Flag 0: NACK flag is not received 1: NACK flag is received This bit can be set to 1 by hardware when one byte is transmitted and NACK is received. It can be cleared to 0 by software by setting NACKCLR bit to 1; or be cleared to 0 by hardware when I2CEN=0.
5	STOPFLG	R	Stop Bit Detection Flag 0: No stop bit is detected 1: Stop bit is detected This bit can be set to 1 by hardware when the peripheral participates in transmission and the stop bit is detected on the bus. This bit can be cleared by software if the peripheral transmits the stop bit as the master or the peripheral is addressed correctly as the slave before this transmission, and STOPCLR=1; or be cleared by hardware when I2CEN=0.
6	TXCFLG	R	Transmit Data Complete Flag 0: Transmitting data is not completed 1: Transmitting data is completed

Field	Name	R/W	Description
			This bit can be set to 1 by hardware when RELOADEN=0, ENDCFG=0 and NUMBYT data have been transmitted ; be cleared when START=1 or STOP=1; or be cleared by hardware when I2CEN=0.
7	TXCRFLG	R	Transfer Complete Reload Flag 0: Transmission is completed 1: Transmission is completed to reload This bit can be set to 1 by hardware when RELOADEN=1 and NUMBYT data have been transmitted; it can be cleared to 0 by software by writing a non-zero value to NUMBYT; or be cleared to 0 by hardware when I2CEN=0. This bit works only in master mode, or in slave mode when SBCEN=1.
8	BERRFLG	R	Bus Error Flag 0: No bus error 1: Bus error occurred This bit can be set to 1 by hardware when wrong start bit or stop bit is detected; be cleared to 0 by software by setting BERRCLR bit; or be cleared to 0 by hardware when I2CEN=0.
9	ALFLG	R	Arbitration Lost Flag 0: No arbitration loss 1: Arbitration loss occurred This bit can be set to 1 by hardware when bus arbitration loss occurs; be cleared to 0 by software by setting ALCLR bit; or be cleared to 0 by hardware when I2CEN=0.
10	OVRURFLG	R	Overrun/Underrun Flag 0: No overrun/underrun 1: Overrun/Underrun occurred This bit can be set to 1 by hardware if overrun/underrun error occurs in slave mode when CLKSTRETCHD=1; be cleared to 0 by software by setting OVRURCLR bit; and be cleared to 0 by hardware when I2CEN=0.
11	PECEFLG	R	PEC Error in Reception Flag 0: No PEC error 1: PEC error occurred This bit can be set to 1 by hardware when the received PEC value does not match the value of PEC register. A NACK will be transmitted automatically when wrong PEC is received. This bit can be cleared to 0 by software by setting PECECLR bit; and be cleared to 0 by hardware when I2CEN=0. If SMBus mode is not supported, this bit will be reserved and be forced to 0 by hardware.
12	TTEFLG	R	Timeout or Tlow Error Flag 0: No timeout error 1: Timeout error occurred This bit can be set to 1 by hardware when timeout or external clock timeout occurs; be cleared to 0 by software by setting TTECLR bit; and be cleared to 0 by hardware when I2CEN=0. If SMBus mode is not supported, this bit will be reserved and be forced to 0 by hardware.

Field	Name	R/W	Description
13	SMBALTFLG	R	SMBus Alert Occur Flag 0: No SMBus alarm 1: SMBus alarm occurred This bit can be set to 1 by hardware if HADDREN=1 (configured by SMBus HOST) and ALTEN=1, and SMBALERT falling edge is detected on SMBALERT pin; be cleared to 0 by software by setting SMBALTCLR bit; and be cleared to 0 by hardware when I2CEN=0. If SMBus mode is not supported, this bit will be reserved and be forced to 0 by hardware.
14	Reserved		
15	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 a start bit is detected; be cleared to 0 by hardware when a stop bit is detected; or be cleared to 0 when I2CEN=0.
16	TXDIRFLG	R	Transfer Direction Flag Update when the address matching event occurs. 0: Write transmission; the slave enters receiving mode 1: Read transmission; the slave enters transmitting mode
23:17	ADDRCMFLG	R	Address Code Match Flag The received address is updated when the address match event occurs. 0: The address code does not match 1: The address code matches In 10-bit address, ADDRCMFLG provides the address after the first two bits of 10-bit address.
31:24	Reserved		

22.7.8 Interrupt flag clear register (I2C_INTFCLR)

Offset address: 0x1C

Reset value: 0x0000 0000

Field	Name	R/W	Description
2:0	Reserved		
3	ADDRMCLR	W	Slave Address Match Flag Clear Set this bit, and the ADDRMFLG flag bit of I2C_STS register and START bit of I2C_CTRL2 register will be cleared.
4	NACKCLR	W	Receive Not Acknowledge Flag Clear Set this bit and NACKFLG flag bit of I2C_STS register will be cleared.
5	STOPCLR	W	Stop Bit Detection Flag Clear Set this bit and STOPFLG flag bit of I2C_STS register will be cleared.
7:6	Reserved		

Field	Name	R/W	Description
8	BERRCLR	W	Bus Error Flag Clear Set this bit and BERRFLG flag bit of I2C_STS register will be cleared.
9	ALCLR	W	Arbitration Lost Flag Clear Set this bit and ALFLG flag bit of I2C_STS register will be cleared.
10	OVRURCLR	W	Overrun/Underrun Flag Clear Set this bit and OVRURFLG flag bit of I2C_STS register will be cleared.
11	PECECLR	W	PEC Error in Reception Flag Clear Set this bit and PECEFLG flag bit of I2C_STS register will be cleared. It SMBus mode is not supported, this bit will be reserved and be forced to 0 by hardware.
12	TTECLR	W	Timeout or Tlow Error Flag Clear Set this bit and TTEFLG flag bit of I2C_STS register will be cleared. It SMBus mode is not supported, this bit will be reserved and be forced to 0 by hardware.
13	SMBALTCLR	W	SMBus Alert Occur Flag Clear Set this bit and SMBALTFLG flag bit of I2C_STS register will be cleared. It SMBus mode is not supported, this bit will be reserved and be forced to 0 by hardware.
31:14	Reserved		

22.7.9 PEC register (I2C_PEC)

Offset address: 0x20

Reset value: 0x0000 0000

Field	Name	R/W	Description
7:0	PEC	R	PEC Value Setup When PECEN=1, this bit field means the internal PEC value. This bit can be cleared to 0 by hardware when I2CEN=0.
31:8	Reserved		

22.7.10 Receive data register (I2C_RXDATA)

Offset address: 0x24

Reset value: 0x0000 0000

Field	Name	R/W	Description
7:0	RXDATA	R	8-Bit Receive Data Byte Data byte received from I2C bus.
31:8	Reserved		

22.7.11 Transmit data register (I2C_TXDATA)

Offset address: 0x28

Reset value: 0x0000 0000

Field	Name	R/W	Description
7:0	TXDATA	R/W	8-Bit Transmit Data Byte Data byte to be transmitted to I2C bus. This bit field can be set only when TXBEFLG=1.
31:8	Reserved		

23 Serial peripheral interface/Inter-IC sound interface (SPI/I2S)

23.1 Full name and abbreviation of terms

Table 79 Full Name and Abbreviation Description of SPI 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
Inter-IC Sound	I2S
Transmit	TX
Receive	RX
Busy	BSY

23.2 Introduction

SPI interface can be configured to support SPI protocol and I2S audio protocol. It works in SPI mode by default, and the functions can be switched in I2S mode by software.

Serial peripheral interface (SPI) provides data transmitting and receiving functions based on SPI protocol, which allows chips to communicate with external devices in half duplex, full duplex, synchronous and serial modes, and can work in master or slave mode.

The inter-IC sound interface (I2S) supports four audio standards: Philips I2S standard, MSB alignment standard, LSB alignment standard and PCM standard. It can work in master/slave mode during half-duplex communication.

23.3 Main characteristics of SPI

（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 4-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） SPI supports Motorola mode

（8） Fast communication in master/slave mode, up to 18MHz

（9） Clock polarity and phase are programmable

（10） Data sequence is programmable; select MSB or LSB in front

（11） An interrupt can be triggered by master mode fault, overrun and CRC error flag

（12） Have DMA transmit and receive buffers

（13） Calculate, transmit and verify by hardware CRC

（14） CRC error flag

（15） Two 32-bit embedded RXFIFO and TXFIFO have DMA function

23.4 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

（9） The master clock can output to an external audio component

23.5 SPI functional description

23.5.1 Description of SPI signal line

Table 80 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
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. Hardware NSS mode of master device hardware: NSS outputs, in single-master mode, NSS OFF output: Operation of multiple master environments is allowed, Slave hardware NSS mode: The NSS signal is set to low as the chip selection signal of the slave

23.5.2 Communication format

In SPI communication, receiving data and transmitting data can be carried out at the same time. SCK transmits and samples the data on the data line synchronously. The communication format depends on the clock phase, clock polarity and data frame format. If the communication is normal, the master device and the slave device must be in the same communication format.

23.5.2.1 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 81 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

Note:

（1）To change CPOL and CPHA bits, SPI must be cleared to 0 and disabled by SPIEN bit.

（2）When SCK is in idle state, if CPOL=1, pull up SCK; if CPOL=0, pull down SCK.

23.5.2.2 Data frame format

Select LSB or MSB first by configuring LSBSEL bit of SPI_CTRL1 register. Select the data word length by configuring DSCFG bit of SPI_CTRL2 register; no matter which data word length is selected, it must be aligned with FRTCFG when performing read access to FIFO. When accessing SPI_DATA register, the data frames are always right aligned. In the process of communication, only the bits within the data word length range will be output with the clock.

23.5.3 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.

23.5.4 SPI mode

23.5.4.1 Initialization of SPI master mode

In master mode, serial clock is generated on SCK pin.

Configure master mode

- Configure MSMCFG=1 in SPI_CTRL1 register, and set to master mode
- Select the serial clock baud rate by configuring BRSEL bit in SPI_CTRL1 register
- Select the polarity and phase by configuring CPOL and CPHA bits in SPI_CTRL1 register
- Select the transmission mode by configuring RXOMEN, BMOEN and BMEN bits in SPI_CTRL1 register
- Select the data bit width by configuring DSCFG bit in SPI_CTRL2 register
- Enable NSS pulse mode by configuring NSSPEN bit in SPI_CTRL2 register (when configuring this bit, CPHA bit must be set to 1)
- Set RXFIFO threshold value for trigging RXBNEFLG event by configuring FRTCFG bit in SPI_CTRL2 register
- If DMA function is used, it is required to configure LDTX and LDRX bits of SPI_CTRL2 register
- If CRC is used, it is required to set CRC polynomial as input and also set CRCEN bit
- 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 SPIEN bit in SPI_CTRL1 register to enable SPI

In master mode: MOSI pin is data output, while MISO is data input.

23.5.4.2 Initialization of SPI slave mode

In slave mode, SCK pin receives the serial clock from the master device.

Configuration of slave mode

- Configure MSMCFG=0 in SPI_CTRL1 register, and set to slave mode
- Select the polarity and phase by configuring CPOL and CPHA bits in SPI_CTRL1 register
- Select the transmission mode by configuring RXOMEN, BMOEN and BMEN bits in SPI_CTRL1 register
- Select the data bit width by configuring DSCFG bit in SPI_CTRL2 register
- Enable NSS pulse mode by configuring NSSPEN bit in SPI_CTRL2 register (when configuring this bit, CPHA bit must be set to 1)

- Set RXFIFO threshold value for trigging RXBNEFLG event by configuring FRTCFG bit in SPI_CTRL2 register
- If DMA function is used, it is required to configure LDTX and LDRX bits of SPI_CTRL2 register
- If CRC is used, it is required to set CRC polynomial as input and also set CRCEN bit
- 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
- Configure SPIEN bit in SPI_CTRL1 register to enable SPI

In slave mode: MOSI pin is data input, while MISO is data output.

23.5.4.3 Full-duplex communication of SPI

Usually, SPI is configured as full-duplex communication, and the master and the slave shift registers are connected through two unidirectional lines MOSI and MISO. During SPI communication, synchronous data transmission is conducted according to SCK clock edge. The data of the master are transmitted to the slave through MOSI pin, and the data of the slave are transmitted to the master through MISO pin. When the data transmission is completed, it means that the information is exchanged successfully.

23.5.4.4 Half-duplex communication of SPI

One clock line and one bidirectional data line

- Enable this mode by setting BMEN bit 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

23.5.4.5 Simplex communication of SPI

One clock line and one unidirectional data line (receive-only or transmit-only)

In this mode, SPI mode is used as receive-only or transmit-only.

Transmit-only mode:

- Data are transmitted on transmitting pin (MOSI in master mode, MISO in slave mode)
- Then the receive pin can be used as general-purpose I/O (MISO in master mode, MOSI in slave mode)

Receive-only mode:

- Disable SPI output function by setting RXOMEN bit in SPI_CTRL1 register
- Release the transmit pin (MOSI in master mode, MISO in slave mode)
- In master mode, enable SPI to start communication, clear SPIEN bit of SPI_CTRL1 register and receiving data can be stopped immediately, not needing to read BSYFLG flag (always 1)
- In slave mode: Pull NSS to low, and as long as SCK is pulsed by clock, SPI will always receive

23.5.4.6 Communication of multiple slave devices of SPI

SPI can be operated by multiple slave devices. The master device uses GPIO pin to manage the chip selection line of the slave device, and can control two or more independent slave devices.

The master device decides using which slave device to transmit data by pulling down the NSS pin of the slave device.

23.5.5 Data transmission and receiving process in different modes of SPI

Table 82 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; MISO transmits
Unidirectional receiving mode of slave device	BMEN=0，RXOMEN=1	MOSI receives; MISO is not used
Bidirectional transmitting mode of slave device	BMEN=1，BMOEN=1	MOSI is not used; MISO transmits
Bidirectional receiving mode of slave device	BMEN=1，BMOEN=0	MOSI receives; MISO is not used

Figure 94 Connection in Full Duplex Mode

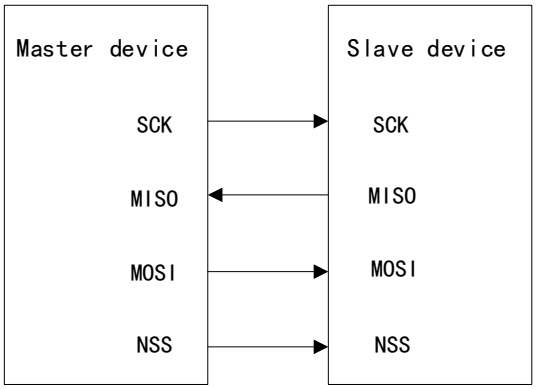


Figure 95 Connection in Half-duplex Mode

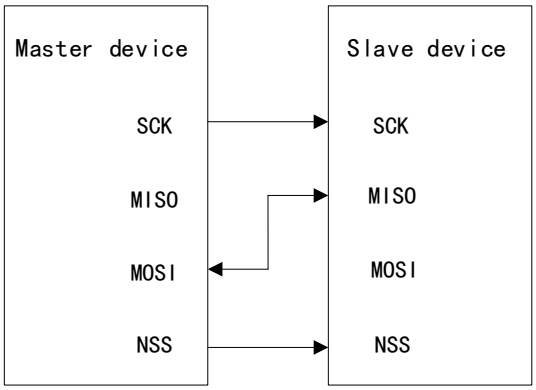


Figure 96 Connection in Simplex Mode (the master is used for receiving, while the slave is used for transmitting)

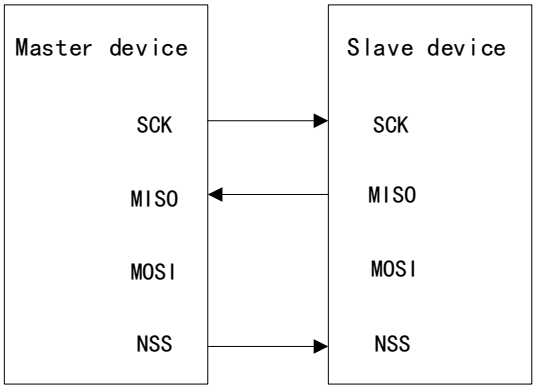
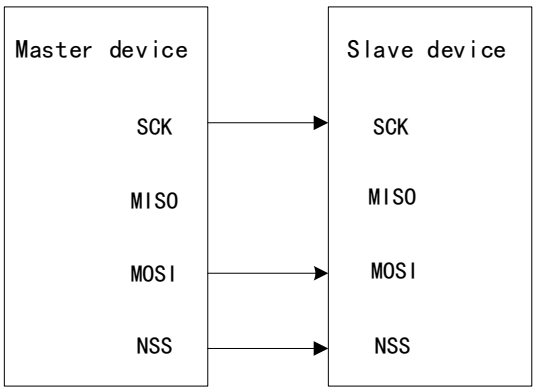


Figure 97 Connection in Simplex Mode (the master only transmits, while the slave receives)



23.5.5.1 Transmitting and receiving of data

In order to prevent overrun when the data frame is short and ensure that SPI can work continuously, all SPI data need to pass through the 32-bit embedded FIFO. Each direction will have its own FIFO, TXFIFO and RXFIFO.

