

1 Introduction

This reference manual supplements the SC32F11T/11G Series datasheet by providing essential information for application development, particularly software. For specific device features, ordering information, and mechanical and electrical specifications, refer to the respective datasheet.

Preliminary

Content

1	Introduction	1
	Content.....	2
2	Document conventions	9
2.1	Glossary	9
2.2	Peripheral availability.....	9
3	Resource diagram	10
4	Flash.....	11
4.1	Overview.....	11
4.2	Storage block diagram.....	11
4.3	Feature	12
4.4	APROM (application program ROM).....	12
4.5	1.5 Kbytes user storage area (generic EEPROM)	14
4.6	4 Kbytes LDROM (system storage area)	14
4.6.1	BootLoader	15
4.7	SRAM	15
4.8	Boot area selection (bootstrap)	15
4.8.1	Bootstrap from application program ROM.....	16
4.8.2	Bootstrap from system memory	16
4.8.3	Bootstrapping from embedded SRAM	16
4.8.4	Bootstrap mode configuration	16
4.9	96 bits Unique ID	16
4.10	User ID area	16
4.11	Programming	17
4.11.1	JTAG specific mode	17
4.11.2	Normal mode (JTAG specific pins disabled)	17
4.12	Security encryption	18
4.12.1	Security encryption access rights.....	18
4.13	In application programming (IAP).....	18
4.13.1	IAP operation related registers.....	19
4.14	Option bytes area (Customer Option)	21
4.14.1	Mapping register for Customer Option	22
5	Power, reset and system clock (RCC).....	24
5.1	Power-on process.....	24
5.1.1	Reset stage	24
5.1.2	Loading information stage	24
5.1.3	Normal operation stage	24
5.2	Reset	24

5.2.1	Hardware reset.....	25
5.2.2	Software reset	26
5.2.3	Boot area after reset.....	26
5.2.4	Initial reset state	26
5.3	Clock.....	26
5.3.1	System clock source	26
5.3.2	Built-in high-frequency 64 MHz oscillator (HIRC)	27
5.3.3	Built-in high-frequency crystal oscillator circuit, can be externally connected to a 2 to 16 MHz high-frequency oscillator (HXT)	27
5.3.4	Built-in low-frequency 37 kHz oscillator (LIRC)	27
5.3.5	Built-in low frequency oscillator circuit, with support for external 32.768 kHz low frequency oscillator (LXT)	28
5.3.6	PLL	28
5.4	Register access clock and peripheral clock source	29
5.4.1	Register access clock.....	29
5.4.2	Peripheral clock source	29
5.4.3	Bus clock	29
5.4.4	Peripheral clock.....	31
5.4.5	Peripheral clock block diagram	33
5.4.6	Peripheral clock enable register.....	33
5.4.7	Low-power mode clock.....	33
5.5	RCC interrupt.....	33
5.6	RCC register	34
5.6.1	RCC register table	34
5.6.2	RCC register mapping.....	47
6	Interrupts.....	49
6.1	External interrupts INT0 to INT15.....	49
6.2	Interrupt and events.....	49
6.3	Interrupt vector table.....	50
6.4	External interrupt registers	54
6.4.1	External interrupt related registers table	54
6.4.2	External interrupt register mapping	57
7	Power saving mode	59
8	GPIO	60
8.1	Clock source	60
8.2	Feature	60
8.3	GPIO block diagram	60
8.3.1	Strong push-pull output mode	60
8.3.2	Input mode with pull-up	61
8.3.3	High impedance input mode (input only).....	61
8.4	GPIO register.....	61

8.4.1	GPIO related register table.....	61
8.4.2	GPIO register mapping.....	64
9	Analog-to-digital converter (ADC).....	65
9.1	Clock source.....	65
9.2	Feature	65
9.3	ADC conversion steps	65
9.4	ADC interrupts	66
9.5	ADC connection circuit diagram	66
9.6	ADC registers	66
9.6.1	ADC register table	66
9.6.2	ADC register mapping	69
10	Operational amplifier and programmable gain amplifier (OP).....	70
10.1	Feature	70
10.2	OP port selection	70
10.2.1	OP precision adjustment	70
10.2.2	OP non-inverting input selection	70
10.2.3	OP inverting input selection.....	70
10.2.4	OP output selection	70
10.3	Block diagram of the OP circuit structure	71
10.4	OP register.....	71
10.4.1	OP related register table	71
10.4.2	OP register mapping	72
11	Analog comparator (CMP).....	73
11.1	Feature	73
11.2	Analog comparator block diagram.....	73
11.3	CMP register.....	74
11.4	CMP related registers table	74
11.4.1	Analog comparator status register (CMP_STS).....	74
11.4.2	Analog comparator configuration register (CMP_CFG).....	74
11.5	CMP register mapping	75
12	UART0 to 5	76
12.1	Clock source.....	76
12.2	Feature	76
12.3	UART interrupt.....	76
12.4	UART0/1/2/3/4/5 registers	77
12.4.1	Register table for UART0/1/2/3/4/5	77
12.4.2	UART0/1/2/3/4/5 register mapping.....	81
13	QSPI0 to 1	83
13.1	Clock source.....	83
13.2	QSPI0/1 feature.....	83

13.3	Pin description	83
13.4	Command sequence	83
13.4.1	8-bit quad-line waveform diagram.....	85
13.4.2	16-bit four-wire waveform.....	86
13.4.3	24-bit four-wire waveform.....	86
13.4.4	32-bit four-wire waveform.....	86
13.5	Transmission line mode.....	87
13.5.1	Single-wire transmission mode	87
13.5.2	Dual-wire transmission mode	87
13.5.3	Four-wire transmission mode	88
13.6	Bypass mode	88
13.7	QSPI configuration method	89
13.7.1	Half-duplex QSPI mode configuration procedure	89
13.7.2	Pass-through mode configuration procedure	89
13.8	QSPI interrupts	90
13.9	QSPI0/1 registers	90
13.9.1	QSPI0/1 related register table	90
13.9.2	QSPI0/1 register mapping	96
14	SPI0 to 3	97
14.1	Clock source	97
14.2	SPI0/1 feature.....	97
14.3	SPI2/3 features.....	97
14.4	Signal description	98
14.5	Operating mode.....	98
14.5.1	Master mode	99
14.5.2	Slave mode	99
14.6	Transmission format.....	100
14.7	Error detection	101
14.8	Comparison between SPI0/1 and SPI2/3.....	101
14.9	SPI0/1 registers	102
14.9.1	SPI0/1 related register table.....	102
14.9.2	SPI0/1 register mapping.....	105
14.10	SPI2/3 registers	106
14.10.1	SPI2/3 related registers table	106
14.10.2	SPI2/3 register mapping.....	109
15	TWI0 to 3	111
15.1	Clock source	111
15.2	TWI0/1 features	111
15.3	Features of TWI2 and TWI3	111
15.4	TWI signal description	111

15.5	Slave operating mode.....	112
15.6	Slave mode operation procedure	114
15.7	Master operation mode.....	115
15.8	Master mode operation procedure	116
15.9	TWI0/1 interrupts	117
15.10	TWI2/3 interrupts	117
15.11	TWI0/1 registers	118
15.11.1	TWI0/1 related register table	118
15.11.2	TWI0/1 register mapping	122
15.12	TWI2/3 registers	123
15.12.1	TWI2/3 related register table	123
15.12.2	TWI2/3 register mapping	128
16	CAN communication port.....	129
16.1	Overview.....	129
16.2	Clock source	129
16.3	Feature	129
16.4	CAN protocol	130
16.4.1	Can2.0 protocol	130
16.4.2	CAN_FD protocol	132
16.5	Message RAM	135
16.6	Function description	137
16.6.1	Baud rate configuration	137
16.6.2	Filter function	137
16.6.3	Transmit buffer	138
16.6.4	Automatic retransmission mask	138
16.7	Operating mode.....	138
16.8	CAN communication configuration	140
16.8.1	CAN module initialization	140
16.8.2	Transmit message configuration	140
16.8.3	Transmit process	141
16.8.4	Receive process	141
16.8.5	Message transmission priority	141
16.9	CAN interrupts	141
16.10	CAN register	142
16.10.1	CAN related register table	142
16.10.2	CAN register mapping	158
17	Hardware watchdog WDT.....	161
17.1	Clock source	161
17.2	WDT registers.....	161
17.2.1	WDT related register table	161

17.2.2	WDT register mapping	162
18	Base timer (BTM).....	163
18.1	Clock source.....	163
18.2	Feature	163
18.3	BTM interrupt.....	163
18.4	BTM register	163
18.4.1	BTM-related register table.....	163
18.4.2	BTM register mapping	165
19	Built-in CRC verification module.....	166
19.1	Clock source.....	166
19.2	Feature	166
19.3	CRC register.....	166
19.3.1	CRC related register table.....	166
19.3.2	CRC register mapping.....	168
20	PWM0: 8-channel 16-bit multifunction PWM.....	170
20.1	Overview.....	170
20.2	Clock source.....	170
20.3	Feature	170
20.4	PWM0 block diagram	171
20.5	PWM0 general configuration	171
20.5.1	Output mode.....	171
20.5.2	Alignment type.....	172
20.5.3	Duty cycle variation feature	173
20.5.4	Period change feature	173
20.5.5	Relationship between period and duty cycle.....	174
20.6	PWM0 independent mode.....	174
20.7	PWM0 complementary mode	175
20.7.1	PWM0 complementary mode dead time configuration	176
20.7.2	PWM0 dead time output waveform	176
20.8	PWM0 fault detection function configuration	176
20.9	PWM0 interrupt.....	177
20.10	PWM0 registers	177
20.10.1	PWM0 related register table.....	177
20.10.2	PWM0 register mapping.....	182
21	LCD/LED driver.....	183
21.1	Clock source.....	183
21.2	Built-in 8 COM x 31 SEG LED driver.....	183
21.3	Built-in 8 COM x 31 SEG LCD driver	183
21.4	LCD/LED registers.....	184
21.4.1	LCD/LED related register table	184

21.4.2	LCD/LED register mapping	188
22	42-channel high sensitivity touch circuit (TK)	190
23	16-bit timers/counters (TIM) Timer0 to Timer7	191
23.1	Clock source	191
23.2	Feature	191
23.3	Counting method	191
23.3.1	TIM counting method in timer mode	191
23.3.2	TIM counting method in PWM output mode	191
23.4	Timer-related signal pins	191
23.5	TIM interrupts and their corresponding flags	192
23.6	TIM operating modes	192
23.6.1	Mode 0: 16-bit capture PWM both-edge capture mode	192
23.6.2	Mode 1: 16-bit auto-reload timer/counter mode	193
23.6.3	Mode 3: programmable clock output mode	193
23.6.4	Mode 4: PWM output mode	193
23.7	TIM interrupt	194
23.8	TIM registers	194
23.8.1	TIM-related register table	194
23.8.2	TIM register mapping	201
24	DMA controller	205
24.1	Overview	205
24.2	Clock source	205
24.3	Feature	205
24.4	Function description	205
24.4.1	Transfer direction	205
24.4.2	DMA access region restriction	205
24.4.3	Channel priority	205
24.4.4	Single transfer and burst transfer	206
24.5	Loop mode	206
24.6	DMA interrupts	206
24.7	DMA registers	207
24.7.1	DMA register table	207
24.7.2	DMA register mapping	212
25	Systick	214
25.1	Clock source	214
25.2	Systick calibration register default value	214
26	Revision history	215
27	Important notice	216

2 Document conventions

2.1 Glossary

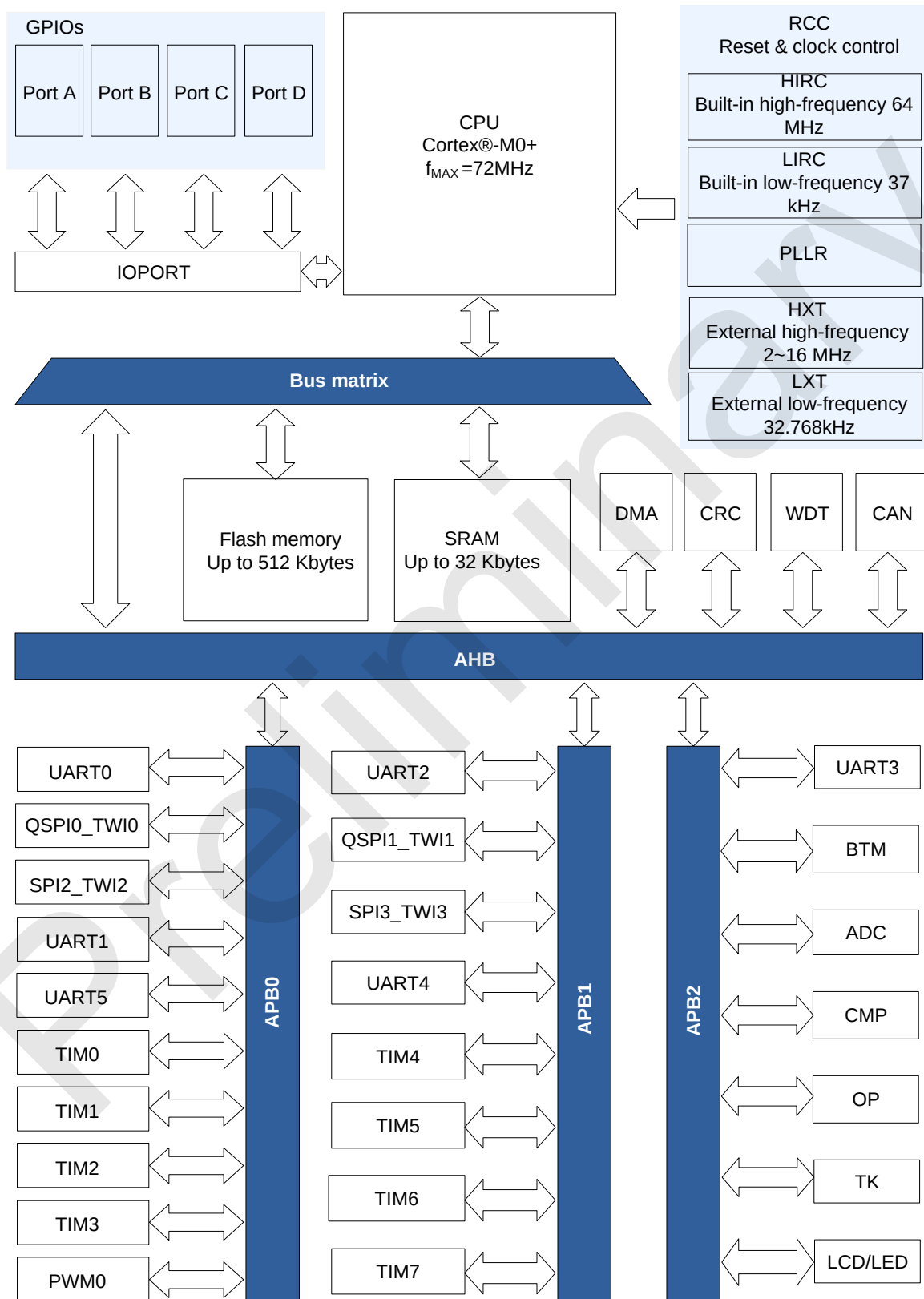
This section explains the definitions of acronyms and abbreviations used in this document:

- Word: 32-bit data
- Half-word: 16-bit data
- Byte: 8-bit data
- Double word: 64-bit data
- IAP (In-Application Programming): refers to reprogramming the MCU's Flash memory during user program execution
- ICP (In-Circuit Programming): Programming the microcontroller's Flash via JTAG, SWD, or bootstrap while the device is installed on the user's board.
- JTAG Protocol: An international standard test protocol primarily used for internal chip testing.
- SWD Protocol: ARM-designed Serial Wire Debug protocol for programming and debugging microcontrollers.
- Option Bytes: Configuration bits stored in Flash.
- AHB: Advanced High-performance Bus.
- APB: Advanced Peripheral Bus.

2.2 Peripheral availability

Refer to the Product Peripheral Resource Table in the latest datasheet for peripheral availability and quantity per model.

3 Resource diagram

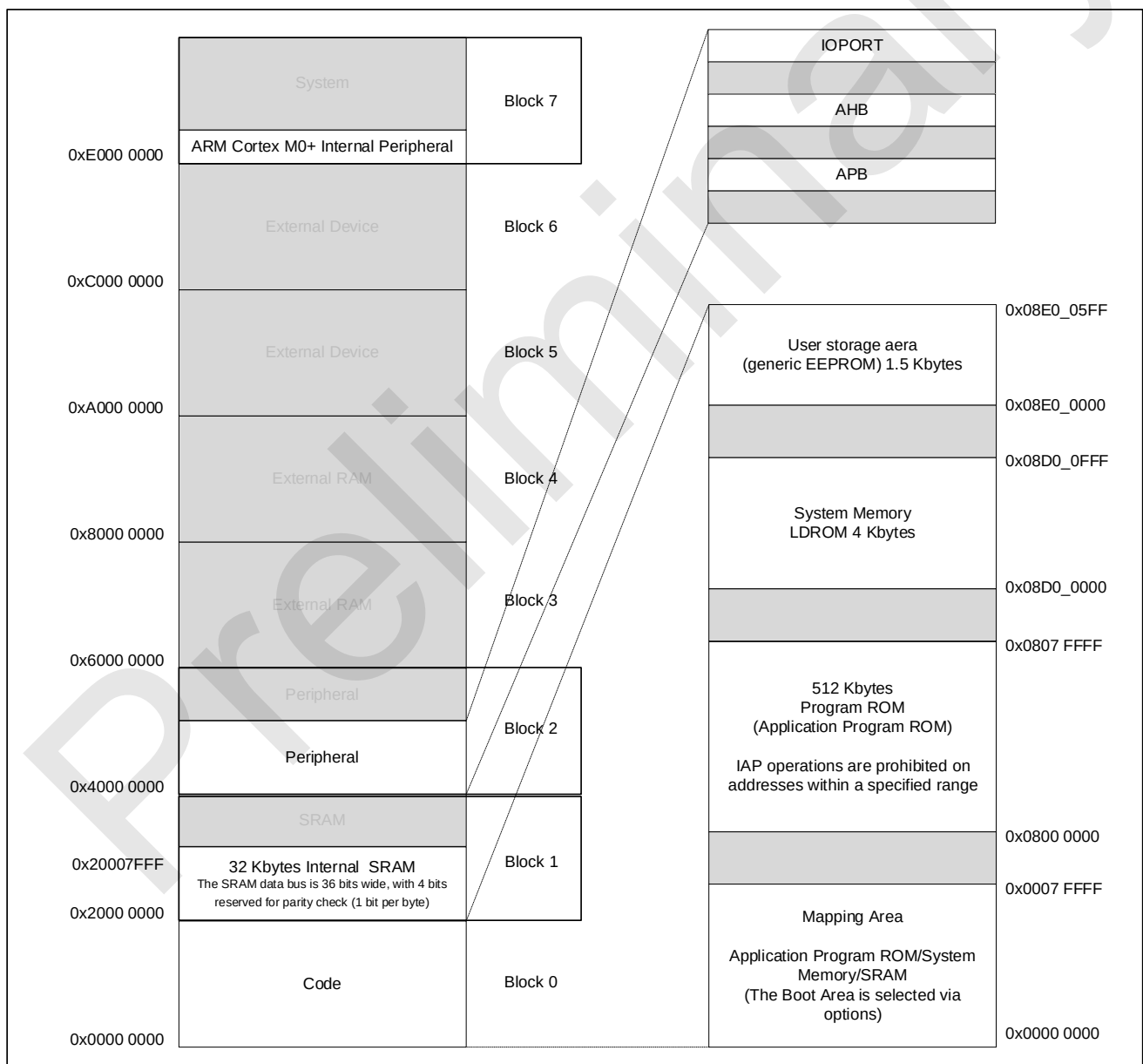


4 Flash

4.1 Overview

Program memory, data memory, and registers are arranged within the same linear (i.e., contiguous address) 4 GB address space. Bytes in memory are encoded using little-endian format. Within a word, the byte with the lowest number is considered the least significant byte, while the byte with the highest number is considered the most significant byte. The addressable Flash is divided into 8 main blocks, each block of 512 MB.

4.2 Storage block diagram



4.3 Feature

- The Flash has a 32-bit width and supports up to 100,000 write cycles.
- Data retention time at room temperature exceeds 10 years.
- The Flash memory is organized as follows:
 - Up to 512 Kbytes APROM (application program ROM)
 - 4 Kbytes LDROM (system storage area)
 - 1.5 Kbytes user storage area (generic EEPROM)
 - 32 Kbytes Internal SRAM with Parity Check Function
 - 96 bits Unique ID.

4.4 APROM (application program ROM)

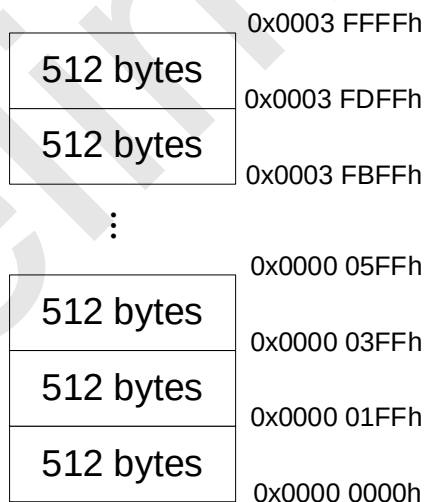
- The APROM of the SC32F11xx9 series has 512 Kbytes.
- The APROM of the SC32F11xx8 Series has 256 Kbytes.
- The APROM of the SC32F11xx7 Series has 128 Kbytes.
- Sector size: 512 bytes.
- Support: Read/Write, Sector Erase, Chip Erase, and Blank Check.
- The CPU (Cortex®-M0+) accesses Flash via the AHB Bus.
- By default, the program starts from the Application Program ROM. Users can select program startup from SRAM, LDROM, or other regions through the Customer Option OP_BL[1:0].
- Read Protection (read encryption): when enabled, only programs launched from the Application Program ROM can read information from it. Programs started from other regions or third-party tools cannot access this information.
- Write Protection: offers two hardware write protection Segments. IAP operation is disabled within the write protected Segments. Users can configure the size of these write protection Segments by sector based on actual requirements.

The 512 Kbytes Flash ROM is divided into 1024 sectors of 512 bytes each. During programming, the sector containing the target address is automatically erased before data is written. For user write operations, sectors must be erased prior to writing data.



SC32F11xx9 series 512 Kbytes Flash ROM sector partition diagram

The 256 Kbytes APROM is divided into 512 sectors of 512 bytes each. During programming, the sector containing the target address is automatically erased before data is written. For user write operations, sectors must be erased prior to writing data.



SC32F11xx8 series 256 Kbytes APROM sector partition diagram

The 128 Kbytes APROM is divided into 256 sectors of 512 bytes each. During programming, all sectors containing the target address are forcibly erased by the programmer before data is written. For user write operations, sectors must be erased before writing.

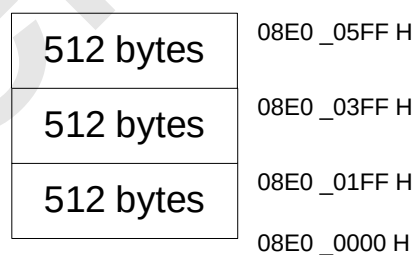


SC32F11xx7 series 128 Kbytes APROM sector partition diagram

4.5 1.5 Kbytes user storage area (generic EEPROM)

The 1.5 Kbytes independent EEPROM area is located at addresses 0x08E0_0000 H to 0x08E0_05FF H, configured by the IAPADE Register. The independent EEPROM can be rewritten 100,000 times, and data retention at room temperature exceeds 100 years. The independent EEPROM supports Blank Check, Programming, Verification, Erase, and Read functions.

EEPROM contains 3 sectors in total, each sector is 512 bytes:



EEPROM sector partition diagram

Note: The EEPROM endurance erase/write cycles are 100,000 times. Users should not exceed the EEPROM's rated programming cycles during erase/write operations, as this may cause malfunctions!

4.6 4 Kbytes LDROM (system storage area)

- The system storage area is a 4 Kbyte LDROM, factory programmed with the ISP Program. This region is not writable or accessible by users.
- Users can select booting from the LDROM area via the Customer Option OP_BL[1:0].

- **Embedded Bootloader Program:** The factory-programmed ISP Program, which is publicly available and enables updating program Flash via UART. The program execution waits for an upgrade command. If no update command is received within 500 ms, it jumps to execute the Application Program ROM at address 0X0800 0000.

4.6.1 BootLoader

Supports both hardware and software BootLoad methods:

- **Software approach:** Directly divide BootLoad and APP segments via software. By modifying VTOR, BootLoad and APP interrupts can be easily shared, and each segment's size can be freely adjusted.
- **Hardware implementation:** 4 Kbytes fixed System Memory as a dedicated BootLoader area, which users cannot read or write:
 - The Loader ROM serves as a fixed BootLoader space. The program is programmed before the production stage, and users cannot read or write it.
 - **Embedded Bootloader Program:** The Embedded Bootloader Program is located in System Memory and programmed during the production stage. **Fixed ISP Program:** This instruction is public and can be used to reprogram Flash via UART using this program.

4.7 SRAM

- **Internal SRAM:** 32 Kbytes, address range 0x2000 0000 to 0x2000 7FFF.
- **Supports parity check:**
 - An additional 4 Kbytes RAM is used for parity check: the SRAM data bus width is 36 bits, of which 4 bits are for parity check (1 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 enabling SRAM parity check, it is recommended to perform software initialization of the entire SRAM at the start of the code to avoid parity check errors when reading uninitialized locations.

- Users can select program startup from SRAM by setting the Customer Option OP_BL[1:0].
- Accessible at the maximum system clock frequency by byte, half-word (16-bit), or full word (32-bit) with no wait states, enabling access by both the CPU and DMA.

4.8 Boot area selection (bootstrap)

After reset, the user can configure the required bootstrap mode.

After exiting standby mode, the startup mode configuration can be resampled. After this startup delay ends, the CPU will fetch the stack top value from address 0x0000 0000, then begin executing code from the bootstrap memory starting at 0x0000 0004.

There are three bootstrap area options: application program ROM, system storage area, and SRAM, described in detail below:

4.8.1 Bootstrap from application program ROM

The application program ROM has an alias in the bootstrap memory space (0x0000 0000), but it can also be accessed from its original memory space (0x0800 0000). In other words, the program can start accessing memory from address 0x0000 0000 or 0x0800 0000.

4.8.2 Bootstrap from system memory

- The system memory area (4 Kbytes independent LDROM) serves as a fixed BootLoader region containing a program pre-programmed at the factory, which cannot be read or written by the user.
- Embedded Bootloader Program: The Embedded Bootloader Program is located in System Memory and programmed during the production stage. The embedded ISP Program is public and can be used to reprogram Flash via various Communication Interfaces (UART, TWI, and SPI).

4.8.3 Bootstrapping from embedded SRAM

SRAM has an Alias in the Bootstrap Memory space (0x0000 0000), but it can also be accessed from its original Memory space (0x2000 0000).

4.8.4 Bootstrap mode configuration

Three bootstrap modes are selected via the BTLD[1:0] register bits in combination with the software reset RST bit. Both BTLD and RST are protected by IAP_KEY:

1. Set BTLD[1:0] = 0x00; after chip software reset, boot from the application program ROM (APROM).
2. Set BTLD[1:0]=0x01; after chip software reset, boot from the system storage area (LDROM).
3. Set BTLD[1:0]=0x10; after chip software reset, boot from embedded SRAM.

The initial chip power-on boot area selection is configured via the Customer Option OP_BL[1:0]:

1. Set OP_BL[1:0]=0x00 in the Customer Option; after chip reset, boot from the application program ROM (APROM).
2. Set OP_BL[1:0]=0x01 in the Customer Option; after chip reset, boot from the system storage area (LDROM).
3. Set OP_BL[1:0]=0x10 in the Customer Option; after chip reset, boot from embedded SRAM.

4.9 96 bits Unique ID

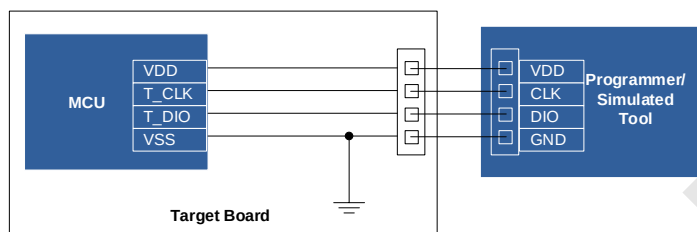
The SC32F11T/11G provides an independent Unique ID Area with a 96-bit unique code pre-programmed before shipment to ensure chip uniqueness. The only method for users to obtain the serial number is via IAP instructions.

4.10 User ID area

The User ID area is programmed at the factory with a user-customized ID. Users may read this ID but are disabled from writing to the User ID area.

4.11 Programming

The SC32F11T/11G Flash is programmed via T_DIO, T_CLK, VDD, and VSS, with the specific connection relationships as follows:



ICP Mode Flash Writer Programming Connection Diagram

T_DIO and T_CLK are two-wire JTAG signals for Flash Programming and Emulation. During programming, users can configure the modes of these two ports through the Customer Option: JTAG Specific Mode and Normal Mode (the JTAG Specific Port is disabled in Normal Mode).

4.11.1 JTAG specific mode

In JTAG Specific Mode, T_DIO and T_CLK are dedicated pins for programming and simulation, and all other functions multiplexed with them are disabled. This mode is typically used during online debugging to facilitate user simulation and debugging. Once JTAG Specific Mode is enabled, the chip can directly enter programming or simulation mode without needing to power cycle.

4.11.2 Normal mode (JTAG specific pins disabled)

In Normal Mode, JTAG functionality is disabled, and other functions multiplexed on the ports operate normally. This mode prevents the programming port from occupying MCU pins, enabling users to maximize MCU resource utilization.

Note: After successfully configuring to disable the JTAG specific pins, the chip must be completely powered off and then powered on again before entering programming or simulation mode. This affects programming and simulation in powered-on mode. SinOne recommends users select the configuration that disables the JTAG dedicated port during mass production programming, and choose JTAG mode during R&D debug.

The related Customer Options are listed below:

Register	Read/Write	Description	Reset Value
OPT_CON1	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 Symbol	Description
6	DISJTG	JTAG Port Switching Control Bit 0: Enable JTAG Mode; the related pins can only function as T_CLK/ T_DIO 1: Normal Mode, JTAG function disabled
4 ~ 2	-	Reserved

4.12 Security encryption

The security encryption function of the SC32F11T/11G Series primarily provides read protection encryption for APROM: during programming, users configure the read protection encryption function in the Customer Option setting of the SinOne dedicated programming suite's host software, select whether to enable flash read protection, and enter encryption mode:

- By default, the chip's flash is non-encrypted at factory shipment.
- The read protection encryption function does not have a mapping register; users can only configure it in the Customer Option setting of the SinOne dedicated programming suite's host software during programming, and modifications require programming.
- Encryption disabled: read, programming, and erase operations can be performed on the Application Program ROM. All operations on option bytes and backup registers are also allowed.
- Encryption Enable:
 - Application Program ROM Bootstrap: Code running in user mode (bootstrapped from user APROM) can perform all operations on the Application Program ROM.
 - 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 of the Application Program ROM first.

4.12.1 Security encryption access rights

Boot Area/Tools	Unlock State					Read Protection Encryption Status				
	Read	Write	Block Erase	Full Erase	Operate on Write Protection Area	Read	Write	Block Erase	Full Erase	Operate on Write Protection Area
Bootstrap from APROM	√	√	√	\	Disable	√	√	√	\	Disable
Debug / Bootstrap from SRAM	√	√	√	√	Disable	Disable	Disable	Disable	Disable	Disable
Bootstrap from LDROM	√	√	√	√	√	Disable	Disable	Disable	√	Disable

4.13 In application programming (IAP)

The IAP area in the APROM of SC32F11T/11G supports In Application Programming (IAP) operations, allowing users to perform remote program updates or read information from the Unique ID Area or User ID Area via IAP. Before performing IAP write operations, users must erase the sector corresponding to the target address.

By default, the chip's APROM permits global IAP operations out of the factory. The chip internally provides

two flash write protection regions, configured in sector units starting at specified addresses; the protected regions prohibit IAP operations. The configuration rules are as follows:

IAPPORx Register value (x = A or B)	IAPPOR Protection Area
IAPPORx_ST = IAPPORx_ED	Sector IAPPORx
IAPPORx_ST > IAPPORx_ED	None (Unprotected)
IAPPORx_ST < IAPPORx_ED	Sectors from IAPPORx_ST to IAPPORx_ED

Users can configure these two APROM write protection segments via the 'Flash sectors protection' option in the Customer Option during programming.

Note: IAP does not support single- or double-byte programming; IAP writes must be word-aligned (4-byte aligned).

4.13.1 IAP operation related registers

IAP operations on APROM outside the write protection area can be performed through the following registers:

4.13.1.1 Data protection register IAP_KEY

Register	Read/Write	Description	Reset Value
IAP_KEY	R/W	Data Protection Register	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 Symbol	Description
31 ~ 0	IAPKEY[31:0]	Data Protection Lock To prevent accidental Flash operations caused by electrical interference or other factors, write operations to the IAP_CON Register must first be unlocked using the IAPKEY. The unlock sequence is as follows: 1. Write KEY1 = 0x1234_5678 2. Write KEY2 = 0xA05F_05FA If the sequence is incorrect, the IAP_CON Register is locked and remains so until the next System Reset.

4.13.1.2 IAP sector number setting register IAP_SNB

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

31	30	29	28	27	26	25	24
IAPADE[7:0]							
23	22	21	20	19	18	17	16

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	IAPSNB[8]
7	6	5	4	3	2	1	0
IAPSNB[7:0]							

Bit Number	Bit Symbol	Description
31 ~ 24	IAPADE[7:0]	IAP Operation Area Extended Address By writing different values to IAPADE, the IAP operation targets different operation areas: 0x00: Invalid 0x4C: APROM 0x69: EEPROM 0xF1: Customer Option Others: Reserved
8 ~ 0	IAPSNB[8:0]	IAP operation sector number setting during sector/page erase: Starting address of the actual operating sector = Flash base address + [IAPSNB[8:0] x 0x200]
23 ~ 9	-	Reserved

4.13.1.3 IAP control register IAP_CON (write protection)

*This register is write-protected and can only be modified by operating the Data Protection Register IAP_KEY.

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

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

Bit Number	Bit Symbol	Description
31	LOCK	After writing 1 to this bit, the IAP_CON Register is locked. When the unlock sequence is detected, hardware clears this bit to 0. If the unlock operation fails, this bit remains set to 1 until the next System Reset.
10 ~ 9	BTLD[1:0]	Application Start Area Selection Bit after Software Reset: 00: Chip boots from APROM (Application Program ROM) after Software Reset 01: Chip boots from LDROM (system storage area) after Software Reset 10: Chip boots from Embedded SRAM after Software Reset 11: Reserved
8	RST	Software Reset Control Bit 0: Normal Program Operation 1: Writing 1 to this bit triggers an immediate system reset
7	ERASE	Chip Erase / Full Erase Control Bit 0: No Operation 1: Writing 1 to this bit and then setting CMD[1:0] = 10 initiates APROM Chip Erase / Full Erase, erasing the entire APROM

Bit Number	Bit Symbol	Description
5	SERASE	Sector Erase Control Bit 0: No Operation 1: Writing 1 to this bit and then setting CMD[1:0] = 10 initiates APROM Sector Erase, erasing the specified APROM Sector
4	PRG	Programming Control Bit 0: Disable Flash Programming 1: Enable Flash Programming
1 ~ 0	CMD[1:0]	IAP command enable control bits 10: Execute Erase Operation Command Others: Reserved Note: 1. After setting the Erase operation command bit to 1, CMD[1:0] must be configured to 10 for the operation to start 2. Only one IAP operation can be executed at a time, so only one of the ERASE or SERASE bits can be set to 1 simultaneously
30 ~ 11 6 3 ~ 2	-	Reserved

4.13.1.4 IAP register mapping

Register	Offset Address	Read/Write	Description	Reset Value
IAP base address: 0x4000_03C0				
IAP_KEY	0x00	R/W	Data Protection Register	0x0000_0000
IAP_SNB	0x04	R/W	IAP Sector Number Setting Register	0x0000_0000
IAP_CON	0x0C	R/W	IAP Control Register	0x0000_0000

4.14 Option bytes area (Customer Option)

The SC32F11T/11G features a dedicated Flash area for storing customers' power-on initial settings, called the Customer Option area. During programming, users can configure Customer Option items through the Host Software. The configuration values are written into the Customer Option area, which the IC uses as initial settings during reset initialization.

