1 Introduction

This technical reference manual serves as a supplement to the SC32F15G datasheet, providing the necessary information for applications, especially software development. For details regarding the functional features, ordering information, as well as mechanical and electrical characteristics of specific SC32F15G devices, please refer to their respective datasheets.

Content

2 Document Conventions

2.1 Glossary

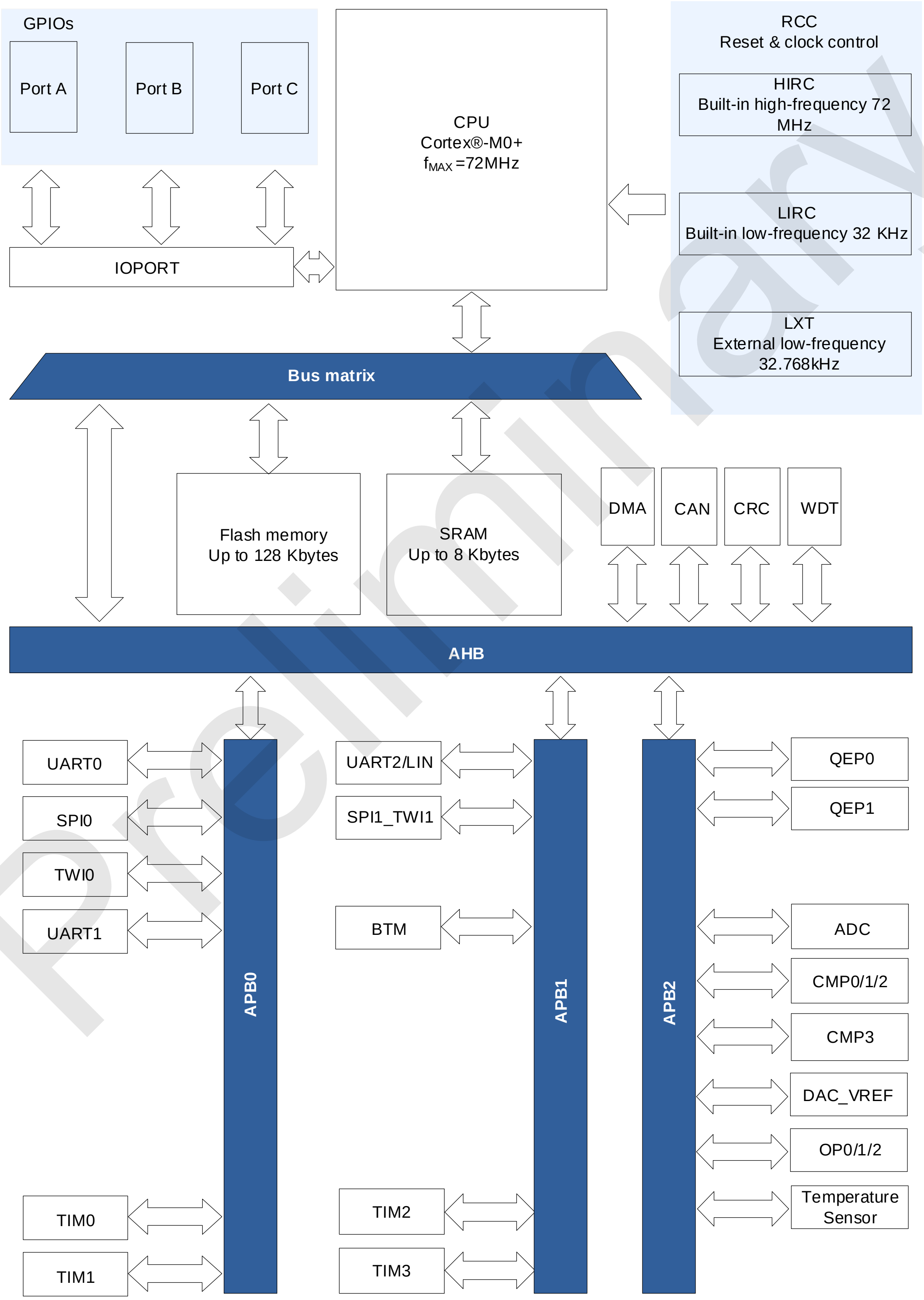
This section primarily explains the definitions of abbreviations and acronyms used in this document:

- Word: 32-bit data
- Half-word: 16-bit data
- Byte: 8-bit data
- Double word: 64-bit data
- IAP (In-Application Programming): IAP refers to the ability to reprogram the microcontroller's Flash during the execution of user programs.
- ICP (In-Circuit Programming): ICP refers to the ability to program the microcontroller's Flash when the device is installed on a user's circuit board, using the JTAG protocol, SWD protocol, or bootloader.
- ISP (In-System Programming): ISP refers to programming using a bootloader in conjunction with peripheral interfaces such as UART/SPI for programming
- JTAG protocol: JTAG protocol is an international standard testing protocol primarily used for internal chip testing.
- SWD protocol: SWD protocol, designed by ARM, represents Serial Wire Debug and is used for programming and debugging ARM microcontrollers.
- Option Byte: Configuration bits stored in Flash.
- AHB: Advanced High-Performance Bus
- APB: Advanced Peripheral Bus

2.2 Availability of peripherals

For information on the availability and quantity of peripherals for various product models, please refer to the latest data sheets in the product peripheral resource table section.

3 Resource Diagram



4 Power, Reset And System Clock (RCC)

4.1 Power-on Reset

After the SC32F15G power-on, the processes carried out before execution of client software are as follows:

① Reset stage
② Loading information stage
③ Normal operation stage

4.1.1 Reset Stage

The SC32F15G will always be reset until the voltage supplied to SC32F15G is higher than a certain voltage, and the internal Clock starts to be effective. The duration of reset stage is related to rising speed of external power. Once the external supply voltage is up to built-in POR voltage, the reset stage would be completed.

4.1.2 Loading Information Stage

There is a warm-up counter inside The SC32F15G. During the reset stage, the warm-up counter is cleared to 0 until the voltage exceeds the POR voltage, the built-in HIRC oscillator starts to oscillate, and the warm-up counter starts counting. When the internal warm-up counter counts to a certain number, every certain number of HIRC clocks will read a byte of data from the IFB (including Customer Option) in the Flash ROM and store it in the internal system register. This reset signal will not end until the warm-up is completed.

4.1.3 Normal Operation Stage

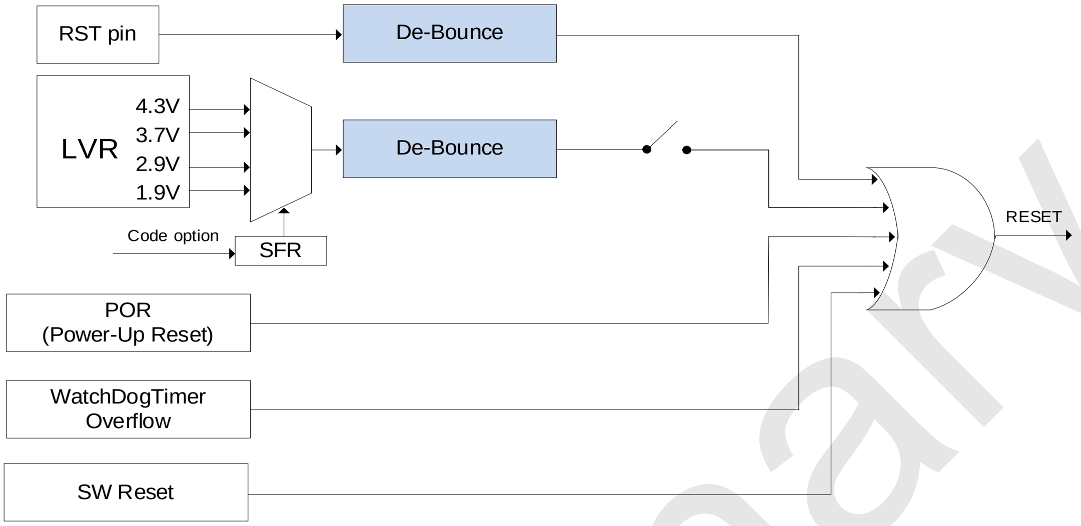
After finishing the Loading Information stage, The SC32F15G starts to read the instruction code from Flash and enters the normal operation stage. The LVR voltage is the set value of Customer Option written by the user.

4.2 Reset Modes

The SC32F15G has 5 reset methods, the first four are hardware reset:

- External reset
- Low-voltage reset LVR
- Power-on reset POR
- Watchdog WDT reset
- Software reset

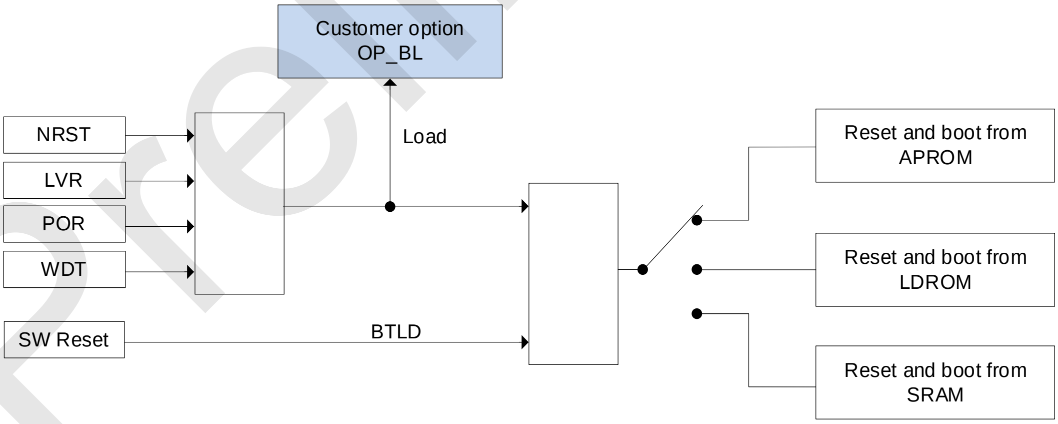
The circuit diagram of the reset part of the SC32F15G is as follows:



SC32F15G Reset Circuit Diagram

4.2.1 Boot area after the reset

After hardware reset through external RST, low voltage reset (LVR), power-on reset (POR), or watchdog reset (WDT), the chip boots from the startup area (APROM / LDROM / SRAM) set by the user in OP_BL. After the software reset, the chip boots from the startup area (APROM / LDROM / SRAM) set by BTLD[1:0].



SC32F15G Boot Area Switching diagram after reset

4.2.2 External RST

External reset is a low-level reset pulse signal of a certain width given to SC32F15G from external RST pin to realize the reset of SC32F15G. User can configure the PC11/NRST pin as RST (reset pin) using the programming host software by Customer Option before programming.

4.2.3 Low-voltage Reset LVR

The SC32F15G provides a low-voltage reset circuit. There are 4-level LVR voltage options: 4.3V, 3.7V, 2.9V, 1.9V. The default value is the Customer Option value written by the user. A reset occurs when the VDD voltage is less than the threshold voltage for low-voltage reset and the duration is greater than TLVR. Among them, T_{LVR} is the buffeting time of LVR, about 30μs.

4.2.4 Power-on Reset(POR)

The SC32F15G has a power-on reset circuit inside. When the power supply voltage VDD reaches the POR reset voltage, the system automatically resets.

4.2.5 Watchdog Reset(WDT)

The SC32F15G has a WDT, the clock source of which is the built-in 32 kHz oscillator. The user can choose whether to enable the watchdog reset function by Customer Option.

4.2.6 Software Reset

Enable RST(IAP_CON.8) will immediately reset the system.

4.2.7 Initial Reset State

When SC32F15G is in the reset state, most registers return to their initial state. The watchdog (WDT) is in the disabled state. 'Hot-start' resets (such as WDT, LVR, software reset, etc.) do not affect SRAM, and SRAM values remain the same as before the reset. Loss of SRAM content occurs when the power supply voltage drops to a level where RAM cannot retain data.

4.3 Clock

4.3.1 System Clock Source

Three different clock sources can be used to drive the system clock (SYSCLK):

- Built-in high-frequency 72MHz oscillator (HIRC)
- Built-in low-frequency 32kHz oscillator (LIRC)
- External low-frequency crystal oscillator (LXT)

Note:

1. **The default system clock source at power-up is HIRC, and its frequency is $f_{HIRC/2}$. Users can switch the clock source through software during normal operation after power-up. Before switching, ensure that the selected clock source is in a stable operating state.**
2. **Regardless of the chosen clock source to switch to, the system clock source must first be switched to HIRC before transitioning to the target clock source.**

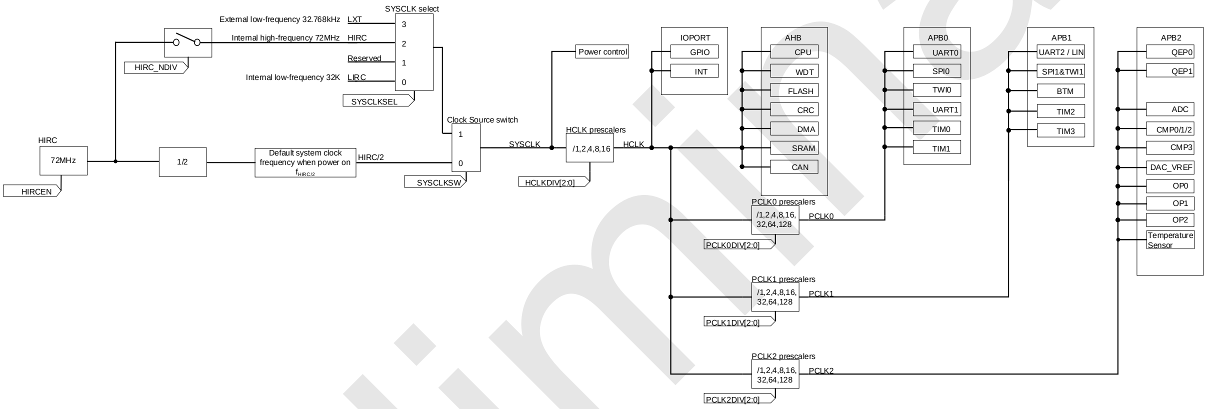
4.3.2 Bus

Users can configure the frequencies of the AHB, APB0, APB1, and APB2 domains through multiple prescalers.

- HCLK: The main clock of the AHB domain, with a maximum frequency of 72MHz. It drives components such as the Cortex®-M0+ core, memory, and DMA.
- PCLK0: The main clock of the APB0 domain, with a maximum frequency equal to the HCLK frequency. Peripheral devices on the APB0 bus are driven by PCLK0.
- PCLK1: The main clock of the APB1 domain, with a maximum frequency equal to the HCLK frequency. Peripheral devices on the APB1 bus are driven by PCLK1.
- PCLK2: The main clock of the APB2 domain, with a maximum frequency equal to the HCLK frequency. Peripheral devices on the APB2 bus are driven by PCLK2.

The RCC divides the AHB clock (HCLK) by 8 to serve as the external clock for SysTick. By setting the control and status registers of SysTick, you can choose either the above-mentioned clock or the core clock as the SysTick clock source.

4.3.3 Clock and Bus Allocation Block Diagram



Clock and Bus Allocation Block Diagram

Note: Default system clock frequency when power on "f_{SYS}" is fHIRC/2, users can change clock source by modify SYSCLKSW or SYSCLKSEL.

4.4 Built-in high-frequency 72MHz Oscillator (HIRC)

HIRC has the following functions and features:

- Can be selected as the system operating clock
- Default system clock frequency when power on "f_{SYS}" is f_{HIRC}/2
- Frequency error: Within ±1% @ -40 to 105℃ @ 2.0V to 5.5V
- The system clock can be automatically calibrated by 32.768 kHz external crystal oscillator, after calibration HIRC accuracy can be infinitely close to the accuracy of external 32.768 kHz crystal oscillator

4.5 Built-in Low-Frequency 32kHz Oscillator (LIRC)

LIRC has the following functions and features:

- Can be selected as the system operating clock
- Can be selected as the Base Timer clock source
- Fixed as the WDT clock source, which will be automaticlly enabled when WDT is enabled
- Frequency error: Within ±4% @ -20 to 85℃ @ 4.0V to 5.5V, after register correction

4.6 External Low-Frequency Oscillator Circuit, Can Connect to 32.768kHz Oscillator (LXT)

LXT has the following functions and features:

- Can be selected as the system operating clock
- Can be selected as the Base Timer clock source
- Allows for an external 32.768kHz low-frequency oscillator
- Automatic calibration of HIRC can be performed using LXT

4.7 RCC Register

4.7.1 RCC Related Register

4.7.1.1 RCC Protect Register (RCC_KEY)

Register	R/W	Description	Reset Value	POR
RCC_KEY	R/W	RCC Protect Register	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
7~0	RCCKEY[7:0]	The operation enable switch and timing limit settings for RCC_CFG0, RCC_CFG1 registers. Write a value “n” greater than or equal to 0x40 means: 1. Enable the write operation function for RCC_CFG0, RCC_CFG1 registers. 2. If no register write command is received after “n” system clock, the RCC rewrite function will be disabled again.
31~8	-	Reserved

4.7.1.2 System Clock Source Selection Register (RCC_CFG0) (Write Protection)

***This register is write-protected and can only be modified by manipulating the RCCprotection register RCC_KEY.**

Register	R/W	Description	Reset Value	POR
RCC_CFG0	R/W	System Clock Source Selection Register	0x0000_1040	0x0000_1040

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
INTEN	HIRC_NDIV	WAIT[1:0]		HPLDO_DP	-	SYSCLKSEL[1:0]	
7	6	5	4	3	2	1	0
SYSCLKSW	HIRCEN	-	-	-	-	LIRCEN	LXTEN

Bit number	Bit Mnemonic	Description
15	INTEN	Interrupt Request CPU Enable Control Bit 0: Disable interrupt request 1: Enable interrupt request
14	HIRC_NDIV	Bulit-in High Frequency Oscillator Output 72MHz Enable Bit 0: Disable 1: Enable
13~12	WAIT[1:0]	00: Reserved, “00” is not recommanded for users to set 01: 1 wait, 36 MHz clock frequency recommanded 10: 2 wait, 72 MHz clock frequency recommanded 11: 3 wait, 72 MHz clock frequency recommanded Note: When the user sets the main frequency to 36M, at least 1 wait is required; when the main frequency is set to 72M, at least 2 waits are required.
11	HPLDO_DP	Low Frequency System Clock Power Consumption Adjust Bit 0: The recommended setting for the system clock source when not using LIRC 1: The recommended setting for the system clock source when using LIRC, when the system clock is set to LIRC, writing this bit to 1 can reduce power consumption
9~8	SYSCLKSEL[1:0]	System Clock Source Selection Bit 00: System clock source is from LIRC 01: Reserved 10: System clock source is from HIRC(72MHz) 11: System clock source is from LXT **Note:** **1. The default system clock source after power-up is HIRC. Users can switch the clock source through software during normal operation after power-up. Before switching, ensure**

Bit number	Bit Mnemonic	Description
		that the selected clock source is in a stable operating state. **2. Regardless of the chosen clock source for switching, the system clock source must first be switched to HIRC before switching to the target clock source.**
7	SYSCLKSW	The system clock source switching bit, when enabled, allows the system clock source to switch from HIRC to the clock selected by SYSCLKSEL: 0: System clock source is HIRC 1: System clock source is the option set by SYSCLKSEL After rewriting this bit, the internal circuit must successfully switch for the updated value to take effect; otherwise, the read value will remain the status before rewriting. Users could determine whether the clock source has successfully switched by reading this bit. This bit will be automatically cleared after a reset/wake-up, meaning that HIRC provides the system clock after a reset/wake-up. **Note:** **1. The default system clock source after power-up is HIRC. Users can switch the clock source through software during normal operation after power-up. Before switching, ensure that the selected clock source is in a stable operating state.** **2. Regardless of the chosen clock source for switching, the system clock source must first be switched to HIRC before switching to the target clock source.**
6	HIRCEN	Built-In High-Frequency 64MHz Oscillator Enable Bit 0: Disable 1: Enable When SYSCLKSW = 0, and HIRC is selected as the system clock, this bit cannot be written. This bit will be set to 1 by hardware after a reset/wake-up, meaning that HIRC provides the system clock after a reset/wake-up.
1	LIRCEN	Built-In Low-Frequency LIRC Oscillator Enable Bit 0: Disable 1: Enable
0	LXTEN	External Low-Frequency LXT Crystal Oscillator Enable Bit 0: Disable 1: Enable
31~16 10 5~2	-	Reserved

4.7.1.3 Peripheral Clock Source Selection Register (RCC_CFG1)(Write Protection)

***This register is write-protected and can only be modified by manipulating the RCCprotection register RCC_KEY.**

Register	R/W	Description	Reset Value	POR
RCC_CFG1	R/W	Peripheral Clock Source Register	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
7~5	STCLKSEL[2:0]	SysTick Clock Source Selection Bit 000: Clock source is from HCLK/8 001: Clock source is from HIRC/4 010: Reserved 011: Clock source is from LIRC 100: Clock source is from LXT **Note: When cofiguring clock source, users should note that if SysTick source is not from HCLK, the clock source frequency of SysTick must equal to or fewer than f_{HCLK}/2.**
0	BTMCLKSEL	BTM Clock Source Selection Bit 0: Clock source is from LIRC 1: Clock source is from LXT After rewriting this bit, the internal circuit must successfully switch for the updated value to take effect; otherwise, the read value will remain the status before rewriting. Users could determine whether the clock source has successfully switched by reading this bit.
31~8 4~1	-	Reserved

4.7.1.4 Clock Status Register (RCC_STS)

Register	R/W	Description	Reset Value	POR
RCC_STS	R/W	Clock Status Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8

-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	SRAMPEIF	-	-	-

Bit number	Bit Mnemonic	Description
3	SRAMPEIF	SRAM Parity Checking Error Flag This bit is set to 1 by hardware when SRAM parity checking error is detected, and cleared by writing 1 through software. 0: SRAM parity checking error detected 1: SRAM parity checking error not detected
31~4 2~0	-	Reserved

4.7.1.5 SysTick Calibration Parameter Register (SYST_CALIB)

Register	R/W	Description	Reset Value	POR
SYST_CALIB	Read Only	SysTick Calibration Parameter Register	0x0000_2327	0x0000_2327

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

Bit number	Bit Mnemonic	Description
23~0	CALIB[23:0]	Calibration Register Default Value: If the default clock after power-up is f_{HCLK}/n (MHz), (n is the default division factor after power-up, and HIRC is the default clock source after power-up). Then the SysTick calibration initial value is set to 1000*(f_{HCLK}/n), this ensures that a default 1ms time base can be generated.
31~24	-	Reserved

4.7.1.6 AHB Bus Peripheral Clock Enable Register (AHB_CFG)

Register	R/W	Description	Reset Value	POR
AHB_CFG	R/W	AHB Bus Peripheral Clock Enable Register	0x0010_0000	0x0010_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-

23	22	21	20	19	18	17	16
-	CLKDIV[2:0]			-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	CANEN	-	IFBEN	CRCEN	DMAEN

Bit number	Bit Mnemonic	Description
22~20	CLKDIV[2:0]	AHB Clock Division Configure Bit The division factor of f_{SYSCLK} to generate $f_{HCLK:}$ 000: f_{HCLK}= f_{SYS} 001: f_{HCLK}= f_{SYS} / 2 010: f_{HCLK}= f_{SYS} / 4 011: f_{HCLK}= f_{SYS} / 8 100: f_{HCLK}= f_{SYS} / 16 Others: Reserved
4	CANEN	CAN Module Clock Enable Bit 0: Disable 1: Enable
2	IFBEN	Customer Option Mapping Register Clock Enable Bit Before rewrite IFB mapping register by OPINX and OPREG,it is necessary to enable IFBEN. 0: Disable 1: Enable
1	CRCEN	CRC Module Clock Enable Bit 0: Disable 1: Enable
0	DMAEN	DMAClock Enable Bit 0: Disable 1: Enable
31~23 19~5 3	-	Reserved

4.7.1.7 APB0 Bus Peripheral Clock Enable Register (APB0_CFG)

Register	R/W	Description	Reset Value	POR
APB0_CFG	R/W	APB0 Bus Peripheral Clock Enable Register	0x0000_0000	0x0000_0000

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

ENAPB	CLKDIV[2:0]			-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
UART1EN	UART0EN	SPI0EN	TWI0EN	-	-	TIM1EN	TIM0EN

Bit number	Bit Mnemonic	Description
23	ENAPB	APB0 Bus Clock Control Bit 0: Disable 1: Enable
22~20	CLKDIV[2:0]	APB0 Clock Division Configure Bit The division factor of f_{HCLK} to generate f_{PCLK0}: 000: $f_{PCLK0} = f_{HCLK}$ 001: $f_{PCLK0}= f_{HCLK} / 2$ 010: $f_{PCLK0}= f_{HCLK} / 4$ 011: $f_{PCLK0}= f_{HCLK} / 8$ 100: $f_{PCLK0}= f_{HCLK} / 16$ 101: $f_{PCLK0}= f_{HCLK} / 32$ 110: $f_{PCLK0}= f_{HCLK} / 64$ 111: $f_{PCLK0}= f_{HCLK} / 128$
7	UART1EN	UART1 Clock Enable Bit 0: Disable 1: Enable
6	UART0EN	UART0 Clock Enable Bit 0: Disable 1: Enable
5	SPI0EN	SPI0 Clock Enable Bit 0: Disable 1: Enable
4	TWI0EN	TWI0 Clock Enable Bit 0: Disable 1: Enable
1	TIM1EN	Timer1 Clock Enable Bit 0: Disable 1: Enable
0	TIM0EN	Timer0 Clock Enable Bit 0: Disable 1: Enable
31~24 19~8 3~2	-	Reserved

4.7.1.8 APB1 Bus Peripheral Clock Enable Register (APB1_CFG)

Register	R/W	Description	Reset Value	POR
APB1_CFG	R/W	APB1 Bus Peripheral Clock Enable Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
ENAPB	CLKDIV[2:0]			-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
UART2EN	-	-	SPI_TWIEN	-	-	TIM3EN	TIM2EN

Bit number	Bit Mnemonic	Description
23	ENAPB	APB1 Bus Clock Control Bit 0: Disable 1: Enable
22~20	CLKDIV[2:0]	APB1 Clock Division Configure Bit The division factor of f_{HCLK} to generate f_{PCLK1}: 000: $f_{PCLK1}= f_{HCLK}$ 001: $f_{PCLK1}= f_{HCLK} / 2$ 010: $f_{PCLK1}= f_{HCLK} / 4$ 011: $f_{PCLK1}= f_{HCLK} / 8$ 100: $f_{PCLK1}= f_{HCLK} / 16$ 101: $f_{PCLK1}= f_{HCLK} / 32$ 110: $f_{PCLK1}= f_{HCLK} / 64$ 111: $f_{PCLK1}= f_{HCLK} / 128$
7	UART2EN	UART2 Clock Enable Bit 0: Disable 1: Enable
4	SPI_TWIEN	SPI_TWI Clock Enable Bit 0: Disable 1: Enable
1	TIM3EN	Timer3 Clock Enable Bit 0: Disable 1: Enable
0	TIM2EN	Timer2 Clock Enable Bit 0: Disable 1: Enable
31~24 19~8 6~5	-	Reserved

Bit number	Bit Mnemonic	Description
3~2		

4.7.1.9 APB2 Bus Peripheral Clock Enable Register (APB2_CFG)

Register	R/W	Description	Reset Value	POR
APB2_CFG	R/W	APB2 Bus Peripheral Clock Enable Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
ENAPB	CLKDIV[2:0]			-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	ADCEN	-	-	QEP1EN	QEP0EN	-	-

Bit number	Bit Mnemonic	Description
23	ENAPB	APB2 Bus Clock Control Bit 0: Disable 1: Enable
22~20	CLKDIV[2:0]	APB2 Clock Division Configure Bit The division factor of f_{HCLK} to generate f_{PCLK2}: 000: $f_{PCLK2}= f_{HCLK}$ 001: $f_{PCLK2}= f_{HCLK} / 2$ 010: $f_{PCLK2}= f_{HCLK} / 4$ 011: $f_{PCLK2}= f_{HCLK} / 8$ 100: $f_{PCLK2}= f_{HCLK} / 16$ 101: $f_{PCLK2}= f_{HCLK} / 32$ 110: $f_{PCLK2}= f_{HCLK} / 64$ 111: $f_{PCLK2}= f_{HCLK} / 128$
6	ADCEN	ADC Clock Enable Bit 0: Disable 1: Enable
3	QEP1EN	QEP1 Clock Enable Bit 0: Disable 1: Enable
2	QEP0EN	QEP0 Clock Enable Bit 0: Disable 1: Enable
31~24 19~7	-	Reserved

Bit number	Bit Mnemonic	Description
5~4 1~0		

4.7.1.10 AHB Bus Peripheral Reset Control Register (AHB_RST)

Register	R/W	Description	Reset Value	POR
AHB_RST	R/W	AHB Bus Peripheral Reset Control Register	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
4	CANRST	CAN Reset Control Bit This bit is set to 1 by software, and is automatically cleared by hardware. 0: None effect 1: Reset CAN
1	CRCRST	CRC Reset Control Bit This bit is set to 1 by software, and is automatically cleared by hardware. 0: None effect 1: Reset RCC
0	DMARST	DMA Reset Control Bit This bit is set to 1 by software, and is automatically cleared by hardware. 0: None effect 1: Reset DMA
31~5 3~2	-	Reserved

4.7.1.11 APB0 Bus Peripheral Reset Control Register (APB0_RST)

Register	R/W	Description	Reset Value	POR
APB0_RST	R/W	APB0 Bus Peripheral Reset Control Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24

-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
UART1RST	UART0RST	SPI0RST	TWI0RST	-	-	TIM1RST	TIM0RST

Bit number	Bit Mnemonic	Description
7	UART1RST	UART1 Reset Control Bit This bit is set to 1 by software, and is automatically cleared by hardware. 0: None effect 1: Reset UART1
6	UART0RST	UART0 Reset Control Bit This bit is set to 1 by software, and is automatically cleared by hardware. 0: None effect 1: Reset UART0
5	SPI0RST	SPI0 Reset Control Bit This bit is set to 1 by software, and is automatically cleared by hardware. 0: None effect 1: Reset SPI0
4	TWI0RST	TWI0 Reset Control Bit This bit is set to 1 by software, and is automatically cleared by hardware. 0: None effect 1: Reset TWI0
1	TIM1RST	Timer1 Reset Control Bit This bit is set to 1 by software, and is automatically cleared by hardware. 0: None effect 1: Reset Timer1
0	TIM0RST	Timer0 Reset Control Bit This bit is set to 1 by software, and is automatically cleared by hardware. 0: None effect 1: Reset Timer0
31~8 3~2	-	Reserved

4.7.1.12 APB1 Bus Peripheral Reset Control Register (APB1_RST)

Register	R/W	Description	Reset Value	POR
APB1_RST	R/W	APB1 Bus Peripheral Reset Control Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
UART2RST	-	-	SPI_TWIRST	-	-	TIM3RST	TIM2RST

Bit number	Bit Mnemonic	Description
7	UART2RST	UART2 Reset Control Bit This bit is set to 1 by software, and is automatically cleared by hardware. 0: None effect 1: Reset UART2
4	SPI_TWIRST	SPI_TWI Reset Control Bit This bit is set to 1 by software, and is automatically cleared by hardware. 0: None effect 1: Reset TWI1
1	TIM3RST	Timer3 Reset Control Bit This bit is set to 1 by software, and is automatically cleared by hardware. 0: None effect 1: Reset Timer3
0	TIM2RST	Timer2 Reset Control Bit This bit is set to 1 by software, and is automatically cleared by hardware. 0: None effect 1: Reset Timer2
31~8 6~5 3~2	-	Reserved

4.7.1.13 APB2 Bus Peripheral Reset Control Register (APB2_RST)

Register	R/W	Description	Reset Value	POR
APB2_RST	R/W	APB2 Bus Peripheral Reset Control Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	ADCRST	-	-	QEP1RST	QEP0RST	-	-

Bit number	Bit Mnemonic	Description
6	ADCRST	ADC Reset Control Bit This bit is set to 1 by software, and is automatically cleared by hardware. 0: None effect 1: Reset ADC
3	QEP1RST	QEP1 Reset Control Bit This bit is set to 1 by software, and is automatically cleared by hardware. 0: None effect 1: Reset QEP1
2	QEP0RST	QEP0 Reset Control Bit This bit is set to 1 by software, and is automatically cleared by hardware. 0: None effect 1: Reset QEP0
31~7 5~4 1~0	-	Reserved

4.7.1.14 NMI Interrupt Configuration Register (NMI_CFG)

Register	R/W	Description	Reset Value	POR
NMI_CFG	R/W	Non-Maskable Interrupt(NMI) Interrupt Configuration Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
KEY[15:8]							

23	22	21	20	19	18	17	16
KEY[7:0]							
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
CMP0EN	CMP3EN	OP1EN	OP2EN	SRAMPEEN	INT0EN	-	-

Bit number	Bit Mnemonic	Description
31~16	KEY[15:0]	NMI_CFG Register Write Protection Switch Writing 0xA05F to KEY[15:0] is required to unlock the lower bits of the current register for modification
7	CMP0EN	CMP0 NMI Enable Bit 0: CMP0 NMI disable 1: CMP0 NMI enable When enabled, CMP0IF flag set will trigger NMI, and the NMI interrupt can only be exited after manually clearing the CMP0IF flag. **Note: If CMP0 interrupt is enable (CMPX_IDE->INTEN=1；CMPX_IDE->CMP0IE=1), NMI will still be given priority.**
6	CMP3EN	CMP3 NMI Enable Bit 0: CMP3 NMI disable 1: CMP3 NMI enable When enabled, CMP3IF flag set will trigger NMI, and the NMI interrupt can only be exited after manually clearing the CMP3IF flag. **Note: If CMP3 interrupt is enable (CMP3_IDE->INTEN=1), NMI will still be given priority.**
5	OP1EN	OP1_CMP NMI Enable Bit 0: OP1_CMP NMI disable 1: OP1_CMP NMI enable When enabled, OP1IF flag set will trigger NMI, and the NMI interrupt can only be exited after manually clearing the OP1IF flag. **Note: If OP1 interrupt is enable (OPX_IDE->INTEN=1；OPX_IDE->OP1IE=1), NMI will still be given priority.**
4	OP2EN	OP2_CMP NMI Enable Bit 0: OP2_CMP NMI disable 1: OP2_CMP NMI enable When enabled, OP2IF flag set will trigger NMI, and the NMI interrupt can only be exited after manually clearing the OP2IF flag. **Note: If OP2 interrupt is enable (OPX_IDE->INTEN=1；OPX_IDE->OP2IE=1), NMI will still be given priority.**
3	SRAMPEEN	SRAM Parity Checking Error Interrupt Enable Bit 0: NMI will not be generated by SRAM parity checking error 1: NMI will be generated by SRAM parity checking error

Bit number	Bit Mnemonic	Description
		When enabled, detect SRAM parity error while reading SRAM will trigger NMI. The corresponding flag must be manually cleared to exit the NMI interrupt.
2	INT0EN	External Interrupt INT0 NMI Enable Bit 0: INT0 NMI disable 1: INT0 NMI enable When enabled, both rising and falling edge interrupts on the INT0 pin will trigger NMI. The corresponding flag must be manually cleared to exit the NMI interrupt. **Note: If INT0 interrupt is enabled, NMI will still be given priority.**
15~8 1~0	-	Reserved

4.7.2 RCC Register Mapping

Register	Offset Address	R/W	Description	Reset Value	POR
AHB Base Address:0x4000_3000					
AHB_CFG	0x00	R/W	AHB Bus Peripheral Clock Enable Register	0x0010_0000	0x0010_0000
AHB_RST	0x04	R/W	AHB Bus Peripheral Reset Control Register	0x0000_0000	0x0000_0000
RCC_KEY	0x0C	R/W	RCC Protect Register	0x0000_0000	0x0000_0000
RCC_CFG0	0x14	R/W	System Clock Source Selection Register	0x0000_1040	0x0000_1040
RCC_CFG1	0x18	R/W	Peripheral Clock Source Selection Register	0x0000_0000	0x0000_0000
RCC_STS	0x20	R/W	Clock Status Register	0x0000_0000	0x0000_0000
SYST_CALIB	0x28	R/W	SysTick Calibration Parameter Register	0x0000_2327	0x0000_2327
NMI_CFG	0x2C	R/W	NMI Interrupt Configuration Register	0x0000_0000	0x0000_0000

Register	Offset Address	R/W	Description	Reset Value	POR
APB0 Base Address:0x4002_0000					
APB0_CFG	0x00	R/W	APB0 Bus Peripheral Clock Enable Register	0x0000_0000	0x0000_0000
APB0_RST	0x04	R/W	APB0 Bus Peripheral Reset Control Register	0x0000_0000	0x0000_0000

Register	Offset Address	R/W	Description	Reset Value	POR
APB1 Base Address:0x4002_1000					
APB1_CFG	0x00	R/W	APB1 Bus Peripheral Clock Enable Register	0x0000_0000	0x0000_0000
APB1_RST	0x04	R/W	APB1 Bus Peripheral Reset Control Register	0x0000_0000	0x0000_0000

Register	Offset Address	R/W	Description	Reset Value	POR
APB2 Base Address:0x4002_2000					
APB2_CFG	0x00	R/W	APB2 Bus Peripheral Clock Enable Register	0x0000_0000	0x0000_0000
APB2_RST	0x04	R/W	APB2 Bus Peripheral Reset Control Register	0x0000_0000	0x0000_0000

5 Interrupts

- M0+ core could provide a maximum of 32 interrupt sources, numbered from 0 to 31, while SC32F15G series has 25 interrupt sources.
- Four-level interrupt priorities can be configured, and the interrupt priorities are set through the Interrupt Priority Registers in the core registers.

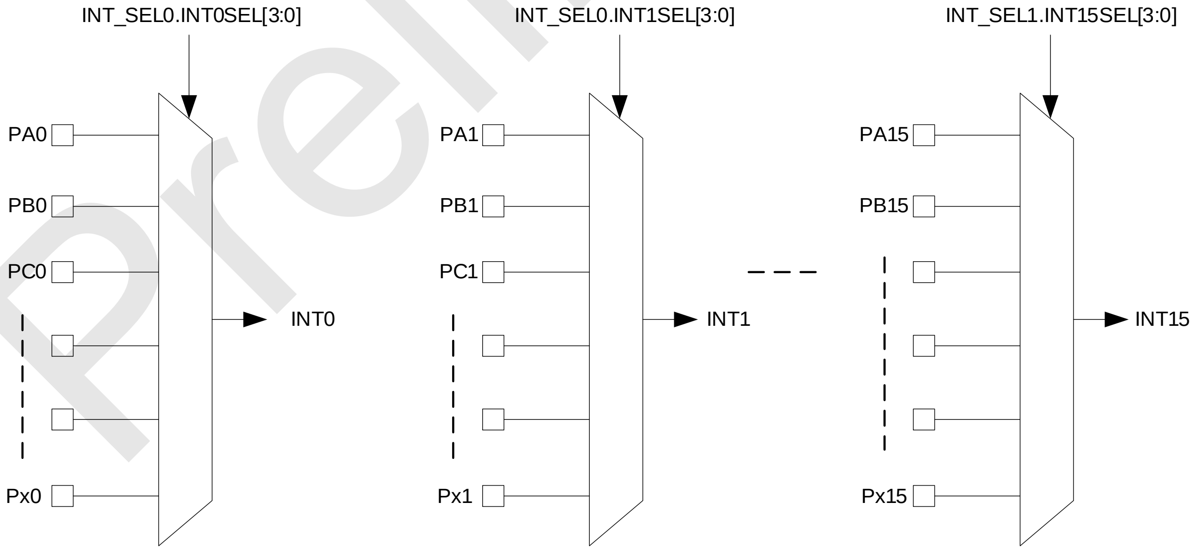
5.1 External interrupts INT0~15

External interrupts comprise 16 interrupt sources, occupying a total of 4 interrupt vectors. All 16 external interrupt sources can be configured to respond to rising edges, falling edges, or both edges. Once configured, these interrupts can cover all GPIO pins. When the corresponding event occurs, software sets the corresponding interrupt flag (RIF/FIF to 1), triggering entry into the corresponding interrupt service.

The external interrupt features of the SC32F15G series are as follows:

- 16 INT interrupt sources, occupying 4 interrupt vectors in total.
- After configuration, INT can cover all GPIO pins.
- All INT sources can be configured for rising edge, falling edge, or both edge interrupts, each having independent corresponding interrupt flag.
- Software sets the corresponding interrupt flag can trigger entry into the corresponding interrupt service.

Note: When using INT functions, users need to manually set the GPIO port corresponding to INTn (n=0~15) to pull-up input mode. External interrupts cannot be detected in output mode.



External Interrupt Port Multiplexer

5.2 Interrupt and Events

- When NVIC is disabled, interrupt request masks are enabled, events can be generated, but interrupt cannot be generated.

- When NVIC is enabled, interrupt request masks act as internal master interrupt control bit in the module.

5.3 Interrupt Source and Vector

Interrupt Vector	Interrupt Number	Priority	Interrupt Vector Address	Interrupt Source	Core/NVIC Enable Bit	Interrupt Request Mask Bit	Interrupt Subroutine Control Bit	Interrupt Flag	Capability of Waking up STOP
0	-	-	0x0000_0000	-		-	\	\	YES
1	-	Fixed	0x0000_0004	RESET	PRIMASK	SCB	\	\	YES
2	-	Fixed	0x0000_0008	NMI_Handler		SCB	\	\	YES
3	-	Fixed	0x0000_000C	HardFault_Handler	PRIMASK	SCB	\	\	YES
4~10	-	-	0x0000_0010 - 0x0000_0028	-		-	\	\	YES
11	-	Settable		SVC_Handler	PRIMASK	SCB	\	\	YES
12~13	-	-	0x0000_0030 0x0000_0034	-		-	\	\	YES
14	-	Settable	0x0000_0038	PendSV_Handler	PRIMASK	SCB	\	\	YES
15	-	Settable	0x0000_003C	SysTick_Handler	PRIMASK	SysTick_CTRL	\	\	YES
16	0	Settable	0x0000_0040	INT0	NVIC->ISER[0].0	INTF_IE->ENFx，x=0 INTR_IE->ENRx	\	INTF_STS->FIFx INTR_STS->RIFx	YES
17	1	Settable	0x0000_0044	INT1-7	NVIC->ISER[0].1	INTF_IE->ENFx，x=1~7 INTR_IE->ENRx	\	INTF_STS->FIFx INTR_STS->RIFx	YES
18	2	Settable	0x0000_0048	INT8-11	NVIC->ISER[0].2	INTF_IE->ENFx，x=8~11 INTR_IE->ENRx	\	INTF_STS->FIFx INTR_STS->RIFx	YES
19	3	Settable	0x0000_004C	INT12-15	NVIC->ISER[0].3	INTF_IE->ENFx，x=12~15 INTR_IE->ENRx	\	INTF_STS->FIFx INTR_STS->RIFx	YES
20	4	Reserved	0x0000_0050	\	NVIC->ISER[0].4	\	\	\	
21	5	Reserved	0x0000_0054	\	NVIC->ISER[0].5	\	\	\	
22	6	Settable	0x0000_0058	BTM	NVIC->ISER[0].6	BTM_CON->INTEN	\	BTM_STS->BTMIF	YES
23	7	Settable	0x0000_005C	UART0	NVIC->ISER[0].7	UART0_IDE->INTEN	UART0_IDE->TXIE UART0_IDE->RXIE	UART0_STS->TXIF UART0_STS->RXIF	YES
				UART2/LIN		UART2_IDE->INTEN	UART2_IDE->TXIE UART2_IDE->RXIE UART2_IDE->BKIE UART2_IDE->SLVHEIE	UART2_STS->TXIF UART2_STS->RXIF UART2_STS->BKIF UART2_STS->SLVHEIF	NO
24	8	Settable	0x0000_0060	UART1	NVIC->ISER[0].8	UART1_IDE->INTEN	UART1_IDE->TXIE UART1_IDE->RXIE	UART1_STS->TXIF UART1_STS->RXIF	YES
25	9	Settable	0x0000_0064	SPI0	NVIC->ISER[0].9	SPI0_IDE->INTEN	SPI0_IDE->RXNEIE SPI0_IDE->TBIE SPI0_IDE->RXIE	SPI0_STS->SPIF SPI0_STS->RXNEIF SPI0_STS->TXEIF SPI0_STS->RXFIF	NO

Interrupt Vector	Interrupt Number	Priority	Interrupt Vector Address	Interrupt Source	Core/NVIC Enable Bit	Interrupt Request Mask Bit	Interrupt Subroutine Control Bit	Interrupt Flag	Capability of Waking up STOP
							SPI0_IDE->RXHIE SPI0_IDE->TXHIE	SPI0_STS->RXHIF SPI0_STS->TXHIF	
26	**10**	Settable	0x0000_0068	SPI1	NVIC->ISER[0].10	SPI1_TWI1_IDE->INTEN	SPI1_TWI1_IDE->TBIE	SPI1_TWI1_STS->QTWIF SPI1_TWI1_STS->TXEIF	NO
				TWI1				SPI1_TWI1_STS->QTWIF	NO
27	**11**	Settable	0x0000_006C	DMA0	NVIC->ISER[0].11	DMA0_CFG->INTEN	DMA0_CFG->TCIE DMA0_CFG->HTIE DMA0_CFG->TEIE	DMA0_STS->GIF DMA0_STS->TCIF DMA0_STS->HTIF DMA0_STS->TEIF	NO
28	**12**	Settable	0x0000_0070	DMA1	NVIC->ISER[0].12	DMA1_CFG->INTEN	DMA1_CFG->TCIE DMA1_CFG->HTIE DMA1_CFG->TEIE	DMA1_STS->GIF DMA1_STS->TCIF DMA1_STS->HTIF DMA1_STS->TEIF	NO
29	**13**	Settable	0x0000_0074	DMA2	NVIC->ISER[0].13	DMA2_CFG->INTEN	DMA2_CFG->TCIE DMA2_CFG->HTIE DMA2_CFG->TEIE	DMA2_STS->GIF DMA2_STS->TCIF DMA2_STS->HTIF DMA2_STS->TEIF	NO
30	**14**	Settable	0x0000_0078	DMA3	NVIC->ISER[0].14	DMA3_CFG->INTEN	DMA3_CFG->TCIE DMA3_CFG->HTIE DMA3_CFG->TEIE	DMA3_STS->GIF DMA3_STS->TCIF DMA3_STS->HTIF DMA3_STS->TEIF	NO
31	**15**	Settable	0x0000_007C	TIM0	NVIC->ISER[0].15	TIM0_IDE->INTEN	TIM0_IDE->TIE TIM0_IDE->EXFIE TIM0_IDE->EXRIE	TIM0_STS->TIF TIM0_STS->EXIF TIM0_STS->EXIR	NO
32	**16**	Settable	0x0000_0080	TIM1	NVIC->ISER[0].16	TIM1_IDE->INTEN	TIM1_IDE->TIE TIM1_IDE->EXFIE TIM1_IDE->EXRIE	TIM1_STS->TIF TIM1_STS->EXIF TIM1_STS->EXIR	NO
33	**17**	Settable	0x0000_0084	TIM2	NVIC->ISER[0].17	TIM2_IDE->INTEN	TIM2_IDE->TIE TIM2_IDE->EXFIE TIM2_IDE->EXRIE	TIM2_STS->TIF TIM2_STS->EXIF TIM2_STS->EXIR	NO
34	**18**	Settable	0x0000_0088	TIM3	NVIC->ISER[0].18	TIM3_IDE->INTEN	TIM3_IDE->TIE TIM3_IDE->EXFIE TIM3_IDE->EXRIE	TIM3_STS->TIF TIM3_STS->EXIF TIM3_STS->EXIR	NO
35	**19**	Reserved	0x0000_008C	\	NVIC->ISER[0].19	\	\	\	
36	**20**	Settable	0x0000_0090	QEP0	NVIC->ISER[0].20	QEP0_IDE->INTEN	QEP0_IDE->PCUIE QEP0_IDE->PCOIE QEP0_IDE->IERIE	QEP0_STS->PCUIF QEP0_STS->PCOIF QEP0_STS->IERIF	NO

Interrupt Vector	Interrupt Number	Priority	Interrupt Vector Address	Interrupt Source	Core/NVIC Enable Bit	Interrupt Request Mask Bit	Interrupt Subroutine Control Bit	Interrupt Flag	Capability of Waking up STOP
							QEP0_IDE->UPEVNTIE	QEP0_STS->UPEVNTIF	
37	**21**	Settable	0x0000_0094	\	NVIC->ISER[0].21	\	\	\	
38	**22**	Settable	0x0000_0098	OP1_CMP	NVIC->ISER[0].22	OP_IDE>INTEN	OP_IDE->OP_CMP1IE	OP_STS->OP_CMP1IF	NO
				OP2_CMP			OP_IDE->OP_CMP2IE	OP_STS->OP_CMP2IF	NO
39	**23**	Settable	0x0000_009C	TWI0	NVIC->ISER[0].23	TWI0_IDE->INTEN	\	TWI0_STS->TWIF	NO
40	**24**	Settable	0x0000_00A0	QEP1	NVIC->ISER[0].24	QEP1_IDE->INTEN	QEP1_IDE->PCUIE QEP1_IDE->PCOIE QEP1_IDE->IERIE QEP1_IDE->UPEVNTIE	QEP1_STS->PCUIF QEP1_STS->PCOIF QEP1_STS->IERIF QEP1_STS->UPEVNTIF	NO
41	**25**	Reserved	0x0000_00A4	\	\	\	\	\	
42	**26**	Reserved	0x0000_00A8	\	\	\	\	\	
43	**27**	Reserved	0x0000_00AC	\	\	\	\	\	
44	**28**	Settable	0x0000_00B0	**CAN**	NVIC->ISER[0].28	CAN_IDE->INTEN	CAN_RTIE->RIE CAN_RTIE->ROIE CAN_RTIE->RFIE CAN_RTIE->RAFIE CAN_RTIE->TPIE CAN_RTIE->TSIE CAN_RTIE->EIE CAN_RTIE->EPIE CAN_RTIE->ALIE CAN_RTIE->BEIE	CAN_RTIE->RIF CAN_RTIE->ROIF CAN_RTIE->RFIF CAN_RTIE->RAFIF CAN_RTIE->TPIF CAN_RTIE->TSIF CAN_RTIE->EIF CAN_RTIE->EPIF CAN_RTIE->ALIF CAN_RTIE->BEIF	NO
45	**29**	Settable	0x0000_00B4	ADC	NVIC->ISER[0].29	ADC_CON->INTEN	ADC_IDE->EOCIE ADC_IDE->EOSIE0 ADC_IDE->EOSIE1 ADC_IDE->EOSIE2 ADC_IDE->EOSIE3 ADC_IDE->UPTHIE ADC_IDE->DOWTHIE	ADC_STS->ADCIF ADC_STS->EOSIF0 ADC_STS->EOSIF1 ADC_STS->EOSIF2 ADC_STS->EOSIF3 ADC_STS->UPTHIF ADC_STS->DOWTHIF	NO
46	**30**	Settable	0x0000_00B8	CMP0	NVIC->ISER[0].30	CMPX_IDE->INTEN	CMPX_IDE->CMP0IE	CMPX_STS->CMP0IF	YES
				CMP1			CMPX_IDE->CMP1IE	CMPX_STS->CMP1IF	

Interrupt Vector	Interrupt Number	Priority	Interrupt Vector Address	Interrupt Source	Core/NVIC Enable Bit	Interrupt Request Mask Bit	Interrupt Subroutine Control Bit	Interrupt Flag	Capability of Waking up STOP
				CMP2			CMPX_IDE->CMP2IE	CMPX_STS->CMP2IF	
47	31	Settable	0x0000_00BC	CMP3	NVIC->ISER[0].31	CMP3_IDE->INTEN	\	CMP3_STS->CMP3IF	YES

5.4 External Interrupt Register

5.4.1 External Interrupt Related Register

5.4.1.1 External Interrupt Falling Edge Interrupt Enable Register (INTF_IE)

Register	R/W	Description	Reset Value	POR
INTF_IE	R/W	INT Falling Edge Enable Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
ENF15	ENF14	ENF13	ENF12	ENF11	ENF10	ENF9	ENF8
7	6	5	4	3	2	1	0
ENF7	ENF6	ENF5	ENF4	ENF3	ENF2	ENF1	ENF0

Bit number	Bit Mnemonic	Description
15~0	ENFx (x=0~15)	INTx Falling Edge Enable Control Bit(x=0~15) 0: Disable 1: Enable
31~16	-	Reserved

5.4.1.2 External Interrupt Rising Edge Interrupt Enable Register (INTR_IE)

Register	R/W	Description	Reset Value	POR
INTR_IE	R/W	INT Rising Edge Enable Register	0x0000_0000	0x0000_0000

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

-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
ENR15	ENR14	ENR13	ENR12	ENR11	ENR10	ENR9	ENR8
7	6	5	4	3	2	1	0
ENR7	ENR6	ENR5	ENR4	ENR3	ENR2	ENR1	ENR0

Bit number	Bit Mnemonic	Description
15~0	ENRx (x=0~15)	INTx Rising Edge Enable Control Bit(x=0~15) 0: Disable 1: Enable
31~16	-	Reserved

5.4.1.3 External Interrupt Port Selection Register0 (INT_SEL0)

Register	R/W	Description	Reset Value	POR
INT_SEL0	R/W	External Interrupt Port Selection Register0	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
INT7SEL[3:0]				INT6SEL[3:0]			
23	22	21	20	19	18	17	16
INT5SEL[3:0]				INT4SEL[3:0]			
15	14	13	12	11	10	9	8
INT3SEL[3:0]				INT2SEL[3:0]			
7	6	5	4	3	2	1	0
INT1SEL[3:0]				INT0SEL[3:0]			

Bit number	Bit Mnemonic	Description
31~0	INTxSEL[3:0] (x=0~7)	External Interrupt INTx Port Selection Bit(x=0~7) 0000: Select PAx Port 0001: Select PBx Port 0010: Select PCx Port Others: Reserved Note: The same external interrupt port can only be selected for one GPIO at a time.

5.4.1.4 External Interrupt Port Selection Register1 (INT_SEL1)

Register	R/W	Description	Reset Value	POR
INT_SEL1	R/W	External Interrupt Port Selection Register1	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
INT15SEL[3:0]				INT14SEL[3:0]			

23	22	21	20	19	18	17	16
INT13SEL[3:0]				INT12SEL[3:0]			
15	14	13	12	11	10	9	8
INT11SEL[3:0]				INT10SEL[3:0]			
7	6	5	4	3	2	1	0
INT9SEL[3:0]				INT8SEL[3:0]			

Bit number	Bit Mnemonic	Description
31~0	INTxSEL[3:0] (x=8~15)	External Interrupt INTx Port Selection Bit(x=8~15) 0000: Select PAx Port 0001: Select PBx Port 0010: Select PCx Port Others: Reserved Note: The same external interrupt port can only be selected for one GPIO at a time.