Handle FIFO according to SPI simplex and duplex mode, data frame format, access size executed on FIFO data register and whether to use data package to process FIFO when accessing FIFO.

After read access to SPI_DATA register, the earliest values that have not been read yet and are stored in RXFIFO will be returned. After write access to SPI_DATA, the written data will be stored in TXFIFO at the end of the transmit queue. Read access must always be aligned with RXFIFO threshold value configured by FRTCFG bit in SPI_CTRL2 register. The FTLSEL and FRLSEL bits indicate the current occupancy levels of the two FIFO.

The read access to SPI_DATA register must be managed by RXBNEFLG event. When the data are stored in RXFIFO and reach the threshold value (defined by previous bit), this event will be triggered; when RXBNEFLG is cleared, RXFIFO will be regarded to be empty, and in the similar way, the write access to the data frame to be transmitted is managed by TXBEFLG event. When TXFIFO is less than or equal to half of its capacity, RXBNEFLG event will be triggered; otherwise, TXBEFLG will be cleared, meanwhile, it will be regarded that there are data stored in TXFIFO. Therefore, when the data frame format is less than or equal to one byte, RXFIFO can store 4 data frames at most, and TXFIFO can store 3 data frames. When the software attempts to write more data to TXFIFO in 16-bit mode, this difference can prevent the three or 8-bit data frames that have been stored in TXFIFO from being damaged. TXBEFLG and RXBNEFLG events can be polled or handled by interrupt.

23.5.5.2 Sequence processing

In transmitting data, multiple data can be formed into a sequence in order.

When the transmission is started, TXFIFO will transmit continuously in order.

In single receive mode, in half-duplex or simplex mode, when SPI is enabled, the master device will immediately receive the sequence until SPI is disabled or the single receive mode is disabled . When the data frame starts transmission, the slave cannot control the data sequence, so the slave must prepare the data before the transmission, to ensure there are data to be transmitted in TXFIFO.

When there are multiple slave devices, each sequence needs to correspond to different slave devices, so NSS pulse should be used to separate the sequence to ensure it is correct.

Note:

（1）Check whether the data transmission is completed according to FTLSEL bit and BSYFLG bit, and the clock output will stop when the transmission is completed.

（1）In packet mode, special attention should be paid to empty bytes when the data being transmitted are odd.

（2）In single receive mode of the master device, it is required to disable SPI or single receive mode to stop clock output.

（3）Master the correct receiving time to ensure the correct data transmission

（4）The action of disabling should be performed between the sampling time of first bit and the first bit of the next byte.

23.5.5.3 Data packing

If the data frame is less than or equal to one byte, when executing 16-bit read and write access to SPI_DATA register, the data will be packed automatically and double data can be processed in parallel. After conducting write access to SPI_DATA, 2-byte data will be transmitted; if the threshold value of RXFIFO is set to 16 bits, a receive RXBNEFLG event will be generated.

For a single RXBNEFLG event, the data receiver shall perform one read operation to SPI_DATA, and only after that, can it obtain all data.

Note: The threshold value of RXFIFO should be consistent with the bit width of follow-up data access.

23.5.6 NSS pulse mode

NSS pulse mode can be set by configuring NSSPEN bit of SPI_CTRL1 register; this mode takes effect only when SPI is configured as Motorola master mode and captures the first edge. In transmitting of this mode, NSS pulse is generated between two continuous data frames, and NSS will remain high for at least one cycle. NSS pulse mode allows the slave to latch data.

23.5.7 TI mode

Master mode of TI protocol

SPI interface can be made compatible with master mode of TI protocol by configuring FRFCFG bit of SPI_CTRL2 register.

In master mode of TI protocol, it is unaffected by the setting of SPI_CTRL1 register, and the clock polarity, phase and NSS management will meet the requirements of TI protocol. In slave mode, SPI baud rate frequency divider is used to control MISO pin to make MISO pin in high-impedance state, and any baud rate can be used, ensuring the best flexibility.

Generally the baud rate is set as the baud rate of external master clock, and the delay for MISO signal to become the high-impedance state depends on the baud rate set synchronously and through BRSEL bit of SPI_CTRL1 register internally. The formula is:

$$T_{baud\_rate}/2+4\times t_{pclk}<t_{release}<t_{baud\_rate}+6\times t_{pclk}$$

Note: This function does not apply to Motorola SPI communication mode (FRFCFG bit is set to 0)

23.5.8 CRC functions

SPI module contains two CRC computing units, which are used for data receiving and data transmission respectively.

CRC computing units are used to define polynomials in SPI_CRCPOLY register (it should be an odd number, and does not support even number).

Enable CRC computing by configuring CRCEN bit in SPI_CTRL1 register; at the same time, reset the CRC register (SPI_RXCRC and SPI_TXCRC).

CRC is managed by CPU during transmission

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; during CRC data transmission, CRC computing will be frozen.

The received CRC data will be stored in RXFIFO. A CRC transaction usually needs one more data frame to communicate at the end of the data sequence. However, when an 8-bit data frame checked by 16-bit CRC is set, two data frames are needed to transmit the complete CRC. When the last CRC data is received, the received value and the value of SPI_RXCRC register will be compared. By checking CRCEFLG flag bit in SPI_STS register, judge whether the data are destroyed in the process of transmission. CRCEFLG bit can be cleared by writing 0. RXBNEFLG bit can be cleared by reading SPI_DATA register.

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)

Note: When SPI works in slave mode, the software must enable CRC operation when the clock is stable. During the peirod of the data phase and CRC phase, the NSS signal needs to be pulled down and maintained.

23.5.9 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 needed to enable DMA transmission channel.

When SPI only receives data, it is only needed to enable DMA receiving channel.

DMA function of SPI mode can be enabled by configuring TXDEN and RXDEN bits of SPI_CTRL2 register.

- When transmitting: When TXBEFLG flag bit is set to 1, issue the DMA request, DMA controller writes data to SPI_DATA, 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. The CRCNXT bit is not controlled by software. The transmitting DMA channel counter of SPI must be set to the number that does not contain CRC data, but the DMA channel counter must contain the length of one more CRC data when receiving.

After reading CRC data in CRC check link, the values of SPI_TXCRC and SPI_RXCRC will be cleared to 0 automatically. Then continuous transmission

can be realized by DMA circular mode (except in single-receive mode).

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.

23.5.10 Disable SPI

After data transmission is over, end the communication by disabling SPI module.

When data are being transmitted or there are data in TXFIFO, it is not allowed to disable SPI by operating SPIEN bit in SPI_CTRL1 register. If SPIEN=0 is set, the clock signal will be transmitted continuously until the peripheral is enabled again. Certain steps are required to disable SPI to prevent the above situations.

Steps of disabling SPI

（1） Wait for clearing FTLSEL to 0

（2） Wait for clearing BSYFLG flag bit to 0

（3） Wait for clearing FRLSEL to 0

（4） Disable SPI (SPIEN=0)

Steps of disabling SPI in some single-receive mode

（1） Wait for clearing RXOMEN to 0 or setting BMOEN to 1

（2） Wait for clearing BSYFLG flag bit to 0

（3） Wait for clearing FRLSEL to 0

（4） Disable SPI (SPIEN=0)

23.5.11 SPI interrupt

An interrupt can be triggered by the following events during SPI operation:

- TXFIFO ready for loading
- RXFIFO receives data
- Master mode error
- CRC error
- TI frame format error

23.5.11.1 Status flag bit

There are three flag bits for fully monitoring the status of SPI bus

Transmit buffer idle flag TXBEFLG

TXBEFLG=1 means that TXFIFO has space to store the transmitted data; TXBEFLG flag bit is connected to TXFIFO bit, and in the process of storing data, if the storage content of TXFIFO is less than or equal to FIFO/2,

TXBEFLG flag bit remains high. When the storage content of TXFIFO is greater than FIFO/2, TXBEFLG flag bit will be cleared to 0. If TXBEIEN bit in SPI_CTRL2 register is set, an interrupt will be generated.

Receive buffer non-empty flag RXBNEFLG

RXBNEFLG flag bit depends on the value of FRTCFG bit in SPI_CTRL2 register:

- If FRTCFG=1, when the storage content of RXFIFO is greater than or equal to 8 bits, RXBNEFLG=1
- If FRTCFG=1, when the storage content of RXFIFO is greater than or equal to 16 bits, RXBNEFLG=1

RXBNEFLG flag bit will be cleared to 0 automatically if not in the above situations.

If RXBNEIEN=1 in SPI_CTRL2 register, an interrupt will be generated.

Busy flag BSYFLG

BSYFLG flag is set and cleared by hardware, which can indicate the state of SPI communication layer. When BSYFLG=1, it indicates SPI is communicating. BSYFLG flag can be used to detect whether transmission is over to avoid destroying the last transmitted data.

BSYFLG flag will be cleared to 0 in the following situations

- End the transmission in master mode
- Master mode fault
- In slave mode, there is at least one SPI cycle between two data transmissions
- Disable SPI

During continuous communication:

- 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 best to use TXBEFLG and RXBNEFLG flags to process the transmitting and receiving of each data item

23.5.11.2 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.

Effect of master mode failure: MEFLG is set to 1, and if ERRIEN is set, SPI

interrupt will be generated; SPIEN is cleared to 0 (output stops, SPI interface is disabled); MSMCFG is cleared to 0 and the device is forced to enter the slave mode.

Operation of clearing the MEFLG flag bit: When MEFLG flag bit is set to 1, it is required to read or write SPI_STS register, and then write SPI_CTRL1 register.

When MEFLG flag bit is 1, it is not allowed to set SPIEN and MSMCFG bits.

Overrun error OVRFLG

An overrun error will be generated when the following events occur

- When RXBNEFLG flag bit is still 1 after the master device has transmitted data
- When the space in RXFIFO cannot store the data to be received when receiving data
- When the software or DMA cannot read the data in RXFIFO in time
- When CRC is only enabled in receiving mode, RXFIFO is not available and the receive buffer is limited to the single data frame buffer

When an overrun error occurs: OVRFLG bit is set to 1; if ERRIEN bit is also set, an interrupt will be generated.

After an overrun error occurs, the data in the receiving buffer are not the data transmitted by the master device, and by reading SPI_DATA value, the data are the data not read before, and the subsequent data will be discarded.

OVRFLG flag can be cleared by reading SPI_DATA register and SPI_STS register according to the sequence.

CRC error flag bit CRCEFLG

Enable CRC operation by setting CRCEN bit of SPI_CTRL1 register, and CRC error flag can be used to check whether the received data are 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 mode frame format error (FREFLG)

Under the slave device and in accordance with TI mode protocol, when a pulse appears in NSS during data communication, a TI mode frame format error will be caused. When TI mode frame format error occurs, FREFLG flag bit of SPI_STS register will be set to 1, SPI will not be disabled, NSS pulse will be ignored, and SPI will wait for the next NSS pulse before retransmission. As the

error detection may cause the loss of two data bytes, the data may have been destroyed.

FREFLG flag can be cleared by reading SPI_STS register. If ERRIEN bit is set, an interrupt will be generated when NSS error occurs. At this time, SPI is disabled, which is because the consistency of data cannot be guaranteed. When SPI is enabled again, the master server needs to be reinitialized.

Table 83 SPI Interrupt Request

Interrupt flag	Interrupt event	Enable control bit	Clearing method
TXBEFLG	Transmit buffer empty flag	TXBEIEN	Write SPI_DATA register
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 mode frame format error		Read SPI_STS register

23.6 I2S functional description

Enable I2S function by setting I2SMOD bit of SPI_I2SCFG.

I2S and SPI share four pins:

- SD: Serial data, transmitting and receiving the data of 2-way time division multiplexing channel
- WS: Chip selection, switching the data of left and right channels
- CK: Serial clock; the clock signal is output in master mode, and is input in slave mode
- MCK: Master clock; in master mode, when MCOEN bit of SPI_I2SPSC register is set to 1, it can be used as the pin for outputting the extra clock signal.

23.6.1 I2S audio standard

I2S audio standard is selected by setting I2SSSEL bit and PFSSEL bit of SPI_I2SCFG register, and four audio standards can be selected in total: I2S Philips standard, MSB alignment standard, LSB alignment standard and PCM standard. Except PCM standard, other audio standards have two channels: left and right channels.

The data length and channel length can be configured by DATALEN and CHLEN bits in SPI_I2SCFG register. The channel length must be greater than

or equal to the data length. There are four data formats to transmit data: 16-bit data packed into 16-bit frame, 16-bit data packed into 32-bit frame, 24-bit data packed into 32-bit frame, and 32-bit data packed into 32-bit frame.

When the 16-bit data is extended to 32 bits, the first 16 bits are valid data, and the last 16 bits are forced to be 0. No external intervention is needed in this process.

Since the data buffers used for transmitting and receiving are all 16 bits, SPI_DATA needs to read/write twice when 24-bit and 32-bit data are transmitted. If DMA is used, DMA transmission twice is required.

For all communication standards and data formats, the most significant bit of data is always transmitted first.

For time division multiplexing, the left channel is always transmitted first, and then the right channel is transmitted.

23.6.1.1 I2S Philips standard

In I2S Philips standard, the pin WS can indicate the data being transmitted comes from the left channel or the right channel.

In I2S Philips standard, both WS and SD change on the falling edge of CK clock signal.

The sender will change the data on the falling edge of the clock signal CK, while the receiver will change the data on the rising edge of the clock signal CK.

Figure 98 I2S Philips Protocol Waveform (16/32 bits)

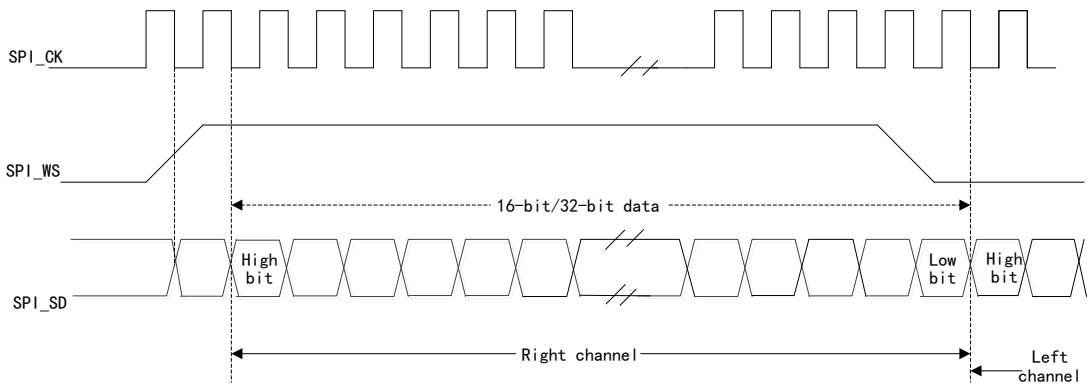
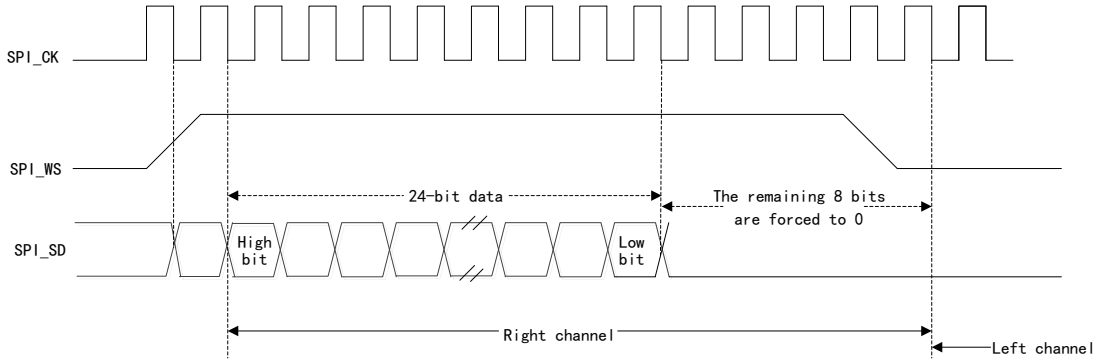


Figure 99 I2S Philips Protocol Waveform (24 bits)



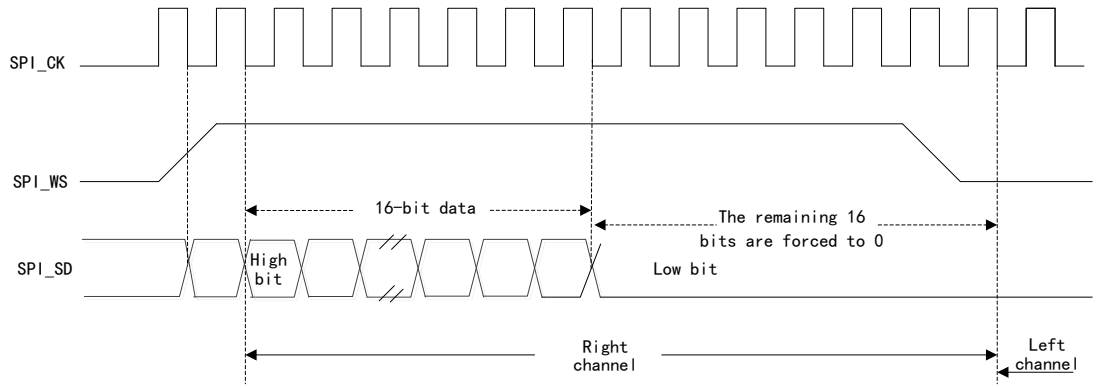
In I2S Philips standard, if you want to transmit/receive 24-bit and 32-bit data, the SPI_DATA register needs to read/write twice; for example:

- If you need to transmit 0x9FBB88 (24-bit data), write 0x9FBB to SPI_DATA register for the first time, and write 0x88XX to the register for the second time.
- If you need to receive 0x9FBB88 (24-bit data), read out 0x9FBB from SPI_DATA register for the first time and read out 0x8800 from the register for the second time.