Customer Option items can also be temporarily modified by accessing the Mapping Register for Customer Option. However, such modifications are temporary and do not affect the values stored in the Customer Option area. After reset, the chip restores initialization settings based on the Customer Option parameters programmed previously.

The operation method of Customer Option related Mapping Registers is as follows:

The read and write operations of Customer Option related SFRs are controlled by two registers, OPINX and OPREG. The specific location of each Customer Option SFR is determined by OPINX, as shown in the table below:

Register	Address	Description	Reset Value
OPINX	0x4000_03F8	Customer Option Pointer	0x0000_0000
OPREG	0x4000_03FC	Customer Option Register	0x0000_0000
OPT_CON0	0XC1 @ OPINX	Customer Option Mapping Register 0	0x0000_0000
OPT_CON1	0XC2 @ OPINX	Customer Option Mapping Register 1	0x0000_0000

4.14.1 Mapping register for Customer Option

Before rewriting the IFB mapping register using OPINX with OPREG, ensure the clock enable switch AHB_CFG.IFBEN for the Customer Option Register is first turned on:

4.14.1.1 AHB bus peripheral clock enable register (AHB_CFG)

Register	Read/Write	Description	Reset Value
AHB_CFG	R/W	AHB Bus Peripheral Clock Enable Register	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 Symbol	Description
2	IFBEN	Customer Option Mapping Register Clock Enable Bit Before rewriting the IFB Mapping Register using OPINX with OPREG, the Clock Enable must be enabled first. 0: Disable 1: Enable
31 ~ 23 19 ~ 5 3	-	Reserved

4.14.1.2 Customer Option mapping register 0 OPT_CON0

Register	Read/Write	Description	Reset Value
OPT_CON0	R/W	Customer Option Mapping Register 0	0x0000_0000

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

Bit Number	Bit Symbol	Description
2	DISLVR	LVR Switch 0: LVR Enabled 1: LVR Disabled
1 ~ 0	LVRS[1:0]	LVR Voltage Selection Control 11: 4.3 V Reset

Bit Number	Bit Symbol	Description
		10: 3.7 V Reset 01: 2.9 V Reset 00: 1.9 V Reset
7 ~ 3	-	Reserved

4.14.1.3 Customer Option mapping register 1 OPT_CON1

Register	Read/Write	Description	Reset Value
OPT_CON1	R/W	Customer Option Mapping Register 1	0x0000_0000

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

Bit Number	Bit Symbol	Description
7	ENWDT	WDT Switch 1: WDT Enabled 0: WDT Disabled
6	DISJTG	JTAG Port Switching Control Bit 0: Enable JTAG Mode; the related pins can only function as T_CLK/ T_DIO 1: Normal Mode, JTAG function disabled
5	DISRST	Reset Signal Port Switch Control Bit This bit is read-only and cannot be modified by the user. 0: The RST pin is used as a reset pin. 1: The RST pin is used as a normal GPIO pin.
4	HIRC_PLL	PLL Clock Source Switch Control Bit 0: PLL 1: Select HIRC/1 to replace PLL, when used set RCC_CFG0.HIRC_NDIV to 1
3	DRV_EH	IOL Drive Capability Selection for the Port with COM: 0: For ports corresponding to COM0 through COM7, IOL has strong drive capability only when the port is enabled for COM function; otherwise, the port maintains normal drive capability. For applications without COM driving, it is recommended to use the normal drive capability for these ports. 1: Ports COM0 to COM7 have a Drive Capability with the sinking current IOL always strong.
1 ~ 0	OP_BL[1:0]	Boot area selection after chip Reset This bit is read-only and cannot be modified by the user. 00: Boot from APROM (Application Program ROM) after chip Reset 01: Boot from LDROM (system storage area) after chip Reset 10: Boot from Embedded SRAM after chip Reset 11: Reserved
2	-	Reserved

5 Power, reset and system clock (RCC)

5.1 Power-on process

After power-on of the SC32F11T/11G, the client software execution will proceed through the following three stages:

1. Reset Stage
2. Loading Information Stage
3. Normal Operation Stage.

5.1.1 Reset stage

The Reset Stage refers to the SC32F11T/11G being continuously held in the reset state until the power supply voltage exceeds a specific threshold, at which point its internal clock mechanism begins to operate effectively.

The exact duration of this stage depends on the rise rate of the external power supply voltage. When the external power supply voltage reaches the built-in POR voltage, the Reset Stage ends.

5.1.2 Loading information stage

An internal warm-up counter is provided inside the SC32F11T/11G. During the reset stage, this warm-up counter is kept at zero. After the supply voltage rises to POR, the internal HIRC oscillator warm-up counter starts counting. When the warm-up counter reaches a certain value, the IFB data (including Customer Option) in Flash ROM will be periodically read to the system register. After all IFB data is read, the warm-up and information loading stages end, and the system enters normal operating mode.

5.1.3 Normal operation stage

After the Loading Information Stage ends, the SC32F11T/11G begins fetching instruction codes from Flash and enters the Normal Operation Stage. At this point, the LVR voltage is the setting value configured by the user in the Customer Option.

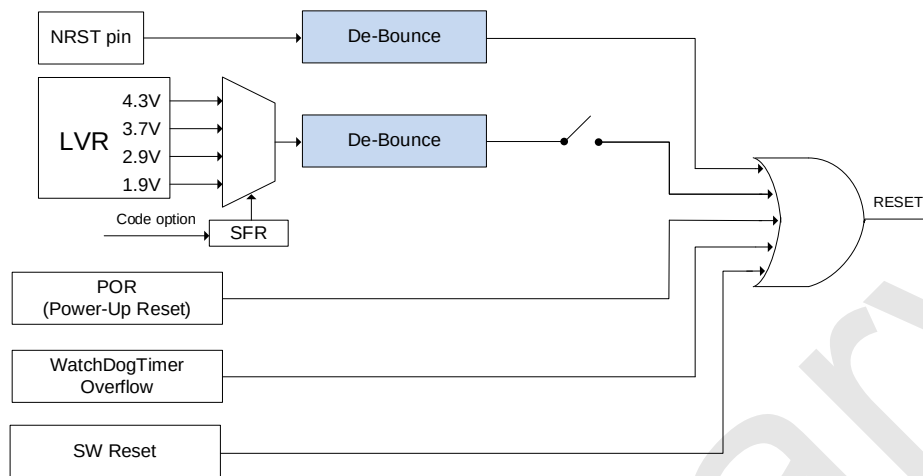
5.2 Reset

The SC32F11T/11G supports two types of Reset: Hardware Reset and Software Reset. When a Reset occurs, all registers except the CAN registers reset to their Reset Value. Refer to the [CAN Register](#) section for details on the CAN registers' Reset Values.

The SC32F11T/11G has five Reset types, the first four of which are Hardware Reset:

1. External RST Reset
2. Low Voltage Reset (LVR)
3. Power-On Reset (POR)
4. Watchdog (WDT) Reset
5. Software Reset

The circuit diagram of the reset section of the SC32F11T/11G is shown below:



SC32F11T/11G Reset Circuit Diagram

5.2.1 Hardware reset

Hardware Reset can be performed by the following four methods:

1. External RST Reset
2. Low Voltage Reset (LVR)
3. Power-On Reset (POR)
4. Watchdog (WDT) Reset

5.2.1.1 External RST reset

The external RST Reset of SC32F11T/11G is triggered by applying a low-level reset pulse longer than 18 μ s to the external NRST pin.

Before programming, users can configure the PD3 / NRST pin as NRST (Reset pin) via the Customer Option in the Host Software.

- When this pin is configured as a reset function, it cannot serve as GPIO.
- When this pin is configured as GPIO, its reset function is disabled.

5.2.1.2 Low voltage reset (LVR)

SC32F11T/11G includes a built-in Low Voltage Reset circuit. There are four options for the reset threshold voltage: 4.3 V, 3.7 V, 2.9 V, and 1.9 V. The factory default is set to 1.9 V, but users can adjust the default by configuring the Customer Option value during flashing. When V_{DD} voltage falls below the set threshold and remains low for longer than the debounce time T_{LVR} of approximately 30 μ s, the chip resets.

5.2.1.3 Power-on reset (POR)

The SC32F11T/11G includes an internal Power-On Reset (POR) circuit that resets the system when V_{DD} reaches the POR reset voltage.

5.2.1.4 Watchdog reset WDT

The SC32F11T/11G features a WDT with its Clock Source as the internal 37 kHz Oscillator. Users can enable or disable the Watchdog Reset function via the programmer's Customer Option. A Watchdog Reset will occur if a watchdog event takes place.

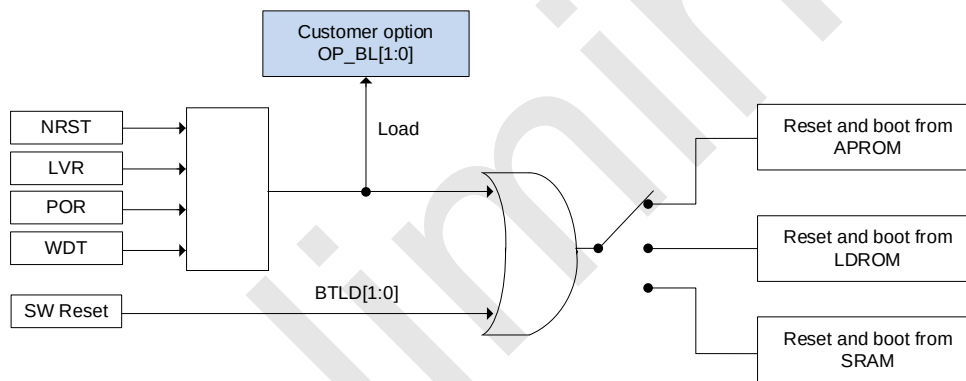
5.2.2 Software reset

The SC32F11T/11G supports software reset. Writing 1 to the RST (IAP_CON.8) bit immediately resets the system.

5.2.3 Boot area after reset

After the four hardware reset types-External RST Reset, Low Voltage Reset (LVR), Power-On Reset (POR), and Watchdog (WDT)-the chip boots from the boot area (APROM / LDROM / SRAM) specified by the user OP_BL[1:0].

After a software reset, the chip boots from the boot area (APROM / LDROM / SRAM) specified by BTLD[1:0].



SC32F11T/11G Reset Boot Area Switching Diagram

5.2.4 Initial reset state

When the SC32F11T/11G is in reset state, most registers return to their initial values. The Watchdog WDT function is disabled during this period. Resets during 'Hot-start' (such as WDT, LVR, software reset, etc.) do not affect SRAM data; thus, SRAM retains the pre-reset values.

Note: If the power supply voltage falls below the level required to maintain RAM data retention, the contents of SRAM will be lost.

5.3 Clock

5.3.1 System clock source

System Clock SYSCLK can be driven by five different clock sources:

- Built-in high-frequency 64 MHz Oscillator (HIRC)
- External high-frequency Crystal, supporting external connection to a 2 to 16 MHz high-frequency

oscillator (HXT)

- Built-in low-frequency 37 kHz Oscillator (LIRC)
- External low-frequency Crystal Oscillator, can connect to a 32.768 kHz low-frequency oscillator (LXT)
- PLL, with a maximum frequency of 72 MHz.

Each oscillator can be individually enabled or disabled to reduce power consumption when not in use.

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 via software during normal operation after power-up. Before switching, ensure the selected clock source is in a stable operating state.**
2. **When switching the System Clock Source, you must first switch to HIRC before switching to the target Clock Source.**

5.3.2 Built-in high-frequency 64 MHz oscillator (HIRC)

The built-in high-frequency clock HIRC can be enabled by setting the HIRCEN bit of Register RCC_CFG0 to 1, or disabled by clearing it to 0. The user must first write a value exceeding 0x40 to RCC Protection Register RCC_KEY before modifying RCC_CFG0 and RCC_CFG1.

HIRC has the following Features:

- Can serve as the System Operating Clock.
- The default clock frequency at system power-on, f_{SYS} , is f_{HIRC} divided by 2
- Frequency Error: Frequency error within the application environment of 2.0 V to 5.5 V and -40 to 105 °C
- The system clock can be automatically calibrated by the 32.768 kHz external Crystal Oscillator, and after calibration, HIRC accuracy can be infinitely close to that of the external 32.768 kHz crystal oscillator.

5.3.3 Built-in high-frequency crystal oscillator circuit, can be externally connected to a 2 to 16 MHz high-frequency oscillator (HXT)

The external high-frequency crystal oscillator clock (HXT) can be enabled by setting the HXTEN bit of Register RCC_CFG0 to 1, or disabled by clearing it to 0; The user must first write a value exceeding 0x40 to RCC Protection Register RCC_KEY before modifying RCC_CFG0 and RCC_CFG1.

Additionally, users must set an appropriate CRY_HF value according to the frequency of the externally connected high-frequency crystal oscillator. When the frequency is below 12 MHz, CRY_HF should be set to 0; When the frequency is above 12 MHz, CRY_HF should be set to 1.

HXT provides the following functions and features:

- Can serve as the System Operating Clock.
- Supports external connection of a 2 to 16 MHz high-frequency oscillator.

5.3.4 Built-in low-frequency 37 kHz oscillator (LIRC) .

The built-in low-frequency clock LIRC can be enabled by setting LIRCEN to 1 in Register RCC_CFG0, or disabled by clearing it to 0; The user must first write a value exceeding 0x40 to RCC Protection Register RCC_KEY before modifying RCC_CFG0 and RCC_CFG1.

LIRC has the following functions and features:

- Can serve as the System Operating Clock.
- Can serve as the clock source for LCD/LED modules
- Can serve as the clock source for the Base Timer and WDT; when WDT is enabled, this clock source must be enabled
- Frequency error: within the voltage range of 2.0 to 5.5 V and at a normal temperature of 25 °C, the frequency error after register correction is $\leq \pm 4\%$.

5.3.5 Built-in low frequency oscillator circuit, with support for external 32.768 kHz low frequency oscillator (LXT)

When using an external low-frequency Crystal clock LXT, enable it by setting the LXTEN bit of Register RCC_CFG0 to 1, or disable it by clearing to 0; The user must first write a value exceeding 0x40 to RCC Protection Register RCC_KEY before modifying RCC_CFG0 and RCC_CFG1.

LXT has the following functions and features:

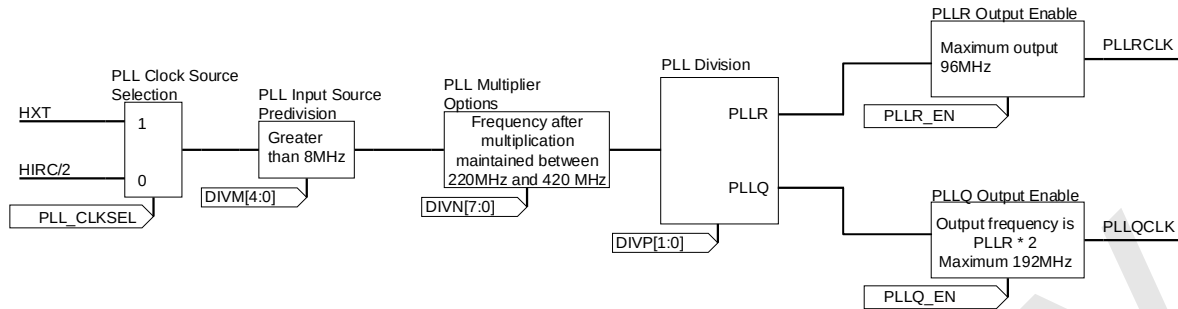
- Can serve as the System Operating Clock.
- Can be selected as the Base Timer clock source
- Can serve as the clock source for LCD/LED modules
- Supports an external 32.768 kHz low-frequency oscillator.
- Automatic calibration of HIRC can be performed using LXT.

5.3.6 PLL

PLL output is divided into PLLRCLK and PLLQCLK.

PLL has the following functions and features:

- PLLRCLK
 - Can be selected as Application Output, with output frequency up to 96 MHz.
 - Can be selected as the System Clock Source, with output frequency up to 72 MHz.
- PLLQCLK
 - Output frequency is twice that of PLLRCLK, up to 192 MHz.
 - Can be selected as the PWM0 clock source.
- Users can configure the required frequency using the following formula:
 - PLL input Clock frequency f_{PLL_IN}
 - PLL output frequency $f_{PLL} = [(f_{PLL_IN}$
- The block diagram of the PLL circuit is as follows:



5.4 Register access clock and peripheral clock source

5.4.1 Register access clock

The Register Access Clock drives the Cortex®-M0+ Core and internal bus operations. It directly controls instruction pipeline execution, register file access, and data transfer between the core and bus.

The register access clock remains strictly synchronous with the AHB (Advanced High-performance Bus) and APB (Advanced Peripheral Bus) bus clocks, ensuring coordinated operation of the CPU core and peripheral interfaces within a unified timing framework. Users must enable bus clocks and control peripheral clock gating via the AHB Bus Peripheral Clock Enable Register (AHB_CFG) or the APB Bus Peripheral Clock Enable Registers (APBn_CFG, n=0 to 2). Read and write operations to a peripheral register can only be performed when both the bus and the register access clock bits for the corresponding peripheral are enabled; otherwise, a hardware protection mechanism will be triggered, invalidating the access.

5.4.2 Peripheral clock source

The Peripheral Clock Source is a clock signal source that provides independent operating clock ticks for specific peripheral modules (such as PWM, UART, ADC, etc.). It can be a divided or multiplied version of the System Clock, or an independent built-in or external Clock Source (such as an internal RC Oscillator or an external Crystal Oscillator). The Peripheral Clock Source allows different peripherals to select the optimal Clock Frequency and stability based on their own needs. Users can also reduce Power Consumption by disabling clocks for unused peripherals individually.

Some peripherals of the SC32F11T/11G provide multiple optional Clock Sources. Users can select the appropriate Clock Source for peripherals through the Peripheral Clock Source selection registers (RCC_CFGn, n=0 to 1).

5.4.3 Bus clock

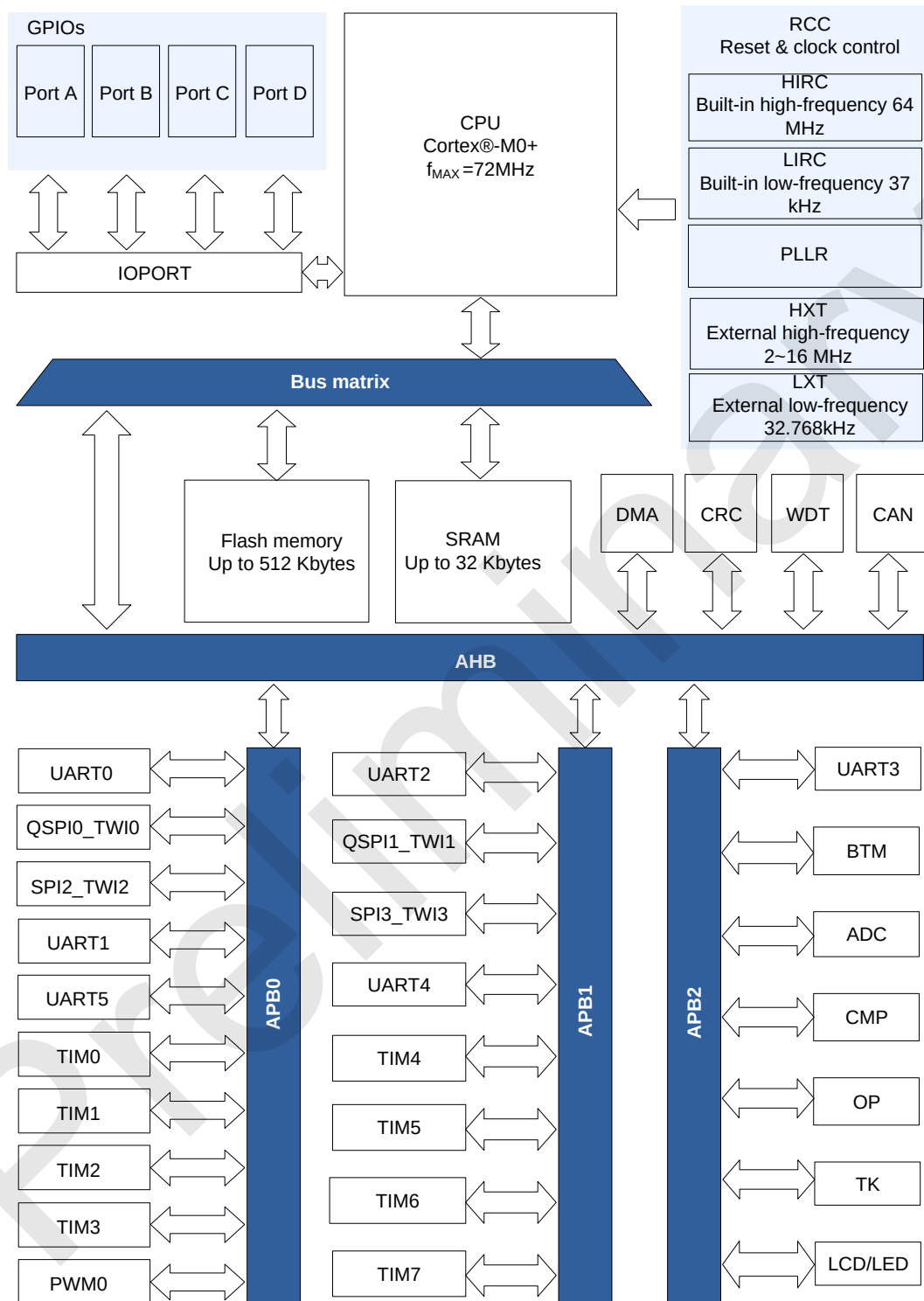
Users may configure the frequencies of the IOPORT, AHB, APB0, APB1, and APB2 domains using multiple prescalers.

- IOPORT: Main clock for the GPIO port domain, derived from System Clock SYSCLK, with a maximum frequency of 72 MHz. All GPIOs are driven by IOPORT.
- HCLK: Main clock for the AHB domain, derived from System Clock SYSCLK. Default division is SYSCLK

divided by 2, adjustable via CLKDIV[2:0] in the AHB CFG register. Maximum frequency is 72 MHz. Drives the Cortex®-M0+ Core, memory, DMA, and other peripherals.

- PCLK0: Main clock for the APB0 domain, derived by dividing HCLK, with a maximum frequency equal to HCLK. All peripherals on the APB0 Bus are driven by PCLK0, which also provides some peripheral clock sources on the bus.
- PCLK1: The main clock for the APB1 domain, derived by dividing HCLK. Its maximum frequency equals that of HCLK. All peripherals on the APB1 bus are driven by PCLK1, and some peripheral clock sources on the bus come from PCLK1.
- PCLK2: The main clock of the APB2 domain, derived from HCLK through division. Its maximum frequency equals the HCLK frequency. Peripherals on the APB2 bus are driven by PCLK2, and some peripheral clock sources on this bus are provided by PCLK2.

The bus architecture diagram is shown below. The frequency for reading peripheral registers depends on the frequency of the bus to which the peripheral is connected.



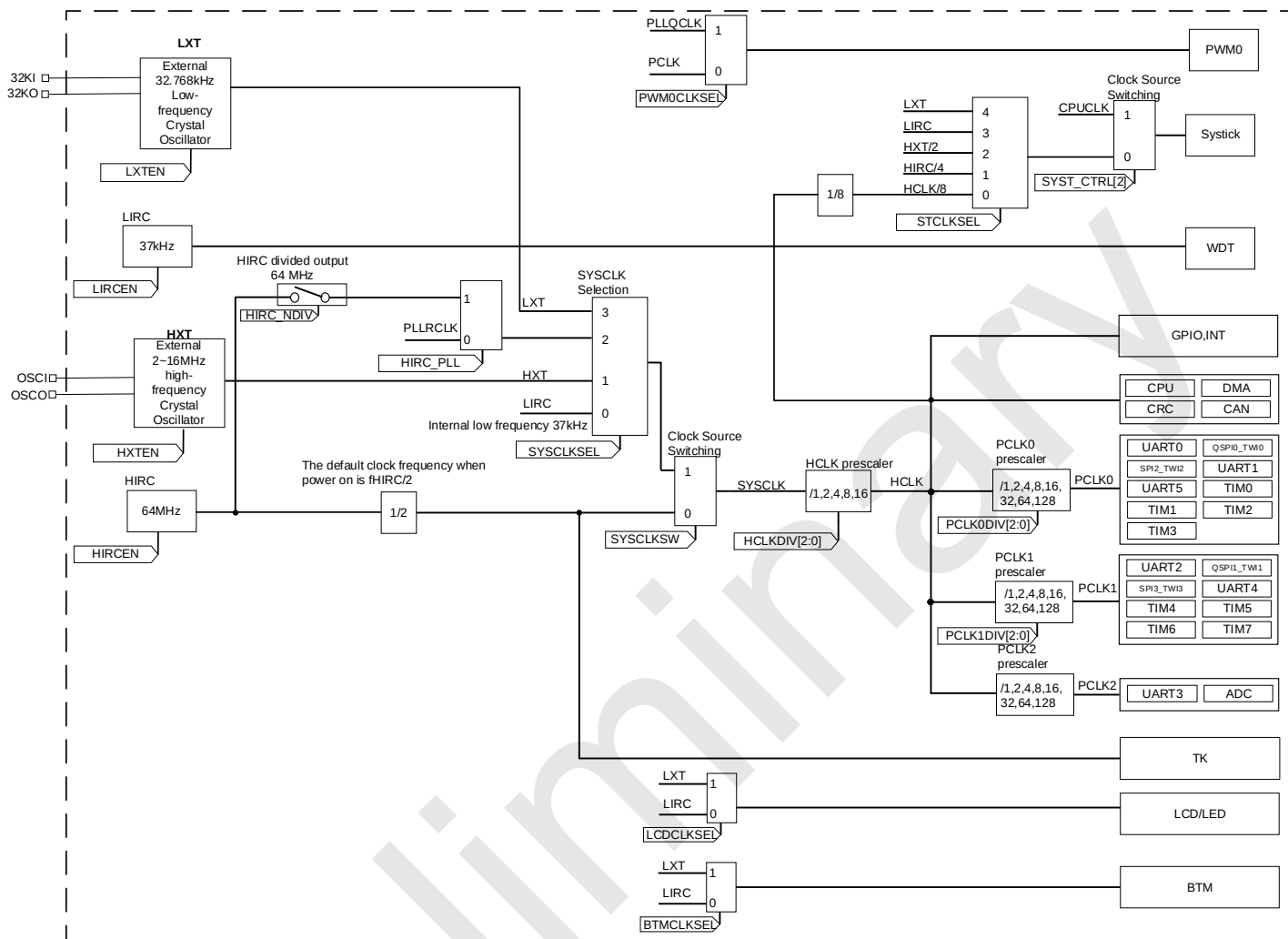
Bus Architecture Block Diagram

5.4.4 Peripheral clock

Peripheral Clocks on the Bus are generally provided by their corresponding Bus Clocks, but the following peripherals can select other Clock Sources:

- PWM0 can select the following Clock Sources:
 - PCLK, the Clock on the Bus that PWM0 resides on
 - PLLQCLK
- LCD/LED can select the following Clock Sources:
 - LIRC, the internal low-frequency Clock
 - LXT, the Low-frequency Crystal Clock
- BTM can select the following Clock Sources:
 - LIRC, the internal low-frequency Clock
 - LXT, the Low-frequency Crystal Clock
- The WDT Clock Source is fixed to LIRC; when WDT is enabled, LIRC automatically starts, and it remains oscillating during WDT operation, which cannot be turned off by the user.
- SysTick can select the following Clock Sources:
 - LXT, the Low-frequency Crystal Clock
 - LIRC, the internal low-frequency Clock
 - HXT/2, the High-frequency Crystal Clock divided by 2
 - HIRC/4, the clock divided from the internal high-frequency clock by 4
 - HCLK/8, the clock divided from the AHB Bus clock by 8
 - CPUCLK, the Core clock

5.4.5 Peripheral clock block diagram



Note: The default system clock frequency at power-on, " f_{SYS} ," is $f_{HIRC}/2$. Users can select the clock source by modifying `SYSCCLKSW` or `SYSCCLKSEL`.

5.4.6 Peripheral clock enable register

Each peripheral clock can be enabled via the corresponding enable bits in the `AHB_CFG` and `APBn_CFG` registers. When the peripheral clock is inactive, read and write access to the peripheral registers is disabled.

5.4.7 Low-power mode clock

In low power STOP mode, the CPU clock, most clock sources, and peripheral clocks stop, and the program halts execution. However, the following clock sources continue to operate normally in low power STOP mode:

- Internal low frequency clock LIRC
- External low frequency oscillator clock LXT

5.5 RCC interrupt

Supporting the oscillation stop detection mechanism, the clock source of SC32F11T/11G provides a user-configurable RCC Interrupt: when the system clock source is HXT, if an abnormality in the clock source is

detected, the oscillation stop detection interrupt flag is set. If the corresponding interrupt is enabled, an oscillation stop detection interrupt is triggered.

5.6 RCC register

5.6.1 RCC register table

5.6.1.1 RCC protection register RCC_KEY

Register	Read/Write	Description	Reset Value
RCC_KEY	R/W	RCC Protection 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
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
RCCKEY[7:0]							

Bit Number	Bit Symbol	Description
7 ~ 0	RCCKEY[7:0]	Enable bits and timeout settings for registers RCC_CFG0 and RCC_CFG1. Writing a value $n \geq 0x40$ means: 1. Enables write access to registers RCC_CFG0 and RCC_CFG1 2. If no write command is received within n system clocks, the RCC rewrite function is disabled.
31 ~ 8	-	Reserved

5.6.1.2 System clock source selection register RCC_CFG0 (write-protected)

*This register is write-protected and can only be modified via the RCC Protection Register RCC_KEY.

Register	Read/Write	Description	Reset Value	Power-on Initial Value
RCC_CFG0	R/W	System Clock Source Selection Register	0x0000_0000	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	HXTEN	CRY_HF	-	-	LIRCEN	LXTEN

Bit Number	Bit Symbol	Description
15	INTEN	CPU Interrupt Request Enable Control Bit 0: Disable interrupt request 1: Enable interrupt request

Bit Number	Bit Symbol	Description
14	HIRC_NDIV	Enable bit for 64 MHz output of internal high-frequency crystal oscillator divided by 1 0: Disable 1: Enable
13 ~ 12	WAIT[1:0]	00: 0 wait states 01: 1 wait state (recommended for 32 MHz system clock) 10: 2 wait states (recommended for 64 MHz system clock) 11: 3 wait states (recommended for 64 MHz and 72 MHz system clocks)
11	HPLDO_DP	Low-frequency system clock power consumption adjustment bit 0: Recommended when system clock source is not LIRC 1: Recommended when system clock source is LIRC; writing 1 reduces overall power consumption when LIRC is selected
9 ~ 8	SYSCLKSEL[1:0]	System clock source selection bits 00: System clock source is LIRC 01: System clock source is HXT 10: System clock source is the HIRC_PLL selected clock 11: System clock source is 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 via software during normal operation after power-up. Before switching, ensure 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.
7	SYSCLKSW	System Clock Source Switch bit: when enabled, the system clock switches from HIRC to the clock selected by SYSCLKSEL. 0: System clock source is HIRC 1: System clock source is the SYSCLKSEL selection After writing this bit, the value only updates once the internal circuit completes the switch; otherwise, reads return the previous state. Users can read this bit to verify if the clock source switch has succeeded. This bit is automatically cleared after reset or wake-up, defaulting the system clock source to HIRC. 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 via software during normal operation after power-up. Before switching, ensure 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.
6	HIRCEN	Enable bit for the internal 64 MHz high-frequency oscillator (HIRC) 0: Disable 1: Enable This bit is not writable when SYSCLKSW = 0 and HIRC is selected as the system clock. This bit is set by hardware after reset or wake-up, indicating the system clock is supplied by HIRC.
5	HXTEN	External high-frequency crystal oscillator HXT enable 0: Disable 1: Enable Note: 1. After any reset event, the HXTEN bit is automatically cleared by hardware.

Bit Number	Bit Symbol	Description
		2. When waking up from STOP mode, the state of the HXTEN bit depends on the system clock source used before entering STOP mode: a) If HXT was used as the system clock before entering STOP, after wake-up HXTEN remains 1 (not cleared), but the system clock is switched to HIRC by hardware; b) If HXT was not used as the system clock before entering STOP, after wake-up HXTEN is cleared by hardware.
4	CRY_HF	External high-frequency crystal oscillator HXT frequency range selection 0: External crystal frequency < 12 MHz 1: External crystal frequency ≥ 12 MHz
1	LIRCEN	Internal low-frequency 37 kHz oscillator LIRC enable 0: Disable 1: Enable
0	LXTEN	External low-frequency crystal oscillator LXT enable 0: Disable 1: Enable
31 ~ 16 10 3 ~ 2	-	Reserved

5.6.1.3 Peripheral clock source selection register RCC_CFG1 (write-protected)

*This register is write-protected and can only be modified via the RCC Protection Register RCC_KEY.

Register	Read/Write	Description	Reset Value
RCC_CFG1	R/W	Peripheral clock source selection 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
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
STCLKSEL[2:0]			-	-	PWM0CLKSEL	LCDCLKSEL	BTMCLKSEL

Bit Number	Bit Symbol	Description
7 ~ 5	STCLKSEL[2:0]	SysTick clock source selection 000: Clock source from HCLK/8 001: Clock source from HIRC/4 010: Clock source from HXT/2 011: Clock source from LIRC 100: Clock source from LXT Note: If the SysTick clock source is not HCLK, its frequency must be less than or equal to $f_{HCLK}/2$.
2	PWM0CLKSEL	PWM0 8-channel 16-bit multifunction clock source selection bit 0: Clock source from PCLK 1: Clock source from PLLQCLK After writing this bit, wait for the internal switch to complete before the update takes effect; otherwise, reads will return the previous value. Users can read this bit to verify if the clock source switch has succeeded.

Bit Number	Bit Symbol	Description
1	LCDCLKSEL	LCD Clock Source Selection Bit 0: Clock source is LIRC 1: Clock source is LXT After writing this bit, wait for the internal switch to complete before the update takes effect; otherwise, reads will return the previous value. Users can read this bit to verify if the clock source switch has succeeded.
0	BTMCLKSEL	BTM Clock Source Selection Bit 0: Clock source is LIRC 1: Clock source is LXT After writing this bit, wait for the internal switch to complete before the update takes effect; otherwise, reads will return the previous value. Users can read this bit to verify if the clock source switch has succeeded.
31 ~ 8 4 ~ 3	-	Reserved

5.6.1.4 PLL configuration register PLL_CFG (write protection)

*This register is write-protected and can only be modified via the RCC Protection Register RCC_KEY.