5.4.1.5 External Interrupt Falling Edge Control Register (INTF_CON)

Register	R/W	Description	Reset Value	POR
INTF_CON	R/W	External Interrupt Falling Edge Control Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
FT15	FT14	FT13	FT12	FT11	FT10	FT9	FT8
7	6	5	4	3	2	1	0
FT7	FT6	FT5	FT4	FT3	FT2	FT1	FT0

Bit number	Bit Mnemonic	Description
15~0	FTx (x=0~15)	INTx Falling Edge Detection Enable Bit(x=0~15) 0: Disable 1: Enable
31~16	-	Reserved

5.4.1.6 External Interrupt Rising Edge Control Register (INTR_CON)

Register	R/W	Description	Reset Value	POR
INTR_CON	R/W	External Interrupt Rising Edge Control Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
RT15	RT14	RT13	RT12	RT11	RT10	RT9	RT8
7	6	5	4	3	2	1	0
RT7	RT6	RT5	RT4	RT3	RT2	RT1	RT0

Bit number	Bit Mnemonic	Description
15~0	RTx (x=0~15)	INTx Rising Edge Detection Enable Bit(x=0~15) 0: Disable 1: Enable
31~16	-	Reserved

5.4.1.7 External Interrupt Falling Edge Flag Register (INTF_STS)

Register	R/W	Description	Reset Value	POR
INTF_STS	R/W	External Interrupt Falling Edge Flag Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
FIF15	FIF14	FIF13	FIF12	FIF11	FIF10	FIF9	FIF8
7	6	5	4	3	2	1	0
FIF7	FIF6	FIF5	FIF4	FIF3	FIF2	FIF1	FIF0

Bit number	Bit Mnemonic	Description
15~0	FIFx (x=0~15)	INTx Falling Edge Capture Flag(x=0~15) This bit will be set to 1 by hardware when a falling edge is detected, and can be cleared by software. It is possible to trigger a falling edge capture interrupt by setting this bit to 1 by software.
31~16	-	Reserved

5.4.1.8 External Interrupt Rising Edge Flag Register (INTR_STS)

Register	R/W	Description	Reset Value	POR
INTR_STS	R/W	External Interrupt Rising Edge Flag Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
RIF15	RIF14	RIF13	RIF12	RIF11	RIF10	RIF9	RIF8
7	6	5	4	3	2	1	0
RIF7	RIF6	RIF5	RIF4	RIF3	RIF2	RIF1	RIF0

Bit number	Bit Mnemonic	Description
15~0	RIFx (x=0~15)	INTx Rising Edge Capture Flag(x=0~15) This bit will be set to 1 by hardware when a rising edge is detected, and can be cleared by software. It is possible to trigger a rising edge capture interrupt by setting this bit to 1 by software.
31~16	-	Reserved

5.4.2 External Interrupt Register Mapping

Register	Offset Address	R/W	Description	Reset Value	POR
External Interrupt Base Address:0x4001_1800					
INTF_IE	0x00	R/W	External Interrupt Falling Edge Interrupt Enable Register	0x0000_0000	0x0000_0000
INTR_IE	0x20	R/W	External Interrupt Rising Edge Interrupt Enable Register	0x0000_0000	0x0000_0000
INT_SEL0	0x40	R/W	External Interrupt Port Selection Register0	0x0000_0000	0x0000_0000
INT_SEL1	0x60	R/W	External Interrupt Port Selection Register1	0x0000_0000	0x0000_0000
INTF_CON	0x80	R/W	External Interrupt Falling Edge Control Register	0x0000_0000	0x0000_0000
INTR_CON	0xA0	R/W	External Interrupt Rising Edge Control Register	0x0000_0000	0x0000_0000
INTF_STS	0xC0	R/W	External Interrupt Falling Edge Flag Register	0x0000_0000	0x0000_0000

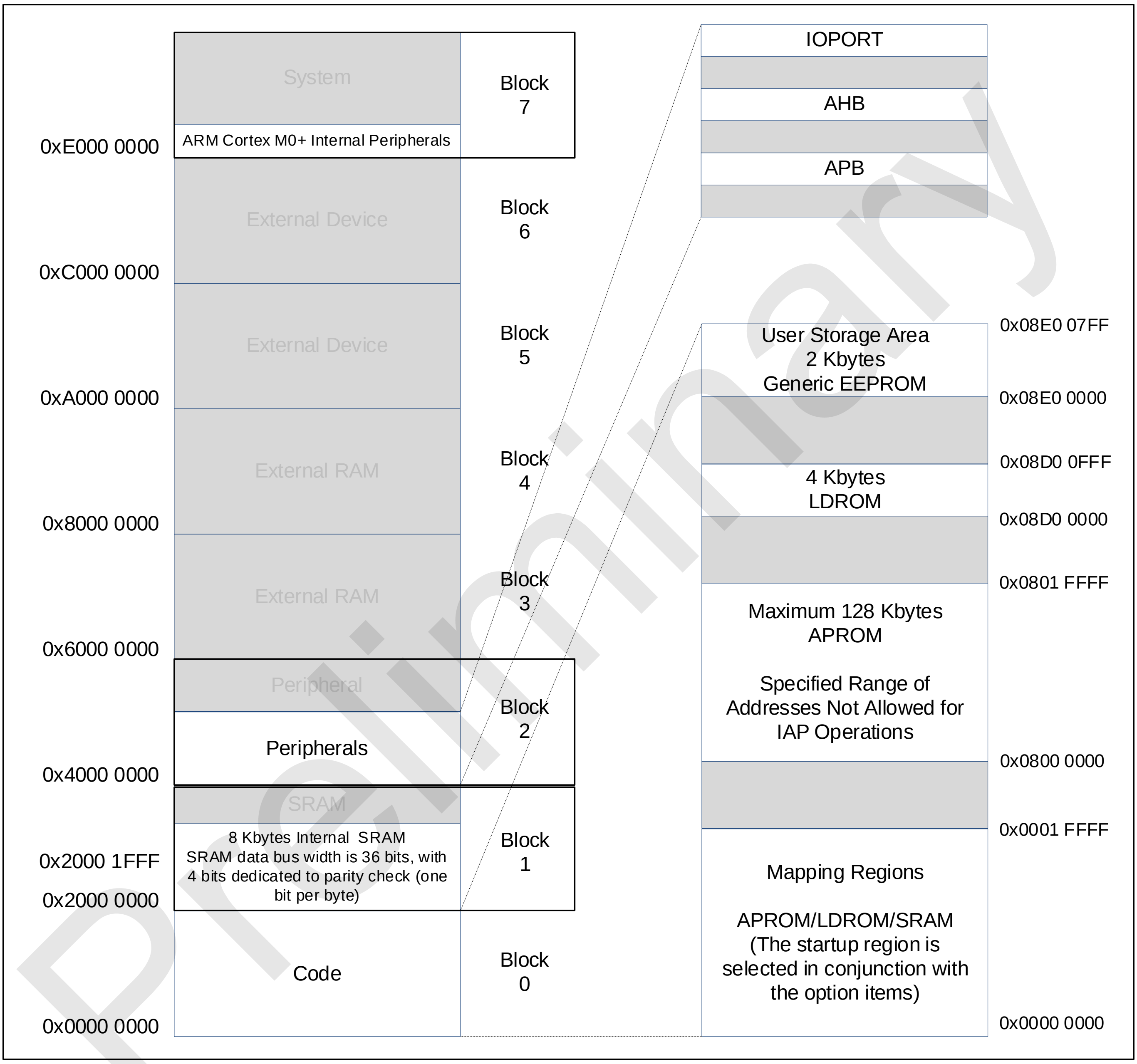
Register	Offset Address	R/W	Description	Reset Value	POR
INTR_STS	0xE0	R/W	External Interrupt Rising Edge Flag Register	0x0000_0000	0x0000_0000

6 Flash

6.1 Description

The program memory, data memory, and registers are arranged in the same linear 4 GB address space. Each byte is encoded in little-endian format in memory. The byte with the lowest index within a word is considered the least significant byte, while the byte with the highest index is considered the most significant byte. The addressable memory space is divided into 8 main blocks, with each block being 512 MB.

6.2 Storage Block Diagram



SC32F15G Series Memory Mapping Diagram

6.3 Feature

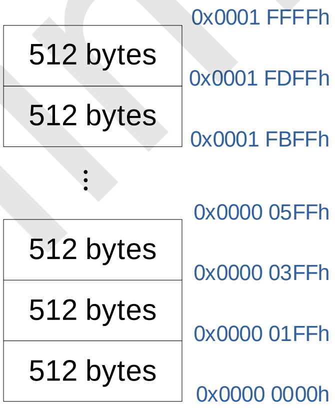
- The Flash width is 32 bits, and it can be rewritten up to 100,000 times
- Data retention time is over 100 years at room temperature
- The structure of the Flash includes:
  - Maximum 128 Kbytes APROM
  - 4 Kbytes LDROM

- 2 Kbytes user storage area (generic EEPROM)
- 8 Kbytes Internal SRAM, support parity check
- 96 bits Unique ID

6.4 APROM

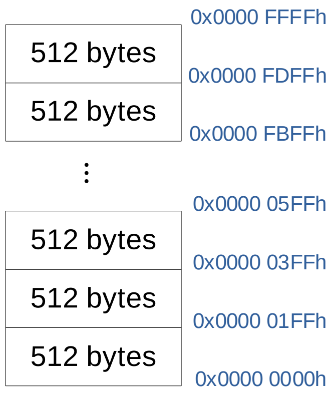
- Maximum 128 Kbytes APROM
- Sector Size: 512 bytes
- Supports: Read/Write/Sector Erase/Chip Erase/Blank Check
- The CPU (Cortex®-M0+) accesses Flash through the AHB bus
- The program defaults to booting from APROM, and users can select programs to boot from other areas such as SRAM/LDROM using the customer option OP_BL[1:0].
- Read Protection: After enabling read protection, only a program that runs from APROM can read information from APROM. Other areas or third-party tools cannot access information from APROM.
- Write Protection: Provides two hardware write protection regions where IAP operations are prohibited. Users can set the range of the two write protection regions in units of sectors based on actual needs.

128 Kbytes of APROM is divided into 256 sectors, with each sector being 512 bytes. During programming, the sector corresponding to the target address is forcibly erased by the programmer before writing data. For user write operations, erasure must precede data writing.



SC32F15G series 128 Kbytes APROM Sector Partition Illustration

64 Kbytes of APROM is divided into 128 sectors, with each sector being 512 bytes. During programming, the sector corresponding to the target address is forcibly erased by the programmer before writing data. For user write operations, erasure must precede data writing.

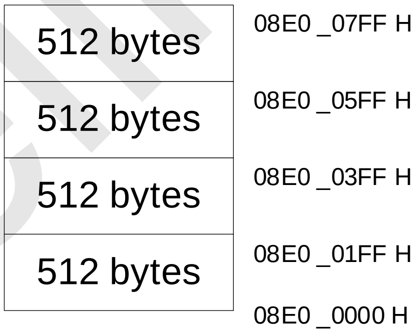


SC32F15G series 64 Kbytes APROM Sector Partition Illustration

6.5 2 Kbytes User Storage Area(Genetic EEPROM)

The 2 Kbytes of independent EEPROM area is addressed from 0x08E0_0000 H to 0x08E0_07FF H, as set by the IAPADE register. This independent EEPROM can be written to repeatedly up to 100,000 times, and it is designed to retain data for over 100 years at room temperature. The independent EEPROM supports various operations including blank check, programming, verification, erasure, and reading functions.

EEPROM has 4 sectors, with each sector being 512 bytes.



EEPROM Sector Partition Illustration

Note: The EEPROM has a write cycle endurance of 100,000 times. Users should avoid exceeding the rated write cycles of the EEPROM to prevent any anomalies!

6.6 4 Kbtyes LDROM

- 4 Kbytes of system storage area, factory-programmed with BootLoader program, Users cannot modify or access this area.
- Embedded Bootloader Program: The fixed ISP program is publicly available, allowing reprogramming of Flash via UART. The program waits for upgrade commands, and if no update command is received within 500 milliseconds, it jumps to APROM for execution (0X0800 0000).

6.6.1 BootLoader

Supports two Bootloader modes:

- Software Approach: Directly partition BootLoader and APP areas in software. Easy sharing interrupts of BootLoader and APP by modifying VTOR. Flexible adjustment of the size of each area；
- Hardware Approach: 4 Kbytes fixed "LDROM" as a dedicated BootLoader area that users cannot read or write
  - LDROM serves as a fixed BootLoader space with factory-programmed program, and users cannot read or write
  - Embedded Bootloader Program: The embedded bootloader program resides in LDROM and is programmed during the production stage. The fixed ISP program is publicly available, allowing reprogramming of Flash via UART.

6.7 SRAM

- Internal SRAM：8 Kbytes，address 0x2000 0000 ~ 0x2000 1FFF
- Supports parity check
  - An additional 1K RAM is used for parity checking, which means SRAM data bus width is 36 bits, with 4 bits dedicated to parity check (one bit per byte).
  - The parity check bits are calculated and saved when writing to the SRAM, and automatically verified upon reading. If a bit fails, an unmaskable interrupt (Cortex®-M0+ NMI) will be generated.
  - Provides an independent SRAM parity error flag, SRAMPEIF.

Note: When SRAM parity check is enabled, it is recommended to perform a software initialization of the entire SRAM at the beginning of the code to prevent parity check errors when reading from uninitialized locations.

- Users can choose to start the program from SRAM by configuring the customer option OP_BL[1:0].
- It supports byte, half-word (16-bit), or word (32-bit) access at the maximum system clock frequency, with no waiting states. Therefore, it can be accessed by both the CPU and DMA

6.8 Boot Area Selection

After a reset, users can independently configure the desired boot mode.

After exiting the standby mode, the startup mode configuration can be resampled. Once this startup delay has ended, the CPU will fetch the stack top value from address 0x0000_0000 and then begin executing code from the boot memory starting at 0x0000_0004.

There are three options for boot area selection: Main Flash Memory Area, System Flash Memory Area and SRAM, described in detail as follows:

6.8.1 Boot from APROM

APROM is aliased in the boot memory space (0x0000_0000) but can also be accessed from its original memory space (0x0800_0000). In other words, the program can start accessing from either address 0x0000_0000 or 0x0800_0000.

6.8.2 Boot from LDROM

- 4 Kbytes LDROM serves as a fixed BootLoader space with factory-programmed program, Users cannot modify or access this area.
- Embedded Bootloader Program: The embedded bootloader program resides in LDROM and is programmed during the production stage. The fixed ISP program is publicly available, allowing reprogramming of Flash via UART.

6.8.3 Boot from SRAM

SRAM has an alias in the boot memory space (0x0000_0000) but can also be accessed from its original memory space (0x2000_0000).

6.8.4 Boot mode config

The boot modes can be controlled by the register bits BTLD[1:0] in conjunction with the software reset (RST) control bit, both protected by the IAP_KEY:

① Set BTLD[1:0]=0x00: the chip boots from APROM after a software reset
② Set BTLD[1:0]=0x01: the chip boots from LDROM after a software reset
③ Set BTLD[1:0]=0x10: the chip boots from SRAM after a software reset

The initial boot region selection during power-up can be configured by customer option bits OP_BL[1:0]:

① Set OP_BL[1:0]=0x00 in customer option: the chip boots from APROM after a software reset
② Set OP_BL[1:0]=0x01 in customer option: the chip boots from LDROM after a software reset
③ Set OP_BL[1:0]=0x10 in customer option: the chip boots from SRAM after a software reset

6.9 96 bits Unique ID

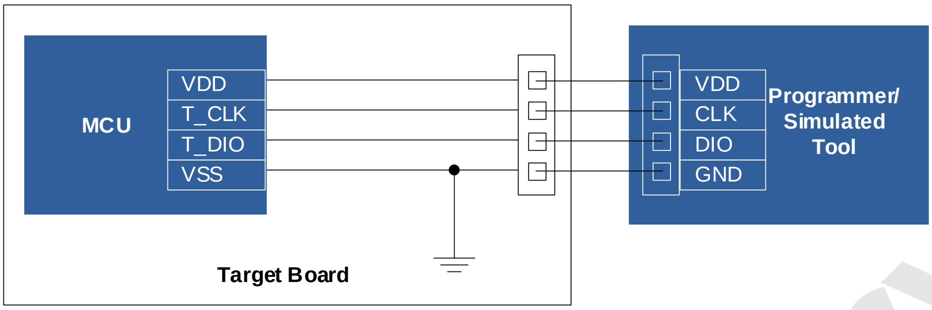
The SC32F15G provides an independent Unique ID area. A 96-bit unique code can be pre-programmed before leaving the factory to ensure the uniqueness of the chip. The only way for the user to obtain the serial number is to read through the IAP instruction.

6.10 User ID Area

User ID area, where user-costomized ID is pre-programmed when leaving the factory. Users can read the User ID area, but cannot write the User ID area.

6.11 Programming

The SC32F15G's Flash can be programmed through T_DIO, T_CLK, VDD, VSS, the specific connection relationship is as follows:



ICP mode Flash Writer programming connection diagram

T_DIO、T_CLK is a 2-wire JTAG programming and emulation signal line. Users can configure the mode of these two ports through the Customer Option when programming.

6.11.1 JTAG Specific Mode

T_DIO,T_CLK are specific port for programming and emulation, and other functions multiplexed with it are not available. This mode is generally used in the online debugging stage, which is convenient for users to simulate and debug. After the JTAG special mode takes effect, the chip can directly enter the programming or emulation mode without powering on and off again.

6.11.2 Normal Mode (JTAG specific port is invalid)

The JTAG function is not available, and other functions multiplexed with it can be used normally. This mode can prevent the programming port from occupying the MCU pins, which is convenient for users to maximize the use of MCU resources.

Note: When the invalid configuration setting of the JTAG dedicated port is successful, the chip must be completely powered off and then on again to enter the programming or emulation mode, which will affect the programming and emulation in the live mode. SinOne recommends that users select the invalid configuration of the JTAG dedicated port during mass production and programming, and select the JTAG mode during the development and debugging phase.

Related Customer Option is as followed:

Register	R/W	Description	Reset Value
COPT1_CFG@0xC2	R/W	Customer Option Mapping Register 1	0x0000_0000

7	6	5	4	3	2	1	0
ENWDT	DISJTG	DISRST	-	-	-	OP_BL[1:0]	

Bit number	Bit Mnemonic	Description
6	DISJTG	JTAG Ports Switch Control Bit 0: JTAG mode enabled, the corresponding pins can only be used as T_CLK and T_DIO

Bit number	Bit Mnemonic	Description
		1: Normal mode, JTAG function disabled

6.12 Security Encryption

The SC32F15G series mainly involves encrypting the APROM for read protection. Users can configure the read protection encryption feature during programming through the customer option in the dedicated programming host; enable flash read protection can enter encryption mode:

- The chip defaults to a non-encrypted state while leaving the factory
- The read protection encryption feature has no mapped registers. Users can only modify it after config the customer option in the dedicated programming host and programming.
- Encryption Disabled: Operations such as reading, programming, and erasing can be performed on APROM. These operations can be also performed on Bytes and backup registers.
- Encryption Enabled:
  - Enable from APROM: Code executed in user mode (booting from user APROM) can perform all operations on APROM.
  - Debug, enable from SRAM and LDROM: In debug mode or when code is booted from SRAM or LDROM, APROM is completely inaccessible.
- Disabling encryption requires a full erase operation on APROM.

6.12.1 Security Encryption Access Rights

Boot Area/Tools	Encryption Disabled Status					Read Protection Encryption Status				
	Read	Write	Block Earse	Full Earse	Operate Write-Protection Region	Read	Write	Block Earse	Full Earse	Operate Write-Protection Region
Boot from APROM	√	√	√	\	Forbid	√	√	√	\	Forbid
Debug/Boot from SRAM	√	√	√	√	Forbid	Forbid	Forbid	Forbid	Forbid	Forbid
Boot from LDROM	√	√	√	√	√	Forbid	Forbid	Forbid	√	Forbid

6.13 In Application Programming (IAP)

The IAP (In Application Programming) area in the APROM of SC32F15G allows users to perform remote program updates through IAP operations. Users can also retrieve information from the Unique ID or User ID areas by IAP read operations. Before performing IAP write operations, users must carry out sector erasure for the target address sector.

The chip allows global IAP operations in the APROM by default while leaving the factory. Internally, the chip provides two sets of flash write protection regions. These regions are set based on sector units, and the protected areas are restricted from IAP operations. The rules for setting these regions are as follows:

IAPPORx Register Value(x=A or B)	IAPPOR Protection Area
IAPPORx_ST = IAPPORx_ED	Sector IAPPORx
IAPPORx_ST > IAPPORx_ED	No protection
IAPPORx_ST < IAPPORx_ED	Sectors from IAPPORx_ST to IAPPORx_ED

User can config these APROM's write protection area through "Customer Option" while programming.

6.13.1 IAP Control Register

To perform IAP operations on APROM outside the write protection regions, it can be achieved using the following registers:

6.13.1.1 Data Protect Register (IAP_KEY)

Register	R/W	Description	Reset Value	POR
IAP_KEY	R/W	Data Protect Register	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
31~0	IAPKEY[31:0]	Data Protection Key To prevent accidental operations on Flash due to electrical interference, IAP_CON Register requires unlocking through IAPKEY before performing a write operation. The unlocking sequence is as follows: 1. Write KEY1 = 0x1234_5678 2. Write KEY2 = 0xA05F_05FA The IAP_CON Register will be locked until the next system reset if the sequence of operations is incorrect.

6.13.1.2 IAP Sector Number Setting Register (IAP_SNB)

Register	R/W	Description	Reset Value	POR
IAP_SNB	R/W	IAP Sector Number Setting Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
IAPADE[7:0]							

23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	IAPSNB[8]
7	6	5	4	3	2	1	0
IAPSNB[7:0]							

Bit number	Bit Mnemonic	Description
31~24	IAPADE[7:0]	IAP Operation Area Extended Address By writing different values to IAPADE, the IAP operations can be directed to different operation areas: 0x00: Invalid 0x4C: APROM 0x69: EEPROM 0xF1: customer option Others: Reserved
8~0	IAPSNB[8:0]	IAP Operation Sector Number Setting for Sector/Page Erase: The actual starting address of the operated sector = Flash Base Address + [IAPSNB[8:0] * 0x200]
23~9	-	Reserved

6.13.1.3 IAP Control Register (IAP_CON)(Write Protection)

*This register is write-protected and can only be modified by manipulating the data protection register IAP_KEY.

Register	R/W	Description	Reset Value	POR
IAP_CON	R/W	IAP Control Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
LOCK	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	CONT[5:0]					
15	14	13	12	11	10	9	8
-	-	-	-	DMAEN	BTLD[1:0]		RST
7	6	5	4	3	2	1	0
ERASE	-	SERASE	PRG	-	-	CMD[1:0]	

Bit number	Bit Mnemonic	Description
31	LOCK	The IAP_CON will be locked after setting this bit to 1. When the unlock sequence is detected, this bit will be cleared by hardware, if the unlock operation fails, this bit will remain 1 until the next system reset.

Bit number	Bit Mnemonic	Description
21~16	CONT[5:0]	IAP Write Data Acceleration Setting Bit Perform continuous IAP write operations in units of 4-byte consecutive addresses. Set the continuous programming length via CONT[5:0], with a valid range of 0x01 to 0x20. This allows a maximum of 32 bits * 32 = 128 bytes to be programmed in one continuous operation. Operation Steps: 1. Temporarily store the data to be programmed in the RAM buffer, the RAM buffer has a maximum size of 128 bytes. 2. Unlock IAP_KEY and enable IAP_CON.PRG 3. Enable IAP_CON.DMAEN and set CONT[5:0] 4. Configure DMA: Select an idle DMA channel. Set the source address to the RAM buffer and the target address to the starting address of the programming area. Note: The starting address of the target area must be 4-byte aligned. 5. Configure DMA channel parameters: Set TPTYPE = 1 to select bulk mode, configure TXWIDTH[1:0] = 32 bits to match the data width, and set DMACNT[31:0] = CONT[5:0] to define the number of transfers. 6. Trigger DMA channel for software transfer: Set SWREQ = 1 to initiate the continuous programming operation.
11	DMAEN	DMA-Assisted Continuous Programming Control Bit 0: Disable DMA-assisted continuous programming. 1: Enable DMA-assisted continuous programming.
10~9	BTLD[1:0]	Boot Area Selections Bit After Software Reset: 00: Boot from APROM after software reset 01: Boot from LDROM after software reset 10: Boot from embedded SRAM after software reset 11: Reserved
8	RST	Software Reset Control Bit: 0: Program running normally 1: System will reset immediately after setting this bit to 1
7	ERASE	All Erase Control Bit 0:No earse operation 1:Setting 1 to this bit and configure CMD[1:0]=10 will initiate a full erase operation on APROM.
5	SERASE	Sector Erase Control Bit:0:No earse operation 1: Setting 1 to this bit and configure CMD[1:0]=10 will initiate a sector erase operation on APROM, and the selected sector will be earsed.
4	PRG	Program Control Bit: 0:Disable Flash Programming 1:Enable Flash Programming
1~0	CMD[1:0]	IAP Command Enable Control Bit:

Bit number	Bit Mnemonic	Description
		10: Execute the erase operation command Others: Reserved **Note:** **1.** The corresponding operation will execute only when CMD[1:0] set to 10 after setting any earse control bit to 1. **2.** Only one IAP operation can be executed at a time, so the ERASE/SERASE bit can only be set to 1 at a time
30~22 15~12 6 3~2	-	Reserved

6.13.2 IAP Register Mapping

Register	Offset Address	R/W	Description	Reset Value	POR
IAP Base Address:0x4000_03C0					
IAP_KEY	0x00	R/W	Data protect Register	0x0000_0000	0x0000_0000
IAP_SNB	0x04	R/W	IAP Sector Number Setting Register	0x0000_0000	0x0000_0000
IAP_CON	0x0C	R/W	IAP Control Register	0x0000_0000	0x0000_0000

6.14 Customer Option

SC32F15G has a separate Flash area dedicated to storing customer-defined power-up default settings, this area is called Customer Option area. Users can configure Customer Option through host, and the configured values are written into the Customer Option area during the programming process. IC will use the Customer Option data as the initial settings during the reset initialization phase.

It is also possible to temporarily modify Customer Option by operating mapping registers. However, it's important to note that modifying the mapping registers only achieves temporary adjustments and does not affect the settings in the Customer Option area. The initialization will still be based on the Customer Option parameters during programming after reset.

The operation method of Customer Option related mapping Register is as follows:

The Customer Option related SFR R/W operations are controlled by OPINX and OPREG registers, the specific location of each Customer Option SFR is determined by OPINX, as shown in the following table:

Register	Address	Description	Reset Value	POR
OPINX	0x4000_03F8	Customer Option Pointer	0x0000_0000	0x0000_0000
OPREG	0x4000_03FC	Customer Option Register	0x0000_0000	0x0000_0000

Register	Address	Description	Reset Value	POR
COPT0_CFG	0XC1 @ OPINX	Customer OptionMapping Register0	0x0000_0000	0x0000_0000
COPT1_CFG	0XC2 @ OPINX	Customer OptionMapping Register1	0x0000_0000	0x0000_0000

6.14.1 Customer Option Mapping Register

Before rewrite IFB mapping register by OPINX and OPREG,it is necessary to enable AHB_CFG.IFBEN, the clock enable switch of Customer Option Register.

6.14.1.1 AHB Bus Peripheral Clock Enable Register (AHB_CFG)

Register	R/W	Description	Reset Value	POR
AHB_CFG	R/W	AHB Bus Peripheral Clock Enable Register	0x0010_0000	0x0010_0000

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

Bit number	Bit Mnemonic	Description
2	IFBEN	Customer Option Mapping Register Clock Enable Bit Before rewrite IFB mapping register by OPINX and OPREG,it is necessary to enable IFBEN. 0: Disable 1: Enable
31~23 19~3	-	Reserved

6.14.1.2 Customer Option Mapping Register0 (COPT0_CFG)

Register	R/W	Description	Reset Value	POR
COPT0_CFG	R/W	Customer Option Mapping Register0	0x0000_0000	0x0000_0000

7	6	5	4	3	2	1	0
-	-	-	-	-	DISLVR	LVRS [1:0]	

Bit number	Bit Mnemonic	Description
2	DISLVR	LVR Switch 0:LVR Enable 1:LVR Disable
1~0	LVRS [1:0]	LVR Voltage Select Control 11:4.3V Reset 10:3.7V Reset 01:2.9V Reset 00:1.9V Reset
7~3	-	Reserved

6.14.1.3 Customer Option Mapping Register1 (COPT1_CFG)

Register	R/W	Description	Reset Value	POR
COPT1_CFG	R/W	Customer Option Mapping Register1	0x0000_0000	0x0000_0000

7	6	5	4	3	2	1	0
ENWDT	DISJTG	DISRST	-	-	-	OP_BL[1:0]	

Bit number	Bit Mnemonic	Description
7	ENWDT	WDT Switch 0: WDT disable 1: WDT enable
6	DISJTG	JTAG Switch Control Bit 0: JTAG Mode Enable,corresponding pin can only work as T_CLK/T_DIO 1: Normal Mode Enable,JTAG function disable
5	DISRST	Reset Pin Switch Control Bit **This bit is read only.Read Only** 0: RST corresponding pin is used as reset pin 1: RST corresponding pin is used as normal GPIO pin
1~0	OP_BL[1:0]	Boot area selection after reset **This bit is read onlyRead Only** 00: Boot from APROM after reset 01: Boot from LDROM after reset 10: Boot from embedded SRAM after reset 11: Reserved
4~2	-	Reserved

7 Analog-to-Digital Converter ADC

7.1 Overview

The SC32F15G series features a 12-bit successive approximation type analog-to-digital converter (ADC). It supports up to 18 multiplexed channels and can measure signals from 16 external sources and 2 internal sources(VDD and chip temperature). The A/D conversion for each channel can be performed in single-shot or continuous sampling modes. The results of the ADC are stored in a 32-bit data register.

7.2 Clock source

- The SC32F15G series ADC has only one clock source, which is derived from PCLK
- Typcial conversion time: 404ns

7.3 Feature

- Precision:12 bits
- Maximum Channels: Supports up to 18 channels
  - External 16 ADC sampling channels can be multiplexed with I/O ports for other function
  - Output port of OP0,OP1,OP2 are multiplexed with ADC channels, and the output value can be directly read through ADCV register
  - One internal ADC can directly measure VDD voltage
  - One internal temperature sample channel
- Configurable ADC upper and lower threshold, which can trigger interrupt
- Selectable trigger mode:
  - Manual single trigger
  - Sequential trigger(Only available through software trigger)
- Configurable ADC conversion completion interrupt
- Single-channel conversion time:404ns
- Supports DMA transmission: DMA request will be generated after ADC conversion complete
- The conversion results feature an overflow flag: OVERRUN, and the OVERRUN flag bit is located in the same register as the ADC conversion results so users can read the information all at once.

7.4 ADC Sampling and Conversion Time

LOWSP[2:0] Value	Sampling cycle	Sampling time @F_{PCLK} = 72MHz unit:ns	Conversion time unit:ns	Totol time unit:ns
000	3	42	404	446
001	6	83	404	487
010	9	125	404	529

LOWSP[2:0] Value	Sampling cycle	Sampling time @F_{PCLK} = 72MHz unit:ns	Conversion time unit:ns	Totol time unit:ns
011	15	208	404	612
100	30	417	404	821
101	60	833	404	1237
110	120	1667	404	2071
111	480	6667	404	7071

7.5 Sampling Mode

Two Triggering Methods for Sampling Are Available:

- Manual Trigger: Write 1 to ADCS to trigger a single sampling and conversion cycle for the channel selected by ADCISA[4:0], and the conversion result is stored in ADCVA[11:0].
- Sequence Conversion: When the sequence conversion trigger conditions are met, sampling and conversion are performed channel by channel in ascending order based on the valid DSn numbers selected in the sequence.

7.6 Conversion Mode

The SC32F15G series ADC has two conversion modes:

7.6.1 Single Conversion Mode (CONT=0)

This mode is selected when ADC_CON.CONT=0.

- Write 1 to ADCS to trigger a single sampling and conversion cycle for the channel selected by ADCISA[4:0], and the conversion result is stored in ADCVA[11:0].

7.6.2 Sequence Conversion Mode (CONT=1)

This mode is selected when ADC_CON.CONT=1.

- Write 1 to ADCS triggers a complete sequence conversion

7.6.2.1 Sequence Configuration

Users can flexibly configure the starting sampling channel and sequence length according to their needs. The sequence segmentation is as follows:

- 16 Channel Configuration Entries (DSn[4:0], n = 0~15): All channels included in the sequence are defined by DSn
- SQSTR0[3:0] Defines the Starting Position of the Sequence
- SQCNT0[3:0] Defines the Number of Samples in the Sequence

For example:

Configure DSn，n=0~15 (Subsequent DSn configurations will follow this example)		**SQSTR0[3:0]=0, Sequence start at: DS0 SQCNT0[3:0]=13，Sequence length：14**			**SQSTR0[3:0]=4, Sequence start at: DS4 SQCNT0[3:0]=15，Sequence length：16**		
DSn	**Channel**	**Sequence**	**DSn**	**Sampling channel**	**Sequence**	**DSn**	**Sampling channel**
DS0	AIN0	**1**	DS0	AIN0	13	DS0	AIN0
DS1	AIN1	2	DS1	AIN1	14	DS1	AIN1
DS2	AIN6	3	DS2	AIN6	15	DS2	AIN6
DS3	AIN7	4	DS3	AIN7	16	DS3	AIN7
DS4	AIN4	5	DS4	AIN4	**1**	DS4	AIN4
DS5	AIN5	6	DS5	AIN5	2	DS5	AIN5
DS6	AIN14	7	DS6	AIN14	3	DS6	AIN14
DS7	AIN15	8	DS7	AIN15	4	DS7	AIN15
DS8	AIN8	9	DS8	AIN8	5	DS8	AIN8
DS9	AIN9	10	DS9	AIN9	6	DS9	AIN9
DS10	AIN10	11	DS10	AIN10	7	DS10	AIN10
DS11	AIN11	12	DS11	AIN11	8	DS11	AIN11
DS12	AIN12	13	DS12	AIN12	9	DS12	AIN12
DS13	AIN13	14	DS13	AIN13	10	DS13	AIN13
DS14	1/4 VDD	\	DS14	1/4 VDD	11	DS14	1/4 VDD
DS15	Temperature Sensor	\	DS15	Temperature Sensor	12	DS15	Temperature Sensor

SQSTR0[3:0]=6，Sequence start at: DS6 SQCNT0[3:0]=7，Sequence length：8			SQSTR0[3:0]=12，Sequence start at: DS12 SQCNT0[3:0]=5，Sequence length：6			SQSTR0[3:0]=4，Sequence start at: DS4 SQCNT0[3:0]=0，Sequence length：1		
Sequence	DSn	Sampling channel	Sequence	DSn	Sampling channel	Sequence	DSn	Sampling channel
\	DS0	AIN0	5	DS0	AIN0	\	DS0	AIN0
\	DS1	AIN1	6	DS1	AIN1	\	DS1	AIN1
\	DS2	AIN6	\	DS2	AIN6	\	DS2	AIN6
\	DS3	AIN7	\	DS3	AIN7	\	DS3	AIN7
\	DS4	AIN4	\	DS4	AIN4	**1**	DS4	AIN4
\	DS5	AIN5	\	DS5	AIN5	\	DS5	AIN5
1	DS6	AIN14	\	DS6	AIN14	\	DS6	AIN14
2	DS7	AIN15	\	DS7	AIN15	\	DS7	AIN15
3	DS8	AIN8	\	DS8	AIN8	\	DS8	AIN8
4	DS9	AIN9	\	DS9	AIN9	\	DS9	AIN9
5	DS10	AIN10	\	DS10	AIN10	\	DS10	AIN10
6	DS11	AIN11	\	DS11	AIN11	\	DS11	AIN11
7	DS12	AIN12	**1**	DS12	AIN12	\	DS12	AIN12
8	DS13	AIN13	2	DS13	AIN13	\	DS13	AIN13
\	DS14	1/4 VDD	3	DS14	1/4 VDD	\	DS14	1/4 VDD
\	DS15	Temperature Sensor	4	DS15	Temperature Sensor	\	DS15	Temperature Sensor

7.7 ADC Overflow

If the converted data is not read promptly by the CPU or DMA before new data is generated, an overflow flag (OVERRUN) will indicate an overflow event.

When an overflow occurs, the ADC will remain in working state and can continue with conversions. However, the OVERRUN flag will be set to 1 by hardware, and the value of ADCV will be overwritten by the latest conversion result and any previously unread data will be lost.

The OVERRUN flag is set to 1 by hardware when an overflow occurs, and it is automatically cleared to 0 after reading ADCV.

7.8 ADC and DMA Controller Collaboration

By selecting one of the DMA channels with REQSRC[5:0]=59 (indicating the DMA channel's request source is ADC) and setting ADC_CON.DMAEN=1, a DMA request will be generated after every ADC conversion. After enabling DMA and ADC, DMA can transfer the converted data from the ADCV Register to the target location selected by the software.

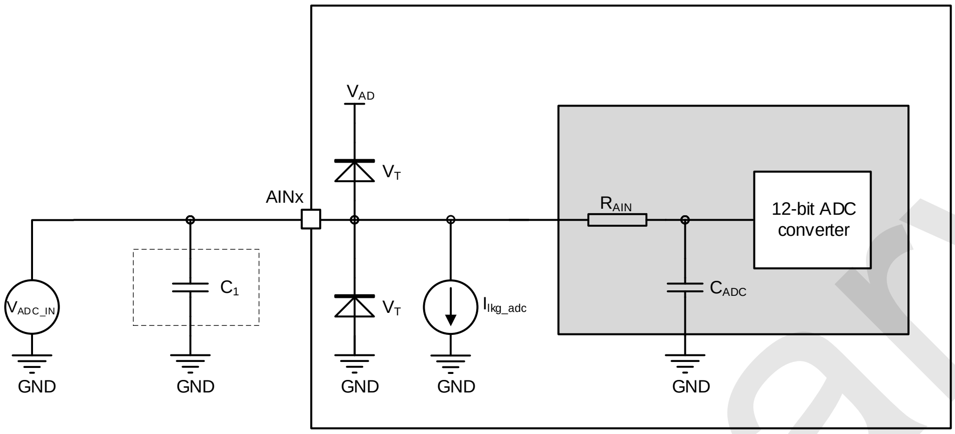
If DMA cannot process the DMA transfer request promptly, an overflow (OVERRUN=1) will be generated by the ADC. However, this does not affect the DMA transfer request. Users can read the ADCV value from the RAM area and check if the most significant bit is 1 to determine whether an overflow has occurred.

7.9 ADC Conversion Steps

The actual operation steps required for the user to perform ADC conversion are as follows:

- Set the ADC input pin; (set the bit corresponding to AINx as ADC input, usually the ADC pin will be fixed in advance);
- Set the ADC reference source using the REFSEL bit: If VREF is selected, the reference value for VREF must be configured separately.
- Enable the ADC module power: Write 1 to the ADCEN bit to power on the ADC module.
- Set ADCISA[4:0]: Select the idle channel for manual trigger sampling.
- Configure upper and lower thresholds using UPTH[11:0] and DOWTH[11:0]: If the ADC conversion result exceeds the thresholds, the corresponding flag will be set. Additionally, users can configure whether threshold comparison is enabled for each channel via the ADC_TH_CFG register
- Choose single or sequence conversion mode:
    - For single conversion, set CONT to 0 and write 1 to ADCS to trigger the conversion for the channel selected by ADCISA.
    - For sequence conversion, pre-configure the sequence order in the ADC_SQ0 register, set the starting position of the sequence using SQSTR0 in the ADC_SQCNT register and the number of samples using SQCNT0, to start sequence conversion, set CONT to 1 and write 1 to ADCS. The conversion will proceed in ascending order based on the valid DSn numbers in the sequence.
- ADCIF flag indicates completion of a conversion: If ADC interrupts are enabled and EOCIE is set, a conversion completion interrupt will be triggered. The user must clear the ADCIF flag in software.
- EOSIF0 flag indicates completion of a sequence conversion: If ADC interrupts are enabled and EOSIE0 is set, a sequence completion interrupt will be triggered. The user must clear the EOSIF0 flag in software
- In sampling mode, the conversion result for the sampled channel is stored in ADCVA[11:0]: If the ADCV register is not read in time, the next conversion result will overwrite the current one, and the OVERRUN bit will be set to indicate overflow. Overflow does not affect sampling or conversion. The OVERRUN bit will be automatically cleared when the ADCV register is read
- If ADC conversion thresholds are set: After the conversion result is stored in ADCVA[11:0], it will be compared with the upper and lower thresholds. If the result exceeds the thresholds, the UPTHIF (upper threshold overflow flag) or DOWTHIF (lower threshold overflow flag) will be set. If ADC interrupts are enabled and UPTHIE/DOWTHIE is set, a threshold overflow interrupt will be triggered.
- DMA can be used to transfer conversion data.

7.10 ADC Connection Circuit Diagram



Note:

1. **C1 is an external 0.01μF capacitor. Users are advised to add this capacitor to improve the performance of the ADC.**
2. **For detailed electrical parameters related to the ADC, please refer to** Section 27.9 ADC Characteristics **in SC32F15G series datasheet.**

7.11 ADC Interrupt

After the SC32F15G series ADC conversion complete, the ADCIF flag will be set, and if ADC_CON.INTEN=1, an interrupt will be generated. Each sequence has its corresponding interrupt enable bit and flag.

Interrupt Event	Interrupt Enable Control Bit	Event Flag	Interrupt Enable Sub-Switch
ADC conversion completion interrupt request	ADC_IDE->INTEN	ADCIF	EOCIE
Sequence sampling and conversion complete interrupt request		EOSIF0	EOSIE0
lower thresholds overflow interrupt reqeust		DOWTHIF	DOWTHIE
upper thresholds overflow interrupt reqeust		UPTHIF	UPTHIE

7.12 ADC Register

7.12.1 ADC Related Register

7.12.1.1 ADC Control Register (ADC_CON)

Register	R/W	Description	Reset Value	POR
ADC_CON	R/W	ADC Control Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
-	-	-	ADCISA[4:0]				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
-	-	-	-	-	LOWSP[2:0]		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADCEN	-	CONT	REFSEL	-	-	-	ADCS

<table>
<tr><th>Bit number</th><th>Bit Mnemonic</th><th>Description</th></tr>
<tr><td>21~20</td><td>ADCISA[4:0]</td><td>Available value is shown as below:
<table>
<tr><td>00000</td><td>ADC0 / OP0</td></tr>
<tr><td>00001</td><td>ADC1 / OP1</td></tr>
<tr><td>00010</td><td>ADC2</td></tr>
<tr><td>00011</td><td>ADC3</td></tr>
<tr><td>00100</td><td>ADC4</td></tr>
<tr><td>00101</td><td>ADC5</td></tr>
<tr><td>00110</td><td>ADC6 / OP2</td></tr>
<tr><td>00111</td><td>ADC7</td></tr>
<tr><td>01000</td><td>ADC8</td></tr>
<tr><td>01001</td><td>ADC9</td></tr>
<tr><td>01010</td><td>ADC10</td></tr>
<tr><td>01011</td><td>ADC11</td></tr>
<tr><td>01100</td><td>ADC12</td></tr>
<tr><td>01101</td><td>ADC13</td></tr>
<tr><td>01110</td><td>ADC14</td></tr>
<tr><td>01111</td><td>ADC15</td></tr>
<tr><td>10000</td><td>1/4 VDD</td></tr>
<tr><td>10001</td><td>Temperature sensor</td></tr>
<tr><td>else</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>10~8</td><td>LOWSP[2:0]</td><td>ADC Sampling Period Selection Control Bit
000: Sampling time is 3 system clock(about 42ns @ f_{PCLK2}=72MHz)
001: Sampling time is 6 system clock(about 83 @ f_{PCLK2}=72MHz)</td></tr>
</table>

Bit number	Bit Mnemonic	Description
		010: Sampling time is 9 system clock(about 125 @ f_{PCLK2}=72MHz) 011: Sampling time is 15 system clock(about 208 @ f_{PCLK2}=72MHz) 100: Sampling time is 30 system clock(about 417 @ f_{PCLK2}=72MHz) 101: Sampling time is 60 system clock(about 833 @ f_{PCLK2}=72MHz) 110: Sampling time is 120 system clock(about 1667 @ f_{PCLK2}=72MHz) 111: Sampling time is 480 system clock(about 6667 @ f_{PCLK2}=72MHz) Others: Reserved Description: The total time for ADC from sampling to completing the conversion is calculated as follows: T_{ADC} = Sampling time + Conversion time ADC conversion time is fixed at 404ns
7	ADCEN	ADC Module Power Startup Control Bit 0: Disable ADC module power 1: Enable ADC module power
5	CONT	Single/Continuous Conversion Mode Select Bit This bit can be set to 1 or cleared by software. 0: Single conversion mode, write 1 to ADCS to trigger a single sampling and conversion cycle for the channel selected by ADCISA[4:0] 1: Continuous conversion mode, write 1 to ADCS triggers a complete sequence conversion
4	REFSEL	ADC Module Reference Source Selection Bit 0: Select VDD as reference source 1: Select VREF as reference source
0	ADCS	ADC Conversion Trigger Control Bit This bit serves as the trigger signal for ADC conversion, set to 1 by software, and cleared to 0 by hardware. Writing 1 to this bit triggers a single ADC conversion. **Note: After setting ADCS to 1, refrain from writing to the ADC_CON Register until the interrupt flag ADCIF is set.**
31~21 15~11 6 3~1	-	Reserved

7.12.1.2 ADC Flag Register (ADC_STS)

Register	R/W	Description	Reset Value	POR
ADC_STS	R/W	ADC Flag Register	0x0000_0000	0x0000_0000

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

15	14	13	12	11	10	9	8
-	-	-	-	-	-	DOWTHIF	UPTHIF
7	6	5	4	3	2	1	0
BUSY	-	-	-	-	-	EOSIF0	ADCIF

Bit number	Bit Mnemonic	Description
9	DOWTHIF	Lower Thresholds Overflow Flag: 0:ADCVA[11:0]≥DOWTHIF[11:0] 1:ADCVA[11:0]< DOWTHIF[11:0]
8	UPTHIF	Upper Thresholds Overflow Flag: 0:ADCVA[11:0]≤UPTH[11:0] 1:ADCVA[11:0]>UPTH[11:0]
7	BUSY	ADC Busy Status Bit: 0: ADC is in idle state 1: ADC sequence is sampling/converting During ADC sampling/converting, any operation to ADCCON register is invalid before BUSY bit is cleared
1	EOSIF0	Sequence Sampling and Coverting Complete Flag This bit is set to 1 by hardware, and is cleared by writing to 1 through software. 0: Sequence sampling and converting incomplete 1: Sequence sampling and converting complete This bit will be set to 1 by hardware when the conversion result of last channel occurs in ADCV register, and if ADC_CON.INTEN=1, an interrupt will be generated.
0	ADCIF	ADC Interrupt Request Flag This bit is set to 1 by hardware, and is cleared by writing to 1 through software. This bit will be set to 1 by hardware after the ADC conversion is complete, and if ADC_CON.INTEN=1, an interrupt will be generated.
31~10 6~2	-	Reserved

7.12.1.3 ADC Conversion Value Register (ADCV)

Register	R/W	Description	Reset Value	POR
ADCV	Read Only	ADC Conversion Value Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
OVERRUN	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8

-	-	-	-	ADCVA[11:8]			
7	6	5	4	3	2	1	0
ADCVA[7:0]							

Bit number	Bit Mnemonic	Description
31	OVERRUN	Flowout Flag(Read Only) If ADC conversion request is not handled promptly by the CPU or DMA, this bit will be set by hardware. This bit will be automatically cleared after reading ADCV. **Note: When overflow occurs, the value of ADCV will be overwritten by the latest conversion result and any previously unread data will be lost.**
11~0	ADCVA[11:0]	12 bits ADC conversion results Current sampling channel conversion result stores at ADCVA[11:0]
30~12	-	Reserved

7.12.1.4 ADC Port Configuration Register (ADC_CFG)

Register	R/W	Description	Reset Value	POR
ADC_CFG	R/W	ADC Port Configuration Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
AIN15	AIN14	AIN13	AIN12	AIN11	AIN10	AIN9	AIN8
7	6	5	4	3	2	1	0
AIN7	AIN6	AIN5	AIN4	AIN3	AIN2	AIN1	AIN0

Bit number	Bit Mnemonic	Description
15~0	AINx (x=0~15)	ADC Port Configuration Register 0: The AINx corresponding port cannot be used as an ADC input channel. 1: The AINx corresponding port can be used as an ADC input channel. When ADCIS[4:0] selects AINx as the ADC input channel, the pull-up resistor on the AINx port will be automatically removed.
31~16	-	Reserved

7.12.1.5 ADC Channel Thresholds Enable Register ADC_TH_CFG

Register	R/W	Description	Reset Value	POR
ADC_TH_CFG	R/W	ADC Channel Thresholds Enable Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
AIN15	AIN14	AIN13	AIN12	AIN11	AIN10	AIN9	AIN8
7	6	5	4	3	2	1	0
AIN7	AIN6	AIN5	AIN4	AIN3	AIN2	AIN1	AIN0

Bit number	Bit Mnemonic	Description
15~0	AINx	ADC Channel Thresholds Enable Control Bit,x=0~15 0:Disable AINx thresholds function 1:Enable AINx thresholds function
31~16	-	Reserved

7.12.1.6 ADC Lower Thresholds Configuration Register ADC_DOWTH

Register	R/W	Description	Reset Value	POR
ADC_DOWTH	R/W	ADC Lower Thresholds Configuration Register	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
11~0	DOWTH[11:0]	ADC Lower Thresholds Configuration Bits When the current conversion result ADCVA[11:0] is less than the value set in DOWTH[11:0], the DOWTHIF flag is set. If DOWTHIE is enabled at this time, an ADC lower threshold comparison interrupt can be triggered.
31~12	-	Reserved

7.12.1.7 ADC Upper Thresholds Configuration Register ADC_UPTH

Register	R/W	Description	Reset Value	POR
ADC_UPTH	R/W	ADC Upper Thresholds Configuration Register	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
11~0	UPTH[11:0]	ADC Upper Thresholds Configuration Bits When the current conversion result ADCVA[11:0] is more than the value set in UPTH[11:0], the UPTHIF flag is set. If UPTHIE is enabled at this time, an ADC upper threshold comparison interrupt can be triggered.
31~12	-	Reserved

7.12.1.8 ADC Interrupt Enable And DMA Control Register ADC_IDE

Register	R/W	Description	Reset Value	POR
ADC_IDE	R/W	ADC Interrupt Enable And DMA Control Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	DOWTHIE	UPTHIE
7	6	5	4	3	2	1	0
INTEN	DMAEN	-	-	-	-	EOSIE0	EOCIE

Bit number	Bit Mnemonic	Description
9	DOWTHIE	Lower Thresholds Overflow Interrupt Enable Bit 0: An interrupt will not be generated when DOWTHIF is set 1: An interrupt will be generated when DOWTHIF is set
8	UPTHIE	Upper Thresholds Overflow Interrupt Enable Bit 0: An interrupt will not be generated when UPTHIF is set 1: An interrupt will be generated when UPTHIF is set
7	INTEN	Interrupt Request CPU Enable Control Bit 0: Disable interrupt request 1: Enable interrupt request
6	DMAEN	Direct Memory Access (DMA) Enable

Bit number	Bit Mnemonic	Description
		This bit is set and cleared by software to enable the generation of DMA requests, allowing the DMA controller to automatically manage the converted data. 0: DMA is disabled. 1: DMA is enabled. Note: Ensure that no conversion is currently in progress before writing to this bit via software.
1	EOSIE0	Sequence Sampling and Converting Complete Interrupt Enable Bit 0: An interrupt will not be generated when EOSIF0 is set 1: An interrupt will be generated when EOSIF0 is set
0	EOCIE	Each ADC Conversion Complete Interrupt Enable Bit 0: An interrupt will not be generated when ADCIF is set 1: An interrupt will be generated when ADCIF is set
31~10 5~2	-	Reserved

7.12.1.9 ADC Sequence Channel Configuration Register ADC_SQCNT

Register	R/W	Description	Reset Value	POR
ADC_SQCNT	R/W	ADC Sequence Channel Configuration Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
SQSTR0[3:0]				SQCNT0[3:0]			

Bit number	Bit Mnemonic	Description
7~4	SQSTR0[3:0]	Sequence Starting Position Configuration Bit This bit is used to set the starting DSn (n = 0~15) for the sequence each time it is initiated.
3~0	SQCNT0[3:0]	Sequence Sample Count Configuration Bit The number of samples in the sequence is calculated as: Number of Samples = SQCNT0[3:0] + 1 This means the sequence can support a maximum of 16 samples.
31~8	-	Reserved

7.12.1.10 ADC Sequence Configuration Register ADC_SQ0

Register	R/W	Description	Reset Value	POR
ADC_SQ0	R/W	ADC Sequence Configuration Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	DS15[4:0]				
23	22	21	20	19	18	17	16
-	-	-	DS14[4:0]				
15	14	13	12	11	10	9	8
-	-	-	DS13[4:0]				
7	6	5	4	3	2	1	0
-	-	-	DS12[4:0]				
31	30	29	28	27	26	25	24
-	-	-	DS11[4:0]				
23	22	21	20	19	18	17	16
-	-	-	DS10[4:0]				
15	14	13	12	11	10	9	8
-	-	-	DS9[4:0]				
7	6	5	4	3	2	1	0
-	-	-	DS8[4:0]				
31	30	29	28	27	26	25	24
-	-	-	DS7[4:0]				
23	22	21	20	19	18	17	16
-	-	-	DS6[4:0]				
15	14	13	12	11	10	9	8
-	-	-	DS5[4:0]				
7	6	5	4	3	2	1	0
-	-	-	DS4[4:0]				
31	30	29	28	27	26	25	24
-	-	-	DS3[4:0]				
23	22	21	20	19	18	17	16
-	-	-	DS2[4:0]				
15	14	13	12	11	10	9	8
-	-	-	DS1[4:0]				
7	6	5	4	3	2	1	0
-	-	-	DS0[4:0]				

Bit number	Bit Mnemonic	Description
28~24 20~16 12~8 4~0	DSn[4:0]	DSn[4:0]: n=0~15, ADC Sampling Sequence Signal Configuration 00000 ADC0 / OP0 00001 ADC1 / OP1 00010 ADC2

<table>
<tr><th>Bit number</th><th>Bit Mnemonic</th><th>Description</th></tr>
<tr><td></td><td></td><td><table>
<tr><td>00011</td><td>ADC3</td></tr>
<tr><td>00100</td><td>ADC4</td></tr>
<tr><td>00101</td><td>ADC5</td></tr>
<tr><td>00110</td><td>ADC6 / OP2</td></tr>
<tr><td>00111</td><td>ADC7</td></tr>
<tr><td>01000</td><td>ADC8</td></tr>
<tr><td>01001</td><td>ADC9</td></tr>
<tr><td>01010</td><td>ADC10</td></tr>
<tr><td>01011</td><td>ADC11</td></tr>
<tr><td>01100</td><td>ADC12</td></tr>
<tr><td>01101</td><td>ADC13</td></tr>
<tr><td>01110</td><td>ADC14</td></tr>
<tr><td>01111</td><td>ADC15</td></tr>
<tr><td>10000</td><td>1/4 VDD</td></tr>
<tr><td>10001</td><td>Temperature sensor</td></tr>
<tr><td>else</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>31~29
23~21
15~13
7~5</td><td>-</td><td>Reserved</td></tr>
</table>

7.12.2 ADC Register Mapping

Register	Offset Address	R/W	Description	Reset Value	POR
ADC Base Address:0x4002_2110					
ADC_CON	0x00	R/W	ADC Control Register	0x0000_0000	0x0000_0000
ADC_STS	0x04	R/W	ADC Flag Register	0x0000_0000	0x0000_0000
ADCV	0x08	R/W	ADC Conversion Value Register	0x0000_0000	0x0000_0000
ADC_CFG	0x0C	R/W	ADC Port Configuration Register	0x0000_0000	0x0000_0000
ADC_TH_CFG	0x10	R/W	ADC Channel Thresholds Enable Register	0x0000_0000	0x0000_0000
ADC_LOWTH	0x14	R/W	ADC Lower Thresholds Configuration Register	0x0000_0000	0x0000_0000

Register	Offset Address	R/W	Description	Reset Value	POR
ADC_UPTH	0x18	R/W	ADC Upper Thresholds Configuration Register	0x0000_0000	0x0000_0000
ADC_IDE	0x1C	R/W	ADC Interrupt Enable And DMA Control Register	0x0000_0000	0x0000_0000
ADC_SQCNT	0x24	R/W	ADC Sequence Channel Configuration Register	0x0000_0000	0x0000_0000
ADC_SQ0	0x28	R/W	ADC Sequence Configuration Register	0x0000_0000	0x0000_0000

8 Internal Reference Source(VREF)

8.1 Overview

The SC32F15G series features an independent internal reference source(VREF), and can be reference source of some peripherals.