In I2S configuration, when selecting the frame format of extending from 16-bit data to 32-bit data frame, it is required to access SPI_DATA register, and the remaining 16-bit data will be set to 0x0000 by hardware by force; for example:

- The data to be received or transmitted is 0x62d8, which becomes 0x62D80000 after it is extended to 32 bits, and it is necessary to write 0x62D8 to SPI_DATA register or read out from SPI_DATA register.

Figure 100 I2S Philips Protocol Waveform (extending from 16 bits to 32 bits)



In the transmission process, the MSB should be written to the register SPI_DATA, and when TXBEFLG flag bit is set to 1, new data can be written; if there is corresponding interrupt, an interrupt can be generated.

In the receiving process, every time the MSB is received, the RXBNEFLG flag bit will be set to 1; if there is corresponding interrupt, an interrupt can be generated.

23.6.1.2 MSB alignment standard

In MSB standard, WS signal and the first data bit are generated at the same time

In the transmission process, the data is changed on the falling edge of the clock signal; in the receiving process, the data is read on the rising edge of the clock signal.

Figure 101 MSB Alignment Standard Waveform (16/32-bit data)

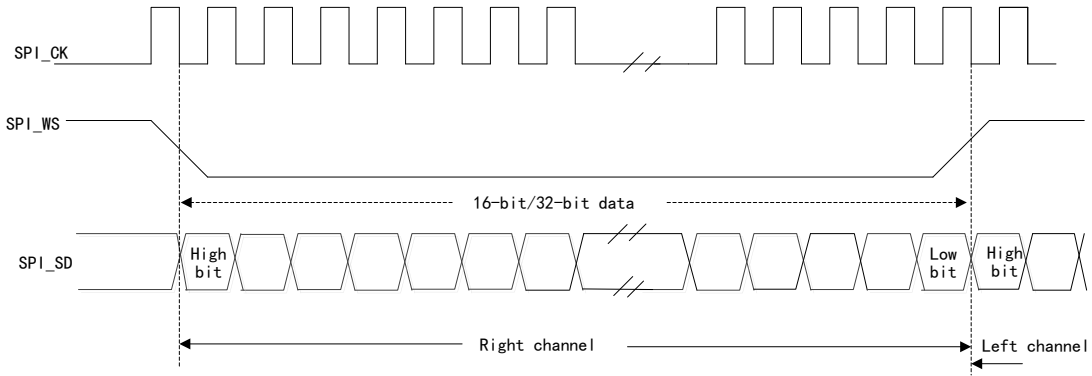


Figure 102 MSB Alignment Standard Waveform (24-bit data)

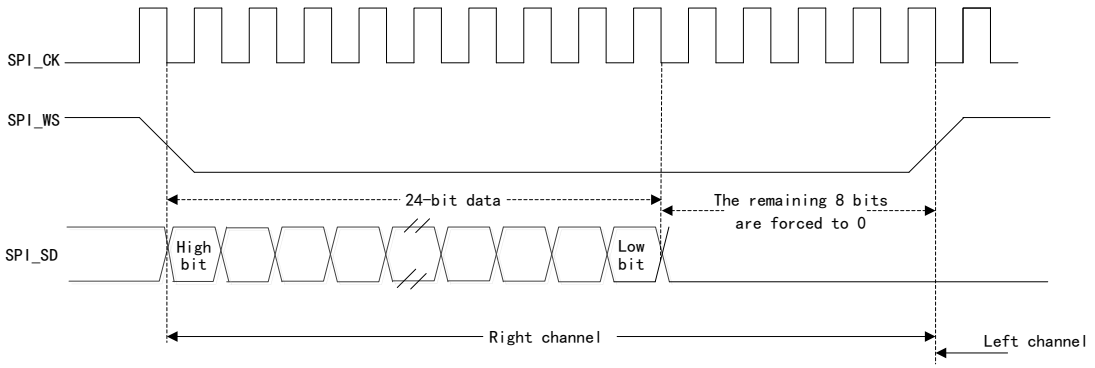
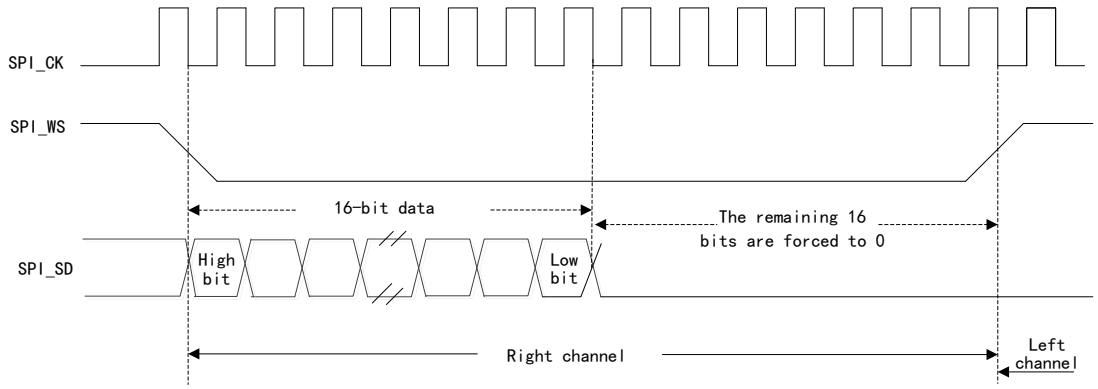


Figure 103 MSB Alignment Standard Waveform (extending from 16 bits to 32 bits)



23.6.1.3 LSB alignment standard

In the transmission process of LSB alignment standard, the data is changed on

the falling edge of the clock signal; in the receiving process, the data is read on the rising edge of the clock signal. When the channel length is the same as the data length, the LSB alignment standard is the same as the MSB alignment standard. If the channel length is larger than the data length, the valid data of the LSB alignment standard is aligned with the least significant bit.

Figure 104 LSB Alignment Standard Waveform (16/32-bit data)

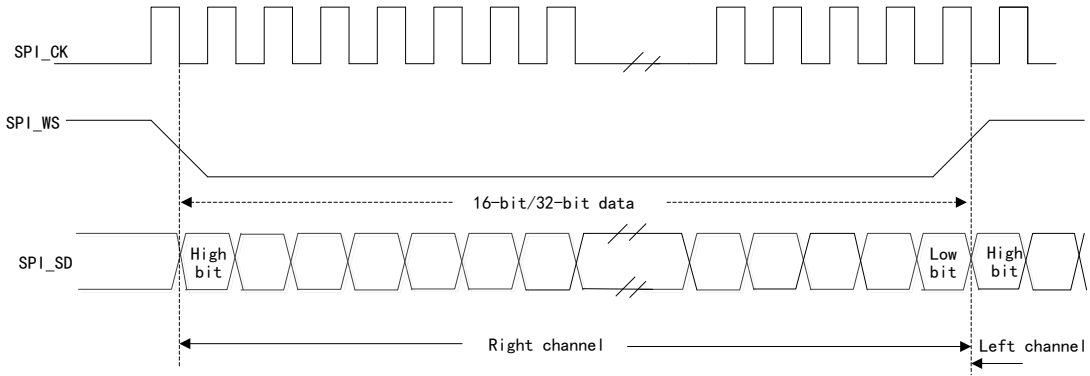
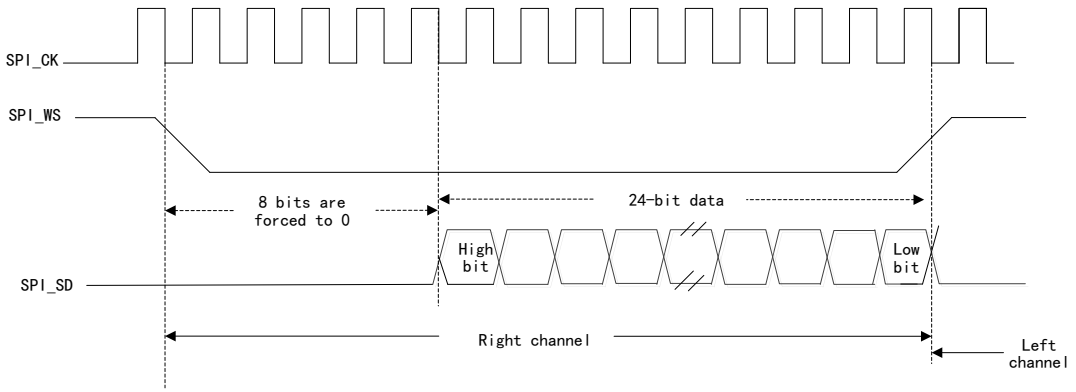


Figure 105 LSB Alignment Standard Waveform (24-bit data)



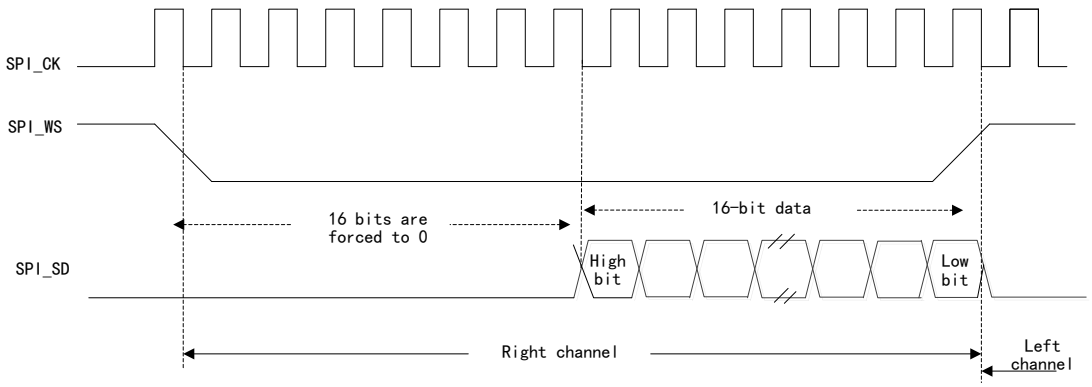
In the transmission process, if you want to transmit/receive 24-bit data, it is required to read/write the SPI_DATA register twice; for example:

- When you need to transmit 0x56EA98, write 0xXX56 to SPI_DATA register for the first time, and write 0xEA98 to SPI_DATA for the second time.
- When you need to receive 0x56EA98, read out 0x0056 from SPI_DATA register for the first time, ad read out 0xEA98 from SPI_DATA register for the second time.

In I2S configuration, when selecting the frame format of extending from 16-bit data to 32-bit data frame, it is required to access SPI_DATA register, and the high 16-bit data will be set to 0x0000 by hardware by force; for example:

- The data to be received or transmitted is 0x98A5, which becomes 0x000098A5 after it is extended to 32 bits, and it is necessary to write 0x98A5 to SPI_DATA register or read out from SPI_DATA register.

Figure 106 Under LSB Alignment Standard (extending from 16 bits to 32 bits)

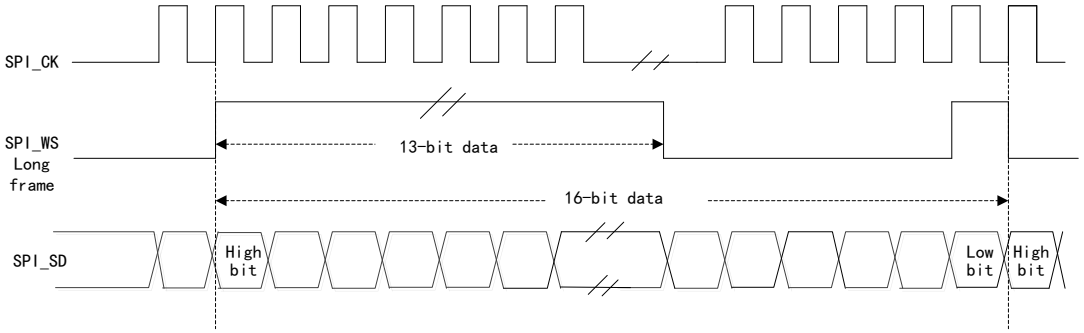


23.6.1.4 PCM standard

There is no sound channel selection in PCM standard. Short frame and long frame of PCM standard are selected by configuring PFSSEL bit in SPI_I2SCFG register.

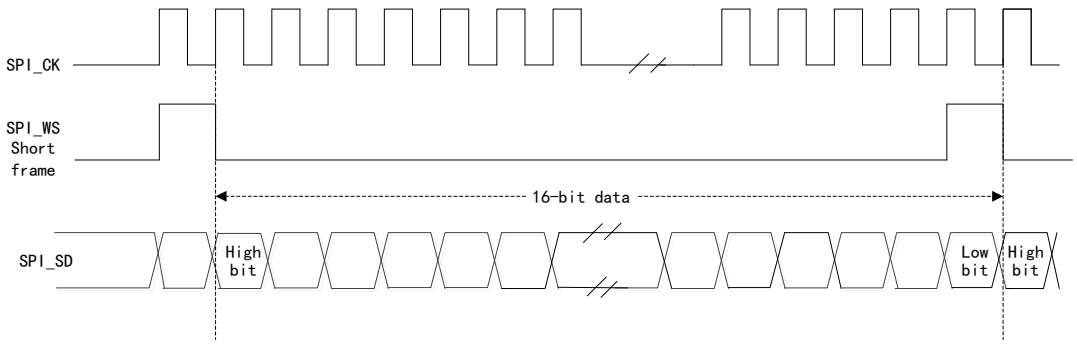
In the master mode, the valid time of synchronous WS signal of the long frame structure is 13 bits.

Figure 107 PCM Standard Waveform



In the master mode, the length of the synchronous WS signal of the short frame structure is 1 bit.

Figure 108 PCM Standard Waveform



23.6.2 I2S clock

The clock source of I2SxCLK is system clock (HSICLK, HSECLK or PLL of AHB clock)

The bit rate of I2S determines the data stream on I2S data line and the clock signal frequency of I2S.