Register	Read/Write	Description	Reset Value
PLL_CFG	R/W	PLL Configuration Register	0x0000_0000

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

Bit Number	Bit Symbol	Description
23	PLLCLKSEL	PLL Clock Source Selection Bit 1: PLL clock source is external high-speed crystal oscillator HXT; PLL input clock frequency $f_{PLL_IN} = f_{HXT}$ 0: PLL clock source is internal high-speed oscillator HIRC; PLL input clock frequency $f_{PLL_IN} = f_{HIRC}$ Note: This bit field can only be written when PLL is disabled.
20 ~ 16	DIVM[4:0]	PLL Clock Source Pre-divider Valid value range: 0x01 to 0x1 F Note: The user must configure this to ensure the PLL input frequency is divided to 8 MHz or higher.
15 ~ 8	DIVN[7:0]	PLL multiplication factor 2 ≤ N ≤ 192 0x00, 0x01, and values above 192: invalid Calculation formula: $f_{PLL} = [(f_{PLL_IN} / DIVM) * (DIVN + 2)] / 2 ^ (DIVP + 1)$ Note: 1. User must configure the multiplier to keep the frequency between 220 MHz and 420 MHz. Then, configure DIVP to ensure

Bit Number	Bit Symbol	Description
		PLL output does not exceed 96 MHz. 2. PLLR can be used as system clock source but must not exceed 72 MHz.
7	PLLON	PLL enable bit 0: PLL off 1: PLL enabled Cleared automatically by hardware when entering stop, standby, or shutdown modes. Note: This bit must not be 0 if PLL clock is selected as the system clock.
6	PLLREN	PLLCLK clock output enable bit 0: Disable 1: Enable This bit cannot be written when PLLCLK output is selected as system clock.
5	PLLQEN	PLLQCLK clock output enable bit 0: Disable 1: Enable
1 ~ 0	DIVP[1:0]	PLLCLK clock output division factor 00: Divide by 2 01: Divide by 4 10: Divide by 8 11: Divide by 16 Writable only when PLL is disabled. PLLCLK can be selected as the system clock.
31 ~ 24 22 ~ 21 4 ~ 2	-	Reserved

5.6.1.5 Clock status register RCC_STS

Register	Read/Write	Description	Reset Value
RCC_STS	R/W	Clock Status 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
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	SRAMPEIF	PLLRDY	LOCKERR	CLKFIF

Bit Number	Bit Symbol	Description
3	SRAMPEIF	SRAM Parity Error Flag Set by hardware when an SRAM parity error is detected. Cleared by writing 1 in software. 0: No SRAM parity error detected 1: SRAM parity error detected
2	PLLRDY	PLL Clock Ready Flag Read-only. Set by hardware to indicate PLL lock. 0: PLL not locked 1: PLL Locked
1	LOCKERR	PLL Loss of Lock Record Bit Read-only bit set by hardware to indicate a PLL loss of lock. 0: PLL has not lost lock

Bit Number	Bit Symbol	Description
		1: PLL loss of lock detected
0	CLKFIF	Clock source fault flag, valid when system clock source is an external crystal 0: No fault in current clock source 1: Fault detected in current clock source; system clock source automatically switched to HIRC. If RCC Interrupt Enable is set (RCC_CFG0.INTEN=1), this generates an interrupt. The CLKFIF flag can be cleared after reset.
31 ~ 4	-	Reserved

5.6.1.6 Systick calibration parameter register (SYST_CALIB)

Register	Read/Write	Description	Reset Value	Power-on Initial Value
SYST_CALIB	R	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 Symbol	Description
23 ~ 0	CALIB[23:0]	Default calibration register value: If the power-on default clock frequency is f_{HCLK}/n (MHz), where n is the default power-on prescaler, the power-on default clock source is HIRC. The initial SysTick calibration value is $1000 * (f_{HCLK}/n)$, ensuring a default 1 ms time base
31 ~ 24	-	Reserved

5.6.1.7 AHB bus peripheral clock enable register (AHB_CFG)

Register	Read/Write	Description	Reset Value
AHB_CFG	R/W	AHB Bus Peripheral Clock Enable Register	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 Symbol	Description
22 ~ 20	CLKDIV[2:0]	AHB Clock Division Bits AHB Bus Clock HCLK is derived from the System Clock SYSCLK by division: 000: $f_{HCLK} = f_{SYSCLK}$ 001: $f_{HCLK} = f_{SYSCLK} / 2$ 010: $f_{HCLK} = f_{SYSCLK} / 4$ 011: $f_{HCLK} = f_{SYSCLK} / 8$ 100: $f_{HCLK} = f_{SYSCLK} / 16$ Others: Reserved
4	CANEN	CAN Module Clock Enable Bit 0: Disable 1: Enable
2	IFBEN	Customer Option Mapping Register Clock Enable Bit Before rewriting the IFB Mapping Register using OPINX with OPREG, the Clock Enable must be enabled first. 0: Disable 1: Enable
1	CRCEN	CRC Module Clock Enable Bit 0: Disable 1: Enable
0	DMAEN	DMA Clock Enable Bit 0: Disable 1: Enable
31 ~ 23 19 ~ 5 3	-	Reserved

5.6.1.8 APB0 peripheral clock enable register APB0_CFG

Register	Read/Write	Description	Reset Value
APB0_CFG	R/W	APB0 Peripheral Clock Enable Register	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
-	-	-	-	-	-	UART5EN	PWM0EN
7	6	5	4	3	2	1	0
UART1EN	UART0EN	SPI2_TWI2EN	QTIW0EN	TIM3EN	TIM2EN	TIM1EN	TIM0EN

Bit Number	Bit Symbol	Description
23	ENAPB	APB0 Bus Clock Gate Control Bit 0: Disable 1: Enable
22 ~ 20	CLKDIV[2:0]	APB0 Clock Division Bits APB0 Bus Clock PCLK0 is derived from HCLK by division: 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: $f_{PCLK0} = f_{HCLK} / 64$

Bit Number	Bit Symbol	Description
		111: $f_{PCLK0} = f_{HCLK} / 128$
9	UART5EN	UART5 Clock Enable Bit 0: Disable 1: Enable
8	PWM0EN	PWM0 Clock Enable Bit 0: Disable 1: Enable
7	UART1EN	UART1 Clock Enable Bit 0: Disable 1: Enable
6	UART0EN	UART0 Clock Enable Bit 0: Disable 1: Enable
5	SPI2_TWI2EN	SPI2_TWI2 Clock Enable Bit 0: Disable 1: Enable
4	QTIW0EN	QSPI0_TWI0 Clock Enable Bit 0: Disable 1: Enable
3	TIM3EN	Timer3 Clock Enable Bit 0: Disable 1: Enable
2	TIM2EN	Timer2 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 ~ 10	-	Reserved

5.6.1.9 APB1 bus peripheral clock enable register (APB1_CFG)

Register	Read/Write	Description	Reset Value
APB1_CFG	R/W	APB1 Bus Peripheral Clock Enable Register	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	UART4EN	SPI3_TWI3EN	QTIW1EN	TIM7EN	TIM6EN	TIM5EN	TIM4EN

Bit Number	Bit Symbol	Description
23	ENAPB	APB1 Bus Clock Switch Control 0: Disable 1: Enable

Bit Number	Bit Symbol	Description
22 ~ 20	CLKDIV[2:0]	APB1 Clock Division Setting APB1 Bus Clock PCLK1 Division Ratio from HCLK: 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
6	UART4EN	UART4 Clock Enable Bit 0: Disable 1: Enable
5	SPI3_TWI3EN	SPI3_TWI3 Clock Enable Bit 0: Disable 1: Enable
4	QTIWI1EN	QSPI1_TWI1 Clock Enable Bit 0: Disable 1: Enable
3	TIM7EN	Timer7 Clock Enable Bit 0: Disable 1: Enable
2	TIM6EN	Timer6 Clock Enable Bit 0: Disable 1: Enable
1	TIM5EN	Timer5 Clock Enable Bit 0: Disable 1: Enable
0	TIM4EN	Timer4 Clock Enable Bit 0: Disable 1: Enable
31 ~ 24 19 ~ 8	-	Reserved

5.6.1.10 APB2 bus peripheral clock enable register (APB2_CFG)

Register	Read/Write	Description	Reset Value
APB2_CFG	R/W	APB2 Bus Peripheral Clock Enable Register	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
-	-	-	-	-	UART3EN	LCDEN	-

Bit Number	Bit Symbol	Description
23	ENAPB	APB2 Bus Clock Switch Control Bit 0: Disable

Bit Number	Bit Symbol	Description
		1: Enable
22 ~ 20	CLKDIV[2:0]	APB2 Clock Division Setting Bit APB2 Bus Clock PCLK2 Division from HCLK: 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: $f_{PCLK2} = f_{HCLK} / 128$
2	UART3EN	UART3 Clock Enable Bit 0: Disable 1: Enable
1	LCDEN	LCD/LED Module Clock Enable Bit 0: LCD/LED Clock Disabled 1: LCD/LED Clock Enabled
31 ~ 24 19 ~ 3 0	-	Reserved

5.6.1.11 AHB bus peripheral reset control register (AHB_RST)

Register	Read/Write	Description	Reset Value
AHB_RST	R/W	AHB Bus Peripheral Reset 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
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	CANRST	-	-	CRCRST	DMARST

Bit Number	Bit Symbol	Description
4	CANRST	CAN Reset Control Bit This bit is set to 1 by software and cleared to 0 automatically by hardware. 0: No effect 1: Reset CAN
1	CRCRST	CRC Reset Control Bit This bit is set to 1 by software and cleared to 0 automatically by hardware. 0: No effect 1: Reset RCC
0	DMARST	DMA Reset Control Bit This bit is set to 1 by software and cleared to 0 automatically by hardware. 0: No effect 1: Reset DMA
31 ~ 5 3 ~ 2	-	Reserved

5.6.1.12 APB0 bus peripheral reset control register (APB0_RST)

Register	Read/Write	Description	Reset Value
APB0_RST	R/W	APB0 Bus Peripheral Reset 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
-	-	-	-	-	-	UART5RST	PWM0RST
7	6	5	4	3	2	1	0
UART1RST	UART0RST	SPI2_TWI2 RST	QTIW0RST	TIM3RST	TIM2RST	TIM1RST	TIM0RST

Bit Number	Bit Symbol	Description
9	UART5RST	UART5 Reset Control Bit This bit is set to 1 by software and cleared to 0 automatically by hardware. 0: No effect 1: Reset UART5
8	PWM0RST	PWM0 Reset Control Bit This bit is set to 1 by software and cleared to 0 automatically by hardware. 0: No effect 1: Reset PWM0
7	UART1RST	UART1 Reset Control Bit This bit is set to 1 by software and cleared to 0 automatically by hardware. 0: No effect 1: Reset UART1
6	UART0RST	UART0 Reset Control Bit This bit is set to 1 by software and cleared to 0 automatically by hardware. 0: No effect 1: Reset UART0
5	SPI2_TWI2RST	SPI2_TWI2 Reset Control Bit This bit is set to 1 by software and cleared to 0 automatically by hardware. 0: No effect 1: Reset SPI2_TWI2
4	QTIW0RST	QSPI0_TWI0 Reset Control Bit This bit is set to 1 by software and cleared to 0 automatically by hardware. 0: No effect 1: Reset QSPI0_TWI0
3	TIM3RST	Timer3 Reset Control Bit This bit is set to 1 by software and cleared to 0 automatically by hardware. 0: No effect 1: Reset Timer3
2	TIM2RST	Timer2 Reset Control Bit This bit is set to 1 by software and cleared to 0 automatically by hardware. 0: No effect

Bit Number	Bit Symbol	Description
		1: Reset Timer2
1	TIM1RST	Timer1 Reset Control Bit This bit is set to 1 by software and cleared to 0 automatically by hardware. 0: No effect 1: Reset Timer1
0	TIM0RST	Timer0 Reset Control Bit This bit is set to 1 by software and cleared to 0 automatically by hardware. 0: No effect 1: Reset Timer0
31 ~ 10	-	Reserved

5.6.1.13 APB bus peripheral reset control register (APB1_RST)

Register	Read/Write	Description	Reset Value
APB1_RST	R/W	APB1 Bus Peripheral Reset 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
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
UART2RST	UART4RST	SPI3_TWI3 RST	QTIWI1RST	TIM7RST	TIM6RST	TIM5RST	TIM4RST

Bit Number	Bit Symbol	Description
7	UART2RST	UART2 Reset Control Bit This bit is set to 1 by software and cleared to 0 automatically by hardware. 0: No effect 1: Reset UART2
6	UART4RST	UART4 Reset Control Bit This bit is set to 1 by software and cleared to 0 automatically by hardware. 0: No effect 1: Reset UART4
5	SPI3_TWI3RST	SPI3_TWI3 Reset Control Bit This bit is set to 1 by software and cleared to 0 automatically by hardware. 0: No effect 1: Reset SPI3_TWI3
4	QTIWI1RST	QSPI1_TWI1 Reset Control Bit This bit is set to 1 by software and cleared to 0 automatically by hardware. 0: No effect 1: Reset QSPI1_TWI1
3	TIM7RST	Timer7 reset control bit This bit is set to 1 by software and cleared to 0 automatically by hardware. 0: No effect

Bit Number	Bit Symbol	Description
		1: Reset Timer7
2	TIM6RST	Timer6 reset control bit This bit is set to 1 by software and cleared to 0 automatically by hardware. 0: No effect 1: Reset Timer6
1	TIM5RST	Timer5 reset control bit This bit is set to 1 by software and cleared to 0 automatically by hardware. 0: No effect 1: Reset Timer5
0	TIM4RST	Timer4 reset control bit This bit is set to 1 by software and cleared to 0 automatically by hardware. 0: No effect 1: Reset Timer4
31 ~ 8	-	Reserved

5.6.1.14 APB2 bus peripheral reset control register (APB2_RST)

Register	Read/Write	Description	Reset Value
APB2_RST	R/W	APB2 bus peripheral reset 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
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	-	UART3RST	LCDRST	-

Bit Number	Bit Symbol	Description
2	UART3RST	UART3 reset control bit This bit is set to 1 by software and cleared to 0 automatically by hardware. 0: No effect 1: Reset UART3
1	LCDRST	LCD/LED module reset control bit This bit is set to 1 by software and cleared to 0 automatically by hardware. 0: No effect 1: Reset
31 ~ 3 0	-	Reserved

5.6.1.15 NMI interrupt configuration register (NMI_CFG)

Register	Read/Write	Description	Reset Value
NMI_CFG	R/W	Non-maskable interrupt (NMI) configuration register	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
-	-	-	-	SRAMPEEN	INT0EN	CM PEN	CSS EN

Bit Number	Bit Symbol	Description
31 ~ 16	KEY[15:0]	NMI_CFG register write protection enable Write 0xA05F to KEY[15:0] to unlock before modifying the register's lower bits.
3	SRAMPEEN	SRAM Parity Check Error Interrupt Enable Bit 0: Disable NMI trigger for SRAM Parity Check Error 1: Enable SRAM Parity Check Error as a Non-Maskable Interrupt source When enabled, detecting an SRAM Parity Check Error during SRAM read triggers an NMI. The corresponding flag must be cleared manually to exit the NMI.
2	INT0EN	External Interrupt INT0 Non-Maskable Interrupt Enable Bit 0: Disable NMI trigger for INT0 1: Enable INT0 as a Non-Maskable Interrupt source When enabled, rising and falling edge interrupts on INT0 trigger an NMI. The corresponding flag must be cleared manually to exit the NMI. Note: When INT0 interrupt is enabled, NMI is still prioritized.
1	CM PEN	Comparator non-maskable interrupt source enable bit 0: CMP disables NMI trigger 1: Enables CMP non-maskable interrupt source When enabled, setting CMPIF triggers NMI. Exiting NMI requires manually clearing the CMPIF flag. Note: If CMP interrupt is enabled (CMPCFG->CMPIM[1:0] = 1), NMI remains prioritized.
0	CSS EN	Clock Security System non-maskable interrupt source enable bit 0: CSS disables NMI trigger 1: Enables CSS system non-maskable interrupt source When enabled, setting CLKFIF triggers NMI. The CLKFIF flag can be cleared after reset. Note: If RCC Interrupts are enabled (RCC_CFG0.INTEN=1), NMI interrupts will retain highest priority.
15 ~ 4	-	Reserved

5.6.2 RCC register mapping

Register	Offset Address	Read/Write	Description	Reset Value
RCC Base Address: 0x4000_3000				
AHB_CFG	0x00	R/W	AHB Bus Peripheral Clock Enable Register	0x0010_0000
AHB_RST	0x04	R/W	AHB Bus Peripheral Reset Control Register	0x0000_0000
RCC_KEY	0x0C	R/W	RCC Protection Register	0x0000_0000
RCC_CFG0	0x14	R/W	System Clock Source Selection Register	0x0000_1040

Register	Offset Address	Read/Write	Description	Reset Value
RCC_CFG1	0x18	R/W	Peripheral clock source selection register	0x0000_0000
PLL_CFG	0x1C	R/W	Peripheral clock source selection register	0x0000_0000
RCC_STS	0x20	R/W	Clock Status Register	0x0000_0000
SYST_CALIB	0x28	R	SysTick Calibration Parameter Register	0x0000_2327
NMI_CFG	0x2C	R/W	NMI Interrupt Configuration Register	0x0000_0000

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

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

Register	Offset Address	Read/Write	Description	Reset Value
RCC Base Address: 0x4002_2000				
APB2_CFG	0x00	R/W	APB2 Bus Peripheral Clock Enable Register	0x0000_0000
APB2_RST	0x04	R/W	APB2 bus peripheral reset control register	0x0000_0000

6 Interrupts

- The M0+ core supports up to 32 interrupt sources, numbered 0 to 31. The SC32F11T/11G Series provides 25 interrupt sources.
- Four interrupt priority levels are configurable. Interrupt priorities are set via the core's Interrupt priority registers.

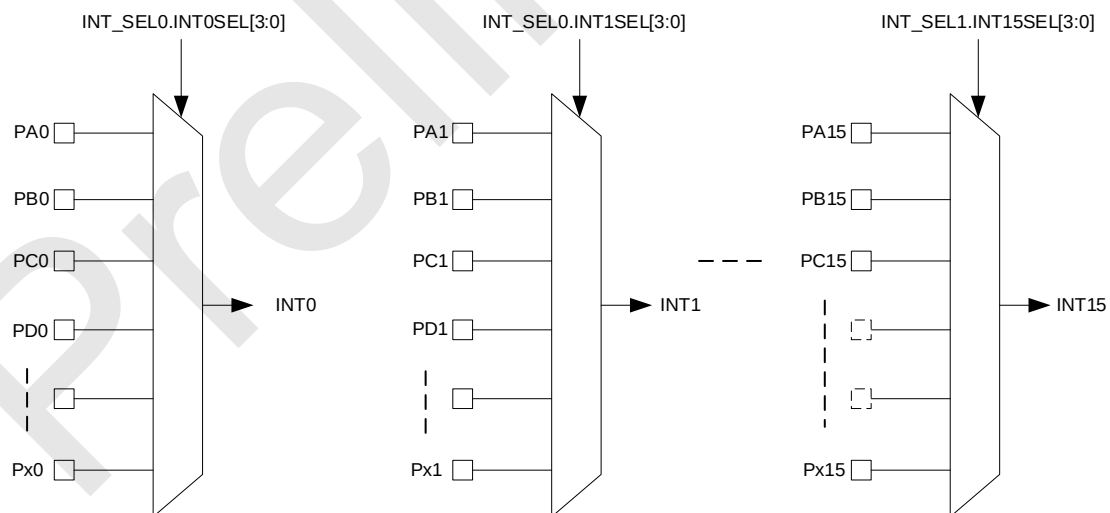
6.1 External interrupts INT0 to INT15

There are 16 external interrupt sources that share 4 interrupt vectors. All 16 sources can be configured to trigger on Rising Edge, Falling Edge, or Both Edge. Once configured, they can map to any GPIO pin. Setting the corresponding interrupt flag (RIF/FIF = 1) in software triggers the respective interrupt.

The external interrupt features of the SC32F11T/11G Series are as follows:

- 16 INT interrupt sources sharing 4 interrupt vectors.
- INT signals can be remapped to all GPIO pins after configuration.
- All interrupts support configurable Rising Edge, Falling Edge, and Both Edge triggers, each with a dedicated Interrupt Flag bit.
- Setting the corresponding Interrupt Flag via software can trigger the corresponding interrupt.

Note: When switching INT functions, users must manually set the GPIO Port of INT_n (n=0 to 15) to input mode with Pull-up enabled; external interrupts cannot be detected when the port is in output mode.



6.2 Interrupt and events

- When NVIC is disabled, the interrupt request mask bit is enabled, generating events without generating Interrupts.
- After NVIC is enabled, the interrupt request mask bit serves as the module's global Interrupt switch.

6.3 Interrupt vector table

Interrupts Vector number	Interrupts	Priority	Interrupt Vector Address	Interrupt sources	Core/NVIC Enable Bit	Interrupt Request Mask Bit	Interrupt Sub Switch	Interrupt Flag Bit	Wake up from 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	-	Configurable		SVC_Handler	PRIMASK	SCB	\	\	YES
12 ~ 13	-	-	0x0000_0030 0x0000_0034	-		-	\	\	YES
14	-	Configurable	0x0000_0038	PendSV_Handler	PRIMASK	SCB	\	\	YES
15	-	Configurable	0x0000_003C	SysTick_Handler	PRIMASK	SysTick_CTRL	\	\	NO
16	0	Configurable	0x0000_0040	INT0	NVIC->ISER[0].0	INTF_IE->ENFx,x=0 INTR_IE->ENRx	\	INTF_STS->FIFx INTR_STS->RIFx	YES
17	1	Configurable	0x0000_0044	INT1-7	NVIC->ISER[0].1	INTF_IE->ENFx,x=1 to 7 INTR_IE->ENRx	\	INTF_STS->FIFx INTR_STS->RIFx	YES
18	2	Configurable	0x0000_0048	INT8-11	NVIC->ISER[0].2	INTF_IE->ENFx,x=8 to 11 INTR_IE->ENRx	\	INTF_STS->FIFx INTR_STS->RIFx	YES
19	3	Configurable	0x0000_004C	INT12-15	NVIC->ISER[0].3	INTF_IE->ENFx,x=12 to 15 INTR_IE->ENRx	\	INTF_STS->FIFx INTR_STS->RIFx	YES
20	4	Configurable	0x0000_0050	RCC Oscillation Failure Detection	NVIC->ISER[0].4	RCC_CFG->INTEN	\	RCC_STS->CLKFIF	NO
21	5	Reserved	0x0000_0054	\	NVIC->ISER[0].5	\	\	\	
22	6	Configurable	0x0000_0058	BTM	NVIC->ISER[0].6	BTM_CON->INTEN	\	BTM_STS->BTMIF	YES
23	7	Configurable	0x0000_005C	UART0	NVIC->ISER[0].7	UART0_IDE->INTEN	UART0_IDE->TXIE UART0_IDE->RXIE	UART0_STS->TXIF UART0_STS->RXIF	YES

Interrupts Vector number	Interrupts	Priority	Interrupt Vector Address	Interrupt sources	Core/NVIC Enable Bit	Interrupt Request Mask Bit	Interrupt Sub Switch	Interrupt Flag Bit	Wake up from stop
				UART2		UART2_IDE->INTEN	UART2_IDE->TXIE UART2_IDE->RXIE	UART2_STS->TXIF UART2_STS->RXIF	YES
				UART4		UART4_IDE->INTEN	UART4_IDE->TXIE UART4_IDE->RXIE	UART4_STS->TXIF UART4_STS->RXIF	YES
24	8	Configurable	0x0000_0060	UART1	NVIC->ISER[0].8	UART1_IDE->INTEN	UART1_IDE->TXIE UART1_IDE->RXIE	UART1_STS->TXIF UART1_STS->RXIF	YES
				UART3		UART3_IDE->INTEN	UART3_IDE->TXIE UART3_IDE->RXIE	UART3_STS->TXIF UART3_STS->RXIF	YES
				UART5		UART5_IDE->INTEN	UART5_IDE->TXIE UART5_IDE->RXIE	UART5_STS->TXIF UART5_STS->RXIF	YES
25	9	Configurable	0x0000_0064	QSPI0/SPI0/TWI0	NVIC->ISER[0].9	TWI_QSPI0_IDE->INTEN	TWI_QSPI0_IDE->QTWIE TWI_QSPI0_IDE->RXNEIE TWI_QSPI0_IDE->TBIE TWI_QSPI0_IDE->RXIE TWI_QSPI0_IDE->RXHIE TWI_QSPI0_IDE->TXHIE	TWI_QSPI0_STS->QTWIF TWI_QSPI0_STS->RXNEIF TWI_QSPI0_STS->TXEIF TWI_QSPI0_STS->RXHFIF TWI_QSPI0_STS->RXHIF TWI_QSPI0_STS->TXHIF	NO
				SPI2/TWI2		TWI_SPI2_IDE->INTEN	TWI_SPI2_IDE->TBIE	TWI_SPI2_STS->QTWIF TWI_SPI2_STS->TXEIF	NO
26	10	Configurable	0x0000_0068	QSPI1/SPI1/TWI1	NVIC->ISER[0].10	TWI_QSPI1_IDE->INTEN	TWI_QSPI1_IDE->QTWIE TWI_QSPI1_IDE->RXNEIE TWI_QSPI1_IDE->TBIE TWI_QSPI1_IDE->RXIE TWI_QSPI1_IDE->RXHIE TWI_QSPI1_IDE->TXHIE	TWI_QSPI1_STS->QTWIF TWI_QSPI1_STS->RXNEIF TWI_QSPI1_STS->TXEIF TWI_QSPI1_STS->RXHFIF TWI_QSPI1_STS->RXHIF TWI_QSPI1_STS->TXHIF	NO
				SPI3/TWI3		TWI_SPI3_IDE->INTEN	TWI_SPI3_IDE->TBIE	TWI_SPI3_STS->QTWIF TWI_SPI3_STS->TXEIF	NO
27	11	Configurable	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	Configurable	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	Configurable	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	Configurable	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	NO

Interrupts Vector number	Interrupts	Priority	Interrupt Vector Address	Interrupt sources	Core/NVIC Enable Bit	Interrupt Request Mask Bit	Interrupt Sub Switch	Interrupt Flag Bit	Wake up from stop
								DMA _n _STS->HTIF DMA _n _STS->TEIF	
31	15	Configurable	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	Configurable	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	Configurable	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	Configurable	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	Configurable	0x0000_008C	TIM4	NVIC->ISER[0].19	TIM4_IDE->INTEN	TIM4_IDE->TIE TIM4_IDE->EXFIE TIM4_IDE->EXRIE	TIM4_STS->TIF TIM4_STS->EXIF TIM4_STS->EXIR	NO
				TIM5		TIM5_IDE->INTEN	TIM5_IDE->TIE TIM5_IDE->EXFIE TIM5_IDE->EXRIE	TIM5_STS->TIF TIM5_STS->EXIF TIM5_STS->EXIR	NO
36	20	Configurable	0x0000_0090	TIM6	NVIC->ISER[0].20	TIM6_IDE->INTEN	TIM6_IDE->TIE TIM6_IDE->EXFIE TIM6_IDE->EXRIE	TIM6_STS->TIF TIM6_STS->EXIF TIM6_STS->EXIR	NO
				TIM7		TIM7_IDE->INTEN	TIM7_IDE->TIE TIM7_IDE->EXFIE TIM7_IDE->EXRIE	TIM7_STS->TIF TIM7_STS->EXIF TIM7_STS->EXIR	NO
37	21	Configurable	0x0000_0094	PWM0	NVIC->ISER[0].21	PWM0_CON->INTEN	\	PWM0_STS->PWMIF	NO
38	22	Reserved	0x0000_0098	\	NVIC->ISER[0].22	\	\	\	
39	23	Reserved	0x0000_009C	\	NVIC->ISER[0].23	\	\	\	
40	24	Reserved	0x0000_00A0	\	NVIC->ISER[0].24	\	\	\	
41	25	Reserved	0x0000_00A4	\	NVIC->ISER[0].25	\	\	\	
42	26	Reserved	0x0000_00A8	\	NVIC->ISER[0].26	\	\	\	
43	27	Reserved	0x0000_00AC	\	NVIC->ISER[0].27	\	\	\	
44	28	Configurable	0x0000_00B0	CAN	NVIC->ISER[0].28	\	CAN_RTIE->RIE CAN_RTIE->ROIE CAN_RTIE->RFIE	CAN_RTIE->RIF CAN_RTIE->ROIF CAN_RTIE->RFIF	NO

Interrupts Vector number	Interrupts	Priority	Interrupt Vector Address	Interrupt sources	Core/NVIC Enable Bit	Interrupt Request Mask Bit	Interrupt Sub Switch	Interrupt Flag Bit	Wake up from stop
							CAN_RTIE->RAFIE CAN_RTIE->TPIE CAN_RTIE->TSIE CAN_RTIE->EIE CAN_RTIE->EPIE CAN_RTIE->ALIE CAN_RTIE->BEIE	CAN_RTIE->RAFIF CAN_RTIE->TPIF CAN_RTIE->TSIF CAN_RTIE->EIF CAN_RTIE->EPIF CAN_RTIE->ALIF CAN_RTIE->BEIF	
45	29	Configurable	0x0000_00B4	ADC	NVIC->ISER[0].29	ADC_CON->INTEN	\	ADC_STS->ADCIF	NO
46	30	Configurable	0x0000_00B8	CMP	NVIC->ISER[0].30	CMPCFG->CMPIM[1:0]	\	CMP_STS->CMPIF	YES
47	31	Configurable	0x0000_00BC	TK	NVIC->ISER[0].31	TKCON->INTEN	\	TKCON->TKIF	YES

6.4 External interrupt registers

6.4.1 External interrupt related registers table

6.4.1.1 INT interrupt falling edge enable register INTF_IE

Register	Read/Write	Description	Reset Value
INTF_IE	R/W	INT Interrupt Falling Edge Enable 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
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 Symbol	Description
15 ~ 0	ENFx (x=0 to 15)	INTx Falling Edge Interrupt Enable Control Bit (x=0 to 15) 0: None 1: Enable
31 ~ 16	-	Reserved

6.4.1.2 INT interrupt rising edge enable register INTR_IE

Register	Read/Write	Description	Reset Value
INTR_IE	R/W	INT Interrupt Rising Edge Enable 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
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 Symbol	Description
15 ~ 0	ENRx (x=0 to 15)	INTx Rising Edge Interrupt Enable Control Bit (x=0 to 15) 0: None 1: Enable
31 ~ 16	-	Reserved

6.4.1.3 External interrupt port selection register 0 INT_SEL0

Register	Read/Write	Description	Reset Value
INT_SEL0	R/W	External Interrupt Port Selection Register 0	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 Symbol	Description
31 ~ 0	INTxSEL[3:0] (x=0 to 7)	External Interrupt INTx Port Select Bit (x=0 to 7) 0000: Select PAX Port 0001: Select PBx Port 0010: Select PCx GPIO port 0011: Select PDx GPIO port Others: Reserved Note: Each external interrupt line can be mapped to only one GPIO port at a time

6.4.1.4 External interrupt port selection register 1 (INT_SEL1)

Register	Read/Write	Description	Reset Value
INT_SEL1	R/W	External Interrupt Port Selection Register 1	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 Symbol	Description
31 ~ 0	INTxSEL[3:0] (x=8 to 15)	External Interrupt INTx Port Select Bit (x=8 to 15) 0000: Select PAX Port 0001: Select PBx Port 0010: Select PCx GPIO port 0011: Select PDx GPIO port Others: Reserved Note: Each external interrupt line can be mapped to only one GPIO port at a time

6.4.1.5 External interrupt falling edge control register (INTF_CON)

Register	Read/Write	Description	Reset Value
INTF_CON	R/W	Falling Edge Control Register for External Interrupts	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 Symbol	Description
15 ~ 0	FTx (x=0 to 15)	INTx Falling Edge Detection Enable (x=0 to 15) 0: None 1: Enable
31 ~ 16	-	Reserved

6.4.1.6 External interrupt rising edge control register (INTR_CON)

Register	Read/Write	Description	Reset Value
INTR_CON	R/W	Rising Edge Control Register for External Interrupts	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 Symbol	Description
15 ~ 0	RTx (x=0 to 15)	INTx Rising Edge Detection Enable (x=0 to 15) 0: None 1: Enable
31 ~ 16	-	Reserved

6.4.1.7 External interrupt falling edge flag register (INTF_STS)

Register	Read/Write	Description	Reset Value
INTF_STS	R/W	External Interrupt Falling Edge Flag 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
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 Symbol	Description
15 ~ 0	FIFx (x=0 to 15)	INTx Port Falling Edge Capture Flag (x=0 to 15) Falling edge detected; this bit is set by hardware and cleared by writing 0 from software. Writing 1 to this bit from software also triggers the falling edge capture interrupt.
31 ~ 16	-	Reserved

6.4.1.8 External interrupt rising edge flag register (INTR_STS)

Register	Read/Write	Description	Reset Value
INTR_STS	R/W	External Interrupt Rising Edge Flag 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
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 Symbol	Description
15 ~ 0	RIFx (x=0 to 15)	INTx Port Rising Edge Capture Flag (x=0 to 15) Rising edge detected; this bit is set by hardware and cleared by writing 0 from software. Writing 1 to this bit from software also triggers the rising edge capture interrupt.
31 ~ 16	-	Reserved

6.4.2 External interrupt register mapping

Register	Offset Address	Read/Write	Description	Reset Value
External Interrupt Base Address: 0x4001_1800				
INTF_IE	0x00	R/W	INT Interrupt Falling Edge Enable Register	0x0000_0000
INTR_IE	0x20	R/W	INT Interrupt Rising Edge Enable Register	0x0000_0000
INT_SEL0	0x40	R/W	External Interrupt Port Selection Register 0	0x0000_0000
INT_SEL1	0x60	R/W	External Interrupt Port Selection Register 1	0x0000_0000
INTF_CON	0x80	R/W	Falling Edge Control Register for External Interrupts	0x0000_0000
INTR_CON	0xA0	R/W	Rising Edge Control Register for External Interrupts	0x0000_0000

Register	Offset Address	Read/Write	Description	Reset Value
INTF_STS	0xC0	R/W	External Interrupt Falling Edge Flag Register	0x0000_0000
INTR_STS	0xE0	R/W	External Interrupt Rising Edge Flag Register	0x0000_0000

Preliminary

7 Power saving mode

After initial power-up, the system runs by default in Normal Mode, with three additional power saving modes available:

- Low-Speed Mode: System clock source selects LIRC; CPU runs at 37 kHz
- IDLE Mode, can be woken up by any Interrupts
- STOP mode wake-up sources include INT0 to 15, Base Timer, UART, TK, and CMP

8 GPIO

8.1 Clock source

The M0+ core can access GPIO through the IOPORT bus in a single cycle, resulting in extremely high data transfer efficiency. The clock for the IOPORT bus is derived from HCLK.

8.2 Feature

The GPIO Port Features of the SC32F11T/11G Series are as follows:

- Up to 60 bidirectional GPIOs independently controllable
- The CPU can access GPIO ports via the IOPORT bus within a single cycle
- Independent pull-up resistor configuration is available
- Drive Capability of the COM port can reach 100 mA@0.8 V
- Drive Capability of the conventional port can reach 55 mA@0.8 V
- 16 GPIOs per group
- In both input and output modes, reading the port data register returns the actual state of the port pins.

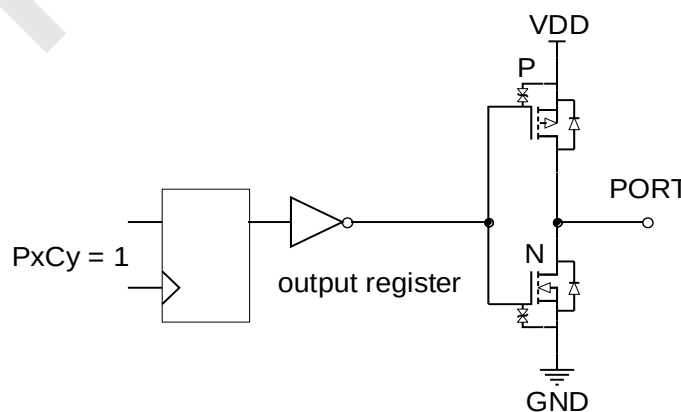
Note: Unused ports and ports not exposed in the package must be set to strong push-pull output mode.

8.3 GPIO block diagram

8.3.1 Strong push-pull output mode

In Strong Push-Pull Output Mode, the pin provides continuous high-current drive. Refer to the GPIO Parameter section of the SC32F11T/11G Datasheet for electrical specifications.