8.2 Clock Source

- The SC32F15G series VREF has only one clock source, which is derived from PCLK2

8.3 Internal Reference Source Configuration

There are four methods to configure internal reference source module:

- VREFCFG1=0,VREFCFG0=0: disable Vref PIN port, disable internal reference source module
- VREFCFG1=0,VREFCFG0=1: the analog circuit uses the internal reference, and the Vref voltage is determined by the VREFS[1:0] selection.
- VREFCFG1=1,VREFCFG0=0: the analog circuit uses an external reference, and Vref is input through the external Vref PIN.
- VREFCFG1=1,VREFCFG0=1: the analog circuit uses the internal reference, and the Vref voltage is determined by the VREFS[1:0] selection.

8.4 Internal Reference Source Output

Once the internal reference source module is enabled, VREF can be utilized as a reference for ADC, DAC, OP, or CMP. Additionally, it can be divided by two and output through the VMID pin.

The specific settings are as follows:

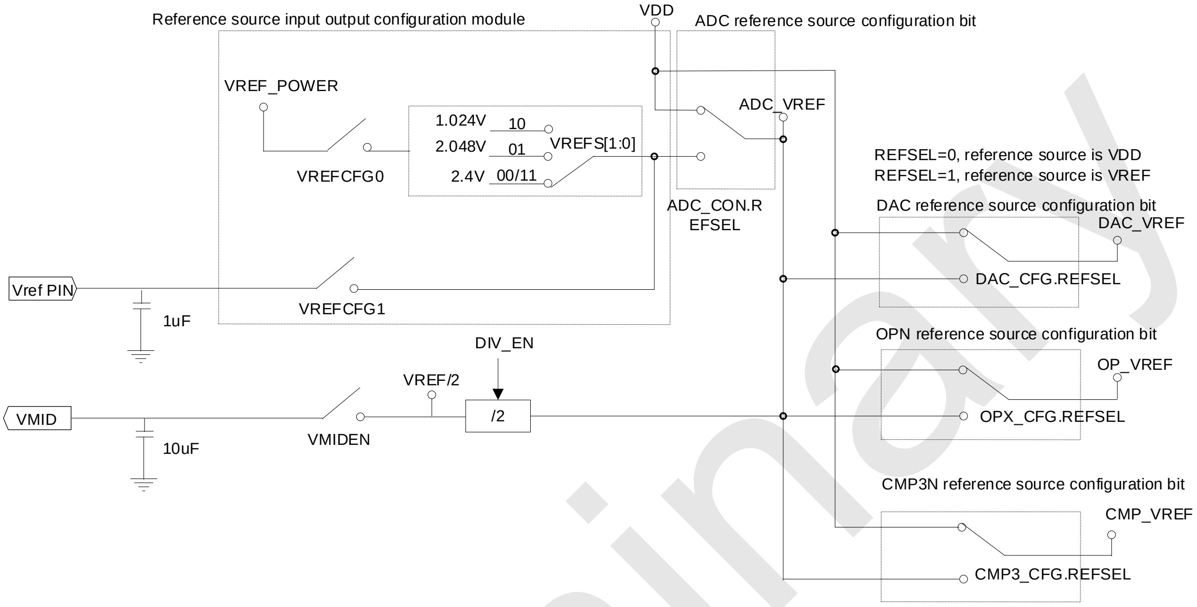
- The default reference source for the ADC/DAC/OP/CMP modules is VDD. When VREF is selected as the reference source, the REFSEL bit in the relevant peripheral register must be enabled. For example, when VREF is selected as the reference source for the DAC module, the REFSEL bit in the DAC configuration register (DAC_CFG) must be enabled.

Note:

- If only DAC/OP/CMP is enabled, the selection of the external peripheral reference source is not affected.
- If ADC and other peripherals (DAC/OP/CMP) are enabled simultaneously, when the ADC reference source is VDD, the reference source for other peripherals can only be VDD. To change the reference source of other peripherals to VREF, the ADC reference source must first be set to VREF!
- When selecting VREF/2 to be output through the VMID pin, the internal reference divider enable bit DIV_EN must first be set to 1 to output a voltage that is half of VREF through the VREF/2 point. Then, the VMID pin enable bit VMIDEN must be set to 1.

8.5 Internal Reference Source Structure Diagram

Vref PIN can be used as input port, and VMID can be only used as output port


Reference source input output configuration module
VDD
ADC reference source configuration bit
VREF_POWER
1.024V 10
2.048V 01
2.4V 00/11
VREFS[1:0]
VREFCFG0
ADC_VREF
ADC_CON.REFSEL
REFSEL=0, reference source is VDD
REFSEL=1, reference source is VREF
DAC reference source configuration bit
DAC_VREF
DAC_CFG.REFSEL
Vref PIN
1uF
VREFCFG1
DIV_EN
OPN reference source configuration bit
OP_VREF
VREF/2
/2
VMID
10uF
VMIDEN
OPX_CFG.REFSEL
CMP3N reference source configuration bit
CMP_VREF
CMP3_CFG.REFSEL

8.6 VREF Register

8.6.1 VREF Related Register

8.6.1.1 VREF Configuration Register VREF_CFG

Register	R/W	Description	Reset Value	POR
VREF_CFG	R/W	VREF Configuration Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	DIV_EN	VMIDEN	-	VREFS[1:0]		VREFCFG1	VREFCFG0

Bit number	Bit Mnemonic	Description
6	DIV_EN	Internal Reference Divider Enable Bit 0: Disabled, no output at the VREF/2 point. 1: Enabled, the VREF/2 point outputs a voltage that is half of VREF.
5	VMIDEN	VMID Port Enable Bit 0: The VMID pin is used for other multiplexed functions.

Bit number	Bit Mnemonic	Description
		1: The VMID pin outputs VREF/2.
3~2	VREFS[1:0]	System Analog Circuit Vref Voltage Selection 00: Reserved (default connection to 2.4V) 01: Set the ADC Vref to an accurate internal 2.048V 10: Set the ADC Vref to an accurate internal 1.024V 11: Set the ADC Vref to an accurate internal 2.4V
1~0	VREFCFG1、VREFCFG0	VREFCFG[1:0] System Analog Circuit Reference Module VREF Setting Bits 00: Disable Vref PIN port, disable internal reference source module 01: The analog circuit uses the internal reference, and the Vref voltage is determined by the VREFS[1:0] selection. 10: The analog circuit uses an external reference, and Vref is input through the external Vref PIN. 11: The analog circuit uses the internal reference, and the Vref voltage is determined by the VREFS[1:0] selection.
31~7 4	-	Reserved

8.6.2 VREF Register Mapping

Register	Offset Address	R/W	Description	Reset Value	POR
VREF Base Address:0x4002_2190					
VREF_CFG	0x0C	R/W	VREF Configuration Register	0x0000_0000	0x0000_0000

9 Digital-to-Analog Converter(DAC)

9.1 Overview

The SC32F15G series features an independent digital-to-analog converter(DAC). The DAC features two independent output ports, DACOUT0 and DACOUT1. Additionally, the DAC can internally route its output to the inverting input port of OP1 or OP2.

9.2 Clock Source

- The SC32F15G series DAC has only one clock source, which is derived from PCLK2

9.3 Feature

- Reference source selectable: VDD or VREF
- Output ports:
  - 2 independent DAC output port DACOUT0 and DACOUT1
  - Inverting input port of OP1/OP2
  - Negative port of CMP0/1/2/3

9.4 DAC Register

9.4.1 DAC Related Register

9.4.1.1 DAC Status Register DAC_STS

Register	R/W	Description	Reset Value	POR
DAC_STS	R/W	DAC Status Register	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
0	STA	DAC Conversion Status Bit This bit is a status bit, set and cleared by hardware: 0: The DAC module is idle/conversion completed.

Bit number	Bit Mnemonic	Description
		1: The DAC module is currently converting.
31~1	-	Reserved

9.4.1.2 DAC Conversion Register DAC_IN

Register	R/W	Description	Reset Value	POR
DAC_IN	R/W	DAC Conversion Register	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
9~0	DACV[9:0]	DAC Output Voltage: VDACOUT = (Vref / 1024) * DACV[9:0] Note: The converted value takes effect immediately after it is written to this register.
31~10	-	Reserved

9.4.1.3 DAC Configuration Register DAC_CFG

Register	R/W	Description	Reset Value	POR
DAC_CFG	R/W	DAC Configuration Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	REFSEL	OUT1EN	OUT0EN	DACEN

Bit number	Bit Mnemonic	Description
3	REFSEL	DAC Module Reference Source Configuration Bit 0: Reference source is VDD

Bit number	Bit Mnemonic	Description
		1: Reference source is VREF
2	DACOUT1	DACOUT1 Port Enable Bit 0: The port where DACOUT1 located is used for other multiplexed functions 1: The port where DACOUT1 located outputs the current converted voltage of the DAC
1	DACOUT0	DACOUT0 Port Enable Bit 0: The port where DACOUT0 located is used for other multiplexed functions 1: The port where DACOUT0 located outputs the current converted voltage of the DAC
0	DACEN	DAC Enable Control Bit 0: Disable 1: Enable
31~4	-	Reserved

9.4.2 DAC Register Mapping

Register	Offset Address	R/W	Description	Reset Value	POR
DAC Base Address: 0x4002_2190					
DAC_STS	0x00	R/W	DAC Status Register	0x0000_0000	0x0000_0000
DAC_IN	0x04	R/W	DAC Conversion Register	0x0000_0000	0x0000_0000
DAC_CFG	0x08	R/W	DAC Configuration Register	0x0000_0000	0x0000_0000

10 Temperature Sensor

10.1 Overview

The SC32F15G series features a temperature sensor, and temperature sensor voltage can be measured though ADC.

10.2 Temperature Sensor Operation Step

When using the temperature sensor, the ADC reference voltage should be set to the internal 2.4V reference. For every 1°C increase in temperature, the ADC conversion value will increase by a fixed amount. SinOne has pre-programmed the ADC conversion result corresponding to 25°C for each chip into a specific address during production.

The steps for operating the temperature sensor are as follows:

- Set the ADC reference voltage (Vref) to the internal 2.4V reference source and configure the ADC sampling period. It is recommended to select a sampling clock of 60 or more cycles. Then, enable the ADC module power.
- Select the ADC input channel as the temperature sensor channel.
- Enable the temperature sensor by setting TS_EN to 1.
- Wait for a delay of 20μs.
- Set TS_CHOP to 0 to initiate the first ADC conversion. Once the conversion is complete, record the conversion value as ADC_{Value1}.
- Set TS_CHOP to 1 to initiate the second ADC conversion. Once the conversion is complete, record the conversion value as ADC_{Value2}.
- Calculate the average of the two conversion values:

$$\mathrm{ADC}_{\mathrm{Value}} = \frac{(\mathrm{ADC}_{\mathrm{Value1}} + \mathrm{ADC}_{\mathrm{Value2}})}{2}$$

- Read the factory-programmed ADC conversion value for 25°C ($ADC_{ValueTest}$) from the corresponding address.
- Substitute the values into the formula to calculate the current temperature:

$$\mathrm{Temperature} = 25^{\circ}\mathrm{C} + \frac{(\mathrm{ADC}_{\mathrm{Value}} - \mathrm{ADC}_{\mathrm{ValueTest}})}{8.53}$$

For more detailed information about the temperature sensor, please refer to the “SinOne SC32F1XXX Series MCU Application Guide V1.4”.

10.3 Temperature Sensor Register

10.3.1 Temperature Sensor Related Register

10.3.1.1 Temperature Sensor Configuration Register TS_CFG

Register	R/W	Description	Reset Value	POR
TS_CFG	R/W	Temperature Sensor Configuration Register	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
7	TS_EN	Temperature Sensor Enable Control Bit 0: Disable temperature sensor 1: Enable temperature sensor
0	TS_CHOP	Temperature Sensor Offset Compensation Control Bit Writing 0 to TS_CHOP initiates an ADC conversion to obtain the first value, then writing 1 to TS_CHOP initiates another ADC conversion to obtain the second value. The final result will be obtained by averaging the two values.

10.3.2 Temperature Sensor Register Mapping

Register	Offset Address	R/W	Description	Reset Value	POR
Temperature Sensor Base Address: 0x4002_21E0					
TS_CFG	0x00	R/W	Temperature Sensor Configuration Register	0x0000_0000	0x0000_0000

11 Operational Amplifier (OP)

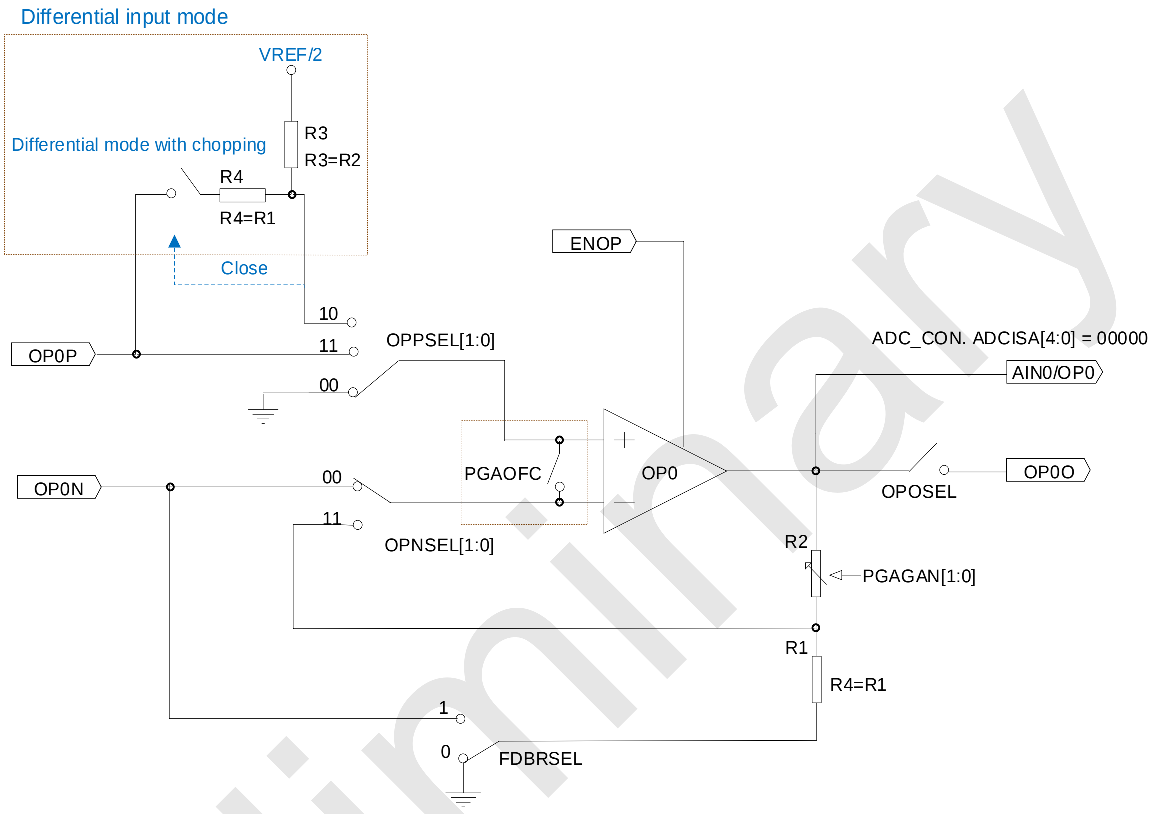
11.1 Overview

The SC32F15G series features 3 independent rail-to-rail operational amplifiers:OP0/OP1/OP2.

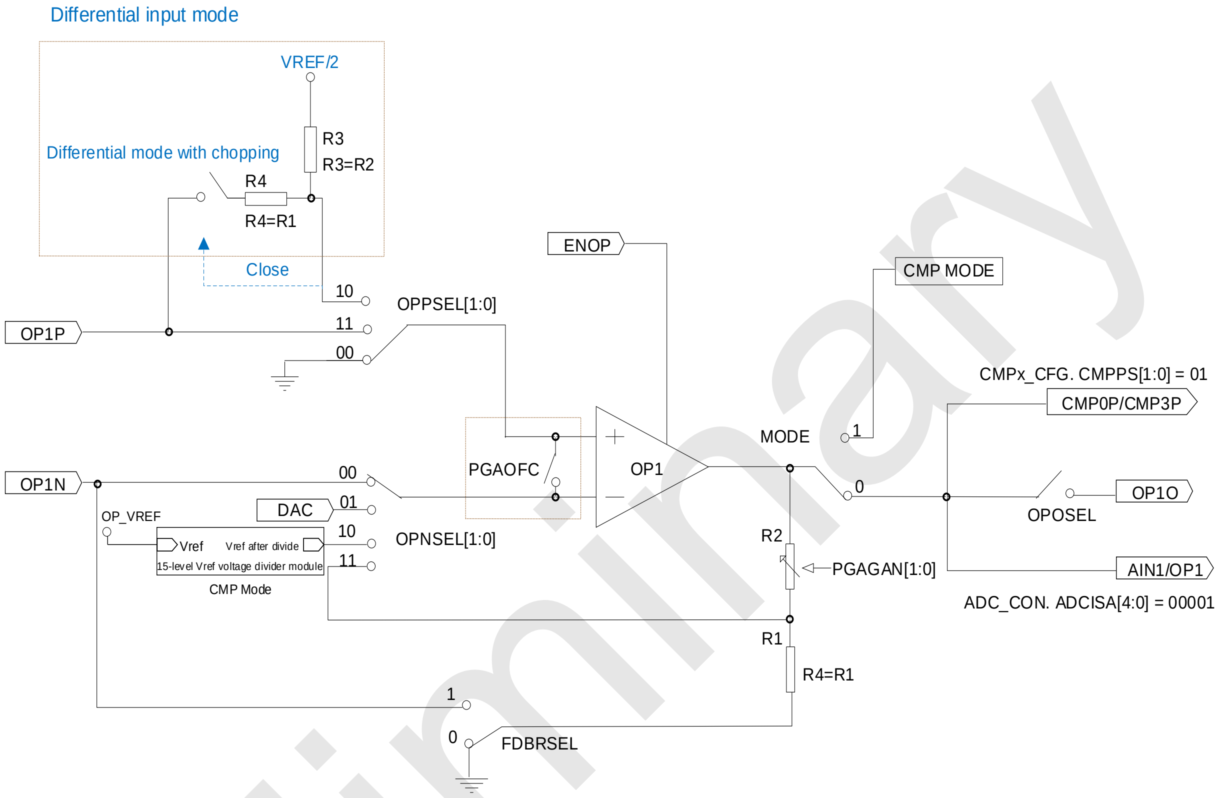
11.2 Feature

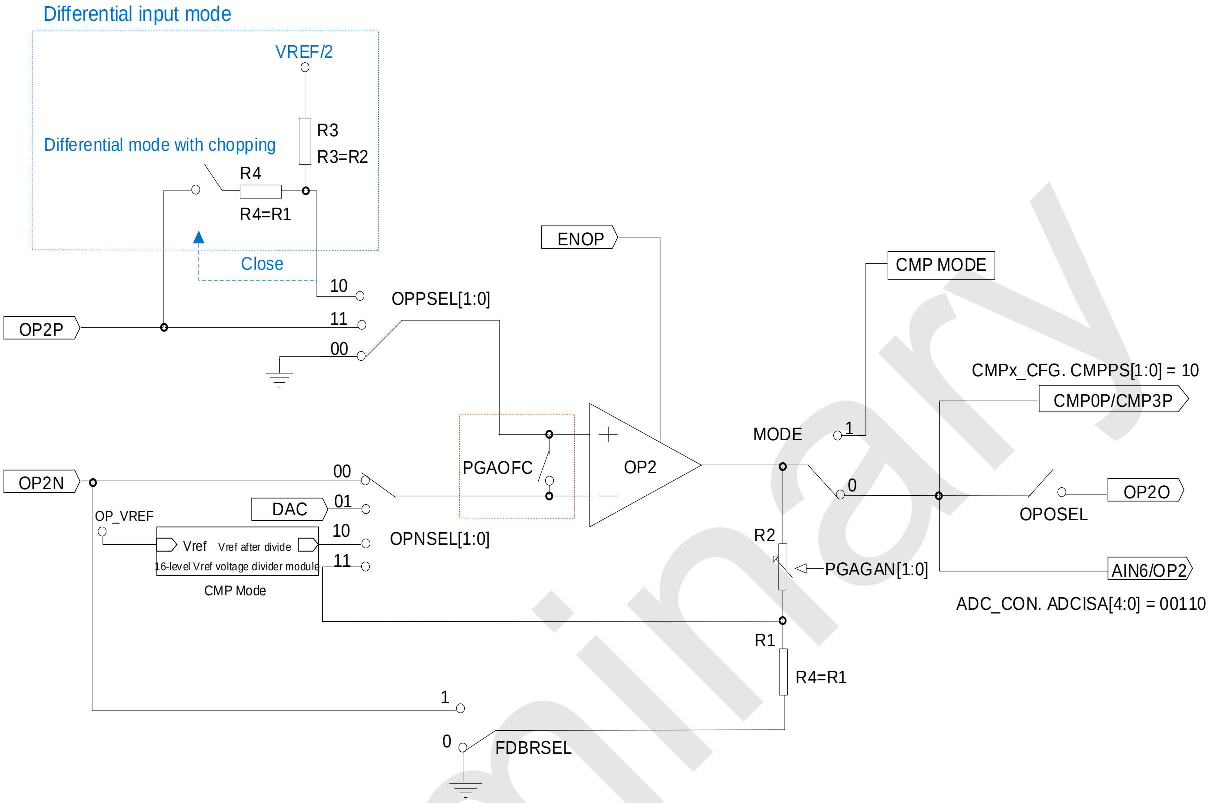
- All three OPs can be configurable as a Programmable Gain Amplifier (PGA)
  - Non-inverting gain: 4/8/16/32
  - Inverting gain: 3/7/15/31
- All three OPs have independent non-inverting input port, inverting input port and output port
- Output port of OP0,OP1,OP2 are multiplexed with ADC channels, and the output value can be directly read through ADCV register
- OP1/OP2 can be configured as CMP
  - Comparator voltage hysteresis: 10-15mV
  - Response time: typical 50ns
- The output of OP1/OP2 can be directly connected to the positive input of CMP0 and CMP3
  - Bandwidth 10MHz
  - Offset voltage≤10mV, and need zero calibration
  - Slew rate≥10V/us

11.3 OP0 Structure Diagram



11.4 OP1/OP2 Structure Diagram





11.5 OP0 Port Selection

11.5.1 OP0 Accuracy Adjustment

The accuracy of OP0 can be adjusted by enabling the PGA offset adjustment control bit (PGAOFC). This is achieved by shorting the positive and negative input terminals of the OP module internally. This process helps to calibrate and minimize any offset errors in the operational amplifier. After calibration, set PGAOFC = 0 to disable the offset adjustment mode and restore normal operation of the OP module.

11.5.2 OP0 Non-Inverting Input Selection

The non-inverting input terminal of OP0 module can be switched and selected by OPPSEL[1:0], and it has three options:

- OP0P external pin
- Internal VSS
- Differential input mode

When differential mode is selected, it is necessary to simultaneously enable the DIV_EN bit in the VREF_CFG register to ensure that the bias voltage V_{REF}/2 is output.

11.5.3 OP0 Inverting Input Selection

The inverting input terminal of the OP0 module has two options:

- OP0N external pin.

When choosing the OP0N external pin as the inverting input for the OP0, the OP0 input control bit OPNSEL[1:0] should be set to 00, and the feedback resistor selection bits FDBRSEL should be set to 1.

- Internal feedback resistor.

When choosing the internal feedback resistor as the inverting input for the OP0, the OP0 input control bit OPNSEL[1:0] should be set to 11, and the feedback resistor selection bits FDBRSEL should be set to 0 or 1, and the internal gain can be selected by internal gain selection bits PGAGAN[1:0].

11.5.4 OP0 Output Selection

The output of the OP0 module has two options:

- Sampling channel of the AD converter

When OP0 output is used as an ADC input, users should set ENOP=1 to enable the OP module, then set ADCEN=1 to power on the ADC. The conversion result of OP can be directly obtained in the ADCV register by selecting the OP output port as the ADC input port through setting ADCISA[4:0]=00000.

- OP0O pin.

When OP outputs through the OP0O pin,users should set ENOP=1 to enable the OP module,then set OPOSEL=1

11.6 OP1/2 Port Selection

11.6.1 OP1/2 Accuracy Adjustment

The accuracy of OP1/2 can be adjusted by enabling the PGA offset adjustment control bit (PGAOFC). This is achieved by shorting the non-inverting and inverting input terminals of the OP module internally. This process helps to calibrate and minimize any offset errors in the operational amplifier. After calibration, set PGAOFC = 0 to disable the offset adjustment mode and restore normal operation of the OP module.

11.6.2 OP1/2 Non-Inverting Input Selection

The non-inverting input terminal of OP1/2 module can be switched and selected by OPPSEL[1:0], and it has three options:

- OP1P/OP2P external pin
- Internal VSS
- Differential input mode

When differential mode is selected, it is necessary to simultaneously enable the DIV_EN bit in the VREF_CFG register to ensure that the bias voltage $V_{REF}/2$ is output.

11.6.3 OP1/2 Inverting Input Selection

The inverting input terminal of the OP1/2 module has four options:

- OP1N/OP2N external pin.

When choosing the OP1N/OP2N external pin as the inverting input for the OP1/2, the OP1/2 input control bit OPNSEL[1:0] should be set to 00, and the feedback resistor selection bits FDBRSEL should be set to 1.

- DAC output

When choosing the DAC output as the inverting input for the OP1/2, the DAC module must be enabled and the OP1/2 input control bit OPNSEL[1:0] should be set to 01

- 15-level voltage divider of OPx_VREF

When choosing the OPRF[3:0] as the inverting input for the OP1/2, the OP1/2 input control bit OPNSEL[1:0] should be set to 10

- Internal feedback resistor.

When choosing the internal feedback resistor as the inverting input for the OP0, the OP1/2 input control bit OPNSEL[1:0] should be set to 11, and the feedback resistor selection bits FDBRSEL should be set to 0 or 1, and the internal gain can be selected by internal gain selection bits PGAGAN[1:0].

11.6.4 OP1/2 Output Selection

The output of the OP module has three options:

- Sampling channel of the AD converter

When OP1/2 output is used as an ADC input, users should set ENOP=1 to enable the OP module and set MODE=0 to configure OP1/2 as amplifer mode, then set ADCEN=1 to power on the ADC. The conversion result of OP can be directly obtained in the ADCV register by selecting the OP output port as the ADC input port through setting ADCISA[4:0]=00001/00110.

- Non-inverting input of the CMP0/3

When OP1/2 is used as the non-inverting input of the CMP0/3, users should set ENOP=1 to enable the OP module, then select OP output port as the CMP input port by channel control bit CMPPS[1:0].

- OP1O/OP2O pin.

When OP outputs through the OP1O/OP2O pin,users should set ENOP=1 to enable the OP module,then set OPOSEL=1

11.7 OP Register

11.7.1 OP0 Related Register

11.7.1.1 OP0 Control Register (OP0_CON)

Register	R/W	Description	Reset Value	POR
OP0_CON	R/W	OP0 Control Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	FDBRSEL	-	TRIMOFFSETN[4:0]				
23	22	21	20	19	18	17	16
PGAOFC	-	-	TRIMOFFSETP[4:0]				
15	14	13	12	11	10	9	8
-	-	-	-	-	-	PGAGAN[1:0]	

7	6	5	4	3	2	1	0
ENOP	-	OPPSEL[1:0]		OPNSEL[1:0]		-	OPOSEL

Bit number	Bit Mnemonic	Description
30	FDBRSEL	Feedback Resister R1 Connection Selection Bit 0: Internal VSS 1: OP0N port
28~24	TRIMOFFSETN[4:0]	Trim for NMOS differential pairs
23	PGAOFC	PGA Input Offset Adjustment Control Bit 0: OP inverting input and non-inverting input are not internally short-circuited 1: OP inverting input and non-inverting input are internally short-circuited (Note: Internally short-circuiting or disconnecting the OP inverting and non-inverting input ends does not affect the selection of OPPSEL and OPNSEL)
20~16	TRIMOFFSETP[4:0]	Trim for PMOS differential pairs
9~8	PGAGAN[1:0]	Internal Gain Selection: 00: Non-inverting gain=4, inverting gain=3 01: Non-inverting gain=8, inverting gain=7 10: Non-inverting gain=16, inverting gain=15 11: Non-inverting gain=32, inverting gain=31
7	ENOP	OP0 Enable Control Bit 0: Disable OP0 1: Enable OP0
5~4	OPPSEL[1:0]	OP Non-inverting signal Connection Selection Bit 00: Internal connect VSS, 0V 10: Differential input mode, with a bias voltage of VREF/2. Note that VREF_CFG.DIV_EN must be enabled simultaneously to provide the VREF/2 voltage output 11: OP0P(external pin)
3~2	OPNSEL[1:0]	OP Inverting signal Connection Selection Bit 00: OP0N(external pin) 01: Reserved 10: Reserved 11: Internal feedback resister R2
0	OPOSEL	OP Output Connection Selection Bit 0: Disconnect from OP0O 1: OP0O(external pin) **Note**: The OP output is always connected to the ADC and the optional CMPxPS.
31	-	Reserved

Bit number	Bit Mnemonic	Description
29 22~21 15~10 6 1		

11.7.2 OP1/OP2 Related Register

11.7.2.1 OP1 Control Register (OP1_CON)

Register	R/W	Description	Reset Value	POR
OP1_CON	R/W	OP1 Control Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	FDBRSEL	-	TRIMOFFSETN[4:0]				
23	22	21	20	19	18	17	16
PGAOFC	-	-	TRIMOFFSETP[4:0]				
15	14	13	12	11	10	9	8
OPRF[3:0]				MODE	-	PGAGAN[1:0]	
7	6	5	4	3	2	1	0
ENOP	-	OPPSEL[1:0]		OPNSEL[1:0]		-	OPOSEL

Bit number	Bit Mnemonic	Description
30	FDBRSEL	Feedback Resister R1 Connection Selection Bit 0: Internal VSS 1: OP1N port
28~24	TRIMOFFSETN[4:0]	Trim for NMOS differential pairs
23	PGAOFC	PGA Input Offset Adjustment Control Bit 0: OP inverting input and non-inverting input are not internally short-circuited 1: OP inverting input and non-inverting input are internally short-circuited (Note: Internally short-circuiting or disconnecting the OP inverting and non-inverting input ends does not affect the selection of OPPSEL and OPNSEL)
20~16	TRIMOFFSETP[4:0]	Trim for PMOS differential pairs
15~12	OPRF[3:0]	The inverting input voltage selection bit for the OP when operating in comparator mode, effective when OPNSEL[1:0] = 10: 0000: 1/16 OPx_VREF 0001: 1/16 OPx_VREF 0010: 2/16 OPx_VREF

Bit number	Bit Mnemonic	Description
		0011: 3/16 OPx_VREF 0100: 4/16 OPx_VREF 0101: 5/16 OPx_VREF 0110: 6/16 OPx_VREF 0111: 7/16 OPx_VREF 1000: 8/16 OPx_VREF 1001: 9/16 OPx_VREF 1010: 10/16 OPx_VREF 1011: 11/16 OPx_VREF 1100: 12/16 OPx_VREF 1101: 13/16 OPx_VREF 1110: 14/16 OPx_VREF 1111: 15/16 OPx_VREF
11	MODE	OP1 Mode Selection Bit 0: Operational amplifier mode 1: Comparator mode
9~8	PGAGAN[1:0]	Internal Gain Selection: 00: Non-inverting gain=4, inverting gain=3 01: Non-inverting gain=8, inverting gain=7 10: Non-inverting gain=16, inverting gain=15 11: Non-inverting gain=32, inverting gain=31
7	ENOP	OP1 Enable Control Bit 0: Disable OP1 1: Enable OP1
5~4	OPPSEL[1:0]	OP Non-inverting signal Connection Selection Bit 00: Internal connect VSS, 0V 10: Differential input mode, with a bias voltage of VREF/2. Note that VREF_CFG.DIV_EN must be enabled simultaneously to provide the VREF/2 voltage output 11: OP1P(external pin)
3~2	OPNSEL[1:0]	OP Inverting signal Connection Selection Bit 00: OP1N(external pin) 01: DAC output 10: Value set by OPRF[3:0] 11: Internal feedback resister R2
0	OPOSEL	OP Output Connection Selection Bit 0: Disconnect from OP1O 1: OP1O(external pin) **Note**: The OP output is always connected to the ADC and the optional CMPxPS.
31 29 22~21	-	Reserved

Bit number	Bit Mnemonic	Description
10 6 1		

11.7.2.2 OP2 Control Register (OP2_CON)

Register	R/W	Description	Reset Value	POR
OP2_CON	R/W	OP2 Control Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	FDBRSEL	-	TRIMOFFSETN[4:0]				
23	22	21	20	19	18	17	16
PGAOFC	-	-	TRIMOFFSETP[4:0]				
15	14	13	12	11	10	9	8
OPRF[3:0]				MODE	-	PGAGAN[1:0]	
7	6	5	4	3	2	1	0
ENOP	-	OPPSEL[1:0]		OPNSEL[1:0]		-	OPOSEL

Bit number	Bit Mnemonic	Description
30	FDBRSEL	Feedback Resister R1 Connection Selection Bit 0: Internal VSS 1: OP2N port
28~24	TRIMOFFSETN[4:0]	Trim for NMOS differential pairs
23	PGAOFC	PGA Input Offset Adjustment Control Bit 0: OP inverting input and non-inverting input are not internally short-circuited 1: OP inverting input and non-inverting input are internally short-circuited (Note: Internally short-circuiting or disconnecting the OP inverting and non-inverting input ends does not affect the selection of OPPSEL and OPNSEL)
20~16	TRIMOFFSETP[4:0]	Trim for PMOS differential pairs
15~12	OPRF[3:0]	The inverting input voltage selection bit for the OP when operating in comparator mode, effective when OPNSEL[1:0] = 10: 0000: 1/16 OPx_VREF 0001: 1/16 OPx_VREF 0010: 2/16 OPx_VREF 0011: 3/16 OPx_VREF 0100: 4/16 OPx_VREF 0101: 5/16 OPx_VREF 0110: 6/16 OPx_VREF

Bit number	Bit Mnemonic	Description
		0111: 7/16 OPx_VREF 1000: 8/16 OPx_VREF 1001: 9/16 OPx_VREF 1010: 10/16 OPx_VREF 1011: 11/16 OPx_VREF 1100: 12/16 OPx_VREF 1101: 13/16 OPx_VREF 1110: 14/16 OPx_VREF 1111: 15/16 OPx_VREF
11	MODE	OP2 Mode Selection Bit 0: Operational amplifier mode 1: Comparator mode
9~8	PGAGAN[1:0]	Internal Gain Selection: 00: Non-inverting gain=4, inverting gain=3 01: Non-inverting gain=8, inverting gain=7 10: Non-inverting gain=16, inverting gain=15 11: Non-inverting gain=32, inverting gain=31
7	ENOP	OP2 Enable Control Bit 0: Disable OP2 1: Enable OP2
5~4	OPPSEL[1:0]	OP Non-inverting signal Connection Selection Bit 00: Internal connect VSS, 0V 10: Differential input mode, with a bias voltage of VREF/2. Note that VREF_CFG.DIV_EN must be enabled simultaneously to provide the VREF/2 voltage output 11: OP0P(external pin)
3~2	OPNSEL[1:0]	OP Inverting signal Connection Selection Bit 00: OP2N(external pin) 01: DAC output 10: Value set by OPRF[3:0] 11: Internal feedback resister R2
0	OPOSEL	OP Output Connection Selection Bit 0: Disconnect from OP2O 1: OP2O(external pin) **Note**: The OP output is always connected to the ADC and the optional CMPxPS.
31 29 22~21 10 6 1	-	Reserved

11.7.2.3 OP1/2 Configuration Register OPX_CFG

Register	R/W	Description	Reset Value	POR
OPX_CFG	R/W	OP1/2 Configuration Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
REFSEL	-	OP_CMPIM2[1:0]		OP_CMPIM1[1:0]		-	-

Bit number	Bit Mnemonic	Description
7	REFSEL	OPx_VREF Reference Source Selection Bit(x=1~2) 0: Reference source is VDD 1: Reference source is VREF
5~4	OP_CMPIM2[1:0]	OP2 Comparator Mode Interrupt Mode Selection Bit 00: No interrupt generated 01: Rising edge interrupt: Interrupt will be generated when IN+ transitions from being less than IN- to being greater than IN- 10: Falling edge interrupt: Interrupt will be generated when IN+ transitions from being greater than IN- to being less than IN- 11: Both edge interrupt: Interrupt will be generated when IN+ transitions from being less than IN- to being greater than IN- or from being greater than IN- to being less than IN-
3~2	OP_CMPIM1[1:0]	OP1 Comparator Mode Interrupt Mode Selection Bit 00: No interrupt generated 01: Rising edge interrupt: Interrupt will be generated when IN+ transitions from being less than IN- to being greater than IN- 10: Falling edge interrupt: Interrupt will be generated when IN+ transitions from being greater than IN- to being less than IN- 11: Both edge interrupt: Interrupt will be generated when IN+ transitions from being less than IN- to being greater than IN- or from being greater than IN- to being less than IN-
31~8 6 1~0	-	Reserved

11.7.2.4 OP1/2 Comparator Mode Status Register OPX_STS

Register	R/W	Description	Reset Value	POR
OPX_STS	R/W	OP1/2 Comparator Mode Status Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	OP_CMP2STA	OP_CMP1STA	-	OP_CMP2IF	OP_CMP1IF	-

Bit number	Bit Mnemonic	Description
5	OP_CMP2STA	OP2 Comparator Mode Output Status Bit 0: OP2 positive terminal voltage is less than negative terminal voltage 1: OP2 positive terminal voltage is greater than negative terminal voltage
4	OP_CMP1STA	OP1 Comparator Mode Output Status Bit 0: OP1 positive terminal voltage is less than negative terminal voltage 1: OP1 positive terminal voltage is greater than negative terminal voltage
2	OP_CMP2IF	OP2 Comparator Mode Interrupt Flag This bit is set to 1 by hardware, and is cleared by writing to 1 through software. 0: OP2 interrupt has not been inturrput 1: This bit will be set to 1 by hardware if OP2 comparator meets the interrupt trigger condition. And OP2 interrupt will be generated if OP_CMP2IE is enable.
1	OP_CMP1IF	OP1 Comparator Mode Interrupt Flag This bit is set to 1 by hardware, and is cleared by writing to 1 through software. 0: OP1 interrupt has not been inturrput 1: This bit will be set to 1 by hardware if OP1 comparator meets the interrupt trigger condition. And OP1 interrupt will be generated if OP_CMP1IE is enable.
31~6 3 0	-	Reserved

11.7.2.5 OP1/2 Comparator Mode Interrupt Enable Register OPX_IDE

Register	R/W	Description	Reset Value	POR
OPX_IDE	R/W	OP1/2 Comparator Mode Interrupt Enable Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
INTEN	-	-	-	-	OP_CMP2IE	OP_CMP1IE	-

Bit number	Bit Mnemonic	Description
7	INTEN	Interrupt Request CPU Enable Control Bit 0: Disable interrupt request 1: Enable interrupt request
2	OP_CMP2IE	OP2 Comparator Mode Interrupt Enable Bit 0: An interrupt will not be generated when OP_CMP2IF is set 1: An interrupt will be generated when OP_CMP2IF is set
1	OP_CMP1IE	OP1 Comparator Mode Interrupt Enable Bit 0: An interrupt will not be generated when OP_CMP1IF is set 1: An interrupt will be generated when OP_CMP1IF is set
31~8 6~3 0	-	Reserved

11.7.3 OP0/1/2 Register Mapping

Register	Offset Address	R/W	Description	Reset Value	POR
OP0/1/2 Base Address: 0x4002_21B0					
OP0_CON	0x00	R/W	OP0 Control Register	0x0000_0000	0x0000_0000
OP1_CON	0x04	R/W	OP1 Control Register	0x0000_0000	0x0000_0000
OP2_CON	0x0C	R/W	OP2 Control Register	0x0000_0000	0x0000_0000
OPX_CFG	0x10	R/W	OP1/2 Configuration Register	0x0000_0000	0x0000_0000
OPX_STS	0x14	R/W	OP1/2 Comparator Mode Status Register	0x0000_0000	0x0000_0000
OPX_IDE	0x18	R/W	OP1/2 Comparator Mode Interrupt Enable Register	0x0000_0000	0x0000_0000

12 Analog Comparator CMP

12.1 Overview

The SC32F15G series features 4 built-in analog comparator (CMP), CMP0/1/2 shared the negative input port and CMP3 is completely independent.

CMP interrupt can wake up the STOP Mode. It can be used for applications such as alarm circuits, power supply voltage monitoring circuits, zero-crossing detection circuits, etc.

12.2 Clock Source

- The SC32F15G series CMP has only one clock source, which is derived from PCLK2

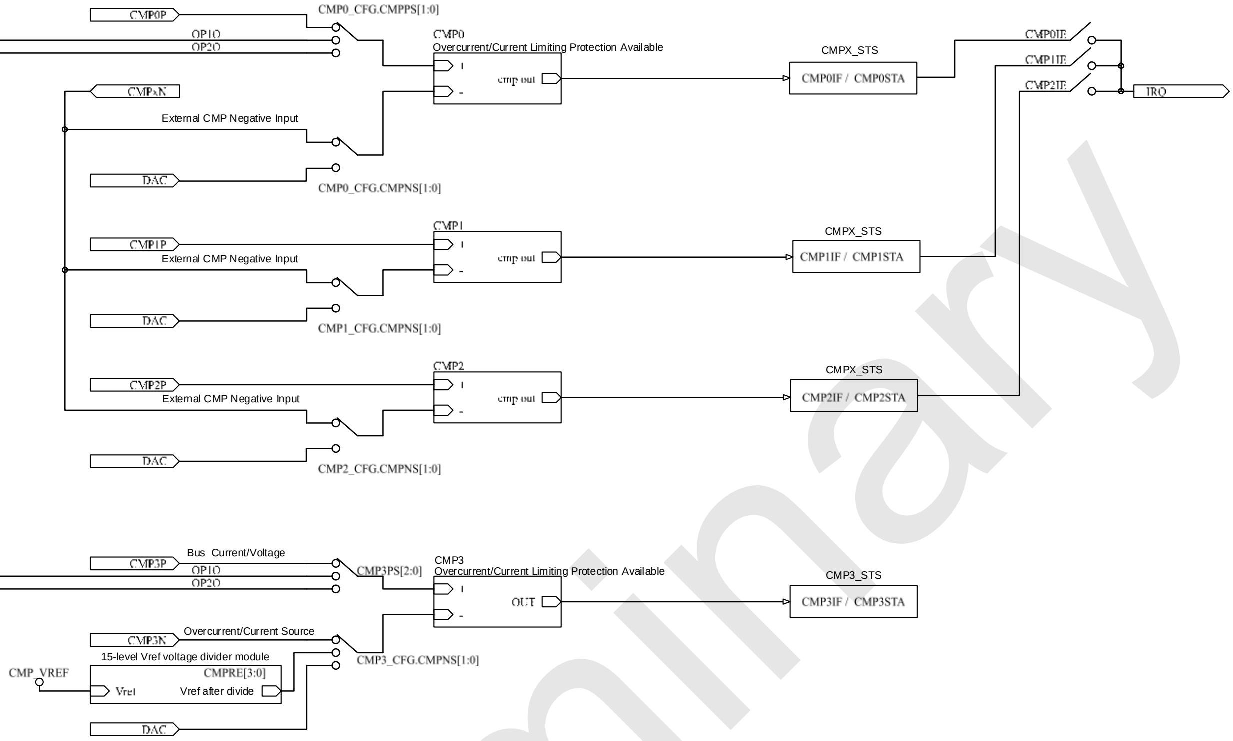
12.3 CMP0/1/2 Feature

- CMP0/1/2 have indenpendent external input port
- Positive input of CMP0 can select the output of OP1 and OP2
- Negative input of CMP0/1/2 selectable:
  - Shared input port CMPxN
  - Internal DAC output
- CMP0/1/2 interrupts can wake up the STOP mode
- Comparator voltage hysteresis: 0/5/10/20mV
- Response time: typical 50ns

12.4 CMP3 Feature

- Positive input of CMP3 selectable:
  - External input port CMP3N
  - Internal OP1 or OP2 output
- Negative input of CMP3 selectable:
  - External input port CMP3N
  - Internal DAC output
  - 15-level Vref voltage divider module output
- CMP3 interrupts can wake up the STOP mode
- Comparator voltage hysteresis: 0/5/10/20mV
- Response time: typical 50ns

12.5 Analog Comparator Structure Diagram



Analog Comparator Structure Diagram

12.6 CMP Interrupt

For CMP0~3, interrupts will be generated when meets the interrupt trigger condition. Separate interrupt enable bits can be used to enhance flexibility.

Interrupt Event	Interrupt Request Control Bit	Event Flag	Interrupt Enable Sub-Switch
CMP0 meets the interrupt trigger condition set by CMPIM[1:0]	CMPX_IDE->INTEN	CMP0IF	CMP0IE
CMP1 meets the interrupt trigger condition set by CMPIM[1:0]		CMP1IF	CMP1IE
CMP2 meets the interrupt trigger condition set by CMPIM[1:0]		CMP2IF	CMP2IE
CMP3 meets the interrupt trigger condition set by CMPIM[1:0]	CMP3_IDE ->INTEN	CMP3IF	CMP3IE

12.7 CMP Register

12.7.1 CMP0/1/2 Related Register

12.7.1.1 CMP0/1/2 Status Register (CMPX_STS)

Register	R/W	Description	Reset Value	POR
CMPX_STS	R/W	CMP0/1/2 Status Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	CMP2STA	CMP1STA	CMP0STA	CMP2IF	CMP1IF	CMP0IF

Bit number	Bit Mnemonic	Description
5	CMP2STA	CMP2 Output Status Bit 0: CMP2 positive terminal voltage is less than negative terminal voltage 1: CMP2 positive terminal voltage is greater than negative terminal voltage
4	CMP1STA	CMP1 Output Status Bit 0: CMP1 positive terminal voltage is less than negative terminal voltage 1: CMP1 positive terminal voltage is greater than negative terminal voltage
3	CMP0STA	CMP0 Output Status Bit 0: CMP0 positive terminal voltage is less than negative terminal voltage 1: CMP0 positive terminal voltage is greater than negative terminal voltage
2	CMP2IF	CMP2 Interrupt Flag This bit is set to 1 by hardware, and is cleared by writing to 1 through software. 0: CMP2 interrupt has not been inturrput 1: This bit will be set to 1 by hardware if CMP2 meets the interrupt trigger condition. And CMP2 interrupt will be generated if CMP2IE is enable
1	CMP1IF	CMP1 Interrupt Flag This bit is set to 1 by hardware, and is cleared by writing to 1 through software. 0: CMP1 interrupt has not been inturrput 1: This bit will be set to 1 by hardware if CMP1 meets the interrupt trigger condition. And CMP1 interrupt will be generated if CMP1IE is enable
0	CMP0IF	CMP0 Interrupt Flag This bit is set to 1 by hardware, and is cleared by writing to 1 through software. 0: CMP0 interrupt has not been inturrput 1: This bit will be set to 1 by hardware if CMP0 meets the interrupt trigger condition. And CMP0 interrupt will be generated if CMP0IE is enable
31~6	-	Reserved

12.7.1.2 CMP0/1/2 Control Register (CMPX_CON)

Register	R/W	Description	Reset Value	POR
CMPX_CON	R/W	CMP0/1/2 Control Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24

-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	-	-	HYS[1:0]	

Bit number	Bit Mnemonic	Description
1~0	HYS[1:0]	CMP0/1/2 Comparator voltage hysteresis Selection Bit 00: 0V 01: 5mV 10: 10mV 11: 20mV
31~2	-	Reserved

12.7.1.3 CMP0/1/2 Interrupt Enable Register (CMPX_IDE)

Register	R/W	Description	Reset Value	POR
CMPX_IDE	R/W	CMP0/1/2 Interrupt Enable Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
INTEN	-	-	-	-	CMP2IE	CMP1IE	CMP0IE

Bit number	Bit Mnemonic	Description
7	INTEN	Interrupt Request CPU Enable Control Bit 0: Disable interrupt request 1: Enable interrupt request
2	CMP2IE	CMP2 Interrupt Enable Bit 0: An interrupt will not be generated when OP_CMP2IF is set 1: An interrupt will be generated when OP_CMP2IF is set
1	CMP1IE	CMP1 Interrupt Enable Bit 0: An interrupt will not be generated when OP_CMP1IF is set 1: An interrupt will be generated when OP_CMP1IF is set
0	CMP0IE	CMP0 Interrupt Enable Bit 0: An interrupt will not be generated when OP_CMP0IF is set 1: An interrupt will be generated when OP_CMP0IF is set

Bit number	Bit Mnemonic	Description
31~8 6~3	-	Reserved

12.7.1.4 CMP0 Configuration Register (CMP0_CFG)

Register	R/W	Description	Reset Value	POR
CMP0_CFG	R/W	CMP0 Configuration Register	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
7	CMPEN	CMP0 Enable Bit 0: Disable CMP0 1: Enable CMP0
6~5	CMPIM[1:0]	CMP Interrupt Mode Selection Bit 00: No interrupt generated 01: Rising edge interrupt: Interrupt will be generated when IN+ transitions from being less than IN- to being greater than IN- 10: Falling edge interrupt: Interrupt will be generated when IN+ transitions from being greater than IN- to being less than IN- 11: Both edge interrupt: Interrupt will be generated when IN+ transitions from being less than IN- to being greater than IN- or from being greater than IN- to being less than IN-
3~2	CMPPS[1:0]	CMP Positive Terminal Input Selection: 00: Select CMP0P 01: Select OP1O 10: Select OP2O 11: Reserved
1~0	CMPNS[1:0]	CMP Negative Terminal Input Selection: 00: Select CMPxN 01: Select DAC output 10: Reserved 11: Reserved
31~8 4	-	Reserved

12.7.1.5 CMP1 Configuration Register (CMP1_CFG)

Register	R/W	Description	Reset Value	POR
CMP1_CFG	R/W	CMP1 Configuration Register	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
7	CMPEN	CMP1 Enable Bit 0: Disable CMP1 1: Enable CMP1
6~5	CMPIM[1:0]	CMP1 Interrupt Mode Selection Bit 00: No interrupt generated 01: Rising edge interrupt: Interrupt will be generated when IN+ transitions from being less than IN- to being greater than IN- 10: Falling edge interrupt: Interrupt will be generated when IN+ transitions from being greater than IN- to being less than IN- 11: Both edge interrupt: Interrupt will be generated when IN+ transitions from being less than IN- to being greater than IN- or from being greater than IN- to being less than IN-
1~0	CMPNS[1:0]	CMP1 Negative Terminal Input Selection: 00: Select CMPxN 01: Select DAC output 10: Reserved 11: Reserved
31~8 4~2	-	Reserved

12.7.1.6 CMP2 Configuration Register (CMP2_CFG)

Register	R/W	Description	Reset Value	POR
CMP2_CFG	R/W	CMP2 Configuration Register	0x0000_0000	0x0000_0000

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

15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
CMPEN	CMPIM[1:0]		-	-	-	CMPNS[1:0]	

Bit number	Bit Mnemonic	Description
7	CMPEN	CMP2 Enable Bit 0: Disable CMP2 1: Enable CMP2
6~5	CMPIM[1:0]	CMP2 Interrupt Mode Selection Bit 00: No interrupt generated 01: Rising edge interrupt: Interrupt will be generated when IN+ transitions from being less than IN- to being greater than IN- 10: Falling edge interrupt: Interrupt will be generated when IN+ transitions from being greater than IN- to being less than IN- 11: Both edge interrupt: Interrupt will be generated when IN+ transitions from being less than IN- to being greater than IN- or from being greater than IN- to being less than IN-
1~0	CMPNS[1:0]	CMP2 Negative Terminal Input Selection: 00: Select CMPxN 01: Select DAC output 10: Reserved 11: Reserved
31~8 4~2	-	Reserved

12.7.2 CMP0/1/2 Register Mapping

Register	Offset Address	R/W	Description	Reset Value	POR
CMP Base Address:0x4002_2150					
CMPX_STS	0x00	R/W	CMP0/1/2 Status Register	0x0000_0000	0x0000_0000
CMPX_CON	0x04	R/W	CMP0/1/2 Control Register	0x0000_0000	0x0000_0000
CMPX_IDE	0x08	R/W	CMP0/1/2 Interrupt Enable Register	0x0000_0000	0x0000_0000
CMP0_CFG	0x0C	R/W	CMP0 Configuration Register	0x0000_0000	0x0000_0000
CMP1_CFG	0x10	R/W	CMP1 Configuration Register	0x0000_0000	0x0000_0000
CMP2_CFG	0x14	R/W	CMP2 Configuration Register	0x0000_0000	0x0000_0000

12.8 CMP3 Register

12.8.1 CMP3 Related Register

12.8.1.1 CMP3 Status Register (CMP3_STS)

Register	R/W	Description	Reset Value	POR
CMP3_STS	R/W	CMP3 Status Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	-	-	CMP3STA	CMP3IF

Bit number	Bit Mnemonic	Description
1	CMP3STA	CMP3 Output Status Bit 0: CMP3 positive terminal voltage is less than negative terminal voltage 1: CMP3 positive terminal voltage is greater than negative terminal voltage
0	CMP3IF	CMP3 Interrupt Flag This bit is set to 1 by hardware, and is cleared by writing to 1 through software. 0: CMP3 interrupt has not been inturrput 1: This bit will be set to 1 by hardware if CMP3 meets the interrupt trigger condition. And CMP3 interrupt will be generated if CMP3IE is enable
31~2	-	Reserved

12.8.1.2 CMP3 Control Register (CMP3_CON)

Register	R/W	Description	Reset Value	POR
CMP3_CON	R/W	CMP3 Control Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-

7	6	5	4	3	2	1	0
REFSEL	-	-	-	-	-	HYS[1:0]	

Bit number	Bit Mnemonic	Description
7	REFSEL	CMP_VREF Reference Source Selection Bit 0: Reference source is VDD 1: Reference source is VREF
1~0	HYS[1:0]	CMP3 Comparator voltage hysteresis Selection Bit 00: 0V 01: 5mV 10: 10mV 11: 20mV
31~8 6~2	-	Reserved

12.8.1.3 CMP3 Interrupt Enable Register (CMP3_IDE)

Register	R/W	Description	Reset Value	POR
CMP3_IDE	R/W	CMP3 Interrupt Enable Register	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
7	INTEN	Interrupt Request CPU Enable Control Bit 0: Disable interrupt request 1: Enable interrupt request
31~8 6~0	-	Reserved

12.8.1.4 CMP3 Configuration Register (CMP3_CFG)

Register	R/W	Description	Reset Value	POR
CMP3_CFG	R/W	CMP3 Configuration Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-

23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	CMPRF[3:0]			
7	6	5	4	3	2	1	0
CMPEN	CMPIM[1:0]		-	CMPPS[1:0]		CMPNS[1:0]	

Bit number	Bit Mnemonic	Description
11~8	CMPRF[3:0]	Comparator Negative Input Voltage Selection Bit The settings for the negative input comparison voltage of the analog comparator are as follows:: 0000: 1/16 CMP_VREF 0001: 1/16 CMP_VREF 0010: 2/16 CMP_VREF 0011: 3/16 CMP_VREF 0100: 4/16 CMP_VREF 0101: 5/16 CMP_VREF 0110: 6/16 CMP_VREF 0111: 7/16 CMP_VREF 1000: 8/16 CMP_VREF 1001: 9/16 CMP_VREF 1010: 10/16 CMP_VREF 1011: 11/16 CMP_VREF 1100: 12/16 CMP_VREF 1101: 13/16 CMP_VREF 1110: 14/16 CMP_VREF 1111: 15/16 CMP_VREF
7	CMPEN	CMP3 Enable Bit 0: Disable CMP3 1: Enable CMP3
6~5	CMPIM[1:0]	CMP3 Interrupt Mode Selection Bit 00: No interrupt generated 01: Rising edge interrupt: Interrupt will be generated when IN+ transitions from being less than IN- to being greater than IN- 10: Falling edge interrupt: Interrupt will be generated when IN+ transitions from being greater than IN- to being less than IN- 11: Both edge interrupt: Interrupt will be generated when IN+ transitions from being less than IN- to being greater than IN- or from being greater than IN- to being less than IN-
3~2	CMPPS[1:0]	CMP3 Positive Terminal Input Selection: 00: Select CMP3P 01: Select OP1O 10: Select OP2O 11: Reserved

Bit number	Bit Mnemonic	Description
1~0	CMPNS[1:0]	CMP3 Negative Terminal Input Selection: 00: Select CMP3N 01: Select DAC output 10: Value set by CMPRF[3:0] 11: Reserved
31~12 4	-	Reserved

12.8.2 CMP3 Register Mapping

Register	Offset Address	R/W	Description	Reset Value	POR
CMP Base Address:0x4002_2170					
CMP3_STS	0x00	R/W	CMP3 Status Register	0x0000_0000	0x0000_0000
CMP3_CON	0x04	R/W	CMP3 Control Register	0x0000_0000	0x0000_0000
CMP3_IDE	0x08	R/W	CMP3 Interrupt Enable Register	0x0000_0000	0x0000_0000
CMP3_CFG	0x0C	R/W	CMP3 Configuration Register	0x0000_0000	0x0000_0000

13 Quadrature Encoder Pulse (QEP) Module

13.1 Overview

The SC32F15G series features 2 QEP modules, they can be connected to linear or rotary incremental encoders to obtain machine position, direction, and speed information.Users can configure the QSRC[1:0] bit in QEPn_CON (n=0~1) register to select the counting method.