- I2S bit rate = the number of bits per channel × the number of sound channels × audio sampling frequency
- There are two channels of 16-bit audio signal: I2S bit rate=16×2×Fs

The relationship between audio sampling frequency (Fs) and I2S bit rate (I2S) is defined by the following formula:

Table 84 Audio Sampling Frequency (Fs) Formula

MCOEN	CHLEN	Audio sampling frequency (Fs)
1	0	I2SxCLK/[（16*2）*（（2*I2SPSC）+ODDPSC）*8]
1	1	I2SxCLK/[（32*2）*（（2*I2SPSC）+ODDPSC）*4]
0	0	I2SxCLK/[（16*2）*（（2*I2SPSC）+ODDPSC）]
0	1	I2SxCLK/[（32*2）*（（2*I2SPSC）+ODDPSC）]

23.6.3 I2S mode

Table 85 I2S Run Mode

Operation Modes	SD	WS	CK	MCK
Master transmitting	Output	Output	Output	Output/Not use
Master receiving	Input	Output	Output	Output/Not use
Slave transmitting	Output	Input	Input	Output/Not use
Slave receiving	Input	Input	Input	Output/Not use

23.6.3.1 I2S master mode configuration process

- Configure I2SPSC bit and ODDPSC bit of SPI_I2SPSC register to define the baud rate of serial clock and the actual frequency division factor corresponding to the audio sampling frequency.
- Configure CPOL bit of SPI_I2SCFG register to define the clock polarity of SPI in idle state.
- Configure I2SMOD bit of SPI_I2SCFG register to activate I2S function and configure I2SMOD and PFSSEL bits of SPI_I2SCFG register to select I2S standard; configure DATALEN bit of SPI_I2SCFG register to select the data bits of the sound channel, and configure I2SMOD bit to select I2S master mode and transmitting terminal/receiving terminal.
- Configure SPI_CTRL2 register to select to enable the interrupt and DMA function or not (select required or not).
- Configure WS pin and CK pin to output mode; when MCOEN bit of SPI_I2SPSC is set to 1, the MCK pin should also be configured to output mode.
- Set the running mode of I2S by configuring the I2SMOD bit of SPI_I2SCFG.
- Set I2SEN bit of SPI_I2SCFG register to 1.

23.6.3.2 I2S master mode transmission process

When the data is written to the transmit buffer, the transmission will start, and the data will be transmitted from the transmit buffer to the shift register, the TXBEFLG flag position is set to 1, and the SCHDIR flag bit indicates the corresponding sound channel of the currently transmitted data. The value of SCHDIR flag bit will be updated when TXBEFLG flag bit is 1.

When transmitting the first bit of data, 16-bit data will be transmitted to the 16-bit shift register in parallel, and then transmitted from the pin MISO/SD in serial. The next data needs to be written to SPI_DATA register when TXBEFLG flag bit is 1. If TXBEIEN bit of SPI_CTRL2 is 1, an interrupt will be generated.

Before the completion of the current data transmission, write the next data to be transmitted to ensure continuous transmission of audio data.

When I2S is disabled, I2SEN can be cleared to 0 only when the flag bit TXBEFLG is 1 and BSYFLG is 0.

23.6.3.3 I2S master mode receiving process

RXBNEFLG flag is used to control the receiving sequence. RXBNEFLG flag indicates whether the receive buffer is empty; when the receive buffer is full, the RXBNEFLG flag bit will be set to 1. If RXBNEIEN bit of SPI_CTRL2 is configured, an interrupt will occur and after the user reads out the data from SPI_DATA register, the RXBNEFLG flag bit will be cleared to 0. Make sure to receive new data after reading operation; otherwise, overrun will occur and the

OVRFLG flag bit will be set to 1.

The value of SCHDIR should be updated immediately after receiving data, and it depends on the WS signal generated by I2S.

Regardless of the data type and the channel length, the audio data is always received in the form of 16 bits. According to the configured data and the length of the channel, the data needs to be transmitted to the receive buffer once or twice.

Disable the I2S function, and for different audio protocols, the data length and channel length operation steps are as follows:

16-bit data length, 32-bit channel length (DATALEN=00, CHLEN=1, I2SSSEL=10), in LSB alignment mode

- Wait until the penultimate RXBNEFLG is set to 1
- Wail for 17 I2S clock cycles (software delay)
- I2SEN flag bit is cleared to 0

16-bit data length, 32-bit channel length (DATALEN=00, CHLEN=1, I2SSSEL=10), in MSB alignment mode

- Wait until the last RXBNEFLG is set to 1
- Wail for 1 I2S clock cycle (software delay)
- I2SEN flag bit is cleared to 0

All the other situations

- Wait until the penultimate RXBNEFLG is set to 1
- Wail for 1 I2S clock cycle (software delay)
- I2SEN flag bit is cleared to 0

BSYFLG flag clock is low during data transmission

23.6.3.4 I2S slave mode configuration process

The configuration method of slave mode is basically the same as that of master mode. In slave mode, the clock signal and WS signal are provided by external I2S device instead of I2S.

- Configure I2SMOD bit of SPI_I2SCFG register to activate I2S function.
- Configure I2SSSEL bit of SPI_I2SCFG register to select the I2S standard; configure DATALEN[1:0] bit of SPI_I2SCFG register to select the bits of data; configure CHLEN bit of SPI_I2SCFG register to select the data bits per channel; configure I2SMOD bit of SPI_I2SCFG register to select I2S slave mode as transmitting terminal/receiving terminal.
- Configure SPI_CTRL2 register to select to enable the interrupt and DMA function or not (select required or not).

- Set I2SEN bit of SPI_I2SCFG register to 1.

23.6.3.5 I2S slave mode transmission process

Enable the slave device, write the data to the I2S data register, the external master device will start to communicate, and the external master device will transmit the clock signal, and when the data transmission starts, the transmitting process will begin.

When the first bit data is transmitted, the 16-bit data will be transmitted to the 16-bit shift register in parallel, and then transmitted from the pin MOSI/SD in series. When the data is transmitted from the data register to the shift register, the TXBEFLG flag bit will be set to 1; at this time if TXBEIEN bit of SPI_CTRL2 register is set, an interrupt will be generated. In order to ensure the continuity of data transmission, before the data transmission is completed, the next data should be written to SPI_DATA register; otherwise, "underrun" will occur, and the UDRFLG flag bit will be set to 1.

SCHDIR bit of SPI_STS register indicates the channel corresponding to the transmitted data. In the slave mode, the SCHDIR bit is determined by the WS signal of the external master device.

In MSB and LSB alignment mode of I2S, the first data written to the data register corresponds to the data of the left channel.

Disable I2S, and after the TXBEFLG flag bit is set to 1, BSYFLG flag bit can be cleared to 0.

23.6.3.6 I2S slave mode receiving process

RXBNEFLG flag is used to control the receiving sequence. The RXBNEFLG flag indicates whether the receive buffer is empty; after the receive buffer is full, the RXBNEFLG flag bit will be set to 1; if RXBNEIEN bit of SPI_CTRL2 register is set, an interrupt will occur, and after the data are read out from SPI_DATA register, RXBNEFLG flag bit will be cleared to 0; make sure to receive new data after read operation; otherwise, "overrun" will occur, and the OVRFLG flag bit will be set to 1.

The value of SCHDIR should be updated immediately after receiving data, and it depends on the WS signal generated by I2S.

Regardless of the data type and the channel length, the audio data is always received in the form of 16 bits. According to the configured data and the length of the channel, the data needs to be transmitted to the receive buffer once or twice.

Disable I2S, and when receiving the last RXBNEFLG set to 1, I2SEN flag bit will be cleared to 0.

23.6.4 I2S interrupt

23.6.4.1 Status flag bit

There are four status flag bits in I2S to monitor the status of I2S bus.

Transmit buffer empty flag bit TXBEFLG

When the TXBEFLG flag bit is 1, it indicates that the transmit buffer is empty, and the data to be transmitted can be written to the transmit buffer; after the data is written, the TXBEFLG flag bit will be cleared to 0. (When I2S is disabled, the TXBEFLG flag bit is 0).

Receive buffer non-empty flag bit RXBNEFLG

When the RXBNEFLG flag bit is 1, it indicates that the receive buffer has data to be received; after reading SPI_DATA register, RXBNEFLG flag bit will be cleared to 0.

Busy flag bit BSYFLG

When the BSYFLG flag bit is 1, it indicates that I2S is in communication state (set and cleared by hardware), but in the master receiving mode, the BSYFLG flag bit is always 0 during the receiving period.

When I2S is disabled and data transmission is over, the BSYFLG flag bit will be cleared to 0.

During continuous communication

- In the master transmitting mode, the BSYFLG flag bit is always high during the transmission period.
- In the slave mode, during transmission of each data item, the BSYFLG flag bit is set to 0 within one I2S clock cycle.

Channel flag bit SCHDIR

In the transmitting mode, SCHDIR flag bit is refreshed when TXBEFLG flag bit is high, indicating the channel of the data transmitted on SD pin at this time. If underrun error occurs in the transmission process of slave mode, the value of SCHDIR flag bit will be invalid, and the communication can be started after disabling I2S function and enabling it again.

In the receiving mode, SCHDIR flag bit is refreshed when SPI_DATA register receives the data, indicating the channel of the received data. if overrun error occurs, the SCHDIR flag bit is invalid, and the communication can be started after disabling I2S function and enabling it again.

As there is no channel selection in PCM standard, the SCHDIR flg bit is meaningless.

When OVRFLG and UDRFLG flag bits of SPI_STS register are 1 and ERRIEN bit of SPI_CTRL2 is 1, an interrupt will be generated. Clear the flag by reading the value of SPI_STS register.

23.6.4.2 Error flag bit

I2S contains two error flag bits

Underrun flag bit UDRFLG

In the transmitting mode, if new data to be transmitted is written to SPI_DATA register before the data is transmitted, UDRFLG flag bit will be set to 1; at this time if ERRIEN bit of SPI_CTRL2 register is set to 1, an interrupt will be generated.

This flag bit will take effect only after I2SMOD bit of SPI_I2SCFG is set to 1.

Clear the UDRFLG flag bit by reading SPI_STS register.

Overrun flag bit OVRFLG

In the receiving mode, if a new data is received before the data is read, OVRFLG flag bit will be set to 1. At this time if ERRIEN bit of SPI_CTRL2 register is set to 1, an interrupt will be generated, indicating the occurrence of the error.

Read SPI_DATA register to return the last correctly received data, and all the other newly received data will be lost.

OVRFLG flag can be cleared by first reading SPI_STS register and then reading SPI_DATA register.

Frame error flag FREFLG

This bit can be set to 1 by hardware only when I2S is configured in slave mode. If the WS line signal is changed during data frame transmission, the flag will be set to 1. At this time, if ERRIEN bit of SPI_CTRL2 register is set to 1, an interrupt will be generated.

Clear the UDRFLG flag bit by reading SPI_STS register.

Table 86 I2S Interrupt Request

Interrupt flag	Interrupt event	Enable control bit	Clearing method
TXBEFLG	Transmit buffer empty flag	TXBEIEN	Write SPI_DATA register
RXBNEFLG	Receive buffer non-empty flag	RXBNEIEN	Read SPI_DATA register

Interrupt flag	Interrupt event	Enable control bit	Clearing method
OVRFLG	Underrun flag bit	ERRIEN	Read SPI_STS register
UDRFLG	Overrun flag bit		Read SPI_STS register Read SPI_DATA register again
FREFLG	TI frame format error flag bit		Read SPI_STS register

23.6.4.3 DMA function

In I2S mode, the work mode of DMA is the same as that of SPI, except that it does not support CRC function.

23.7 Register address mapping

Table 87 SPI and I2S 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_I2S_CFG	SPI_I2S configuration register	0x1C
SPI_I2SPSC	SPI_I2S prescaler register	0x20

23.8 Register functional description

These peripheral registers can be operated by half word (16 bits) or word (32 bits).

23.8.1 SPI control register 1 (SPI_CTRL1) (not used in I2S mode)

Offset address: 0x00

Reset value: 0x0000

Field	Name	R/W	Description
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. Except that CRC function is used in TI mode, this bit is not used in I2S mode and SPI TI mode.
1	CPOL	R/W	Clock Polarity Configure The state maintained by SCK when SPI is in idle state. 0: SCK low 1: SCK high Note: This bit cannot be modified during communication. Except that CRC function is used in TI mode, this bit is not used in I2S mode and SPI TI mode.
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.
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=Fmaster/DIV Note: This bit cannot be modified during communication
6	SPIEN	R/W	SPI Device Enable 0: Disable 1: Enable Note: When SPI device is disabled, please operate according to the process of disabling SPI.
7	LSBSEL	R/W	LSB First Transfer Select 0: First transmit the most significant bit (MSB) 1: First transmit the least significant bit (LSB)
8	ISSEL	R/W	Internal Slave Device Select Determine the level on NSS pin This bit can be set effectively only when CTRL1_SSEN=1.
9	SSEN	R/W	Software Slave Device Enable 0: Disable 1: Enable When SSEN is set, the level of NSS pin is determined by SSEN.

Field	Name	R/W	Description
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.
11	CRCLSEL	R/W	CRC Length Select 0: Use 8-bit CRC 1: Use 16-bit CRC Note: This bit can be written only when SPIEN=0; otherwise, an error will occur.
12	CRCNXT	R/W	CRC Transfer Next Enable 0: Next value to be transmitted is from transmit buffer 1: Next value to be transmitted is from transmit CRC register Note: After the last data is written to SPI_DATA register, set CRCNXT bit immediately.
13	CRCEN	R/W	CRC Calculate Enable 0: CRC check is disabled 1: CRC check is enabled CRC check function only applies to full-duplex mode; only when SPIEN=0, can this bit be changed.
14	BMOEN	R/W	Bidirectional Mode Output Enable 0: Disable (receive-only ode) 1: Enable (transmit-only mode) When BMEN=1, namely in single-line bidirectional mode, this bit decides the transmission direction of transmission line.
15	BMEN	R/W	Bidirectional Mode Enable 0: Double-line bidirectional mode 1: Single-line bidirectional mode Single-line bidirectional transmission means: transmission between MOSI pin of data master and MISO pin of slave.

23.8.2 SPI control register 2 (SPI_CTRL2)

Offset address: 0x04

Reset value: 0x0700

Field	Name	R/W	Description
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
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

Field	Name	R/W	Description
			1: Enable
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. Note: Not available in I2S and SPI TI modes.
3	NSSPEN	R/W	NSS Pulse Management Enable 0: Disable 1: Enable Notes: During continuous transmission, it is allowed to generate NSS pulse between transmission of two data. During single data transmission, NSS pin will be forced to be pulled up at the end of transmission. This bit is invalid when CPHA=1 or FRFCFG=1. This bit can be written only when SPIEN=0. Not available in I2S and SPI TI modes.
4	FRFCFG	R/W	Frame Format Configure 0: SPI Motorola mode 1: SPI TI mode Notes: This bit can be written only when SPIEN=0. Not available in I2S mode.
5	ERRIEN	R/W	Error interrupt Enable 0: Disable 1: Enable When an error occurs, ERRIEN bit controls whether to generate the interrupt.
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
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
11:8	DSCFG	R/W	Data Size Configure Configure the bit width of SPI transmission date: 0000: Reserved 0001: Reserved 0010: Reserved 0011: 4 bits 0100: 5 bits 0101: 6 bits 0110: 7 bits

Field	Name	R/W	Description
			0111: 8 bits 1000: 9 bits 1001: 10 bits 1010: 11 bits 1011: 12 bits 1100: 13 bits 1101: 14 bits 1110: 15 bits 1111: 16 bits Notes: When reserved bit is written by software, the value will be forced to be 0111 (8 bits) Not used in I2S mode
12	FRTCFG	R/W	FIFO Reception Threshold Configure Configure FIFO threshold, and when the value exceeds this threshold, RXBNEFLG will occur 0: 16 bits 1: 8 bits Note: Not available in I2S mode
13	LDRX	R/W	Last DMA Receive These bits are used in data packing mode to define the total number received by DMA to be odd or even. 0: Even 1: Odd Notes: These bits are meaningful only when RXDEN bit of SPI_CTRL2 register is set and the packing mode is enabled. This bit can be written only when SPIEN=0. Disable SPI according to the steps in 错误!未找到引用源。. Not used in I2S mode.
14	LDTX	R/W	Last DMA Transmit These bits are used in data packing mode to define the total number transmitted by DMA to be odd or even. 0: Even 1: Odd Notes: These bits are meaningful only when RXDEN bit of SPI_CTRL2 register is set and the packing mode is enabled. This bit can be written only when SPIEN=0. Close SPI according to the steps in "23.4.10". Not used in I2S mode.
15	Reserved		

23.8.3 SPI status register (SPI_STS)

Offset address: 0x08

Reset value: 0x0002

Field	Name	R/W	Description
0	RXBNEFLG	R	Receive Buffer Not Empty Flag This bit indicates that the receive buffer is empty or not 0: Empty 1: Not empty
1	TXBEFLG	R	Transmit Buffer Empty Flag This bit indicates that the transmit buffer is empty or not 0: Not empty 1: Empty
2	SCHDIR	R	Sound Channel Direction Flag 0: Indicate that the left channel is transmitting or receiving the required data 1: Indicate that the right channel is transmitting or receiving the required data Note: Not used in SPI mode, without left and right channels in PCM mode.
3	UDRFLG	R	Underrun Occur Flag This bit indicates whether the underrun occurs or not 0: Not occur 1: Occurred This flag bit is set by hardware and reset by software. Not used in SPI mode
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 and reset by software. Not used in I2S mode
5	MEFLG	R	Mode Error Occur Flag This bit indicates whether mode error occurs or not 0: Not occur 1: Occurred This bit can be set by hardware and reset by software. Not used in I2S mode
6	OVRFLG	R	Overrun Occur Flag This bit indicates whether overrun occurs or not 0: Not occur 1: Occurred This bit can be set by hardware and reset by software.
7	BSYFLG	R	Busy Flag This bit indicates the work state of SPI 0: SPI is idle 1: SPI is communicating This bit can be set or reset by hardware

Field	Name	R/W	Description
8	FREFLG	R	Frame Format Error Flag 0: Not occur 1: Occurred Note: This bit is set to 1 by hardware and cleared to 0 by reading SPI_STS register.
10:9	FRLSEL	R	FIFO Receive Leve Select p00: FIFO is empty 01：FIFO/4 10：FIFO/2 11: FIFO is full Note: This bit is set to 1 and cleared to 0 by hardware. It is not used in 2S mode and SPI single receiving mode with CRC check.
12:11	FTLSEL	R	FIFO Transmit Leve Select p00: FIFO is empty 01：FIFO/4 10：FIFO/2 11: FIFO is full (it can be considered as full when the threshold value of FIFO is greater than 1/2) Note: This bit is set to 1 and cleared to 0 by hardware. It is not used in I2S mode
15:13	Reserved		

23.8.4 SPI data register (SPI_DATA)

Offset address: 0x0C

Reset value: 0x0000

Field	Name	R/W	Description
15:0	DATA	R/W	Transmit Receive Data register Store the data to be transmitted or received. 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, 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.