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

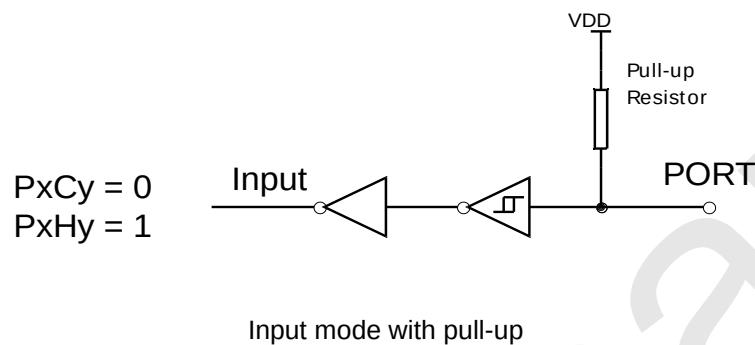


Strong push-pull output mode

8.3.2 Input mode with pull-up

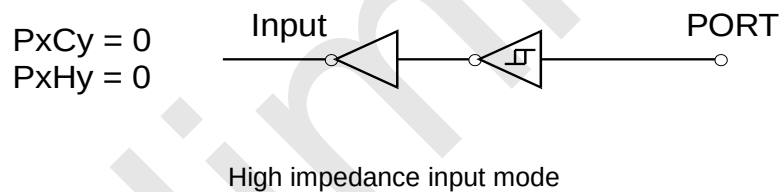
In the input mode with Pull-up, a pull-up resistor is permanently connected to the input port; a low-level signal is detected only when the input port voltage is pulled down.

The port structure diagram for the input mode with Pull-up is shown below:



8.3.3 High impedance input mode (input only)

The port structure diagram for High Impedance Input Mode is shown below:



8.4 GPIO register

8.4.1 GPIO related register table

8.4.1.1 PX port data register PX

Register	Read/Write	Description	Reset Value
PX X=A,B,C,D	R/W	PX Port 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
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 Symbol	Description
15 ~ 0	PDn (n=0 to 15)	PXn Port Data Register, X = A, B, C, D; n = 0 to 15 Port latch register data. Values read from the port data register represent the actual port state.
31 ~ 16	-	Reserved

8.4.1.2 PX port data register PXn_BIT

Register	Read/Write	Description	Reset Value
PXn_BIT X=A,B,C,D n=0 ~ 15	R/W	PX Port 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
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	BSRn

Bit Number	Bit Symbol	Description
0	BSRn	PXn Port Bit Assignment Control, n = 0 to 15 Used for individual assignment to PXn port
31 ~ 1	-	Reserved

8.4.1.3 PX port data register PXn_XR

Register	Read/Write	Description	Reset Value
PXn_XR X=A,B,C,D n=0 ~ 15	R/W	Toggle PXn	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 Symbol	Description
0	XRn	PXn Port Bit Toggle Control, n = 0 to 15 0: None

Bit Number	Bit Symbol	Description
		1: Toggles PXn output
31 ~ 1	-	Reserved

8.4.1.4 PX port input/output control register PXCON

Register	Read/Write	Description	Reset Value
PXCON X=A,B,C,D	R/W	PX Input/Output 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
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 Symbol	Description
15 ~ 0	MODEn (n=0 to 15)	Strong Push-Pull Mode Enable Bit for PXn, n=0 to 15 0: PXn in Input Mode (Power-on Default) 1: PXn in Strong Push-Pull Output Mode
31 ~ 16	-	Reserved

8.4.1.5 PX pull-up resistor control register PXPB

Register	Read/Write	Description	Reset Value
PXPB X=A,B,C,D	R/W	PX Pull-up Resistor 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
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 Symbol	Description
15 ~ 0	PUPDn (n=0 to 15)	Pull-up Resistor Enable Bit for PXn, n=0 to 15 0: PXn in High Impedance Input Mode (Power-on Default), Pull-up Disabled 1: PXn Pull-up Enabled

Bit Number	Bit Symbol	Description
31 ~ 16	-	Reserved

8.4.2 GPIO register mapping

Register	Offset Address	Read/Write	Description	Reset Value
PA Base Address: 0x40011000				
PA	0x00	R/W	PA Data Register	0x0000_0000
PACON	0x20	R/W	PA Input/Output Control Register	0x0000_0000
PAPH	0x40	R/W	PA Pull-up Resistor Control Register	0x0000_0000

Register	Offset Address	Read/Write	Description	Reset Value
PB Base Address: 0x4001_1100				
PB	0x00	R/W	PB GPIO Port Data Register	0x0000_0000
PBCON	0x20	R/W	PB GPIO Port Input/Output Control Register	0x0000_0000
PBPH	0x40	R/W	PB GPIO Port Pull-up Control Register	0x0000_0000

Register	Offset Address	Read/Write	Description	Reset Value
PC Base Address: 0x4001_1200				
PC	0x00	R/W	PC GPIO Port Data Register	0x0000_0000
PCCON	0x20	R/W	PC GPIO Port Input/Output Control Register	0x0000_0000
PCPH	0x40	R/W	PC GPIO Port Pull-up Control Register	0x0000_0000

Register	Offset Address	Read/Write	Description	Reset Value
PD Base Address: 0x4001_1300				
PD	0x00	R/W	PD GPIO Port Data Register	0x0000_0000
PDCON	0x20	R/W	PD GPIO Port Input/Output Control Register	0x0000_0000
PDPH	0x40	R/W	PD GPIO Port Pull-up Control Register	0x0000_0000

9 Analog-to-digital converter (ADC)

9.1 Clock source

- The ADC clock source for the SC32F11T/11G Series is a single source: PCLK2
- ADC conversion time is approximately 950 ns

9.2 Feature

- Precision: 14-bit
- Supports up to 20 channels: 18 external ADC sampling channels with other IO port functions multiplexed
 - One internal ADC channel can directly measure V_{DD} voltage
 - One Internal ADC Channel can directly measure OP output
- Built-in 2.4 V, 2.048 V, and 1.024 V reference voltages
- The ADC reference voltage has 4 options: V_{DD} , 2.4 V, 2.048 V, 1.024 V
- ADC Input Channel Selection
 - The ADC input channel is selected by configuring the ADCIS[4:0] bits
- Conversion can be started through software
- ADC Conversion Completion Interrupt can be enabled
- The total time from sampling to conversion completion is as low as 2 μ s
- Supports DMA transfer: a DMA Request is generated upon ADC conversion completion
- Supports Single Channel Continuous Conversion Mode
- The ADC conversion result includes an overflow flag. The OVERRUN flag and the ADC conversion result share the ADCV register, enabling a single read operation.

9.3 ADC conversion steps

The steps users need to follow for performing ADC conversion are as follows:

1. Configure the ADC input Pin; (Configure the bit corresponding to AINx as an ADC input, usually ADC Pins are preset)
2. Set the ADC Reference Voltage V_{ref} and the frequency for ADC Conversion
3. Write 1 to ADCEN to Enable the ADC module Power Supply
4. Select the ADC input Channel by setting the ADCIS bit.
5. Start ADCS to begin Conversion
6. Wait until ADCIF=1. If ADC Interrupts are Enabled, an ADC Interrupt will be generated, and the user must clear the ADCIF Flag in software
7. Obtain 14-bit data from ADCV after each conversion is completed
8. If not changing the input channel, set CONT to 1 in software to Enable Single Channel Continuous Conversion Mode. Conversion continues until this bit is cleared.
9. When ADC conversion results overflow, the OVERRUN Flag is set to 1.
10. Conversion data can be transferred via DMA.

Note: Before setting ADC_CON[8] (INTEN), users should clear ADCIF in software first and also clear

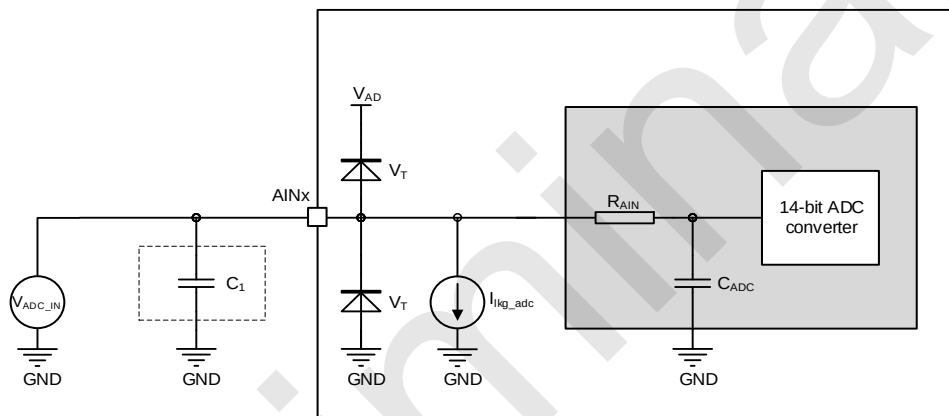
ADCIF at the end of the ADC Interrupt Service Routine to prevent continuous ADC Interrupts.

9.4 ADC interrupts

After ADC conversion completion in the SC32F11T/11G Series, ADCIF will be set. If ADC_CON.INTEN=1, an interrupt will be generated.

Interrupt Event	Event Flag	Interrupt Enable Control Bit
ADC Conversion Complete Interrupt Request	ADCIF	ADC_CON->INTEN

9.5 ADC connection circuit diagram



Description: It is recommended to connect a 103 capacitor externally at C1 to improve ADC sampling performance. ADC-related electrical parameters are detailed in the 'SC32F11T/11G_Datasheet'.

9.6 ADC registers

9.6.1 ADC register table

9.6.1.1 ADC control register ADC_CON

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

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

Bit Number	Bit Symbol	Description
21 ~ 20	VREFS[1:0]	Reference Voltage Selection Bits 00: Select ADC V_{REF} as V_{DD} 01: Select the ADC V_{VREF} 10: Select the ADC V_{REF} as the internally accurate $1.0V_{REF}$ as the internally accurate $2.4 V$
18 ~ 16	LOWSP[2:0]	ADC Sampling Time Selection Bits 100: Sampling time is 3 system clocks (approximately 100 ns @ $f_{PCLK2} = 32$ MHz) 101: Sampling time approximately 6 system clocks (about 200 ns @ $f_{PCLK2} = 32$ MHz) 110: Sampling time approximately 16 system clocks (about 500 ns @ $f_{PCLK2} = 32$ MHz) 111: Sampling time approximately 32 system clocks (about 1000 ns @ $f_{PCLK2} = 32$ MHz) Others: Reserved Description: Total ADC time from sampling to conversion completion is calculated as: $T_{ADC} = \text{Sampling time} + \text{Conversion time}$ ADC conversion time is approximately 950 ns.
15	ADCEN	ADC module power control bit 0: ADC module power off 1: ADC module power on
12	DMAEN	DMA request enable bit This bit enables DMA request generation. Writing 1 enables the DMA controller to automatically handle ADC conversion data. 0: Disable DMA Request 1: Enable DMA Request Note: When writing to this bit in software, ensure no conversion is ongoing.
11	CONT	Single/Continuous Conversion Mode Selection Bit This bit is set and cleared by software. When set, conversion continues until cleared. 0: Single Conversion Mode 1: Continuous Conversion Mode
8	INTEN	CPU Interrupt Request Enable Control Bit 0: Disable interrupt request 1: Enable interrupt request
7	ADCS	ADC Conversion Trigger Control Bit This bit triggers an ADC conversion. It is set by software and cleared by hardware. Writing 1 starts a single conversion. Note: After setting ADCS to 1, do not write to the ADC_CON register until the ADCIF interrupt flag is set.
4 ~ 0	ADCIS[4:0]	ADC Input Channel Selection Bits 00000: Select AIN0 as ADC input 00001: Select AIN1 as ADC input 00010: Select AIN2 as ADC input 00011: Select AIN3 as ADC input 00100: Select AIN4 as ADC input 00101: Select AIN5 as ADC input 00110: Select AIN6 as ADC input 00111: Select AIN7 as ADC input 01000: Select AIN8 as ADC input 01001: Select AIN9 as ADC input 01010: Select AIN10 as ADC input 01011: Select AIN11 as ADC input 01100: Select AIN12 as ADC input 01101: Select AIN13 as ADC input 01110: Select AIN14 as ADC input 01111: Select AIN15 as ADC input

Bit Number	Bit Symbol	Description
		10000: Select AIN16 as ADC input 10001: Select AIN17 as ADC input 10010 to 11101: Reserved 11110: Select OP output as ADC input 11111: ADC input is 1/4 V _{DD} , used to measure power supply voltage
31 ~ 22 19 14 ~ 13 10 ~ 9 6 ~ 5	-	Reserved

9.6.1.2 ADC flag status register ADC_STS

Register	Read/Write	Description	Reset Value
ADC_STS	R/W	ADC Flag Status 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
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	ADCIF

Bit Number	Bit Symbol	Description
0	ADCIF	ADC Interrupt Request Flag This bit is set by hardware and cleared by writing 1 in software. This bit is set by hardware when ADC conversion completes. If ADC_CON.INTEN=1, an interrupt is generated.
31 ~ 1	-	Reserved

9.6.1.3 ADC conversion value register ADCV

Register	Read/Write	Description	Reset Value
ADCV	R	ADC Conversion Value Register	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
-	-	ADCV[13:8]					
7	6	5	4	3	2	1	0
ADCV[7:0]							

Bit Number	Bit Symbol	Description
31	OVERRUN	Overflow Flag (Read-only) This bit is set by hardware when CPU or DMA fails to handle ADC conversion requests on time, causing overflow. It clears automatically after reading ADCV.

Bit Number	Bit Symbol	Description
		Note: When overflow occurs, the new ADC conversion result overwrites the previous one.
13 ~ 0	ADCV[13:0]	14-bit ADC conversion result
30 ~ 14	-	Reserved

9.6.1.4 ADC port configuration register ADC_CFG

Register	Read/Write	Description	Reset Value
ADC_CFG	R/W	ADC port configuration register	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	AIN17	AIN16
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 Symbol	Description
17 ~ 0	AINx (x=0 to 17)	ADC port configuration register 0: AINx port cannot be used as an ADC input channel 1: AINx port can be used as an ADC input channel. When ADCIS[4:0] selects AINx as an ADC input channel, its pull-up resistor is automatically disabled.
31 ~ 18	-	Reserved

9.6.2 ADC register mapping

Register	Offset Address	Read/Write	Description	Reset Value
ADC base address: 0x4002_2110				
ADC_CON	0x00	R/W	ADC Control Register	0x0000_0000
ADC_STS	0x04	R/W	ADC Flag Status Register	0x0000_0000
ADCV	0x08	R	ADC Conversion Value Register	0x0000_0000
ADC_CFG	0x0C	R/W	ADC port configuration register	0x0000_0000

10 Operational amplifier and programmable gain amplifier (OP)

A built-in OP integrates an operational amplifier and a Programmable Gain Amplifier, providing rail-to-rail operation. The OP can be configured as PGA Mode, featuring 5 Non-inverting Inputs, 2 Inverting Inputs, and 3 Outputs, and supports non-inverting gains of 8, 16, 32, or 64 times and inverting gains of 7, 15, 31, or 63 times.

10.1 Feature

- One Rail-to-Rail Operational Amplifier
- Configurable as PGA Mode
 - Non-inverting Input Gain: 8/16/32/64
 - Inverting Input Gain: 7/15/31/63
- Non-inverting Input routed to two external Pins: OP_P0 or OP_P1
- Inverting Input routed to one external Pin: OP_N
- An external pin is output at the output terminal: OP_O
- The output can be directly connected to the ADC input
- The output can be directly connected to the positive input of CMP
- Precision adjustment is achieved by enabling the PGA offset control bit (PGAOFC=1), which shorts the non-inverting and inverting inputs of the OP module.

10.2 OP port selection

10.2.1 OP precision adjustment

Precision adjustment is achieved by setting the PGA input offset control bit PGOFC=1, shorting the OP module's non-inverting and inverting inputs. Otherwise, PGOFC is set to 0.

10.2.2 OP non-inverting input selection

The OP module has five types of non-inverting inputs: OP_P0 external pin, OP_P1 external pin, internal VSS, internal 1.2 V reference, and V_{DD}. These are selectable via OPPSEL[2:0].

10.2.3 OP inverting input selection

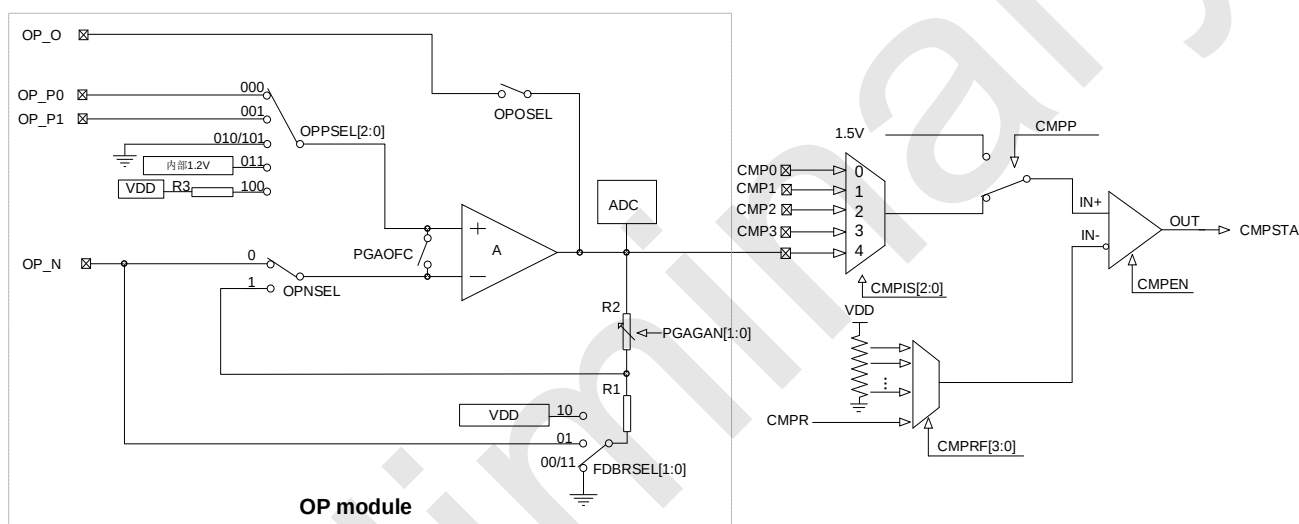
The OP module's inverting input has two options: OP_N external Pin and internal feedback resistor.

- When selecting the OP_N external Pin as the OP's inverting input, set the OP input control bit OPNSEL=0 and the feedback resistor connection selection bits FDBRSEL[1:0]=01.
- When selecting the internal feedback resistor as the OP's inverting input, set OPNSEL=1, FDBRSEL[1:0]=00, 11, or 10, and choose the internal gain stage via the PGAGAN[1:0] bits.

10.2.4 OP output selection

The OP module has three outputs: three pins corresponding to the AD converter sampling channel, the positive input of the CMP, and the OP_O pin.

- To use the OP as the ADC input, first enable the OP module by setting ENOP=1, then enable the ADC power supply by setting ADCEN=1, and select the OP output as the ADC input via ADCIS[4:0]. The OP conversion result can then be read directly from the ADCV register.
- When OP is used as the positive input of the CMP, and OP is enabled, if the CMP input channel control bits CMPIS[2:0] are set to OP, then the CMP's positive input is selected from the OP output.



Register	Read/Write	Description	Reset Value
OP_CON	R/W	OP Control Register	0x0000_0000

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

Bit Number	Bit Symbol	Description
28 ~ 24	TRIMOFFSETN[4:0]	Trim for NMOS differential pairs NMOS Differential Offset Calibration Value of OP
23	PGAOFC	PGA Input Offset Adjustment Control Bit 0: OP Non-inverting and Inverting Inputs Not Internally Shorted 1: OP Non-inverting and Inverting Inputs Internally Shorted Note: Internal short or open between OP non-inverting and inverting inputs does not affect OPPSEL and OPNSEL selection
20 ~ 16	TRIMOFFSETP[4:0]	Trim for PMOS differential pairs PMOS Differential Offset Calibration Value of OP
13 ~ 12	FDBRSEL[1:0]	Feedback Resistor Connection Selection Bit 00/11: Connected Internally to VSS 01: OP_N Pin 10: VDD
9 ~ 8	PGAGAN[1:0]	Internal Gain Stage Selection: 00: Non-inverting 8, Inverting 7 01: Non-inverting 16, Inverting 15 10: Non-inverting 32, Inverting 31 11: Non-inverting 64, Inverting 63
7 ~ 5	OPPSEL[2:0]	OP non-inverting input signal selection 000: OP_P0 (external pin) 001: OP_P1 (external pin) 010: Internally connected to VSS (0 V) 011: Internal 1.2 V Reference 100: VDD 101: Internally connected to VSS (0 V)
3	OPNSEL	OP inverting input signal selection 0: Select OP_N (external pin) 1: Select internal feedback resistor
1	OPOSEL	OP output connection selection bit 0: Disconnect from OP_O 1: Connect OP output to OP_O (external pin)
0	ENOP	OP module power supply enable bit 0: Disable OP module power 1: Enable OP module power
31 ~ 29 22 ~ 21 15 ~ 14 11 ~ 10 4,2	-	Reserved

10.4.2 OP register mapping

Register	Offset Address	Read/Write	Description	Reset Value
OP Base Address: 0x4002_2140				
OP_CON	0x00	R/W	OP Control Register OP_CON	0x0000_0000

11 Analog comparator (CMP)

The SC32F11T/11G Series integrates an analog comparator (CMP), and CMP interrupts can wake the device from STOP Mode. It can be applied to alarm circuits, power supply voltage monitoring circuits, zero-cross detection circuits, and similar applications.

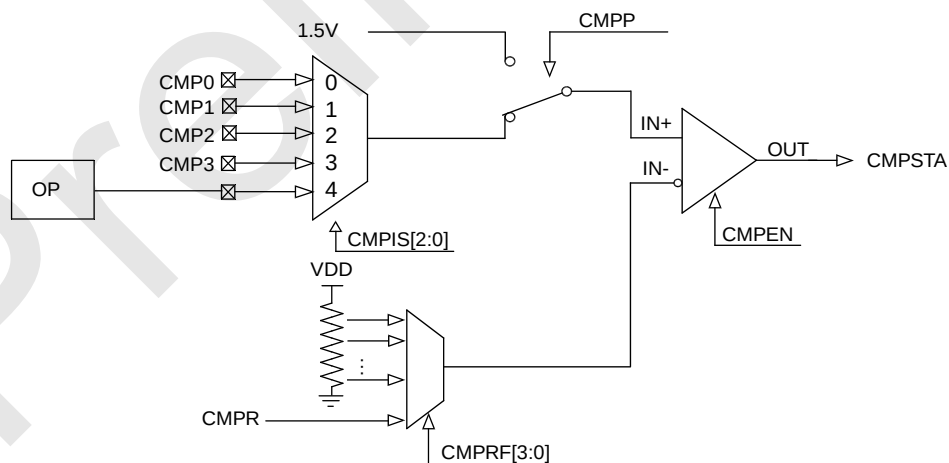
This comparator has five analog positive input terminals: CMP0 through CMP3, and the OP output terminal, selectable via CMPIS[2:0]. The negative input voltage can be switched by CMPRF[3:0] to either the external voltage at the CMPP pin or one of the 15 internal reference voltages.

The comparator interrupt mode can be conveniently set via CMPIM[1:0]. When the interrupt condition defined by CMPIM[1:0] occurs, the comparator interrupt flag CMPIF is set to 1 and must be cleared by software.

11.1 Feature

- Positive input signal selectable from five options:
 - Four analog signal positive input pins CMP0 to CMP3
 - OP output terminal
- The negative input terminal voltage can be selected from the CMPP pin or one of 15 internal V_{DD} voltage division reference levels.
- CMP Interrupts can wake up STOP Mode

11.2 Analog comparator block diagram



Analog comparator block diagram

11.3 CMP register

11.4 CMP related registers table

11.4.1 Analog comparator status register (CMP_STS)

Register	Read/Write	Description	Reset Value
CMP_STS	R/W	Analog Comparator Status 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
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	-	-	CMPSTA	CMPIF

Bit Number	Bit Symbol	Description
1	CMPSTA	Analog Comparator Output Status Bit 0: Comparator positive input (V+) voltage is less than negative input (V-) 1: Comparator positive input (V+) voltage is greater than negative input (V-)
0	CMPIF	Analog Comparator Interrupt Flag Bit This bit is set by hardware and cleared by writing 1 in software. 0: Comparator interrupt not triggered 1: Set by hardware when the comparator meets interrupt trigger conditionsA comparator interrupt is generated if CMPIM[1:0] ≠ 00
31 ~ 2	-	Reserved

11.4.2 Analog comparator configuration register (CMP_CFG)

Register	Read/Write	Description	Reset Value
CMP_CFG	R/W	Analog Comparator 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
-	-	-	-	CMPRF[3:0]			
7	6	5	4	3	2	1	0
CMPEN	CMPIM[1:0]		CMPP	-	CMPIS[2:0]		

Bit Number	Bit Symbol	Description
11 ~ 8	CMPRF[3:0]	Analog Comparator Negative Input Voltage Reference Selection Bit Analog Comparator negative input voltage reference options are as follows: 0000:CMPR 0001:1/16 V _{DD}

Bit Number	Bit Symbol	Description
		0010:2/16 V _{DD} 0011:3/16 V _{DD} 0100:4/16 V _{DD} 0101:5/16 V _{DD} 0110:6/16 V _{DD} 0111:7/16 V _{DD} 1000:8/16 V _{DD} 1001:9/16 V _{DD} 1010:10/16 V _{DD} 1011:11/16 V _{DD} 1100:12/16 V _{DD} 1101:13/16 V _{DD} 1110:14/16 V _{DD} 1111:15/16 V _{DD}
7	CMPEN	Analog Comparator Enable Bit 0: Disable Analog Comparator 1: Enable Analog Comparator
6 ~ 5	CMPIM[1:0]	Analog Comparator Interrupt Mode Selection Bits 00: No Interrupt 01: Rising Edge Interrupt: Triggered when IN+ crosses from below IN- to above IN- 10: Falling Edge Interrupt: Triggered when IN+ crosses from above IN- to below IN- 11: Both Edge Interrupt: Triggered on either rising or falling edge crossing between IN+ and IN-
4	CMPP	Analog Comparator Positive Input Selection: 0: Positive input is selected from CMP0 to CMP3, configured by CMPIS[1:0] 1: Positive input is the internal 1.5 V reference voltage
2 ~ 0	CMPIS[2:0]	Analog Comparator Positive Input Channel Selection Bit This bit is invalid when CMPP = 1: 000: Select CMP0 as analog comparator positive input 001: Select CMP1 as analog comparator positive input 010: Select CMP2 as analog comparator positive input 011: Select CMP3 as analog comparator positive input 100 to 110: Not used 111: Select PGA output as analog comparator input
31 ~ 12 3	-	Reserved

11.5 CMP register mapping

Register	Offset Address	Read/Write	Description	Reset Value
CMP Base Address: 0x40022130				
CMP_STS	0x00	R/W	Analog Comparator Status Register	0x0000_0000
CMP_CFG	0x04	R/W	Analog Comparator Configuration Register	0x0000_0000

12 UART0 to 5

12.1 Clock source

- All UARTs in the SC32F11T/11G Series share a single clock source derived from PCLK.

12.2 Feature

- 6 UARTs: UART0 through 5.
- UART0 through UART5 support signal port mapping and can be mapped to another IO group.
- Independent Baud Rate Generator.
- Four selectable communication modes.
- SM0 and SM1 Serial Communication Mode Control Bits
 - Mode 0: 8-bit half-duplex synchronous communication mode, transmitting and receiving serial data on the RXD pin. The TXD pin serves as the transmit shift clock. Each frame consists of 8 bits, with the least significant 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, with variable communication Baud Rate.
 - Mode 2: Reserved
 - Mode 3: 11-bit full-duplex Asynchronous Communication, consisting of 1 Start bit, 8 data bits, one programmable 9th bit, and 1 Stop bit, with variable communication Baud Rate.
- Transmit and receive completion can generate interrupts and set the corresponding flags TXIF and RXIF. Interrupt flags must be cleared by software.
- UART0 and UART1 can generate DMA Requests.
- UART2 to 5 cannot generate DMA Requests.
- Independent Baud Rate Generator.
- UART0/1/2/3/4/5 support Wake up from STOP Mode:
 - START bit Falling Edge can Wake up STOP Mode.
 - Provides wake up Interrupt Enable bit WKIE and wake up Interrupt Flag WKIF

12.3 UART interrupt

For UARTn (n=0 to 5), an interrupt is generated on “Wake up” or “Data transmit/receive complete” events. Individual interrupt enable bits improve flexibility.

Interrupt Event	Interrupt Request Control Bit	Event Flag	Interrupt Enable Sub-Bit
UART Wake-up from STOP Mode	UARTn_IDE ->INTEN	WKIF	WKIE
Data Transmit Complete		TXIF	TXIE
Data Receive Complete		RXIF	RXIE

12.4 UART0/1/2/3/4/5 registers

12.4.1 Register table for UART0/1/2/3/4/5

12.4.1.1 UART control register UARTn_CON

Register	Read/Write	Description	Reset Value
UARTn_CON (n=0/1/2/3/4/5)	R/W	UART 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
-	-	SPOS	-	-	-	-	-
7	6	5	4	3	2	1	0
TXEN	RXEN	-	PRESCALER	-	SM2	SM[1:0]	

Bit Number	Bit Symbol	Description																																																																								
13	SPOS	- UART0 Signal Port Mapping Control Bit @UART0_CON <table> <tr> <th>Signal</th><th>RXD0</th><th>TXD0</th></tr> <tr> <td>SPOS Value</td><td></td><td></td></tr> <tr> <td>SPOS =0</td><td>PB12</td><td>PB11</td></tr> <tr> <td>SPOS =1</td><td>PA3</td><td>PA2</td></tr> </table> - UART1 Signal Port Mapping Control Bit @UART1_CON <table> <tr> <th>Signal</th><th>RXD1</th><th>TXD1</th></tr> <tr> <td>SPOS Value</td><td></td><td></td></tr> <tr> <td>SPOS =0</td><td>PD4</td><td>PD5</td></tr> <tr> <td>SPOS =1</td><td>PC10</td><td>PC11</td></tr> </table> - UART2 Signal Port Mapping Control Bit @UART2_CON <table> <tr> <th>Signal</th><th>RXD2</th><th>TXD2</th></tr> <tr> <td>SPOS Value</td><td></td><td></td></tr> <tr> <td>SPOS =0</td><td>PB9</td><td>PB10</td></tr> <tr> <td>SPOS =1</td><td>PA6</td><td>PA5</td></tr> </table> - UART3 Signal Port Mapping Control Bit @UART3_CON <table> <tr> <th>Signal</th><th>RXD3</th><th>TXD3</th></tr> <tr> <td>SPOS Value</td><td></td><td></td></tr> <tr> <td>SPOS =0</td><td>PC12</td><td>PC13</td></tr> <tr> <td>SPOS =1</td><td>PC5</td><td>PC6</td></tr> </table> - UART4 Signal Port Mapping Control Bit @UART4_CON <table> <tr> <th>Signal</th><th>RXD4</th><th>TXD4</th></tr> <tr> <td>SPOS Value</td><td></td><td></td></tr> <tr> <td>SPOS =0</td><td>PB7</td><td>PB6</td></tr> <tr> <td>SPOS =1</td><td>PC3</td><td>PC4</td></tr> </table> - UART5 Signal Port Mapping Control Bit @UART5_CON <table> <tr> <th>Signal</th><th>RXD5</th><th>TXD5</th></tr> <tr> <td>SPOS Value</td><td></td><td></td></tr> <tr> <td>SPOS =0</td><td>PA0</td><td>PA1</td></tr> <tr> <td>SPOS =1</td><td>PA13</td><td>PA14</td></tr> </table>	Signal	RXD0	TXD0	SPOS Value			SPOS =0	PB12	PB11	SPOS =1	PA3	PA2	Signal	RXD1	TXD1	SPOS Value			SPOS =0	PD4	PD5	SPOS =1	PC10	PC11	Signal	RXD2	TXD2	SPOS Value			SPOS =0	PB9	PB10	SPOS =1	PA6	PA5	Signal	RXD3	TXD3	SPOS Value			SPOS =0	PC12	PC13	SPOS =1	PC5	PC6	Signal	RXD4	TXD4	SPOS Value			SPOS =0	PB7	PB6	SPOS =1	PC3	PC4	Signal	RXD5	TXD5	SPOS Value			SPOS =0	PA0	PA1	SPOS =1	PA13	PA14
Signal	RXD0	TXD0																																																																								
SPOS Value																																																																										
SPOS =0	PB12	PB11																																																																								
SPOS =1	PA3	PA2																																																																								
Signal	RXD1	TXD1																																																																								
SPOS Value																																																																										
SPOS =0	PD4	PD5																																																																								
SPOS =1	PC10	PC11																																																																								
Signal	RXD2	TXD2																																																																								
SPOS Value																																																																										
SPOS =0	PB9	PB10																																																																								
SPOS =1	PA6	PA5																																																																								
Signal	RXD3	TXD3																																																																								
SPOS Value																																																																										
SPOS =0	PC12	PC13																																																																								
SPOS =1	PC5	PC6																																																																								
Signal	RXD4	TXD4																																																																								
SPOS Value																																																																										
SPOS =0	PB7	PB6																																																																								
SPOS =1	PC3	PC4																																																																								
Signal	RXD5	TXD5																																																																								
SPOS Value																																																																										
SPOS =0	PA0	PA1																																																																								
SPOS =1	PA13	PA14																																																																								

Bit Number	Bit Symbol	Description
7	TXEN	UART Transmit Enable Control Bit 0: Transmit disabled; TXD signal does not affect the pin state. If transmit is disabled in user code while receiving is enabled, other functions multiplexed on the TXD pin remain unaffected. 1: Transmit enabled; the pin operates in TXD signal port mode.
6	RXEN	UART Receive Enable Control Bit 0: Receive disabled 1: Receive enabled
4	PRESCALER	Baud Rate Multiplier Bit This bit is defined differently depending on the UART mode: - When SM0 to 1 = 01 (UART Mode 1) or SM0 to 1 = 11 (UART Mode 3): 0: Serial port runs at system clock divided by 1 1: Serial port runs at system clock divided by 16 - When SM0 to 1 = 00 (UART Mode 0), the Baud Rate Multiplier Bit is: 0: Serial port runs at system clock divided by 12 1: Serial port runs at system clock divided by 4
2	SM2	RB8 Set Interrupt Enable Bit This bit is valid only in Mode 3 0: RI is set to generate an interrupt request for each complete data frame received 1: RI is set to generate an interrupt request only if RB8=1 when a complete data frame is received.
1 ~ 0	SM[1:0]	UART Communication Mode Control Bits 00: Mode 0, 8-bit Half-Duplex Synchronous Communication, transmitting and receiving serial data on the RXD pin. The TXD pin serves as the transmit shift clock. Each frame transmits or receives 8 bits, LSB first. When RXEN is enabled in this mode, UART outputs a complete frame clock and sets RXIF to 1. 01: Mode 1, 10-bit Full-Duplex Asynchronous Communication, consisting of 1 Start bit, 8 data bits, and 1 Stop bit; Baud Rate is configurable. 10: Reserved. 11: Mode 3, 11-bit Full-Duplex Asynchronous Communication, consisting of 1 Start bit, 8 data bits, one programmable 9th bit, and 1 Stop bit; Baud Rate is configurable.
31 ~ 14 12 ~ 8 5 3	-	Reserved

12.4.1.2 UART flag status register UARTn_STS

Register	Read/Write	Description	Reset Value
UARTn_STS (n=0/1/2/3/4/5)	R/W	UART Flag Status 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
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	WKIF	-	-	TXIF	RXIF

Bit Number	Bit Symbol	Description
4	WKIF	UART Wake-up Flag After UART wakes up from STOP mode, this bit is set by hardware. If WKIE = 1, an interrupt is generated. This bit is cleared by software writing 1.
1	TXIF	Transmit Interrupt Flag This bit is set by hardware upon completion of data transmission. If TXIE = 1, an interrupt is generated. This bit is cleared by software writing 1.
0	RXIF	Receive Interrupt Flag This bit is set by hardware upon completion of data reception. If RXIE = 1, an interrupt is generated. This bit is cleared by software writing 1.
31 ~ 5 3 ~ 2	-	Reserved

12.4.1.3 UART baud rate configuration register UARTn_BAUD

Register	Read/Write	Description	Reset Value
UARTn_BAUD (n=0/1/2/3/4/5)	R/W	UART Baud Rate 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 Symbol	Description
15 ~ 0	BAUD[15:0]	UART Baud Rate Configuration Bits After writing BAUD[15:0], the UART baud rate is set by the following formula: $\text{BaudRate} = f_{\text{UART}} / \text{BAUD}[15:0]$ Where f_{UART} is the UART clock frequency after prescaling; see the PRESCALER bit description. Note: BAUD[15:0] must be greater than 0x0010.
31 ~ 16	-	Reserved