The SC32F15G series provides 3 counting methods: Quadrature Counting, Direction Counting and Dual Pulse Counting.

13.2 Feature

- Each QEP module (n = 0~1) provides three input signal pins: QEPnA, QEPnB and QEPnI
  - QEPnA and QEPnB can be swapped in direction
  - The polarity of QEPnA and QEPnB can be individually configured
  - Provides a configurable digital filter with a maximum division factor of 128 for QEPnA, QEPnB, and QEPnI signals
- In Direction Counting and Dual Pulse Counting modes, counting can be configured for:
  - Rising edge
  - Falling edge
  - Both edges (rising and falling)
- Position Counter Reset Modes:
  - Index Event Reset
  - overflow Reset(When PCNT=PMAX)

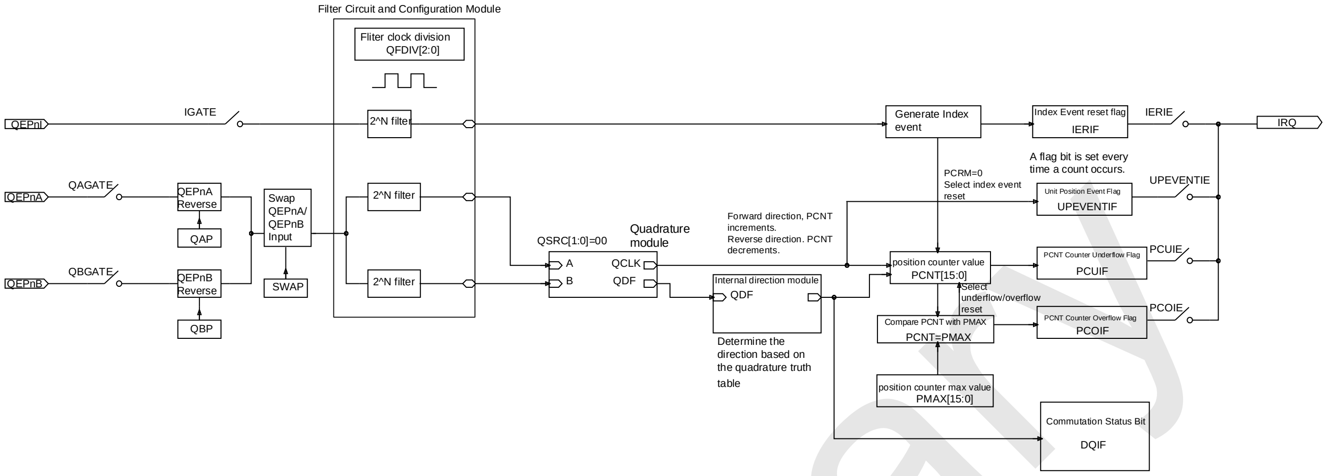
13.3 Quadrature Counting

When the position count source is selected as quadrature counting, i.e., QSRC[1:0] = 00, quadrature counting requires a 90° phase difference between the two input signals QEPnA and QEPnB. The QEP module generates a quadrature pulse QCLK based on the rising/falling edges of QEPnA and QEPnB.

If the waveform of QEPnA is leading, it indicates the forward direction, and the position counter PCNT increments. The direction flag QDF is set to 1, and PCNT + 1 for each pulse of QCLK.

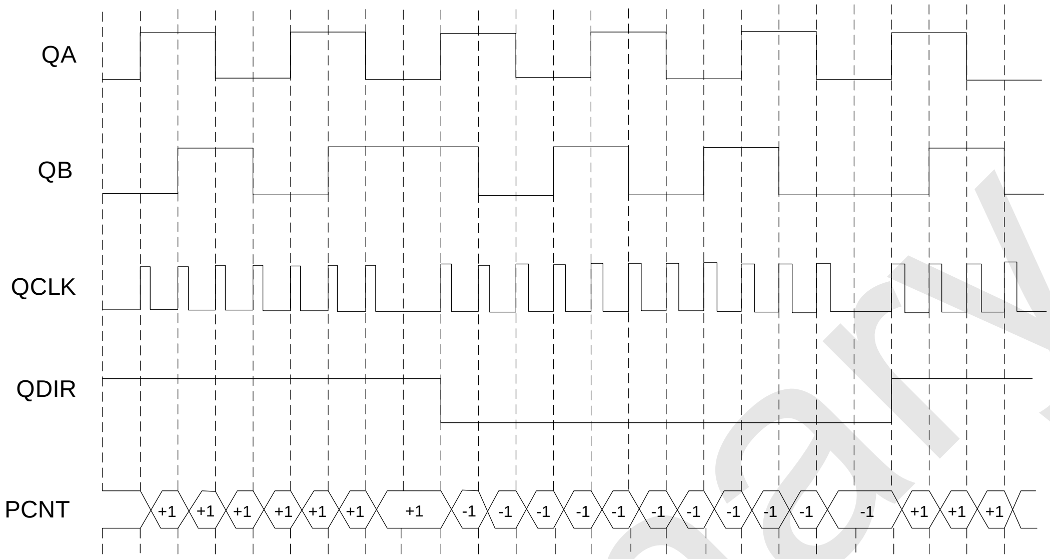
If the waveform of QEPnB leads QEPnA, it indicates the reverse direction, and the position counter PCNT decrements. The direction flag QDF is set to 0, and PCNT - 1 for each pulse of QCLK.

13.3.1 Quadrature Mode Structure Diagram



13.3.2 Quadrature Truth Table and Waveform

Last Edge	Current Edge	DQIF	PCNT
QEPnA↑	QEPnB↑	1	Increase
	QEPnB↓	0	Decrease
	QEPnA↓	Reverse	Increase or decrease
QEPnA↓	QEPnB↓	1	Increase
	QEPnB↑	0	Decrease
	QEPnA↑	Reverse	Increase or decrease
QEPnB↑	QEPnA↑	1	Increase
	QEPnA↓	0	Decrease
	QEPnB↓	Reverse	Increase or decrease
QEPnB↓	QEPnA↓	1	Increase
	QEPnA↑	0	Decrease
	QEPnB↑	Reverse	Increase or decrease



In the waveform diagram, QEPnA and QEPnB are two input signals with a phase difference of 90°. The QCLK signal is derived from the edges of the A and B phase signals. The DQIF direction signal can determine the direction based on the truth table of QEPnA and QEPnB. Together, the QCLK signal and the DQIF signal determine the counting direction of the position counter (QEPn_PCNT): when in forward direction, the counter increments by +1 for each QCLK; when in reverse direction, the counter decrements by -1 for each QCLK.

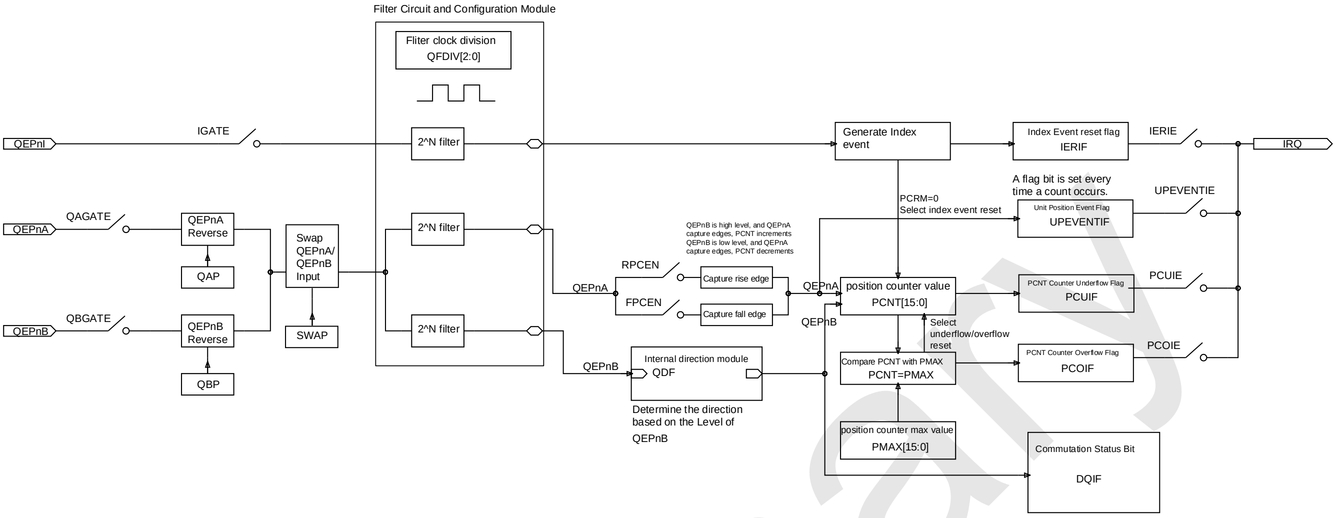
13.4 Direction Counting

When the position count source is selected as directional counting, i.e., QSRC[1:0] = 01, directional counting requires that the input signal QEPnA serves as the counting pulse, while the input signal QEPnB indicates the direction. Dual-edge counting can be achieved by enabling rising-edge counting (RPCEN), falling-edge counting (FPCEN), or both simultaneously.

When the input signal QEPnB is high, it indicates the forward direction, and the direction flag QDF is set to 1. In this case, if a counting pulse corresponding to QEPnA is detected, the position counter PCNT increments by 1.

When the input signal QEPnB is low, it indicates the reverse direction, and the direction flag QDF is set to 0. In this case, if a counting pulse corresponding to QEPnA is detected, the position counter PCNT decrements by 1.

13.4.1 Direction Counting Mode Structure Diagram



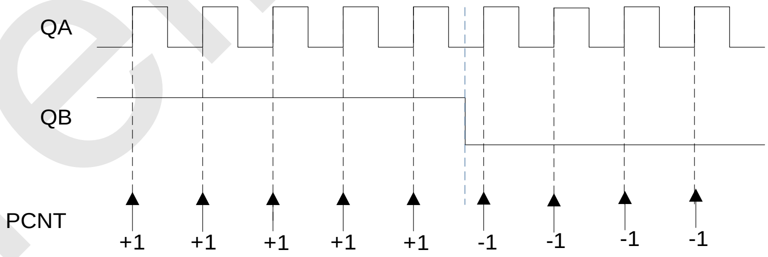
13.4.2 Direction Counting Mode Waveform

In the directional counting mode, the direction is determined based on the logic level of QEPnB.

If rising-edge counting is enabled, when QEPnB is high (indicating the CW direction), the position counter (QEPn_PCNT) increments by 1 on each rising edge of the QEPnA signal.

When QEPnB is low (indicating the CCW direction), the position counter (QEPn_PCNT) decrements by 1 on each rising edge of the QEPnA signal.

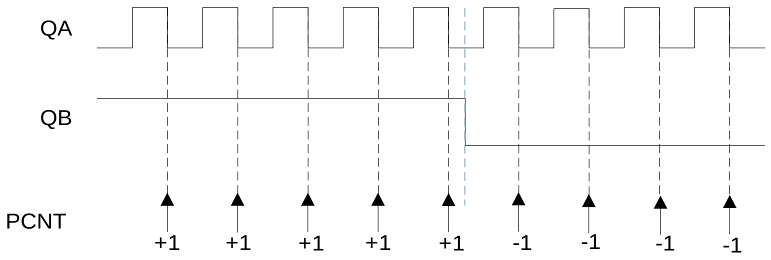
The waveform diagram is as follows:



If falling-edge counting is enabled, when QEPnB is high (indicating the CW direction), the position counter (QEPn_PCNT) increments by 1 on each falling edge of the QEPnA signal.

When QEPnB is low (indicating the CCW direction), the position counter (QEPn_PCNT) decrements by 1 on each falling edge of the QEPnA signal.

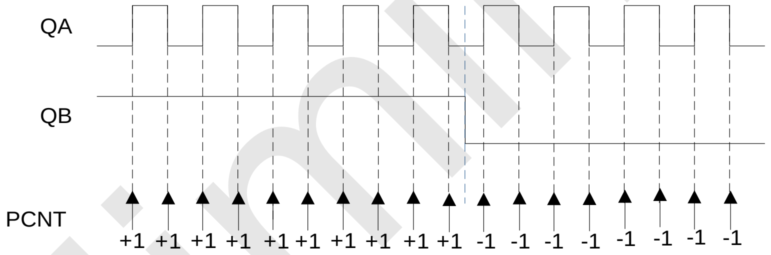
The waveform diagram is as follows:



If dual-edge counting is enabled, when QEPnB is high (indicating the CW direction), the position counter (QEPn_PCNT) increments by 1 on each rising edge and falling edge of the QEPnA signal.

When QEPnB is low (indicating the CCW direction), the position counter (QEPn_PCNT) decrements by 1 on each rising edge and falling edge of the QEPnA signal.

The waveform diagram is as follows:



13.5 Dual Pulse Counting

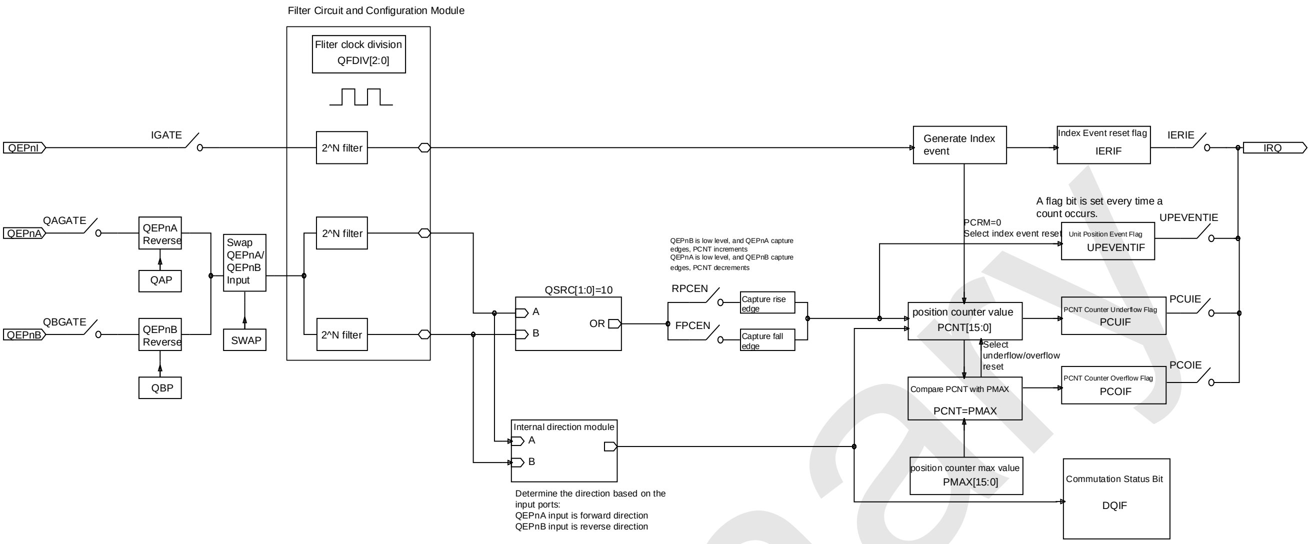
When the position count source is selected as dual-pulse counting, i.e., QSRC[1:0] = 10, the counting direction is determined based on the logic levels of the input signals QEPnA and QEPnB:

If QEPnA has a pulse and QEPnB is low, it indicates the forward direction. The direction flag QDF is set to 1, and the position counter PCNT increments by 1 for each pulse.

If QEPnB has a pulse and QEPnA is low, it indicates the reverse direction. The direction flag QDF is set to 0, and the position counter PCNT decrements by 1 for each pulse of QEPnB.

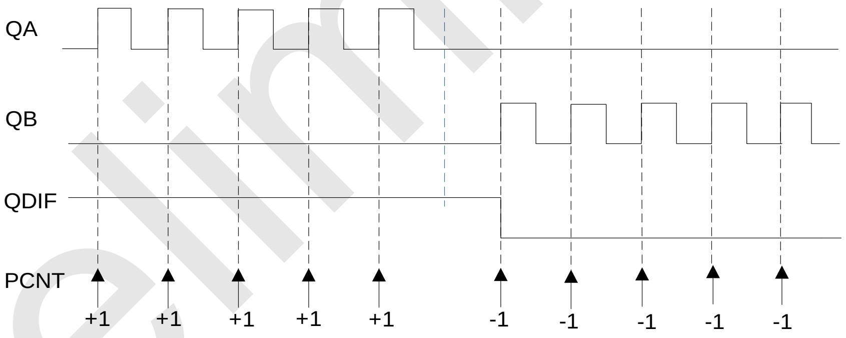
Dual-edge counting can be enabled by configuring rising-edge counting (RPCEN), falling-edge counting (FPCEN), or both simultaneously.

13.5.1 Dual Pulse Counting Mode Structure Diagram

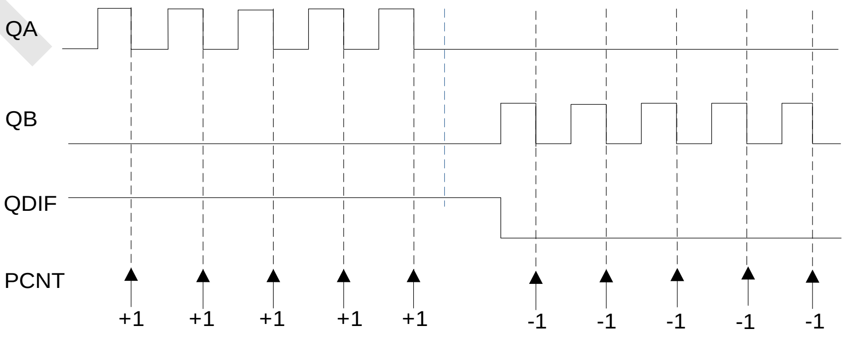


13.5.2 Dual Pulse Counting Mode Waveform

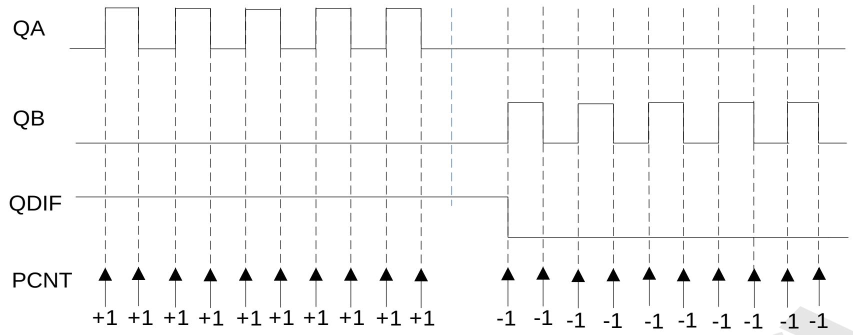
When rising-edge counting is enabled, the waveform diagram is as follows:



When falling-edge counting is enabled, the waveform diagram is as follows:



When both rising-edge and falling-edge counting are enabled, the waveform diagram is as follows:



13.6 Reset Method

When index event reset is selected, i.e., PCRM = 0, the position counter PCNT is reset upon detection of an index signal on the index input pin. When rotating forward, PCNT is reset to 0; when rotating in reverse, PCNT is reset to the PMAX value. In this case, the index event reset flag IERIF is set. If IERIE is enabled, an interrupt can also be triggered simultaneously.

When maximum value reset is selected, i.e., PCRM = 1, the position counter PCNT behaves as follows:

When PCNT counts upward and reaches the PMAX value, the overflow flag of the PCNT counter is set, and PCNT is reset to 0.

When PCNT counts downward and reaches 0, the underflow flag of the PCNT counter is set, and PCNT is reset to the PMAX value.

13.7 Unit Positon Event

To facilitate the measurement of speed information, the SC32F15G QEP module provides a unit position event interrupt. In different counting modes, the QEP module enters a unit position event interrupt each time it captures a counting edge. This makes it convenient to use an external timer to capture the time interval between unit position events, thereby measuring speed information using the following formula:

$$v(k) \approx \frac{X}{t(k) - t(k-1)} = \frac{X}{\Delta T}$$

where X represents the count value collected between t(k) and t(k−1).

13.8 QEP Register

13.8.1 QEP Related Register

13.8.1.1 QEPn Control Register QEPn_CON

Register	R/W	Description	Reset Value	POR
QEPn_CON	R/W	QEPn Control Register (n=0~1)	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	SPOS	-	-	-	-	QIP
15	14	13	12	11	10	9	8
SWAP	PCRM	QBP	QAP	-	QFDIV[2:0]		
7	6	5	4	3	2	1	0
QEPEN	IGATE	QBGATE	QAGATE	FPCEN	RPCEN	QSRC[1:0]	

<table>
<tr><th>Bit number</th><th>Bit Mnemonic</th><th>Description</th></tr>
<tr><td>21</td><td>SPOS</td><td>QEP1 Port Mapping Control Bit@QEP1_CON
<table>
<tr><th>Port
SPOS value</th><th>QEP1A</th><th>QEP1B</th><th>QEP1I</th></tr>
<tr><td>SPOS=0</td><td>PA0</td><td>PA1</td><td>PA2</td></tr>
<tr><td>SPOS=1</td><td>PC4</td><td>PC5</td><td>PC6</td></tr>
</table></td></tr>
<tr><td>16</td><td>QIP</td><td>QEPnI Input Polarity
0: Forward direction
1: Reverse direction</td></tr>
<tr><td>15</td><td>SWAP</td><td>QEPnA and QEPnB Input Swap Bit
0: QEPnA and QEPnB inputs are not swapped.
1: QEPnA and QEPnB inputs are swapped before filtering.</td></tr>
<tr><td>14</td><td>PCRM</td><td>Index Event Reset Selection Bit
0: Enable index event. When an index event occurs, the position counter is reset. For positive direction reset, PCNT = 0; for negative direction reset, PCNT = PMAX.
1: Mask the index event.</td></tr>
<tr><td>13</td><td>QBP</td><td>QEPnB Input Polarity
0: Forward direction
1: Reverse direction</td></tr>
<tr><td>12</td><td>QAP</td><td>QEPnA Input Polarity
0: Forward direction
1: Reverse direction</td></tr>
<tr><td>10~8</td><td>QFDIV[2:0]</td><td>QEPnA, QEPnB, QEPnI Digital Input Filter Width Selection Bits
111 = 1:128 clock division
110 = 1:64 clock division
101 = 1:32 clock division
100 = 1:16 clock division
011 = 1:8 clock division
010 = 1:4 clock division
001 = 1:2 clock division
000 = 1:1 clock division</td></tr>
<tr><td>7</td><td>QEPEN</td><td>QEP Module Enable Bit
0: Disable
1: Enable</td></tr>
</table>

Bit number	Bit Mnemonic	Description
6	IGATE	QEPnI Pin Enable Bit 0: Disable QEPnI pin 1: Enable QEPnI pin
5	QBGATE	QEPnB Pin Enable Bit 0: Disable the QEPnB pin. 1: Configure the QEPnB pin as a QEP input.
4	QAGATE	QEPnA Pin Enable Bit 0: Disable the QEPnA pin. 1: Configure the QEPnA pin as a QEP input.
3	FPCEN	Falling-Edge Counting Enable 0: Disabled 1: Enabled
2	RPCEN	Rising-Edge Counting Enable 0: Disabled 1: Enabled
1~0	QSRC[1:0]	Position Count Source Selection 00: Quadrature Counting Mode: The direction is determined based on the truth table of QEPnA and QEPnB. The position counter (QPOSCNT) increments by 1 for each QCLK in the positive direction and decrements by 1 for each QCLK in the negative direction. 01: Directional Counting Mode: QEPnA provides the clock for the position counter, while QEPnB provides the direction information. 10: Dual-Pulse Counting Mode: In this mode, the periodic count directly represents the speed signal. 11: Reserved
31~22 20~17 11	-	Reserved

13.8.1.2 QEPn Position Counter Value Register QEPn_PCNT

Register	R/W	Description	Reset Value	POR
QEPn_PCNT	R/W	QEPn Position Counter Value Registe(n=0~1)	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
15~0	PCNT[15:0]	Position counter value
31~16	-	Reserved

13.8.1.3 QEPn Position Counter Maximum Value Register QEPn_PMAX

Register	R/W	Description	Reset Value	POR
QEPn_PMAX	R/W	QEPn Position Counter Maximum Value Register(n=0~1)	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
15~0	PMAX[15:0]	Position Counter Maximum Value When PCNT counts upward and reaches the PMAX value, the overflow flag of the PCNT counter is set, and PCNT is reset to 0. When PCNT counts downward and reaches 0, the underflow flag of the PCNT counter is set, and PCNT is reset to the PMAX value.
31~16	-	Reserved

13.8.1.4 QEPn Status Register QEPn_STS

Register	R/W	Description	Reset Value	POR
QEPn_STS	R/W	QEPn Status Register(n=0~1)	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
4	DQIF	Quadrature Direction Flag 0: Counterclockwise rotation (or reverse motion) 1: Clockwise rotation (or forward motion)
3	UPEVNTIF	Unit Position Event Flag 0: No unit position event detected 1: Unit position event detected, write 1 to clear.
2	IERIF	Index Event Reset Interrupt Flag 0: No interrupt generated 1: Set after resetting PCNT
1	PCOIF	Position Counter Overflow Interrupt Flag 0: No interrupt generated 1: Set when the position counter overflows upward
0	PCUIF	Position Counter Underflow Interrupt Flag 0: No interrupt generated 1: Set when the position counter underflows downward
31~5	-	Reserved

13.8.1.5 QEPn Interrupt Enable Register QEPn_IDE

Register	R/W	Description	Reset Value	POR
QEPn_IDE	R/W	QEPn Interrupt Enable Register(n=0~1)	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
7	INTEN	Interrupt Request CPU Enable Control Bit 0: Disable interrupt request 1: Enable interrupt request
3	UPEVNTIE	Unit Position Event Interrupt Enable Bit 0: Disable interrupt 1: Enable interrupt
2	IERIE	Index Event Reset Interrupt Enable Bit 0: Disable interrupt 1: Enable interrupt
1	PCOIE	Position Counter Overflow Interrupt Enable Bit

Bit number	Bit Mnemonic	Description
		0: Disable interrupt 1: Enable interrupt
0	PCUIE	Position Counter Underflow Interrupt Enable Bit 0: Disable interrupt 1: Enable interrupt
31~8 6~4	-	Reserved

13.8.2 QEP Register Mapping

Register	Offset Address	R/W	Description	Reset Value	POR
QEP0 Base Address: 0x4002_2020					
QEP0_CON	0x00	R/W	QEP0 Control Register	0x0000_0000	0x0000_0000
QEP0_PCNT	0x04	R/W	QEP0 Position Counter Value Register	0x0000_0000	0x0000_0000
QEP0_PMAX	0x08	R/W	QEP0 Position Counter Maximum Value Register	0x0000_0000	0x0000_0000
QEP0_STS	0x0C	R/W	QEP0 Status Register	0x0000_0000	0x0000_0000
QEP0_IDE	0x10	R/W	QEP0 Interrupt Enable	0x0000_0000	0x0000_0000

Register	Offset Address	R/W	Description	Reset Value	POR
QEP1 Base Address: 0x4002_2040					
QEP1_CON	0x00	R/W	QEP1 Control Register	0x0000_0000	0x0000_0000
QEP1_PCNT	0x04	R/W	QEP1 Position Counter Value Register	0x0000_0000	0x0000_0000
QEP1_PMAX	0x08	R/W	QEP1 Position Counter Maximum Value Register	0x0000_0000	0x0000_0000
QEP1_STS	0x0C	R/W	QEP1 Status Register	0x0000_0000	0x0000_0000
QEP1_IDE	0x10	R/W	QEP1 Interrupt Enable	0x0000_0000	0x0000_0000

14 16-bit Timers (Timer0~Timer3)

14.1 Clock Source

- In timer mode/PWM output mode, the TIM clock source is derived from PCLK
- In counter mode, the Tn pin serves as the counting source input

14.2 Feature

- Support 8-stage TIM clock pre-scaling
- 4 independent 16-bit auto-reload counters: Timer0 to Timer7
- 16-bit incremental, decremental, and increment/decrement auto-reload counters
- Support rising/falling edge capture, enabling PWM duty and period capture
- Overflow and capture events of TIM1/2 can generate DMA requests
- All timer pins(Tn and TnEX) can be remapped

14.3 Counting method

14.3.1 Counting Method in Timer Mode

- Upward Counting: Counts from the set value upwards to overflow at 0xFFFF
- Downward Counting: Counts from 0xFFFF downwards to the set value

14.3.2 Counting Method in PWM Mode

Only upward counting mode is available in PWM output mode: The counter starts from 0 and counts up until PDT, then PWM output waveform will switch between the high and low levels. The counting will then continue up to RLD, causing an overflow and the counter reset to 0.

The formula of TPWM is shown as follows:

$$T_{PWM} = \frac{RLD[15:0] + 1}{PCLK}$$

The formula of duty is shown as follows:

$$duty = \frac{PDT[15:0]}{RLD[15:0] + 1}$$

14.4 Timer Signal Port

- Tn, n=0~3
    - Clock input/output
    - Both rising and falling edges can be captured
- TnEX, n=0~3
    - In reload mode, the external event input (falling edge) on the TnEX pin is used for reload

enable/disable control
  - In capture mode, when FSEL = 1, it serves as a falling edge capture signal input. Detection of a falling edge on the TnEX pin generates a capture, sets EXIF, and captures the value of the TnCNT register into the FCAP register
- TnPWM, n=0~3
  - TIM0~3 can provide PWM with independently adjustable duty cycle through the Tn port: TnPWMA
  - TIM0~3 can provide PWM with independently adjustable duty cycle through the TnEX port: TnPWMB
  - Optional clock source follows TIM

Note: TIM's PWM capture function and PWM output function cannot be enabled simultaneously

14.5 Interrupts and Corresponding Flags for TIM:

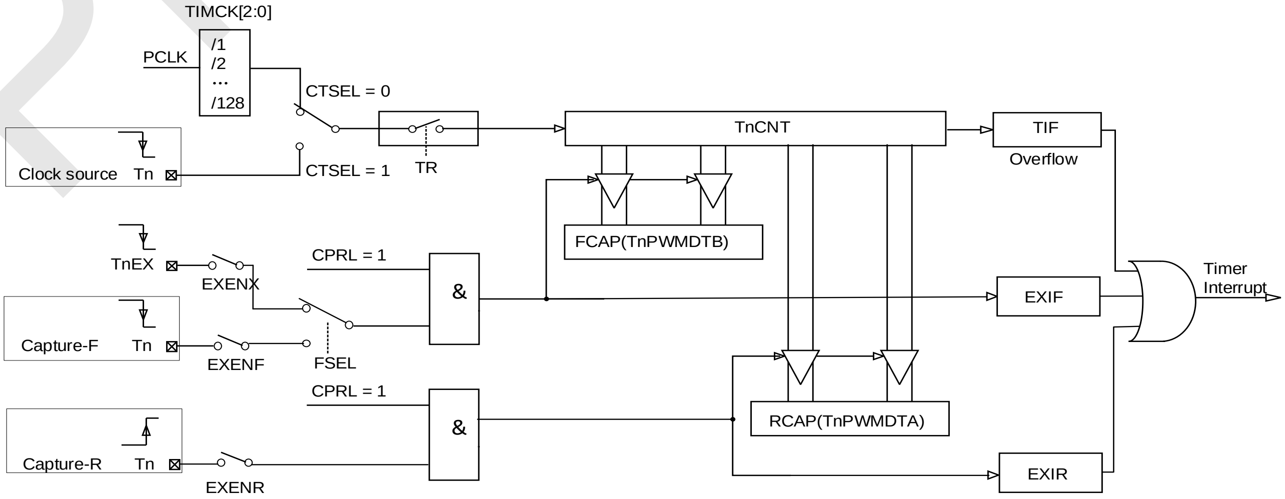
- Overflow/underflow of the counter share the interrupt flag TIF
- Capture status flags:
  - EXIF: Flag indicating detection of a falling edge on the external event input
  - EXIR: Flag indicating detection of a rising edge on the external event input
- Interrupt and priority configuration control bits are merged into the NVIC module

14.6 Timer Operating Mode

- Mode 0: 16-bit capture mode, capable of PWM edge capture on both rising and falling edges
- Mode 1: 16-bit auto-reload timer/counter mode
- Mode 3: Programmable clock output mode
- Mode 4: PWM output mode

14.6.1 Operating Mode 0: 16-bit Capture

14.6.1.1 Dual-Edge Capture Structure Diagram



Dual-Edge Capture Mode

TnEX is a PWM capture port. The internal measurement is divided into separate channels for sampling the rising and falling edges:

- A dedicated 16-bit falling-edge capture counter register — multiplexed with the PWM duty register.
- A dedicated 16-bit rising-edge capture counter register — multiplexed with the PWM duty register.
- Capture flag bits:
  - Separate enable bits for rising-edge and falling-edge capture.
  - Separate flags for rising-edge and falling-edge capture.

14.6.2 Operating Mode 1: 16-bit Auto-Reload Timer

In the 16-bit auto-reload mode, the timer can be selected to count either upwards or downwards. This functionality is achieved by setting the DEC bit in TIMn_CON (n = 0~3) to 1, and then selecting the counting direction through the TnEX pin. After system reset, the default value of the DEC bit is 0, and Timer n counts upwards by default. When DEC is set to 1, whether Timer n counts upwards or downwards depends on the level of the TnEX pin.

When DEC = 0, two options are selected through the EXENX bit in TIMn_CON:

If EXENX = 0, when TIMn_CNT increments to 0xFFFF, the timer overflow flag TIF is set, and the timer automatically loads the 16-bit value written by the user software in the reload register TIMn_RLD into the TIMn_CNT register.

If EXENX = 1, an overflow or a falling edge on the external input TnEX can trigger a 16-bit reload. When a falling edge occurs on TnEX, the EXIF flag is set. If TIE is enabled, both TIF and EXIF flags can generate an interrupt.

When DEC = 1, the TnEX pin controls the counting direction, and the EXENX control is invalid.

If TnEX = 1, TIMn counts upwards. When TIMn_CNT increments to 0xFFFF, the timer overflow flag TIF is set, and the timer automatically loads the 16-bit value written by the user software in the reload register TIMn_RLD into the TIMn_CNT register.

If TnEX = 0, TIMn counts downwards. When the value of TIMn_CNT decrements from 0xFFFF to the value equal to TIMn_RLD, the timer overflows, the timer overflow flag TIF is set, and 0xFFFF is reloaded into TIMn_CNT.

In this operating mode, regardless of whether Timer n overflows or not, EXFIF does not act as an interrupt flag.

14.6.3 Operating Mode 3: Programmable Clock Output

In this way, TIMn(n=0~3) can be programmed to output a 50% duty cycle clock cycle: when CTSEL = 0; TnOE = 1, TIMn is enabled as a clock generator

In this way, the clock frequency output by Tn is:

$$f_{OUT} = \frac{f_{TIM}}{(65536 - TIMn_{RLD}) * 4}$$

14.6.4 Operating Mode 4: PWM Output

- PWM Duty Cycle Change Characteristics

When the value of PDTx[15:0] is changed, the duty cycle does not change immediately. Instead, it waits until the current period ends and then changes in the next period.

- PWM Period Change Characteristics

This is achieved by modifying the values in the period setting registers [RCAPXL / RCAPXH]. Define the current period count value as Tn. When writing to the period register, the value counted by the timer is Tm, and the new period count value to be updated is Tx. Then:

- If Tm ≤ Tx: The period changes in real-time according to Tx.
- If Tm > Tx: The period change is divided into two stages. In the first stage, after writing to the period register, the period counter continues to count up from the current value until it overflows and resets to zero. In the second stage, the period changes according to Tx.

14.7 TIM Interrupt

In timed or counting mode, when the count value of the CNT counter reaches the TIMn count value, TIF (Timer Interrupt Flag) will be set, and an interrupt will be generated if TIMn_IDE.INTEN = 1.

In external event input mode, when a valid edge transition is detected, EXIR/EXIF will be set, and an interrupt will be generated if TIMn_IDE.INTEN = 1.

Interrupt Event	Event Flag	Interrupt Enable Control Bit	Interrupt Enable Sub-Switch
Timer overflow	TIF	TIMn_IDE->INTEN (n=0~3)	TIMn_IDE->TIE
External event input rising edge interrupt	EXIR		TIMn_IDE->EXRIE
External event input falling edge interrupt	EXIF		TIMn_IDE->EXFIE

14.8 TIM Register

14.8.1 TIM Related Register

14.8.1.1 Timer Control Register (TIMn_CON)

Register	R/W	Description	Reset Value	POR
TIMn_CON (n=0~3)	R/W	Timer Control Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	SPOS[1:0]		-	-	-	-	-
15	14	13	12	11	10	9	8
TXOE	EPWMNA	EPWMNB	INVNA	INVNB	TIMCK[2:0]		
7	6	5	4	3	2	1	0
TR	DEC	EXENX	FSEL	EXENF	EXENR	CTSEL	CPRL

<table>
<tr><th>Bit number</th><th>Bit Mnemonic</th><th>Description</th></tr>
<tr><td>22~21</td><td>SPOS[1:0]</td><td>
● TIM0 Port Mapping Control Bit @TIM0_CON
<table>
<tr><th>Port
SPOS Value</th><th>T0CAP</th><th>T0EX</th></tr>
<tr><td>SPOS[1:0]=00</td><td>PA4</td><td>PA5</td></tr>
<tr><td>SPOS[1:0]=01</td><td>PB0</td><td>PB1</td></tr>
<tr><td>SPOS[1:0]=10</td><td>PC6</td><td>PC7</td></tr>
</table>
● TIM1 Port Mapping Control Bit @TIM1_CON
<table>
<tr><th>Port
SPOS Value</th><th>T1CAP</th><th>T1EX</th></tr>
<tr><td>SPOS[1:0]=00</td><td>PC11</td><td>PC12</td></tr>
<tr><td>SPOS[1:0]=01</td><td>PB2</td><td>PB3</td></tr>
</table>
● TIM2 Port Mapping Control Bit @TIM2_CON
<table>
<tr><th>Port
SPOS Value</th><th>T2CAP</th><th>T2EX</th></tr>
<tr><td>SPOS[1:0]=00</td><td>PC14</td><td>PC13</td></tr>
<tr><td>SPOS[1:0]=01</td><td>PB4</td><td>PB5</td></tr>
</table>
● TIM3 Port Mapping Control Bit @TIM3_CON
<table>
<tr><th>Port
SPOS Value</th><th>T3CAP</th><th>T3EX</th></tr>
<tr><td>SPOS[1:0]=00</td><td>PC2</td><td>PC1</td></tr>
<tr><td>SPOS[1:0]=01</td><td>PB6</td><td>PB7</td></tr>
</table>
</td></tr>
<tr><td>15</td><td>TXOE</td><td>Tn Pin Signal Direction Control Bit
0: Tn is used as clock input or I/O
1: Tn is used as programmable clock output</td></tr>
<tr><td>14</td><td>EPWMNA</td><td>Tn_PWMA Pin PWM Waveform Output Enable Bit
0: Disable
1: Enable</td></tr>
<tr><td>13</td><td>EPWMNB</td><td>Tn_PWMB Pin PWM Waveform Output Enable Bit
0: Disable
1: Enable</td></tr>
<tr><td>12</td><td>INVNA</td><td>TPWMnA Waveform Output Inversion Control Bit
0: Normal
1: Waveform Output Inverted</td></tr>
<tr><td>11</td><td>INVNB</td><td>TPWMnB Waveform Output Inversion Control Bit
0: Normal
1: Waveform Output Inverted</td></tr>
<tr><td>10~8</td><td>TIMCK[2:0]</td><td>TIM Clock Frequency Prescaler Bit
This clock frequency of Timer “f_{TIM}” is:
000: f_{SOURCE}/1
001: f_{SOURCE}/2
010: f_{SOURCE}/4
011: f_{SOURCE}/8
100: f_{SOURCE}/16</td></tr>
</table>

Bit number	Bit Mnemonic	Description
		101: f_{SOURCE}/32 110: f_{SOURCE}/64 111: f_{SOURCE}/128 The clock corresponding to f_{SOURCE} may be either PCLK or the input Tn.
7	TR	TIMn Start/Stop Control Bit 0: Stop the TIMn/TPWMn counter 1: Start the TIMn/TPWMn counter
6	DEC	Increment/Decrement Direction Control Bit 0: TIMn is an incrementing timer/counter 1: TIMn is an incrementing/decrementing timer/counter, and TnEX is used to select the counting direction
5	EXENX	TnEX Setting Bit, n=0~3 The function of this bit varies in different modes: ● Reload mode (CPRL = 0): This bit controls the external event input (falling edge) on the TnEX pin for reload enable/disable control: 0: Ignore events on the TnEX pin. 1: Generate a reload when detect a falling edge on the TnEX pin. ● Capture mode (CPRL = 1): This bit serves as the TnEX falling edge signal capture selection bit: 0: Ignore events on the TnEX pin. 1: When FSEL = 1, generate a capture when detect a falling edge on the TnEX pin, set EXIF, and capture the value of the TnCNT register into the register FCAP.
4	FSEL	Falling Edge Signal Selection Bit This bit is only valid in capture mode (CPRL=1): 0: Generate a capture when detect a falling edge on the Tn pin,. Ignore events on the TnEX pin. 1: Generate a capture when detect a falling edge on the TnEX pin. Ignore events on the Tn pin.
3	EXENF	Falling Edge Signal Capture Enable Bit: 0: Ignore events on the Tn pin 1: Generate a capture when detect a falling edge on the Tn pin, set EXIF, and capture the value of the TnCNT register into the register FCAP
2	EXENR	Rising Edge Signal Capture Enable Bit: 0: Ignore events on the Tn pin 1: Generate a capture when detect a rising edge on the Tn pin, set EXIR, and capture the value of the TnCNT register into the register RCAP

Bit number	Bit Mnemonic	Description
1	CTSEL	Timer/Counter Selection Bit 0: Timer 1: Counter
0	CPRL	Capture/Reload Function Selection Bit 0: Reload function 1: Capture function
31~23 20~16	-	Reserved

14.8.1.2 Timer Count Value Register (TIMn_CNT)

Register	R/W	Description	Reset Value	POR
TIMn_CNT (n=0~3)	R/W	Timer Count Value Register	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
15~0	CNT[15:0]	TIMn count value
31~16	-	Reserved

14.8.1.3 Timer Reload Register (TIMn_RLD)

Register	R/W	Description	Reset Value	POR
TIMn_RLD (n=0~3)	R/W	Timer Reload Register	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
15~0	RLD[15:0]	A 16-bit reload can be triggered by either a timer overflow or a falling edge on the external input TnEX. When a reload is triggered, the timer automatically loads the user-programmed RLD[15:0] value into TnCNT register
31~16	-	Reserved

14.8.1.4 Timer Flag Register (TIMn_STS)

Register	R/W	Description	Reset Value	POR
TIMn_STS (n=0~3)	R/W	Timer Flag Register	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
2	EXIF	Flag indicating the detection of a falling edge on the external event input. This bit is set by hardware and cleared by writing 1 through software. 0: No external event input detected 1: External input detected (set to 1 by hardware if EXENF = 1) **Note: In capture mode, updating the TnFCAP value is not allowed before clearing this bit through software.**
1	EXIR	Flag indicating the detection of a rising edge on the external event input. This bit is set by hardware and cleared by writing 1 through software. 0: No external event input detected 1: External input detected (set to 1 by hardware if EXENF = 1) **Note: In capture mode, updating the TnRCAP value is not allowed before clearing this bit through software.**
0	TIF	Timer Overflow Flag. This bit is set by hardware and cleared by writing 1 through software. 0: No overflow (must be cleared by software). 1: Overflow (set to 1 by hardware if RCLK = 0 and TCLK = 0).
31~3	-	Reserved

14.8.1.5 TnPWMA Duty Cycle Configuration Register (TIMn_PDTA)(@CPRL = 0)

Register	R/W	Description	Reset Value	POR
TIMn_PDTA(n=0~3)	R/W	TnPWMA Duty Cycle Configuration Register (@CPRL = 0)	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
15~0	PDT[15:0]	TPWMnA Duty Cycle Register, n=0~3. The high-level width of the TPWMnA waveform is PDT[15:0] TIM clocks.
31~16	-	Reserved

14.8.1.6 TnPWMB Duty Cycle Configuration Register (TIMn_PDTB)(@CPRL = 0)

Register	R/W	Description	Reset Value	POR
TIMn_PDTB (n=0~3)	R/W	TnPWMB Duty Cycle Configuration Register (@CPRL = 0)	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
15~0	PDT[15:0]	TPWMnB Duty Cycle Register, n=0~3. The high-level width of the TPWMnB waveform is PDT[15:0] TIM clocks.

Bit number	Bit Mnemonic	Description
31~16	-	Reserved

14.8.1.7 Rising Edge Data Capture Register (TIMn_RCAP)(@CPRL = 1)

Register	R/W	Description	Reset Value	POR
TIMn_RCAP (n=0~3)	R/W	Rising Edge Data Capture Register (@CPRL = 1)	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
15~0	RCAP [15:0]	In PWM capture mode of TIMn, when the rising edge capture condition occurs, the value of the CNT counter will be saved in this register.
31~16	-	Reserved

14.8.1.8 Falling Edge Data Capture Register (TIMn_FCAP) (@CPRL = 1)

Register	R/W	Description	Reset Value	POR
TIMn_FCAP (n=0~3)	R/W	Falling Edge Data Capture Register (@CPRL = 1)	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
15~0	FCAP [15:0]	In PWM capture mode of TIMn, when the falling edge capture condition occurs, the value of the CNT counter will be saved in this register.
31~16	-	Reserved

14.8.1.9 TIMn Interrupt Enable And DMA Control Register (TIMn_IDE)

Register	R/W	Description	Reset Value	POR
TIMn_IDE (n=0~3)	R/W	TIMn Interrupt Enable And DMA Control Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	CAPFDE	CAPRDE	TIDE	EXFIE	EXRIE	TIE	INTEN

Bit number	Bit Mnemonic	Description
6	CAPFDE	Trigger DMA Request On Falling Edge Capture Event Enable Bit 0: Disable DMA request on falling edge capture event 1: Trigger DMA request on the occurrence of a new falling edge capture, DMA will transfer the value of the FCAP register.
5	CAPRDE	Trigger DMA Request On Rising Edge Capture Event Enable Bit 0: Disable DMA request on rising edge capture event 1: Trigger DMA request on the occurrence of a new rising edge capture, DMA will transfer the value of the RCAP register.
4	TIDE	Trigger DMA Request On Timer Overflow Event Enable Bit 0: Disable DMA request on timer overflow 1: Enable DMA request on timer overflow
3	EXFIE	External Event Input Falling Edge Interrupt Enable Bit 0: Disable falling edge interrupt 1: Enable falling edge interrupt
2	EXRIE	External Event Input Rising Edge Interrupt Enable Bit 0: Disable rising edge interrupt 1: Enable rising edge interrupt
1	TIE	Timer Overflow Interrupt Enable Bit 0: Disable overflow interrupt 1: Enable overflow interrupt
0	INTEN	Interrupt Request CPU Enable Control Bit 0: Disable interrupt request 1: Enable interrupt request
31~7	-	Reserved

14.8.2 TIM Register Mapping

Register	Offset Address	R/W	Description	Reset Value	POR
TIM0 Base Address:0x4002_0100					
TIM0_CON	0x00	R/W	Timer0 Control Register	0x0000_0000	0x0000_0000
TIM0_CNT	0x04	R/W	Timer0 Count Value Register	0x0000_0000	0x0000_0000
TIM0_RLD	0x08	R/W	Timer0 Reload Register	0x0000_0000	0x0000_0000
TIM0_STS	0x0C	R/W	Timer0 Flag Register	0x0000_0000	0x0000_0000
TIM0_PDTA	0x10	R/W	TnPWMA Duty Cycle Configuration Register (@CPRL = 0)	0x0000_0000	0x0000_0000
TIM0_RCAP	0x10	R/W	Rising Edge Data Capture Register (@CPRL = 1)	0x0000_0000	0x0000_0000
TIM0_PDTB	0x14	R/W	TnPWMB Duty Cycle Configuration Register (@CPRL = 0)	0x0000_0000	0x0000_0000
TIM0_FCAP	0x14	R/W	Falling Edge Data Capture Register (@CPRL = 1)	0x0000_0000	0x0000_0000
TIM0_IDE	0x18	R/W	TIM0 Interrupt Enable And DMA Control Register	0x0000_0000	0x0000_0000
TIM1 Base Address:0x4002_0140					
TIM1_CON	0x00	R/W	Timer1 Control Register	0x0000_0000	0x0000_0000
TIM1_CNT	0x04	R/W	Timer1 Count Value Register	0x0000_0000	0x0000_0000
TIM1_RLD	0x08	R/W	Timer1 Reload Register	0x0000_0000	0x0000_0000
TIM1_STS	0x0C	R/W	Timer1 Flag Register	0x0000_0000	0x0000_0000
TIM1_PDTA	0x10	R/W	TnPWMA Duty Cycle Configuration Register (@CPRL = 0)	0x0000_0000	0x0000_0000
TIM1_RCAP	0x10	R/W	Rising Edge Data Capture Register (@CPRL = 1)	0x0000_0000	0x0000_0000
TIM1_PDTB	0x14	R/W	TnPWMB Duty Cycle Configuration Register (@CPRL = 0)	0x0000_0000	0x0000_0000
TIM1_FCAP	0x14	R/W	Falling Edge Data Capture Register (@CPRL = 1)	0x0000_0000	0x0000_0000
TIM1_IDE	0x18	R/W	TIM1 Interrupt Enable And DMA Control Register	0x0000_0000	0x0000_0000
TIM2 Base Address:0x4002_1100					
TIM2_CON	0x00	R/W	Timer2 Control Register	0x0000_0000	0x0000_0000

Register	Offset Address	R/W	Description	Reset Value	POR
TIM2_CNT	0x04	R/W	Timer2 Count Value Register	0x0000_0000	0x0000_0000
TIM2_RLD	0x08	R/W	Timer2 Reload Register	0x0000_0000	0x0000_0000
TIM2_STS	0x0C	R/W	Timer2 Flag Register	0x0000_0000	0x0000_0000
TIM2_PDTA	0x10	R/W	TnPWMA Duty Cycle Configuration Register (@CPRL = 0)	0x0000_0000	0x0000_0000
TIM2_RCAP	0x10	R/W	Rising Edge Data Capture Register (@CPRL = 1)	0x0000_0000	0x0000_0000
TIM2_PDTB	0x14	R/W	TnPWMB Duty Cycle Configuration Register (@CPRL = 0)	0x0000_0000	0x0000_0000
TIM2_FCAP	0x14	R/W	Falling Edge Data Capture Register (@CPRL = 1)	0x0000_0000	0x0000_0000
TIM2_IDE	0x18	R/W	TIM2 Interrupt Enable And DMA Control Register	0x0000_0000	0x0000_0000
TIM3 Base Address:0x4002_1140					
TIM3_CON	0x00	R/W	Timer3 Control Register	0x0000_0000	0x0000_0000
TIM3_CNT	0x04	R/W	Timer3 Count Value Register	0x0000_0000	0x0000_0000
TIM3_RLD	0x08	R/W	Timer3 Reload Register	0x0000_0000	0x0000_0000
TIM3_STS	0x0C	R/W	Timer3 Flag Register	0x0000_0000	0x0000_0000
TIM3_PDTA	0x10	R/W	TnPWMA Duty Cycle Configuration Register (@CPRL = 0)	0x0000_0000	0x0000_0000
TIM3_RCAP	0x10	R/W	Rising Edge Data Capture Register (@CPRL = 1)	0x0000_0000	0x0000_0000
TIM3_PDTB	0x14	R/W	TnPWMB Duty Cycle Configuration Register (@CPRL = 0)	0x0000_0000	0x0000_0000
TIM3_FCAP	0x14	R/W	Falling Edge Data Capture Register (@CPRL = 1)	0x0000_0000	0x0000_0000
TIM3_IDE	0x18	R/W	TIM3 Interrupt Enable And DMA Control Register	0x0000_0000	0x0000_0000

15 Power Saving Mode

Upon initial power-up, the system runs in Normal Mode. Additionally, three power-saving modes are available:

- Low-Speed Mode: The system clock source can be LIRC, and the CPU can operate at 32kHz.
- IDLE Mode: The system can be awakened by any interrupt.
- STOP Mode: The system can be awakened by INT0~15, Base Timer and CMP.