23.8.5 SPI CRC polynomial register (SPI_CRCPOLY) (not used in I2S mode)

Offset address: 0x10

Reset value: 0x0007

Field	Name	R/W	Description
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.

23.8.6 SPI receive CRC register (SPI_RXCRC) (not used in I2S mode)

Offset address: 0x14

Reset value: 0x0000

Field	Name	R/W	Description
15:0	RXCRC	R	Receive Data CRC Value The CRC data of receive bytes 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.

23.8.7 SPI transmit CRC register (SPI_TXCRC) (not used in I2S mode)

Offset address: 0x18

Reset value: 0x0000

Field	Name	R/W	Description
15:0	TXCRC	R	Transmit Data CRC Value The CRC data of transmitted bytes calculated by hardware are stored in TXCRC; 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. Note: When BSYFLG bit is set to 1, the value of reading RXCRC register may be wrong.

23.8.8 SPI_I2S configuration register (SPI_I2SCFG) (not applicable in SPI mode)

Offset address: 0x1C

Reset value: 0x0000

Field	Name	R/W	Description
0	CHLEN	R/W	Channel Length Configure The channel length refers to the data bits per audio channel 0: 16-bit width 1: 32-bit width Write operation is meaningful only when DATALEN=00 for this bit; otherwise, the channel length will be fixed as 32 bits by hardware. Note: This bit can be set only when I2S is disabled. Not used in SPI mode.
2:1	DATALEN	R/W	Length of Data To Be Transferred Configure 00: 16-bit data length 01: 24-bit data length 10: 32-bit data length 11: Disable Note: This bit can be set only when I2S is disabled. Not used in SPI mode.

Field	Name	R/W	Description
3	CPOL	R/W	Steady State Clock Polarity Select Level state when I2S clock is in static state 0: Low level 1: High level Note: This bit can be set only when I2S is disabled. Not used in SPI mode.
5:4	I2SSSEL	R/W	I2S Standard Select 00: I2S Philips standard 01: High-byte alignment standard (left alignment) 10: Low-byte alignment standard (right alignment) 11: PCM standard Note: This bit can be set only when I2S is disabled. Not used in SPI mode.
6	Reserved		
7	PFSSEL	R/W	PCM Frame Synchronization Mode Select 0: Synchronization of short frames 1: Synchronization of long frames Note: This bit is meaningful only when I2SSSEL=11. Not used in SPI mode.
9:8	I2SMOD	R/W	I2S Master/Slave Transmit/Receive Mode Configure 00: Slave device transmits 01: Slave device receives 10: Master device transmits 11: Master device receives Note: This bit can be set only when I2S is disabled. Not used in SPI mode.
10	I2SEN	R/W	I2S Enable 0: Disable I2S 1: Enable I2S Note: It is not used in SPI mode.
11	MODESEL	R/W	SPI/I2C Mode Select 0: Select SPI mode 1: Select I2S mode Note: This bit can be set only when SPI or I2S is disabled.
15:12	Reserved		

23.8.9 SPI_I2S prescaler register (SPI_I2SPSC) (not used in SPI mode)

Offset address: 0x20

Reset value: 0x0002

Field	Name	R/W	Description
7:0	I2SPSC	R/W	I2S Linear Prescaler Factor Configure Disabl setting I2SPSC [7:0]=0 or I2SPSC [7:0]=1 Note: This bit can be set only when I2S is disabled. This bit is used only when I2S is in master device mode. Not used in SPI mode.

Field	Name	R/W	Description
8	ODDPSC	R/W	Configure the prescaler factor to be odd 0: Actual division factor=I2SPSC*2 1: Actual division factor=(I2SPSC*2)+1 Note: This bit can be set only when I2S is disabled. This bit is used only when I2S is in master device mode. Not used in SPI mode.
9	MCOEN	R/W	Master Device Clock Output Enable 0: Disable 1: Enable Note: This bit can be set only when I2S is disabled. This bit is used only when I2S is in master device mode. Not used in SPI mode.
15:10	Reserved		

24 Analog/Digital converter (ADC)

24.1 Introduction

ADC with 12-bit precision and 19 channels, including 16 external channels and 3 internal channels, and there are single, continuous or intermittent A/D conversion modes for each channel. ADC conversion results can be left-aligned or right-aligned and stored in 16-bit data register.

24.2 Main characteristics

（1） ADC power supply requirements: From 2.4V to 3.6V

（2） ADC input range: $V_{SSA} \leq V_{IN} \leq V_{DDA}$

（3） Conversion mode

- Single conversion mode
- Continuous conversion mode
- Intermittent mode

（4） Analog input channel category

- External GPIO input channel
- One internal temperature sensor (V_{SENSE}) input channel
- One internal reference voltage (V_{REFINT}) input channel
- One (1/2*VDD) pin voltage input channel

（5） High performance

- 12-bit, 10-bit, 8-bit or 6-bit configurable resolution.
- Self-calibration
- Programmable sampling time
- Data alignment
- DMA supported

（6） Low power

- Low-power operation reduces PCLK frequency and maintains optimum ADC performance
- Automatic delay mode: Run in PCLK low speed, to prevent ADC over-limit
- Automatic shutdown mode: ADC can power off automatically at other times except during conversion period

（7） Interrupt

- End of conversion interrupt
- End of sequence conversion interrupt
- End of sampling phase interrupt
- ADC ready interrupt

- Overrun interrupt
- Analog watchdog state reset interrupt

（8） Trigger mode

- External pin signal trigger
- Internal signal trigger generated by on-chip timer

（9） Data register

- 19 data registers, used to store the results of sequence segmented sampling

24.3 Functional description

24.3.1 ADC pin and internal signal

Table 88 ADC Internal Signal

Name	Description	Signal type
TMRx_TRG	Internal information from timer	Input
V_{SENSE}	Output voltage of internal temperature sensor	Input
V_{REFINT}	Output of internal reference voltage	Input
1/2*VDD	Input voltage of 1/2*VDD pin	Input

Table 89 ADC Pins

Name	Description	Signal type
V_{DDA}	Analog power supply, positive ADC reference voltage, $V_{DDA} \geq V_{DD}$	Input, analog power supply
V_{SSA}	Analog ground, $V_{SSA}=V_{SS}$	Input, analog power ground
ADC_IN[15:0]	16-way analog input	Analog input signal

24.3.2 Calibration

The function of calibration is to eliminate the offset error of A/D conversion of each chip, so calibration should be conducted before A/D conversion, and ADC module cannot be used during calibration.

Calibration configuration process:

- Configure ADCEN bit of register ADC_CTRL to 0, and disable ADC
- Configure CAL bit of register ADC_CTRL to 1, and enable calibration
- After calibration is completed, CAL bit is automatically cleared to 0 by hardware
- The calibration factor is read in CDATA[6:0] bit of register ADC_DATA

24.3.3 ADC conversion mode

24.3.3.1 Single conversion mode

In this mode, for single channel, only one conversion is performed for this channel, and for multiple channels, only one conversion is performed for this

group of channels .

When CMODESEL bit of configuration register ADC_CFG1 is 0, ADC is set to single conversion mode; ADC conversion can be enabled by setting STARTCEN bit of configuration register ADC_CTRL to 1 by software or by hardware-triggered event.

After the conversion of each channel, the converted data will be stored in the 16-bit ADC_DATA register, EOCFLG bit will be set to 1, and if EOCIEN bit is set to 1, an interrupt will be generated. After the channel sequence conversion, EOSEQFLG bit will be set to 1, and if EOSEQIEN bit is set to 1, an interrupt will be generated.

24.3.3.2 Continuous conversion mode

In this mode, for single channel, continuous conversion is conducted for this channel; for multiple channels, continuous conversion is only conducted for this group of channel.

When CMODESEL bit of configuration register ADC_CFG1 is set to 1, ADC is set to continuous conversion mode; ADC conversion can be enabled by setting STARTCEN bit of configuration register ADC_CTRL to 1 by software or by hardware-triggered event.

After the conversion of each channel, the converted data will be stored in the 16-bit ADC_DATA register, EOCFLG bit will be set to 1, and if EOCIEN bit is set to 1, an interrupt will be generated. After the channel sequence conversion, EOSEQFLG bit will be set to 1, and if EOSEQIEN bit is set to 1, an interrupt will be generated.

24.3.3.3 Intermittent mode

When DISCEN bit of configuration register ADC_CFG1 is set to 1, ADC is set to intermittent mode; ADC conversion can be enabled by software or by hardware-triggered event. In this mode, only one channel of one sequence is converted at a time. If DISCEN bit is cleared to 0, all channels of one sequence will be converted at a time.

For example:

- DISCEN bit is set to 1, and the channel sequence is 0, 1, 5
  - 1st trigger, Channel 0 is converted and generates an EOCFLG event
  - 2nd trigger, Channel 1 is converted and generates an EOCFLG event
  - 3rd trigger, Channel 5 is converted and generates an EOCFLG event
- DISCEN bit is set to 0, and the channel sequence is 0, 1, 5
  - 1st trigger, channels 0, 1 and 5 are converted in sequence. After the conversion of each channel, an EOCFLG event will be

generated. After the conversion of the whole sequence, an EOSEQFLG event will be generated

24.3.4 ADC channel classification

24.3.4.1 Analog input channel introduced by GPIO pin

In total 16 channels are connected to ADC_IN0…ADC_IN15.

24.3.4.2 Internal analog input channel

Internal reference voltage VREFINT

（1） The internal reference voltage is used to provide a stable voltage output for ADC

（2） Internal reference voltage V_{REFINT} selects ADC1_IN17 input channel

Voltage of 1/2*VDD pin

1/2*VDD pin input voltage selects ADC1_IN18 input channel.

Temperature sensor

（1） The temperature sensor is used to measure the internal temperature of the chip
（2） The temperature sensor selects ADC1_IN16 input channel
（3） Enable by TSEN bit of configuration register ADC_CCFG
（4） Select sampling time
（5） Supported temperature range: -40~105℃

The temperature sensor is used to measure the contact temperature of the product. The sensor transmits the voltage value by ADC_ IN16 to ADC, and then the voltage value will be converted to a value by ADC.

Configuration

- Enable ADC, select input channel
- Select sampling time (recommend 17.1us)
- Temperature sensor can be enabled by configuring the TSEN bit in the ADC_CCFG register
- Start conversion by configuring the STARTCEN bit in the SDC_CTRL register
- The conversion result can be read by accessed the ADC_DATA register
- Convert the result to temperature T though the formula as:

$$T(℃) = 25 + \frac{V_{25} - V_{sensor}}{Slope}$$

Note: (1) V25: The value of Vsensor at 25℃, see the Datasheet for more details.

(2) Slope: Average slope value of Vsensor and temperature (Unit: mV/℃), see the Datasheet for more details.

(3) When configuring the start up time, the ADCEN bit and TSEN bit need to be configured at the same time.

24.3.5 External trigger and trigger polarity

（1） The external trigger of the first segment of the sequence segmented sampling function and the external trigger event of other functions can be configured by EXTTRGSEL1 bit of configuration register ADC_CFG1.

Table 90 External Trigger of the First Segment

Trigger source	EXTTRGSEL	Trigger type
TMR1_TRGO	0000	Internal signal generated by on-chip timer
TMR1_CC4	0001	
TMR2_TRGO	0010	
TMR3_TRGO	0011	
TMR4_TRGO	0100	
TMR1_TRGO2	0101	
TMR1_TRGO3	0110	
TMR2_TRGO2	0111	
TMR3_TRGO2	1000	
TMR4_TRGO2	1001	
Reserved	1010/1100/1101/1110/1111	External pin

When the bit EXTPOLSEL1≠"0b00" for the register ADC_CFG1, the external event can trigger conversion on its selected polarity.

Table 91 Configuration Trigger Polarity

EXTPOLSEL	Source
00	Disable trigger detection
01	Detection on rising edge
10	Detection on falling edge
11	Detection on both rising edge and falling edge

（2） The external trigger of the second segment of the sequence segmented sampling function and the external trigger event of other functions can be configured by EXTTRGSEL2 bit of configuration register ADC_CFG3.

Table 92 External Trigger of the Second Segment

Trigger source	EXTTRGSEL2	Trigger type
TMR1_TRGO	0000	Internal signal generated by on-chip timer
TMR1_CC4	0001	
TMR2_TRGO	0010	
TMR3_TRGO	0011	
TMR4_TRGO	0100	
TMR1_TRGO2	0101	

Trigger source	EXTTRGSEL2	Trigger type
TMR1_TRGO3	0110	
TMR2_TRGO2	0111	
TMR3_TRGO2	1000	
TMR4_TRGO2	1001	
Reserved	1010/1100/1101/1110/1111	External pin

When the bit EXTPOLSEL2≠"0b00" for the register ADC_CFG3, the external event can trigger conversion on its selected polarity.

Table 93 Configuration Trigger Polarity

EXTPOLSEL2	Source
00	Disable trigger detection
01	Detection on rising edge
10	Detection on falling edge
11	Detection on both rising edge and falling edge

（3） The external trigger of the third segment of the sequence segmented sampling function and the external trigger event of other functions can be configured by EXTTRGSEL3 bit of configuration register ADC_CFG3.

Table 94 External Trigger of the Third Segment

Trigger source	EXTTRGSEL3	Trigger type
TMR1_TRGO	0000	Internal signal generated by on-chip timer
TMR1_CC4	0001	
TMR2_TRGO	0010	
TMR3_TRGO	0011	
TMR4_TRGO	0100	
TMR1_TRGO2	0101	
TMR1_TRGO3	0110	
TMR2_TRGO2	0111	
TMR3_TRGO2	1000	
TMR4_TRGO2	1001	
Reserved	1010/1100/1101/1110/1111	External pin

When the bit EXTPOLSEL3≠"0b00" for the register ADC_CFG3, the external event can trigger conversion on its selected polarity.

Table 95 Configuration Trigger Polarity

EXTPOLSEL3	Source
00	Disable trigger detection
01	Detection on rising edge

EXTPOLSEL3	Source
10	Detection on falling edge
11	Detection on both rising edge and falling edge

24.3.6 Data register

The data can be left-aligned or right-aligned, which is determined by DALIGCFG bit of configuration register ADC_CFG1; when DALIGCFG is set to 0, it means right-aligned, and if DALIGCFG is set to 1, it means left-aligned. ADC conversion results can be left-aligned or right-aligned and stored in 16-bit data register.

The data of the sequential segmented sampling mode is stored in ADC_DATA(0-18) register and can be left or right-aligned.

For example, to configure three-segment sequential sampling, if the total number of channels in the first segment is 1 (sampling channel 0), the total number of channels in the second segment is 2 (sampling channels 1, 0), and the total number of channels in the third segment is 3 (sampling channels 2, 3, 1), (configure ADC_SEQ_NUM = 0x20000820; ADC_CHANNEL1.CH0=0; ADC_CHANNEL1.CH1=1, ADC_CHANNEL1.CH2=0; ADC_CHANNEL1.CH3=2, ADC_CHANNEL1.CH4=3, ADC_CHANNEL1.CH5=1), the data result of the first segment will appear in ADC_DATA0, the data result of the second segment will appear in ADC_DATA1-2, and the data result of the third segment will appear in ADC_DATA3-5. Although the ADC segmented sampling mode supports out of sequence (namely, sampling of the second and third segments can precede the first segment), the corresponding relationship of the corresponding result register of each segment is fixed (for example, after configuration is completed, enable ADC, and if ADC first receives the hardware trigger of sampling of the third segment, the result of analog channel 2 will appear in ADC_DATA3 after the sampling of this segment is over, the result of analog channel 3 will appear in ADC_DATA3, and the result of analog channel 1 will appear in ADC_DATA5).