12.4.1.4 UART data register UARTn_DATA

Register	Read/Write	Description	Reset Value
UARTn_DATA (n=0/1/2/3/4/5)	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 Symbol	Description
8	SBUF8	The 9th data bit for UART transmit/receive This bit is valid only in Mode 3.
7 ~ 0	SBUF[7:0]	UART Data Buffer Read: returns the content of the receive latch Write: data in SBUF is loaded into the transmit shift register and initiates transmission
31 ~ 9	-	Reserved

12.4.1.5 UART interrupt enable and DMA control register UARTn_IDE

Register	Read/Write	Description	Reset Value
UARTn_IDE (n=0/1/2/3/4/5)	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
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
TXDMAEN	RXDMAEN	-	WKIE	-	TXIE	RXIE	INTEN

Bit Number	Bit Symbol	Description
7	TXDMAEN	DMA Transmit Channel Enable 0: Disable DMA Transmit 1: Enable DMA Transmit Enabling this bit sets TXIF to trigger a DMA channel transmit request Note: 1. UART0 and UART1 can generate DMA Requests. 2. UART2/3/4/5 do not generate DMA requests
6	RXDMAEN	DMA receive channel enable bit 0: Disable DMA receive function 1: Enable DMA receive function This bit enables RXIF to trigger DMA channel receive requests Note: 1. UART0 and UART1 can generate DMA Requests. 2. UART2/3/4/5 do not generate DMA requests
4	WKIE	UART wake-up interrupt enable bit 0: No interrupt generated when WKIF is set 1: Interrupt generated when WKIF is set
2	TXIE	UART transmit interrupt enable bit 0: No interrupt generated when TXIF is set 1: Interrupt generated when TXIF is set
1	RXIE	UART receive interrupt enable bit 0: No interrupt generated when RXIF is set 1: Interrupt generated when RXIF is set
0	INTEN	CPU Interrupt Request Enable Control Bit 0: Disable interrupt request 1: Enable interrupt request
31 ~ 8 5 3	-	Reserved

12.4.2 UART0/1/2/3/4/5 register mapping

Register	Offset Address	Read/Write	Description	Reset Value	Access Restriction
UART0 Base Address: 0x4002_0020					
UART0_CON	0x00	R/W	UART Control Register	0x0000_0000	-
UART0_STS	0x04	R/W	UART Flag Register	0x0000_0000	-
UART0_BAUD	0x08	R/W	UART Baud Rate Register	0x0000_0000	-
UART0_DATA	0x0C	R/W	UART Data Register	0x0000_0000	Byte/Half-word Access Not Supported
UART0_IDE	0x10	R/W	Communication DMA Control Register	0x0000_0000	-

Register	Offset Address	Read/Write	Description	Reset Value	Access Restriction
UART1 Base Address: 0x4002_0080					
UART1_CON	0x00	R/W	UART Control Register	0x0000_0000	-
UART1_STS	0x04	R/W	UART Flag Register	0x0000_0000	-
UART1_BAUD	0x08	R/W	UART Baud Rate Register	0x0000_0000	-
UART1_DATA	0x0C	R/W	UART Data Register	0x0000_0000	Byte/Half-word Access Not Supported
UART1_IDE	0x10	R/W	Communication DMA Control Register	0x0000_0000	-

Register	Offset Address	Read/Write	Description	Reset Value	Access Restriction
UART2 Base Address: 0x4002_1020					
UART2_CON	0x00	R/W	UART Control Register	0x0000_0000	-
UART2_STS	0x04	R/W	UART Flag Register	0x0000_0000	-
UART2_BAUD	0x08	R/W	UART Baud Rate Register	0x0000_0000	-
UART2_DATA	0x0C	R/W	UART Data Register	0x0000_0000	Byte/Half-word Access Not Supported
UART2_IDE	0x10	R/W	Communication DMA Control Register	0x0000_0000	-

Register	Offset Address	Read/Write	Description	Reset Value	Access Restriction
UART3 Base Address: 0x4002_2020					

Register	Offset Address	Read/Write	Description	Reset Value	Access Restriction
UART3_CON	0x00	R/W	UART Control Register	0x0000_0000	-
UART3_STS	0x04	R/W	UART Flag Register	0x0000_0000	-
UART3_BAUD	0x08	R/W	UART Baud Rate Register	0x0000_0000	-
UART3_DATA	0x0C	R/W	UART Data Register	0x0000_0000	Byte/Half-word Access Not Supported
UART3_IDE	0x10	R/W	Communication DMA Control Register	0x0000_0000	-

Register	Offset Address	Read/Write	Description	Reset Value	Access Restriction
UART4 Base Address: 0x4002_1080					
UART4_CON	0x00	R/W	UART Control Register	0x0000_0000	-
UART4_STS	0x04	R/W	UART Flag Register	0x0000_0000	-
UART4_BAUD	0x08	R/W	UART Baud Rate Register	0x0000_0000	-
UART4_DATA	0x0C	R/W	UART Data Register	0x0000_0000	Byte/Half-word Access Not Supported
UART4_IDE	0x10	R/W	Communication DMA Control Register	0x0000_0000	-

Register	Offset Address	Read/Write	Description	Reset Value	Access Restriction
UART5 Base Address: 0x4002_00A0					
UART5_CON	0x00	R/W	UART Control Register	0x0000_0000	-
UART5_STS	0x04	R/W	UART Flag Register	0x0000_0000	-
UART5_BAUD	0x08	R/W	UART Baud Rate Register	0x0000_0000	-
UART5_DATA	0x0C	R/W	UART Data Register	0x0000_0000	Byte/Half-word Access Not Supported
UART5_IDE	0x10	R/W	Communication DMA Control Register	0x0000_0000	-

13 QSPI0 to 1

13.1 Clock source

The SPI clock source in the SC32F11T/11G Series is single and derived from PCLK.

13.2 QSPI0/1 feature

- Shares register addresses and pins with SPI0/1 and TWI0/1, but functions operate independently.
- QSPI supports Master Mode only.
- Each QSPI supports four transfer widths (8/16/24/32 bits) with a fixed 8-level FIFO per transfer width, featuring independent TX and RX.
- Supports Single-, Dual-, and Quad-Wire communication.
- Supports two modes: QSPI half-duplex communication and pass-through.
- Communication speed up to 32 Mbps.
- Both QSPI units support direct pass-through without FIFO.
- Supports DMA.

13.3 Pin description

Signal Name	Signal Type	Description
QSPI0/1 CLK	Digital Output	QSPI0/1 Clock Output
QSPI0/1 IO0	Digital Input/Output	In dual-wire and four-wire modes, it operates as bidirectional IO; in single-wire mode, it serves as a serial output.
QSPI0/1 IO1	Digital Input/Output	In dual-wire and four-wire modes, it operates as bidirectional IO; in single-wire mode, it serves as a serial input.
QSPI0/1 IO2	Digital Input/Output	In four-wire mode, it operates as bidirectional IO.
QSPI0/1 IO3	Digital Input/Output	In four-wire mode, it operates as bidirectional IO.
QSPI0/1 CS	Digital Output	Software-controlled chip select signal output; users must configure the pin via software.

13.4 Command sequence

QUADSPI communicates with FLASH through commands, each comprising instruction, address, alternate bytes, dummy cycles, and data phases. Any of these five phases can be omitted, but each command must include at least one of instruction, address, alternate bytes, or data. The user must drive the CS signal low before each data transfer and drive it high after completion. On the SC32F11T/11G, CS must be implemented via software.

- **Instruction phase**

During the instruction phase, the transfer width is set to 8 bits via the DWIDTH[1:0] field in the

TWIn_QSPIn_CON register ($n = 0$ to 1). An 8-bit instruction is then written to the TWIn_QSPIn_DATA register ($n = 0$ to 1) and transmitted to the FLASH to specify the operation type.

Most FLASH devices receive instructions in single-line mode; however, QSPI can send instructions in dual-line or quad-line mode during the instruction phase. This is configured through the LMODE[1:0] field in the TWIn_QSPIn_CON register ($n = 0$ to 1).

If this is not configured, the instruction phase is skipped, and the command sequence begins at the address phase, if present.

- Address phase

In the address phase, users configure the number of address bytes to transmit via the DWIDTH[1:0] field in TWIn_QSPIn_CON ($n = 0$ to 1), then write these bytes into the TWIn_QSPIn_DATA ($n = 0$ to 1) register. This transmits 1 to 4 bytes to the FLASH, specifying the operation address.

Users configure the transmission line mode for the address phase-single, dual, or quad wire-via the LMODE[1:0] field in TWIn_QSPIn_CON ($n = 0$ to 1).

If not configured, the address phase is skipped, and the command sequence advances directly to the next phase, if any.

- Alternate byte phase

In the alternate byte phase, users configure the number of bytes to transmit via DWIDTH[1:0] in TWIn_QSPIn_CON ($n = 0$ to 1) and write them into TWIn_QSPIn_DATA ($n = 0$ to 1). This sends 1 to 4 bytes to the FLASH, controlling the operation mode.

The user configures the alternate byte phase transmission line mode as single-line, dual-line, or quad-line via the LMODE[1:0] field in the TWIn_QSPIn_CON ($n = 0$ to 1) register.

If not configured, the alternate byte phase is skipped, and the command sequence proceeds directly to the next phase, if present.

- Idle instruction cycle phase

During the idle instruction cycle phase, the user must configure a number of cycles during which no data is transmitted or received. This provides the FLASH with preparation time when operating at higher clock frequencies.

In the idle instruction cycle phase, set the LMODE[1:0] field in the TWIn_QSPIn_CON ($n = 0$ to 1) register to 00 and configure the number of cycles via the TWIn_QSPIn_DATA ($n = 0$ to 1) register. Writing 1 byte of data corresponds to transmitting two clock cycles.

If the above configuration is not performed, the empty command period stage will be skipped, and the command sequence will proceed directly to the data stage (if present).

Note: When receiving data, to ensure sufficient conversion time for the data signal to switch from Output Mode to Input Mode, at least one empty command period is required when receiving data from FLASH in dual-line and quad-line modes.

At least one empty command period is required when receiving data from FLASH in dual-line and quad-line modes.

- Data Stage

During the data segment, any number of bytes can be received from or transmitted to FLASH.

In write mode (WR=0), the number of bytes to be transmitted is determined by how many data registers are written, and transmission is triggered immediately after writing to the data registers.

In read mode (WR=1), the number of bytes to be received can be configured via the TWIn_QSPIn_DL (n = 0 to 1) register; after enabling clock-only mode (CLKONLY=1), the clock triggers immediately.

In read-write mode operation, the bit width of data for transmit and receive is configured by the DWIDTH[1:0] field of the TWIn_QSPIn_CON register (n = 0 to 1).

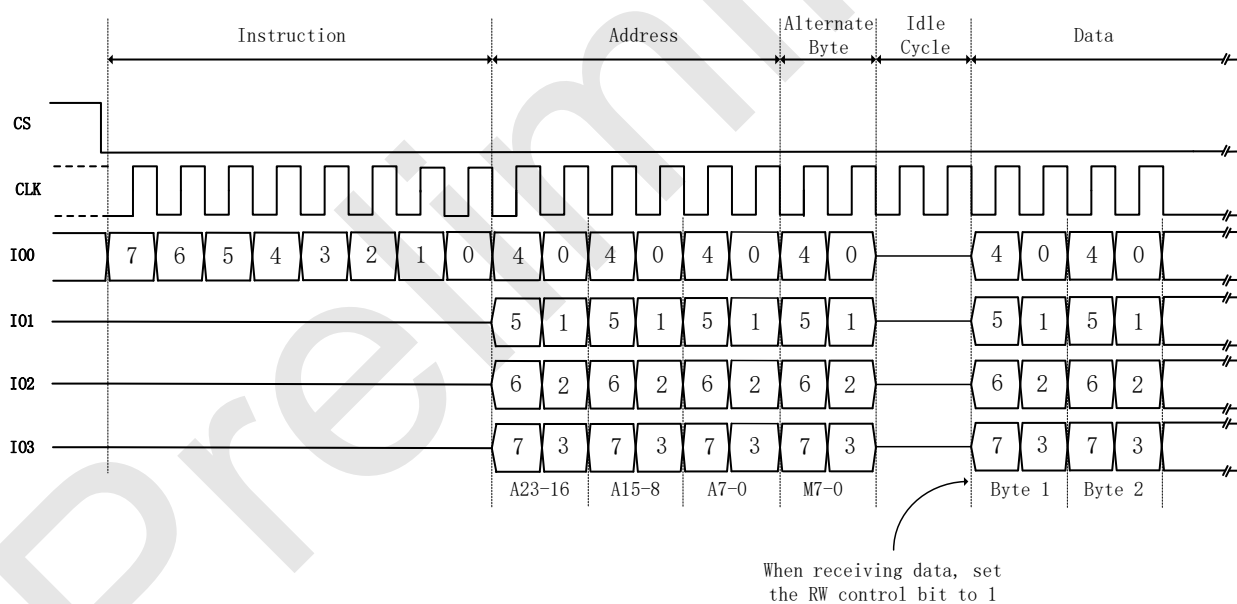
Users can configure the data phase transfer line mode as single-line, dual-line, or quad-line via the LMODE[1:0] field in the TWIn_QSPIn_CON register (n = 0 to 1).

If the above configuration is not set, the data phase is skipped and the command sequence completes as soon as CS is released high.

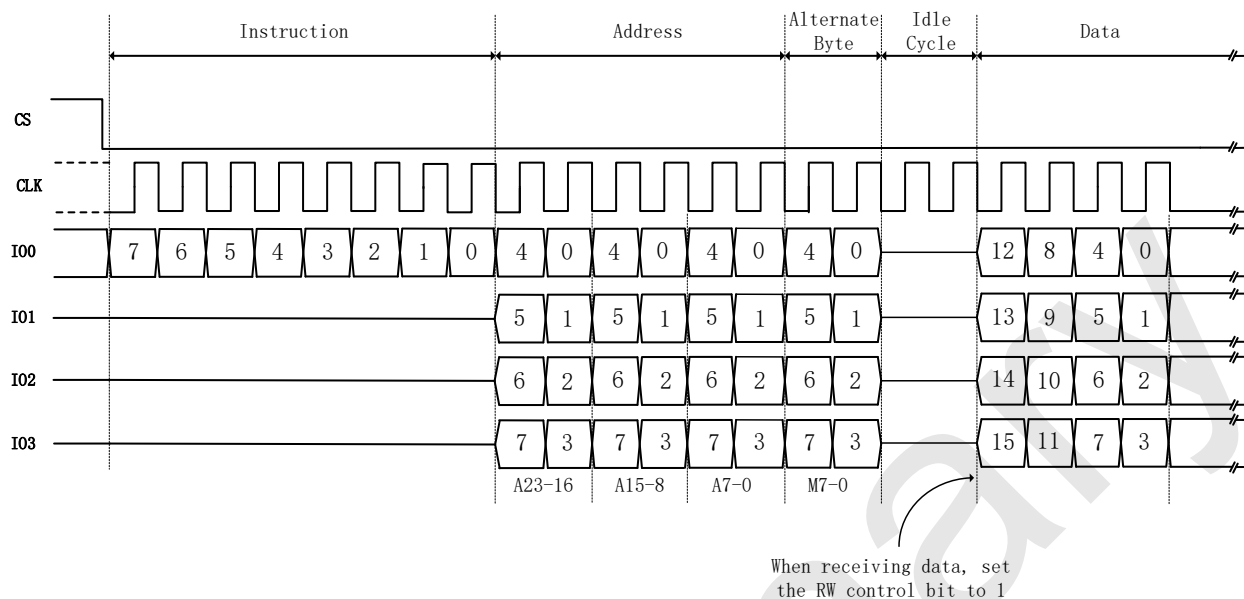
- Phase complete

Users can monitor the BUSY status bit in the TWIn_QSPIn_STS register (n = 0 to 1) to determine whether each phase has finished transmitting, then proceed with the next phase's transmission and configuration.

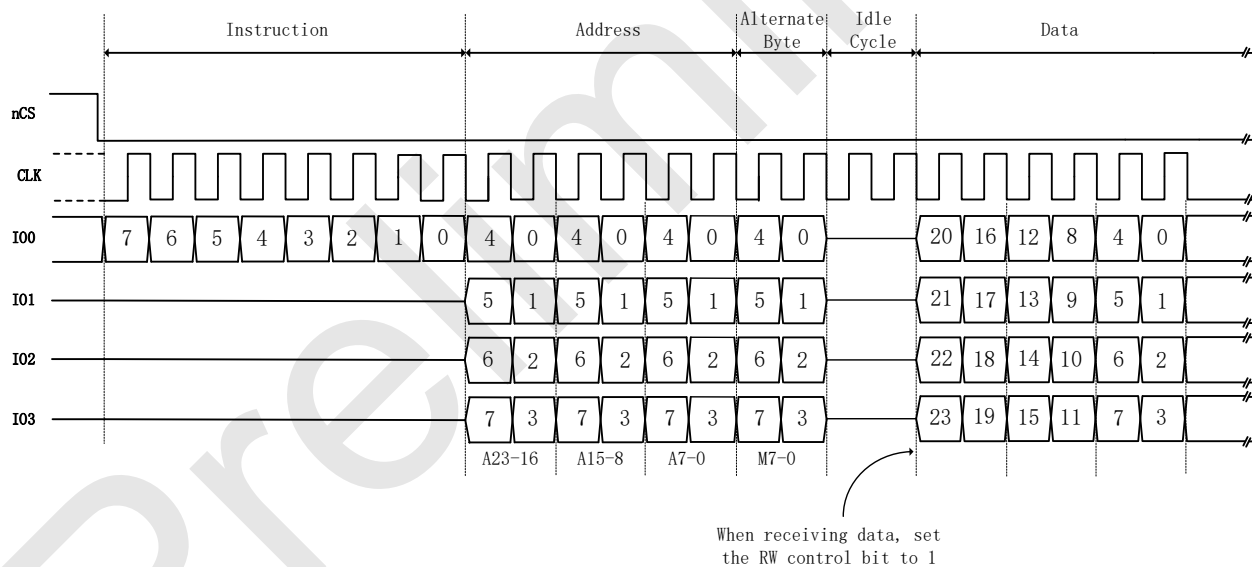
13.4.1 8-bit quad-line waveform diagram



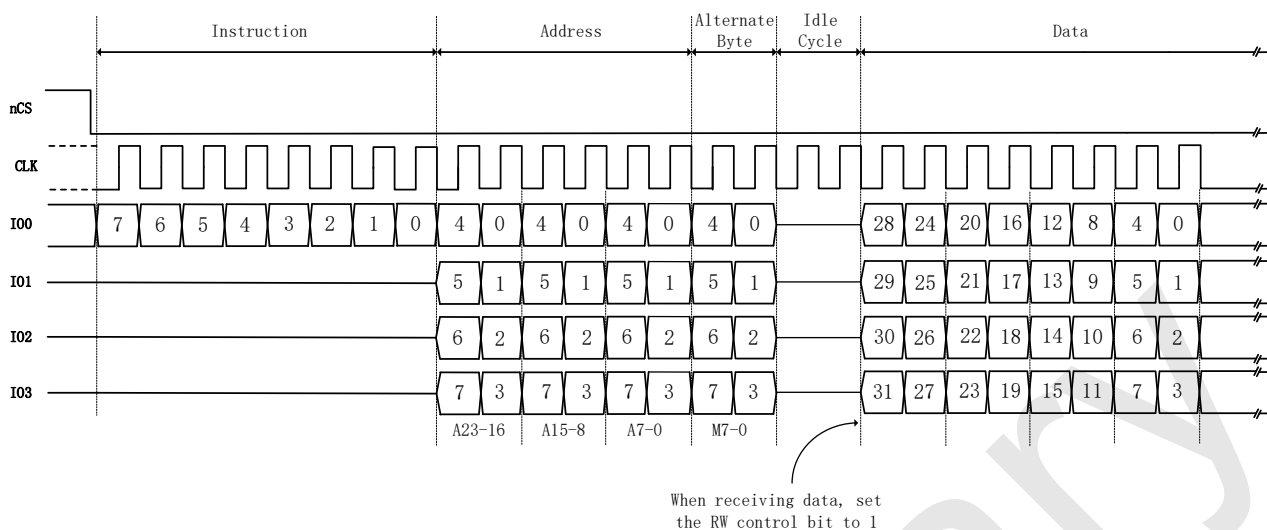
13.4.2 16-bit four-wire waveform



13.4.3 24-bit four-wire waveform



13.4.4 32-bit four-wire waveform



13.5 Transmission line mode

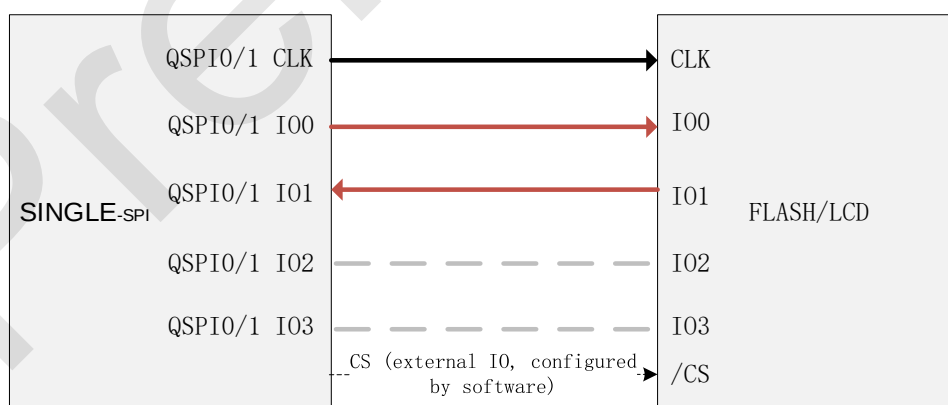
The user configures the transmission line mode for each phase via the LMODE field in the $TWIn_QSPIn_CON$ register ($n=0$ to 1).

Note: The dummy instruction phase always operates in 4-wire mode, writing 1 byte and lasting 2 clock cycles.

13.5.1 Single-wire transmission mode

In single-wire mode:

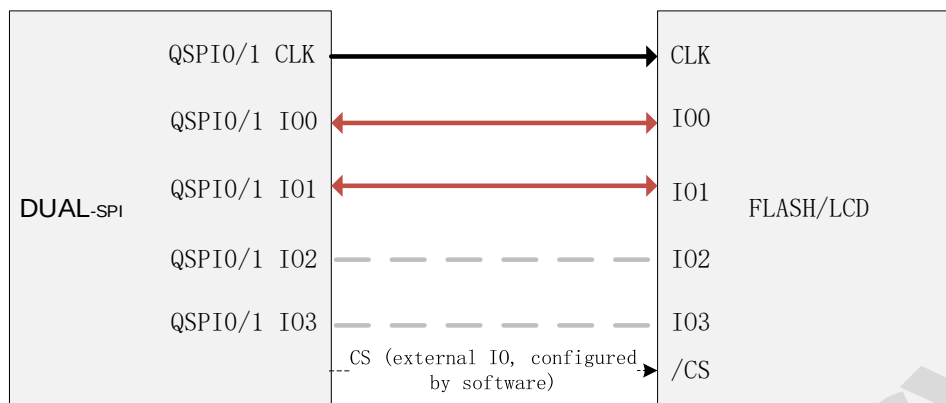
- $IO0$ is set to output mode
- $IO1$ is set to input mode (high impedance)
- $IO2$ and $IO3$ are in high impedance mode



13.5.2 Dual-wire transmission mode

In dual-wire mode:

- During data read phases, $IO0$ and $IO1$ are in high impedance (input); in all other phases, they operate as outputs.
- $IO2/3$ are in high-impedance mode.

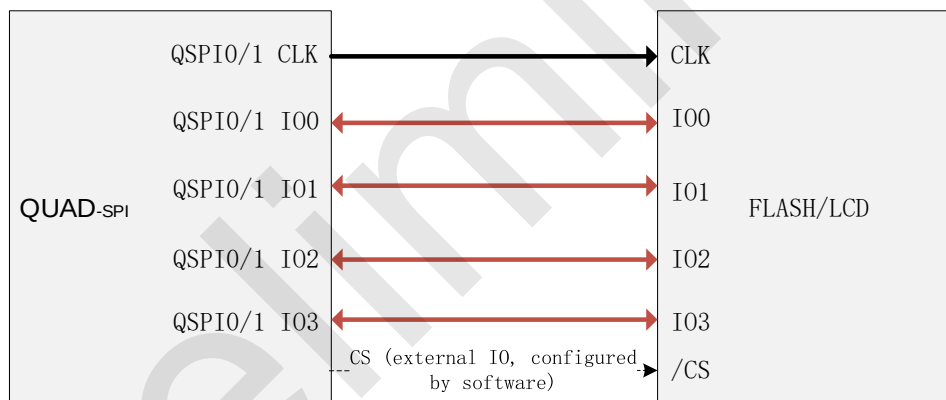


13.5.3 Four-wire transmission mode

In four-wire mode:

- During data phase reads, IO0/IO1/IO2/IO3 remain in high-impedance (input); otherwise, they operate as outputs.

Note: IO2 and IO3 are only used in four-wire mode. If four-wire mode is unused in all phases, these pins can be assigned other functions even when the QSPI module is enabled.



13.6 Bypass mode

QSPIn (n = 0 to 1) supports an internal bypass mode, enabling one QSPI to read FLASH data while the other drives a display, allowing direct data transfer to improve efficiency.

Setting TWI0_QSPI0_CON and TWI1_QSPI1_CON with MODE[1:0] = 11 selects QSPI0/1 to operate in pass-through mode.

To read FLASH via QSPI, select internal transfer direction, enable internal transfer, and disable internal receive (STRDIR=0, STRTEN=1, STRREN=0).

For driving the screen via QSPI, select internal receive direction, enable internal receive, and disable internal transfer (STRDIR=1, STRREN=1, STRTEN=0).

Other communication configurations follow QSPI half-duplex mode. Once configured, data read from FLASH is directly transmitted for screen driving, with a delay of about one frame (or slightly more than 8 clocks for 32-bit transfers).

13.7 QSPI configuration method

QUADSPI Configuration Consists of Two Stages

- General Communication Configuration

Configure the communication clock rate by setting the clock prescaler QTWCK[3:0] in the TWIn_QSPIn_CON register (n = 0 to 1). Set the communication mode using CPOL and CPHA.

Note: Most FLASH devices support only modes 0 and 3 (CPOL=0, CPHA=0 or CPOL=1, CPHA=1). QSPI supports MSB-first transmission only (DORD=0) and operates exclusively in Master Mode (SMSTR=1).

- QSPI Communication Configuration

After completing general communication configuration, select operation in half-duplex QSPI mode or passthrough mode. Configure read/write mode, transfer line mode, transfer width, and internal passthrough direction according to application requirements.

13.7.1 Half-duplex QSPI mode configuration procedure

Set MODE[1:0] bits of TWIn_QSPIn_CON (n=0 to 1) to 10 to select half-duplex QSPI mode. Set RW to 0 to select write mode and transmit data to FLASH or QSPI-compatible displays; Set RW to 1 to select read mode and read data from FLASH.

In half-duplex QSPI mode, communication with FLASH follows these steps:

1. Clear TWIn_QSPIn_CON before configuration. Note that LMODE[1:0] and DWIDTH[1:0] must be cleared before each setup
2. Set RW read/write mode in TWIn_QSPIn_CON
3. Set LMODE[1:0] transmission line mode and DWIDTH[1:0] data width in TWIn_QSPIn_CON
4. In read mode, specify the number of data to read in TWIn_QSPIn_DL
5. Drive the CS signal low
6. In read mode, enable CLKONLY to output the clock and trigger clock output immediately
7. In write mode, write transmit data to TWIn_QSPIn_DATA and trigger transmission immediately
8. Check the BUSY flag to confirm whether the current stage transmission is complete
9. In read mode, receive data by reading TWIn_QSPIn_DATA
10. After confirming transmission completion, repeat steps 1 to 9 to configure the next stage
11. Drive the CS signal high after the current command completes.

13.7.2 Pass-through mode configuration procedure

Configure MODE[1:0] bits of TWIn_QSPIn_CON (n=0 to 1) to 11 to select QSPI pass-through mode.

When QUADSPI0 and QUADSPI1 operate in pass-through mode, FLASH communicates with the display supporting QSPI as follows (the following description uses LCD as an example):

1. Initialize QSPI0 and QSPI1
2. Initialize the LCD in half-duplex QSPI mode according to screen requirements. Refer to the half-duplex QSPI mode configuration steps
3. Pull the QSPI chip select CS0 low to communicate with the LCD
4. Set the LCD to wait for data transmission from the controller
5. Configure the QSPI connected to the LCD for pass-through mode by selecting internal receive direction

- (STRDIR=1) and enabling internal reception (STRREN=1) to wait for data from FLASH
6. Pull the QSPI chip select CS1 low to communicate with the FLASH
 7. Configure FLASH read operations in half-duplex QSPI mode according to FLASH specifications. Before the data read phase, transmit all preceding phases (instruction, address, alternate bytes, dummy cycles) following the half-duplex QSPI configuration steps.
 8. Set the QSPI connected to FLASH to passthrough mode, select internal transfer direction (STRDIR=0), and enable internal transfer (STRTEN=1).
 9. Data read from FLASH is directly transmitted to the LCD via passthrough. The transmitted data length is determined by the value in the FLASH-connected QSPI TWIn_QSPIn_DL register.
 10. After the FLASH read completes, set CS1 high.
 11. After the LCD driver finishes, set CS0 high.

Note: In pass-through mode, both QSPIs must be set to the same line transmission mode LMODE[1:0] and transmission width DWIDTH[1:0].

13.8 QSPI interrupts

For QSPIn (n=0 to 1), interrupts are generated when data transfer completes or FIFO changes state. Separate interrupt enable bits provide enhanced flexibility.

Interrupt Event	Interrupt Request Control Bit	Event Flag	Interrupt Enable Sub-Bit
One frame of data transmission/reception complete	TWIn_QSPIn_IDE ->INTEN	QTWIF	TXIE
Receive FIFO valid data exceeds half full		RXHIF	RXHIE
Transmit FIFO valid data less than half full		TXHIF	TXHIE
Receive FIFO full		RXFIF	RXIE
Transmit FIFO empty		TXEIF	TBIE
Receive FIFO not empty		RXNEIF	RXNEIE
TWIn_QSPIn_DL counter underflow		DLUFIF	DLUFIE

13.9 QSPI0/1 registers

13.9.1 QSPI0/1 related register table

13.9.1.1 TWI_QSPI control register TWIn_QSPIn_CON

Register	Read/Write	Description	Reset Value
TWIn_QSPIn_CON (n=0/1)	R/W	TWIN_QSPI Control Register	0x0000_0000

31	30	29	28	27	26	25	24
LMODE[1:0]		DWIDTH[1:0]		SSHIFT	STRREN	STRTEN	STRDIR
23	22	21	20	19	18	17	16
MODE[1:0]		RW	CPOL	CPHA	DORD	CLKONLY	SMSTR
15	14	13	12	11	10	9	8
-	-	-	-	QTWCK[3:0]			

7	6	5	4	3	2	1	0
QTWEN	-	START	STOP	-	-	AA	STRETCH

Bit Number	Bit Symbol	Description
31 ~ 30	LMODE[1:0]	QSPI Transfer Mode Select Bit 00: No Data 01: Single-Line Transfer 10: Dual-Line Transfer 11: Quad-Line Transfer
29 ~ 28	DWIDTH[1:0]	Full-Duplex SPI and Half-Duplex QSPI Interface Transfer Width Select Bit 00: 8 Bits 01: 16 Bits 10: 24 Bits 11: 32 Bits
27	SSHIFT	Sample Shift Select Bit By default, QSPI samples data starting half a clock cycle after FLASH drives the data. Setting this bit to 1 delays sampling to accommodate external signal latency. 0: No Shift 1: Shift by Half Clock Cycle Note: This bit can only be modified when BUSY=0.
26	STRREN	Internal Receive Pass-Through Enable Bit This bit controls whether the current QSPI can be triggered from another QSPI. 0: Disable 1: Enable Note: This bit is valid only when MODE=11 and STRDIR=1.
25	STRTEN	Pass-through internal transfer enable bit This bit controls whether the current QSPI can trigger another QSPI. 0: Disable 1: Enable Note: This bit is valid only when MODE=11 and STRDIR=0.
24	STRDIR	Pass-through mode direction selection bit 0: Internal transfer direction is current QSPI → QSPIX (pass-through TX mode, QSPI for FLASH read) 1: Internal transfer direction is current QSPIX → current QSPI (pass-through RX mode, QSPI used to drive the display) Note: When using pass-through mode, ensure that DL[31:0] values match on both pass-through QSPIs.
23 ~ 22	MODE[1:0]	TWI/QSPI Mode Selection Bit 00: TWI Mode 01: Full-Duplex SPI Mode 10: Half-Duplex QSPI Mode 11: Pass-Through Mode
21	RW	R/W Control Bit 0: Master transmits data, Slave receives data 1: Master reads data, Slave transmits data Note: When the master reads data from the slave, configure DL[31:0] and set CLKONLY to 1.

Bit Number	Bit Symbol	Description
20	CPOL	QSPI/SPI Clock Polarity Control Bit 0: SCK is low when idle 1: SCK is high when idle
19	CPHA	QSPI/SPI Clock Phase Control Bit 0: Data sampled on the first clock edge of SCK 1: Data sampled on the second clock edge of SCK
18	DORD	QSPI/SPI Transfer Direction Selection Bit 0: MSB transmitted first 1: LSB transmitted first Note: This bit is valid only in full-duplex SPI mode (MODE=01). In half-duplex QSPI mode (MODE=10) and passthrough mode (MODE=11), MSB transmission is fixed.
17	CLKONLY	Transmit clock-only mode enable bit 0: Transmit clock-only mode disabled 1: Transmit clock-only mode enabled The number of output clocks is determined jointly by DWIDTH[1:0] and DL[31:0]. IO0 to IO3 behavior is controlled by LMODE[1:0] and RW. Note: This bit is recommended only when receiving data.
16	SMSTR	SPI master/slave mode selection bit 0: SPI configured as slave 1: SPI configured as master Note: This bit is valid only in full-duplex SPI mode (MODE=01). In half-duplex QSPI mode (MODE=10) and passthrough mode (MODE=11), the device is fixed as master.
11 ~ 8	QWCK[3:0]	SPI Clock Prescaler Control Bit 0000: f_{PCLK} 0001: $f_{PCLK} / 2$ 0010: $f_{PCLK} / 4$ 0011: $f_{PCLK} / 8$ 0100: $f_{PCLK} / 16$ 0101: $f_{PCLK} / 32$ 0110: $f_{PCLK} / 64$ 0111: $f_{PCLK} / 128$ 1000: $f_{PCLK} / 256$ 1001: $f_{PCLK} / 512$ 1010: $f_{PCLK} / 1024$ 1011: $f_{PCLK} / 2048$ 1100: $f_{PCLK} / 4096$ Others: $f_{PCLK} / 4096$ Note: To ensure correct operation of the SC32F11T/11G Series QSPI0/1, use a communication frequency below 32 MHz.
7	QWEN	QSPI Module Enable Control Bit 0: Disable the module 1: Enable the module
15 ~ 12 6 3 ~ 2	-	Reserved

13.9.1.2 TWI_QSPI Flag flag status register TWIn_QSPIn_STS

Register	Read/Write	Description	Reset Value
TWIn_QSPIn_STS (n=0/1)	R/W	TWIn_QSPIn Flag Status Register	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
BUSY	TMSTR	GCA	TXnE/RXnE	-	STATE[2:0]		
7	6	5	4	3	2	1	0
WCOL	DLUFIF	TXHIF	RXHIF	RXFIF	TXEIF	RXNEIF	QTWIF

Bit Number	Bit Symbol	Description
15	BUSY	Busy Status Bit Set by hardware during data transmission; automatically cleared when transmission completes. Note: This bit is valid only in Master mode. In continuous transmit/receive mode, this bit remains set until transmission finishes, then clears automatically.
7	WCOL	Write Conflict Flag Set to 1 by hardware and cleared by software writing 1. Indicates whether a write conflict has occurred: 0: No write conflict detected 1: Write conflict detected
6	DLUFIF	DL Underflow Flag Set to 1 by hardware and cleared by software writing 1. Indicates whether the DL[31:0] counter has underflowed. 0: DL count has not reached zero 1: DL count has reached zero
5	TXHIF	Transmit FIFO Less Than Half Full Status/Interrupt Flag Read-only bit, set or cleared by hardware, indicating the current state of the transmit FIFO: 0: Valid data in the transmit FIFO exceeds half its depth 1: Valid data count in the transmit FIFO is less than or equal to half of FIFO depth. If TXHIE = 1, an interrupt is generated.
4	RXHIF	Status bit / interrupt status bit indicating valid data in the receive FIFO exceeds half of FIFO depth. This bit is read-only, set or cleared by hardware to indicate the current receive FIFO status: 0: Valid data count in the receive FIFO is less than or equal to half of FIFO depth. 1: Valid data count in the receive FIFO exceeds half of FIFO depth. If RXHIE = 1, an interrupt is generated.
3	RXFIF	Receive FIFO full status bit. This bit is read-only, set or cleared by hardware to indicate whether the receive FIFO is full: 0: Receive FIFO is not full. 1: Receive FIFO is full.
2	TXEIF	Transmit FIFO Empty Flag Indicates if the transmit FIFO is empty: 0: Transmit FIFO not empty 1: Transmit FIFO empty This bit is set by hardware and cleared by writing 1 in 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 via software.