16 GPIO

16.1 Clock Source

M0+ core can achieve single-cycle access to GPIO through the IOPORT bus, resulting in highly efficient data transfer. The IOPORT bus clock is derived from HCLK.

16.2 Feature

The GPIO port features of the SC32F15G series are as follows:

- A maximum of 45 bidirectional independently controlled GPIOs
- CPU can access GPIO ports through the IOPORT bus in a single cycle
- Independent setting of pull-up resistors
- All ports have four levels of source driving capability
- All I/Os have high sink current driving capability (50mA)
- 16 I/Os in one group
- Whether input mode or output mode, reading from the port data register retrieves the actual status value of the port

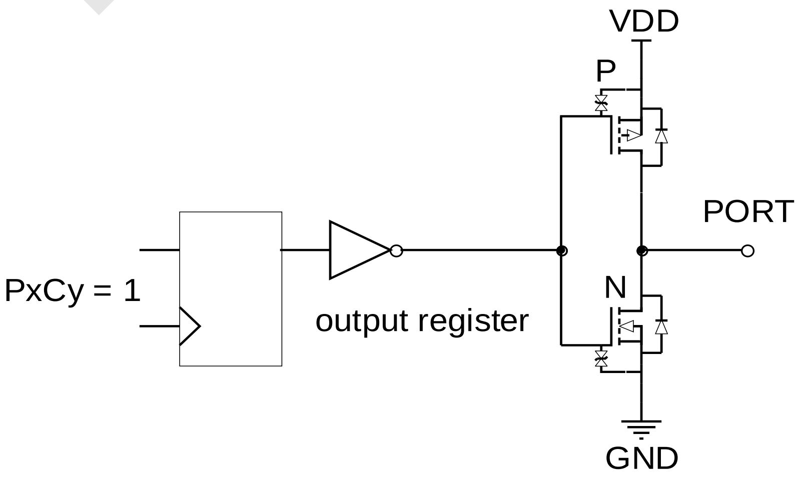
Note: Unused and non-exported ports should be set to strong push-pull output mode

16.3 GPIO Structure Diagram

16.3.1 Strong Push-pull Output Mode

In the strong push-pull output mode, it can provide continuous high-current drive: For detailed electrical parameters, please refer to the "GPIO Parameters" section.

The schematic diagram of the port structure of the strong push-pull output mode is as follows:

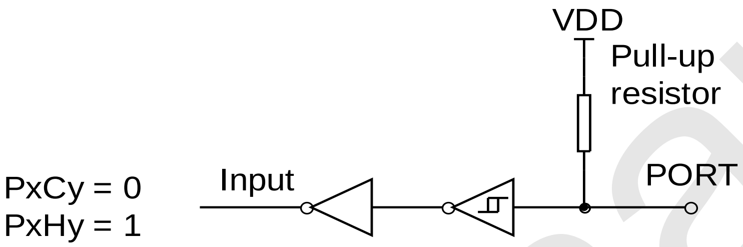


Strong push-pull output mode

16.3.2 Pull-up Input Mode

In the pull-up input mode, a pull-up resistor is constantly connected to the input port. Only when the input port is pulled low, the low-level signal is detected.

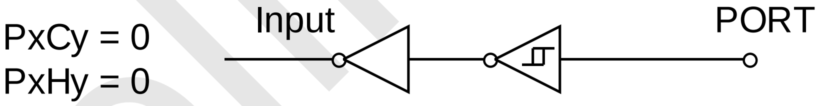
The schematic diagram of the port structure with pull-up input mode is as follows:



Input mode with pull-up resistor

16.3.3 High Impedance Input Mode (Input only)

The schematic diagram of the port structure of the high impedance input mode is as follows:



High impedance input mode

16.4 GPIO Register

16.4.1 GPIO Related Register

16.4.1.1 Port PX Data Register (PX)

Register	R/W	Description	Reset Value	POR
PX X=A,B,C	R/W	Port PX Data Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8

PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8
7	6	5	4	3	2	1	0
PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0

Bit number	Bit Mnemonic	Description
15~0	PDn (n=0~15)	Port PXn Data Register, X=A,B,C, n=0~15 Port latch register data, value read from port data register is the actual state value of the port.
31~16	-	Reserved

16.4.1.2 Port PX Data Register (PXn_BIT)

Register	R/W	Description	Reset Value	POR

PXn_BIT
X=A,B,C
n=0~15
	R/W	Port PX Data Register	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
0	BSRn	Port PXn Bit Assignment Control, n=0~15 Used for individual assignment of the PXn port bit.
31~1	-	Reserved

16.4.1.3 Port PX Data Register (PXn_XR)

Register	R/W	Description	Reset Value	POR

PXn_XR
X=A,B,C
n=0~15
	R/W	Toggle PXn	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8

-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	XRn

Bit number	Bit Mnemonic	Description
0	XRn	Port PXn Bit Toggle Control, n=0~15 0: Invalid 1: Toggle the output of PXn
31~1	-	Reserved

16.4.1.4 Port PX Input/Output Control Register (PXCON)

Register	R/W	Description	Reset Value	POR
PXCON X=A,B,C	R/W	Port PX Input/Output Control Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
MODE15	MODE14	MODE13	MODE12	MODE11	MODE10	MODE9	MODE8
7	6	5	4	3	2	1	0
MODE7	MODE6	MODE5	MODE4	MODE3	MODE2	MODE1	MODE0

Bit number	Bit Mnemonic	Description
15~0	MODEn (n=0~15)	Port PXn Strong Push-Pull Mode Enable Bit, n=0~15 0: PXn in input mode (default at power-up) 1: PXn in strong push-pull mode
31~16	-	Reserved

16.4.1.5 Port PX Pull-up Resister Control Register (PXPH)

Register	R/W	Description	Reset Value	POR
PXPH X=A,B,C	R/W	Port PX Pull-Up Resister Control Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
PUPD15	PUPD14	PUPD13	PUPD12	PUPD11	PUPD10	PUPD9	PUPD8
7	6	5	4	3	2	1	0

PUPD7	PUPD6	PUPD5	PUPD4	PUPD3	PUPD2	PUPD1	PUPD0

Bit number	Bit Mnemonic	Description
15~0	PUPDn (n=0~15)	Port PXn Pull-Up Resister Enable Bit, n=0~15 0: PXn in high-impedance input mode (default at power-up), pull-up resister disable 1: PXn pull-up resistor enable
31~16	-	Reserved

16.4.1.6 GPIO Drive Level Register (PXLEV)

Register	R/W	Description	Reset Value	POR
PXLEV X=A,B,C	R/W	GPIO Drive Level Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
LEV15[1:0]		LEV14[1:0]		LEV13[1:0]		LEV12[1:0]	
23	22	21	20	19	18	17	16
LEV11[1:0]		LEV10[1:0]		LEV9[1:0]		LEV8[1:0]	
15	14	13	12	11	10	9	8
LEV7[1:0]		LEV6[1:0]		LEV5[1:0]		LEV4[1:0]	
7	6	5	4	3	2	1	0
LEV3[1:0]		LEV2[1:0]		LEV1[1:0]		LEV0[1:0]	

Bit number	Bit Mnemonic	Description
31~0	LEVn[1:0] (n=0~15)	Port PXn Level Control Bit, n=0~15 Used for configuring the I_{OH} level of Port PXn 00: Level 0(Maximum) 01: Level 1 10: Level 2 11: Level 3(Minimum)

16.4.2 GPIO Register Mapping

Register	Offset Address	R/W	Description	Reset Value	POR
PA Base Address:0x4001_1000					
PA	0x00	R/W	Port PA Data Register Register	0x0000_0000	0x0000_0000
PACON	0x20	R/W	Port PA Input/Output Control Register	0x0000_0000	0x0000_0000

Register	Offset Address	R/W	Description	Reset Value	POR
PAPH	0x40	R/W	Port PA Pull-Up Resister Control Register	0x0000_0000	0x0000_0000
PALEV	0x60	R/W	GPIO Drive Level Register	0x0000_0000	0x0000_0000

Register	Offset Address	R/W	Description	Reset Value	POR
PB Base Address: 0x4001_1100					
PB	0x00	R/W	Port PB Data Register Register	0x0000_0000	0x0000_0000
PBCON	0x20	R/W	Port PB Input/Output Control Register	0x0000_0000	0x0000_0000
PBPH	0x40	R/W	Port PB Pull-Up Resister Control Register	0x0000_0000	0x0000_0000
PBLEV	0x60	R/W	GPIO Drive Level Register	0x0000_0000	0x0000_0000

Register	Offset Address	R/W	Description	Reset Value	POR
PC Base Address:0x4001_1200					
PC	0x00	R/W	Port PC Data Register Register	0x0000_0000	0x0000_0000
PCCON	0x20	R/W	Port PC Input/Output Control Register	0x0000_0000	0x0000_0000
PCPH	0x40	R/W	Port PC Pull-Up Resister Control Register	0x0000_0000	0x0000_0000
PCLEV	0x60	R/W	GPIO Drive Level Register	0x0000_0000	0x0000_0000

17 UART0~2

17.1 Clock Source

The SC32F15G series UART has only one clock source, which is derived from PCLK

17.2 Feature

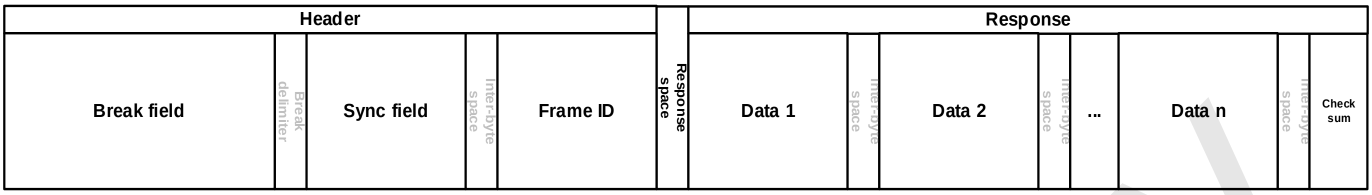
- Three UARTs，UART0~2
- UART2 has a complete LIN interface
  - Can switch between master and slave modes
  - Supports hardware break sending in master mode (10/13 bits)
  - Supports hardware break detection in slave mode (10/11 bits)
  - Supports baud rate synchronization in slave mode
  - Provides related interrupts/status bits/flags/fault tolerance range
- UART0~2 support signal port mapping and can be mapped to another set of I/Os
- Each UART has four communication modes to choose from:
  - Mode 0: 8-bit half-duplex synchronous communication mode, serial data is transmitted and received on the RX pin. The TX pin is used as the transmit shift clock. Each frame transmits or receives 8 bits, with the low bit transmitted or received first
  - Mode 1: 10-bit full-duplex asynchronous communication, consisting of 1 start bit, 8 data bits, and 1 stop bit. The communication baud rate is variable
  - Mode 2: Reserved
  - Mode 3: 11-bit full-duplex asynchronous communication, consisting of 1 start bit, 8 data bits, 1 programmable 9th bit and 1 stop bit. The communication baud rate is variable
- Interrupts will be generated and corresponding flags TXIF and RXIF will be set when transmission and reception are complete. Interrupt flags need to be cleared by software
- UART0 and UART1 can generate DMA requests
- UART2 cannot generate DMA requests
- Independent baud rate generator
- UART2 does not support waking up from STOP Mode
- UART0/1 support waking up from STOP Mode:
  - The falling edge of the START bit can wake up STOP Mode
  - Provides corresponding wake-up interrupt enable bit WKIE and wake-up interrupt flag WKIF

17.3 UART2-LIN

UART2 supports standard LIN communication protocol.

17.3.1 LIN Frame Structure

Under the LIN protocol, all communication information is encapsulated into frames. A frame is composed of a header (provided by the master task) and a response (provided by the slave task). The header (provided by the master task) consists of a break field, a sync (synchronization) field and a frame ID. The frame ID serves solely to define the purpose of the frame and the slave is responsible for responding to the relevant frame ID. The response consists of a data field and a checksum field.



LIN Frame Structure Diagram

17.3.2 LIN Master Mode

By setting FUNCSEL=1 and SLVEN=0, the UART will support LIN master mode. In LIN mode, according to the LIN protocol, each byte is initiated with a dominant bit, followed by 8 data bits with no parity bit, LSB first, and ends with a recessive STOP bit.

The initialization process for LIN master mode is as follows:

① Configure the UART_BAUD register to set the baud rate.
② Set FUNCSEL=1 to select the LIN function mode.
③ Set SM[1:0] to 01 to configure the UART in Mode 1.

A complete header consists of a break field, a sync field, and a frame ID. The UART controller can choose the 'break field' as the transmitted header. The 'sync field' and 'frame ID field' need to be written by the user through software, that is to say, to send a complete header to the bus, the software must sequentially fill in the sync data (0x55) and the frame ID data into the UART_DAT register.

17.3.3 LIN Slave Mode

By setting FUNCSEL=1 and SLVEN=1, the UART will support LIN slave mode. In LIN mode, according to the LIN protocol, each byte is initiated with a dominant bit, followed by 8 data bits with no parity bit, LSB first, and ends with a recessive STOP bit.

The initialization process for LIN slave mode is as follows:

① Configure the UART_BAUD register to set the baud rate.
② Set FUNCSEL=1 to select the LIN function mode.
③ Set SM[1:0] to 01 to configure the UART in Mode 1.
④ Set SLVEN to 1 to enable LIN slave mod

In LIN slave mode, the slave break field detection function is enabled by setting LBDL to detect and receive 'break field'. After receiving a break, the BKIF flag will be set and an interrupt will be generated if BKIE is set to 1. To avoid bit rate deviation, users can set SLVAREN to enable automatic resynchronization feature to prevent clock errors.

17.3.4 Synchronization Error Detection

In automatic resynchronization mode, the controller will detect errors in the sync field. The error detection compares the current baud rate with the baud rate of the received sync field, and the following both detections are performed simultaneously.

Check 1: Based on the measurements from the first falling edge to the last falling edge of the sync field

- If the error exceeds 15%, the header error flag SLVHEIF will be set.
- If the error is between 14% and 15%, the header error flag SLVHEIF may be set (depending on data dephasing).

Check 2: Based on the measurements from each falling edge of the sync field

- If the error exceeds 19%, the header error flag SLVHEF will be set.
- If the error is between 15% and 19%, the header error flag SLVHEIF may be set (depending on data dephasing).

Note: Error detection is based on the current baud rate clock. Therefore, to ensure the accuracy of error detection, it is recommended that users reload the baud rate to its initial value through software before a new break field is received.

17.4 UART Interrupt

For UARTn, n=0~2, interrupts will be generated upon “wake-up” or “data transmission/reception completion”. Separate interrupt enable bits can be used to enhance flexibility.

Interrupt Event	Interrupt Request Control Bit	Event Flag	Interrupt Enable Sub-Switch
Uart wake up from STOP mode	UARTn_IDE ->INTEN	WKIF	WKIE
Data transmission completion		TXIF	TXIE
Data reception completion		RXIF	RXIE
Break detected	UART2_IDE->INTEN	BKIF	BKIE
Header error detected by LIN slave		SLVHEIF	SLVHEIE
Baud rate synchronization complete		SYNCIF	SYNCIE

17.5 UART0/1 Register

17.5.1 UART0/1 Related Register

17.5.1.1 UART Control Register (UARTn_CON)

Register	R/W	Description	Reset Value	POR
UARTn_CON (n=0/1)	R/W	UART Control Register	0x0000_0000	0x0000_0000

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

TXEN	RXEN	-	PRESCALER	-	SM2	SM1	SM0

<table>
<tr><th>Bit number</th><th>Bit Mnemonic</th><th>Description</th></tr>
<tr><td>13</td><td>SPOS[1:0]</td><td>● UART0 Port Mapping Control Bit@UART0_CON
<table>
<tr><th>Port / SPOS value</th><th>RX0</th><th>TX0</th></tr>
<tr><td>SPOS=0</td><td>PC7</td><td>PC8</td></tr>
<tr><td>SPOS=1</td><td>PB6</td><td>PB5</td></tr>
</table>
● UART1 Port Mapping Control Bit@UART1_CON
<table>
<tr><th>Port / SPOS value</th><th>RX1</th><th>TX1</th></tr>
<tr><td>SPOS=0</td><td>PC1</td><td>PC2</td></tr>
<tr><td>SPOS=1</td><td>PB3</td><td>PB4</td></tr>
</table></td></tr>
<tr><td>7</td><td>TXEN</td><td>UART Transmission Enable Control Bit
0: Disallow data transmission, and the TXD signal no longer affects the state of the associated pin. If the user program restricts the sending function and only utilizes reception, other functions multiplexed with the TX pin will not be affected
1: Allow data transmission, and the pin associated with TXD switches to the TXD signal mode</td></tr>
<tr><td>6</td><td>RXEN</td><td>UART Reception Enable Control Bit
0: Disallow data reception
1: Allow data reception</td></tr>
<tr><td>4</td><td>PRESCALER</td><td>Baud Rate Multiplier Setting Bit
This bit has different definitions in different modes of UART:
● When SM0~1=01(UART mode 1) or SM0~1=11(UART mode 3):
■ 0: Serial port runs at 1/1 frequency of the system clock
■ 1: Serial port runs at 1/16 frequency of the system clock
● When SM0~1=00(UART mode 0):
■ 0: Serial port runs at 1/12 frequency of the system clock
■ 1: Serial port runs at 1/4 frequency of the system clock</td></tr>
<tr><td>2</td><td>SM2</td><td>SBUF8 Set Interrupt Enable Bit
This bit is only valid in mode 3
0: Set RI interrupt request upon receiving each complete data frame
1: Set RI interrupt request only when SBUF8=1 upon receiving a complete data frame</td></tr>
<tr><td>1~0</td><td>SM[1:0]</td><td>UART Communication Mode Control Bits
00: Mode 0, 8-bit half-duplex synchronous communication mode. Serial data is transmitted and received on the RX pin. The TX pin is used as the transmit shift clock. Each frame transmits and receives 8 bits, with the low bit first. Enabling the RXEN bit in this mode will cause the UART to generate a complete frame clock, and set RXIF to 1</td></tr>
</table>

Bit number	Bit Mnemonic	Description
		01: Mode 1, 10-bit full-duplex asynchronous communication, consisting of 1 start bit, 8 data bits, and 1 stop bit. Communication baud rate is variable 10: Reserved 11: Mode 3, 11-bit full-duplex asynchronous communication, consisting of 1 start bit, 8 data bits, a programmable 9th bit, and 1 stop bit. Communication baud rate is variable
31~14 12~8 5,3	-	Reserved

17.5.1.2 UART Flag Register (UARTn_STS)

Register	R/W	Description	Reset Value	POR
UARTn_STS (n=0/1)	R/W	UART Flag Register	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
4	WKIF	UART Wake Up Flag This bit will be set to 1 after UART wake up from STOP mode, and an interrupt will be generated if WKIE=1. This bit is cleared by writing to 1 through software.
1	TXIF	Transmission Interrupt Flag This bit will be set to 1 upon data transmission complete, and an interrupt will be generated if TXIE=1. This bit is cleared by writing to 1 through software. **Note: In DMA mode, after DMA writes to the transmit buffer, this bit is cleared by the DMA module, users do not need to clear it by software.**
0	RXIF	Reception Interrupt Flag This bit will be set to 1 upon data reception complete, and an interrupt will be generated if RXIE=1. This bit is cleared by writing to 1 through software.

Bit number	Bit Mnemonic	Description
		Note: In DMA mode, after DMA writes to the receive buffer, this bit is cleared by the DMA module, users do not need to clear it by software.
31~5 3~2	-	Reserved

17.5.1.3 UART Baud Configuration Register (UARTn_BAUD)

Register	R/W	Description	Reset Value	POR
UARTn_BAUD (n=0/1)	R/W	UART Baud Configuration Register	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
15~0	BAUD[15:0]	UART Baud Configuration Bit After writing to BAUD[15:0], the UART baud rate will be configured according to the following formula: BaudRate=f_{UART} / BAUD[15:0] f_{UART} is the final frequency of the UART clock source after prescaling, as described in the PRESCALER bit description. **Note:BAUD[15:0] must be greater than 0x0010。**
31~16	-	Reserved

17.5.1.4 UART Data Register (UARTn_DATA)

Register	R/W	Description	Reset Value	POR
UARTn_DATA (n=0/1)	R/W	UART Data Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	SBUF8

7	6	5	4	3	2	1	0
SBUF[7:0]							

Bit number	Bit Mnemonic	Description
8	SBUF8	The 9th bit of UART transmission/reception This bit is only valid in mode 3
7~0	SBUF[7:0]	UART Data Buffer Read operation: Returns the content of the receive buffer Write operation: The data in SBUF will be sent to the transmit shift register, initiating the transmission process.
31~9	-	Reserved

17.5.1.5 UART Interrupt Enable And DMA Control Register (UARTn_IDE)

Register	R/W	Description	Reset Value	POR
UARTn_IDE (n=0/1)	R/W	UART Interrupt Enable And DMA Control Register	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
7	TXDMAEN	DMA Transmission Channel Enable Bit 0: Disable DMA transmission function 1: Enable DMA transmission function The set of TXIF can trigger DMA channel sending request after enabling this bit.
6	RXDMAEN	DMA Reception Channel Enable Bit 0: Disable DMA reception function 1: Enable DMA reception function The set of RXIF can trigger DMA channel receving request after enabling this bit.
4	WKIE	UART Wake Up Interrupt Enable Bit 0: An interrupt will not be generated after WKIF is set 1: An interrupt will be generated after WKIF is set
2	TXIE	UART Transmission Interrupt Enable Bit 0: An interrupt will not be generated after TXIF is set 1: An interrupt will be generated after TXIF is set

Bit number	Bit Mnemonic	Description
1	RXIE	UART Receving Interrupt Enable Bit 0: An interrupt will not be generated after RXIF is set 1: An interrupt will be generated after RXIF is set
0	INTEN	Interrupt Request CPU Enable Control Bit 0: Disable interrupt request 1: Enable interrupt request
31~8 5,3	-	Reserved

17.5.2 UART0~1 Register Mapping

Register	Offset Address	R/W	Description	Reset Value	POR	Access Restriction
UART0 Base Address:0x4002_0020						
UART0_CON	0x00	R/W	UART0 Control Register	0x0000_0000	0x0000_0000	
UART0_STS	0x04	R/W	UART0 Flag Register	0x0000_0000	0x0000_0000	
UART0_BAUD	0x08	R/W	UART0 Baud Configuration Register	0x0000_0000	0x0000_0000	
UART0_DATA	0x0C	R/W	UART0 Data Register	0x0000_0000	0x0000_0000	Do not support byte/half word access
UART0_IDE	0x10	R/W	UART0 Interrupt Enable and DMA Control Register	0x0000_0000	0x0000_0000	

Register	Offset Address	R/W	Description	Reset Value	POR	Access Restriction
UART1 Base Address:0x4002_0080						
UART1_CON	0x00	R/W	UART1 Control Register	0x0000_0000	0x0000_0000	
UART1_STS	0x04	R/W	UART1 Flag Register	0x0000_0000	0x0000_0000	
UART1_BAUD	0x08	R/W	UART1 Baud Configuration Register	0x0000_0000	0x0000_0000	
UART1_DATA	0x0C	R/W	UART1 Data Register	0x0000_0000	0x0000_0000	Do not support byte/half word access
UART1_IDE	0x10	R/W	UART1 Interrupt Enable and DMA Control Register	0x0000_0000	0x0000_0000	

17.6 UART2 Register

17.6.1 UART2 Related Register

17.6.1.1 UART Control Register (UARTn_CON)

Register	R/W	Description	Reset Value	POR
UARTn_CON (n=2)	R/W	UART Control Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	BKSIZE	-	BKTR
23	22	21	20	19	18	17	16
-	-	-	-	-	LBDL	SLVAREN	SLVEN
15	14	13	12	11	10	9	8
-	-	SPOS	-	-	-	-	FUNCSEL
7	6	5	4	3	2	1	0
TXEN	RXEN	-	PRESCALER	-	SM2	SM1	SM0

Bit number	Bit Mnemonic	Description
26	BKSIZE	Break Field Size Selection Bit 0: 10 bits 1: 13 bits
24	BKTR	LIN Mode Break Bit Transmit Trigger Bit 0: Do not trigger break bit transmission 1: Trigger break bit transmission **Note:** **1. This bit is invalid when SLVEN=1** **2.This bit will be cleared after break field transmission is complete**
18	LBDL	LIN Break Detection Size Selection Bit 0: Detect 10 bits break 1: Detect 11 bits break **Note:** **1.The slave detection size needs to be set according to master break field size** **2.This bit is invalid when SLVEN=0**
17	SLVAREN	Slave Baud Rate Automatic Resynchronization Enable Bit 0: Disable slave baud rate automatic resynchronization 1: Enable slave baud rate automatic resynchronization

<table>
<tr><th>Bit number</th><th>Bit Mnemonic</th><th>Description</th></tr>
<tr><td></td><td></td><td>When the automatic resynchronization is enabled, the system continuously samples for 5 falling edges using the LIN working clock after each LIN break field. The measured result is stored in the internal baud rate buffer register, and the value of the UARTn_BAUD register will be automatically updated

<b>Note:</b>
<b>This bit is invalid when SLVEN=0</b></td></tr>
<tr><td>16</td><td>SLVEN</td><td>LIN Slave Mode Enable Bit
0: LIN slave mode disable(LIN master mode enable)
1: LIN slave mode enable(LIN master mode disable)

<b>Note:</b>
<b>1.SLVAREN, LBDL is invalid in LIN master mode</b>
<b>2.Break can be detected in LIN slave mode</b></td></tr>
<tr><td>13</td><td>SPOS</td><td>● UART2 Port Mapping Control Bit@UART2_CON
<table>
<tr><th>Port
SPOS value</th><th>RX2</th><th>TX2</th></tr>
<tr><td>SPOS=0</td><td>PA1</td><td>PA0</td></tr>
<tr><td>SPOS=1</td><td>PB1</td><td>PB0</td></tr>
</table></td></tr>
<tr><td>8</td><td>FUNCSEL</td><td>Function Selection Bit
0: UART function
1: LIN function, LIN hardware module and UART module are both enabled, with the LIN module being responsible for break detection/generation, baud rate synchronization/updating
Note: When FUNCSEL = 1, bits 31 to 16 of UART2_CON are valid.</td></tr>
<tr><td>7</td><td>TXEN</td><td>UART Transmission Enable Control Bit
0: Disallow data transmission, and the TXD signal no longer affects the state of the associated pin. If the user program restricts the sending function and only utilizes reception, other functions multiplexed with the TX pin will not be affected
1: Allow data transmission, and the pin associated with TXD switches to the TXD signal mode</td></tr>
<tr><td>6</td><td>RXEN</td><td>UART Reception Enable Control Bit
0: Disallow data reception
1: Allow data reception</td></tr>
<tr><td>4</td><td>PRESCALER</td><td>Baud Rate Multiplier Setting Bit
This bit has different definitions in different modes of UART:
● When SM0~1=01(UART mode 1) or SM0~1=11(UART mode 3):
■ 0: Serial port runs at 1/1 frequency of the system clock
■ 1: Serial port runs at 1/16 frequency of the system clock
● When SM0~1=00(UART mode 0):
■ 0: Serial port runs at 1/12 frequency of the system clock</td></tr>
</table>

Bit number	Bit Mnemonic	Description
		■ 1: Serial port runs at 1/4 frequency of the system clock
2	SM2	SBUF8 Set Interrupt Enable Bit This bit is only valid in mode 3 0: Set RI interrupt request upon receiving each complete data frame 1: Set RI interrupt request only when SBUF8=1 upon receiving a complete data frame
1~0	SM[1:0]	UART Communication Mode Control Bits 00: Mode 0, 8-bit half-duplex synchronous communication mode. Serial data is transmitted and received on the RX pin. The TX pin is used as the transmit shift clock. Each frame transmits and receives 8 bits, with the low bit first. Enabling the RXEN bit in this mode will cause the UART to generate a complete frame clock, and set RXIF to 1 01: Mode 1, 10-bit full-duplex asynchronous communication, consisting of 1 start bit, 8 data bits, and 1 stop bit. Communication baud rate is variable 10: Reserved 11: Mode 3, 11-bit full-duplex asynchronous communication, consisting of 1 start bit, 8 data bits, a programmable 9th bit, and 1 stop bit. Communication baud rate is variable
31~27 25 23~19 15~14 12~9 5,3	-	Reserved

17.6.1.2 UART Flag Register (UARTn_STS)

Register	R/W	Description	Reset Value	POR
UARTn_STS (n=2)	R/W	UART Flag Register	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
11	SYNCIF	LIN Mode Baud Rate Synchronization Complete Flag

Bit number	Bit Mnemonic	Description
		This flag will be set after sync field(0x55)
10	SLVSYNIF	LIN Slave Sync Field Status Bit This bit indicates the LIN sync field is being analyzed in automatic resynchronization mode. If the receiver header detects some errors, the user must reset the internal circuit by writing a 1 to this bit to search for a new frame header. 0: The current character is not in the LIN sync state 1: The current character is in the LIN sync state **Note:** **1.This bit is only valid in LIN slave mode** **2.This bit is read-only and used as a status bit** **3.When 1 is written to this bit, the hardware will reload the initial baud rate and search for a new frame header**
9	SLVHEIF	LIN Slave Header Error Flag When LIN header error is detected in LIN slave mode, this bit will be set to 1 by hardware and writing a 1 to this bit will clear this bit 0: Header error is not detected 1: Header error is detected Error conditions include: 1. The break field interval is too short (less than the time of 0.5 bit periods). 2. In non-automatic resynchronization mode, the sync field data is not 0x55. 3. In automatic resynchronization mode, the sync field deviates from the expected value. **Note:This bit is only valid in LIN slave mode**
8	BKIF	LIN Mode Break Interrupt Flag This bit will be set to 1 upon data transmission is complete, and an interrupt will be generated if BKIE=1. This bit is cleared by writing to 1 through software.
1	TXIF	Transmission Interrupt Flag This bit will be set to 1 upon data transmission complete, and an interrupt will be generated if TXIE=1. This bit is cleared by writing to 1 through software. **Note: In DMA mode, after DMA writes to the transmit buffer, this bit is cleared by the DMA module, users do not need to clear it by software.**
0	RXIF	Reception Interrupt Flag This bit will be set to 1 upon data reception complete, and an interrupt will be generated if RXIE=1.

Bit number	Bit Mnemonic	Description
		This bit is cleared by writing to 1 through software. **Note: In DMA mode, after DMA writes to the receive buffer, this bit is cleared by the DMA module, users do not need to clear it by software.**
31~12 7~2	-	Reserved

17.6.1.3 UART Baud Configuration Register (UARTn_BAUD)

Register	R/W	Description	Reset Value
UARTn_BAUD (n=2)	R/W	UART Baud Configuration Register	0x0000_0000

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

Bit number	Bit Mnemonic	Description
15~0	BAUD[15:0]	UART Baud Configuration Bit After writing to BAUD[15:0], the UART baud rate will be configured according to the following formula: BaudRate=f_{UART} / BAUD[15:0] f_{UART} is the final frequency of the UART clock source after prescaling, as described in the PRESCALER bit description. **Note:BAUD[15:0] must be greater than 0x0010。**
31~16	-	Reserved

17.6.1.4 UART Data Register (UARTn_DATA)

Register	R/W	Description	Reset Value
UARTn_DATA (n=2)	R/W	UART Data Register	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8

-	-	-	-	-	-	-	SBUF8
7	6	5	4	3	2	1	0
SBUF[7:0]							

Bit number	Bit Mnemonic	Description
8	SBUF8	The 9th bit of UART transmission/reception This bit is only valid in mode 3
7~0	SBUF[7:0]	UART Data Buffer Read operation: Returns the content of the receive buffer Write operation: The data in SBUF will be sent to the transmit shift register, initiating the transmission process.
31~9	-	Reserved

17.6.1.5 UART Interrupt Enable And DMA Control Register (UARTn_IDE)

Register	R/W	Description	Reset Value
UARTn_IDE (n=2)	R/W	UART Interrupt Enable And DMA Control Register	0x0000_0000

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

Bit number	Bit Mnemonic	Description
11	SYNCIE	LIN Mode Baud Rate Synchronization Complete Interrupt Enable Bit
9	SLVHEIE	LIN Slave Header Error Interrupt Enable Bit This bit is only valid in LIN slave mode When LIN header error is detected in LIN slave mode, SLVHEIE will be set to 1 by hardware, and an interrupt will be generated if SLVHEIE=1 Error conditions include: 1. The break field interval is too short (less than the time of 0.5 bit periods). 2. In non-automatic resynchronization mode, the sync field data is not 0x55. 3. In automatic resynchronization mode, the sync field deviates from the expected value.
8	BKIE	LIN Mode Break Interrupt Enable Bit

Bit number	Bit Mnemonic	Description
		0: An interrupt will not be generated after BKIF is set 1: An interrupt will be generated after BKIF is set
2	TXIE	UART Transmission Interrupt Enable Bit 0: An interrupt will not be generated after TXIF is set 1: An interrupt will be generated after TXIF is set
1	RXIE	UART Receving Interrupt Enable Bit 0: An interrupt will not be generated after RXIF is set 1: An interrupt will be generated after RXIF is set
0	INTEN	Interrupt Request CPU Enable Control Bit 0: Disable interrupt request 1: Enable interrupt request
31~12 10 7~3	-	Reserved

17.6.2 UART2 Register Mapping

Register	Offset Address	R/W	Description	Reset Value	POR	Access Restriction
UART2 Base Address:0x4002_1020						
UART2_CON	0x00	R/W	UART2 Control Register	0x0000_0000	0x0000_0000	
UART2_STS	0x04	R/W	UART2 Flag Register	0x0000_0000	0x0000_0000	
UART2_BAUD	0x08	R/W	UART2 Baud Configuration Register	0x0000_0000	0x0000_0000	
UART2_DATA	0x0C	R/W	UART2 Data Register	0x0000_0000	0x0000_0000	Do not support byte/half word access
UART2_IDE	0x10	R/W	UART2 Interrupt Enable and DMA Control Register	0x0000_0000	0x0000_0000	

18 SPI0~1

18.1 Clock Source

The SC32F15G series SPI has only one clock source, which is derived from PCLK.

18.2 SPI0 Feature

- Supports 11-stage SPI clock pre-scaling
- Signal ports can be mapped to another set of ports
- SPI0 signal ports strong driving
    - In SPI communication mode, the corresponding signal port's pin output driving capability will be enhanced, while in other modes, it remains consistent with the characteristics of a regular I/O.
    - Its mapped signal port can also be set to strong driving to ensure the consistency of SPI0 across any port
- Features a 16-bit 8-level FIFO with independent transmission and reception
    - SPI0's FIFO function allows continuous writing of 8 or fewer 16-bit transmit data to the SPI send buffer (SPI0_DATA). During SPI transmission, the data written into the FIFO first is also sent first. When the data written by the user to the FIFO is sent, the FIFO empty flag TXEIF will be set; if the FIFO is full, the write conflict flag WCOL will be set, and the user cannot write data to the FIFO until the data in the FIFO is sent out and the FIFO is not full. The interrupt flag SPIF will be set only when all the data in the FIFO has been sent
    - Continuously read 8 or fewer 16-bit receive data from the SPI receive buffer (SPI0_DATA), with the first received data being the first to be read
    - FIFO data transfer half-interrupt and corresponding flags for timely reading/writing of data:
        - Provides an interrupt and corresponding flag TXHIF when there is less than half of the valid data in the transmit FIFO
        - Provides an interrupt and corresponding flag RXHIF when there is more than half of the data in the receive FIFO
    - Support recieve buffer overflow interrupt and corresponding flag to promptly notify exceptions
- Support DMA
    - Enable TXDMAEN, and the DMA request can be triggered after the transmit buffer empty flag TXEIF is set, and TXEIF will be automatically cleared after DMA write transmit buffer.
    - Enable RXDMAEN, and the DMA request can be triggered after the receive buffer not empty status flag RXNEIF is set, and RXEIF will be automatically cleared after DMA read receive buffer.

18.3 SPI1 Feature

- SPI1 and TWI1 operate independently with multiplexed register addresses and signal pins
- Supports 13-stage SPI clock pre-scaling
- Signal ports can be mapped to three additional sets of ports
- No FIFO
- Supports DMA

18.4 SPI Function Description

18.4.1 Signal Description

Master Output Slave Input (MOSI):

This signal connects the master device to a slave device. Data is transmitted serially from the master device to the slave device through MOSI, which is an output from the master and an input to the slave.

Master Input Slave Output (MISO):

This signal connects the slave device to the master device. Data is transmitted serially from the slave device to the master device through MISO, which is an output from the slave and an input to the master. When the SPI is configured as a slave and not selected, the MISO pin of the slave device will be in a high-impedance state.

SPI Serial Clock (SCK):

The SCK signal is used to control the synchronous movement of input and output data on the MOSI and MISO lines. One byte is transferred on the line every 8 clock cycles. If a slave device is not selected, the SCK signal will be ignored by that slave device.

18.4.2 Working Mode

SPI can be configured in either master or slave mode. The configuration and initialization of the SPI module are accomplished by setting SPI control registers (SPI0_CON/SPI1_TWI1_CON) and SPI interrupt enable and DMA control register (SPI0_IDE/SPI1_TWI1_IDE). Once configured, data transmission will be achieved by setting the SPI data register (SPI0_DATA/SPI1_TWI1_DATA) during SPI communication.

During SPI communication, data is serially shifted in and out in a synchronous manner. The serial clock line (SCK) synchronizes the movement and sampling of data on the two serial data lines (MOSI and MISO). If a slave device is not selected, it will not participate in activities on the SPI bus.

When the SPI master device transmits data to the slave device through the MOSI line, the slave device responds by sending data to the master device through the MISO line. This achieves synchronous full-duplex transmission of data at the same clock. The transmit shift register and receive shift register share the same special function register address. Writing to the SPI data register (SPD) will write to the transmit shift register, and reading from SPD will retrieve data from the receive shift register.

Some devices with SPI interfaces may have an SS pin (slave select pin, active low). When communicating with SC32F15G through the SPI, the connection of the SS pins of other devices on the SPI bus should be configured according to the different communication modes. The table below outlines the connection methods for the SS pins of other devices on the SPI bus in different communication modes for SC32F15G:

SC32F15G SPI	Other Devices On SPI Bus	Mode	Slave SS
Master	Slave	One Master One Slave	Pull low
		One Master Multiple	SC32F15G has multiple I/O pins, each

SC32F15G SPI	Other Devices On SPI Bus	Mode	Slave SS
		Slave	connected to the SS pin of different slave devices. Before data transmission, the SS pin of the specific slave device must be pulled low.
Slave	Master	One Master One Slave	Pull high

Master Mode

- Mode Active:

The SPI master device controls the initiation of all data transfers on the SPI bus. When SPI0_CON.MSTR=1/SPI1_TWI1_CON.SMSTR=1, the SPI operates in master mode, and only one master device can initiate the transfer.

- Transmission:

In SPI master mode, users can perform the following operation on SPD: write a byte of data to SPD[7:0] in 8-bit mode or write a 16-bit data to SPD[15:0], then the data will be written to the transmit shift buffer. If there is already data in the transmit shift register, the master SPI will generate a WCOL signal to indicate that the write is too fast. However, the data in the transmit shift register will not be affected, and the transmission will not be interrupted. Additionally, if the transmit shift register is not empty, the master device immediately serially shifts out the data from the transmit shift register to MOSI at the SPI clock frequency on SCK. When the transfer is complete, the SPIF/QTWIF bit in the SPI status register SPI0_STS/SPI1_TWI1_STS will be set to 1. If SPI interrupts are enabled, an interrupt will also be generated when SPIF/QTWIF is set to 1.

- Reception:

When the master device sends data to the slave device via MOSI, the corresponding data will be simultaneously transmitted by the slave device via MISO to the receive shift register of the master device, achieving full-duplex operation. Therefore, when the SPIF/QTWIF flag is set to 1, it indicates that the transmission is complete and the reception of data is also complete. The received data from the slave device is stored in the receive shift register of the master device according to the MSB or LSB priority transmission direction. When a byte of data is fully moved into the receive register, processor can obtain the data by reading SPD.

Slave Mode

- Mode Active:

When SPI0_CON.MSTR/SPI1_TWI1_CON.SMSTR is cleared, the SPI operates in slave mode.

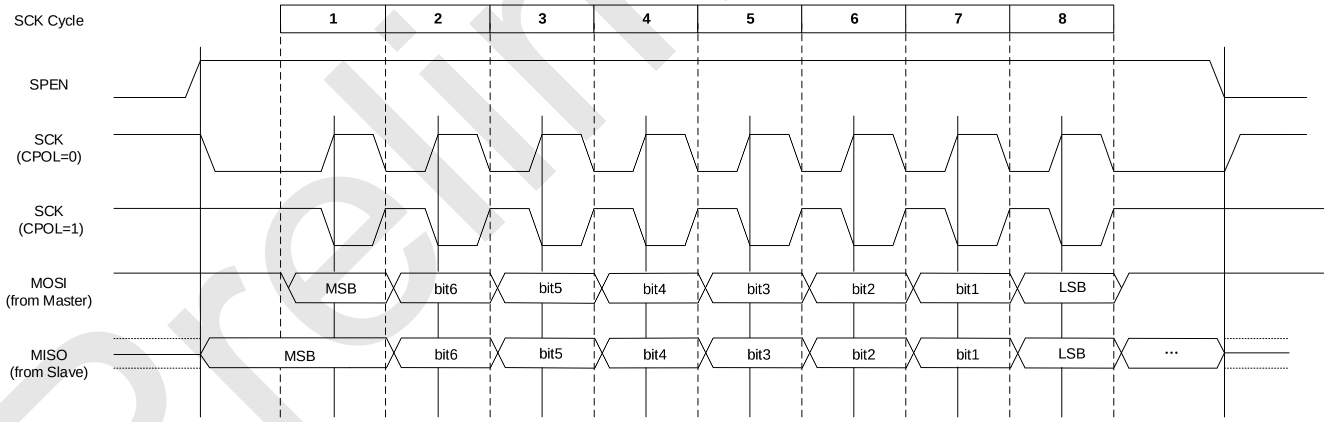
- Transmission And Reception:

In slave mode, data will be input through MOSI and output through MISO according to the SCK signal controlled by the master device. A bit counter records the number of SCK edges. When the receive shift register moves in 8 bits of data (one byte) while the transmit shift register moves out 8 bits of data (one byte), the SPIF/QTWIF flag will be set to 1. The data can be obtained by reading the SPD register. If SPI interrupts are enabled, an interrupt will be generated when SPIF/QTWIF is set to 1. At this time, the receive shift register retains the original data and SPIF/QTWIF is set to 1, indicating that the SPI slave device will not receive any data until SPIF/QTWIF is cleared. The SPI slave device must write the data to be transmitted into the transmit shift register before the master device starts a new transmission. If no data is written before starting transmission, the slave device will send the "0x00" to the master device. If a write to SPD occurs during the transmission process, the WCOL flag of the SPI slave device will be set to 1, indicating a write SPD conflict. However, the data in the shift register is not affected, and the transmission will not be interrupted.

18.4.3 Transmission format

Setting the CPOL (Clock Polarity) and CPHA (Clock Phase) bits in the SPI control register SPI0_CON/SPI1_TWI1_CON by software, users can choose from four combinations of SPI clock polarity and phase. CPOL determines the clock polarity, indicating the electrical level of idle state. It has minimal impact on the SPI transfer format. CPHA defines the clock phase, determining the clock edge at which data is sampled and shifted. In a communication link between a master and a slave device, the settings of clock polarity and phase should be consistent.

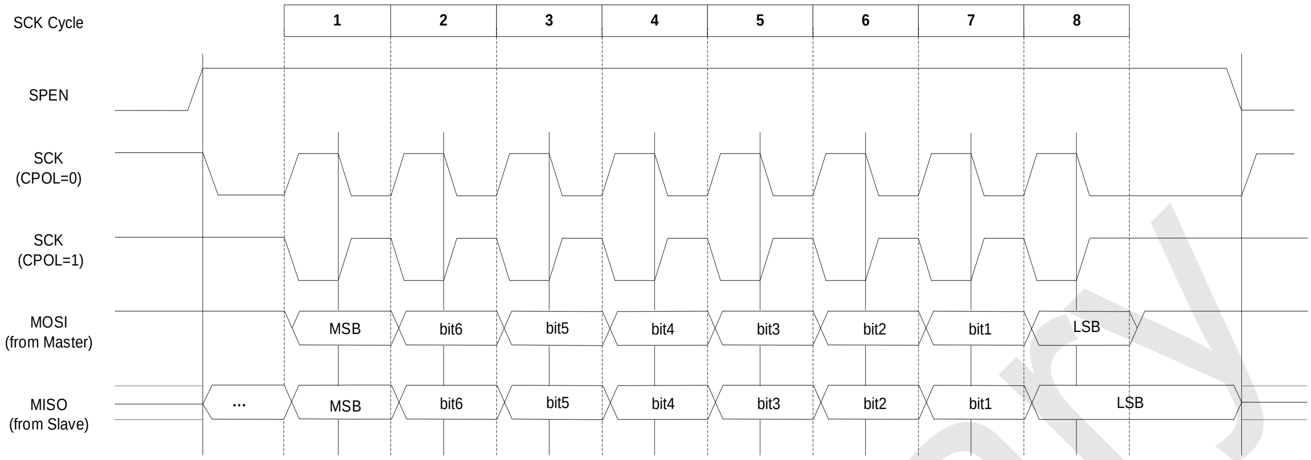
When CPHA = 0, data will be captured at the first edge of SCK. and the data must be prepared by the slave device before the first edge of SCK.



CPHA = 0 Data Transfer Diagram

When CPHA = 1, the master device outputs data to MOSI on the first edge of SCK, and the slave device treats the first edge of SCK as the start of the transmission. The second edge of SCK is used to capture the data, so users must complete the write operation to SPD register within the first two edges of SCK.

This data transmission format is a preferred mode for communication between one master device and one slave device in the SPI protocol.



CPHA = 1 Data Transfer Diagram

18.4.4 Error Detection

Writing to SPD during the transmission of a data sequence will lead to a write collision, resulting in the setting of the WCOL bit. This will not trigger an interrupt, and the transmission will not stop, and the WCOL bit needs to be cleared by software.

18.5 SPI0 and SPI1 Comparison

Comparison BIT	SPI0	SPI1
Signal Port Strong Driving	Available	None
WCOL	When the send FIFO is full, attempting to write to the FIFO will fail, and WCOL will be set, indicating a buffer write conflict	When one frame is sending, attempting to write to the FIFO will fail, and WCOL will be set, indicating a buffer write conflict
SPIF	This position being set indicates the completion of receiving/sending one frame of data	None
QTWIF	None	This position being set indicates the completion of receiving/sending one frame of data
RXHIE	Interrupt enable bit for the valid data in the recieve FIFO is more than half	None
TXHIE	Interrupt enable bit for the valid data in the transmit FIFO is less than half	None
RXIE	Interrupt enable bit for the receive FIFO full	None
TBIE	Interrupt enable bit for the transmit FIFO empty	Interrupt enable bit for the transmit FIFO empty
RXNEIE	Interrupt enable bit for the receive FIFO not empty	None

Comparison BIT	SPI0	SPI1
RXHIF	Set when the valid data in the receive FIFO is more than half	None
TXHIF	Set when the valid data in the receive FIFO is less than half	None
RXFIF	Set when the receive FIFO is full	None
TXEIF	Set when the receive FIFO is empty	Set when the receive FIFO is empty
RXNEIF	Recieve FIFO not empty flag	None
DMA	Triggering DMA requests through the TXEIF flag and the RXNEIF flag	A request is uniformly set at the end of a frame

18.6 SPI Interrupt

As for SPI0, interrupts will be generated when “transmission complete”, “FIFO half transmit”, or “transmit buffer empty”. Separate interrupt enable bits can be used to enhance flexibility.

Interrupt Event	Event Flag	Interrupt Request Control Bit	Sub-Event Flag	Interrupt Enable Sub-Switch
The valid data in the recieve FIFO is more than half	SPIF	SPI0_IDE ->INTEN	RXHIF	RXHIE
The valid data in the transmit FIFO is less than half			TXHIF	TXHIE
The receive FIFO is full			RXFIF	RXIE
The transmit FIFO is empty			TXEIF	TBIE
The receive FIFO is not full			RXNEIF	RXNEIE

As for SPI1, interrupts will be generated when “transmission complete”, or “transmit buffer empty”. Separate interrupt enable bits can be used to enhance flexibility.