24.3.7 Programmable conversion resolution

Reducing the resolution can improve the conversion time and 12, 10, 8 or 6-bit modes can be selected by DATARESCFG bit of configuration register ADC_CFG1.

Table 96 Conversion Time of tSAR Related to Conversion Resolution

DATARESCFG bit	tSAR	tSAR (ns)@f_{ADC}=14MHz	$tSMPL_{(min)}$	tADC	tADC(μs)@f_{ADC}=14MHz
6	7.5	535ns	1.5	9	643ns
8	9.5	678ns	1.5	11	785ns
10	11.5	821ns	1.5	13	928ns

DATARESCFG bit	tSAR	tSAR (ns)@f_{ADC}=14MHz	$tSMPL_{(min)}$	tADC	tADC(μs)@f_{ADC}=14MHz
12	12.5	893ns	1.5	14	1000ns

24.3.8 Interrupt

Table 97 ADC Interrupt

Interrupt event	Event flag	Enable control
End of conversion	EOCFLG	EOCIEN
End of sequence conversion	EOSEQFLG	EOSEQIEN
End of sampling phase	EOSMPFLG	EOSMPIEN
ADC ready	ADCRDYFLG	ADCRDYIEN
Overrun	OVREFLG	OVRIEN
Analog watchdog state reset	AWDFLG	AWDIEN
End of the third segment of sequential segmented sampling	SEQ_NUM3_FIN	SEQ_NUM3_FINIEN
End of the second segment of sequential segmented sampling	SEQ_NUM2_FIN	SEQ_NUM2_FINIEN
End of the first segment of sequential segmented sampling	SEQ_NUM1_FIN	SEQ_NUM1_FINIEN

24.3.9 ADC overrun

ADC overrun means when the converted data is not read by DMA or CPU on time, another converted data will take effect.

When EOCFLG bit is 1 but another new conversion has been completed, an overrun event will occur, and OVREFLG bit of register ADC_STS will be set to 1; if OVRIEN bit is set to 1, an overrun interrupt will be generated.

It is determined by OVRMAG bit of configuration register ADC_CFG1 that the data in the ADC data register are held or overwritten when an overrun event occurs:

- OVRMAG is 0: When an overrun event is detected, old data will be held in ADC_DATA register
- OVRMAG is set to 1: When an overrun event is detected, ADC_DATA register will overwrite the data by the last converted data

24.3.10 Data conversion management

24.3.10.1 Data conversion management without participation of DMA

The software controls data conversion. Every time the conversion is completed, EOCFLG bit will be set to 1, and the conversion results can be read from ADC_DATA register. OVRMAG bit in ADC_CFG1 register should be 0.

24.3.10.2 Data conversion management without participation of DMA and overrun

When one or more channels are converted and each conversion result does not need to be read, OVRMAG bit will be set to 1, the overrun event cannot prevent ADC conversion and the register ADC_Data only saves the last converted data.

24.3.10.3 Data conversion management by DMA

Transmission by DMA can be used to transmit the conversion results from the data register to the memory in time to prevent loss of the conversion results in the ADC_DATA register.

DMA can be enabled by setting DMAEN bit of the register ADC_CFG1 to 1. After each conversion, a DMA request will be generated to transmit the converted data of data register to the memory.

When DMA fails to respond to DMA request in time, an overrun event will be generated, and OVREFLG bit will be set to 1. After that, ADC will not generate a DMA request and DMA will not transmit new conversion results. DMA will start to work again when OVREFLG bit is cleared to 0.

Note: The sequential segmented sampling mode does not support DMA function.

DMA mode is selected by DMACFG bit of configuration register ADC_CFG1:

- When DMACFG is 0, DMA is in single mode
  - DMA programming is used to transmit the fixed-length data
  - In this mode, ADC will generate a DMA request every time it converts data effectively. When ADC conversion is restarted, ADC will stop generating DMA request
  - When the number of ADC conversions reaches the length of DMA, software is required to configure the STOP bit to stop ADC
- When DMACFG is set to 1, DMA is in circular mode
  - DMA programming is in circular mode or double-buffer mode
  - In this mode, when ADC conversion is started again and the converted data is valid, a DMA request will be generated

24.3.11 Low-power characteristics

24.3.11.1 Automatic delay conversion mode

This mode can greatly reduce the power consumption of application, and is suitable for applications with relatively few conversions or long conversion request time interval. Automatic shutdown mode can be used in combination with automatic delay conversion mode in low-frequency application.

Automatic shutdown mode can be enabled by setting AOEN bit of configuration register ADC_CFG1 to 1. When AOEN bit is set to 1 and there is no ADC conversion, it will be powered off automatically, and when the conversion is started, ADC will be woken up automatically.

Note: The sequential segmented sampling mode does not support.

24.3.11.2 Automatic shutdown mode

This mode can greatly reduce the power consumption of application, and is suitable for applications with relatively few conversions or long conversion request time interval. Automatic shutdown mode can be used in combination with automatic delay conversion mode in low-frequency application.

Automatic shutdown mode can be enabled by setting AOEN bit of configuration register ADC_CFG1 to 1. When AOEN bit is set to 1 and there is no ADC conversion, it will be powered off automatically, and when the conversion is started, ADC will be woken up automatically.

Note: The sequential segmented sampling mode does not support.

24.4 Register address mapping

Table 98 ADC Register Address Mapping

Register name	Description	Offset address
ADC_STS	ADC status register	0x00
ADC_IEN	ADC interrupt enable register	0x04
ADC_CTRL	ADC control register	0x08
ADC_CFG1	ADC configuration register 1	0x0C
ADC_CFG2	ADC configuration register 2	0x10
ADC_SMPTIM	ADC sampling time register	0x14
ADC_CFG3	ADC configuration register 3	0x18
ADC_AWDT	ADC watchdog threshold register	0x20
ADC_CHSEL	ADC channel selection register	0x28
ADC_DATA	ADC data register	0x40
ADC_DATAx（x=0~18）	ADC sequential segmented sampling data register	0x44+4x
ADC_SEQ_NUM	ADC sequential segmented sampling control register	0x100
ADC_CHANNEL1	ADC1st-6th-time transmission channel register	0x104
ADC_CHANNEL2	ADC7th-12th-time transmission channel register	0x108
ADC_CHANNEL3	ADC13th-18th-time transmission channel register	0x10C
ADC_CHANNEL4	ADC19th-time transmission channel register	0x110
ADC_CCFG	ADC general-purpose configuration register	0x308

Note: The above number of times is not the actual number of transmission times. The actual number of times is obtained based on ADC_SEQ_NUM register.

24.5 Register functional description

24.5.1 ADC status register (ADC_STS)

Offset address: 0x00

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	ADCRDYFLG	RC_W1	ADC Ready Flag 0: ADC not ready 1: ADC has been ready to start conversion
1	EOSMPFLG	RC_W1	End of Sampling Flag This bit is set to 1 by hardware and cleared to 0 by software 0: Not in the phase of end of sampling 1: Reach the condition for end of sampling phase
2	EOCFLG	RC_W1	End Of Conversion Flag This bit is set to 1 by hardware and cleared to 0 by software 0: Conversion does not end 1: Conversion ends
3	EOSEQFLG	RC_W1	End of Sequence Flag This bit is set to 1 by hardware and cleared to 0 by software 0: Sequence conversion not completed 1: Sequence conversion completed
4	OVREFLG	RC_W1	ADC Overrun Event Flag This bit is set to 1 by hardware and cleared to 0 by software 0: No overrun event 1: Overrun event occurred
6:5	Reserved		
7	AWDFLG	RC_W1	Analog Watchdog Flag This bit is set to 1 by hardware and cleared to 0 by software, indicating whether an analog watchdog event occurs. 0: Not occur 1: Occurred
8	SEQ_NUM3_FIN	RC_W1	NUM.3 of Sequential Section Sampling Finish 0: Not occur 1: Occurred
9	SEQ_NUM1_FIN	RC_W1	NUM.1 of Sequential Section Sampling Finish 0: Not occur 1: Occurred
10	SEQ_NUM2_FIN	RC_W1	NUM.2 of Sequential Section Sampling Finish 0: Not occur 1: Occurred
31:11	Reserved		

24.5.2 ADC interrupt enable register (ADC_IEN)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	ADCRDYIEN	R/W	ADC Ready Interrupt Enable 0: Disable 1: Enable
1	EOSMPIEN	R/W	End of Sampling Flag Interrupt Enable 0: Disable 1: Enable
2	EOCIEN	R/W	End of Conversion Interrupt Enable 0: Disable 1: Enable
3	EOSEQIEN	R/W	End of Conversion Sequence Interrupt Enable 0: Disable 1: Enable
4	OVRIEN	R/W	Overrun Interrupt Enable 0: Disable 1: Enable
6:5	Reserved		
7	AWDIEN	R/W	Analog Watchdog Interrupt Enable 0: Disable 1: Enable
8	SEQ_NUM3_FINIEN	R/W	NUM.3 of Sequential Section Sampling Finish Interrupt Enable 0: Disable 1: Enable
9	SEQ_NUM1_FINIEN	R/W	NUM.1 of Sequential Section Sampling Finish Interrupt Enable 0: Disable 1: Enable
10	SEQ_NUM2_FINIEN	R/W	NUM.2 of Sequential Section Sampling Finish Interrupt Enable 0: Disable 1: Enable
31:11	Reserved		

Note: These bits can be rewritten only when STARTCEN=0.

24.5.3 ADC control register 1 (ADC_CTRL)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	ADCEN	R/S	ADC Enable This bit is set to 1 by software and cleared to 0 by hardware. 0: Disable ADC 1: Enable ADC Note: ADCEN bit can be set by software only when all bits of ADC_CTRL register are 0.

Field	Name	R/W	Description
1	ADCD	R/S	ADC Disable This bit is set to 1 by software and cleared to 0 by hardware. 0: Invalid 1: Disable ADC, and enter power-down mode Note: ADCD bit can be set by software only when ADCEN=1 and STARTCEN=0.
2	STARTCEN	R/S	ADC Start Conversion Enable This bit is set to 1 by software and cleared to 0 by hardware. 0: Disable ADC conversion 1: Enable ADC conversion Note: STARTCEN bit can be set by software only when ADCEN=1 and ADCD=0.
3	Reserved		
4	STOPCEN	R/S	ADC Stop Conversion Enable This bit is set to 1 by software and cleared to 0 by hardware. 0: Invalid 1: Stop ADC conversion Note: This bit can be set by software only when STARTCEN=1 and ADCD=0.
30:5	Reserved		
31	CAL	R/S	ADC Calibrate This bit is set to 1 by software and cleared to 0 by hardware. 0: Calibration is completed 1: Start calibration Note: CAL bit can be set by software only when ADC is disabled.

24.5.4 ADC configuration register 1 (ADC_CFG1)

Offset address: 0x0C

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	DMAEN	R/W	DMA Enable 0: Enable DMA 1: Enable DMA
1	DMACFG	R/W	DMA Mode Configure This bit is valid only when DMAEN=1. 0: DMA single mode 1: DMA circular mode
2	SCANSEQDIR	R/W	Scan Sequence Direction Configure 0: Scan forward (from CHSEL0 to CHSEL16) 1: Scan backward (from CHSEL16 to CHSEL0)
4:3	DATARESCFG	R/W	Data Resolution Configure 00: 12 bits 01: 10 bits 10: 8 bits 11: 6 bits

Field	Name	R/W	Description
5	DALIGCFG	R/W	Data Alignment Configure 0: Right-aligned 1: Left-aligned
9:6	EXTTRGSEL1	R/W	External Trigger Event Select These bits are used to select the external event for triggering ADC 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 1000: Event 8 1001: Event 9 1010: Event 10 1011: Event 11 1100: Event 12 1101: Event 13: 1110: Event 14 1111: Event 15
11:10	EXTPOLSEL1	R/W	External Trigger Enable and Polarity Select 00: Hardware trigger detection is disabled (conversion can be started by software) (this bit cannot be 00 in segmented sampling function) 01: Hardware trigger detection on rising edge 10: Hardware trigger detection on falling edge 11:: Hardware trigger detection on both rising and falling edges
12	OVRMAG	R/W	Overrun Management Mode 0: When an overrun event is detected, ADC_DATA register saves previous data 1: When an overrun event is detected, ADC_DATA register saves the last converted data
13	CMODESEL	R/W	Select Single/Continuous Conversion Mode 0: Single conversion mode 1: Continuous conversion mode
14	WAITCEN	R/W	Wait Conversion Mode Enable 0: Disable 1: Enable
15	AOEN	R/W	Auto-Off Mode Enable 0: Disable 1: Enable
16	DISCEN	R/W	Discontinuous Mode Enable 0: Disable 1: Enable

Field	Name	R/W	Description
21:17	Reserved		
22	AWDCHEN	R/W	Enable The Watchdog On A Single Channel or on All Channels 0: Enable analog watchdog on all channels 1: Enable analog watchdog on a single channel
23	AWDEN	R/W	Analog Watchdog Enable 0: Disable 1: Enable
25:24	Reserved		
30:26	AWDCHSEL	R/W	Analog Watchdog Channel Select These bits are used to configure the input channel for the analog watchdog to monitor ADC 00000: Channel 0 00001: Channel 1 ..... 10010: Channel 18 Other values: Reserved, not used Note: The channels selected by AWDCHSEL bit must be written to CHSELR register. Sequential segmented sampling mode is used to detect in which data register the data is stored 00000: Data in ADC_DATA0 00001: Data in ADC_DATA1 ..... 10010: Data in ADC_DATA18 Others: Reserved
31	Reserved		

Note: These bits can be rewritten only when STARTCEN=0 (confirming no ongoing conversion).

24.5.5 ADC configuration register 2 (ADC_CFG2)

Offset address: 0x10

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	SEQEN	R/E	Sequential Section Sampling Enable 0: Disable 1: Enable
7:1	Reserved		
22:8	TGAP	R/W	Time of The Gap between Conversions in Sequential Section Configure This bit configures the time interval (in PCLK) from the completion of the last conversion to the start of the next sampling in each segment of the sequence. 0x0: No interval 0x1: 1 PCLK clock cycle interval 0x2: 2 PCLK clock cycle intervals .....

Field	Name	R/W	Description
			0x7FFF: 32767 PCLK clock cycle intervals
25:23	Reserved		
26	CLKSEL	R/W	ADC Conversion and Analog Clock Select When CLKCFG=00: 0: Use ADCCLK (asynchronous clock) 1: Use PCLK or PCLK frequency division clock; refer to [29:27] bits for frequency division factor
29:27	PCLKDIV	R/W	PCLK Division Factor Configure When CLKSEL=1, this bit configures how many frequency divisions PCLK will go through before it is used as the clock of ADC analog part. 000: No frequency division 001: 2 divided frequency 010: 3 divided frequency 011: 4 divided frequency 100: 5 divided frequency 101: 6 divided frequency 110: 7 divided frequency 111: 8-divided frequency
31:30	CLKCFG	R/W	ADC Clock Mode Configure 00: Frequency division clock of ADCCLK (asynchronous clock mode) or PCLK or PCLK 01: PCLK/2 (synchronous clock mode) 10: PCLK/4 (synchronous clock mode) 11: Reserved Note: The software allows writing these bits only when ADC is disabled.

Note: This register can be rewritten only when ADC is disabled (STARTCEN=0,DISCEN=0,CAL=0,STOPCEN=0,ADCEN=0).

24.5.6 ADC sampling time register 1 (ADC_SMPTIM)

Offset address: 0x14

Reset value: 0x0000 0000

Field	Name	R/W	Description
2:0	SMPCYCSEL	R/W	Sampling Cycles Selecte 000: 1.5 ADC clock cycles 001: 7.5 ADC clock cycles 010: 13.5 ADC clock cycles 011: 28.5 ADC clock cycles 100: 41.5 ADC clock cycles 101: 55.5 ADC clock cycles 110: 71.5 ADC clock cycles 111: 239.5 ADC clock cycles Note: These bits can be rewritten only when STARTCEN=0.
31:3	Reserved		

24.5.7 ADC configuration register 3 (ADC_CFG3)

Offset address: 0x18

Reset value: 0x0000 0000

If the sequential segmented sampling function is not used, there is no need to configure this register.