Bit Number	Bit Symbol	Description
1	RXNEIF	Receive FIFO Non-Empty Flag Indicates if the receive FIFO is empty: 0: Receive FIFO empty 1: Receive FIFO not empty This bit is set by hardware and cleared by writing 1 in software. Note: In DMA mode, this bit is cleared by the DMA module after reading the receive buffer. Software clearing is unnecessary.
0	QTWIF	QSPI Data Transfer Flag Set by hardware and cleared by writing 1 in software, this bit indicates if the current QSPI data frame transfer is complete: 0: Transfer incomplete/in progress 1: Transfer complete
31 ~ 24 11	-	Reserved

13.9.1.3 TWI_QSPI register TWIn_QSPIn_DATA

Register	Read/Write	Description	Reset Value
TWIn_QSPIn_DATA (n=0/1)	R/W	TWIn_QSPIn Data Register	0x0000_0000

31	30	29	28	27	26	25	24
QTWIDAT[31:24]							
23	22	21	20	19	18	17	16
QTWIDAT[23:16]							
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 Symbol	Description
31 ~ 0	QTWIDAT[31:0]	QSPI Data Buffer Read operation: Read received data from SPI RX FIFO Write operation: Write data to SPI TX FIFO for transmission

13.9.1.4 TWI_QSPI interrupt enable and DMA control register TWIn_QSPIn_IDE

Register	Read/Write	Description	Reset Value
TWIn_QSPIn_IDE (n=0/1)	R/W	Interrupt Enable and DMA Control Register for TWIn_QSPIn	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
-	-	-	-	-	-	DLUFIE	QTWIE
7	6	5	4	3	2	1	0
TXDMAEN	RXDMAEN	TXHIE	RXHIE	RXIE	TBIE	RXNEIE	INTEN

Bit Number	Bit Symbol	Description
9	DLUFIE	DL Underflow Interrupt Enable Bit

Bit Number	Bit Symbol	Description
		0: No interrupt generated when DLUFIF flag is set 1: Interrupt generated when DLUFIE is enabled and DLUFIF flag is set
8	QTWIE	QSPI/SPI Frame Transfer Complete Interrupt Enable Bit 0: No interrupt generated when QTWIF flag is set 1: Interrupt generated when QTWIF flag is set
7	TXDMAEN	DMA Transmit Channel Enable 0: Disable DMA Transmit 1: Enable DMA Transmit When enabled, setting TXEIF triggers DMA channel transmit request
6	RXDMAEN	DMA receive channel enable bit 0: Disable DMA receive function 1: Enable DMA receive function When enabled, setting RXNEIF triggers DMA channel receive request
5	TXHIE	Interrupt enable bit for transmit FIFO less than half full 0: No Interrupt generated when TXHIF is asserted 1: Generates Interrupt when TXHIF is asserted
4	RXHIE	Interrupt enable bit for receive FIFO exceeding half full 0: No Interrupt generated when RXHIF is set 1: Interrupt generated when RXHIF is set
3	RXIE	Interrupt enable bit for receive FIFO full 0: No interrupt generated when RXFIF is set 1: Interrupt generated when RXFIF is set
2	TBIE	Interrupt enable bit when transmit FIFO is empty 0: No interrupt generated when TXEIF is set 1: Interrupt generated when TXEIF is set
1	RXNEIE	Interrupt enable bit for receive FIFO non-empty 0: RXNEIF set, interrupt generation disabled 1: RXNEIF set, interrupt generated
0	INTEN	CPU Interrupt Request Enable Control Bit 0: Disable interrupt request 1: Enable interrupt request
31 ~ 10	-	Reserved

13.9.1.5 TWI_QSPI data length register TWIn_QSPIn_DL

Register	Read/Write	Description	Reset Value
TWIn_QSPIn_DL (n=0/1)	R/W	TWIn_QSPIn data length register	0x0000_0000

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

Bit Number	Bit Symbol	Description
31 ~ 0	DL[31:0]	QSPI data length configuration bits When QSPI reads data (CLKONLY=1), it reads DL[31:0] units of data, each with a bit width defined by DWIDTH[1:0]; Each time DWIDTH[1:0]-bit data is read, DL[31:0] decrements by 1. When DL reaches 0 and DLUFIE=1, an underflow interrupt is triggered.

13.9.2 QSPI0/1 register mapping

Register	Offset Address	Read/Write	Description	Reset Value	Access Restriction
TWI0_QSPI0 base address: 0x4002_0040					
TWI0_QSPI0_CON	0x00	R/W	TWI0_QSPI0 control register	0x0000_0000	-
TWI0_QSPI0_STS	0x04	R/W	TWI0_QSPI0 flag status register	0x0000_0000	-
TWI0_QSPI0_DATA	0x0C	R/W	TWI0_QSPI0 Data Register	0x0000_0000	Byte access not supported Half-word access
TWI0_QSPI0_IDE	0x10	R/W	TWI0_QSPI0 Interrupt Enable and DMA Control Register	0x0000_0000	-
TWI0_QSPI0_DL	0x14	R/W	TWI0_QSPI0 Data Length Register	0x0000_0000	-

Register	Offset Address	Read/Write	Description	Reset Value	Access Restriction
TWI1_QSPI1 Base Address: 0x4002_1040					
TWI1_QSPI1_CON	0x00	R/W	TWI1_QSPI1 Control Register	0x0000_0000	-
TWI1_QSPI1_STS	0x04	R/W	TWI1_QSPI1 Flag Status Register	0x0000_0000	-
TWI1_QSPI1_DATA	0x0C	R/W	TWI1_QSPI1 Data Register	0x0000_0000	Byte access not supported Half-word access
TWI1_QSPI1_IDE	0x10	R/W	TWI1_QSPI1 Interrupt Enable and DMA Control Register	0x0000_0000	-
TWI1_QSPI1_DL	0x14	R/W	TWI1_QSPI1 Data Length Register	0x0000_0000	-

14 SPI0 to 3

The SC32F11T/11G features up to four SPI ports: SPI0/1 are from the advanced two in one interface, and SPI2/3 are from the general two in one interface. SPI0/1 have 8-level FIFO, whereas SPI2/3 have no FIFO. In SPI communication mode, the output drive capability of SPI0/1 signal pins is enhanced, while SPI2/3 does not support this feature.

14.1 Clock source

- The SPI clock source in the SC32F11T/11G Series is single and derived from PCLK.

14.2 SPI0/1 feature

- Shares register addresses and signal pins with QSPI0/1 and TWI0/1; functions are fully independent.
- Supports 13-level SPI clock prescaler, user-configurable to lower frequencies.
- SPI0/1 Signal Pin Strong Drive:
 - In SPI communication mode, the output drive capability of the corresponding signal pins is enhanced; in other modes, it matches standard IO characteristics.
- Features 8-level FIFO buffer with independent transmit and receive
 - The FIFO function of SPI0/1 allows continuous writing of up to 8 pieces of 8/16/24/32-bit data into the SPI transmit buffer (SPI0/1_DATA). During transmission, data written first is transmitted first. When the data written by the user into the FIFO is completely transmitted, the transmit FIFO empty flag TXEIF is set to 1. If the FIFO is full, the write collision flag WCOL is set, preventing the user from writing data into the FIFO until data inside the FIFO is transmitted and space is available. The interrupt flag QTWIF is set once after each frame of data is transmitted.
 - When continuously reading up to 8 units of 8/16/24/32-bit data from the SPI receive buffer (SPI0/1_DATA), the earliest received data is read first.
 - Half FIFO data transfer interrupts and their corresponding flags are provided to help users read or write data in a timely manner:
 - ◆ An interrupt and corresponding flag TXHIF are provided for when valid transmit FIFO data is less than half full.
 - ◆ Provides interrupt when receive FIFO is more than half full and the corresponding flag RXHIF
 - Supports receive buffer overflow interrupt and the corresponding flag to promptly notify exceptions
- Supports Master and Slave Mode
- Supports DMA
 - Enable TXDMAEN; after the transmit buffer empty flag TXEIF is set, a DMA request can be triggered. After DMA writes to the transmit buffer, the TXEIF flag is automatically cleared
 - Enable RXDMAEN; after the receive buffer non-empty flag RXNEIF is set, a DMA request can be triggered. After DMA reads from the receive buffer, the RXNEIF flag is automatically cleared

14.3 SPI2/3 features

- Shares register addresses and signal pins with TWI2/3, but functions are fully independent.
- Supports 13-level SPI clock prescaler, user-configurable to lower frequencies.

- Signal pins can be mapped to three other IO groups.
- No FIFO.
- Supports Master and Slave Mode
- Supports DMA.
 - SPI2 can generate DMA requests
 - SPI3 cannot generate DMA requests

14.4 Signal description

Master Out Slave In (MOSI):

This signal connects the master device to a slave device. Data is serially transmitted from the master to the slave via MOSI; master outputs, slave inputs.

Master In Slave Out (MISO):

This signal connects the slave device to the master device. Data is serially transmitted from the slave to the master via MISO; slave outputs, master inputs. When SPI is configured as a slave and not selected, the slave's MISO pin remains in a high-impedance state.

SPI Serial Clock (SCK):

SCK synchronizes data transmission on MOSI and MISO lines. One byte transfers every 8 clock cycles. If the slave is not selected, it ignores the SCK signal.

14.5 Operating mode

SPI is configurable as either master or slave mode. SPI configuration and initialization are done by setting the SPI control registers TWI0_QSPI0_CON/TWI1_QSPI1_CON/TWI2_SPI2_CON/TWI3_SPI3_CON and enabling SPI interrupts and DMA via TWI0_QSPI0_IDE / TWI1_QSPI1_IDE / TWI2_SPI2_IDE / TWI3_SPI3_IDE. Once configured, data transfer uses the SPI data registers TWI0_QSPI0_DATA / TWI1_QSPI1_DATA / TWI2_SPI2_DATA / TWI3_SPI3_DATA (referred to as SPD).

During SPI communication, data is shifted in and out synchronously. The serial clock line (SCK) synchronizes data transfer and sampling on the MOSI and MISO lines. If the device is not selected, it cannot participate in any activity on the SPI bus.

When the SPI master transmits data to the slave via MOSI, the slave simultaneously transmits data to the master via MISO, enabling synchronous full-duplex communication under the same clock. The transmit and receive shift registers share the same special function address. Writing to the SPI data register (SPD) loads data into the transmit shift register, while reading from SPD fetches data from the receive shift register.

Some devices' SPI interfaces include an SS pin (slave select, active low). When communicating with the SC32F11T/11G SPI, SS pin connections of other devices on the bus must be configured according to the communication mode.

The table below shows the SS pin connections of other devices on the SPI bus for different SPI communication modes in SC32F11T/11G:

SC32F11T/11G SPI	Other devices on the SPI bus	Mode	Slave SS (Slave Select Pin)
Master mode	Slave mode	Single master, single slave	Drive low
		Single master, multiple slaves	SC32F11T/11G provides multiple I/O lines connected individually to the slave SS pins. Before data transfer, each slave's SS pin must be driven low.
Slave mode	Master mode	Single master, single slave	Drive high

14.5.1 Master mode

- Mode activation

The SPI master controls the start of all data transfers on the SPI bus. When the SMSTR bit in the SPI control registers TWI0_QSPI0_CON, TWI1_QSPI1_CON, TWI2_SPI2_CON, or TWI3_SPI3_CON is set, SPI runs in master mode, allowing only one master to initiate transfers.

- Transmit

In SPI master mode, write data to SPD in 8/16/24/32-bit mode; the data is loaded into the transmit shift register. If the transmit shift register already contains data, the master SPI generates a WCOL signal indicating a write collision caused by writing too fast. However, the data in the transmit shift register remains unaffected, and transmission continues uninterrupted. If the transmit shift register is not empty, the master device immediately shifts out its data serially on MOSI according to the SPI clock frequency on SCK. After transmission completes, the QTWIF bit in the status registers TWI0_QSPI0_STS, TWI1_QSPI1_STS, TWI2_SPI2_STS, or TWI3_SPI3_STS is set. If the SPI interrupt is enabled, an interrupt is also generated when QTWIF is set to 1.

- Receive

When the master transmits data via the MOSI line, the slave simultaneously sends its transmit shift register content to the master's receive shift register via the MISO line, achieving full-duplex communication. Thus, setting the QTWIF flag to 1 indicates both transmission and reception are complete. Data received by the slave is stored in the master's receive shift register according to the MSB- or LSB-first transfer order. Once a byte is fully shifted into the receive register, the processor can read it through SPD.

14.5.2 Slave mode

- Mode activation

SPI operates in slave mode when the SMSTR bit in the TWI0_QSPI0_CON, TWI1_QSPI1_CON, TWI2_SPI2_CON, or TWI3_SPI3_CON control register is cleared to 0.

- Transmit and Receive

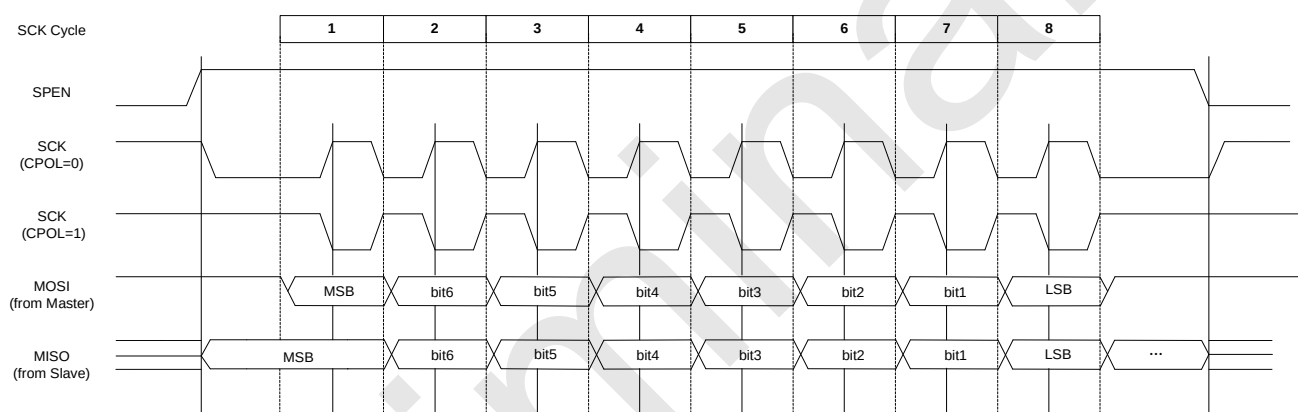
In slave mode, data is shifted in via MOSI and shifted out via MISO, synchronized by the master-controlled SCK signal. A bit counter tracks SCK edges. When the receive shift register has shifted in 8 bits (one byte) and the transmit shift register has shifted out 8 bits, the QTWIF flag is set. Data is available by reading the SPD register. If SPI interrupts are enabled, setting QTWIF also triggers an interrupt. The receive shift register holds its data and QTWIF remains set, preventing the SPI slave from receiving new data until QTWIF is cleared. The SPI slave must write the data to be transmitted into the transmit shift register before the master starts a new data transfer. If data is not written before transmission begins, the slave will transmit a 0x00

byte to the master. If a write to the SPI data register occurs during transmission, the SPI slave's WCOL flag is set. This indicates a write collision if the transmit shift register already contains data. However, the shift register content is unaffected, and transmission continues uninterrupted.

14.6 Transmission format

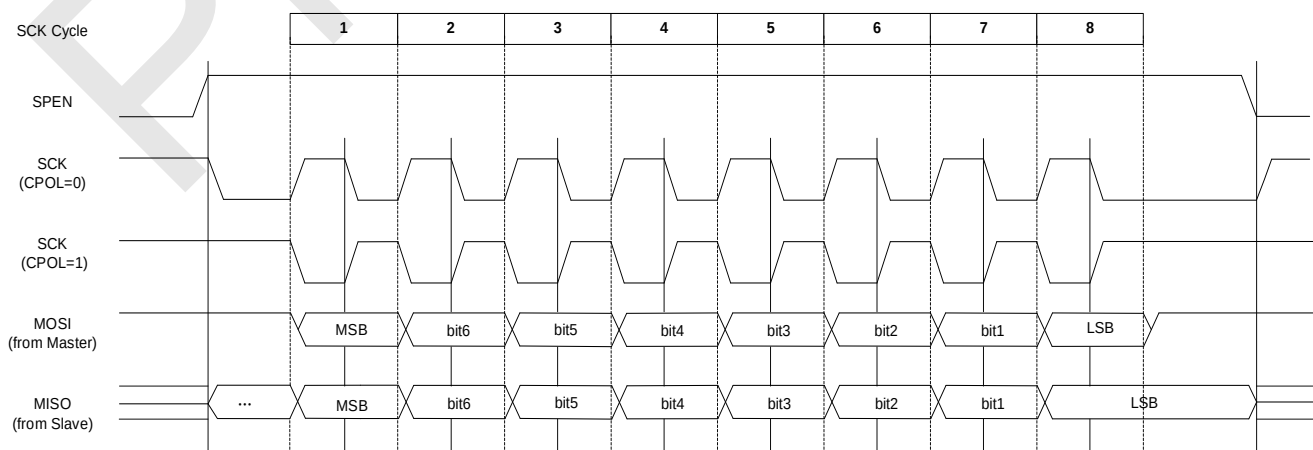
By configuring the CPOL and CPHA bits in the SPI control registers TWI0_QSPI0_CON, TWI1_QSPI1_CON, TWI2_SPI2_CON, or TWI3_SPI3_CON via software, users can select one of four SPI clock polarity and phase combinations. The CPOL bit sets the clock polarity, indicating the idle level. It has minimal effect on the SPI transfer format. The CPHA bit sets the clock phase, specifying the clock edge for data sampling. Clock polarity and phase settings must match between master and slave devices.

With CPHA = 0, data is captured on the first SCK edge. The slave must prepare data before this edge.



Data transfer timing diagram for CPHA = 0

With CPHA = 1, the master outputs data on the first SCK edge to MOSI. The slave treats this edge as the start signal and begins capturing data on the second edge. Therefore, the user must complete writing SPD within the first two SCK edges. This data transfer method is the preferred communication form between a single master and a single slave device.



CPHA = 1 data transfer diagram

14.7 Error detection

Writing to SPD during data sequence transmission causes a write collision, setting the WCOL bit in the SPI flag status registers TWI0_QSPI0_STS, TWI1_QSPI1_STS, TWI2_SPI2_STS, or TWI3_SPI3_STS. Setting WCOL does not generate an interrupt or abort transmission. The WCOL bit must be cleared by software.

14.8 Comparison between SPI0/1 and SPI2/3

Comparison BIT	SPI0/1	SPI2/3
Strong drive on signal pin	Available	None
WCOL	When the transmit FIFO is full, writing to it is not possible; WCOL will be set, indicating a buffer write collision.	While a frame is being transmitted, writing to the transmit buffer is prohibited; WCOL will be set, indicating a buffer write collision.
QTWIF	From this bit onwards, it indicates the completion of a frame's data reception / transmission.	From this bit onwards, it indicates the completion of a frame's data reception / transmission.
QTWIE	QTWIF interrupt enable bit set	None
RXHIE	Interrupt enable bit for receive FIFO exceeding half full	None
TXHIE	Interrupt enable bit for transmit FIFO less than half full	None
RXIE	Interrupt enable bit for receive FIFO full	None
TBIE	Interrupt enable bit for transmit FIFO empty	Interrupt enable bit for transmit buffer empty
RXNEIE	Interrupt enable bit for receive FIFO non-empty	None
RXHIF	Starting at this bit position, indicates valid data in the receive FIFO exceeds half full.	None
TXHIF	Starting at this bit position, indicates valid data in the transmit FIFO is less than half full.	None
RXFIF	Starting at this bit position, indicates the receive FIFO is full.	None
TXEIF	Starting at this bit position, indicates the transmit FIFO is empty.	Starting at this bit position, indicates the transmit buffer is empty.
RXNEIF	Receive FIFO non-empty flag	None
DMA	DMA requests are triggered by the transmit buffer empty flag TXEIF and the receive buffer non-empty status bit RXNEIF.	SPI2: Frame-end unified set request SPI3: DMA not supported

14.9 SPI0/1 registers

14.9.1 SPI0/1 related register table

14.9.1.1 TWI_QSPI control register TWIn_QSPIn_CON

Register	Read/Write	Description	Reset Value
TWIn_QSPIn_CON (n=0/1)	R/W	TWIN_QSPI Control Register	0x0000_0000

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

Bit Number	Bit Symbol	Description
29 ~ 28	DWIDTH[1:0]	Full-Duplex SPI and Half-Duplex QSPI Interface Transfer Width Select Bit 00: 8 Bits 01: 16 Bits 10: 24 Bits 11: 32 Bits
27	SSHIFT	Sample Shift Select Bit By default, SPI starts sampling data half a clock cycle after FLASH drives the data. Setting this bit to 1 compensates for external signal delay and postpones data sampling. 0: No Shift 1: Shift by Half Clock Cycle Note: This bit can only be modified when BUSY=0.
23 ~ 22	MODE[1:0]	TWI/QSPI Mode Selection Bit 00: TWI Mode 01: Full-Duplex SPI Mode 10: Half-Duplex QSPI Mode 11: Pass-Through Mode
20	CPOL	QSPI/SPI Clock Polarity Control Bit 0: SCK is low when idle 1: SCK is high when idle
19	CPHA	QSPI/SPI Clock Phase Control Bit 0: Data sampled on the first clock edge of SCK 1: Data sampled on the second clock edge of SCK
18	DORD	QSPI/SPI Transfer Direction Selection Bit 0: MSB transmitted first 1: LSB transmitted first
16	SMSTR	QSPI/SPI Master/Slave Mode Selection Bit 0: SPI configured as slave 1: SPI configured as master
11 ~ 8	QTWCK[3:0]	SPI Clock Prescaler Control Bit 0000:fPCLK

Bit Number	Bit Symbol	Description
		0001:f _{PCLK} /2 0010:f _{PCLK} /4 0011:f _{PCLK} /8 0100:f _{PCLK} /16 0101:f _{PCLK} /32 0110:f _{PCLK} /64 0111:f _{PCLK} /128 1000:f _{PCLK} /256 1001:f _{PCLK} /512 1010:f _{PCLK} /1024 1011:f _{PCLK} /2048 1100:f _{PCLK} /4096 Others: f _{PCLK} /4096 Note: To ensure correct communication of SC32F11T/11G Series SPI0/1, select a communication frequency below 16 MHz.
7	QWEN	TWI_QSPI Module Enable Control Bit 0: Disable the module 1: Enable the module
15 ~ 12 6 3 ~ 2	-	Reserved

14.9.1.2 TWI_QSPI flag status register TWIn_QSPIn_STS

Register	Read/Write	Description	Reset Value
TWIn_QSPIn_STS (n=0/1)	R/W	TWIn_QSPIn Flag Status Register	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
BUSY	TMSTR	GCA	TXnE/RXnE	-	STATE[2:0]		
7	6	5	4	3	2	1	0
WCOL	DLUFIF	TXHIF	RXHIF	RXFIF	TXEIF	RXNEIF	QWIF

Bit Number	Bit Symbol	Description
15	BUSY	Busy Status Bit Set by hardware during data transmission; automatically cleared when transmission completes. Note: This bit is valid only in Master mode. In continuous transmit/receive mode, this bit remains set until transmission finishes, then clears automatically.
7	WCOL	Write Conflict Flag Set to 1 by hardware and cleared by software writing 1. Indicates whether a write conflict has occurred: 0: No write conflict detected 1: Write conflict detected
5	TXHIF	Transmit FIFO Less Than Half Full Status/Interrupt Flag Read-only bit, set or cleared by hardware, indicating the current state of the transmit FIFO: 0: Valid data count in the transmit FIFO is greater than half the FIFO depth 1: Valid data count in the transmit FIFO is less than or equal to half the FIFO depth. If TXHIE = 1, an interrupt will be generated.

Bit Number	Bit Symbol	Description
4	RXHIF	Status bit / interrupt status bit indicating valid data in the receive FIFO exceeds half of FIFO depth. This bit is read-only, set or cleared by hardware to indicate the current receive FIFO status: 0: Valid data count in the receive FIFO is less than or equal to half the FIFO depth 1: When the valid data count in the receive FIFO exceeds half its depth, an interrupt is generated if RXHIE = 1.
3	RXFIF	Receive FIFO full status bit. This bit is read-only, set or cleared by hardware to indicate whether the receive FIFO is full: 0: Receive FIFO is not full. 1: Receive FIFO is full.
2	TXEIF	Transmit FIFO Empty Flag Indicates if the transmit FIFO is empty: 0: Transmit FIFO not empty 1: Transmit FIFO empty This bit is set by hardware and cleared by writing 1 in 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 via software.
1	RXNEIF	Receive FIFO Non-Empty Flag Indicates if the receive FIFO is empty: 0: Receive FIFO empty 1: Receive FIFO not empty This bit is set by hardware and cleared by writing 1 in software. Note: In DMA mode, this bit is cleared by the DMA module after reading the receive buffer. Software clearing is unnecessary.
0	QTWIF	SPI data transfer flag This bit is set by hardware and cleared by writing 1 in software. It indicates whether the current SPI data frame transfer is complete. 0: Transfer incomplete/in progress 1: Transfer complete
31 ~ 24 11	-	Reserved

14.9.1.3 TWI_QSPI data register TWIn_QSPIn_DATA

Register	Read/Write	Description	Reset Value
TWIn_QSPIn_DATA (n=0/1)	R/W	TWIn_QSPIn Data Register	0x0000_0000

31	30	29	28	27	26	25	24
QTWIDAT[31:24]							
23	22	21	20	19	18	17	16
QTWIDAT[23:16]							
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 Symbol	Description
31 ~ 0	QTWIDAT[31:0]	SPI data buffer Read operation: Read received data from SPI RX FIFO Write operation: Write data to SPI TX FIFO for transmission

14.9.1.4 TWI_QSPI interrupt enable and DMA control register TWIn_QSPIn_IDE

Register	Read/Write	Description	Reset Value
TWIn_QSPIn_IDE (n=0/1)	R/W	TWIn_QSPIn 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
-	-	-	-	-	-	DLUFIE	QTWIE
7	6	5	4	3	2	1	0
TXDMAEN	RXDMAEN	TXHIE	RXHIE	RXIE	TBIE	RXNEIE	INTEN

Bit Number	Bit Symbol	Description
8	QTWIE	QSPI/SPI Frame Transfer Complete Interrupt Enable Bit 0: No interrupt generated when QTWIF flag is set 1: Interrupt generated when QTWIF flag is set
7	TXDMAEN	DMA Transmit Channel Enable 0: Disable DMA Transmit 1: Enable DMA Transmit When enabled, setting TXEIF triggers DMA channel transmit request
6	RXDMAEN	DMA receive channel enable bit 0: Disable DMA receive function 1: Enable DMA receive function When enabled, setting RXNEIF triggers DMA channel receive request
5	TXHIE	Interrupt enable bit for transmit FIFO less than half full 0: No Interrupt generated when TXHIF is asserted 1: Generates Interrupt when TXHIF is asserted
4	RXHIE	Interrupt enable bit for receive FIFO exceeding half full 0: No Interrupt generated when RXHIF is set 1: Interrupt generated when RXHIF is set
3	RXIE	Interrupt enable bit for receive FIFO full 0: No interrupt generated when RXFIF is set 1: Interrupt generated when RXFIF is set
2	TBIE	Interrupt enable bit when transmit FIFO is empty 0: No interrupt generated when TXEIF is set 1: Interrupt generated when TXEIF is set
1	RXNEIE	Interrupt enable bit for receive FIFO non-empty 0: RXNEIF set, interrupt generation disabled 1: RXNEIF set, interrupt generated
0	INTEN	CPU Interrupt Request Enable Control Bit 0: Disable interrupt request 1: Enable interrupt request
31 ~ 10	-	Reserved

14.9.2 SPI0/1 register mapping

Register	Offset Address	Read/Write	Description	Reset Value	Access Restriction
TWI0_QSPI0 base address: 0x4002_0040					
TWI0_QSPI0_CON	0x00	R/W	TWI0_QSPI0 control register	0x0000_0000	-

Register	Offset Address	Read/Write	Description	Reset Value	Access Restriction
TWI0_QSPI0_STS	0x04	R/W	TWI0_QSPI0 flag status register	0x0000_0000	-
TWI0_QSPI0_DATA	0x0C	R/W	TWI0_QSPI0 Data Register	0x0000_0000	Byte/Half-word Access Not Supported
TWI0_QSPI0_IDE	0x10	R/W	TWI0_QSPI0 Interrupt Enable and DMA Control Register	0x0000_0000	-

Register	Offset Address	Read/Write	Description	Reset Value	Access Restriction
TWI1_QSPI1 Base Address: 0x4002_1040					
TWI1_QSPI1_CON	0x00	R/W	TWI1_QSPI1 Control Register	0x0000_0000	-
TWI1_QSPI1_STS	0x04	R/W	TWI1_QSPI1 Flag Status Register	0x0000_0000	-
TWI1_QSPI1_DATA	0x0C	R/W	TWI1_QSPI1 Data Register	0x0000_0000	Byte/Half-word Access Not Supported
TWI1_QSPI1_IDE	0x10	R/W	TWI1_QSPI1 Interrupt Enable and DMA Control Register	0x0000_0000	-

14.10 SPI2/3 registers

14.10.1 SPI2/3 related registers table

14.10.1.1 TWI_SPI control register TWIn_SPIn_CON

Register	Read/Write	Description	Reset Value
TWIn_SPIn_CON (n=2/3)	R/W	TWIn_SPIn control register	0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	DWIDTH	-	-	-	-
23	22	21	20	19	18	17	16
MODE[1:0]		-	CPOL	CPHA	DORD	-	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 Symbol	Description
28	DWIDTH	SPI transfer width selection bit 0: 8-bit mode 1: 16-bit mode
23 ~ 22	MODE[1:0]	TWI_SPI Mode Selection Bit 00: TWI Mode 01: SPI Mode 10: Reserved

Bit Number	Bit Symbol	Description																																								
		11: Reserved																																								
20	CPOL	SPI Clock Polarity Control Bit 0: SCK is low when idle 1: SCK is high when idle																																								
19	CPHA	SPI Clock Phase Control Bit 0: Data sampled on the first clock edge of SCK 1: Data sampled on the second clock edge of SCK																																								
18	DORD	SPI Transfer Direction Selection Bit 0: MSB transmitted first 1: LSB transmitted first																																								
16	SMSTR	SPI master/slave mode selection bit 0: SPI configured as slave 1: SPI configured as master																																								
14 ~ 13	SPOS[1:0]	SPI2 Signal Port Mapping Control Bit<table><tr><th>Signal SPOS Value</th><th>MISO2</th><th>MOSI2</th><th>SCK2</th></tr><tr><td>SPOS[1:0]=00</td><td>PA2</td><td>PA1</td><td>PA0</td></tr><tr><td>SPOS[1:0]=01</td><td>PA10</td><td>PA11</td><td>PA12</td></tr><tr><td>SPOS[1:0]=10</td><td>PB11</td><td>PB10</td><td>PB9</td></tr><tr><td>SPOS[1:0]=11</td><td>PC11</td><td>PC10</td><td>PC9</td></tr></table> SPI3 Signal Port Mapping Control Bit<table><tr><th>Signal SPOS Value</th><th>MISO3</th><th>MOSI3</th><th>SCK3</th></tr><tr><td>SPOS[1:0]=00</td><td>PA4</td><td>PA3</td><td>PA2</td></tr><tr><td>SPOS[1:0]=01</td><td>PA9</td><td>PA8</td><td>PA7</td></tr><tr><td>SPOS[1:0]=10</td><td>PB7</td><td>PB6</td><td>PB5</td></tr><tr><td>SPOS[1:0]=11</td><td>PB11</td><td>PB12</td><td>PB13</td></tr></table>	Signal SPOS Value	MISO2	MOSI2	SCK2	SPOS[1:0]=00	PA2	PA1	PA0	SPOS[1:0]=01	PA10	PA11	PA12	SPOS[1:0]=10	PB11	PB10	PB9	SPOS[1:0]=11	PC11	PC10	PC9	Signal SPOS Value	MISO3	MOSI3	SCK3	SPOS[1:0]=00	PA4	PA3	PA2	SPOS[1:0]=01	PA9	PA8	PA7	SPOS[1:0]=10	PB7	PB6	PB5	SPOS[1:0]=11	PB11	PB12	PB13
Signal SPOS Value	MISO2	MOSI2	SCK2																																							
SPOS[1:0]=00	PA2	PA1	PA0																																							
SPOS[1:0]=01	PA10	PA11	PA12																																							
SPOS[1:0]=10	PB11	PB10	PB9																																							
SPOS[1:0]=11	PC11	PC10	PC9																																							
Signal SPOS Value	MISO3	MOSI3	SCK3																																							
SPOS[1:0]=00	PA4	PA3	PA2																																							
SPOS[1:0]=01	PA9	PA8	PA7																																							
SPOS[1:0]=10	PB7	PB6	PB5																																							
SPOS[1:0]=11	PB11	PB12	PB13																																							
11 ~ 8	QWCK[3:0]	SPI Clock Prescaler Control Bit 0000:f_{PCLK} 0001:f_{PCLK} /2 0010:f_{PCLK} /4 0011:f_{PCLK} /8 0100:f_{PCLK} /16 0101:f_{PCLK} /32 0110:f_{PCLK} /64 0111:f_{PCLK} /128 1000:f_{PCLK} /256 1001:f_{PCLK} /512 1010:f_{PCLK} /1024 1011:f_{PCLK} /2048 1100:f_{PCLK} /4096 Others: f_{PCLK1} / 4096 Note: To ensure proper communication of SC32F11T/11G Series SPI2/3, select a communication frequency below 16 MHz.																																								
7	QWEN	TWI_QSPI Enable Control Bit 0: Disable the module 1: Enable the module																																								
31 ~ 29 27 ~ 24 21,17 15,12 6,3 ~ 2	-	Reserved																																								

14.10.1.2 TWI_SPI flag status register TWIn_SPIn_STS

Register	Read/Write	Description	Reset Value
TWIn_SPIn_STS (n=2/3)	R/W	TWIn_SPIn Flag Status Register	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
-	TMSTR	GCA	TXnE/RXnE	-	STATE[2:0]		
7	6	5	4	3	2	1	0
WCOL	-	-	-	-	TXEIF	-	QTWIF

Bit Number	Bit Symbol	Description
7	WCOL	Write Conflict Flag Set to 1 by hardware and cleared by software writing 1. Indicates whether a write conflict has occurred: 0: No write conflict detected 1: Write conflict detected
2	TXEIF	Transmit Buffer Empty Flag Indicates whether the transmit buffer is empty: 0: Transmit buffer not empty 1: Transmit buffer empty This bit is set by hardware and cleared by writing 1 in 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 via software.
0	QTWIF	SPI data transfer flag This bit is set by hardware and cleared by writing 1 via software. It indicates whether the current SPI data transfer is complete: 0: Transfer incomplete/in progress 1: Transfer complete
31 ~ 24 15,11 6 ~ 3 1	-	Reserved

14.10.1.3 TWI_SPI data register TWIn_SPIn_DATA

Register	Read/Write	Description	Reset Value
TWIn_SPIn_DATA (n=2/3)	R/W	TWIn_SPIn 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
QTWIDAT[15:8]							
7	6	5	4	3	2	1	0
QTWIDAT[7:0]							

Bit Number	Bit Symbol	Description
15 ~ 0	QTWIDAT[15:0]	SPI data buffer Read operation: Reads received data from the SPI receive buffer Write operation: Writes data to be transmitted into the SPI transmit buffer
31 ~ 16	-	Reserved

14.10.1.4 TWI_SPI interrupt enable and DMA control register TWIn_SPIn_IDE

Register	Read/Write	Description	Reset Value
TWIn_SPIn_IDE (n=2/3)	R/W	TWIn_SPIn 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
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
TXDMAEN	RXDMAEN	-	-	-	TBIE	-	INTEN

Bit Number	Bit Symbol	Description
7	TXDMAEN	DMA Transmit Channel Enable 0: Disable DMA Transmit 1: Enable DMA Transmit When enabled, setting TXEIF triggers DMA channel transmit request Note: SPI3 does not support DMA
6	RXDMAEN	DMA receive channel enable bit 0: Disable DMA receive function 1: Enable DMA receive function When this bit is enabled, the QTWIF flag can trigger a DMA channel receive request Note: SPI3 does not support DMA
2	TBIE	Interrupt enable bit for empty transmit buffer 0: No interrupt generated when TXEIF is set 1: Interrupt generated when TXEIF is set
0	INTEN	CPU Interrupt Request Enable Control Bit 0: Disable interrupt request 1: Enable interrupt request
31 ~ 8 5 ~ 3 1	-	Reserved

14.10.2 SPI2/3 register mapping

Register	Offset Address	Read/Write	Description	Reset Value	Access Restriction
TWI2_SPI2 Base Address: 0x4002_0060					
TWI2_SPI2_CON	0x00	R/W	TWI2_SPI2 Control Register	0x0000_0000	-
TWI2_SPI2_STS	0x04	R/W	TWI2_SPI2 Flag Register	0x0000_0000	-

Register	Offset Address	Read/Write	Description	Reset Value	Access Restriction
TWI2_SPI2_DATA	0x0C	R/W	TWI2_SPI2 Data Register	0x0000_0000	Byte access not supported Half-word access
TWI2_SPI2_IDE	0x10	R/W	TWI2_SPI2 Interrupt Enable and DMA Control Register	0x0000_0000	-

Register	Offset Address	Read/Write	Description	Reset Value	Access Restriction
TWI3_SPI3 Base Address: 0x4002_1060					
TWI3_SPI3_CON	0x00	R/W	TWI3_SPI3 Control Register	0x0000_0000	-
TWI3_SPI3_STS	0x04	R/W	TWI3_SPI3 Flag Register	0x0000_0000	-
TWI3_SPI3_DATA	0x0C	R/W	TWI3_SPI3 Data Register	0x0000_0000	Byte access not supported Half-word access
TWI3_SPI3_IDE	0x10	R/W	TWI3_SPI3 Interrupt Enable and DMA Control Register	0x0000_0000	-

15 TWI0 to 3

15.1 Clock source

- TWI clock source of the SC32F11T/11G Series is only from PCLK.