Interrupt Event	Event Flag	Interrupt Request Control Bit	Sub-Event Flag	Interrupt Enable Sub-Switch
One frame transmit/receive complete	QTWIF	SPI1_TWI1_IDE ->INTEN	\	\
The transmit buffer is empty			TXEIF	TBIE

18.7 SPI0 Register

18.7.1 SPI Related Register

18.7.1.1 SPI0 Control Register (SPI0_CON)

Register	R/W	Description	Reset Value	POR
SPI0_CON	R/W	SPI0 Control Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	SPOS	-	SPR[3:0]			
7	6	5	4	3	2	1	0
SPEN	-	-	CPOL	CPHA	DORD	SPMD	MSTR

<table>
<tr><th>Bit number</th><th>Bit Mnemonic</th><th>Description</th></tr>
<tr><td>13</td><td>SPOS</td><td>SPI0 Port Mapping Control Bit
<table>
<tr><th>Port
SPOS Value</th><th>MISO0</th><th>MOSI0</th><th>SCK0</th></tr>
<tr><td>SPOS=0</td><td>PC1</td><td>PC2</td><td>PC3</td></tr>
<tr><td>SPOS=1</td><td>PB5</td><td>PB6</td><td>PB7</td></tr>
</table></td></tr>
<tr><td>11~8</td><td>SPR[3:0]</td><td>SPI Clock Presclar Control Bit
0000: f_{PCLK0}
0001: f_{PCLK0} /2
0010: f_{PCLK0} /4
0011: f_{PCLK0} /8
0100: f_{PCLK0} /16
0101: f_{PCLK0} /32
0110: f_{PCLK0} /64
0111: f_{PCLK0} /128
1000: f_{PCLK0} /256
1001: f_{PCLK0} /512
1010: f_{PCLK0} /1024
Others: f_{PCLK0} /1024
Note: To ensure correct communication for the SPI0 in the SC32F15G series, the communication frequency should be selected to be below 16 MHz.</td></tr>
<tr><td>7</td><td>SPEN</td><td>SPI Enable Control Bit
0: Disable SPI0
1: Enable SPI0</td></tr>
</table>

Bit number	Bit Mnemonic	Description
4	CPOL	SPI Clock Polarity Control Bit 0: SCK is at low level in the idle state 1: SCK is at high level in the idle state
3	CPHA	SPI Clock Phase Control Bit 0: Capture data at the first edge of SCK 1: Capture data at the second edge of SCK
2	DORD	SPI Transmission Direction Selection Bit 0: MSB sending priority 1: LSB sending priority
1	SPMD	SPI Transmission Mode Selection Bit 0: 8-bit mode 1: 16-bit mode
0	MSTR	SPI Master/Slave Selection Bit 0: SPI0 is slave device 1: SPI0 is master device
31~14 12 6~5	-	Reserved

18.7.1.2 SPI0 Flag Register (SPI0_STS)

Register	R/W	Description	Reset Value	POR
SPI0_STS	R/W	SPI0 Flag Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
WCOL	-	TXHIF	RXHIF	RXFIF	TXEIF	RXNEIF	SPIF

Bit number	Bit Mnemonic	Description
7	WCOL	Write Conflict Flag This bit is set to 1 by hardware, and cleared by writing 1 through software, indicating whether a write conflict has occured: 0: No writing conflict detected 1: Writing conflict detected
5	TXHIF	Valid Data In Transmit FIFO Is Less Than Half Flag This bit is read only, and can be set or cleared by hardware, indicating the current status of the transmit FIFO: 0: The valid data in the transmit FIFO is not less than half

Bit number	Bit Mnemonic	Description
		1: The valid data in the transmit FIFO is less than half, an interrupt will be generated if TXHIE=1
4	RXHIF	Valid Data In Recieve FIFO Is More Than Half Flag This bit is read only, and can be set or cleared by hardware, indicating the current status of the receive FIFO: 0: The valid data in the recieve FIFO is not more than half flag 1: The valid data in the recieve FIFO is more than half flag, an interrupt will be generated if RXHIE=1
3	RXFIF	Receive FIFO Is Full Fag This bit is read only, and can be set or cleared by hardware, indicating whether current recieve FIFO is full: 0: Receive FIFO is not full 1: Receive FIFO is full
2	TXEIF	Transmit FIFO Is Empty Flag This bit is set to 1 by hardware, and cleared by writing 1 through software, indicating whether current transmit FIFO is empty: 0: Transmit FIFO is not empty 1: Transmit FIFO is empty **Note: In DMA mode, after DMA writes to the transmit buffer, this bit is cleared by the DMA module, and users do not need to clear it through software.**
1	RXNEIF	Receive FIFO Is Not Full Flag This bit is read only, and can be set or cleared by hardware, indicating whether current receive FIFO is empty: 0: Receive FIFO is empty 1: Receive FIFO is not empty
0	SPIF	SPI Data Transmission Flag This bit is set to 1 by hardware, and cleared by writing 1 through software, indicating whether current SPI transmission is complete: 0: Data transmission is ongoing 1: Data transmission is complete
31~8 6	-	Reserved

18.7.1.3 SPI0 Data Register (SPI0_DATA)

Register	R/W	Description	Reset Value	POR
SPI0_DATA	R/W	SPI0 Data Register	0x0000_0000	0x0000_0000

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

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

Bit number	Bit Mnemonic	Description
15~0	SPD[15:0]	SPI Data Buffer Read operation: Read the received data from the SPI0 receive FIFO Write operation: write data to the SPI0 transmit FIFO
31~16	-	Reserved

18.7.1.4 SPI0 Interrupt Enable And DMA Control Register (SPI0_IDE)

Register	R/W	Description	Reset Value	POR
SPI0_IDE	R/W	SPI0 Interrupt Enable And DMA Control Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
TXDMAEN	RXDMAEN	TXHIE	RXHIE	RXIE	TBIE	RXNEIE	INTEN

Bit number	Bit Mnemonic	Description
7	TXDMAEN	DMA Transmit Channel Enable Bit 0: Disable DMA transmit function 1: Enable DMA transmit function The set of TXEIF can trigger DMA channel transmit request after enabling this bit.
6	RXDMAEN	DMA Recieve Channel Enable Bit 0: Disable DMA receive function 1: Enable DMA receive function The set of RXNEIF can trigger DMA channel receive request after enabling this bit.
5	TXHIE	Valid Data In Transmit FIFO Is Less Than Half Interrupt Enable Bit 0: An interrupt will not be generated when TXHIF is set 1: An interrupt will be generated when TXHIF is set
4	RXHIE	Valid Data In Recieve FIFO Is More Than Half Interrupt Enable Bit 0: An interrupt will not be generated when RXHIF is set 1: An interrupt will be generated when RXHIF is set

Bit number	Bit Mnemonic	Description
3	RXIE	Receive FIFO Is Full Interrupt Enable Bit 0: An interrupt will not be generated when RXFIF is set 1: An interrupt will be generated when RXFIF is set
2	TBIE	Transmit FIFO Is Empty Interrupt Enable Bit 0: An interrupt will be not generated when TXEIF is set 1: An interrupt will be generated when TXEIF is set
1	RXNEIE	Receive FIFO Is Not Full Interrupt Enable Bit 0: An interrupt will not be generated when RXNEIF is set 1: An interrupt will be generated when RXNEIF is set
0	INTEN	Interrupt Request CPU Enable Control Bit 0: Disable interrupt request 1: Enable interrupt request
31~8	-	Reserved

18.7.2 SPI0 Register Mapping

Register	Offset Address	R/W	Description	Reset Value	POR	Access Restriction
SPI0 Base Address:0x4002_0040						
SPI0_CON	0x00	R/W	SPI0 Control Register	0x0000_0000	0x0000_0000	
SPI0_STS	0x04	R/W	SPI0 Flag Register	0x0000_0000	0x0000_0000	
SPI0_DATA	0x0C	R/W	SPI0 Data Register	0x0000_0000	0x0000_0000	Do not support byte/half word access
SPI0_IDE	0x10	R/W	SPI0 Interrupt Enable And DMA Control Register	0x0000_0000	0x0000_0000	

18.8 SPI1 Register

18.8.1 SPI1 Related Register

18.8.1.1 SPI1_TWI1 Dual-Function Control Register (SPI1_TWI1_CON)

Register	R/W	Description	Reset Value	POR
SPI1_TWI1_CON	R/W	SPI1 Dual-Function Control Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
MODE[1:0]		-	CPOL	CPHA	DORD	SPMD	SMSTR
15	14	13	12	11	10	9	8
-	SPOS[1:0]		-	QTWCK[3:0]			
7	6	5	4	3	2	1	0
QTWEN	-	START	STOP	-	-	AA	STRETCH

Bit number	Bit Mnemonic	Description
23~22	MODE[1:0]	SPI1_TWI1 Mode Selection Bit 00: TWI Mode 01: SPI Mode 10: Reserved 11: Reserved
20	CPOL	SPI Clock Polarity Control Bit 0: SCK is at low level in the idle state 1: SCK is at high level in the idle state
19	CPHA	SPI Clock Phase Control Bit 0: Capture data at the first edge of SCK 1: Capture data at the second edge of SCK
18	DORD	SPI Transmission Direction Selection Bit 0: MSB sending priority 1: LSB sending priority
17	SPMD	SPI Transmission Mode Selection Bit 0: 8-bit mode 1: 16-bit mode
16	SMSTR	SPI Master/Slave Selection Bit 0: SPI1 is slave device 1: SPI1 is master device
14~13	SPOS[1:0]	SPI1 Port Mapping Control Bit

Port SPOS Value	MISO1	MOSI1	SCK1
SPOS[1:0]=00	PC0	PB15	PB14
SPOS[1:0]=01	PA8	PA7	PA6
SPOS[1:0]=10	PA13	PA12	PA11
SPOS[1:0]=11	PB4	PB3	PB2

Bit number	Bit Mnemonic	Description
11~8	SPR[3:0]	SPI Clock Presclar Control Bit 0000: f_{PCLK1} 0001: f_{PCLK1} /2 0010: f_{PCLK1} /4 0011: f_{PCLK1} /8 0100: f_{PCLK1} /16 0101: f_{PCLK1} /32 0110: f_{PCLK1} /64 0111: f_{PCLK1} /128 1000: f_{PCLK1} /256 1001: f_{PCLK1} /512 1010: f_{PCLK1} /1024 Others: f_{PCLK1} /1024 Note: To ensure correct communication for the SPI0 in the SC32F15G series, the communication frequency should be selected to be below 8 MHz.
7	QTWEN	SPI1_TWI1 Module Enable Control Bit 0: Disable module 1: Enable module
31~24 21 15 12 6 3~2	-	Reserved

18.8.1.2 SPI1_TWI1 Dual-Function Flag Register (SPI1_TWI1_STS)

Register	R/W	Description	Reset Value	POR
SPI1_TWI1_STS	R/W	SPI1_TWI1 Dual-Function Flag Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
NBYTES[7:0]							
15	14	13	12	11	10	9	8
-	-	-	-	-	STATE[2:0]		
7	6	5	4	3	2	1	0
WCOL	TXEIF	-	-	TMSTR	GCA	TXnE/RXnE	QTWIF

Bit number	Bit Mnemonic	Description
7	WCOL	Write Conflict Flag This bit is set to 1 by hardware, and cleared by writing 1 through software, indicating whether a write conflict has occured: 0: No writing conflict detected 1: Writing conflict detected
6	TXEIF	Transmit Buffer Is Empty Flag This bit is set to 1 by hardware, and cleared by writing 1 through software, indicating whether current transmit Buffer is empty: 0: Transmit Buffer is not empty 1: Transmit Buffer is empty **Note: In DMA mode, after DMA writes to the transmit buffer, this bit is cleared by the DMA module, and users do not need to clear it through software.**
0	QTWIF	SPI Data Transmission Flag This bit is set to 1 by hardware, and cleared by writing 1 through software, indicating whether current SPI transmission is complete: 0: Data transmission is ongoing 1: Data transmission is complete
31~24 15~11 5~4	-	Reserved

18.8.1.3 SPI1_TWI1 Dual-Function Data Register (SPI1_TWI1_DATA)

Register	R/W	Description	Reset Value	POR
SPI1_TWI1_DATA	R/W	SPI1_TWI1 Dual-Function Data Register	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
15~0	QTWIDAT [15:0]	SPI Data Buffer Read operation: Read the received data from the SPI1 receive buffer Write operation: write data to the SPI1 transmit buffer
31~16	-	Reserved

18.8.1.4 SPI1_TWI1 Dual-Function Interrupt Enable And DMA Control Register (SPI1_TWI1_IDE)

Register	R/W	Description	Reset Value	POR
SPI1_TWI1_IDE	R/W	SPI1_TWI1 Dual-Function Interrupt Enable And DMA Control Register	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
7	TXDMAEN	DMA Transmit Channel Enable Bit 0: Disable DMA transmit function 1: Enable DMA transmit function The set of TXEIF can trigger DMA channel transmit request after enabling this bit.
6	RXDMAEN	DMA Recieve Channel Enable Bit 0: Disable DMA receive function 1: Enable DMA receive function The set of QTWIF can trigger DMA channel receive request after enabling this bit.
1	TBIE	Transmit Buffer Is Empty Interrupt Enable Bit 0: An interrupt will be not generated when TXEIF is set 1: An interrupt will be generated when TXEIF is set
0	INTEN	Interrupt Request CPU Enable Control Bit 0: Disable interrupt request 1: Enable interrupt request
31~8 5~2	-	Reserved

18.8.2 SPI1 Register Mapping

Register	Offset Address	R/W	Description	Reset Value	POR	Access Restriction
SPI1_TWI1 Base Address:0x4002_1040						
SPI1_TWI1_CON	0x00	R/W	SPI1_TWI1 Dual-Function Control Register	0x0000_0000	0x0000_0000	

Register	Offset Address	R/W	Description	Reset Value	POR	Access Restriction
SPI1_TWI1_STS	0x04	R/W	SPI1_TWI1 Dual-Function Flag Register	0x0000_0000	0x0000_0000	
SPI1_TWI1_DATA	0x0C	R/W	SPI1_TWI1 Dual-Function Data Register	0x0000_0000	0x0000_0000	Do not support byte/half word access
SPI1_TWI1_IDE	0x10	R/W	SPI1_TWI1 Dual-Function Interrupt Enable And DMA Control Register	0x0000_0000	0x0000_0000	

19 TWI0~1

19.1 Clock Source

The SC32F15G series TWI has only one clock source, which is derived from PCLK

19.2 TWI0 Feature

- Supports 11-stage TWI clock pre-scaling
- Signal ports can be mapped to two additional set of ports
- Support master/slave mode
- Bidirectional data transmission between master and slave
- Communication speed can reach up to 1 Mbps
- Support DMA

19.3 TWI1 Feature

- SPI1 and TWI1 operate independently with multiplexed register addresses and signal pins
- Supports 11-stage TWI clock pre-scaling
- Signal ports can be mapped to two additional set of ports
- Support master/slave mode
- Bidirectional data transmission between master and slave
- Communication speed can reach up to 1 Mbps

19.4 TWI Functhion Description

19.4.1 TWI Signal Description

On the TWI bus, data is synchronously transmitted between the master and slave devices using the clock line (SCL) and the data line (SDA). Each data byte has a length of 8 bits, and one data bit is transferred with each SCL clock pulse. The data is transmitted starting from the most significant bit (MSB), and after each byte, an acknowledgment bit follows. Each bit is sampled when SCL is high. Therefore, the SDA line may change when SCL is low, but it must remain stable when SCL is high. When SCL is high, any transition on the SDA line is considered a command (START or STOP)

TWI Clock Signal Line (SCL):

The clock signal is generated by the master and is connected to all the slaves. It transmits one byte of data every 9 clock cycles. The first 8 cycles are used for data transmission, and the last one is used as the acknowledgment clock for receiver. It should be pulled up by the pull-up resistor on the SDA line when idle.

TWI Data Signal Line (SDA):

SDA is a bidirectional signal line and should be pulled up by the pull-up resistor on the SDA line when idle.

19.4.2 Slave Operating Mode

- **Mode Initiation:**

When TWEN/QTWEN = 1 and the slave receives the start signal sent by the master, the mode will be initiated.

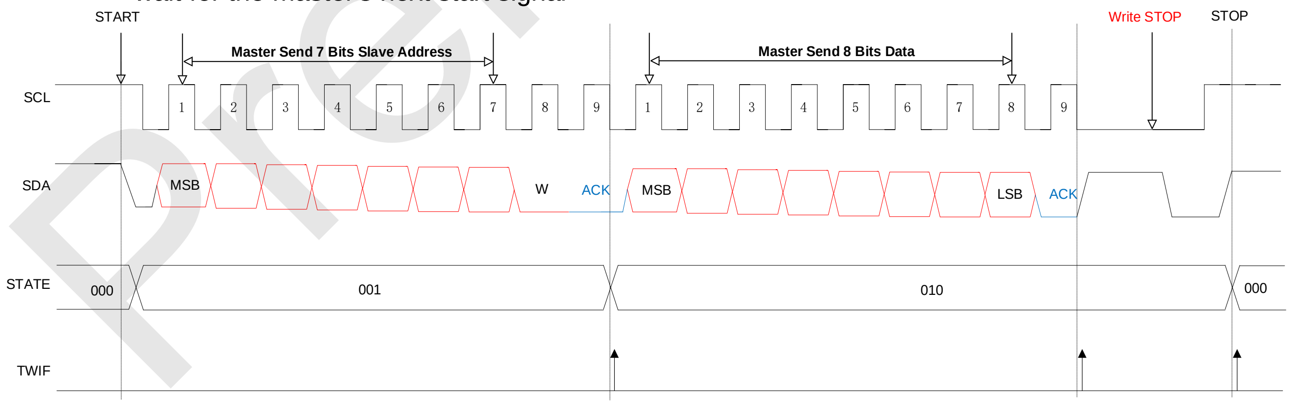
The slave from idle state (STATE[2:0] = 000) change to first frame address reception state (STATE[2:0] = 001), waiting for the master's first frame of data. The first frame of data is sent by the master and includes 7 address bits and 1 read/write bit. All slaves on the TWI bus can receive the master's first frame of data. The SDA signal line will be released after transmiting the first frame of data. If the address sent by the master matches the value in the slave's own address register, the selected slave will be selected and will check the 8th bit on the bus, which is the data read/write bit (1 for read command; 0 for write command). The selected slave then holds the SDA signal line, gives a low-level acknowledgment signal to the master on the 9th clock cycle, and then releases the bus. After being selected, the slave will enter different states depending on the read/write bit:

- **Non-general call address response, slave reception mode:**

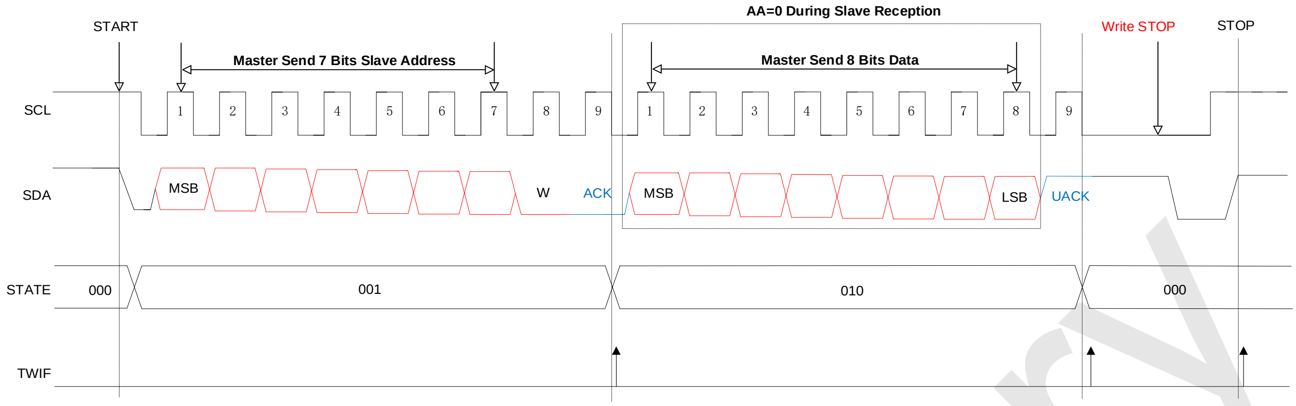
If the read/write bit received in the first frame is a write (0), the slave will enter the slave receive state (STATE[2:0] = 010) to wait for the master to transmit data. The bus will be released every 8 bits the master transmits, and the slave awaits the 9th clock cycle for the acknowledgment signal.

If the slave's acknowledgment signal is low, the master will have the following three actions:

① Continue transmiting data
② Resend the start signal, at which point the slave re-enters the reception of the first address frame state (STATE[2:0] = 001).
③ Transmit a stop signal, indicating the end of this transmission. The slave will return to the idle state and wait for the master's next start signal



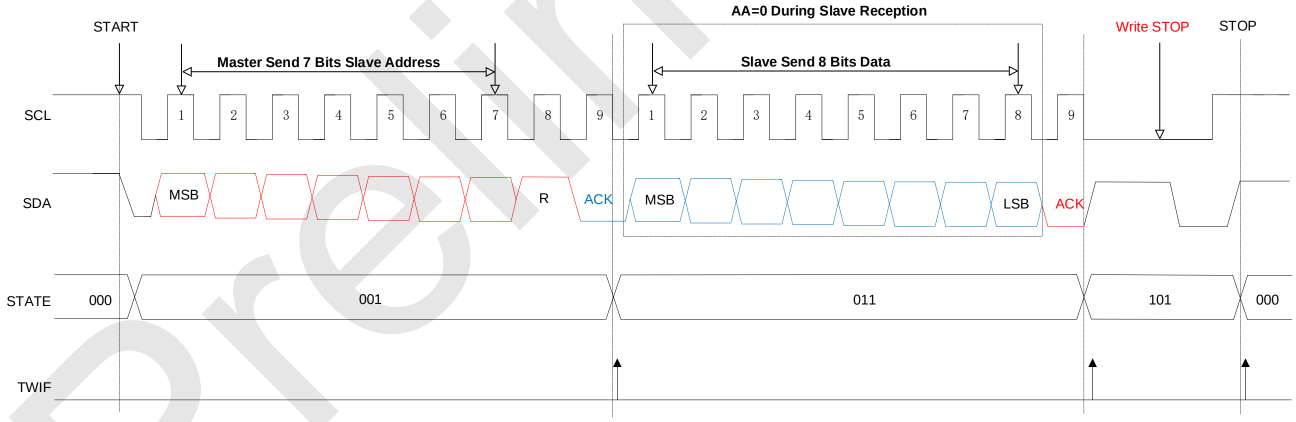
If the slave responds with a high level (during the reception process, the value of AA in the slave's register will be rewritten to 0), it indicates that after the current byte transmission is complete, the slave will actively terminate this transmission, returning to the idle state (STATE[2:0] = 000), and will no longer receive data sent by the master



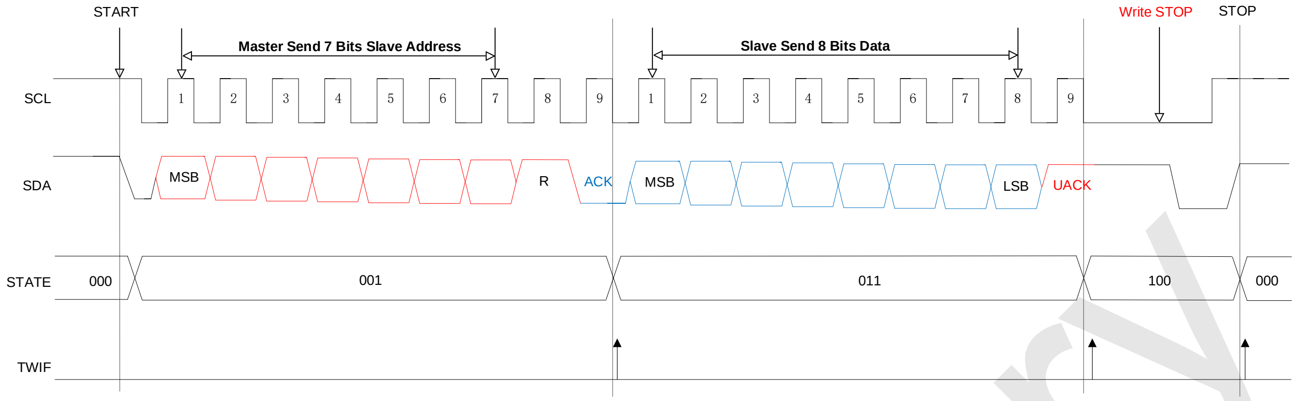
- **Non-general call address response, slave transmission mode:**

If the read/write bit received in the first frame is read, the slave will occupy the bus and transmit data to the master. After transmiting each 8 bits of data, the slave will release the bus and wait for the master's acknowledgment:

If the master responds with a low level, the slave will continue to transmit data. During the transmission, if the value of AA in the slave register is modified to 0, the slave will actively terminate the transmission and release the bus after completing the current byte transmission. It then waits for the master's stop signal or a restart signal (STATE[2:0] = 101).



If the master responds with a high level, the slave’s STATE[2:0] = 100. It then waits for the master's stop signal or a restart signal.

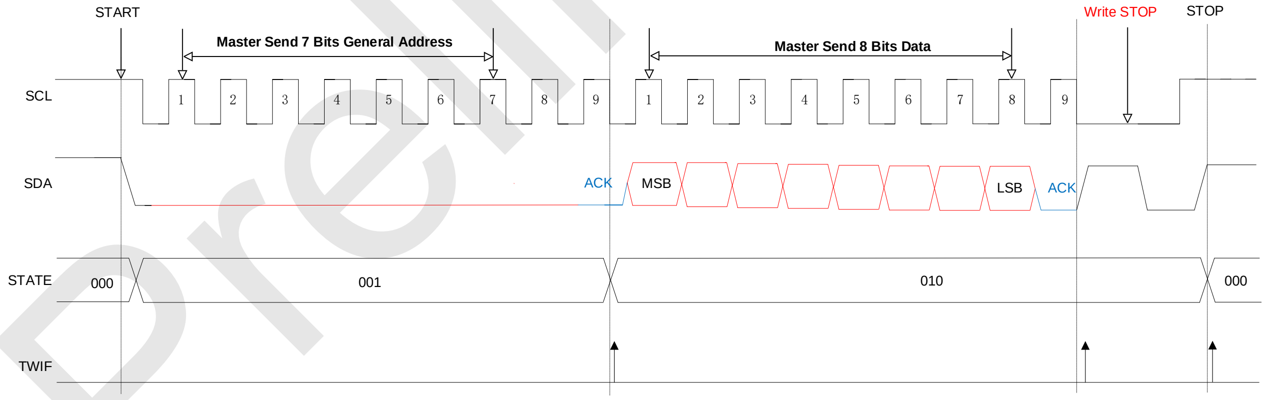


- **General call address response:**

When GC is set to 1, general call address will be allowed. the slave enters the state of receiving the first address frame (STATE[2:0] = 001). If the address bits of first frame is 0x00, all slave will respond to the master. The master transmits a read/write bit, which must be set to write. All slaves then enter the state of receiving data (STATE[2:0] = 010). The master will release the SDA line every 8 data transmissions and read the status on the SDA line:

If there is a slave acknowledgment, the master will have the following three actions:

① Continue transmiting data
② Restart the communication
③ Transmit a stop signal, indicating the end of this transmission



If there is no acknowledgment from any slave, SDA line will be in idle state

Note: In one master multiple slaves mode using a general call address, the read/write bit sent by the master must not be set to read. Otherwise, all devices on the bus will respond, except the one transmiting data.

19.4.3 Slave Mode Operation Steps

① Configure TWI control register (TWI0_CON.TWEN = 1/SPI1_TWI1_CON.QTWEN = 1) to enable TWI
② Configure TWI control register (TWI0_CON/SPI1_TWI1_CON)
③ Configure TWI address register (TWI0_ADD/TWI1_ADD)
④ If slave recieve data, wait for the interrupt flag TWIF in the TWI status register

(TWI0_STS/SPI1_TWI1_STS) to be set to 1. The TWIF/QTWIF flag will be set each time the slave receives 8 bits of data, and TWIF need to be cleared manually.

⑤ If slave transmit data, write the data to be transmitted into the TWI data register (TWIDAT/QTWIDAT), than TWI will automatically transmit the data, and the TWIF flag will be set for every 8 bits transmitted.

19.4.4 Master Operating Mode

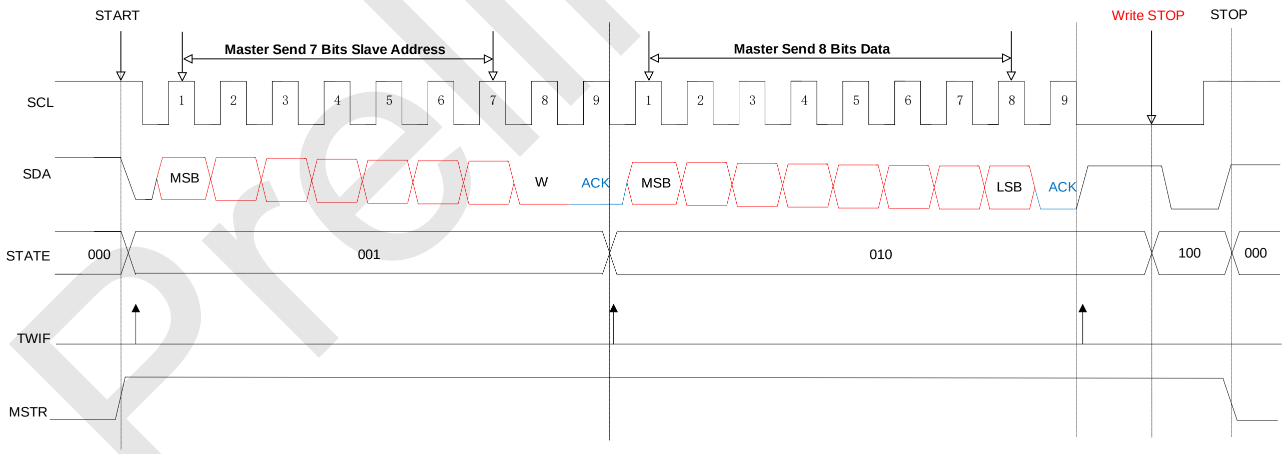
- **Mode Initiation:**

When the TWI interface transmits a start condition to the bus, it automatically switches to master mode, and the hardware will set the MSTR/TMSTR bit to 1. The master status bits STATE[2:0] change from 000 to 001, and simultaneously, the interrupt condition TWIF/QTWIF is set to 1.

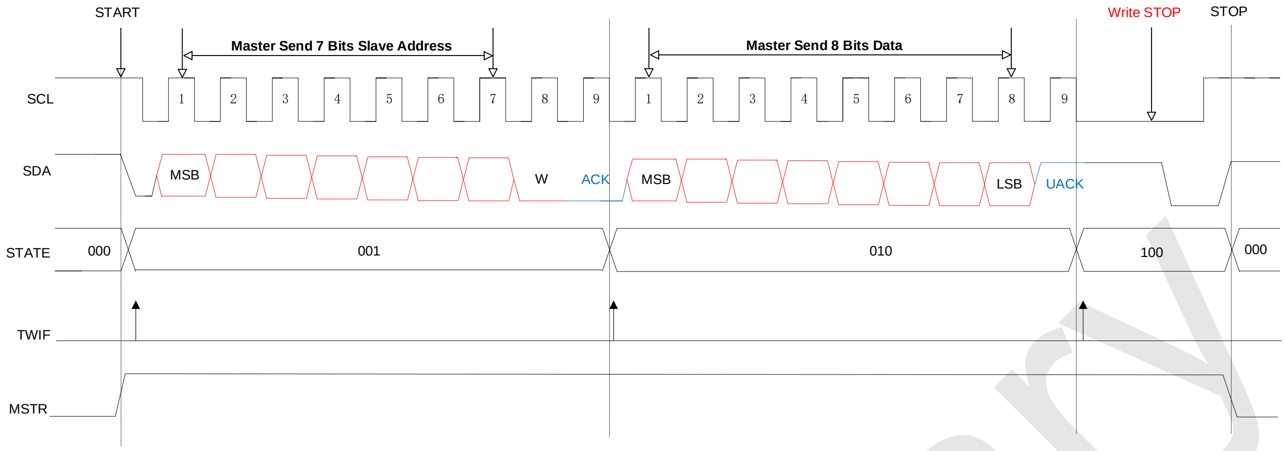
- **TWI Master Transmission Mode:**

In the master transmission mode, the first frame of data sent by the master includes 7 bits of address (the address of the selected slave) and 1 bit of read/write indicator (0 for write command). All slaves on the TWI bus will receive this first frame of data from the master. After transmiting the first frame, master will release the SDA signal line. The selected slave, upon receiving the first frame, responds to the master with an acknowledgment signal on the 9th clock cycle of the SCL. Afterward, the slave releases the bus and enters the slave receive state to await the reception of data from the master. The master will release the bus after transmiting each 8 bits, then wait for the acknowledgment signal from the slave on the 9th cycle.

If the slave responds with a low level, the master can continue transmiting data. It can also resend the start signal:



If the slave responds with a high level, it indicates that the current byte transmission is complete, and the slave will actively terminate the current transmission. The slave will no longer receive data from the master, and the master's STATE[2:0] will change from 010 to 100:

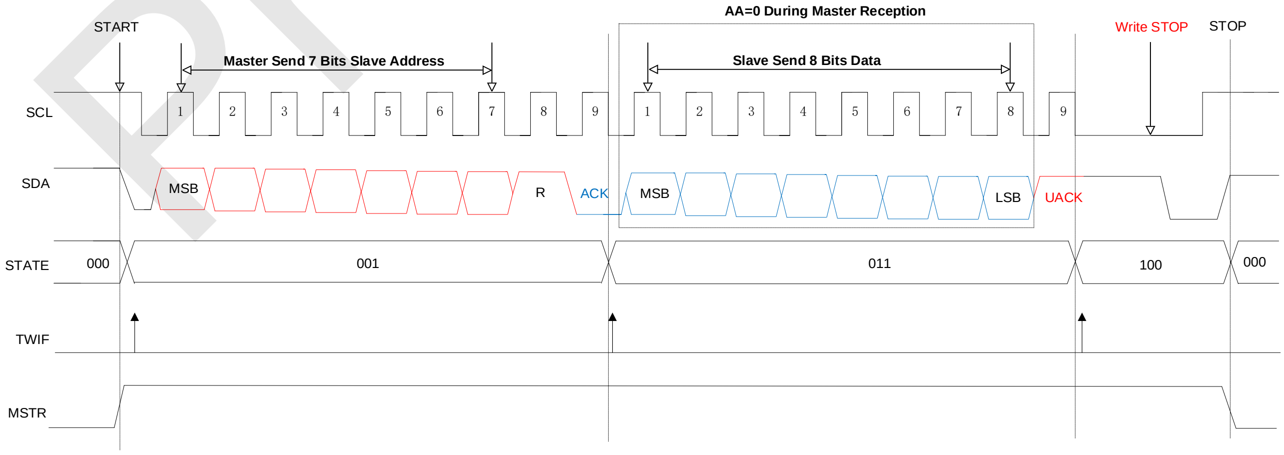


- **TWI Master Reception Mode:**

In master transmission mode, the first frame of data sent by the master includes 7 address bits (the address of the selected slave) and 1 read/write bit (1 for read command). All slaves on the TWI bus will receive this first frame of data from the master. After transmiting the first frame of data, the master will release the SDA signal line. The selected slave will respond to the master with an acknowledgment signal on the 9th clock cycle of SCL. Subsequently, the slave will occupy the bus and transmit data to the master. After transmiting 8 bits of data, the slave will release the bus and wait for the master's acknowledgment. Upon receiving a successful acknowledgment (ACK) from the slave after matching address, the master will begin to receive data from the slave (STATE=011):

1. If the master acknowledgment bit is enabled (AA=1), the master will respond with an acknowledgment signal (ACK) after receiving each byte of data, and TWIF/QTWIF will be set.
2. Before receiving the last byte of data, if the acknowledgment enable bit is disabled (AA=0), the master will respond with a unacknowledge (UACK) after receiving the last byte of data. Then, the master can transmit a stop signal.

In master receiving mode, the method for actively releasing the bus is as follows:



19.4.5 Master Mode Operation Steps

① Configure TWI control register (TWI0_CON.TWEN = 1/SPI1_TWI1_CON.QTWEN = 1) to enable TWI

② Configure TWI control register (TWI0_CON/SPI1_TWI1_CON): confiugure TWI communication rate bit(TWCK/QTWCK[3:0]) and set start bit STA/START to “1”

③ Configure the TWI address register (TWI0_ADD/TWI1_ADD): Write the slave address and read/write bit into TWIDAT to transmit address frame on the bus

④ If the master is receiving data, wait for the interrupt flag TWIF/QTWIF in the TWI0_STS/SPI1_TWI1_STS to be set to 1. The interrupt flag will be set to 1 for every 8 bits of data received and need to be cleared manually

⑤ If the master is sending data, write the data to be transmitted into TWIDAT/QTWIDAT. TWI will automatically transmit the data. The interrupt flag TWIF/QTWIF will be set to 1 for every 8 bits transmitted

⑥ Once data reception or transmission is complete, the master can transmit a stop signal (STO/STOP=1), and the master's state transitions to 000. Alternatively, the master can transmit a repeated start signal to begin a new round of data transmission.

Note: The master's TWIF/QTWIF flag will not be set after generating a stop condition!

19.5 TWI0 Interrupt

For TWI0, the following events can trigger an interrupt. All TWI events share a common interrupt flag

Interrupt Event	Event Flag	Interrupt Request Control Bit
Master mode: start signal transmission complete	TWIF	TWI0_IDE ->INTEN
Master mode: address frame transmission complete		
Master mode: data frame reception or transmission complete		
Slave mode: first frame address successfully match		
Slave mode: successfully receive or transmit 8 bits data		
Slave mode: receive restart signal		
Slave mode: receive stop signal		

19.6 TWI1 Interrupt

For TWI1, the following events can trigger an interrupt. All TWI events share a common interrupt flag

Interrupt Event	Event Flag	Interrupt Request Control Bit
Master mode: start signal transmission complete	qTWIF	SPI1_TWI1_IDE ->INTEN
Master mode: address frame transmission complete		
Master mode: data frame reception or transmission complete		
Slave mode: first frame address successfully match		
Slave mode: successfully receive or transmit 8 bits data		
Slave mode: receive restart signal		
Slave mode: receive stop signal		

19.7 TWI0 Register

19.7.1 TWI0 Related Register

19.7.1.1 TWI0 Control Register (TWI0_CON)

Register	R/W	Description	Reset Value	POR
TWI0_CON	R/W	TWI0 Control Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	SPOS[1:0]		-	TWCK[3:0]			
7	6	5	4	3	2	1	0
TWEN	-	STA	STO	-	-	AA	STRETCH

<table>
<tr><th>Bit number</th><th>Bit Mnemonic</th><th>Description</th></tr>
<tr><td>14~13</td><td>SPOS[1:0]</td><td>● TWI0 Port Mapping Control Bit@TWI0_CON
<table>
<tr><th>Port
SPOS Value</th><th>SCL0</th><th>SDA0</th></tr>
<tr><td>SPOS[1:0]=00</td><td>PA0</td><td>PA1</td></tr>
<tr><td>SPOS[1:0]=01</td><td>PB7</td><td>PB6</td></tr>
<tr><td>SPOS[1:0]=10</td><td>PB3</td><td>PB4</td></tr>
</table></td></tr>
<tr><td>11~8</td><td>TWCK[3:0]</td><td>TWI Master Mode Clock Presclar Control Bit:
0000: f_{PCLK} /4096
0001: f_{PCLK} /2048
0010: f_{PCLK} /1024
0011: f_{PCLK} /512
0100: f_{PCLK} /256
0101: f_{PCLK} /128
0110: f_{PCLK} /64
0111: f_{PCLK} /32
1000: f_{PCLK} /16
1001: f_{PCLK} /8
1010: f_{PCLK} /4
Others: f_{PCLK} /4
f_{PCLK} = f_{PCLK0} while choose TWI0 as master.
Note: When TWI0 operates as a master, the maximum frequency is 4 MHz; when it operates as a slave, the maximum frequency is also 4 MHz.</td></tr>
<tr><td>7</td><td>TWEN</td><td>TWI Enable Control Bit</td></tr>
</table>

Bit number	Bit Mnemonic	Description
		0: Disable TWI 1: Enable TWI
5	STA	TWI Initial Position Trigger Switch Start condition will be generated when this bit is set to 1, and the TWI will switch to master mode. Software can set or clear this bit, or it can be cleared by hardware after the start condition is issued.
4	STO	TWI Stop Bit Trigger Switch In master mode, writing 1 to this bit will generate a stop condition after the current byte transmission or the start condition is issued. Software can set or clear this bit, or it can be cleared by hardware when a stop condition is detected.
1	AA	TWI Acknowledge Enable Bit 0: No acknowledgment, returns UACK (acknowledge bit is high level). 1: Returns an acknowledgment (ACK) after receiving a matching address or data
0	STRETCH	TWI Clock Stretching Enable Bit This bit is only valid in slave mode. 0: Disable clock stretching. 1: Enable clock stretching, and the master needs to support clock stretching. Description: Clock stretching will occur after data transmission is complete and ACK is 0.
31~15 12 6,3~2	-	Reserved

19.7.1.2 TWI0 Status Flag Register (TWI0_STS)

Register	R/W	Description	Reset Value	POR
TWI0_STS	R/W	TWI0 Status Flag Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
NBYTES[7:0]							
15	14	13	12	11	10	9	8
-	-	-	-	-	STATE[2:0]		
7	6	5	4	3	2	1	0
-	-	-	-	MSTR	GCA	TXnE/RXnE	TWIF

Bit number	Bit Mnemonic	Description
23~16	NBYTES[7:0]	Transmission/Reception Buffer Number Setting Bit

Bit number	Bit Mnemonic	Description
		Used to set the number of bytes to be transmitted/received. For each successful transmission/reception, NBYTES will automatically decrease by 1. When NBYTES reaches 0, the TC flag will be set. **Note: Modification is not allowed when STA is set to 1.**
10~8	STATE[2:0]	TWI Status Bits Used to indicate the TWI status, with different meanings in master/slave mode ● Slave mode: 000: Slave is in idle state, waiting for TWEN to be set, detecting the TWI start signal. The slave will transit to this state after receiving a stop condition 001: Slave is receiving the first frame of address and read/write bit (the 8th bit is the read/write bit, 1 for read, 0 for write). The slave will transit to this state after receiving the start condition 010: Slave is in the data reception state 011: Slave is in the data transmission state 100: Slave will transit to this state when the master responds with UACK in the data transmission state, waiting for a restart signal or stop signal 101: Slave will transit to this state when writing AA to 0 in transmission state, waiting for a restart signal or stop signal 110: Slave will transit to this state if the slave's address does not match the address sent by the master, waiting for a new start condition or stop condition ● Master mode: 000: State machine is in idle state 001: Master is transmitting the start condition or the address of slave device 010: Master is transmitting data 011: Master is receiving data 100: Master has transmitted the stop condition or received the UACK signal from the slave
3	MSTR	TWI Master/Slave Mode Flag Bit 0: Slave mode 1: Master mode Description: 1. When the TWI interface transmit a start condition to the bus, it automatically switches to master mode, and the hardware will set this bit. 2. When a stop condition is detected on the bus, the hardware will clear this bit.
2	GCA	TWI General Call Address Response Flag Bit

Bit number	Bit Mnemonic	Description
		0: Non-response to general call address 1: When GC is set to 1 and there is a match with the general call address, this bit will set to 1 by the hardware and then automatically cleared
1	TXnE/RXnE	TWI Transmission Complete Flag Bit TXnE/RXnE will be set to 1 by the hardware in the following cases ● Master mode: ■ Master transmits an address frame (write), and receives ACK from the slave ■ Master finishes sending data and receives ACK from the slave ■ Master receives data and responds with ACK to the slave ● Slave mode: ■ Slave receives an address frame (read), and the received address matches the slave address (TWA) ■ Slave receives data and responds with ACK to the master ■ Slave finishes sending data and receives ACK from the master (AA=1) After a read/write operation on TWIDAT, this bit will be cleared by the hardware
0	TWIF	TWI Interrupt Flag Bit This bit is set to 1 by the hardware and can be cleared by writing 1 through software ● Master mode: ■ Transmit start signal ■ Finish transmitting the address frame ■ Receive or finish transmitting a data frame ● Slave mode: ■ Successful match of the first address frame ■ Successfully receive or transmit 8 bits of data ■ Receive a repeated start condition ■ Slave receives a stop signal
31~24 15~11 7~4	-	Reserved

19.7.1.3 TWI0 Address Register (TWIn_ADD)

Register	R/W	Description	Reset Value	POR
TWI0_ADD	R/W	TWI0 Address Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24

-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
TWA[6:0]							GC

Bit number	Bit Mnemonic	Description
7~1	TWA[6:0]	TWI Address Register TWA[6:0] cannot be written as all 0; 00H is reserved for general call address. This bit is not valid in master mode
0	GC	TWI General Call Address Response Enable Bit 0: Disable response to general call address 00H 1: Enable response to general call address 00H
31~8	-	Reserved

19.7.1.4 TWI0 Data Register (TWI0_DATA)

Register	R/W	Description	Reset Value	POR
TWI0_DATA	R/W	TWI0 Data Register	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
7~0	TWIDAT[7:0]	TWI Data Buffer Read operation: Read the received data from the TWI reception buffer. Write operation: Write the data to be transmitted into the TWI transmission buffer.
31~8	-	Reserved

19.7.1.5 TWI0 Interrupt Enable And DMA Control Register (TWI0_IDE)

Register	R/W	Description	Reset Value	POR
TWI0_IDE	R/W	TWI0 Interrupt Enable And DMA Control Register	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
7	TXDMAEN	DMA Transmission Channel Enable Bit 0: Disable DMA transmission function 1: Enable DMA transmission function When this bit is enabled, setting TXnE can trigger DMA channel transmit requests.
6	RXDMAEN	DMA Receive Channel Enable Bit 0: Disable DMA receive function 1: Enable DMA receive function When this bit is enabled, setting RXnE can trigger DMA channel transmit requests.
0	INTEN	Interrupt Request CPU Enable Control Bit 0: Disable interrupt request 1: Enable interrupt request
31~8 5~1	-	Reserved

19.7.2 TWI0 Register Mapping

Register	Offset Address	R/W	Description	Reset Value	POR
TWI0 Base Address:0x4002_0060					
TWI0_CON	0x00	R/W	TWI0 Control Register	0x0000_0000	0x0000_0000
TWI0_STS	0x04	R/W	TWI0 Status Flag Register	0x0000_0000	0x0000_0000
TWI0_ADD	0x08	R/W	TWI0 Address Register	0x0000_0000	0x0000_0000
TWI0_DATA	0x0C	R/W	TWI0 Data Register	0x0000_0000	0x0000_0000

Register	Offset Address	R/W	Description	Reset Value	POR
TWI0_IDE	0x10	R/W	TWI0 Interrupt Enable and DMA Control Register	0x0000_0000	0x0000_0000

19.8 TWI1 Register

19.8.1 TWI1 Related Register

19.8.1.1 SPI1_TWI1 Dual-Function Control Register (SPI1_TWI1_CON)

Register	R/W	Description	Reset Value	POR
SPI1_TWI1_CON	R/W	SPI1_TWI1 Dual-Function Control Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
MODE[1:0]		-	CPOL	CPHA	DORD	SPMD	SMSTR
15	14	13	12	11	10	9	8
-	SPOS[1:0]		-	QTWCK[3:0]			
7	6	5	4	3	2	1	0
QTWEN	-	START	STOP	-	-	AA	STRETCH

<table>
<tr><th>Bit number</th><th>Bit Mnemonic</th><th>Description</th></tr>
<tr><td>23~22</td><td>MODE[1:0]</td><td>SPI1_TWI1 Mode Selection Bit
00: TWI Mode
01: SPI Mode
10: Reserved
11: Reserved</td></tr>
<tr><td>14~13</td><td>SPOS[1:0]</td><td>● TWI1 Port Mapping Control Bit@SPI1_TWI1_CON
<table>
<tr><th>Port / SPOS Value</th><th>SCL1</th><th>SDA1</th></tr>
<tr><td>SPOS[1:0]=00</td><td>PB14</td><td>PB15</td></tr>
<tr><td>SPOS[1:0]=01</td><td>PA6</td><td>PA7</td></tr>
<tr><td>SPOS[1:0]=10</td><td>PA11</td><td>PA12</td></tr>
<tr><td>SPOS[1:0]=11</td><td>PB2</td><td>PB3</td></tr>
</table></td></tr>
<tr><td>11~8</td><td>QTWCK[3:0]</td><td>TWI Master Mode Clock Presclar Control Bit:
0000：f_{PCLK1}
0001：f_{PCLK1} /2
0010：f_{PCLK1} /4
0011：f_{PCLK1} /8
0100：f_{PCLK1} /16</td></tr>
</table>

Bit number	Bit Mnemonic	Description
		0101：f_{PCLK1} /32 0110：f_{PCLK1} /64 0111：f_{PCLK1} /128 1000：f_{PCLK1} /256 1001：f_{PCLK1} /512 1010：f_{PCLK1} /1024 1011：f_{PCLK1} /2048 1100：f_{PCLK1} /4096 Others：f_{PCLK1} /4096 Note: When TWI1 operates as a master, the maximum frequency is 4 MHz; when it operates as a slave, the maximum frequency is also 2 MHz.
7	QTWEN	SPI1_TWI1 Module Enable Control Bit 0: Disable module 1: Enable module
5	START	TWI Initial Position Trigger Switch Start condition will be generated when this bit is set to 1, and the TWI will switch to master mode. Software can set or clear this bit, or it can be cleared by hardware after the start condition is issued.
4	STOP	TWI Stop Bit Trigger Switch In master mode, writing 1 to this bit will generate a stop condition after the current byte transmission or the start condition is issued. Software can set or clear this bit, or it can be cleared by hardware when a stop condition is detected.
1	AA	TWI Acknowledge Enable Bit 0: No acknowledgment, returns UACK (acknowledge bit is high level). 1: Returns an acknowledgment (ACK) after receiving a matching address or data
0	STRETCH	TWI Clock Stretching Enable Bit This bit is only valid in slave mode. 0: Disable clock stretching. 1: Enable clock stretching, and the master needs to support clock stretching. Description: Clock stretching will occur after data transmission is complete and ACK is 0.
31~24 21 15 12 6 3~2	-	Reserved

19.8.1.2 SPI1_TWI1 Dual-Function Status Flag Register (SPI1_TWI1_STS)

Register	R/W	Description	Reset Value	POR
SPI1_TWI1_STS	R/W	SPI1_TWI1 Dual-Function Status Flag Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
NBYTES[7:0]							
15	14	13	12	11	10	9	8
-	-	-	-	-	STATE[2:0]		
7	6	5	4	3	2	1	0
WCOL	TXEIF	-	-	TMSTR	GCA	TXnE/RXnE	QTWIF

Bit number	Bit Mnemonic	Description
23~16	NBYTES[7:0]	Transmission/Reception Buffer Number Setting Bit Used to set the number of bytes to be transmitted/received. For each successful transmission/reception, NBYTES will automatically decrease by 1. When NBYTES reaches 0, the TC flag will be set. **Note: Modification is not allowed when STA is set to 1.**
10~8	STATE[2:0]	TWI Status Bits Used to indicate the TWI status, with different meanings in master/slave mode ● Slave mode: 000: Slave is in idle state, waiting for TWEN to be set, detecting the TWI start signal. The slave will transit to this state after receiving a stop condition 001: Slave is receiving the first frame of address and read/write bit (the 8th bit is the read/write bit, 1 for read, 0 for write). The slave will transit to this state after receiving the start condition 010: Slave is in the data reception state 011: Slave is in the data transmission state ● Master mode: 001: Master is transmitting the start condition or the address of slave device 010: Master is transmitting data 011: Master is receiving data 111: The module is in an unexpected state. The conditions for entering the unexpected state are as follows: In Master Mode: ● Address mismatch;

Bit number	Bit Mnemonic	Description
		● Receiving a NACK while in transmit mode; ● Sending a NACK while in receive mode; In Slave Mode: ● Address mismatch or writing AA; ● Sending a NACK while in receive mode; ● Writing AA or receiving a NACK while in transmit mode.
3	TMSTR	TWI Master/Slave Mode Flag Bit 0: Slave mode 1: Master mode Description: ● When the TWI interface transmit a start condition to the bus, it automatically switches to master mode, and the hardware will set this bit. ● When a stop condition is detected on the bus, the hardware will clear this bit.
2	GCA	TWI General Call Address Response Flag Bit 0: Non-response to general call address 1: When GC is set to 1 and there is a match with the general call address, this bit will set to 1 by the hardware and then automatically cleared
1	TXnE/RXnE	TWI Transmission Complete Flag Bit TXnE/RXnE will be set to 1 by the hardware in the following cases ● Master mode: ■ Master transmits an address frame (write), and receives ACK from the slave ■ Master finishes sending data and receives ACK from the slave ■ Master receives data and responds with ACK to the slave ● Slave mode: ■ Slave receives an address frame (read), and the received address matches the slave address (TWA) ■ Slave receives data and responds with ACK to the master ■ Slave finishes sending data and receives ACK from the master (AA=1) After a read/write operation on TWIDAT, this bit will be cleared by the hardware
0	QTWIF	TWI Interrupt Flag Bit This bit is set to 1 by the hardware and can be cleared by writing 1 through software ● Master mode: ■ Transmit start signal ■ Finish transmitting the address frame ■ Receive or finish transmitting a data frame

Bit number	Bit Mnemonic	Description
		● Slave mode: ■ Successful match of the first address frame ■ Successfully receive or transmit 8 bits of data ■ Receive a repeated start condition ■ Slave receives a stop signal
31~24 15~11 5~4	-	Reserved

19.8.1.3 TWI1 Address Register (TWI1_ADD)

Register	R/W	Description	Reset Value	POR
TWI1_ADD	R/W	TWI1 Address Register	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
7~1	QTWADD[6:0]	TWI Address Register TWA[6:0] cannot be written as all 0; 00H is reserved for general call address. This bit is not valid in master mode
0	GC	TWI General Call Address Response Enable Bit 0: Disable response to general call address 00H 1: Enable response to general call address 00H
31~8	-	Reserved

19.8.1.4 SPI1_TWI1 Dual-Function Data Register (SPI1_TWI1_DATA)

Register	R/W	Description	Reset Value	POR
SPI1_TWI1_DATA	R/W	SPI1_TWI1 Dual-Function Data Register	0x0000_0000	0x0000_0000

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

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

Bit number	Bit Mnemonic	Description
7~0	QTWIDAT[7:0]	TWI Data Buffer Read operation: Read the received data from the TWI reception buffer. Write operation: Write the data to be transmitted into the TWI transmission buffer.
31~16	-	Reserved

19.8.1.5 SPI1_TWI1 Dual-Function Interrupt Enable And DMA Control Register (SPI1_TWI1_IDE)

Register	R/W	Description	Reset Value	POR
SPI1_TWI1_IDE	R/W	SPI1_TWI1 Dual-Function Interrupt Enable And DMA Control Register	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
0	INTEN	Interrupt Request CPU Enable Control Bit 0: Disable interrupt request 1: Enable interrupt request
31~8 5~1	-	Reserved

19.8.2 TWI1 Register Mapping

Register	Offset Address	R/W	Description	Reset Value	POR
TWI1 Base Address:0x4002_1040					
SPI1_TWI1_CON	0x00	R/W	SPI1_TWI1 Dual-Function Control Register	0x0000_0000	0x0000_0000

Register	Offset Address	R/W	Description	Reset Value	POR
SPI1_TWI1_STS	0x04	R/W	SPI1_TWI1 Dual-Function Status Flag Register	0x0000_0000	0x0000_0000
TWI1_ADD	0x08	R/W	TWI1 Address Register	0x0000_0000	0x0000_0000
SPI1_TWI1_DATA	0x0C	R/W	SPI1_TWI1 Dual-Function Data Register	0x0000_0000	0x0000_0000
SPI1_TWI1_IDE	0x10	R/W	SPI1_TWI1 Dual-Function Interrupt Enable and DMA Control Register	0x0000_0000	0x0000_0000

20 Controller Area Network(CAN)

20.1 Overview

The Controller Area Network (CAN) in the SC32F15G series supports communication using both the CAN 2.0B protocol and the CAN FD protocol. Compared to the CAN 2.0B protocol, CAN FD offers greater flexibility, with a bit rate that can be adjusted (unlike the fixed 1 Mbit/s in CAN 2.0B) and a data field length of up to 64 bytes. The CAN module supports four different operating modes, including a low-power standby mode and wake-up functionality from standby.

The transmit buffer supports two types of transmission buffers: the PTB (Primary Transmission Buffer) and the STB (Secondary Transmission Buffer). The transmission order can be determined using either FIFO mode or priority mode. The receive buffer can store up to 8 frames simultaneously, and each received frame has an individual timestamp. Additionally, there are 8 configurable receive filters, each of which can be independently enabled and configured with specific filtering conditions.