Field	Name	R/W	Description
1:0	EXTPOLSEL2	R/W	External Trigger Enable and Polarity Select 00: Invalid (this bit cannot be 00 in segmented sampling function) 01: Hardware trigger detection on rising edge 10: Hardware trigger detection on falling edge 11:: Hardware trigger detection on both rising and falling edges
5:2	EXTTRGSEL2	R/W	External Trigger Event Select These bits are used to select the external event for triggering ADC 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 1000: Event 8 1001: Event 9 1010: Event 10 1011: Event 11 1100: Event 12 1101: Event 13: 1110: Event 14 1111: Event 15
7:6	Reserved		
9:8	EXTPOLSEL3	R/W	External Trigger Enable and Polarity Select 00: Invalid (this bit cannot be 00 in segmented sampling function) 01: Hardware trigger detection on rising edge 10: Hardware trigger detection on falling edge 11:: Hardware trigger detection on both rising and falling edges
13:10	EXTTRGSEL3	R/W	External Trigger Event Select These bits are used to select the external event for triggering ADC 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

Field	Name	R/W	Description
			1000: Event 8 1001: Event 9 1010: Event 10 1011: Event 11 1100: Event 12 1101: Event 13: 1110: Event 14 1111: Event 15
31:14	Reserved		

Note: These bits can be rewritten only when STARTCEN=0 (confirming no ongoing conversion).

24.5.8 ADC watchdog threshold register (ADC_AWDT)

Offset address: 0x20

Reset value: 0x0FFF 0000

Field	Name	R/W	Description
11:0	AWDLT[11:0]	R/W	Analog Watchdog Low Threshold
15:12	Reserved		
27:16	AWDHT[11:0]	R/W	Analog Watchdog High Threshold
31:28	Reserved		

Note: These bits can be rewritten only when STARTCEN=0.

24.5.9 ADC channel selection register (ADC_CHSEL)

Offset address: 0x28

Reset value: 0x0000 0000

Field	Name	R/W	Description
17:0	CHxSEL	R/W	Channel-x Select 0: Input channel x is not selected as conversion channel 1: Input channel x is selected as conversion channel
31:18	Reserved		

Note: These bits can be rewritten only when STARTCEN=0.

24.5.10 ADC data register (ADC_DATA)

Offset address: 0x40

Reset value: 0x0000 0000

Field	Name	R/W	Description
15:0	CDATA[15:0]	R	Converted Data These bits are read-only. Include the conversion result values of last conversion channel. CDATA[6:0] value is calibration factor only when calibration is completed.
31:16	Reserved		

24.5.11 ADC sequential segmented sampling data register (ADC_DATAx) (x=0...18)

Offset address: 0x44+4x (x=0...18)

Reset value: 0x0000 0000

Field	Name	R/W	Description
15:0	ADC_DATAx	R/W	Sequential Section Sampling Data (x=0...18) The result of sequential segmented sampling will be put into the 19 data registers according to the inherent rules. For example, to configure three-segment sequential sampling, if the total number of channels in the first segment is 1, the total number of channels in the second segment is 2, and the total number of channels in the third segment is 3, the data result of the first segment will appear in ADC_DATA0, the data result of the second segment will appear in ADC_DATA1-2, and the data result of the third segment will appear in ADC_DATA3-5. Although the ADC segmented sampling mode supports out of sequence (namely, sampling of the second and third segments can precede the first segment), the corresponding result register of each segment is fixed.
31:16	Reserved		

24.5.12 ADC sequential segmented sampling control register (ADC_SEQ_NUM)

Offset address: 0x100

Reset value: 0x0000 0000

Field	Name	R/W	Description
4:0	SEQ_NUM1	R/W	Section1 Transmission Time Setup 00000: Transmission once 00001: Transmission twice ..... 10000: Transmission for 17 times 10001: Transmission for 18 times 10010: Transmission for 19 times Others: Invalid
9:5	SEQ_NUM2	R/W	Section2 Transmission Time Setup 00000: Transmission once 00001: Transmission twice ..... 10000: Transmission for 17 times 10001: Transmission for 18 times Others: Invalid
14:10	SEQ_NUM3	R/W	Section2 Transmission Time Setup 00000: Transmission once 00001: Transmission twice ..... 10000: Transmission for 17 times Others: Invalid
27:15	Reserved		

Field	Name	R/W	Description
29:28	SG_NUM	R/W	The Number of Sequencial Section 00: Segment 1 01: Segment 2 10: Segment 3 11: Invalid
31:30	Reserved		

24.5.13 ADC1st-6th-time transmission channel register (ADC_CHANNEL1)

Offset address: 0x104

Reset value: 0x0000 0000

Field	Name	R/W	Description
4:0	CH0	R/W	Number of Channels Converted for The First Time
9:5	CH1	R/W	Number of Channels Converted for The Second Time
14:10	CH2	R/W	Number of Channels Converted for The Third Time
19:15	CH3	R/W	Number of Channels Converted for The Forth Time
24:20	CH4	R/W	Number of Channels Converted for The Fifth Time
29:25	CH5	R/W	Number of Channels Converted for The Sixth Time
31:30	Reserved		

24.5.14 ADC7th-12th-time transmission channel register (ADC_CHANNEL2)

Offset address: 0x108

Reset value: 0x0000 0000

Field	Name	R/W	Description
4:0	CH6	R/W	Number of Channels Converted for The 7th Time
9:5	CH7	R/W	Number of Channels Converted for The 8th Time
14:10	CH8	R/W	Number of Channels Converted for The 9th Time
19:15	CH9	R/W	Number of Channels Converted for The 10th Time
24:20	CH10	R/W	Number of Channels Converted for The 11th Time
29:25	CH11	R/W	Number of Channels Converted for The 12th Time
31:30	Reserved		

24.5.15 ADC13th-18th-time transmission channel register (ADC_CHANNEL3)

Offset address: 0x10C

Reset value: 0x0000 0000

Field	Name	R/W	Description
4:0	CH12	R/W	Number of Channels Converted for The 13th Time

Field	Name	R/W	Description
9:5	CH13	R/W	Number of Channels Converted for The 14th Time
14:10	CH14	R/W	Number of Channels Converted for The 15th Time
19:15	CH15	R/W	Number of Channels Converted for The 16th Time
24:20	CH16	R/W	Number of Channels Converted for The 17th Time
29:25	CH17	R/W	Number of Channels Converted for The 18th Time
31:30	Reserved		

24.5.16 ADC19th-time transmission channel register (ADC_CHANNEL4)

Offset address: 0x110

Reset value: 0x0000 0000

Field	Name	R/W	Description
4:0	CH18	R/W	Number of Channels Converted for The 19th Time
31:5	Reserved		

24.5.17 ADC general-purpose configuration register (ADC_CCFG)

Offset address: 0x308

Reset value: 0x0000 0000

Field	Name	R/W	Description
21:0	Reserved		
22	VREFEN	R/W	V_{REFINT} Enabl 0: Disable 1: Enable
23	TSEN	R/W	Temperature Sensor Enable 0: Disable 1: Enable
24	HLAF_VDDEN	R/W	1/2*VDD Enable 0: Disable 1: Enable
31:25	Reserved		

Note: These bits can be rewritten only when STARTCEN=0.

25 Comparator (COMP)

25.1 Full name and abbreviation of terms

Table 99 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

25.2 Introduction

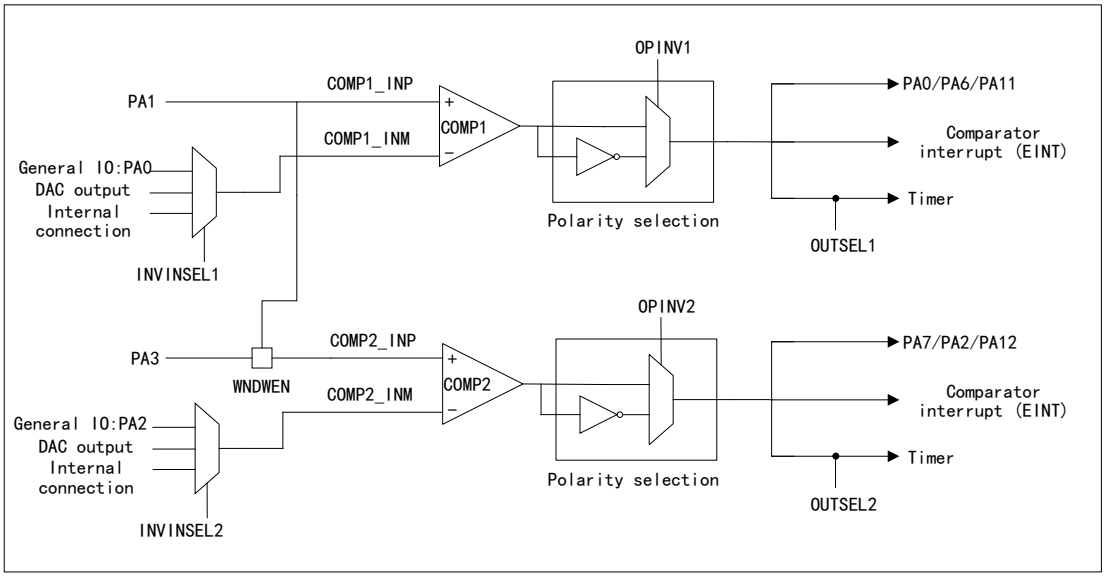
Two independent general-purpose comparators (COMP1 and COMP2) are embedded in MCU, and they can be used in combination with the timer.

25.3 Main characteristics

- Rail-to-rail input
- Two comparators can be combined to form a window comparator
- Hysteresis, rate and loss are programmable
- Can generate an interrupt
- Can wake up from the sleep mode and stop mode (by EINT)
- Can work in stop mode

25.4 Structure block diagram

Figure 109 COMP Structure Block Diagram



25.5 Functional description

25.5.1 COMP clock

COMP has no separate clock enable control bit and works independent of PCLK clock, but its clock is synchronized with PCLK.

COMP can reset the module only by system reset.

25.5.2 COMP input

When input as a comparator, GPIO is required to configure as analog mode.

COMP input consists of non-inverting input and inverting input. All non-inverting inputs are connected to external IO; the inverting input can be programmed and selected, and the external connection has IO pin and DAC output pin; the internal connection has internal reference voltage (V_{REFINT}), and 1/4 or 1/2 or 3/4 of internal reference voltage.

25.5.3 COMP output

The output of the comparator can be connected to the external IO port;

It can also be connected to the following signals of the internal timer:

- Braking signal of PWM
- Input signal of OCREF_CLR
- Input capture channel of timer

The output polarity can be modified by programming the OPINVx bit in COMP_CSTS register

25.5.4 COMP mode

The rate and loss of the comparator are programmable. Considering the practical application, we can program the MODx bit in the register to achieve the most appropriate state.

The comparator has programmable hysteresis function, and can choose the appropriate hysteresis time to avoid the invalid output caused by noise.

Window comparator mode

Connect the non-inverting input of COMP1 and COMP2, and disconnect the connection with PA3.

25.5.5 COMP interrupt

The comparator output is internally connected to the external interrupt and event controller. If the external interrupt is configured correctly, through DAC output, the an interrupt can be generated or MCU entering the sleep and stop mode can be woken up.

25.6 Register address mapping

Table 100 Register Address Mapping

Register name	Description	Offset address
COMP_CSTS	COMP control status register	0x1C

25.7 Register functional description

25.7.1 COMP control status register (COMP_CSTS)

Address offset: 0x1C

Reset value: 0x0000 0000

The control bit of COMP can have R/W two states. RW/R means read/write or read-only state.

Field	Name	R/W	Description
0	EN1	RW/R	COMP1 Enable 0: Disable 1: Enable
1	SW1	RW/R	Comparator COMP1 non-inverting input DAC switch (COMP1 Non Inverting Input on PA1 And PA4 Switch) When this bit is set, disable the switch between the non-inverting input of the general-purpose comparator COMP1 on PA1 and the I/O of PA4 (DAC). 0: Open 1: Close
3:2	MOD1	RW/R	COMP1 Mode

Field	Name	R/W	Description
			Control the working mode of the general-purpose comparator COMP1, namely working rate and loss. 00: High rate/Full power 01: Medium rate/Medium power 10: Low rate/Low power 11: Very low rate/Very low power
6:4	INVINSEL1	RW/R	COMP1 Inverting Input Select Select the inverting input signal source connected to the comparator COMP1. 000: 1/4 of V_{REFINT} 001: 1/2 of V_{REFINT} 010: 3/4 of V_{REFINT} 011: V_{REFINT} (internal reference voltage) 100: COMP1_INM4 (PA4, i.e. DAC_OUT1) 101：COMP1_INM5（PA5） 110：COMP1_INM6（PA0） 111: Reserved.
7	Reserved		
10:8	OUTSEL1	RW/R	COMP1 Output Select These bits are used to select the output direction of the Comparator COMP1. 000: No selection 001: TMR1 interrupt input 010: TMR1 input capture 1 011: TMR 1OCrefclear input 100: TMR 2 input capture 4 101: TMR 2OCrefclear input 110: TMR 3 input capture 1 111: TMR 3OCrefclear input
11	OPINV1	RW/R	COMP1 Output Polarity Invert Output polarity of inverting comparator COMP1 0: Non-inverting ouput 1: Inverting output.
13:12	HYSCFG1	RW/R	COMP1 Hysteresis Level Configure Configure the hysteresis level of COMP1 00: No hysteresis 01: Low degree of hysteresis 10: Medium degree of hysteresis 11: High degree of hysteresis
14	OUTSTS1	R	COMP1 Output State 0: Low output Under the condition of non-inverting output: the non-inverting input is lower than the inverting input, and the output is low level Under the condition of inverting output: the non-inverting input is higher than the inverting input, and the output is low level 1: High output

Field	Name	R/W	Description
			Under the condition of non-inverting output: the non-inverting input is higher than the inverting input, and the output is high level Under the condition of inverting output: the non-inverting input is lower than the inverting input, and the output is high level
15	LOCK1	R/S	COMP1 Lock This bit can be written once only and is set by software and can only be cleared by system reset. When locked, all control bits of COMP1 will become read-only. 0: COMP1 control bit is readable and writable 1: COMP1 control bit is read-only
16	EN2	RW/R	COMP2 Enable 0: Disable 1: Enable
17	Reserved		
19:18	MOD2	RW/R	COMP2 Mode Control the working mode of the general-purpose comparator COMP2, namely working rate and loss. 00: High rate/Full power 01: Medium rate/Medium power 10: Low rate/Low power 11: Very low rate/Very low power
22:20	INVINSEL2	RW/R	COMP2 Inverting Input Select Select the inverting input signal source connected to the comparator COMP2. 000: 1/4 of V_{REFINT} 001: 1/2 of V_{REFINT} 010: 3/4 of V_{REFINT} 011: V_{REFINT} (internal reference voltage) 100: COMP2_INM4 (PA4, namely DAC_OUT1) 101：COMP2_INM5（PA5） 110：COMP2_INM6（PA0） 111: Reserved
23	WMODEN	RW/R	Window Mode Enable Two general-purpose comparators can be connected to become the window comparator mode. 0: Disable 1: Enable
26:24	OUTSEL2	RW/R	COMP2 Output Select Select the output direction of the comparator COMP2. 000: No selection 001: Timer 1 interrupt input 010: Timer 1 input capture 1 011: Timer 1OCrefclear input 100: Timer 2 input capture 4 101: Timer 2OCrefclear input 110: Timer 3 input capture 1 111: Timer 3OCrefclear input

Field	Name	R/W	Description
27	OPINV2	RW/R	COMP2 Output Polarity Invert Output polarity of reverse comparator COMP2 0: Non-inverting ouput 1: Inverting output
29:28	HYSCFG2	RW/R	COMP2 Hysteresis Level Configure 00: No hysteresis 01: Low degree of hysteresis 10: Medium degree of hysteresis 11: High degree of hysteresis
30	OUTSTS2	R	COMP2 Output State 0: Low output Under the condition of non-inverting output: the non-inverting input is lower than the inverting input, and the output is low level Under the condition of inverting output: the non-inverting input is higher than the inverting input, and the output is low level 1: High output Under the condition of non-inverting output: the non-inverting input is higher than the inverting input, and the output is high level Under the condition of inverting output: the non-inverting input is lower than the inverting input, and the output is high level
31	LOCK2	R/S	COMP2 Lock This bit can be written once only and is set by software and can only be cleared by system reset. When locked, all control bits of COMP2 will become read-only. 0: COMP2 control bit is readable and writable 1: COMP2 control bit is read-only

26 Operational amplifier (OPA)

26.1 Introduction

Four independent operational amplifiers (OPA1, OPA2, OPA3 and OPA4) are embedded in MCU, and they can be used in combination withCOMP and ADC.