15.2 TWI0/1 features

- Shares register addresses and signal pins with QSPI0/1 and SPI0/1, but functions are completely independent.
- Supports an 11-step TWI clock prescaler; in Master Mode, the TWI communication rate defaults to the lowest prescaler setting ($f_{PCLK}/4$).
- Configurable for Master Mode or Slave Mode
- Bidirectional data transfer between Master and Slave
- Communication rate up to 1 Mbps
- Optional clock stretching
- Supports DMA
 - Both TWI0 and TWI1 can generate DMA requests

15.3 Features of TWI2 and TWI3

- Shares register addresses and signal pins with SPI2 and SPI3 but functions completely independently
- Supports an 11-step TWI clock prescaler; in Master Mode, the TWI communication rate defaults to the lowest prescaler setting ($f_{PCLK}/4$).
- Three selectable signal pin mapping groups
- Configurable for Master Mode or Slave Mode
- Bidirectional data transfer between Master and Slave
- Communication rate up to 1 Mbps
- Optional clock stretching
- Supports DMA
 - TWI2 can generate DMA requests
 - TWI3 cannot generate DMA Request

15.4 TWI signal description

On the TWI Bus, data is transferred synchronously byte by byte between the Master and Slave devices via the clock line SCL and the data line SDA. Each data byte consists of 8 bits; one SCL clock pulse transmits one data bit, starting with the most significant bit (MSB). After each byte, an acknowledgment bit follows. Each bit is sampled when SCL is high. Therefore, the SDA line may change while SCL is low but must remain stable when SCL is high. When SCL is high, transitions on the SDA line are interpreted as commands (START or STOP).

- **TWI Clock Signal Line (SCL)**

This clock signal is generated by the Master and connected to all Slave devices. One data byte is transferred every 9 clock periods. The first 8 cycles are used for data transmission, and the last clock serves as the acknowledgment clock from the receiver. The idle state should be at a logic high level, pulled up by the pull-up resistor on the SCL line.

- **TWI data signal line (SDA)**

SDA is a bidirectional signal line, and the idle state should be at a logic high level, pulled up by the pull-up resistor on the SDA line.

15.5 Slave operating mode

- **Mode activation**

When the TWI enable flag (QWEN=1) is set and a start condition is received from the master, the mode is activated.

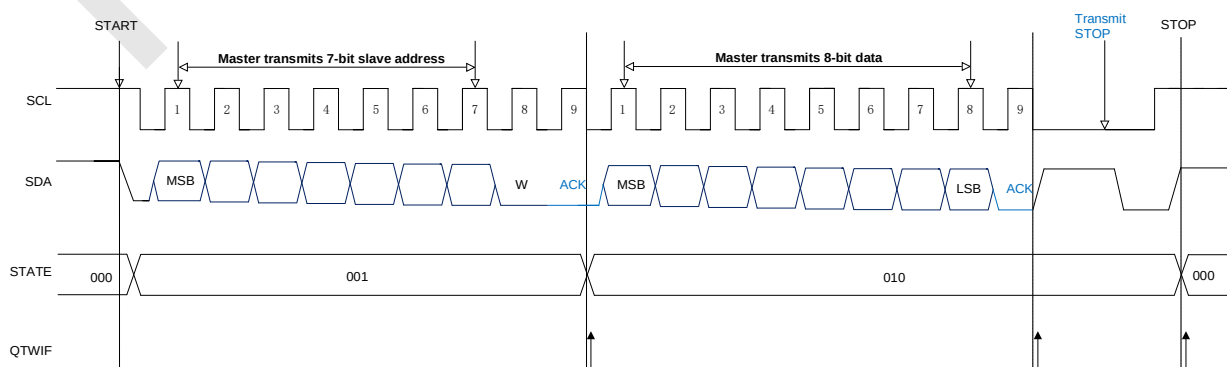
The slave transitions from IDLE mode (STATE[2:0] = 000) to receiving the first frame address state (STATE[2:0] = 001), awaiting the master's first frame data. The master's first frame contains a 7-bit address and a 1-bit read/write flag. All slaves on the TWI bus receive this frame. After transmission, the master releases control of the SDA line. If the master's address matches a slave's address register, that slave is selected. The selected slave then examines the 8th bit on the bus, the read/write flag (=1 for read command; =0, write command), then drives the SDA line low and, on the 9th SCL clock cycle, generates a low-level ACK to the master before releasing the bus. After the slave is selected, it enters different states based on the read/write bit.

- **Non-general address response, slave receive mode.**

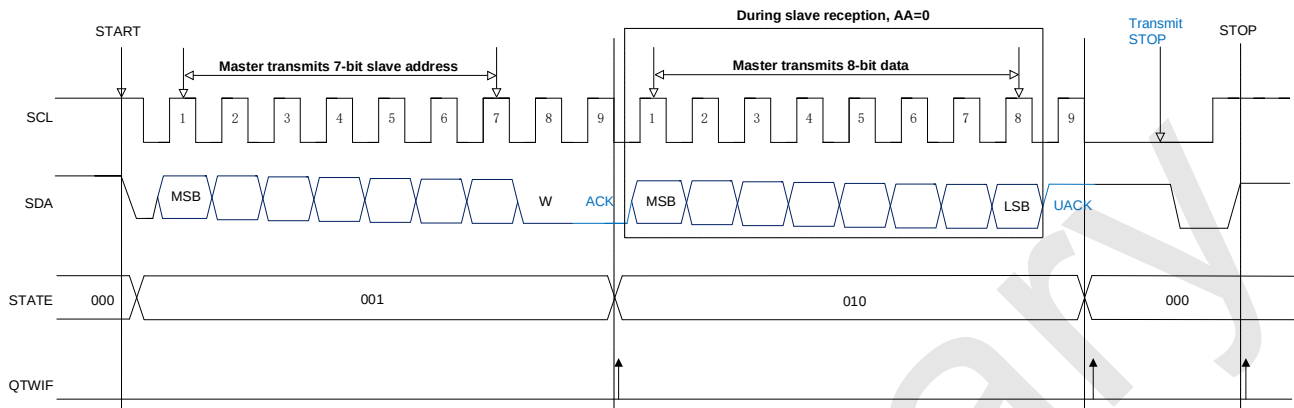
If the read/write bit in the first frame is write (0), the slave enters the receive state (STATE[2:0] = 010) to await data from the master. After every 8 bits transmitted, the master releases the bus to wait for the slave's ACK during the 9th clock cycle.

If the slave's ACK is low, the master's communication can proceed in one of three ways:

1. Continue transmitting data.
2. Retransmit the start signal; the slave reenters the state of receiving the first frame address (STATE[2:0] = 001).
3. Transmit the stop signal to indicate the end of the current transmission. The slave returns to idle and waits for the master's next start signal.



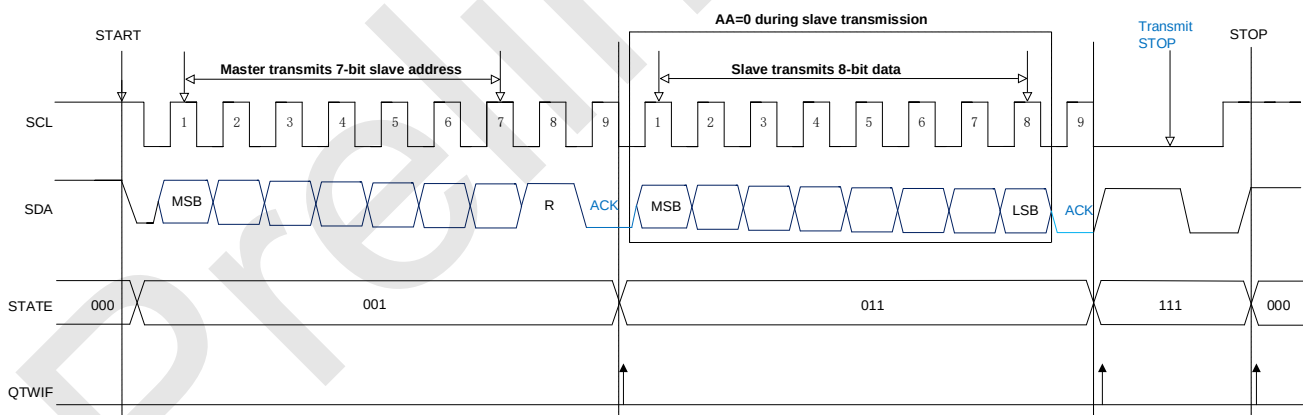
If the slave's acknowledgment is high (during reception, the AA value in the slave register is cleared to 0), it indicates that after the current byte transmission, the slave will terminate the transmission and return to idle (STATE[2:0] = 000), no longer receiving data from the master.



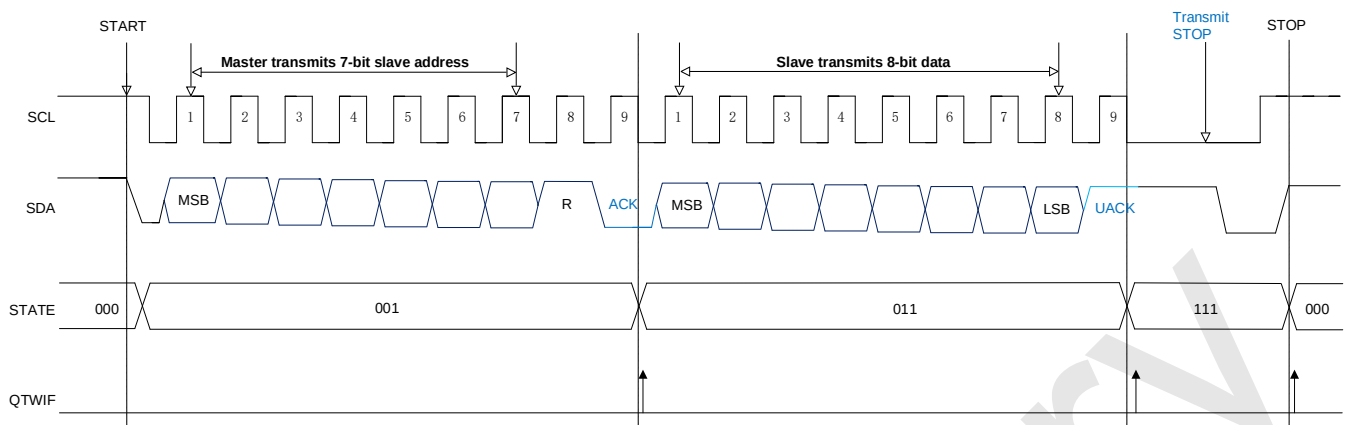
● Non-general address response, slave transmit mode.

If the read/write bit in the first frame is read (1), the slave takes control of the bus and transmits data to the master. After transmitting 8 bits, the slave releases the bus and waits for the master's acknowledgment.

If the master responds with a low level, the slave continues transmitting data. During transmission, if the AA value in the slave register is overwritten to 0, the slave will terminate transmission and release the bus after sending the current byte, then wait for the master's stop or restart signal (STATE[2:0] = 101).



If the master responds with a high level, the slave's STATE[2:0] is set to 100, waiting for the master's stop or restart signal.

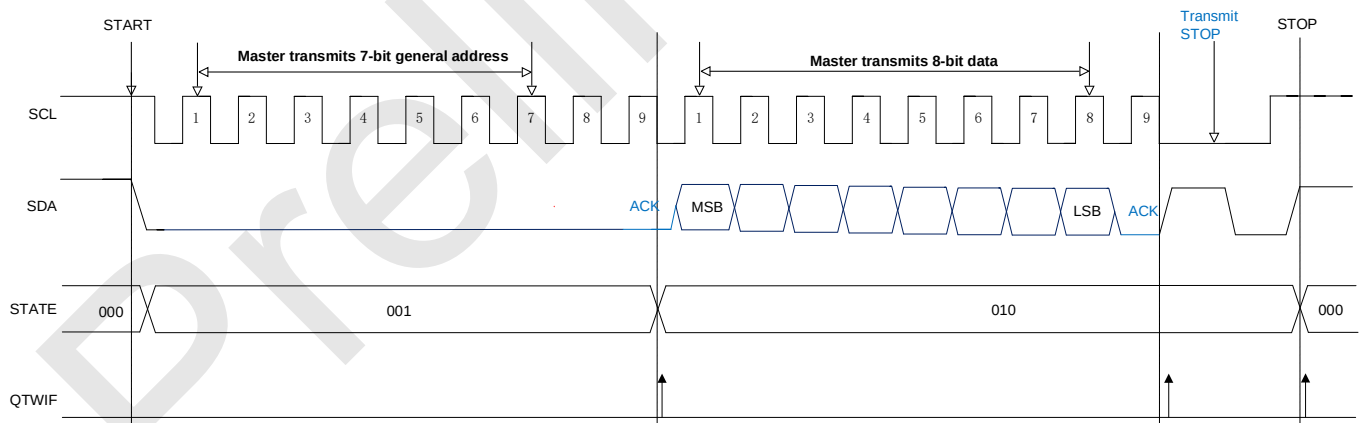


● Response to general call address

When GC=1, the general call address is enabled. The slave enters the state of receiving the first frame address (STATE[2:0] = 001). If the address bits in the first frame are 0x00, all slaves respond to the master. The master's transmitted R/W bit must be set to Write (0). After reception, all slaves enter the Receive Data state (STATE[2:0] = 010). The master releases the SDA line after every 8 transmitted data bits and reads the SDA line status.

If a slave acknowledges, the master communication can proceed in one of three ways:

1. Continue transmitting data.
2. Repeated start
3. Send a stop signal to terminate communication.



If no slave acknowledges, the SDA line remains idle.

Note: When using the general call address in single-master multi-slave mode, the master's R/W bit must NOT be Read (1); otherwise, all devices except the transmitter will respond on the bus.

15.6 Slave mode operation procedure

1. Configure TWI0/1 control registers TWI0_QSPI0_CON/TWI1_QSPI1_CON or TWI2/3 control registers TWI2_SPI2_CON/TWI3_SPI3_CON with QTWEN = 1 to enable TWI
2. Configure TWI0/1 control registers TWI0_QSPI0_CON/TWI1_QSPI1_CON or TWI2/3 control registers

TWI2_SPI2_CON/TWI3_SPI3_CON.

3. Configure TWI0/1 address registers TWI0_QSPI0_ADD/TWI1_QSPI1_ADD or TWI2/3 address registers TWI2_SPI2_ADD/TWI3_SPI3_ADD.
4. When the slave receives data, wait for the QTWIF interrupt flag bit to be set in TWI0_QSPI0_STS/TWI1_QSPI1_STS or TWI2_SPI2_STS/TWI3_SPI3_STS. The QTWIF flag is set after every 8 bits received and must be cleared manually.
5. When the slave transmits data, write the data to be sent into the lower 8 bits of TWI0_QSPI0_QTWIDAT/TWI1_QSPI1_QTWIDAT or TWI2_SPI2_QTWIDAT/TWI3_SPI3_QTWIDAT. TWI will then automatically transmit the data. After every 8 bits, the interrupt flag QTWIF is set.

15.7 Master operation mode

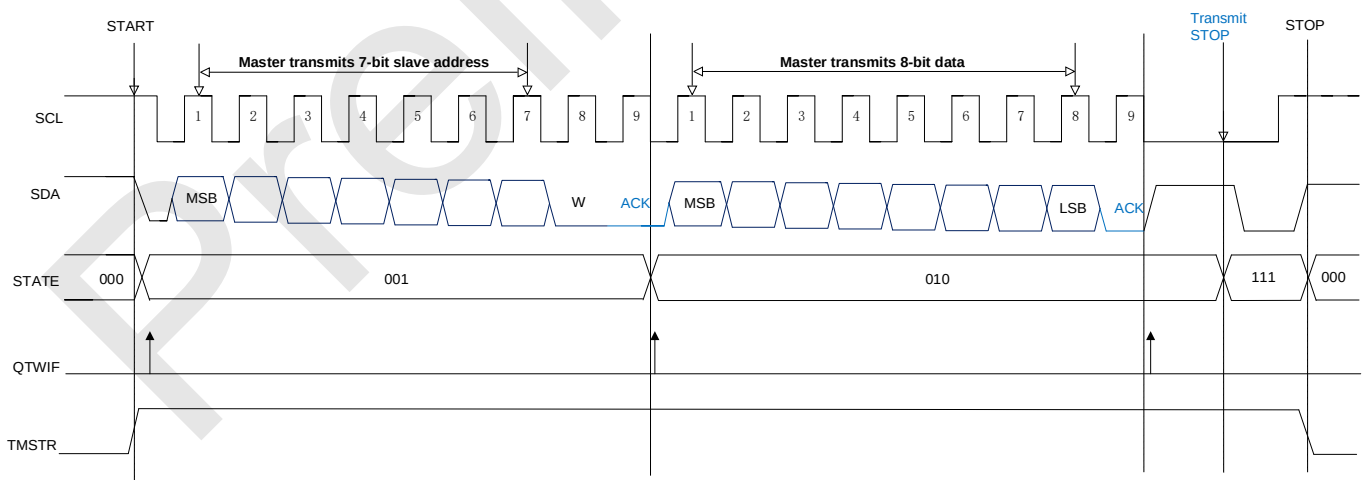
● Mode activation

When the TWI interface issues a start condition on the bus, it automatically switches to master mode. The hardware sets TMSTR to 1 in TWI0_QSPI0/TWI1_QSPI1 or TWI2_SPI2/TWI3_SPI3. The master status bits STATE[2:0] change from 000 to 001, and the interrupt flag QTWIF is set.

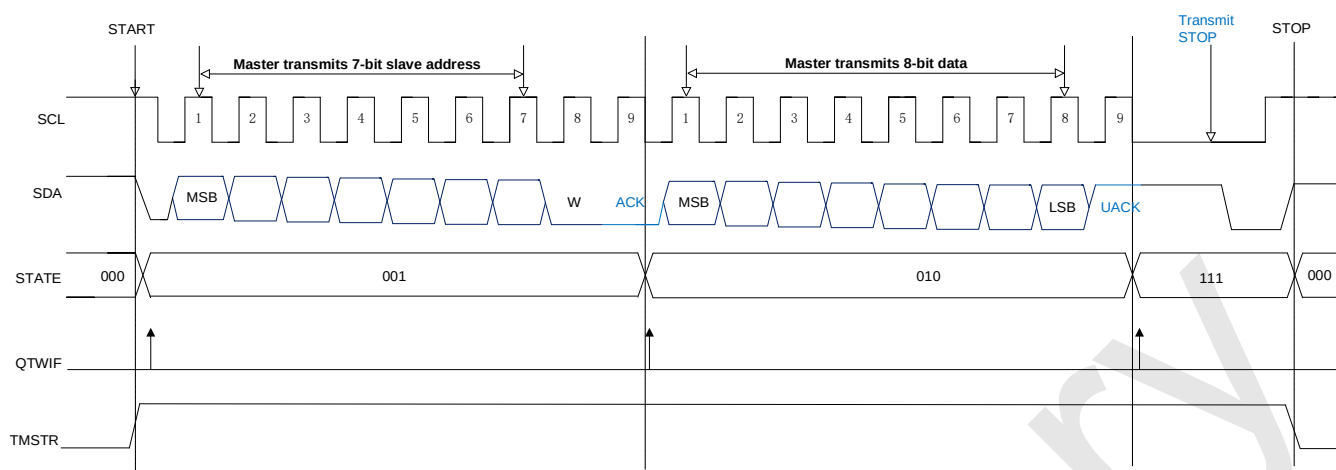
● TWI Master Transmit Mode

In master transmit mode, the first frame sent by the master includes 7 address bits (the selected slave address) and 1 read/write bit (=0 for write). All slaves on the TWI bus receive this first frame. After transmitting the first frame, the master releases the SDA line. The selected slave acknowledges on the 9th SCL clock cycle, then releases the bus and switches to slave receive mode, ready to accept data from the master. After every 8 bits transmitted, the master releases the bus to wait for the slave's ACK during the 9th clock cycle.

If the slave acknowledges low, the master can continue transmitting data or resend the start condition.



If the slave acknowledges high, it indicates the slave will terminate the transmission after the current byte, no longer receiving data. The master's STATE[2:0] changes from 010 (sending data) to 100.

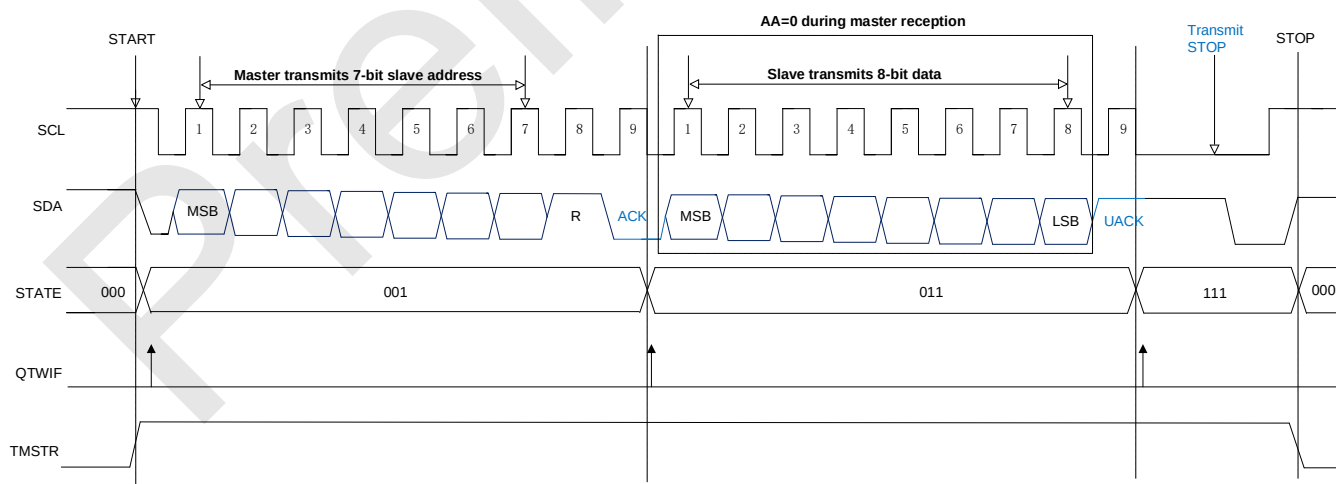


● TWI Master Receive Mode

In master transmit mode, the first frame sent by the master includes a 7-bit slave address and a 1-bit read/write flag (=1 for read). All slaves on the TWI bus receive this first frame. After sending it, the master releases the SDA line. The selected slave sends an acknowledgment on the 9th SCL clock cycle, then takes control of the bus to transmit data to the master. After transmitting each 8-bit data, the slave releases the bus and waits for the master's acknowledgment. The master receives an ACK after successful slave address matching and starts receiving slave data (STATE=011):

1. If AA=1, the master sends an ACK after each received byte and sets QTWIF
2. If AA=0 before the last byte, the master sends a UACK after receiving it, then can transmit a stop signal.

In master receive mode, the bus is actively released as follows:



15.8 Master mode operation procedure

1. Configure TWI0/TWI1 control registers TWI0_QSPI0_CON/TWI1_QSPI1_CON or TWI2/3 control registers TWI2_SPI2_CON/TWI3_SPI3_CON by setting QTWEN = 1 to enable TWI.
2. Configure TWI0/1 control registers TWI0_QSPI0_CON/TWI1_QSPI1_CON to set TWI0/1

communication speed (QWCK[3:0]) and set the START bit. Or configure TWI2/3 control registers SPI2_TWI2_CON/SPI3_TWI3_CON to set TWI2/3 communication speed (QWCK[3:0]) and set the START bit.

3. Configure TWI0/1 data registers TWI0_QSPI0_DATA/TWI1_QSPI1_DATA by writing the slave address with the read/write bit to QWIDAT to send the address frame over the bus. Or configure TWI2/3 data registers TWI2_SPI2_DATA/TWI3_SPI3_DATA by writing the slave address with the read/write bit to QWIDAT to send the address frame over the bus.
4. When the master receives data, wait for the QWIF interrupt flag in TWI0_QSPI0_STS/TWI1_QSPI1_STS or in TWI2_SPI2_STS/TWI3_SPI3_STS to be set. The QWIF interrupt flag is set each time the master receives 8 bits of data. This flag must be cleared manually.
5. When the master transmits data, write the data to QWIDAT; TWI automatically transmits it. The QWIF interrupt flag is set after every 8 bits transmitted.
6. Upon completion of data reception and transmission, the master can send a stop condition (STOP=1) to reset the master state to 000, or send a repeated start to initiate a new transfer.

Note: After the Master generates a STOP condition, the Master's QWIF flag will not be set.

15.9 TWI0/1 interrupts

For TWI0/1, interrupts are triggered by the events below. All TWI events share a common interrupt flag.

Interrupt Event	Event Flag	Interrupt Request Control Bit
Master Mode: after transmitting the start condition	QWIF	TWI0_QSPI0_IDE ->INTEN TWI1_QSPI1_IDE ->INTEN
Master Mode: after transmitting the address frame		
Master Mode: after transmitting or receiving the data frame		
Slave Mode: upon successful first-frame address match		
Slave Mode: upon successful reception or transmission of 8-bit data		
Slave Mode: upon receiving a repeated start condition		
Slave Mode: upon receiving a stop condition		

15.10 TWI2/3 interrupts

For TWI2/3, interrupts are triggered by the events below. All TWI events share a common interrupt flag.

Interrupt Event	Event Flag	Interrupt Request Control Bit
Master Mode: after transmitting the start condition	QWIF	TWI2_SPI2_IDE ->INTEN TWI3_SPI3_IDE ->INTEN
Master Mode: after transmitting the address frame		
Master Mode: after transmitting or receiving the data frame		
Slave Mode: upon successful first-frame address match		

Interrupt Event	Event Flag	Interrupt Request Control Bit
Slave Mode: upon successful reception or transmission of 8-bit data		
Slave Mode: upon receiving a repeated start condition		
Slave Mode: upon receiving a stop condition		

15.11 TWI0/1 registers

15.11.1 TWI0/1 related register table

15.11.1.1 TWI_QSPI control register TWIn_QSPIn_CON

Register	Read/Write	Description	Reset Value
TWIn_QSPIn_CON (n=0/1)	R/W	TWIN_QSPI Control Register	0x0000_0000

31	30	29	28	27	26	25	24
LMODE[1:0]		DWIDTH[1:0]		SSHIFT	STRREN	STRTEN	STRDIR
23	22	21	20	19	18	17	16
MODE[1:0]		RW	CPOL	CPHA	DORD	CLKONLY	SMSTR
15	14	13	12	11	10	9	8
-	-	-	-	QWCK[3:0]			
7	6	5	4	3	2	1	0
QWEN	-	START	STOP	-	-	AA	STRETCH

Bit Number	Bit Symbol	Description
23 ~ 22	MODE[1:0]	TWI/QSPI Mode Selection Bit 00: TWI Mode 01: Full-Duplex SPI Mode 10: Half-Duplex QSPI Mode 11: Pass-Through Mode
11 ~ 8	QWCK[3:0]	TWI Clock Prescale Control Bit 0000: f _{PCLK} 0001: f _{PCLK} /2 0010: f _{PCLK} /4 0011: f _{PCLK} /8 0100: f _{PCLK} /16 0101: f _{PCLK} /32 0110: f _{PCLK} /64 0111: f _{PCLK} /128 1000: f _{PCLK} /256 1001: f _{PCLK} /512 1010: f _{PCLK} /1024 1011: f _{PCLK} /2048 1100: f _{PCLK} /4096 Others: f _{PCLK} /4096 Note: To ensure correct communication for SC32F11T/11G Series TWI0/1, select a communication frequency below 1 MHz.
7	QWEN	TWI_QSPI Module Enable Control Bit 0: Disable the module 1: Enable the module

Bit Number	Bit Symbol	Description
5	START	TWI Start Condition Trigger Setting this bit to 1 generates a start condition, switching TWI to Master Mode. This bit can be set or cleared by software, or cleared by hardware after the start condition is issued.
4	STOP	TWI Stop Condition Trigger In Master Mode, setting this bit to 1 generates a stop condition after the current byte transfer or start condition. This bit can be set or cleared by software, or cleared by hardware upon detecting the stop condition.
1	AA	TWI Acknowledge Enable Bit 0: No acknowledge, returns UACK (ack bit is high level) 1: Returns an ACK after receiving a matching address or data
0	STRETCH	TWI Clock Stretching Enable Bit This bit is effective only in Slave Mode 0: Clock stretching disabled 1: Clock stretching enabled; the Master must support this feature Description: Clock stretching occurs after data transfer completion when ACK is 0.
15 ~ 12 6 3 ~ 2	-	Reserved

15.11.1.2 TWI_QSPI flag status register TWIn_QSPIn_STS

Register	Read/Write	Description	Reset Value
TWIn_QSPIn_STS (n=0/1)	R/W	TWI_QSPI Flag Status Register	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
BUSY	TMSTR	GCA	TXnE/RXnE	-	STATE[2:0]		
7	6	5	4	3	2	1	0
WCOL	DLUFIF	TXHIF	RXHIF	RXFIF	TXEIF	RXNEIF	QTWIF

Bit Number	Bit Symbol	Description
23 ~ 16	NBYTES[7:0]	Transmit and Receive Buffer Byte Count Setting Bit Sets the number of bytes to be transmitted or received. NBYTES decrements automatically by 1 after each successful transmit/receive. When NBYTES reaches 0, the TC flag is set. Note: Cannot be modified when STA is set.
15	BUSY	Busy Status Bit Set by hardware during data transmission; automatically cleared when transmission completes. Note: This bit is valid only in Master mode. In continuous transmit/receive mode, this bit remains set until transmission finishes, then clears automatically.
14	TMSTR	TWI Master/Slave Mode Flag Bit 0: Slave Mode 1: Master Mode Description:

Bit Number	Bit Symbol	Description
		1. When the TWI interface issues a START condition on the bus, it automatically switches to Master Mode, and the hardware sets this bit to 1. 2. When a STOP condition is detected on the bus, the hardware clears this bit.
13	GCA	TWI General Call Address Acknowledge Flag 0: No response to General Call Address 1: When GC is enabled (set to 1) and the General Call Address matches, this bit is set by hardware and automatically cleared.
12	TXnE/RXnE	TWI Transfer Complete Flag In the following cases, TXnE/RXnE flags are set to 1 by hardware. ● Master Mode ■ Master sends address frame (write) and receives ACK from slave ■ Master completes data transmission and receives ACK from slave ■ Master receives data and sends ACK to slave ● Slave Mode ■ Slave receives address frame with read flag matching slave address (TWA) ■ Slave received data and acknowledged the master ■ Slave completed data transmission and received master ACK (AA=1) This bit is cleared by hardware after read/write access to TWIDAT.
10 ~ 8	STATE[2:0]	TWI status bits Indicates TWI status; meanings differ between master and slave modes. ● Slave Mode 000: Slave is idle, waiting for QTWEN to be set and detecting TWI start condition. The slave returns to this state after receiving a stop condition. 001: Slave is receiving the first frame address and read/write bit (bit 8 indicates read/write: 1 = read, 0 = write). Slave enters this state after detecting a start condition. 010: Slave Receive Data State 011: Slave Transmit Data State 111: Slave in Unexpected State Conditions for the Slave Entering Unexpected State: ■ In Slave Transmit Data State, upon receiving UACK from the Master, it transitions to this state, awaiting a restart or stop condition. ■ While the Slave is in Transmit State, writing AA = 0 forces it into this state, awaiting a restart or stop condition. ■ When the Slave is in Receive State and acknowledges with UACK, it transitions to this state, awaiting a restart or stop condition. ■ If the Slave address mismatches the Master's transmitted address, it transitions to this state, awaiting a new start or stop condition. ● Master Mode 000: State Machine Idle 001: Master Sends Start Condition or Transmits Slave Address 010: Master Transmit Data 011: Master Receive Data 111: Module in Unexpected State The master enters the unexpected state under the following conditions: ■ The state transitions here when the master sends a stop condition or receives a UACK from the slave

Bit Number	Bit Symbol	Description
		■ The state transitions here when the slave address does not match the address sent by the master
0	QTWIF	TWI Interrupt Flag Bit This bit is set by hardware and cleared by writing 1 in software. ● Master Mode ■ Transmit Start Signal ■ Address Frame Transmission Complete ■ Data Frame Transmission or Reception Complete ● Slave Mode ■ First Frame Address Matched Successfully ■ Successfully Received or Transmitted 8-bit Data ■ Repeated Start Condition Received Slave Received Stop Signal
31 ~ 24 11	-	Reserved