20.2 Clock Source

The SC32F15G series CAN has only one clock source, which is derived from HCLK

20.3 Feature

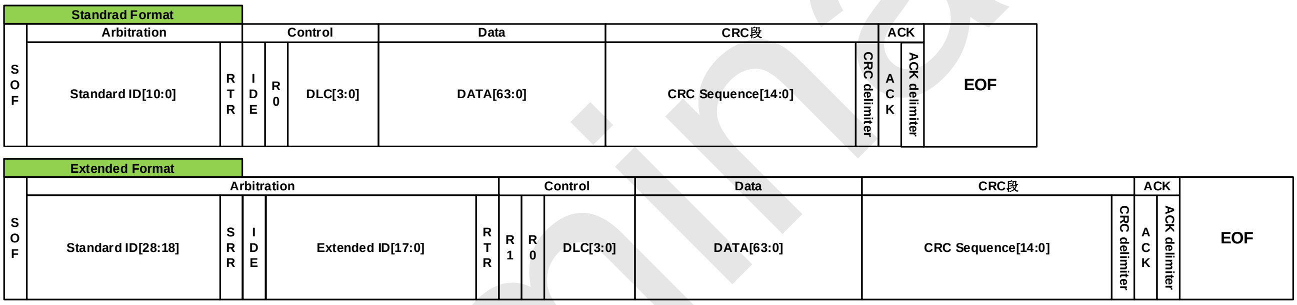
- Protocol Support:
  - CAN 2.0B
    - Support standard format and extend format, maximum load 8 bytes data
    - Bit rate: 1Mbit/s
  - CAN FD
    - Support standard format and extend format, maximum load 64 bytes data
    - Variable bit rate
- Supports low-power standby mode to reduce power consumption when the CAN interface is idle
- Time-Stamping:
  - CiA 603 Compliance, provides a 64-bit time-stamp for precise timing, each transmitted frame has one time-stamp stored in a register, and all received frames have individual time-stamps
- Transmit and Receive Buffers:
  - 8 Receive Buffers (RB)
  - 9 Transmit Buffers (TB)
    - 1 Primary Transmit Buffer(PTB)
    - 8 Secondary Transmit Buffer(STB), support FIFO mode or priority mode
  - 8 Receive Filters: Support 29-bit identifiers for filtering incoming messages

20.4 CAN Protocol

20.4.1 CAN2.0 Protocol

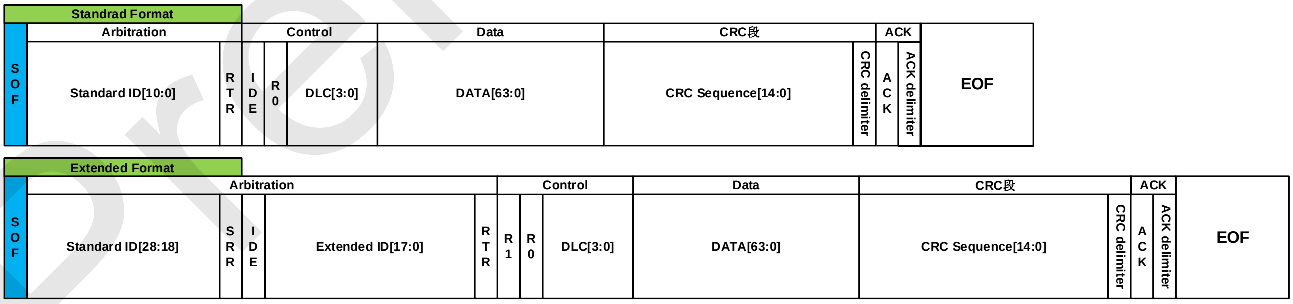
The traditional CAN2.0 protocol has the following features:

- It includes a Start of frame(SOF), arbitration field, control field, data field, CRC field, ACK field, and End of frame(EOF).
- The ID identifier can be selected as either a standard frame or an extended frame.
- It is possible to transmit either a data frame or a remote frame.
- The maximum data length is 8 bytes.
- The bit rate is fixed and configured by the low-speed bit rate register.
- The CRC check field is 15 bits.



20.4.1.1 Start of Frame

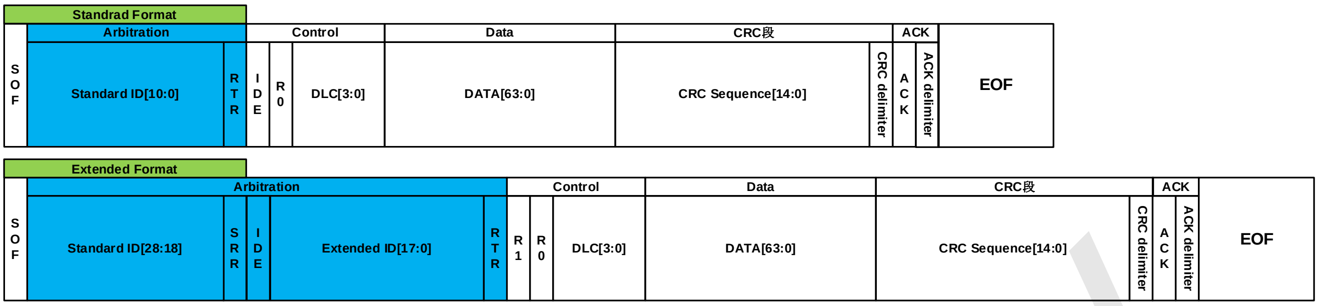
In the CAN 2.0 protocol, SOF is the single dominant start of frame (SOF) bit marks the start of a message, and is used to synchronize the nodes on a bus after being idle.



20.4.1.2 Arbitration Field

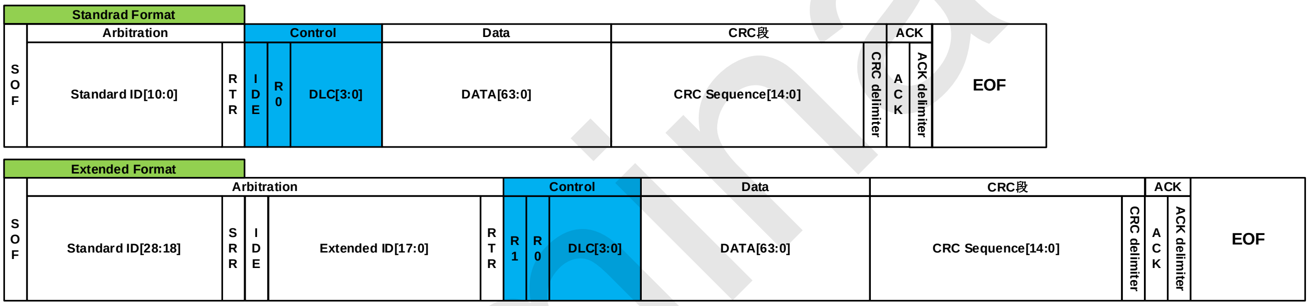
In the CAN 2.0 protocol, the arbitration field includes the ID identifier segment, the RTR bit, and the IDE bit (for extended frames). Users can enable or disable the extended identifier by setting the IDE bit for a particular frame.

The ID identifier in the arbitration field serves as a basis for priority determination. When multiple nodes need to access the bus at the same time, the principle of dominant bits having higher priority than recessive bits applies. The smaller the ID identifier, the higher the priority. If the IDs are the same, a data frame with a dominant RTR bit has higher priority than a remote frame with a recessive RTR bit. In addition to priority determination, the ID identifier can also be used as a basis for filtering criteria when the filtering function is enabled.



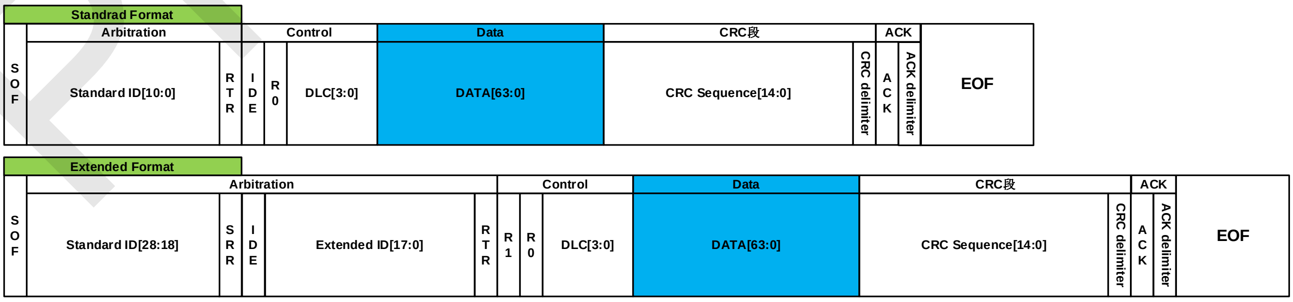
20.4.1.3 Control Field

In the CAN 2.0 protocol, the control field includes the IDE bit (for standard frames), the reserved bit, and the DLC bit. In a CAN 2.0 frame, the maximum data length can be set to 8 bytes.



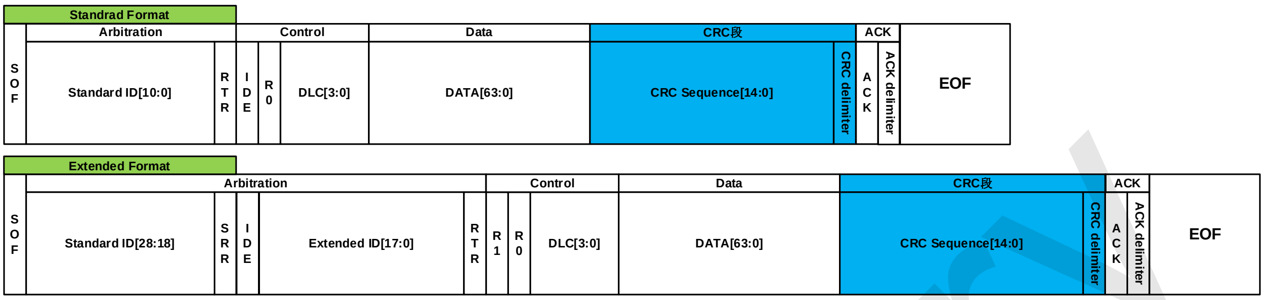
20.4.1.4 Date Field and Remote Field

In the CAN 2.0 protocol, there is an RTR bit that can control whether the transmitted message is a data frame or a remote frame. When selecting a data frame, the user writes the data to be sent into the data field and sends it through the normal transmission process. When selecting a remote frame, a node that needs data can send a remote frame to request another node to transmit a data frame with the same ID. Compared to a data frame, the RTR bit of a remote frame is recessive, while the RTR bit of a data frame is dominant. Moreover, a remote frame does not have a data field, and its DLC indicates the data length of the requested data frame.



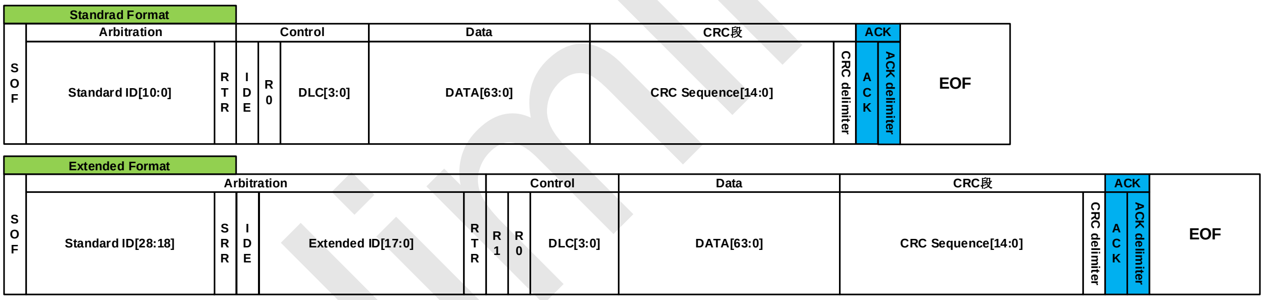
20.4.1.5 CRC Field

The CRC field is a part of the CAN 2.0 protocol used for error detection. It consists of the CRC sequence and the CRC delimiter. The CRC sequence is a value calculated based on a specific polynomial, and its computation covers the frame start field, arbitration field, control field, and data field. Both the transmitter and the receiver use the same algorithm to calculate the CRC sequence value. If the calculated values do not match, it indicates a communication error. The CRC delimiter is a single recessive bit following the CRC sequence, used to separate the CRC field from the ACK field.



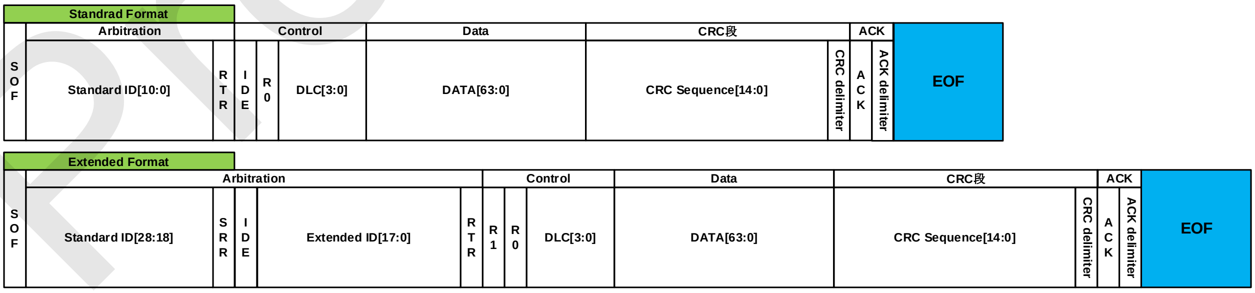
20.4.1.6 ACK Field

In the CAN 2.0 protocol, the ACK segment consists of an ACK bit and an ACK delimiter. The ACK bit is recessive in the transmitted frame. If the receiver confirms that the frame has been received without error, it will respond with a dominant ACK bit, indicating that the transmission has been successfully completed. The ACK delimiter is always recessive and serves as a separator.



20.4.1.7 End of Frame

In the CAN 2.0 protocol, EOF consists of 7 recessive bits, indicating the end of a message frame.

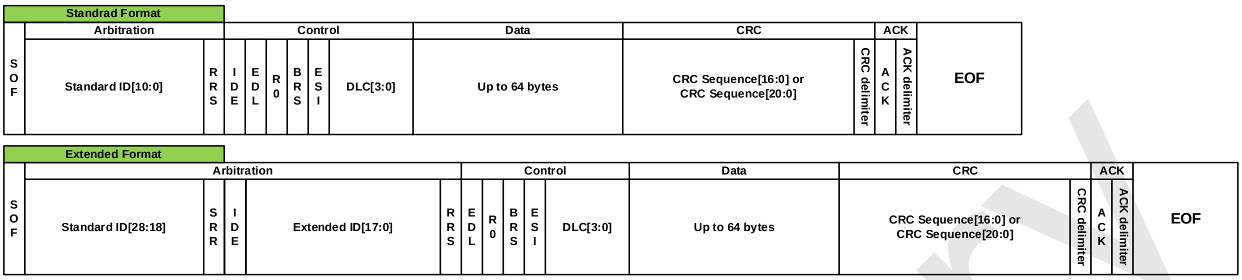


20.4.2 CAN_FD Protocol

The CAN_FD protocol, also known as the CAN flexible data-rate protocol, has the following features:

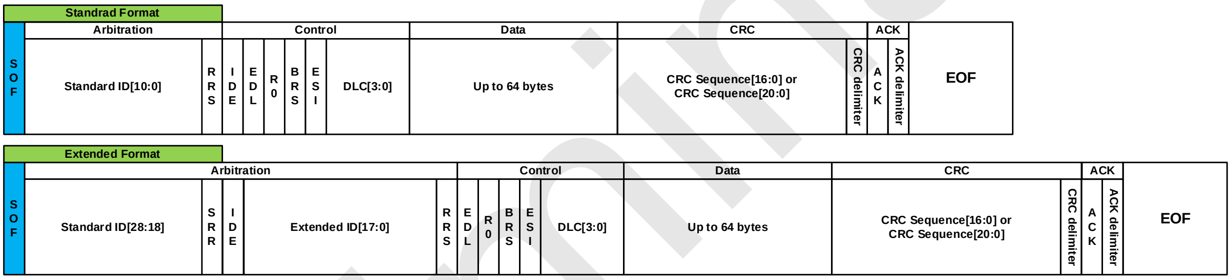
- It is compatible with the CAN 2.0 protocol.
- It includes a Start of frame(SOF), arbitration field, control field, data field, CRC field, ACK field, and End of frame(EOF).
- The ID identifier can be selected as either a standard frame or an extended frame.
- Only data frames can be transmitted.
- The maximum data length is 64 bytes.

- The data field can achieve a higher bit rate than the arbitration field by setting the BRS bit.
- The control field includes an ESI bit to indicate the error status.
- CRC includes bit stuffing in the calculation, with up to 21 CRC check bits.



20.4.2.1 Start of Frame

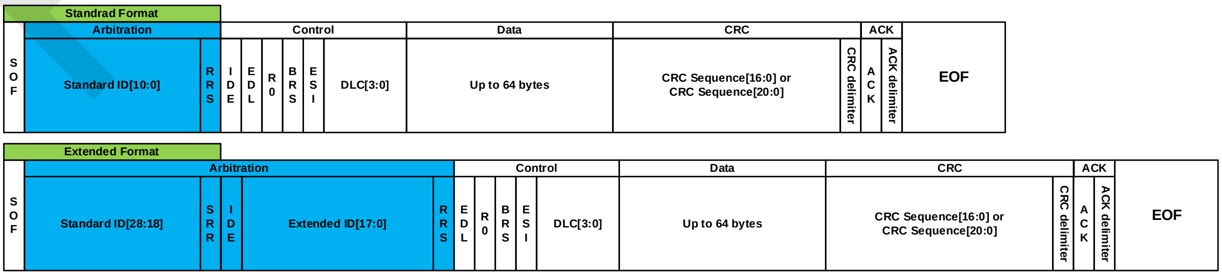
In both the CAN_FD protocol and the CAN 2.0 protocol, SOF is the single dominant start of frame (SOF) bit marks the start of a message, and is used to synchronize the nodes on a bus.



20.4.2.2 Arbitration Field

In CAN_FD, the arbitration field includes the ID identifier segment and the IDE (Identifier Extension) bit (for extended frames). Compared to CAN 2.0, the RTR bit is replaced by the RRS bit (referred to as r1 in the figure below), which is always dominant. Users can control whether to enable the extended identifier for a frame by setting the IDE bit.

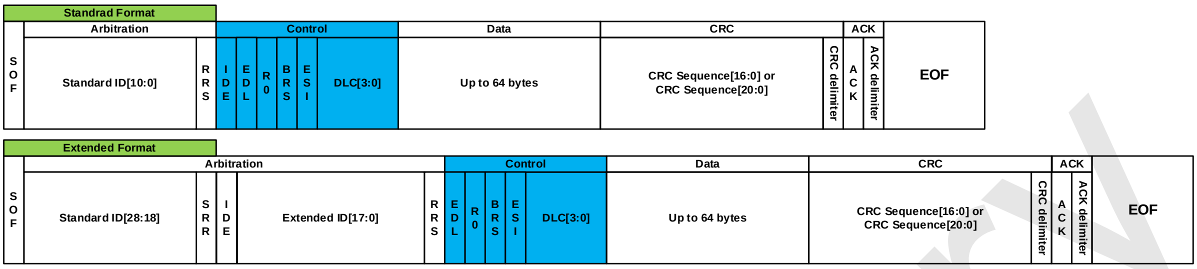
The ID identifier in the arbitration field serves as a basis for priority determination. When multiple nodes need to access the bus at the same time, the principle of dominant bits having higher priority than recessive bits applies. The smaller the ID identifier, the higher the priority. In addition to priority determination, the ID identifier can also be used as a basis for filtering criteria when the filtering function is enabled.



20.4.2.3 Control Field

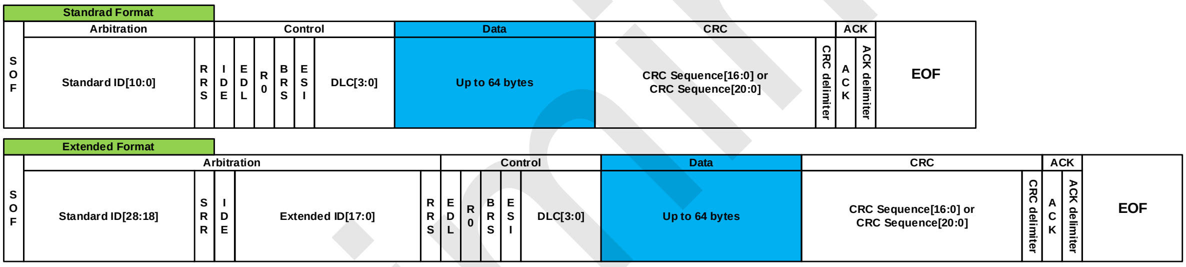
In CAN_FD, the control field includes the IDE bit (for standard frames), reserved bits, the EDL bit, the BRS bit, the ESI bit, and the DLC bit. The maximum data length in a CAN_FD frame is increased to 64 bytes.

In the CAN_FD protocol, the newly added EDL bit indicates that the frame is a CAN_FD frame when it is recessive. The BRS bit can control the use of different bit rates for the arbitration and data segments. The ESI bit can indicate whether a node is in the active error state or the passive error state.



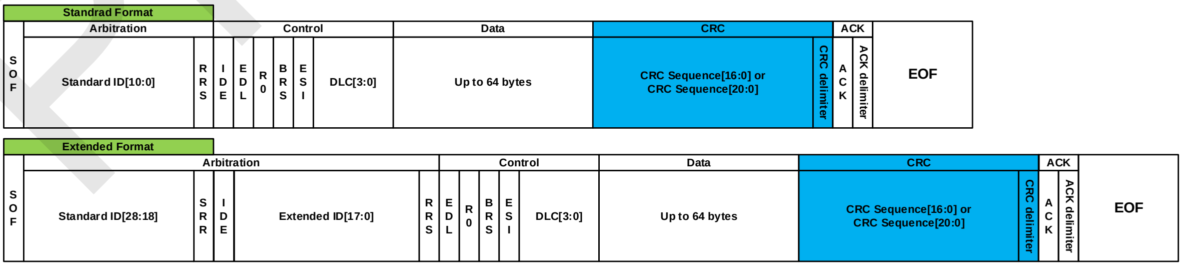
20.4.2.4 Date Field and Remote Field

In the CAN_FD protocol, only data frames can be selected; remote frames are not supported. When selecting a data frame, the user writes the data to be transmitted into the data field and sends it through the normal transmission process.



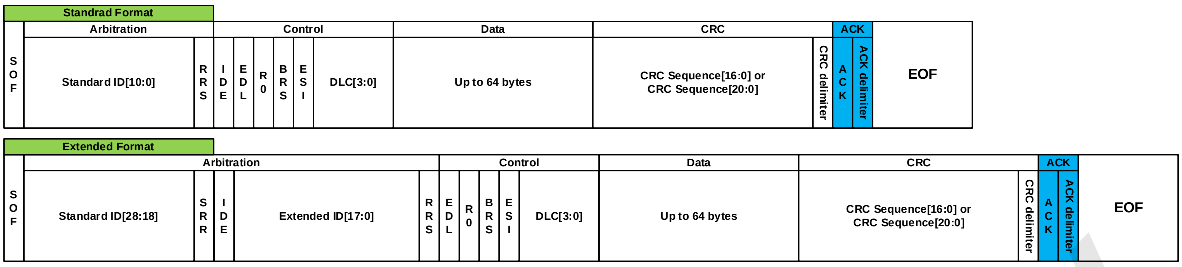
20.4.2.5 CRC Field

The composition of the CRC segment in the CAN_FD protocol is essentially the same as in CAN 2.0, but improvements have been made to the CRC algorithm based on CAN 2.0. Firstly, since CAN_FD supports a maximum data length of up to 64 bytes, the CRC sequence length has been increased to 17 bits, with a maximum of 21 bits, depending on the length of the transmitted data. Secondly, CAN_FD supports bit stuffing, which includes stuffing bits in the CRC sequence calculation, ensuring the robustness of CAN_FD.



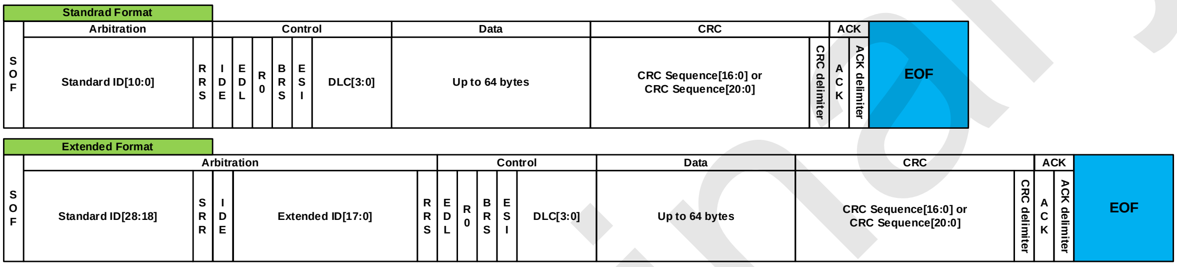
20.4.2.6 ACK Field

The ACK segment in both CAN_FD and CAN 2.0 protocols is constructed identically, consisting of an ACK bit and an ACK delimiter. The ACK bit is recessive in the transmitted frame. If the receiver confirms that the frame has been received without error, it will respond with a dominant ACK bit, indicating that the transmission has been successfully completed. The ACK delimiter is always recessive and serves as a separator.



20.4.2.7 End of Frame

In both CAN_FD and CAN 2.0, EOF consists of 7 recessive bits, indicating the end of a message frame.



20.5 Function Description

20.5.1 Baud Rate Configuration

The baud rate during CAN communication can be configured through the slow speed register CAN_S_SEG and the fast speed register CAN_F_SEG. In the CAN_FD mode, enabling the BRS bit supports the switching of different baud rates between the data field and the arbitration field, so as to increase the transmission rate of the data field. Among them, the slow speed register CAN_S_SEG configures the baud rate of the arbitration field the fast speed register CAN_F_SEG configures the baud rate of the data field. In the CAN2.0 mode, the baud rates of both the arbitration field and the data field can only be configured through the slow speed register CAN_S_SEG, and the fast speed register CAN_F_SEG is not used.

The baud rate calculation formula is as follows:

$$\mathbf{Baud} = \mathbf{clock}/\left(\left((\mathbf{SEG1}+\mathbf{2})+(\mathbf{SEG2}+\mathbf{1})\right)*(\mathbf{PRESC}+\mathbf{1})\right)$$

Where Baud is the baud rate, clock is the CAN clock, SEG1 is the phase segment 1 obtained by subtracting the synchronization segment from the time period before the sampling point, SEG2 is the phase segment 2 after the sampling point, and PRESC is the prescaler value. In addition, SJW also needs to be configured in the register. SJW is the maximum value of the resynchronization jump width, which does not directly affect the baud rate. Its function is to adjust the widths of SEG1 and SEG2 according to the phase difference when a phase difference occurs in each bit of the message due to uncontrollable factors. Therefore, it is required that the value of SJW is less than or equal to SEG1.

20.5.2 Fliter Function

When users want to receive messages with specific IDs, they can enable the filter function for screening. The filter mainly achieves the filtering function through the cooperation of ACODE and AMASK, and both ACODE and AMASK are configured in the CAN_ACF register. The ID will be compared with each bit of the value written in ACODE. If all bits are the same, the message passes through the filter. The bits with a value of 1 in the value written in AMASK will mask the comparison of the corresponding bits in ACODE, so that when the ID is compared with ACODE, regardless of whether the corresponding bits where AMASK is set to 1 are the same or not, the comparison result is considered the same. If the ID is an extended frame, then bits ID28~0 will be compared; if the ID is a standard frame, only bits ID10~0 will be compared, and bits ID28~11 will not affect the result. For example, in the case of a standard frame format, if ACODE is set to 0x212 and AMASK is set to 0x000, then only the message with the ID number of 0x212 can pass through the filter. If AMASK is set to 0x7F0, then the filter will allow all messages with the last digit of 2 (i.e., the ID is 0xXX2) to pass through.

There are a total of 8 filters for users to choose from. The ID will be compared successively from the enabled filter with the smallest number to the enabled filter with the largest number. If it passes the comparison, the message will be received; if it fails, it will continue to be compared with the next filter. If it fails all comparisons, the message will not be received. Enabling a hardware reset, that is, setting the RESET bit in the CAN_CFG_STAT register to 1, by default only filter 0 is enabled, and all messages with any ID are received by default, while other filters are disabled. Only filter 0 is affected by the power-on reset, and all other filters remain in an uninitialized state. The selection and enabling of the filters are configured in the CAN_ACFCTRL register, and the pointer directions of AMASK and ACODE are also configured in the CAN_ACFCTRL register. When the pointer points to AMASK, the CAN_ACF register can also achieve the function of receiving only standard frames or only extended frames by configuring the AIDE and AIDEE bits. When the pointer points to ACODE, configuring the AIDE and AIDEE bits in the CAN_ACF register will have no effect.

20.5.3 Transmit Buffer

The CAN module provides two types of transmit buffers for users to choose from: the PTB (Primary Transmit Buffer) and the STB (Secondary Transmit Buffer). The PTB has only one slot, meaning it can store only one message frame, while the STB has eight slots, allowing it to store up to eight message frames.

When a message frame is written into the PTB and fills it up, the TPE bit must be set to initiate message transmission.

When a slot in the STB is filled with a message frame, the TSNEXT bit must be set to move the STB pointer to the next slot to store the next message frame. After each move to a new slot, the random number in the related transmit register must be cleared to avoid any impact on message writing. The number of stored messages in the STB can be checked using the TSSTAT bit. Regardless of whether the STB is full or not, message transmission can be initiated by setting either the TSONE or TSALL bit. The difference between TSONE and TSALL is that TSONE transmits one message frame from one STB slot at a time, while TSALL transmits all messages together. TSONE can generate an interrupt after each message frame is transmitted, whereas TSALL generates an interrupt only after all filled slots have been transmitted. During STB message transmission, the order of transmission can be determined by the TSMODE bit. The transmission mode can be set to FIFO mode, where messages are transmitted in the order they were written into the STB, or based on ID priority, where messages with smaller ID identifiers have higher transmission priority.

Since the PTB has higher transmission priority than the STB, if TPE is enabled to start PTB transmission during STB transmission, the message transmission in the PTB will begin immediately after the current frame in the STB is transmitted. The STB message transmission will be delayed until the PTB transmission is completed or TPA (Transmit Primary Abort) terminates the PTB message transmission.

20.5.4 Auto Retransmission

The CAN module includes automatic retransmission bits for two types of buffers, PTB (Primary Transmit Buffer) and STB (Secondary Transmit Buffer), which are the TPSS and TSSS bits in the CAN_CFG_STAT register. These two masking bits control the automatic retransmission function of their respective transmit buffers. In the default mode, and the CAN module will automatically retransmit messages that encounter errors or lose arbitration. After enabling the single shot mode, the automatic retransmission function of the corresponding buffer will be disabled, and messages with errors or arbitration loss will not be automatically resent.

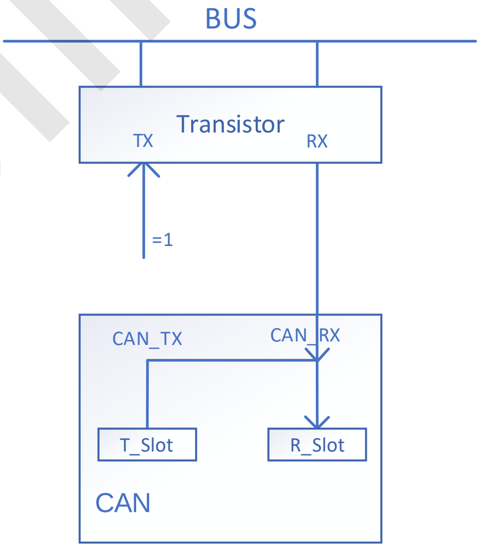
20.6 Operating Mode

The CAN module can operate in four different special modes: Silent Mode, External Loopback Mode, Internal Loopback Mode, and Standby Mode. The specific characteristics of each mode are as follows:

- Silent Mode

When CAN operates in Silent Mode, it cannot transmit messages onto the CAN bus but can receive messages from the bus into the receive buffer. It cannot respond with an ACK nor can it set the receive interrupt.

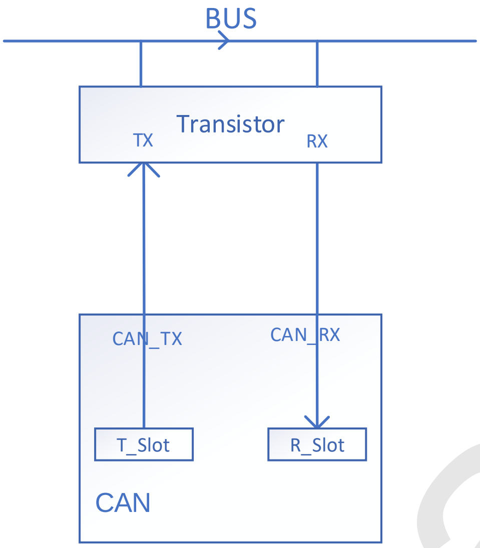
If CAN operates in Silent Mode and External Loopback Mode is enabled simultaneously, the MCU can transmit messages onto the bus. However, it can only respond with an ACK to the messages it has transmitted and set the interrupt for those messages. Other messages on the bus can be stored in the buffer but will not be acknowledged with an ACK.



Silent mode diagram

- External Loopback Mode

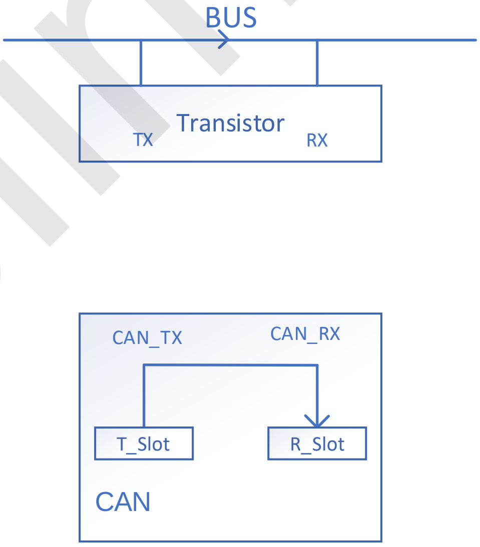
When CAN operates in External Loopback Mode, if the MCU device is connected to the bus, the MCU will receive both the messages it transmits and those from external sources, with interrupts being normally set. If the MCU is not connected to the bus, by connecting CAN_TX to CAN_RX and setting the SACK bit to 1, the MCU will also receive the messages it transmits, with interrupts being normally set.



External Loopback Mode

- Internal Loopback Mode

When CAN operates in Internal Loopback Mode, the MCU directly sends messages internally to the receiver without passing through the transceiver. The receiver will only receive the messages transmitted by itself and will not receive any messages from the bus.



Internal Loopback Mode

- Standby Mode

When CAN operates in Standby Mode, the CAN module will not enable the message transmission function after the MCU is powered on. When the MCU receives a message from the bus, that is, upon detecting a dominant level on the bus, the CAN module will exit Standby Mode regardless of whether the message passes through the filter. After exiting Sleep Mode, it can then transmit messages normally.

20.7 CAN Transmission Configuration Step

20.7.1 CAN Module Initialization

Before enabling the CAN module-related functions of the MCU, the CAN module must be initialized. The initialization steps include:

- Enabling the CAN clock on the AHB clock bus and performing clock initialization.
- Initializing the IO according to the actual functional requirements.
- Initializing the CAN structure.

Since both the transmit buffer (TBUF) and receive buffer (RBUF) are composed of RAM, they contain random values after each power-on or reset. Additionally, the receive-related registers are read-only and cannot be cleared. Therefore, users should avoid reading the receive buffer before the CAN module has received any data, as this will result in reading meaningless random numbers.

20.7.2 Transmission Message Configuration

When configuring a message for transmission, the user needs to clear and configure the following registers:

- Baud Rate Configuration: Set the baud rate by configuring the slow speed baud rate register (CAN_S_SEG) and the fast speed baud rate register (CAN_F_SEG).
- Transmit Buffer Selection: Choose between the PTB (Primary Transmit Buffer) and STB (Secondary Transmit Buffer) by configuring the transmit buffer selection bit (TBSEL).
- CAN Mode Selection: Select between CAN 2.0 and CAN_FD modes by configuring the CAN_FD mode bit (FDF).
- Identifier Type Configuration: Choose between standard and extended frame IDs by configuring the identifier extension bit (IDE).
- Remote Transmission Request: Select between remote and data frames by configuring the remote transmission request bit (RTR). This bit can be ignored if CAN_FD mode is selected.
- Data Length Configuration: Set the length of the data to be transmitted by configuring the data length code bit (DLC).
- Transmit Buffer Configuration: Configure the TBUF (Transmit Buffer) to set the data to be transmitted.

20.7.3 Transmission Step

- After enabling the transmit function, the user writes the configured message into the buffer. If the STB (Secondary Transmit Buffer) is selected, the user must enable the TSNEXT bit after writing each message frame to indicate that the current frame is fully configured and to move the TBUF pointer to the next slot for the next message frame.
- Once all the messages to be transmitted are written, if the PTB (Primary Transmit Buffer) is selected, the user enables the PTE bit to start transmission. If the STB is selected, the user enables the TSONE bit to transmit one frame at a time or the TSALL bit to transmit all frames together.
- If the corresponding interrupt is enabled, transmitting the message will set the corresponding interrupt flag and enter the interrupt service routine. If the interrupt is not enabled, the interrupt flag will not be set.

20.7.4 Reception Step

- Enable the TXEN (transmit enable) and RXEN (receive enable) bits in the CAN_IDE register. Enabling TXEN ensures that the ACK bit is correctly replied after a message is received.
- Configure the slow speed baud rate register (CAN_S_SEG) and the fast speed baud rate register (CAN_F_SEG).
- The RBUF has 8 slots and can store up to 8 message frames. Message reception can be filtered to select messages with specific identifiers. Messages that do not pass the filter will also enter the receive buffer but will be overwritten by the next frame until a filtered message arrives and moves the RBUF pointer to the next slot.
- After reading data from the receive buffer (RBUF), the RREL bit in the CAN_CFG_STAT register must be set to release the current slot and move the RBUF pointer to the next slot in order to successfully read the next message frame.
- If the corresponding interrupt is enabled, transmitting a message will set the corresponding interrupt flag and enter the interrupt service routine. If the interrupt is not enabled, the interrupt flag will not be set.

If the RBUF is full and new messages need to be stored in the receive buffer, the ROM bit can be set during initialization to choose whether to discard the newest or oldest messages.

20.7.5 Message Transmission Priority

The priority of message transmission and reception is related to the message identifier; the smaller the ID identifier, the higher the priority. When more than one message requests access to the bus at the same time, these messages will be processed in order based on their priority.

20.8 CAN Interrupt

When the CAN global interrupt enable bit is turned on and the corresponding sub-interrupt enable bit is set, an interrupt event will set the corresponding sub-interrupt flag and trigger the interrupt service routine. If the sub-interrupt enable bit is not set, the corresponding sub-interrupt flag will not be set.

Interrupt Event	Event Flag	Interrupt Enable Control Bit	Interrupt Enable Sub-Switch
Reception complete	RIF	CAN_IDE_INTEN	CAN_RTIE->RIE
RBUF full	ROIF		CAN_RTIE -> ROIE
RBUF overflow	RFIF		CAN_RTIE -> RFIE
RBUF slot Almost Full	RAFIF		CAN_RTIE -> RAFIE
PTB transmission complete	TPIF		CAN_RTIE -> TPIE
STB transmission complete	TSIF		CAN_RTIE -> TSIE
Error interrupt	EIF		CAN_RITE->EIE
Arbitration lost	ALIF		CAN_RTIE -> ALIE
Bus error	BEIF		CAN_RTIE -> BEIE
Error passive	EPIF		CAN_RTIE -> EPIE

20.9 CAN Register

20.9.1 CAN Related Register

20.9.1.1 Received Frame ID Register CAN_RX_ID—Standard Format

Register	R/W	Description	Reset Value	POR
CAN_RX_ID	R	Received Frame ID Register	0xnnnn_nnnn	0xnnnn_nnnn

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

Bit number	Bit Mnemonic	Description
31	ESI	Error State Indicator This is a read-only status bit for RBUF and is not available in TBUF. The protocol machine automatically embeds the correct value of ESI into transmitted frames. ESI is only included in CAN FD frames and does not exist in CAN 2.0 frames. 0: CAN node is error active 1: CAN node is error passive ESI in RBUF is always low for CAN 2.0 frames. The error state for transmission is shown with bit EPASS in register ERRINT.
10~0	ID[10:0]	The 11-bit ID of the received standard frame
30~11	-	Reserved

20.9.1.2 Received Frame ID Register CAN_RX_ID—Extend Format

Register	R/W	Description	Reset Value	POR
CAN_RX_ID	R	Received Frame ID Register	0xnnnn_nnnn	0xnnnn_nnnn

31	30	29	28	27	26	25	24
ESI	-	-	ID[28:24]-extend format				
23	22	21	20	19	18	17	16
ID[23:16]-extend format							
15	14	13	12	11	10	9	8

ID[15:8]-extend format							
7	6	5	4	3	2	1	0
ID[7:0]-extend format							

Bit number	Bit Mnemonic	Description
31	ESI	Error State Indicator This is a read-only status bit for RBUF and is not available in TBUF. The protocol machine automatically embeds the correct value of ESI into transmitted frames. ESI is only included in CAN FD frames and does not exist in CAN 2.0 frames. 0: CAN node is error active 1: CAN node is error passive ESI in RBUF is always low for CAN 2.0 frames. The error state for transmission is shown with bit EPASS in register ERRINT.
28~0	ID[28:0]-extend format	The 29-bit ID of the received extend frame
30~29	-	Reserved

20.9.1.3 Received Frame Control and Status Register CAN_RX_CTRL

Register	R/W	Description	Reset Value	POR
CAN_RX_CTRL	R	Received Frame Control and Status Register	0xnnnn_nnnn	0xnnnn_nnnn

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
KOER[2:0]			TX	-	-	-	-
7	6	5	4	3	2	1	0
IDE	RTR	FDF	BRS	DLC[3:0]			

Bit number	Bit Mnemonic	Description
15~13	KOER[2:0]	Kind Of Error (Error code) 000: No error 001: Bit error 010: Form error 011: Stuff error 100: Acknowledgement error 101: CRC error 110: Other error(dominant bits after own error flag, received active Error Flag too long, dominant bit during Passive-Error-Flag after ACK error) 111: Not used KOER is updated with each new error. Therefore it

<table>
<tr><th>Bit number</th><th>Bit Mnemonic</th><th>Description</th></tr>
<tr><td></td><td></td><td>stays untouched when frames are successfully transmitted or received.</td></tr>
<tr><td>12</td><td>TX</td><td>The receive status bit in loopback mode:
0: Not received.
1: In loopback mode, the frame sent by itself is received.</td></tr>
<tr><td>7</td><td>IDE</td><td>IDentifier Extension
0: Standard Format: ID(10:0)
1: Extended Format: ID(28:0)</td></tr>
<tr><td>6</td><td>RTR</td><td>Remote Transmission Request
0: data frame
1: remote frame
Only CAN 2.0 frames can be remote frames. There is no remote frame for CAN FD. Therefore RTR is forced to 0 if FDF =1 in TBUF and RBUF.
If a CAN FD frame is received with bit RRS=1, then this is ignored, a data payload is expected for reception instead and RTR in RBUF is overridden but the CRC of the frame is calculated with RRS=1.</td></tr>
<tr><td>5</td><td>FDF</td><td>CAN FD frame
0: CAN 2.0 frame (up to 8 bytes payload)
1: CAN FD frame (up to 64 bytes payload)</td></tr>
<tr><td>4</td><td>BRS</td><td>Bit Rate Switch
0: nominal / slow bit rate for the complete frame
1: switch to data / fast bit rate for the data payload and the CRC
Only CAN FD frames can switch the bit rate. Therefore BRS is forced to 0 if FDF =0.</td></tr>
<tr><td>3~0</td><td>DLC[3:0]</td><td>Received Frame Data Byte Length
For CAN 2.0, the maximum data byte length of a received frame is 8 bytes, and the minimum is 0 bytes.
For CAN FD, the maximum data byte length of a received frame is 64 bytes, and the minimum is 0 bytes.
The correspondence between the data byte lengths for CAN 2.0 and CAN FD is shown in the table below:
<table>
<tr><th>DLC</th><th>Frame Type</th><th>Payload in Bytes</th></tr>
<tr><td>0000 to 1000</td><td>CAN 2.0 and CAN FD</td><td>0 to 8</td></tr>
<tr><td>1001 to 1111</td><td>CAN 2.0</td><td>8</td></tr>
<tr><td>1001</td><td>CAN FD</td><td>12</td></tr>
<tr><td>1010</td><td>CAN FD</td><td>16</td></tr>
<tr><td>1011</td><td>CAN FD</td><td>20</td></tr>
<tr><td>1100</td><td>CAN FD</td><td>24</td></tr>
<tr><td>1101</td><td>CAN FD</td><td>32</td></tr>
<tr><td>1110</td><td>CAN FD</td><td>48</td></tr>
<tr><td>1111</td><td>CAN FD</td><td>64</td></tr>
</table></td></tr>
</table>

Bit number	Bit Mnemonic	Description
31~16 11~8	-	Reserved

20.9.1.4 Received Frame Data Register CAN_RBUF0~CAN_RBUF15

Register	R/W	Description	Reset Value	POR
CAN_RBUF0~CAN_RBUF15	R	Received Frame Data Register	0xnnnn_nnnn	0xnnnn_nnnn

31	30	29	28	27	26	25	24
Datax[7:0] (x=N*4+4 N=0~15)							
23	22	21	20	19	18	17	16
Datax[7:0] (x=N*4+3 N=0~15)							
15	14	13	12	11	10	9	8
Datax[7:0] (x=N*4+2 N=0~15)							
7	6	5	4	3	2	1	0
Datax[7:0] (x=N*4+1 N=0~15)							

Bit number	Bit Mnemonic	Description
31~24	Datax[7:0] (x=N*4+4 N=0~15)	The fourth byte of the N-th word in the received frame data
23~16	Datax[7:0] (x=N*4+3 N=0~15)	The third byte of the N-th word in the received frame data
15~8	Datax[7:0] (x=N*4+2 N=0~15)	The second byte of the N-th word in the received frame data
7~0	Datax[7:0] (x=N*4+1 N=0~15)	The first byte of the N-th word in the received frame data

20.9.1.5 Received Frame Timestamp Storage Lower 32 Bits Register CAN_RTSL

Register	R/W	Description	Reset Value	POR
CAN_RTSL	R	Received Frame Timestamp Storage Lower 32 Bits Register	0xnnnn_nnnn	0xnnnn_nnnn

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

RTS[7:0]

Bit number	Bit Mnemonic	Description
31~0	RTS[31:0]	The Reception Time Stamps (RTS) for CiA 603 time stamping are stored for each received message at the end of the RBUF address range. Depending on a generic parameter the time-stamp can be 32 or 64 wide. Unused bits are forced to 0.The position of the timestamp acquisition is controlled by the TIMEPOS Register in CAN_ACFCTRL. The CAN module provides 8 receive frame slots, each with an independent RTS (Receive Timestamp Storage) space. Therefore in contrast to TTS, RTS is related to the actual selected RBUF slot.

20.9.1.6 Received Frame Timestamp Storage Upper 32 Bits Register CAN_RTSH

Register	R/W	Description	Reset Value	POR
CAN_RTSH	R	Received Frame Timestamp Storage Upper 32 Bits Register	0xnnnn_nnnn	0xnnnn_nnnn

31	30	29	28	27	26	25	24
RTS[63:56]							
23	22	21	20	19	18	17	16
RTS[55:48]							
15	14	13	12	11	10	9	8
RTS[47:40]							
7	6	5	4	3	2	1	0
RTS[39:32]							

Bit number	Bit Mnemonic	Description
31~0	RTS[63:32]	The Reception Time Stamps (RTS) for CiA 603 time stamping are stored for each received message at the end of the RBUF address range. Depending on a generic parameter the time-stamp can be 32 or 64 wide. Unused bits are forced to 0.The position of the timestamp acquisition is controlled by the TIMEPOS Register in CAN_ACFCTRL. The CAN module provides 8 receive frame slots, each with an independent RTS (Receive Timestamp Storage) space. Therefore in contrast to TTS, RTS is related to the actual selected RBUF slot.

20.9.1.7 Transmit Frame ID Register CAN_TX_ID—Standard Format

Register	R/W	Description	Reset Value	POR
CAN_TX_ID	R/W	Transmit Frame ID Register	0xnnnn_nnnn	0xnnnn_nnnn

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

Bit number	Bit Mnemonic	Description
31	TTSEN	Transmit Time-Stamp Enable For CiA 603 time-stamping the acquisition of a transmit time stamp TTS can be selected in the TBUF: 0: No acquisition of a transmit time stamp for this frame 1: TTS update enabled
10~0	ID[10:0]	The 11-bit ID of the transmit standard frame
30~11	-	Reserved

20.9.1.8 Transmit Frame ID Register CAN_TX_ID—Extend Format

Register	R/W	Description	Reset Value	POR
CAN_TX_ID	R/W	Transmit Frame ID Register	0xnnnn_nnnn	0xnnnn_nnnn

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

Bit number	Bit Mnemonic	Description
31	TTSEN	Transmit Time-Stamp Enable For CiA 603 time-stamping the acquisition of a transmit time stamp TTS can be selected in the TBUF: 0: No acquisition of a transmit time stamp for this frame

Bit number	Bit Mnemonic	Description
		1: TTS update enabled
28~0	ID[28:0]	The 29-bit ID of the transmit extend frame
30~29	-	Reserved

20.9.1.9 Transmit Frame Control and Status Register CAN_TX_CTRL

Register	R/W	Description	Reset Value	POR
CAN_TX_CTRL	R/W	Transmit Frame Control and Status Register	0xnnnn_nnnn	0xnnnn_nnnn

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

Bit number	Bit Mnemonic	Description
7	IDE	IDentifier Extension 0: Standard Format: ID(10:0) 1: Extended Format: ID(28:0)
6	RTR	Remote Transmission Request 0: Data frame 1: Remote frame Only CAN 2.0 frames can be remote frames. There is no remote frame for CAN FD. Therefore RTR is forced to 0 if FDF =1 in TBUF and RBUF. If a CAN FD frame is received with bit RRS=1, then this is ignored, a data payload is expected for reception instead and RTR in RBUF is overridden but the CRC of the frame is calculated with RRS=1.
5	FDF	CAN FD frame 0: CAN 2.0 frame (up to 8 bytes payload) 1: CAN FD frame (up to 64 bytes payload)
4	BRS	Bit Rate Switch 0: Nominal / slow bit rate for the complete frame 1: Switch to data / fast bit rate for the data payload and the CRC Only CAN FD frames can switch the bit rate. Therefore BRS is forced to 0 if FDF =0.

<table>
<tr><th>Bit number</th><th>Bit Mnemonic</th><th colspan="3">Description</th></tr>
<tr><td rowspan="11">3~0</td><td rowspan="11">DLC[3:0]</td><td colspan="3">Received Frame Data Byte Length
For CAN 2.0, the maximum data byte length of a received frame is 8 bytes, and the minimum is 0 bytes.
For CAN FD, the maximum data byte length of a received frame is 64 bytes, and the minimum is 0 bytes.
The correspondence between the data byte lengths for CAN 2.0 and CAN FD is shown in the table below:</td></tr>
<tr><td>DLC</td><td>Frame Type</td><td>Payload in Bytes</td></tr>
<tr><td>0000 to 1000</td><td>CAN 2.0 and CAN FD</td><td>0 to 8</td></tr>
<tr><td>1001 to 1111</td><td>CAN 2.0</td><td>8</td></tr>
<tr><td>1001</td><td>CAN FD</td><td>12</td></tr>
<tr><td>1010</td><td>CAN FD</td><td>16</td></tr>
<tr><td>1011</td><td>CAN FD</td><td>20</td></tr>
<tr><td>1100</td><td>CAN FD</td><td>24</td></tr>
<tr><td>1101</td><td>CAN FD</td><td>32</td></tr>
<tr><td>1110</td><td>CAN FD</td><td>48</td></tr>
<tr><td>1111</td><td>CAN FD</td><td>64</td></tr>
<tr><td>31~8</td><td>-</td><td colspan="3">Reserved</td></tr>
</table>

20.9.1.10 Transmit Frame Data Register CAN_TBUF0~CAN_TBUF15

Register	R/W	Description	Reset Value	POR
CAN_TBUF0~CAN_TBUF15	R/W	Transmit Frame Data Register	0xnnnn_nnnn	0xnnnn_nnnn

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">Datax[7:0] (x=N*4+4 N=0~15)</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">Datax[7:0] (x=N*4+3 N=0~15)</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">Datax[7:0] (x=N*4+2 N=0~15)</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">Datax[7:0] (x=N*4+1 N=0~15)</td></tr>
</table>

Bit number	Bit Mnemonic	Description
31~24	Datax[7:0] (x=N*4+4 N=0~15)	The fourth byte of the N-th word in the transmit frame data
23~16	Datax[7:0] (x=N*4+3 N=0~15)	The third byte of the N-th word in the transmit frame data
15~8	Datax[7:0] (x=N*4+2 N=0~15)	The second byte of the N-th word in the transmit frame data
7~0	Datax[7:0] (x=N*4+1	The first byte of the N-th word in the transmit frame data

Bit number	Bit Mnemonic	Description
	N=0~15)	

20.9.1.11 Transmit Frame Timestamp Storage Lower 32 Bits Register CAN_TTSL

Register	R/W	Description	Reset Value	POR
CAN_TTSL	R/W	Transmit Frame Timestamp Storage Lower 32 Bits Register	0xnnnn_nnnn	0xnnnn_nnnn

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

Bit number	Bit Mnemonic	Description
31~0	TTS[31:0]	Transmission Time Stamp TTS holds the time stamp of the last transmitted frame for CiA 603 time stamping. Every new frame overwrites TTS if TTSEN=1. Depending on a generic parameter the time-stamp can be 32 or 64 wide. Unused bits are forced to 0. The TTS is intended to be used by the time master to acquire the time-stamp of the SYNC message.

20.9.1.12 Transmit Frame Timestamp Storage Upper 32 Bits Register CAN_TTSH

Register	R/W	Description	Reset Value	POR
CAN_TTSH	R/W	Transmit Frame Timestamp Storage Upper 32 Bits Register	0xnnnn_nnnn	0xnnnn_nnnn

31	30	29	28	27	26	25	24
TTS[63:56]							
23	22	21	20	19	18	17	16
TTS[55:48]							
15	14	13	12	11	10	9	8
TTS[47:40]							
7	6	5	4	3	2	1	0
TTS[39:32]							

Bit number	Bit Mnemonic	Description
31~0	TTS[63:32]	Transmission Time Stamp TTS holds the time stamp of the last transmitted frame for CiA 603 time stamping. Every new frame overwrites TTS if TTSEN=1. Depending on a generic parameter the time-stamp can be 32 or 64 wide. Unused bits are forced to 0. The TTS is intended to be used by the time master to acquire the time-stamp of the SYNC message.