26.2 Main characteristics

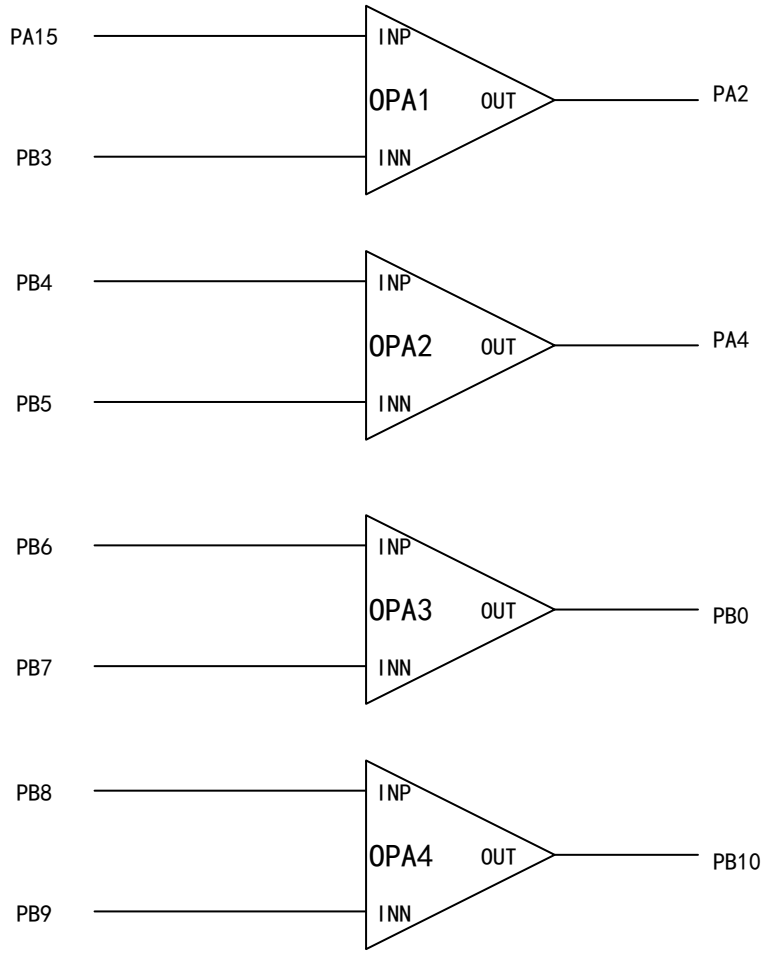
（1） The input and output ends of the operational amplifiers are open. The gain of the operational amplifiers can be adjusted by external resistance, or the internal preset gain gear can be selected by setting.

（2） The amplifier output can be used as the input of the ADC conversion module.

26.3 Structure block diagram

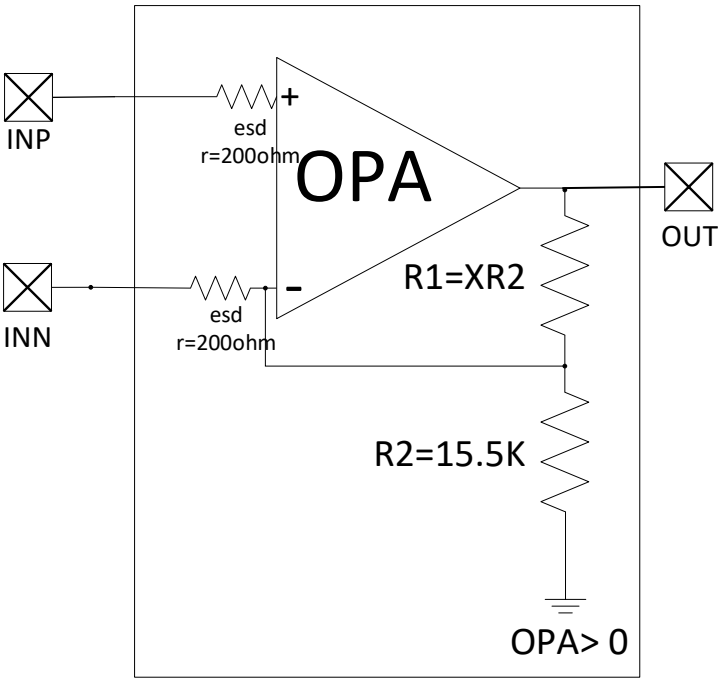
26.3.1 Structure block diagram

Figure 110 OPA Structure Block Diagram



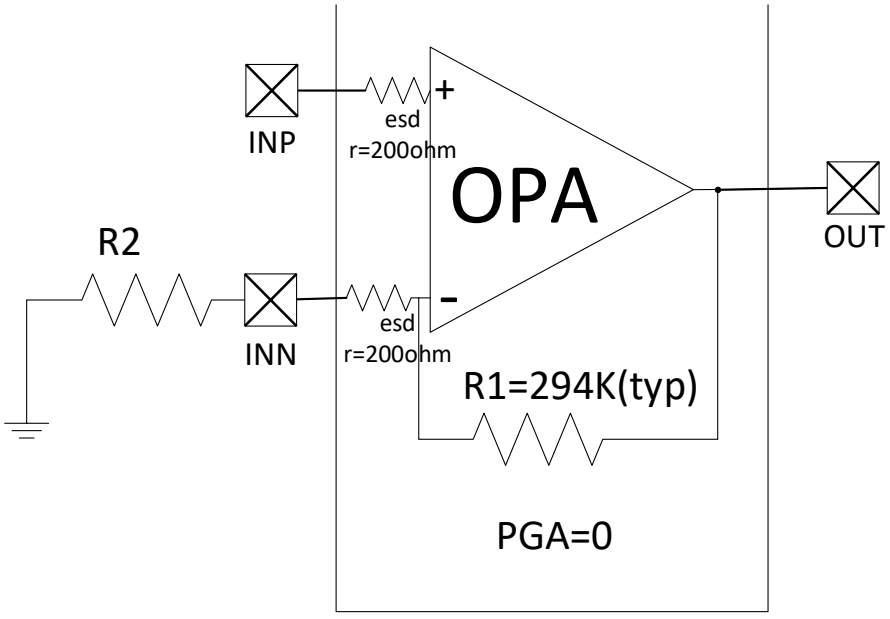
26.3.2 Internal Gain Application Diagram

Figure 111 Internal Gain Application Diagram



26.3.3 External Gain Application Diagram

Figure 112 External Gain Application Diagram



26.4 Register address mapping

Table 101 OPA Register Address Mapping

Register name	Description	Offset address
OPA_CFG	OPA configuration register	0x34

26.5 Register functional description

26.5.1 OPA configuration register (OPA_CFG)

Offset address: 0x34

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	OPA1EN	R/W	OPA1 Enable 0: Disable 1: Enable
1	Reserved		
4:2	OPA1G	R/W	OPA1 Gain Factor 000: Determined by external resistor network, OP1N is inverting input pin of the operational amplifier, and OP1OUT is the output pin of the operational amplifier 001: X1 is determined by internal gain coefficient, and OP1N/OP1OUT is general IO 010: X2 is determined by internal gain coefficient, and OP1N/OP1OUT is general IO 011: X4 is determined by internal gain coefficient, and OP1N/OP1OUT is general IO 100: X8 is determined by internal gain coefficient, and OP1N/OP1OUT is general IO 101: X10 is determined by internal gain coefficient, and OP1N/OP1OUT is general IO 110: X16 is determined by internal gain coefficient, and OP1N/OP1OUT is general IO 111: X20 is determined by internal gain coefficient, and OP1N/OP1OUT is general IO
7:5	Reserved		
8	OPA2EN	R/W	OPA2 Enable 0: Disable 1: Enable
9	Reserved		
12:10	OPA2G	R/W	OPA2 Gain Factor 000: Determined by external resistor network, OP2N is inverting input pin of the operational amplifier, and OP2OUT is the output pin of the operational amplifier 001: X1 is determined by internal gain coefficient, and OP2N/OP2OUT is general IO

Field	Name	R/W	Description
			010: X2 is determined by internal gain coefficient, and OP2N/OP2OUT is general IO 011: X4 is determined by internal gain coefficient, and OP2N/OP2OUT is general IO 100: X8 is determined by internal gain coefficient, and OP2N/OP2OUT is general IO 101: X10 is determined by internal gain coefficient, and OP2N/OP2OUT is general IO 110: X16 is determined by internal gain coefficient, and OP2N/OP2OUT is general IO 111: X20 is determined by internal gain coefficient, and OP2N/OP2OUT is general IO
15:13	Reserved		
16	OPA3EN	R/W	OPA3 Enable 0: Disable 1: Enable
17	Reserved		
20:18	OPA3G	R/W	OPA3 Gain Factor 000: Determined by external resistor network, OP3N is inverting input pin of the operational amplifier, and OP3OUT is the output pin of the operational amplifier 001: X1 is determined by internal gain coefficient, and OP3N/OP3OUT is general IO 010: X2 is determined by internal gain coefficient, and OP3N/OP3OUT is general IO 011: X4 is determined by internal gain coefficient, and OP3N/OP3OUT is general IO 100: X8 is determined by internal gain coefficient, and OP3N/OP3OUT is general IO 101: X10 is determined by internal gain coefficient, and OP3N/OP3OUT is general IO 110: X16 is determined by internal gain coefficient, and OP3N/OP3OUT is general IO 111: X20 is determined by internal gain coefficient, and OP3N/OP3OUT is general IO
23:21	Reserved		
24	OPA4EN	R/W	OPA4 Enable 0: Disable 1: Enable
25	Reserved		
28:26	OPA4G	R/W	OPA4 Gain Factor 000: Determined by external resistor network, OP4N is inverting input pin of the operational amplifier, and OP4OUT is the output pin of the operational amplifier 001: X1 is determined by internal gain coefficient, and OP4N/OP4OUT is general IO 010: X2 is determined by internal gain coefficient, and OP4N/OP4OUT is general IO

Field	Name	R/W	Description
			011: X4 is determined by internal gain coefficient, and OP4N/OP4OUT is general IO 100: X8 is determined by internal gain coefficient, and OP4N/OP4OUT is general IO 101: X10 is determined by internal gain coefficient, and OP4N/OP4OUT is general IO 110: X16 is determined by internal gain coefficient, and OP4N/OP4OUT is general IO 111: X20 is determined by internal gain coefficient, and OP41N/OP4OUT is general IO
31:29	Reserved		

27 Cyclic redundancy check computing unit (CRC)

27.1 Introduction

The cyclic redundancy check (CRC) computing unit can get 8/16/32-bit CRC computing result by calculating the input data through a fixed generator polynomial, which is mainly used to detect or verify the correctness and integrity of the data after transmission or saving.

27.2 Functional description

27.2.1 Calculation method

Use CRC-32 (Ethernet) polynomial: 0x4C11DB7

$$(X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^{8}+X^{7}+X^{5}+X^{4}+X^{2}+X+1)$$

27.2.2 Calculating Time

- When processing 32-bit data, the calculation time is 4 AHB clock cycles
- When processing 16-bit data, the calculation time is 2 AHB clock cycles
- When processing 8-bit data, the calculation time is 1 AHB clock cycle

27.2.3 Functional characteristics

- Process 8-bit, 16-bit and 32-bit data
- It can use fully programmable polynomial of which the bits are programmable (7 bits, 8 bits, 16 bits, or 32 bits)
- Programmable CRC initial value
- Independent 32-bit input/output register
- It can be used as a general-purpose 8-bit register for temporary storage
- Reversible option of I/O data
- The data width can be dynamically adjusted to reduce the number of times of calculating and writing
- The high and low bits of input data can be inverted in order to adapt to different data storage methods (byte, half word or word, little-endian and big-endian system)
- Word or byte calculation can be performed, depending on the different data formats written
- Have input buffer to reduce wait cycles and avoid bus blocking

CRC unit contains a 32-bit read/write register CRC_DATA, used to write new data and give CRC computing results. Every time a new data is written, the result will be a combination of the last calculation result and the new calculation result. (Execute operation for the whole word). CRC_Data can access word or

right-aligned half word or right-aligned bytes, while other registers can only access 32 bits.

Programmable polynomial

Only after the application program is reset or after CRC_DATA is read, can the size of the polynomial be selected or changed by setting the POLSEL bit in CRC_CTRL register, which means that in CRC computing, the value or size of the polynomial cannot be changed.

27.3 Register address mapping

Table 102 CRC Register Address Mapping

Register name	Description	Offset address
CRC_DATA	Data register	0x00
CRC_INDATA	Independent data register	0x04
CRC_CTRL	Control register	0x08
CRC_INITVAL	CRC initial value register	0x10
CRC_POL	CRC polynomial register	0x14

27.4 Register functional description

27.4.1 Data register (CRC_DATA)

Offset address: 0x00

Reset value: 0xFFFF FFFF

Field	Name	R/W	Description
31:0	DATA	R/W	32bit Data) As an input register: Store the new data of CRC calculator when writing. As an output register: Return the results of CRC computing when reading.

27.4.2 Independent data register (CRC_INDATA)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
7:0	INDATA	R/W	Independent 8bit Data Can be used for temporary storage of 1-byte data. CRC rest generated by RST bit of the register CRC_CTRL has no effect on this register.
31:8	Reserved.		

Note: This register does not take part in calculation and can store any data.

27.4.3 Control register (CRC_CTRL)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	RST	R/S	Reset CRC Calculation Unit Set the data register to 0xFFFF FFFF. It can only set this bit, which shall be automatically cleared to 0 by hardware.
2:1	Reserved		
4:3	POLSEL	R/W	Programmable Polynomial Size Select 00: 32 bits 01: 16 bits 10: 8 bits 11: 7 bits
6:5	REVI	R/W	Input Data Reverse Reverse the input data in different units. 00: Not reverse 01: In byte 10: In unit 11: In word
7	REVO	R/W	Output Data Reverse 0: Not reverse 1: Reverse
31:8	Reserved		

27.4.4 CRC initial value register (CRC_INITVAL)

Offset address: 0x10

Reset value: 0xFFFF FFFF

Field	Name	R/W	Description
31:0	VALUE	R/W	Initial CRC Value The CRC initial value is programmable, and this bit is used to set the initial value of CRC.

27.4.5 CRC polynomial register (CRC_POL)

Offset address: 0x14

Reset value: 0x04C11DB7

Field	Name	R/W	Description
31:0	PPOL	R/W	Programmable Polynomial Programmable polynomial coefficients can be written. If the coefficient is less than 32 bits, the correct value must be programmed with the least significant bit.

28 Chip Electronic Signature

28.1 Introduction

The chip electronic signature includes Flash capacity information of main memory and 96-bit unique chip ID, which have been written into the system memory area of the chip before leaving the factory, and are read-only and can not be modified by users.

28.2 Functional description

Main use of 96-bit chip ID:

- Used as serial number
- As the password, when writing the flash memory, the code and password can be combined by algorithm to improve the security of the codes in Flash.
- Used for startup configuration
- The reference number provided by the identity label is unique to any MCU series. Users cannot change the unique ID under any circumstances. According to different usages, users can choose to read the identity label in byte, half word, or full word.

28.3 Register functional description

28.3.1 96-bit unique chip ID

Base address: 0x1FFF F7AC

Offset address: 0x00

Field	Name	R/W	Description
31:0	U_ID[31:0]	R	Unique identity flag 31:0 bits

Offset address: 0x04

Read-only, the value has been prepared before leaving the factory

Field	Name	R/W	Description
31:0	U_ID[63:32]	R	Unique identity flag 63:32 bits

Offset address: 0x08

Read-only, the value has been prepared before leaving the factory

Field	Name	R/W	Description
31:0	U_ID[95:64]	R	Unique identity flag 95:64 bits

28.3.2 Flash capacity register (16 bits)

Base address: 0x1FFF F7CC

Offset address: 0x00

Read-only, the value has been prepared before leaving the factory

Field	Name	R/W	Description
15:0	F_SIZE	R	Flash Size Indicate the capacity of main memory area of the product (in Kb). For example: 0x0040=64 Kb

29 Version history

Table 103 Document Version History

Date	Version	Revision History
2023.3	V0.1	New
2023.7	V0.2	Modify the format
2023.11	V0.3	(1) Modify the I/O structure diagram of the GPIO chapter

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, it shall not be deemed that Geehy authorizes users to use the aforesaid third party’s products, services or 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 may occur inevitably. 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 loses 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 promise 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 "as is". 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 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 "as is" shall not be liable for damages, including any general damages, special direct, indirect or collateral damages arising from the use or no use of the information in this manual (including without limitation data loss or inaccuracy, or losses suffered by users or third parties).

8. Scope of application

The information in this manual replaces the information provided in all previous versions of the manual.