20.9.1.13 CAN Configuration and State Register CAN_CFG_STAT

Register	R/W	Description	Reset Value	POR
CAN_CFG_STAT	R/W	CAN Configuration and State Register	0x0090_0080	0xnnnn_nnnn

31	30	29	28	27	26	25	24
SACK	ROM	ROV	RREL	RBALL	-	RSTAT[1:0]	
23	22	21	20	19	18	17	16
FD_ISO	TSNEXT	TSMODE	-	-	-	TSSTAT[1:0]	
15	14	13	12	11	10	9	8
TBSEL	LOM	STBY	TPE	TPA	TSONE	TSALL	TSA
7	6	5	4	3	2	1	0
RESET	LBME	LBMI	TPSS	TSSS	RACTIVE	TACTIVE	BUSOFF

Bit number	Bit Mnemonic	Description
31	SACK	Self-ACKnowledge 0: no self-ACK 1: self-ACK when LBME=1 This bit is only writeable if RESET=0
30	ROM	Receive Buffer Overflow Mode In case of a full RBUF when a new message is received, then ROM selects the following: 0: The oldest message will be overwritten. 1: The new message will not be stored.
29	ROV	Receive Buffer Overflow 0: No Overflow. 1: Overflow. At least one message is lost. ROV is cleared by setting RREL=1.
28	RREL	Receive Buffer Release The host controller has read the actual RB slot and releases it. Afterwards the CAN-CTRL core points to the next RB slot. RSTAT gets updated. 0: No release 1: Release: The host has read the RB.
27	RBALL	Receive Buffer stores ALL data frames

Bit number	Bit Mnemonic	Description
		0: normal operation 1: RB stores correct data frames as well as data frames with error
25~24	RSTAT[1:0]	Receive Buffer Status 00: R_slot is empty. 01: R_slot is non-empty, not full, and no overflow has occurred. The number of stored frames is less than the AFWL threshold. 10: R_slot is not full and no overflow has occurred, but the number of stored frames is greater than or equal to the AFWL threshold. 11: R_slot is full. If ROV = 1 at this time, it indicates that the RBUF has overflowed.
23	FD_ISO	CAN FD ISO mode 0: Bosch CAN FD (non-ISO) mode 1: ISO CAN FD mode (ISO 11898-1:2015) This bit is only writeable if RESET=1.
22	TSNEXT	Transmit Buffer Secondary NEXT 0 - no action 1 - STB slot filled, select next slot.
21	TSMODE	Transmit Buffer Secondary Operation Mode 0: FIFO mode 1: priority decision mode
17~16	TSSTAT[1:0]	Transmission Secondary Status bits 00: STB is empty 01: STB is less than or equal to half full 10: STB is more than half full 11: STB is full
15	TBSEL	Transmit Buffer Select Selects the transmit buffer to be loaded with a message. Use the TBUF registers for access. TBSEL needs to be stable all the time the TBUF registers are written and when TSNEXT is set. 0: PTB (high-priority buffer) 1: STB
14	LOM	Listen Only Mode 0: Disabled 1: Enabled This bit is only writeable if RESET=0
13	STBY	Transceiver Standby Mode 0: Disabled 1: Enabled This bit is only writeable if RESET=0
12	TPE	Transmit Primary Enable 0: No transmission for the PTB 1: Transmission enable for the message in the high-priority PTB
11	TPA	Transmit Primary Abort

Bit number	Bit Mnemonic	Description
		0: no abort 1: Aborts a transmission from PTB which has been requested by TPE=1 but not started yet. (The data bytes of the message remains in the PTB.)
10	TSONE	Transmit Secondary One frame 0: No transmission for the STB. 1: Transmission enable of one in the STB.
9	TSALL	Transmit Secondary ALL frames 0: No transmission for the STB. 1: Transmission enable of all messages in the STB.
8	TSA	Transmit Secondary Abort 0: No abort 1: Aborts a transmission from STB which has been requested but not started yet.
7	RESET	RESET request bit 1: The host controller performs a local reset of CAN-CTRL. 0: no local reset of CAN-CTRL The some register (e.g for node configuration) can only be modified if RESET=1 or RESET = 0.
6	LBME	Loop Back Mode 0: Disabled 1: Enabled LBME should not be enabled while a transmission is active. This bit is only writeable if RESET=0
5	LBMI	Loop Back Mode, Internal 0: Disabled 1: Enabled LBMI should not be enabled while a transmission is active. This bit is only writeable if RESET=0
4	TPSS	Transmission Primary Single Shot mode for PTB 0: Disabled 1: Enabled
3	TSSS	Transmission Secondary Single Shot mode for STB 0: Disabled 1: Enabled
2	RACTIVE	Reception Active (Receive Status bit) 0: No receive activity. 1: The controller is currently receiving a frame.
1	TACTIVE	Transmission ACTIVE (Transmit Status bit) 0: No transmit activity. 1: The controller is currently transmitting a frame.
0	BUSOFF	Bus Off 0: The controller status is “bus on”.

Bit number	Bit Mnemonic	Description
		1: The controller status is “bus off”. Writing a 1 to BUSOFF will reset TECNT and RECNT. This should be done only for debugging.
26 20~18	-	Reserved

20.9.1.14 CAN Interrupt Control and Flag Register CAN_RTIE

Register	R/W	Description	Reset Value	POR
CAN_RTIE	R/W	CAN Interrupt Control and Flag Register	0x1B00_00FE	0xnnnn_nnnn

31	30	29	28	27	26	25	24
AFWL[3:0]				EWL[3:0]			
23	22	21	20	19	18	17	16
EWARN	EPASS	EPIE	EPIF	ALIE	ALIF	BEIE	BEIF
15	14	13	12	11	10	9	8
RIF	ROIF	RFIF	RAFIF	TPIF	TSIF	EIF	AIF
7	6	5	4	3	2	1	0
RIE	ROIE	RFIE	RAFIE	TPIE	TSIE	EIE	TSFF

Bit number	Bit Mnemonic	Description
31~28	AFWL[3:0]	Receive Buffer Threshold When the number of filled RB slots equals the value set by AFWL(3:0), the RB slots threshold warning flag RAFIF will be set. An interrupt will be generated if RAFIE = 1 at this time.
27~24	EWL[3:0]	Programmable Error Warning Limit = (EWL+1)*8. Possible Limit values: 8, 16, … 128. RECNT and TECNT count errors during reception and transmission, respectively. If either of these error counters reaches the error count threshold, both EWARN and EIF will be set. An interrupt will be generated if EIE = 1 at this time.
23	EWARN	Error WARNing limit reached 0: Both the counters RECNT or TECNT are less than (EWL[3:0]+1)*8 1: One of the error counters RECNT or TECNT is equal or bigger than (EWL[3:0]+1)*8
22	EPASS	Error Passive mode active 0: not active (node is error active) 1: active (node is error passive)
21	EPIE	Error Passive Interrupt Enable 0: Disabled 1: Enabled
20	EPIF	Error Passive Interrupt Flag.

Bit number	Bit Mnemonic	Description
		EPIF will be activated if the error status changes from error active to error passive or vice versa and if this interrupt is enabled.
19	ALIE	Arbitration Lost Interrupt Enable 0: Disabled 1: Enabled
18	ALIF	Arbitration Lost Interrupt Flag 0: No arbitration lost happened 1: Arbitration lost has been happened
17	BEIE	Bus Error Interrupt Enable 0: Disabled 1: Enabled
16	BEIF	Bus Error Interrupt Flag 0: No bus Error happened 1: Bus Error has been happened
15	RIF	Receive Interrupt Flag 0: No frame has been received. 1: Data or a remote frame has been received and is available in the receive buffer.
14	ROIF	RB Overrun Interrupt Flag 0: No RB overwritten. 1: At least one received message has been overwritten in the RB. In case of an overrun both ROIF and RFIF will be set.
13	RFIF	RB Full Interrupt Flag 0: The RB FIFO is not full. 1: All RBs are full. If no RB will be released until the next valid message is received,the oldest message will be lost.
12	RAFIF	RB Almost Full Interrupt Flag 0: number of filled RB slots < AFWL 1: number of filled RB slots ≥ AFWL
11	TPIF	Transmission Primary Interrupt Flag 0: No transmission of the PTB has been completed. 1: The requested transmission of the PTB has been successfully completed.
10	TSIF	Transmission Secondary Interrupt Flag 0: No transmission of the STB has been completed successfully. 1: The requested transmission of the STB has been successfully completed.
9	EIF	Error Interrupt Flag 0: There has been no change. 1: The border of the error warning limit has been crossed in either direction, or the BUSOFF bit has been changed in either direction.
8	AIF	Abort Interrupt Flag

Bit number	Bit Mnemonic	Description
		0: No abort has been executed. 1: After setting TPA or TSA the appropriated message(s) have been aborted.
7	RIE	Receive Interrupt Enable 0: Disabled 1: Enabled
6	ROIE	RB Overrun Interrupt Enable 0: Disabled 1: Enabled
5	RFIE	RB Full Interrupt Enable 0: Disabled 1: Enabled
4	RAFIE	RB Almost Full Interrupt Enable 0: Disabled 1: Enabled
3	TPIE	Transmission Primary Interrupt Enable 0: Disabled 1: Enabled
2	TSIE	Transmission Secondary Interrupt Enable 0: Disabled 1: Enabled
1	EIE	Error Interrupt Enable 0: Disabled 1: Enabled
0	TSFF	Transmit Secondary buffer Full Flag 0: The STB is not filled with the maximal number of messages 1: The STB is filled with the maximal number of messages. If the STB is disabled using STB_DISABLE, then TSFF=0.

20.9.1.15 Slow Speed Clock Configuration Register CAN_S_SEG

Register	R/W	Description	Reset Value	POR
CAN_S_SEG	R/W	Slow Speed Clock Configuration Register	0x0102_0203	0xnnnn_nnnn

31	30	29	28	27	26	25	24
S_PRESC[7:0]							
23	22	21	20	19	18	17	16
-	S_SJW[6:0]						
15	14	13	12	11	10	9	8
-	S_Seg_2[6:0]						
7	6	5	4	3	2	1	0
S_Seg_1[7:0]							

Bit number	Bit Mnemonic	Description
31~24	S_PRESC[7:0]	Prescaler (slow speed) The prescaler divides the system clock to get the time quanta clock tq_clk. Valid range PRESC=[0x00, 0xff] results in divider values 1 to 256. divider values = S_PRESC+1
22~16	S_SJW[6:0]	Synchronization Jump Width (slow speed) The Synchronization Jump Width t_{SJW} =(SJW+1)*TQ is the maximum time for shortening or lengthening the Bit Time for resynchronization, where TQ is a time quanta.
14~8	S_Seg_2[6:0]	Bit Timing Segment 2 (slow speed) The sample point will be set to t_{Seg_2} =(Seg_2+1)*TQ after the sample point.
7~0	S_Seg_1[7:0]	Bit Timing Segment 1 (slow speed) The sample point will be set to t_{Seg_1} =(Seg_1+2)*TQ after start of bit time.
23 15	-	Reserved

20.9.1.16 Fast Speed Clock Configuration Register CAN_F_SEG

Register	R/W	Description	Reset Value	POR
CAN_F_SEG	R/W	Fast Speed Clock Configuration Register	0x0102_0203	0xnnnn_nnnn

31	30	29	28	27	26	25	24
F_PRESC[7:0]							
23	22	21	20	19	18	17	16
--	-	-	-	F_SJW[3:0]			
15	14	13	12	11	10	9	8
--	-	-	-	F_Seg_2[3:0]			
7	6	5	4	3	2	1	0
-	-	-	F_Seg_1[4:0]				

Bit number	Bit Mnemonic	Description
31~24	F_PRESC[7:0]	Prescaler (fast speed) The prescaler divides the system clock to get the time quanta clock tq_clk. Valid range PRESC=[0x00, 0xff] results in divider values 1 to 256.
19~16	F_SJW[3:0]	Synchronization Jump Width (fast speed) The Synchronization Jump Width t_{SJW} =(SJW+1)*TQ is the maximum time for

Bit number	Bit Mnemonic	Description
		shortening or lengthening the Bit Time for resynchronization, where TQ is a time quanta.
11~8	F_Seg_2[3:0]	Bit Timing Segment 2 (fast speed) The sample point will be set to t_{Seg_1} =(Seg_2+1)*TQ after the sample point
4~0	F_Seg_1[4:0]	Bit Timing Segment 1 (fast speed) The sample point will be set to t_{Seg_1} =(Seg_1+2)*TQ after start of bit time.
23~20 15~12 7~5	-	Reserved

20.9.1.17 Delay Compensation and Transmission Error Register CAN_EALCAP

Register	R/W	Description	Reset Value	POR
CAN_EALCAP	R/W	Delay Compensation and Transmission Error Register	0x0000_0000	0xnnnn_nnnn

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

Bit number	Bit Mnemonic	Description
31~24	TECNT[7:0]	Transmit Error Count (number of errors during transmission) TECNT is incremented and decremented as defined in the CAN specification.
23~16	RECNT[7:0]	Receive Error Count (number of errors during reception) RECNT is incremented and decremented as defined in the CAN specification.
15	TDCEN	Transmitter Delay Compensation Enable 0: Transceiver delay compensation disabled 1: Transceiver delay compensation enabled
14~8	SSPOFF	Secondary Sample Point Offset The transmitter delay plus SSPOFF defines the time of the secondary sample point for TDC. SSPOFF is given as a number of TQ.
7~5	KOER[2:0]	Kind Of Error (Error code)

Bit number	Bit Mnemonic	Description
		000: No error 001: Bit error 010: Form error 011: Stuff error 100: Acknowledgement error 101: CRC error 110: Other error(dominant bits after own error flag, received active Error Flag too long, dominant bit during Passive-Error-Flag after ACK error) 111: Not used KOER is updated with each new error. Therefore it stays untouched when frames are successfully transmitted or received.
4~0	ALC[4:0]	Arbitration Lost Capture (bit position in the frame where the arbitration has been lost) This event can be signaled by the ALIF interrupt if it is enabled. The value of ALC stays unchanged if the node is able to win the arbitration. Then ALC holds the old value of the last loss of arbitration. The value of ALC is defined as follows: A frame starts with the SOF bit and then the first bit of the ID is transmitted. This first ID bit has ALC value 0, the second ID bit ALC value 1 and so on.

20.9.1.18 Acceptance Fliter Control Register CAN_ACFCTRL

Register	R/W	Description	Reset Value	POR
CAN_ACFCTRL	R/W	Acceptance Fliter Control Register	0x00010200	0xnnnn_nnnn

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
AE_7	AE_6	AE_5	AE_4	AE_3	AE_2	AE_1	AE_0
15	14	13	12	11	10	9	8
	-	-	-	-	-	TIMEPOS	TIMEEN
7	6	5	4	3	2	1	0
-	-	SELMASK	-	ACFADR[3:0]			

Bit number	Bit Mnemonic	Description
23~16	AE_X X=(0~7)	Acceptance filter Enable 0: acceptance filter disable 1: acceptance filter enabled
9	TIMEPOS	TIME-stamping Position 0: SOF

Bit number	Bit Mnemonic	Description
		1: EOF TIMEPOS can only be changed if TIMEEN=0, but it is possible to modify TIMPOS with the same write access that sets TIMEEN=1.
8	TIMEEN	TIME-stamping Enable 0: Disabled 1: Enabled
5	SELMASK	Select Acceptance MASK 0: Registers ACF_x point to acceptance code 1: Registers ACF_x point to acceptance mask. ACFADR selects one specific acceptance filter.
3~0	ACFADR[3:0]	Acceptance Filter Address ACFADR points to a specific acceptance filter. The selected filter is accessible using the registers ACF_x. Bit SELMASK selects between acceptance code and mask for the selected acceptance filter.
31~24 15~10 7~6 4	-	Reserved

20.9.1.19 Acceptance Fliter Data Register CAN_ACF

Register	R/W	Description	Reset Value	POR
CAN_ACF	R/W	Acceptance Fliter Data Register	0x0000_0000	0xnnnn_nnnn

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

Bit number	Bit Mnemonic	Description
30	AIDEE	Acceptance Mask IDE Bit Check Enable 0: Acceptance filter accepts both standard or extended frames 1: Acceptance filter accepts either standard or extended as defined by AIDE Note: Only filter 0 is affected by the power-on reset. All other filters stay uninitialized
29	AIDE	Acceptance Mask IDE Bit Value

Bit number	Bit Mnemonic	Description
		If AIDEE=1 then: 1: Acceptance filter accepts only extended frames 0: Acceptance filter accepts only standard frames Only filter 0 is affected by the power-on reset. All other filters stay uninitialized
28~0	AMASK /ACODE[28:0]	Acceptance MASK 0: Acceptance check for these bits of receive identifier enable 1: Acceptance check for these bits of receive identifier disabled Acceptance CODE 0 - ACC bit value to compare with ID bit of the received message 1 - ACC bit value to compare with ID bit of the received message
31	-	Reserved

20.9.1.20 CAN Interrupt Control Enable Register CAN_IDE

Register	R/W	Description	Reset Value	POR
CAN_IDE	R/W	CAN Interrupt Control Enable Register	0x8000_0000	0xnnnn_nnnn

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

Bit number	Bit Mnemonic	Description
31	FDEN	FD Format Enable Bit 0: Disabled 1: Enabled
15	TXEN	CAN Transmit Enable Bit 0: Disabled 1: Enabled
14	RXEN	CAN Recieve Enable Bit 0: Disabled 1: Enabled
11	TIM_EN	TIME-stamp Counter Enable Bit 0: Disabled 1: Enabled
0	INTEN	Interrupt Request CPU Enable Control Bit 0: Disable interrupt request

Bit number	Bit Mnemonic	Description
		1: Enable interrupt request
30~16 13~12 10~1	-	Reserved

20.9.1.21 CAN Timestamp Timer Low Register CAN_TIML

Register	R/W	Description	Reset Value	POR
CAN_TIML	R/W	CAN Timestamp Timer Low Register	0x0000_0000	0xnnnn_nnnn

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

Bit number	Bit Mnemonic	Description
31~0	CAN_TIML[31:0]	64-bit Timestamp Timer Lower 32 Bits

20.9.1.22 CAN Timestamp Timer High Register Register CAN_TIMH

Register	R/W	Description	Reset Value	POR
CAN_TIMH	R/W	CAN Timestamp Timer High Register	0x0000_0000	0xnnnn_nnnn

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

Bit number	Bit Mnemonic	Description
31~0	CAN_TIMH[31:0]	64-bit Timestamp Timer Upper 32 Bits

20.9.2 CAN Register Mapping

Register	Offset Address	R/W	Description	Reset Value	POR
CAN Base Address: 0x4001_0C00					
CAN_RX_ID	0x00	R	Received Frame ID Register	0xnnnn_nnnn	0xnnnn_nnnn
CAN_RX_CTRL	0x04	R	Received Frame Control and Status Register	0xnnnn_nnnn	0xnnnn_nnnn
CAN_RBUF0	0x08	R	Received Frame Data Register0	0xnnnn_nnnn	0xnnnn_nnnn
CAN_RBUF1	0x0C	R	Received Frame Data Register1	0xnnnn_nnnn	0xnnnn_nnnn
CAN_RBUF2	0x10	R	Received Frame Data Register2	0xnnnn_nnnn	0xnnnn_nnnn
CAN_RBUF3	0x14	R	Received Frame Data Register3	0xnnnn_nnnn	0xnnnn_nnnn
CAN_RBUF4	0x18	R	Received Frame Data Register4	0xnnnn_nnnn	0xnnnn_nnnn
CAN_RBUF5	0x1C	R	Received Frame Data Register5	0xnnnn_nnnn	0xnnnn_nnnn
CAN_RBUF6	0x20	R	Received Frame Data Register6	0xnnnn_nnnn	0xnnnn_nnnn
CAN_RBUF7	0x24	R	Received Frame Data Register7	0xnnnn_nnnn	0xnnnn_nnnn
CAN_RBUF8	0x28	R	Received Frame Data Register8	0xnnnn_nnnn	0xnnnn_nnnn
CAN_RBUF9	0x2C	R	Received Frame Data Register9	0xnnnn_nnnn	0xnnnn_nnnn
CAN_RBUF10	0x30	R	Received Frame Data Register10	0xnnnn_nnnn	0xnnnn_nnnn
CAN_RBUF11	0x34	R	Received Frame Data Register11	0xnnnn_nnnn	0xnnnn_nnnn
CAN_RBUF12	0x38	R	Received Frame Data Register12	0xnnnn_nnnn	0xnnnn_nnnn
CAN_RBUF13	0x3C	R	Received Frame Data Register13	0xnnnn_nnnn	0xnnnn_nnnn
CAN_RBUF14	0x40	R	Received Frame Data Register14	0xnnnn_nnnn	0xnnnn_nnnn
CAN_RBUF15	0x44	R	Received Frame Data Register15	0xnnnn_nnnn	0xnnnn_nnnn
CAN_RTSL	0x48	R	Received Frame Timestamp Storage Lower 32 Bits Register	0xnnnn_nnnn	0xnnnn_nnnn

Register	Offset Address	R/W	Description	Reset Value	POR
CAN_RTSH	0x4C	R	Received Frame Timestamp Storage Upper 32 Bits Register	0xnnnn_nnnn	0xnnnn_nnnn
CAN_TX_ID	0x50	R/W	Transmit Frame ID Register	0xnnnn_nnnn	0xnnnn_nnnn
CAN_TX_CTRL	0x54	R/W	Transmit Frame Control and Status Register	0xnnnn_nnnn	0xnnnn_nnnn
CAN_TBUF0	0x58	R/W	Transmit Frame Data Register0	0xnnnn_nnnn	0xnnnn_nnnn
CAN_TBUF1	0x5C	R/W	Transmit Frame Data Register1	0xnnnn_nnnn	0xnnnn_nnnn
CAN_TBUF2	0x60	R/W	Transmit Frame Data Register2	0xnnnn_nnnn	0xnnnn_nnnn
CAN_TBUF3	0x64	R/W	Transmit Frame Data Register3	0xnnnn_nnnn	0xnnnn_nnnn
CAN_TBUF4	0x68	R/W	Transmit Frame Data Register4	0xnnnn_nnnn	0xnnnn_nnnn
CAN_TBUF5	0x6C	R/W	Transmit Frame Data Register5	0xnnnn_nnnn	0xnnnn_nnnn
CAN_TBUF6	0x70	R/W	Transmit Frame Data Register6	0xnnnn_nnnn	0xnnnn_nnnn
CAN_TBUF7	0x74	R/W	Transmit Frame Data Register7	0xnnnn_nnnn	0xnnnn_nnnn
CAN_TBUF8	0x78	R/W	Transmit Frame Data Register8	0xnnnn_nnnn	0xnnnn_nnnn
CAN_TBUF9	0x7C	R/W	Transmit Frame Data Register9	0xnnnn_nnnn	0xnnnn_nnnn
CAN_TBUF10	0x80	R/W	Transmit Frame Data Register10	0xnnnn_nnnn	0xnnnn_nnnn
CAN_TBUF11	0x84	R/W	Transmit Frame Data Register11	0xnnnn_nnnn	0xnnnn_nnnn
CAN_TBUF12	0x88	R/W	Transmit Frame Data Register12	0xnnnn_nnnn	0xnnnn_nnnn
CAN_TBUF13	0x8C	R/W	Transmit Frame Data Register13	0xnnnn_nnnn	0xnnnn_nnnn
CAN_TBUF14	0x90	R/W	Transmit Frame Data Register14	0xnnnn_nnnn	0xnnnn_nnnn
CAN_TBUF15	0x94	R/W	Transmit Frame Data Register15	0xnnnn_nnnn	0xnnnn_nnnn
CAN_TTSL	0x98	R/W	Transmit Frame Timestamp Storage Lower 32 Bits Register	0xnnnn_nnnn	0xnnnn_nnnn

Register	Offset Address	R/W	Description	Reset Value	POR
CAN_TTSH	0x9C	R/W	Transmit Frame Timestamp Storage Upper 32 Bits Register	0xnnnn_nnnn	0xnnnn_nnnn
CAN_CFG_STAT	0xA0	R/W	CAN Configuration and State Register	0x0090_0080	0xnnnn_nnnn
CAN_RTIE	0xA4	R/W	CAN Interrupt Control and Flag Register	0x1B00_00FE	0xnnnn_nnnn
CAN_S_SEG	0xA8	R/W	Slow Speed Clock Configuration Register	0x0102_0203	0xnnnn_nnnn
CAN_F_SEG	0xAC	R/W	Fast Speed Clock Configuration Register	0x0102_0203	0xnnnn_nnnn
CAN_EALCAP	0xB0	R/W	Delay Compensation and Transmission Error Register	0x0000_0000	0xnnnn_nnnn
CAN_ACFCTRL	0xB4	R/W	Acceptance Fliter Control Register	0x0001_0200	0xnnnn_nnnn
CAN_ACF	0xB8	R/W	Acceptance Fliter Data Register	0x0000_0000	0xnnnn_nnnn
CAN_IDE	0xCC	R/W	CAN Interrupt Control Enable Register	0x8000_0000	0xnnnn_nnnn
CAN_TIML	0xD0	R/W	CAN Timestamp Timer Low Register	0x0000_0000	0xnnnn_nnnn
CAN_TIMH	0xD4	R/W	CAN Timestamp Timer High Register	0x0000_0000	0xnnnn_nnnn

21 Hardware Watchdog WDT

21.1 Overview

The SC32F15G series features a built-in hardware watchdog (WDT) with an internal 32kHz oscillator as its clock source. Users can choose to enable the watchdog reset function by setting the ENWDT control bit in the Code Option through a programmer.

Hardware watchdog (WDT) features high security, accurate timing, and flexibility in use. This watchdog peripheral can detect and resolve faults caused by software errors, triggering a system reset when the counter reaches a predefined overflow time.

The WDT is driven by its internal low-frequency oscillator, ensuring it remains operational even in the event of a failure in the main clock.

21.2 Clock Source

The SC32F15G series WDT is fixed to LIRC. Once the WDT is enabled, LIRC will automatically start, and it will remain oscillating throughout the operation of the WDT and users cannot turn off LIRC while the WDT is active.

21.3 WDT Register

21.3.1 WDT Related Register

21.3.1.1 WDT Control Register (WDTCON)

Register	R/W	Description	Reset Value	POR
WDTCON	R/W	WDT Control Register	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
0	CLRWDT	WDT Counter Clear Bit This bit is set to 1 by software, and is automatically cleared by hardware.

Bit number	Bit Mnemonic	Description
		0:None effect 1: WDT counter count from 0
31~1	-	Reserved

21.3.1.2 WDT Configuration Register (WDTCFG)

Register	R/W	Description	Reset Value	POR
WDTCFG	R/W	WDT Configuration Register	0x0000_0000	0x0000_0000

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

<table>
<tr><th>Bit number</th><th>Bit Mnemonic</th><th>Description</th></tr>
<tr><td>2~0</td><td>WDTCKS[2:0]</td><td>Watchdog Clock Selection:
<table>
<tr><th>WDTCKS[2:0]</th><th>WDT Overflow Time</th></tr>
<tr><td>000</td><td>500ms</td></tr>
<tr><td>001</td><td>250ms</td></tr>
<tr><td>010</td><td>125ms</td></tr>
<tr><td>011</td><td>62.5ms</td></tr>
<tr><td>100</td><td>31.5ms</td></tr>
<tr><td>101</td><td>15.75ms</td></tr>
<tr><td>110</td><td>7.88ms</td></tr>
<tr><td>111</td><td>3.94ms</td></tr>
</table></td></tr>
<tr><td>31~3</td><td>-</td><td>Reserved</td></tr>
</table>

21.3.2 WDT Register Mapping

Register	Offset Address	R/W	Description	Reset Value	POR	Access Restriction
WDT Base Address:0x4000_0330						
WDTCON	0x0C	R/W	WDT Control Register	0x0000_0000	0x0000_0000	Do not support byte/half word access

Register	Offset Address	R/W	Description	Reset Value	POR	Access Restriction
WDTCFG	0x10	R/W	WDT Configuration Register	0x0000_0000	0x0000_0000	Do not support byte/half word access

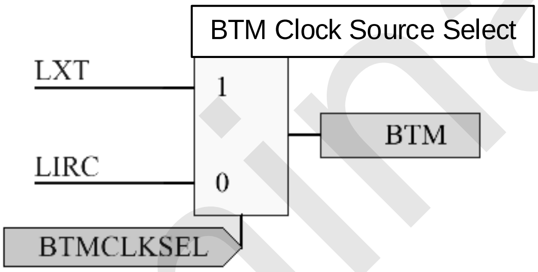
22 Base Timer(BTM)

22.1 Overview

The SC32F15G series features a Base Timer (BTM) that can generate interrupts at intervals ranging from 15.625ms to 32s. The BTM can use either 32kHz LIRC or external 32.768kHz crystal oscillator (LXT) as its clock source. The interrupts generated by the BTM can wake up the CPU from STOP mode.

22.2 Clock Sourse

SC32F15G series BTM can choose LXT or LIRC as its clock sourse



22.3 Feature

- Selectable interrupt frequency intervals from 15.625ms to 32s
- Can wake up from STOP Mode

22.4 BTM Interrupt

When the SC32F15G series BTM counter reaches the conditions set by BTMFS, the BTMIF will be set. If BTM_CON.INTEN = 1, an interrupt will be generated.

Interrupt Event	Event Flag	Interrupt Enable Control Bit
BTM interrupt request	BTMIF	BTM_CON->INTEN

22.5 BTM Register

22.5.1 BTM Related Register

22.5.1.1 BTM Control Register (BTM_CON)

Register	R/W	Description	Reset Value	POR
BTM_CON	R/W	BTM Control Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24

-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
ENBTM	INTEN	-	-	BTMFS[3:0]			

Bit number	Bit Mnemonic	Description
7	ENBTM	Base Timer Enable Control Bit 0:Base Timer disable 1:Base Timer enable
6	INTEN	Interrupt Request CPU Enable Control Bit 0: Disable interrupt request 1: Enable interrupt request
3~0	BTMFS[3:0]	BTM Interrupt Frequency 0000: generate an interrupt every 16.625ms 0001: generate an interrupt every 31.25ms 0010: generate an interrupt every 62.5ms 0011: generate an interrupt every 125ms 0100: generate an interrupt every 0.25s 0101: generate an interrupt every 0.5s 0110: generate an interrupt every 1s 0111: generate an interrupt every 2s 1000: generate an interrupt every 4s 1001: generate an interrupt every 8s 1010: generate an interrupt every 16s 1011: generate an interrupt every 32s 1100~1111: Reserved
31~8 5~4	-	Reserved

22.5.1.2 BTM Flag Register (BTM_STS)

Register	R/W	Description	Reset Value	POR
BTM_STS	R/W	BTM Flag Register	0x0000_0000	0x0000_0000

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

-	-	-	-	-	-	-	BTMIF

Bit number	Bit Mnemonic	Description
0	BTMIF	BTM Interrupt Flag This bit is set to 1 by hardware, and cleared by writing 1 through software. BTMIF will be set when BTM counter meets the conditions set by BFMFS.
31~1	-	Reserved

22.5.2 BTM Register Mapping

Register	Offset Address	R/W	Description	Reset Value	POR
BTM Base Address:0x4002_1080					
BTM_CON	0x00	R/W	BTM Control Register	0x0000_0000	0x0000_0000
BTM_STS	0x04	R/W	BTM Flag Register	0x0000_0000	0x0000_0000

23 Built-in CRC Module

23.1 Overview

The SC32F15G series has a built-in CRC (Cyclic Redundancy Check) module that utilizes a polynomial generator to generate CRC codes from an 8-bit/16-bit/32-bit data word. In numerous applications, CRC-based techniques are commonly used to verify the integrity of data transmission or storage. According to the functional safety standards, these techniques offer a means to verify the integrity of Flash. The CRC calculation unit helps compute the software signature during runtime, and this signature is then compared with the reference signature generated at link time and stored in a designated storage unit.

23.2 Clock Source

The SC32F15G series CRC has only one clock source, which is derived from HCLK.

23.3 Feature

- 1 built-in hardware CRC module
- Configurable initial value, with a default of 0xFFFF_FFFF
- Supports 8-bit/16-bit/32-bit data units
- Programmable polynomial, with a default of 0x04C1_1DB7
- Only supports software-driven data computation mode
- Supports DMA: CRC_DR can serve as the DMA destination address or be accessed directly via registers
- Calculating CRC for a single byte requires 1 system clock

CRC algorithm	CRC-32/MPEG-2
Polynomial Formula	$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^{8}+x^{7}+x^{5}+x^{4}+x^{2}+x+1$
Data Width	32bit
Inital Value	0xFFFF_FFFF
Result XOR Value	0x0000_0000
Input Value Reversal	false
Output Value Reversal	false
LSB/MSB	MSB

Note: The written and read data in CRCDR cannot be the same.

23.4 CRC Register

23.4.1 CRC Related Register

23.4.1.1 CRC Data Register (CRC_DR)

Register	R/W	Description	Reset Value	POR
CRC_DR	R/W	CRC Data Register (calculation result)	0xFFFF_FFFF	0x0000_0000

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

Bit number	Bit Mnemonic	Description
31~0	CRCDR[31:0]	CRC Data Register This register is used to write new data to the CRC calculator. When reading the register, the previous CRC calculation result can be obtained. If the data size is less than 32 bits, the least significant bits can be used to write/read the correct value. The operation requirements for this register are as follows: 1. First, CRC_CON.CRCRST need to be set to 1 to reset CRCDR 2. When "CRCREG" is written, the hardware automatically calculates the CRC result and continues to store it in CRCDR When needed, read out the CRC calculation result instantly.

23.4.1.2 CRC Control Register (CRC_CON)

Register	R/W	Description	Reset Value	POR
CRC_CON	R/W	CRC Control Register	0x0000_0000	0x0000_0000

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

POLYSIZE[1:0]	-	-	-	-	-	CRCRST

Bit number	Bit Mnemonic	Description
7~6	POLYSIZE[1:0]	CRC Polynomial Size Setting Bits 00:32 bits polynomial 01: 16 bits polynomial 10: 8 bits polynomial 11: 7 bits polynomial
0	CRCRST	CRCDR Register Reset Bit(Q31~Q0) This bit is set to 1 by software, and is automatically cleared by hardware. 0: None effect 1: Reset CRCDR, and the reset value is the value of CRC_INIT register user write in.
31~8 5~1	-	Reserved

23.4.1.3 CRC Initial Value Register (CRC_INT)

Register	R/W	Description	Reset Value	POR
CRC_INT	R/W	CRC Initial Value Register	0xFFFF_FFFF	0x0000_0000

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

Bit number	Bit Mnemonic	Description
31~0	CRC_INIT[31:0]	Programmable CRC intial value, reset value:0xFFFF FFFF This register is used for users to write in CRC initial value.

23.4.1.4 CRC Polynomial Setting Register (CRC_POL)

Register	R/W	Description	Reset Value	POR
CRC_POL	R/W	CRC Polynomial Setting Register	0x04C1_1DB7	0x0000_0000

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

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

Bit number	Bit Mnemonic	Description
31~0	POL[31:0]	Programmable polynomial, reset value:0x04C1_1DB7 This register is used to write the coefficients of the polynomial to be used for CRC calculation. If the polynomial size is less than 32 bits, the least significant bits must be used to program the correct values.

23.4.2 CRC Register Mapping

Register	Offset Address	R/W	Description	Reset Value	POR	Access Restriction
CRC Base Address:0x4000_2000						
CRC_DR	0x00	R/W	CRC Data Register	0xFFFF_FFFF	0x0000_0000	
CRC_CON	0x04	R/W	CRC Control Register	0x0000_0000	0x0000_0000	
CRC_INT	0x08	R/W	CRC Initial Value Register	0xFFFF_FFFF	0x0000_0000	Do not support byte/half word access
CRC_POL	0x0C	R/W	CRC Polynomial Setting Register	0x04C1_1DB7	0x0000_0000	Do not support byte/half word access

24 Direct Memory Access (DMA)

24.1 Overview

The DMA controller is designed for high-speed data transfer, allowing the movement of data from one address to another without the need for CPU intervention. Leveraging DMA for data transfer can reduce the workload on the CPU, enabling the saved CPU resources to be utilized for other applications. The DMA controller comprises 4 channels, each directly connected to dedicated hardware DMA requests. Additionally, each channel supports software triggering. The DMA controller features support for 4-level channel priority, facilitating the management of priority between DMA requests to ensure that only one DMA channel operates at any given time. It also supports both single and batch transfers, with the request source being either a software request or an interface request. Data transfer between memories is accomplished using software requests.

Note: For a bidirectional data transfer application, two DMA channels are required to handle sending and receiving operations.

24.2 Clock Source

The clock source of DMA is derived from HCLK, and the external peripheral clock of DMA is enabled through AHB_CFG.DMAEN.

24.3 Feature

- Support 4 independent configurable channels
- Support 4 priority levels for requests
- Support 8-bit, 16-bit, 32-bit data transfers
- Support automatic increment or fixed source and destination addresses, with data widths of byte, half-word, and word
- Support single and burst transfer modes

24.4 Function Description

24.4.1 Transmission

No transmit limitation between peripheral and memory for DMA:

Memory-to-Memory	Memory-to-Peripheral	Peripheral-to-Memory	Peripheral-to-Periphral
No limitation	No limitation	No limitation	No limitation

24.4.2 DMA Access Restriction

Users are not allowed to perform write operations on Flash or access the core through DMA. Violating these restrictions may lead to unpredictable exceptions.

24.4.3 Channel Priority

There are 4 priority levels can be configured through PL[1:0]:

- 00: Low
- 01: Medium
- 10: High
- 11: Very High

24.4.4 Single Transmission and Burst Transmission

The DMA controller supports single and burst data transfer types, and the request source can be a software request or an interface request while data transfer between memory is done by software requests. Single transfer means that the software or interface is ready to transfer one data (each data requires one request), while burst transfer means that the software or interface will transfer multiple data (multiple data requiring only one request).

The modes of single and burst transfer can be set through TPTYPE register (DMAn_CFG[15]).

In single transfer mode, each transfer of data requires one request. As each data is transferred, the values in the register DMAn_CNT[31:0](n=0~3) decrease by 1, the transfer of data is completed when the count in DMAn_CNT[31:0] becomes 0. In this mode, BURSIZE (DMAn_CFG[14:12]) is not used to control the size of the transferred data and its value is fixed at 1.

In burst transfer mode, DMA transfer DMAn_CNT[31:0] data with only one request. After transferring BURSIZE (DMAn_CFG[14:12]) data, the value in DMAn_CNT[31:0] is decreased by BURSIZE. The transfer of data is completed when the count in DMAn_CNT[31:0] becomes 0.

24.4.5 Loop Mode

The loop mode can be used to handle circular buffers and continuous data streams (such as ADC scan mode). During the loop mode transfer, the number of data to be transferred will automatically reload to the initial value set in the channel configuration phase and continue to respond to DMA requests. To stop loop transfer, the software needs to stop the generation of DMA requests by the peripheral before disabling the DMA channel (for example, exiting ADC scan mode). The software must explicitly set the DMACNT value before starting/enabling the transfer and after stopping the loop transfer.

The SC32F15G series DMA controller supports normal mode and loop mode:

- When CIRC=0 (DMA channel is in non-loop mode), it will no longer accept any DMA requests after reaching the set number of data to be transferred
- When CIRC=1 (DMA channel is in loop mode), after the transfer is complete, the DMACNT of the channel will automatically reload the previously set value and wait for the next loop

Users can flexibly choose according to their actual needs.

24.4.6 DMA Channel Control Bit Restrictions After Enable

Once a DMA channel is enabled (CHEN = 1), certain control bits become read-only to prevent modifications during an active DMA transfer, which could lead to unpredictable data transmission behavior.

After DMA channel is enabled, register bit fields/bits, source and destination addresses, priority settings and transfer control configurations are locked

24.5 DMA Interrupt

For each DMAn (n=0~3) channel, an interrupt will be generated when "transmission complete," "half transmission," or "transmission error." Separate interrupt enable bits can be used to enhance flexibility.

Interrupt Event	Event Flag	Interrupt Request Control Bit	Sub-Event Flag	Interrupt Enable Sub-Switch
DMAn transmission complete	GIF	DMAn_CFG ->INTEN	TCIF	TCIE
DMAn half transmission			HTIF	HTIE
DMAn transmission error			TEIF	TEIE

24.6 DMA Register

24.6.1 DMA Related Register

24.6.1.1 DMAn Transmission Source Address Cache Register (DMAn_SADR)

Register	R/W	Description	Reset Value	POR
DMAn_SADR n = 0~3	R/W	DMAn Transmission Source Address Cache Register	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
31~0	SADR[31:0]	DMA Transmission Source Address Cache ● Read: ■ When the channel is enabled, what is read is the internal source address working register. ■ When the channel is disabled, what is read is the apparent source address cache register. ● Update: ■ After each transmission, the source address working register will automatically change based on the SAINC[1:0] settings, and the width of the change is determined by TXWIDTH[1:0]. ■ In the loop mode (SAINC = 11), the source address cache register will reload into the source address working register.

Bit number	Bit Mnemonic	Description
		● Write: ■ The conditions for writing to the source address cache register: CHEN=1, and DMA channel has completed the transmission and stay in the IDLE state, or CHEN=0.

24.6.1.2 DMAn Transmission Target Address Cache Register (DMAn_DADR)

Register	R/W	Description	Reset Value	POR
DMAn_DADR n = 0~3	R/W	DMAn Transmission Target Address Cache Register	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
31~0	DADR[31:0]	DMA Transmission Target Address Cache ● Read: ■ When the channel is enabled, what is read is the internal target address working register. ■ When the channel is disabled, what is read is the apparent target address cache register. ● Update: ■ After each transmission, the target address working register will automatically change based on the DAINC[1:0] settings, and the width of the change is determined by TXWIDTH[1:0]. ■ In the loop mode (SAINC = 11), the target address cache register will reload into the target address working register. ● Write: ■ The conditions for writing to the target address cache register: first, CHEN=1, and DMA channel has completed the transmission and stay in the IDLE state, or CHEN=0.

24.6.1.3 DMAn Control/Configuration Register (DMAn_CFG)

Register	R/W	Description	Reset Value	POR
DMAn_CFG n = 0~3	R/W	DMAn Control/Configuration Register	0x0000_0000	0x0000_0000

31	30	29	28	27	26	25	24
-	-	REQSRC[5:0]					
23	22	21	20	19	18	17	16
CHRQ	-	-	-	TEIE	HTIE	TCIE	INTEN
15	14	13	12	11	10	9	8
TPTYPE	BURSIZE[2:0]			SAINC[1:0]		DAINC[1:0]	
7	6	5	4	3	2	1	0
CHEN	CHRST	PAUSE	CIRC	TXWIDTH[1:0]		PL[1:0]	

Bit number	Bit Mnemonic	Description
29~24	REQSRC[5:0]	DMA Channel Request Source Selection Bit 0: Disable peripheral request for the current DMA channel Select the following configuration values, if the peripheral DMA request enable in the selected setting, corresponding request source will be generated: 2: UART0_IDE->TXDMAEN 3: UART0_IDE->RXDMAEN 4: UART1_IDE->TXDMAEN 5: UART1_IDE->RXDMAEN 12: SPI0_IDE->TXDMAEN 13: SPI0_IDE->RXDMAEN 14: SPI1_TWI1_IDE->TXDMAEN 15: SPI1_TWI1_IDE->RXDMAEN 20: TWI0_IDE->TXDMAEN 21: TWI0_IDE->RXDMAEN 33: TIM1_IDE->TIDE 34: TIM1_IDE->CAPFDE 35: TIM1_IDE->CAPRDE 36: TIM2_IDE->TIDE 37: TIM2_IDE->CAPFDE 38: TIM2_IDE->CAPRDE 59: ADCCON->DMAEN 60: DMA0_CFG->CHRQ 61: DMA1_CFG->CHRQ 62: DMA2_CFG->CHRQ 63: DMA3_CFG->CHRQ Others:Disable DMA peripheral request
23	CHRQ	DMA Request Enable Bit for DMA Channel: 0: Disable, the current DMA channel is prohibited from serving as the request source for other DMA channels 1: Enable, the current DMA channel can serve as the request source for other DMA channels, meaning it can generate DMA requests. like other peripherals When this bit is enabled, it allows DMA to request DMA.For example:

Bit number	Bit Mnemonic	Description
		If CHRQ =1, after DMA channell n completes data transmission, it will generate a DMA request to DMA channel m. Channel m will respond to the request and update the pre-configured parameter table to the register of channel n, thereby achieving automatic parameter updates for channel n. Note: After CHRQ is set, the DMA acting as the request source is able to perform data transfer, but it will not set the flag or enter the corresponding interrupt. The flag will only be set and the interrupt will only be entered after CHRQ is cleared to 0
19	TEIE	DMA Transmission Error Interrupt Enable Bit 0: Disable DMA transmission error interrupt 1: Enable DMA transmission error interrupt
18	HTIE	DMA Half Transmission Interrupt Enable Bit 0: Disable DMA half transmission interrupt 1: Enable DMA half transmission interrupt
17	TCIE	DMA Transmission Complete Interrupt Enable Bit 0: Disable DMA transmission complete interrupt 1: Enable DMA transmission complete interrupt
16	INTEN	Interrupt Request CPU Enable Control Bit 0: Disable interrupt request 1: Enable interrupt request
15	TPTYPE	DMA Channel Transmission Type Selection Bit 0: Single transmission 1: Burst transmission. In burst transmission mode, The DMA controller moves DMACNT data with just one request. Once the channel responds to this request, the data will be transferred in a burst mode, meaning it moves in units of BURSIZE until DMACNT decrements to 0. The data processing for a single burst transfer is considered complete only when DMACNT reaches zero.
14~12	BURSIZE[2:0]	In Burst transmission, based on the definition of Burst transmission mode, burst size can be selected as: 000: 128 001: 64 010: 32 011: 16 100: 8 101: 4 110: 2 111: 1
11~10	SAINC[1:0]	DMA Channel Transmission Source Address Increment/Decrement Mode Configuration Bit 00: No increment (Fixed address mode) 01: Increment mode

Bit number	Bit Mnemonic	Description
		10: Decrement mode 11: Incremental circular mode (Refer to the DMA transmission source address cache register) The values of SAINC[1:0] can be modified freely and take effect immediately when the channel is disabled. When the channel is enabled, the modified values will take effect during the reload in circular mode.
9~8	DAINC[1:0]	DMA Channel Transmission Target Address Increment/Decrement Mode Configuration Bit 00: No increment (Fixed address mode) 01: Increment mode 10: Decrement mode 11: Incremental circular mode (Refer to the DMA transmission target address cache register) The values of DAINC [1:0] can be modified freely and take effect immediately when the channel is disabled. When the channel is enabled, the modified values will take effect during the reload in circular mode.
7	CHEN	DMA Channel Enable Bit 0: Disable DMA channel 1: Enable DMA channel
6	CHRST	DMA Channel Reset Control Bit This bit is used to control the reset of DMA channel. 0: Invalid 1: Reset the current DMA channel. At this point, CHEN for the current DMA channel is disabled, the interrupt flag is cleared, and the values of other registers remain unchanged.
5	PAUSE	DMA Channel Transfer Pause Control Bit 0: Invalid 1: Pause the current DMA channel. At this point, CHEN for the current DMA channel is disabled, and the state machine returns to state=1 after completing the current read/write cycle. The internal register values (source/destination address register, counters) are maintained. To resume the transfer gain, users need to set CHEN=1 and PAUSE=0 simultaneously.
4	CIRC	DMA Channel Loop Mode Enable Bit 0: The channel is not in loop mode. When the set number of data to be transferred is reached, the DMACNT for that channel will remain at zero. 1: The channel is in loop mode. After the transfer is complete, the DMACNT for that channel will automatically reload the previously set value.

Bit number	Bit Mnemonic	Description
		Loop mode can be used for handling circular buffers and continuous data streams (such as ADC scan mode). During the loop mode transfer, the number of data to be transferred will automatically reload to the initial value set during the channel configuration phase, and the channel will continue to respond to DMA requests. To stop the loop transfer, software needs to stop the peripheral from generating DMA requests before disabling the DMA channel (for example, exiting ADC scan mode). Software must explicitly set the DMACNT value before starting/enabling the transfer and after stopping the loop transfer.
3~2	TXWIDTH[1:0]	DMA Channel Transmission Width Selection Bit Choose the data width of the source and target addresses for each transmission of the current DMA channel: 00: 8bit 01: 16bit 10: 32bit 11: 32bit The values of TXWIDTH[1:0] can be freely modified and take effect immediately when the channel is disabled. When the channel is enabled, the modified values will take effect during the reload in loop mode.
1~0	PL[1:0]	DMA Channel Priority Setting Bit When DMA has a channel in operation, and other channels also receive requests but are pending, priority arbitration will be initiated once the currently active channel completes its operation. 00: Low 01: Medium 10: High 11: Very High **Note: For equal priority configurations, lower channel numbers have higher priority.**
31~30 22~20	-	Reserved

24.6.1.4 DMAn Counter Cache Register (DMAn_CNT)

Register	R/W	Description	Reset Value	POR
DMAn_CNT n = 0~3	R/W	DMAn Counter Cache Register	0x0000_0000	0x0000_0000

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

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

Bit number	Bit Mnemonic	Description
31~0	DMACNT[31:0]	DMA Channel Counter Cache Register ● Write: ■ The value of DMACNT refers to the remaining transfer count for the current DMA channel. ■ Each DMA channel has an internal “working counter” that decrements by the TXWIDTH units after each transmission: ◆ When CIRC=0 (DMA channel is not in loop mode), the 'working counter' will stop accepting any further DMA requests after decrementing to 0. ◆ When CIRC=1 (DMA channel is in loop mode), after the “working counter” decrements to 0, it will reload the value of DMACNT into the “working counter” and wait for the next loop. ● Read: ■ When the channel is disabled, reading returns the value of DMACNT. ■ When the channel is enabled, reading returns the real-time data of the internal “working counter”.

24.6.1.5 DMAn Status Register (DMAn_STS)

Register	R/W	Description	Reset Value	POR
DMAn_STS n = 0~3	R/W	DMAn Status Register	0x0000_0000	0x0000_0000

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

Bit number	Bit Mnemonic	Description
8	SWREQ	DMA Channel Software Request Trigger Bit

Bit number	Bit Mnemonic	Description
		When this bit is written to 1, the current DMA channel will remain pending software requests until the channel responds, and this bit is automatically cleared by hardware.
7~4	STATUS[3:0]	DMA Channel Status Bit 0000: Idle 0001: Write to source address 0010: Read source address data and write to target address 0011: Write to target address data 0100: Reserved 0101: Pending (When a channel is busy, requests from other channels are suspended.) 0110: Pause pending (In burst transmission mode, after writing PAUSE to 1) 0111: Burst transmission in progress 1000:Burst transmission stopped(PAUSE is enabled, DMACNT counts to 0, or bursize counts to 0, will enter this state.)
3	TEIF	DMA Transmission Error Interrupt Flag When DMA reads or writes to an undefined address, TEIF will be set to 1 by the hardware. Writing 1 clears the bit to zero.
2	HTIF	DMA HalfTransmission Interrupt Flag When the counter value of DMACNT reaches DMACNT/2, HTIF will be set to 1 by the hardware. Writing 1 clears the bit to zero.
1	TCIF	DMA Transmission Complete Interrupt Flag When the counter value of DMACNT reaches 0, TCIF will be set to 1 by the hardware. Writing 1 clears the bit to zero.
0	GIF	DMA Channel Global Interrupt Flag 0: The current DMA channel has no interrupt generated. 1: The current DMA channel has generated an interrupt: transmission error, half-transmission, or transmission complete.
31~9	-	Reserved

24.6.2 DMA Register Mapping

Register	Offset Address	R/W	Description	Reset Value	POR
DMA0 Base Address:0x4001_0800					
DMA0_SADR	0x00	R/W	DMA0 Transmission Source Address Cache Register	0x0000_0000	0x0000_0000

Register	Offset Address	R/W	Description	Reset Value	POR
DMA0_DADR	0x04	R/W	DMA0 Transmission Target Address Cache Register	0x0000_0000	0x0000_0000
DMA0_CFG	0x08	R/W	DMA0 Control/Configuration Register	0x0000_0000	0x0000_0000
DMA0_CNT	0x0C	R/W	DMA0 Counter Cache Register	0x0000_0000	0x0000_0000
DMA0_STS	0x10	R/W	DMA0 Status Register	0x0000_0000	0x0000_0000
DMA1 Base Address:0x4001_0840					
DMA1_SADR	0x00	R/W	DMA1 Transmission Source Address Cache Register	0x0000_0000	0x0000_0000
DMA1_DADR	0x04	R/W	DMA1 Transmission Target Address Cache Register	0x0000_0000	0x0000_0000
DMA1_CFG	0x08	R/W	DMA1 Control/Configuration Register	0x0000_0000	0x0000_0000
DMA1_CNT	0x0C	R/W	DMA1 Counter Cache Register	0x0000_0000	0x0000_0000
DMA1_STS	0x10	R/W	DMA1 Status Register	0x0000_0000	0x0000_0000
DMA2 Base Address:0x4001_0880					
DMA2_SADR	0x00	R/W	DMA2 Transmission Source Address Cache Register	0x0000_0000	0x0000_0000
DMA2_DADR	0x04	R/W	DMA2 Transmission Target Address Cache Register	0x0000_0000	0x0000_0000
DMA2_CFG	0x08	R/W	DMA2 Control/Configuration Register	0x0000_0000	0x0000_0000
DMA2_CNT	0x0C	R/W	DMA2 Counter Cache Register	0x0000_0000	0x0000_0000
DMA2_STS	0x10	R/W	DMA2 Status Register	0x0000_0000	0x0000_0000
DMA3 Base Address:0x4001_08C0					
DMA3_SADR	0x00	R/W	DMA3 Transmission Source Address Cache Register	0x0000_0000	0x0000_0000
DMA3_DADR	0x04	R/W	DMA3 Transmission Target Address Cache Register	0x0000_0000	0x0000_0000
DMA3_CFG	0x08	R/W	DMA3 Control/Configuration Register	0x0000_0000	0x0000_0000
DMA3_CNT	0x0C	R/W	DMA3 Counter Cache Register	0x0000_0000	0x0000_0000
DMA3_STS	0x10	R/W	DMA3 Status Register	0x0000_0000	0x0000_0000

25 SysTick

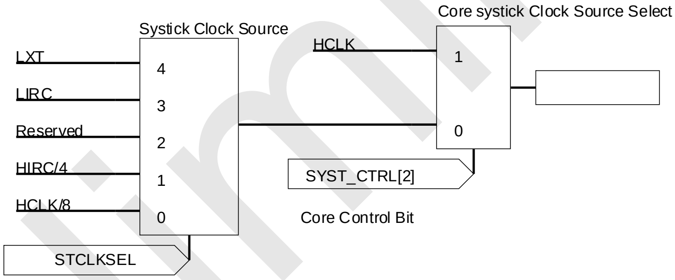
SysTick is a simple, 24-bit, writable-clear, decrementing automatic reload counter with a flexible control mechanism. This counter can be used as a tick timer for a Real-Time Operating System (RTOS) or as a simple counter.

25.1 Clock Source

SysTick (Cortex®-M0+ Core System Timer) has internal clock source and external clock source:

- Internal clock source: CPU Clock
- 4 external clock sources

SysTick clock sourse diagram is as follow:



25.2 SysTick Calibration Register Default Value

The calibration value for the SysTick Calibration Register is set as follows:

- If the default clock after power-up is f_{HCLK}/n (MHz), (n is the default division factor after power-up, and HIRC is the default clock source after power-up).
- Then the SysTick calibration initial value is set to $1000*(f_{HCLK}/n)$, this ensures that a default 1ms time base can be generated.

26 Revision History

Version	Notes	Date
V0.1	Initial Release	2025.02.14

27 Important Notice

Shenzhen SinOne Microelectronics Co., Ltd. (hereinafter referred to as SinOne) reserves the right to change, correct, enhance, modify and improve SinOne products, documents or services at any time without prior notice. SinOne considers the information provided to be accurate and reliable. The information in this document will be used in February 2025. In the actual production design, please refer to the latest data manual of each product and other relevant materials.