15.11.1.3 TWI_QSPI address register TWIn_QSPIn_ADD

Register	Read/Write	Description	Reset Value	Power-on Initial Value
TWIn_QSPIn_ADD (n=0/1)	R/W	TWI_QSPI 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 Symbol	Description
7 ~ 1	QTWADD[6:0]	TWI Address Register QTWADD[6:0] must not be written as all zeros; 00H is reserved for general call addressing. This bit is ineffective in Master Mode.
0	GC	TWI General Call Address Response Enable Bit 0: Disable response to general call address 0x00 1: Enable response to general call address 0x00
31 ~ 8	-	Reserved

15.11.1.4 TWI_QSPI data register TWIn_QSPIn_DATA

Register	Read/Write	Description	Reset Value
TWIn_QSPIn_DATA (n=0/1)	R/W	TWI_QSPI Data Register	0x0000_0000

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

31	30	29	28	27	26	25	24
QTDWIDAT[23:16]							
15	14	13	12	11	10	9	8
QTDWIDAT[15:8]							
7	6	5	4	3	2	1	0
QTDWIDAT[7:0]							

Bit Number	Bit Symbol	Description
7 ~ 0	QTDWIDAT[7:0]	TWI Data FIFO Read: retrieve data from TWI receive FIFO Write: load data into TWI transmit FIFO

15.11.1.5 TWI_QSPI interrupt enable and DMA control register TWIn_QSPIn_IDE

Register	Read/Write	Description	Reset Value
TWIn_QSPIn_IDE (n=0/1)	R/W	TWI_QSPI 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
-	-	-	-	-	-	DLUFIE	QTDWIE
7	6	5	4	3	2	1	0
TXDMAEN	RXDMAEN	TXHIE	RXHIE	RXIE	TBIE	RXNEIE	INTEN

Bit Number	Bit Symbol	Description
7	TXDMAEN	DMA Transmit Channel Enable 0: Disable DMA Transmit 1: Enable DMA Transmit Enabling this bit allows TXnE to trigger DMA transmit requests.
6	RXDMAEN	DMA receive channel enable bit 0: Disable DMA receive function 1: Enable DMA receive function Enabling this bit allows RXnE to trigger DMA receive requests.
0	INTEN	CPU Interrupt Request Enable Control Bit 0: Disable interrupt request 1: Enable interrupt request
31 ~ 10	-	Reserved

15.11.2 TWI0/1 register mapping

Register	Offset Address	Read/Write	Description	Reset Value	Access Restriction
TWI0_QSPI0 base address: 0x4002_0040					
TWI0_QSPI0_CON	0x00	R/W	TWI0_QSPI0 control register	0x0000_0000	-
TWI0_QSPI0_STS	0x04	R/W	TWI0_QSPI0 flag status register	0x0000_0000	-
TWI0_QSPI0_ADD	0x08	R/W	TWI0_QSPI0 Address Register	0x0000_0000	-

Register	Offset Address	Read/Write	Description	Reset Value	Access Restriction
TWI0_QSPI0_DATA	0x0C	R/W	TWI0_QSPI0 Data Register	0x0000_0000	Byte access not supported Half-word access
TWI0_QSPI0_IDE	0x10	R/W	TWI0_QSPI0 Interrupt Enable and DMA Control Register	0x0000_0000	-

Register	Offset Address	Read/Write	Description	Reset Value	Access Restriction
TWI1_QSPI1 Base Address: 0x4002_1040					
TWI1_QSPI1_CON	0x00	R/W	TWI1_QSPI1 Control Register	0x0000_0000	-
TWI1_QSPI1_STS	0x04	R/W	TWI1_QSPI1 Flag Status Register	0x0000_0000	-
TWI1_QSPI1_ADD	0x08	R/W	TWI1_QSPI1 Address Register	0x0000_0000	-
TWI1_QSPI1_DATA	0x0C	R/W	TWI1_QSPI1 Data Register	0x0000_0000	Byte access not supported Half-word access
TWI1_QSPI1_IDE	0x10	R/W	TWI1_QSPI1 Interrupt Enable and DMA Control Register	0x0000_0000	-

15.12 TWI2/3 registers

15.12.1 TWI2/3 related register table

15.12.1.1 TWI_SPI control register TWIn_SPIn_CON

Register	Read/Write	Description	Reset Value
TWIn_SPIn_CON (n=2/3)	R/W	TWI_SPI Control Register	0x0000_0000

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

Bit Number	Bit Symbol	Description
23 ~ 22	MODE[1:0]	TWI_SPI Mode Selection Bit 00: TWI Mode 01: SPI Mode 10: Reserved 11: Reserved

Bit Number	Bit Symbol	Description															
14 ~ 13	SPOS[1:0]	TWI2 Signal Port Mapping Control Bits															
		<table><tr><th>Signal SPOS Value</th><th>SCL2</th><th>SDA2</th></tr><tr><td>SPOS[1:0]=00</td><td>PA0</td><td>PA1</td></tr><tr><td>SPOS[1:0]=01</td><td>PA12</td><td>PA11</td></tr><tr><td>SPOS[1:0]=10</td><td>PB9</td><td>PB10</td></tr><tr><td>SPOS[1:0]=11</td><td>PC9</td><td>PC10</td></tr></table>	Signal SPOS Value	SCL2	SDA2	SPOS[1:0]=00	PA0	PA1	SPOS[1:0]=01	PA12	PA11	SPOS[1:0]=10	PB9	PB10	SPOS[1:0]=11	PC9	PC10
		Signal SPOS Value	SCL2	SDA2													
		SPOS[1:0]=00	PA0	PA1													
		SPOS[1:0]=01	PA12	PA11													
		SPOS[1:0]=10	PB9	PB10													
		SPOS[1:0]=11	PC9	PC10													
		TWI3 Signal Port Mapping Control Bits															
		<table><tr><th>Signal SPOS Value</th><th>SCL3</th><th>SDA3</th></tr><tr><td>SPOS[1:0]=00</td><td>PA2</td><td>PA3</td></tr><tr><td>SPOS[1:0]=01</td><td>PA7</td><td>PA8</td></tr><tr><td>SPOS[1:0]=10</td><td>PB5</td><td>PB6</td></tr><tr><td>SPOS[1:0]=11</td><td>PB13</td><td>PB12</td></tr></table>	Signal SPOS Value	SCL3	SDA3	SPOS[1:0]=00	PA2	PA3	SPOS[1:0]=01	PA7	PA8	SPOS[1:0]=10	PB5	PB6	SPOS[1:0]=11	PB13	PB12
		Signal SPOS Value	SCL3	SDA3													
SPOS[1:0]=00	PA2	PA3															
SPOS[1:0]=01	PA7	PA8															
SPOS[1:0]=10	PB5	PB6															
SPOS[1:0]=11	PB13	PB12															
11 ~ 8	QTWCK[3:0]	SPI Clock Prescaler Control Bit 0000:f _{PCLK} 0001:f _{PCLK} /2 0010:f _{PCLK} /4 0011:f _{PCLK} /8 0100:f _{PCLK} /16 0101:f _{PCLK} /32 0110:f _{PCLK} /64 0111:f _{PCLK} /128 1000:f _{PCLK} /256 1001:f _{PCLK} /512 1010:f _{PCLK} /1024 1011:f _{PCLK} /2048 1100:f _{PCLK} /4096 Others: f _{PCLK} 1 / 4096 Note: To ensure proper operation of SC32F11T/11G Series TWI2/3 communication, keep the frequency below 1 MHz.															
7	QTWEN	TWI_SPI Enable Control Bits 0: Disable the module 1: Enable the module															
5	START	TWI Start Condition Trigger Setting this bit to 1 generates a start condition, switching TWI to Master Mode. This bit can be set or cleared by software, or cleared by hardware after the start condition is issued.															
4	STOP	TWI Stop Condition Trigger In Master Mode, setting this bit to 1 generates a stop condition after the current byte transfer or start condition. This bit can be set or cleared by software, or cleared by hardware upon detecting the stop condition.															
1	AA	TWI Acknowledge Enable Bit 0: No acknowledge, returns UACK (ack bit is high level) 1: Returns an ACK after receiving a matching address or data															
0	STRETCH	TWI Clock Stretching Enable Bit This bit is effective only in Slave Mode 0: Clock stretching disabled 1: Clock stretching enabled; the Master must support this feature Description: Clock stretching occurs after data transfer completion when ACK is 0.															
31 ~ 29 27 ~ 24 21,17 15,12 6	-	Reserved															

Bit Number	Bit Symbol	Description
3 ~ 2		

15.12.1.2 TWI_SPI flag status register TWIn_SPIn_STS

Register	Read/Write	Description	Reset Value
TWIn_SPIn_STS (n=2/3)	R/W	TWI_SPI Flag Status Register	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
-	TMSTR	GCA	TXnE/RXnE	-	STATE[2:0]		
7	6	5	4	3	2	1	0
WCOL	-	-	-	-	TXEIF	-	QTWIF

Bit Number	Bit Symbol	Description
23 ~ 16	NBYTES[7:0]	Transmit and Receive Buffer Byte Count Setting Bit Sets the number of bytes to be transmitted or received. NBYTES decrements automatically by 1 after each successful transmit/receive. When NBYTES reaches 0, the TC flag is set. Note: Cannot be modified when STA is set.
14	TMSTR	TWI Master/Slave Mode Flag Bit 0: Slave Mode 1: Master Mode Description: 1. When the TWI interface issues a START condition on the bus, it automatically switches to Master Mode, and the hardware sets this bit to 1. 2. When a STOP condition is detected on the bus, the hardware clears this bit.
13	GCA	TWI General Call Address Acknowledge Flag 0: No response to General Call Address 1: When GC is set and general call address matches, this bit is set by hardware and cleared automatically.
12	TXnE/RXnE	TWI Transfer Complete Flag In the following cases, TXnE/RXnE flags are set to 1 by hardware. - Master Mode - Master sends address frame (write) and receives ACK from slave - Master completes data transmission and receives ACK from slave - Master receives data and sends ACK to slave - Slave Mode - Slave receives address frame with read flag matching slave address (TWA) - Slave received data and acknowledged the master - Slave completed data transmission and received master ACK (AA=1) This bit is cleared by hardware after read/write access to TWIDAT.
10 ~ 8	STATE[2:0]	TWI status bits Indicates TWI status; meanings differ between master and slave modes. Slave Mode

Bit Number	Bit Symbol	Description
		000: Slave idle, waiting for QTWEN to be set to detect TWI start signal. After receiving a stop condition, the slave returns to this state. 001: Slave receiving first address frame and read/write bit (bit 8 is read/write, 1 = read, 0 = write). Slave enters this state upon detecting start condition. 010: Slave Receive Data State 011: Slave Transmit Data State 111: Slave in Unexpected State Conditions for the Slave Entering Unexpected State: ■ In the slave transmit data state, when the master returns a UACK, it transitions to this state, waiting for a restart or stop condition. ■ When the slave is in the transmit state, writing 0 to AA causes a transition to this state, waiting for a restart or stop condition. ■ When the Slave is in Receive State and acknowledges with UACK, it transitions to this state, awaiting a restart or stop condition. ■ If the Slave address mismatches the Master's transmitted address, it transitions to this state, awaiting a new start or stop condition. ● Master Mode 000: State Machine Idle 001: Master Sends Start Condition or Transmits Slave Address 010: Master Transmit Data 011: Master Receive Data 111: Module in Unexpected State The master enters the unexpected state under the following conditions: ■ A transition to this state occurs when the master sends a stop condition or receives a UACK from the slave. If the slave address does not match the address sent by the master, the device transitions to this state.
0	QTWIF	TWI data transfer flag This bit is set by hardware and cleared by writing 1 via software. It indicates whether the current SPI data transfer is complete: 0: Transfer incomplete/in progress 1: Transfer complete
31 ~ 24 15,11 6 ~ 3 1	-	Reserved

15.12.1.3 TWI_SPI address register TWIn_SPIn_ADD

Register	Read/Write	Description	Reset Value	Power-on Initial Value
TWIn_SPIn_ADD (n=2/3)	R/W	TWI_SPI 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 Symbol	Description
7 ~ 1	QTWADD[6:0]	TWI Address Register QTWADD[6:0] must not be written as all zeros; 00H is reserved for general call addressing. This bit is ineffective in Master Mode.
0	GC	TWI General Call Address Response Enable Bit 0: Disable response to general call address 0x00 1: Enable response to general call address 0x00
31 ~ 8	-	Reserved

15.12.1.4 TWI_SPI data register TWIn_SPIn_DATA

Register	Read/Write	Description	Reset Value
TWIn_SPIn_DATA (n=2/3)	R/W	TWI_SPI 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
QTWIDAT[15:8]							
7	6	5	4	3	2	1	0
QTWIDAT[7:0]							

Bit Number	Bit Symbol	Description
7 ~ 0	QTWIDAT[7:0]	TWI Data FIFO Read operation: read received data from the TWI receive buffer Write operation: write data to be transmitted to the TWI transmit buffer
31 ~ 16	-	Reserved

15.12.1.5 TWI_SPI interrupt enable and DMA control register TWIn_SPIn_IDE

Register	Read/Write	Description	Reset Value
TWIn_SPIn_IDE (n=2/3)	R/W	TWI_SPI 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
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
TXDMAEN	RXDMAEN	-	-	-	TBIE	-	INTEN

Bit Number	Bit Symbol	Description
7	TXDMAEN	DMA Transmit Channel Enable 0: Disable DMA Transmit 1: Enable DMA Transmit Enabling this bit allows TXnE high level to trigger a DMA channel transmit request
6	RxDMAEN	DMA receive channel enable bit 0: Disable DMA receive function 1: Enable DMA receive function Enabling this bit allows RXnE high level to trigger a DMA channel receive request
0	INTEN	CPU Interrupt Request Enable Control Bit 0: Disable interrupt request 1: Enable interrupt request
31 ~ 8 5 ~ 3 1	-	Reserved

15.12.2 TWI2/3 register mapping

Register	Offset Address	Read/Write	Description	Reset Value	Access Restriction
TWI2_SPI2 Base Address: 0x4002_0060					
TWI2_SPI2_CON	0x00	R/W	TWI2_SPI2 Control Register	0x0000_0000	-
TWI2_SPI2_STS	0x04	R/W	TWI2_SPI2 Flag Register	0x0000_0000	-
TWI2_SPI2_ADD	0x08	R/W	TWI2_QSPI2 Address Register	0x0000_0000	-
TWI2_SPI2_DATA	0x0C	R/W	TWI2_SPI2 Data Register	0x0000_0000	Byte access not supported Half-word access
TWI2_SPI2_IDE	0x10	R/W	TWI2_SPI2 Interrupt Enable and DMA Control Register	0x0000_0000	-

Register	Offset Address	Read/Write	Description	Reset Value	Access Restriction
TWI3_SPI3 Base Address: 0x4002_1060					
TWI3_SPI3_CON	0x00	R/W	TWI3_SPI3 Control Register	0x0000_0000	-
TWI3_SPI3_STS	0x04	R/W	TWI3_SPI3 Flag Register	0x0000_0000	-
TWI3_SPI3_ADD	0x08	R/W	TWI3_SPI3 Address Register	0x0000_0000	-
TWI3_SPI3_DATA	0x0C	R/W	TWI3_SPI3 Data Register	0x0000_0000	Byte access not supported Half-word access
TWI3_SPI3_IDE	0x10	R/W	TWI3_SPI3 Interrupt Enable and DMA Control Register	0x0000_0000	-

16 CAN communication port

16.1 Overview

The Controller Area Network (CAN) in the SC32F11T/11G Series supports communication protocols CAN2.0B and CAN_FD. Compared to CAN2.0B, CAN_FD offers greater flexibility with a variable bit rate- originally fixed at 1 Mbit/s- and supports data segments up to 64 bytes in length. Supports four different working modes, including configurable low power standby mode, and supports wake up from standby.

The transmit buffer provides two options: PTB Primary Transmission Buffer and STB Secondary Transmission Buffer. Transmission order can be determined using either FIFO mode or priority mode. The receive buffer can store 8 frames simultaneously, with each received frame having its own time-stamp. There are 8 groups of receive filters available; each filter can be enabled independently and configured with filtering conditions.

16.2 Clock source

AHB clock bus uses HCLK as its clock source.

16.3 Feature

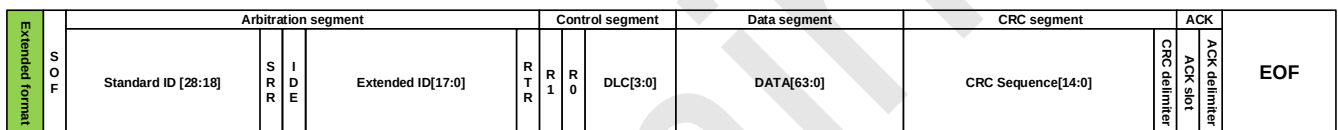
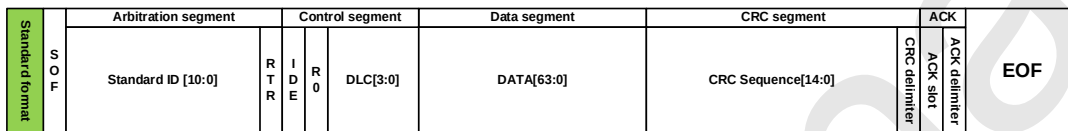
- Protocol support
 - CAN 2.0B
 - ◆ Supports standard and extended formats, with up to 8 bytes of data payload
 - ◆ Supports data rates up to 1 Mbit/s
 - CAN FD
 - ◆ Supports standard and extended formats, with up to 64 bytes of data payload
 - ◆ Variable rate
- Standby Mode: When enabled, the CAN transceiver enters a low power state, stops receiving data frames, and only detects dominant levels on the CAN bus.
- Time-Stamping
 - CiA 603: 64-bit time-stamping; transmit frames (TTS) have a single time-stamp stored in a register, while every receive frame (RTS) has its own individual time-stamp.
- Transmit and Receive Buffers
 - The message buffer area is located in SRAM, and its address is configurable.
 - There are 8 Receive Buffers (RB). Each received frame has an independent time-stamp stored together with the data in the RB. The RB operates as a FIFO.
 - There are 9 Transmit Buffers (TB).
 - ◆ 1 Primary Transmit Buffer (PTB)
 - ◆ 8 Secondary Transmit Buffers (STB), supporting two transmission modes: FIFO mode and priority-based mode.
 - 8 Receive Filters (supports 29-bit ID).

16.4 CAN protocol

16.4.1 Can2.0 protocol

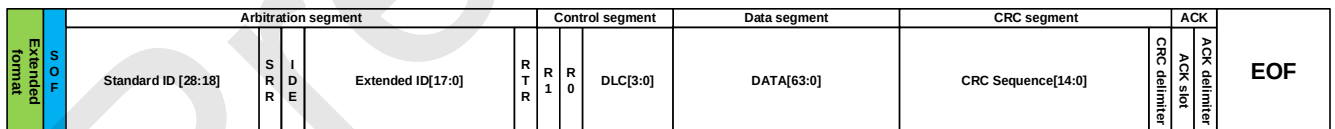
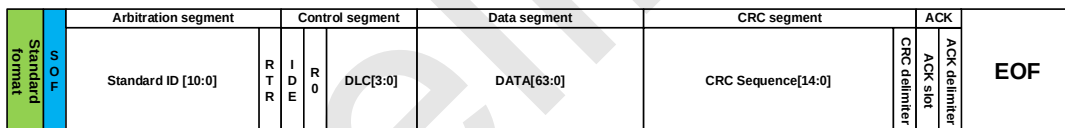
The CAN2.0 traditional protocol has the following features:

- Frame Start Segment, Arbitration Segment, Control Segment, Data Segment, CRC Segment, ACK Segment, Frame End Segment
- ID Identifier supports Standard or Extended Frame format
- Can transmit Data Frame or Remote Frame
- Maximum data length is 8 Bytes
- Bitrate is fixed by low-speed bit-rate register settings
- CRC checksum length is 15 bits



16.4.1.1 Frame start segment

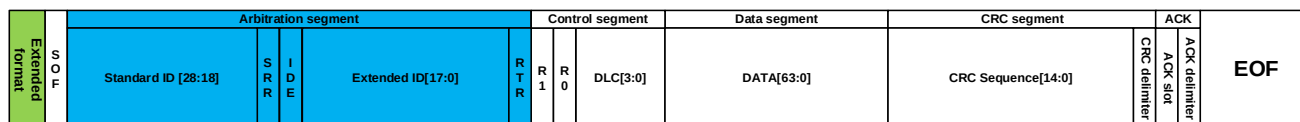
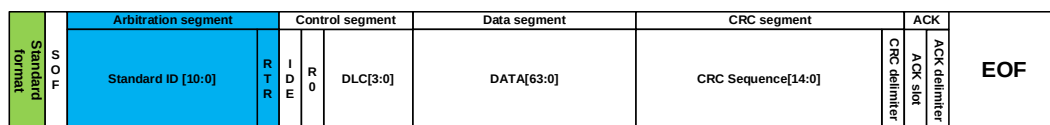
In the CAN2.0 protocol, the SOF dominant bit indicates the start of a message and serves as synchronization on the bus.



16.4.1.2 Arbitration segment

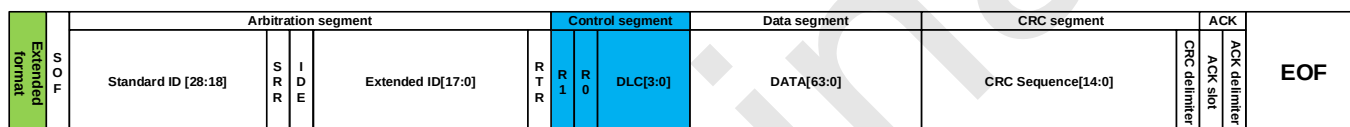
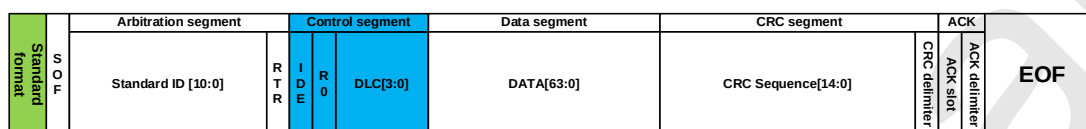
In the CAN 2.0 protocol, the arbitration segment includes the ID segment, the RTR remote frame Control Bit, and the IDE extension bit (for extended frames). Users can enable or disable the Identifier extension bit by setting the IDE.

In the arbitration segment, the ID identifier determines priority. When multiple nodes request bus access simultaneously, a lower ID value has higher priority, following the rule that dominant bits override recessive bits. When IDs are identical, a data frame with RTR dominant has higher priority than a remote frame with RTR recessive. Additionally, the ID identifier can serve as a filter criterion in filtering functions.



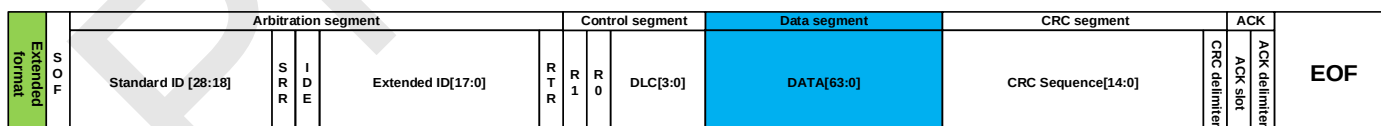
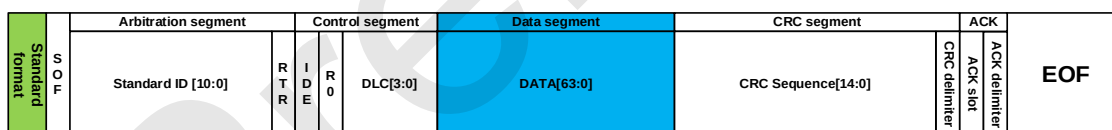
16.4.1.3 Control segment

In the CAN2.0 protocol, the control segment includes the IDE identifier extension bit (for standard frames), reserved bits, and DLC data length bits. The maximum configurable data length in a CAN2.0 frame is 8 bytes.



16.4.1.4 Data frame and remote frame

In the CAN 2.0 protocol, the RTR bit controls whether the transmitted message contains a data frame or a remote frame. When a data frame is selected, the user writes the data to the data segment and transmits it through the standard process. When a remote frame is selected, a node requesting data sends a remote frame to prompt another node to transmit a data frame with the same ID. The RTR bit in a remote frame is recessive, whereas in a data frame, it is dominant. Remote frames do not contain a data segment; their DLC indicates the requested data frame's data length.



16.4.1.5 CRC segment

The CRC segment contains the cyclic redundancy check sequence and a delimiter. In CAN 2.0, the CRC sequence is calculated with a specific polynomial and covers the frame start, arbitration, control, and data segments. Both sender and receiver compute this value using the same algorithm. If the values differ, a communication error has occurred. The CRC delimiter is a single recessive bit following the CRC sequence, used to separate the CRC segment from the ACK segment.

Standard format	Arbitration segment				Control segment		Data segment		CRC segment		ACK		EOF
	Standard ID [10:0]				R	I	R	D	DATA[63:0]	CRC Sequence[14:0]	CRC delimiter	ACK slot	ACK delimiter

S O U R C E	Extended format	Arbitration segment				Control segment			Data segment	CRC segment		ACK	E O F		
		S R R		I D E	Extended ID[17:0]		R T R	R 1	R 0	DLC[3:0]	DATA[63:0]	CRC Sequence[14:0]		CRC delimiter	ACK slot
		Standard ID [28:18]													

16.4.1.6 ACK segment

In CAN 2.0, the ACK segment consists of one ACK bit and one ACK delimiter. The ACK bit is recessive in the transmitting frame; if the receiver acknowledges correct reception, it responds with a dominant ACK bit, indicating successful transmission. The ACK delimiter is always recessive and acts as a separator.

Standard format	Arbitration segment				Control segment		Data segment		CRC segment		ACK		EOF
	Standard ID [10:0]				R	I	R	D	DATA[63:0]	CRC Sequence[14:0]	CRC delimiter	ACK slot	ACK delimiter

Extended format	SOF	Arbitration segment				Control segment			Data segment	CRC segment	ACK		EOF
		Standard ID [28:18]	SRR	IDE	Extended ID[17:0]	RTT	R1	R0	DLC[3:0]	DATA[63:0]	CRC Sequence[14:0]	CRC delimiter	

16.4.1.7 Frame end segment

In CAN 2.0, the frame end segment consists of seven recessive bits, marking the end of the frame.

Standard format	Arbitration segment				Control segment		Data segment		CRC segment		ACK		EOF
	Standard ID [10:0]				R	I	R	D	DATA[63:0]	CRC Sequence[14:0]	CRC delimiter	ACK slot	ACK delimiter

SOL	Extended format	Arbitration segment				Control segment			Data segment	CRC segment	ACK	EOF		
		Standard ID [28:18]		SRR	IDE	Extended ID[17:0]		RTR	R1	R0	DLC[3:0]		DATA[63:0]	CRC Sequence[14:0]

16.4.2 CAN_FD protocol

The CAN_FD protocol, also known as the CAN Flexible Data-Rate protocol, features:

- Compatible with the CAN 2.0 protocol
- Frame start segment, arbitration segment, control segment, data frame, CRC segment, ACK segment, and frame end segment
- ID Identifier supports Standard or Extended Frame format
- Only data frames are allowed for transmission
- Maximum data length is 64 bytes
- Setting the BRS bit allows the data segment to have a higher bit rate than the arbitration segment
- The control segment contains an ESI bit indicating an error state
- CRC calculation includes bit stuffing, with up to 21 CRC check bits

SOF	Standard format	Arbitration segment			Control segment				Data segment		CRC segment			ACK		EOF			
		Standard ID [10:0]			R R S	I D E	E D L	R 0	E S I	DLC[3:0]		Up to 64 bytes		CRC Sequence[16:0] or CRC Sequence[20:0]			CRC delimiter	ACK slot	ACK delimiter

Standard format	SOF	Arbitration segment		Control segment						Data segment	CRC segment	ACK		EOF
		Standard ID [10:0]	R R S	I D E	E D L	R O	B R S	E S I	DLC[3:0]	Up to 64 bytes	CRC Sequence[16:0] or CRC Sequence[20:0]	CRC delimiter	ACK slot	

SOF	Extended format	Arbitration segment					Control segment					Data segment		CRC segment		ACK		EOF
		Standard ID [28:18]		SRR	IDE	Extended ID[17:0]	RRL	EDL	R0	BRS	ESI	DLC[3:0]	Up to 64 bytes	CRC Sequence[16:0] or CRC Sequence[20:0]	CRC delimiter	ACK slot	ACK delimiter	

16.4.2.4 Data frame

Only data frames can be selected in CAN_FD; remote frames are not supported. When selecting a data frame, the user writes the data to the data segment and initiates transmission.

Standard format	SOF	Arbitration segment			Control segment					Data segment	CRC segment		ACK		EOF
		Standard ID [10:0]			R R S	I D E	E D L	R 0	B R S	E S I	DLC[3:0]	Up to 64 bytes	CRC Sequence[16:0] or CRC Sequence[20:0]		

Extended format	SOF	Arbitration segment						Control segment						Data segment	CRC segment		ACK		EOF
		Standard ID [28:18]		SRR	IDE	Extended ID[17:0]		RRS	EDL	R0	BR	ES	I	DLC[3:0]	Up to 64 bytes	CRC Sequence[16:0] or CRC Sequence[20:0]		CRC delimiter	

16.4.2.5 CRC segment

In the CAN_FD protocol, the CRC segment structure is fundamentally the same as in CAN 2.0, but the CRC algorithm has been enhanced. Since CAN_FD supports a maximum data length of 64 bytes, the CRC sequence length ranges from 17 up to 21 bits, depending on the transmitted data length. Additionally, CAN_FD incorporates bit stuffing into the CRC calculation by including stuffed bits, which enhances protocol robustness.

Standard format	SOF	Arbitration segment			Control segment					Data segment		CRC segment		ACK		EOF
		Standard ID [10:0]			R R S	I D E	E D L	R 0	B R S	E S I	DLC[3:0]	Up to 64 bytes	CRC Sequence[16:0] or CRC Sequence[20:0]		CRC delimiter	

Extended format	SOF	Arbitration segment						Control segment					Data segment		CRC segment		ACK		EOF										
		Standard ID [28:18]						S	R	R	I	D	E	Extended ID[17:0]	R	R	I	D		E	R	B	R	E	S	I	DLC[3:0]	Up to 64 bytes	CRC Sequence[16:0] or CRC Sequence[20:0]

16.4.2.6 ACK segment

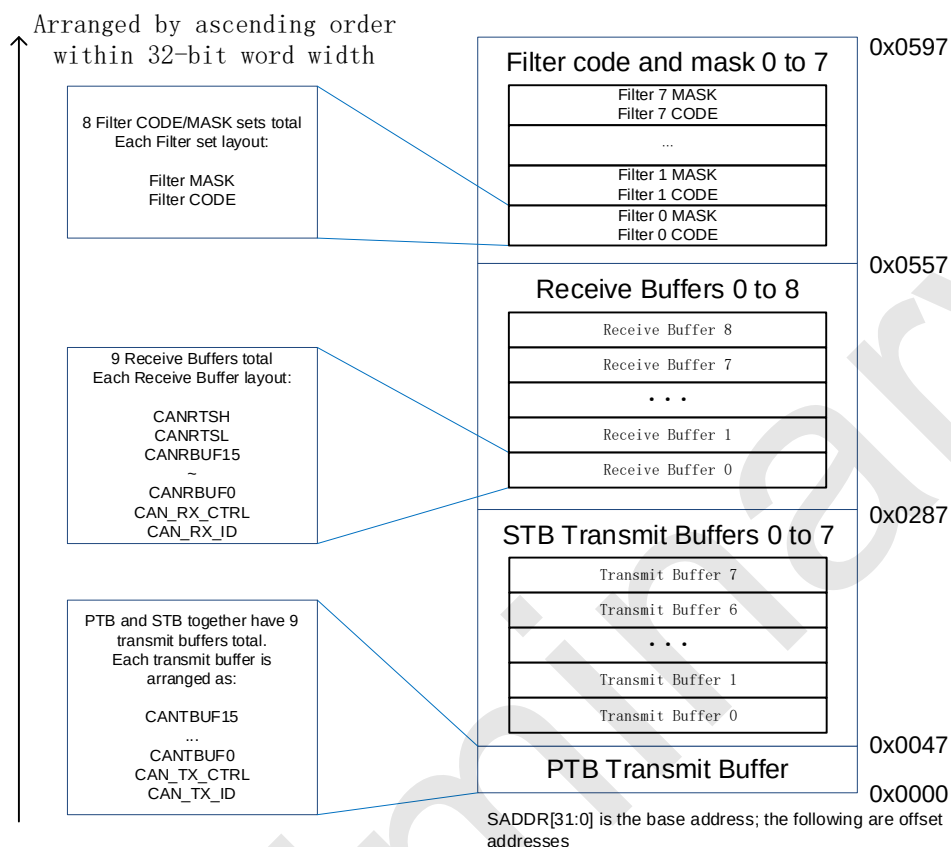
The ACK segment is identical in both CAN_FD and CAN 2.0, consisting of an ACK bit followed by an ACK delimiter. The ACK bit is recessive in the transmitting frame; if the receiver acknowledges correct reception, it responds with a dominant ACK bit, indicating successful transmission. The ACK delimiter is always recessive and acts as a separator.

Extended format	SOL	Arbitration segment				Control segment				Data segment	CRC segment	ACK		EOF
		Standard ID [28:18]	SRR	IDE	Extended ID[17:0]	RSS	EDL	R0	BRS	ESI	DLC[3:0]	Up to 64 bytes	CRC Sequence[16:0] or CRC Sequence[20:0]	

In both CAN_FD and CAN2.0, the frame end segment consists of 7 recessive bits, marking the end of a frame.

S.O.U. Extended format	Arbitration segment				Control segment				Data segment	CRC segment		ACK	EOF	
	Standard ID [28:18]	S R R	I D E	Extended ID[17:0]	R R S	E D L	R O	B R S	E S I	DLC[3:0]	Up to 64 bytes	CRC Sequence[16:0] or CRC Sequence[20:0]		CRC delimiter

Message RAM in the SC32F11T/G series is a RAM area allocated for storing transmit and receive message buffers, as well as the receive CODE and MASK values for 8 filter groups. The CAN module initialization includes defining the Message RAM, which is 32-bit wide. Its start address is set by the SADDR[31:0] field in the CAN_SADDR register. The address range spans from this start address to start address + 0x597, covering 1432 bytes (358 words). The address layout details are as follows:



Message RAM address block diagram

- Offset address 0x0000 to 0x0287 corresponds to the transmit buffer.

When transmitting messages using PTB, data written to CAN_TX_ID, CAN_TX_CTRL, and CAN_TBUF0 through TBUF15 registers is simultaneously transferred as 32-bit units to the PTB transmit buffer. When transmitting messages using STB, messages are written sequentially from transmit buffer 0 to 7. After writing the transmit message to the registers and transferring it to the STB transmit buffer, setting TSNEXT to 1 allows storing the next message in the following transmit buffer.

- Offset address 0x0288 to 0x0557 corresponds to the receive buffer.

Received messages are stored sequentially from receive buffer 0 to 8; however, only eight messages can be accessed via the receive frame registers. For example, in discard oldest mode, after receiving 8 frames, the 9th frame is stored in receive buffer 8. Although the message in receive buffer 0 remains, read operations only access messages in receive buffers 1 through 8. If an additional frame is received, the newest frame overwrites the data in receive buffer 0. Read operations then retrieve messages from receive buffers 0 and 2 through 8.

- Offset address 0x0558 to 0x0597 is allocated for filter code and mask buffers.

After writing filter 0 to 7 code and mask values to the CAN_ACF register, these values are written to the corresponding RAM area.

Note: Before defining the message storage area in RAM, clear the target address region first. After defining the message storage area, ensure the address region is not overwritten by other functions.

16.6 Function description

16.6.1 Baud rate configuration

CAN baud rate is configured via the low-speed register CAN_S_SEG and the high-speed register CAN_F_SEG. In CAN_FD mode, enabling the BRS bit allows different baud rates for the data segment and arbitration segment, increasing data segment transmission speed. The low-speed register CAN_S_SEG sets the arbitration segment baud rate, while the high-speed register CAN_F_SEG sets the data segment baud rate. In CAN2.0 mode, the baud rate of the arbitration Segment and data Segment can only be configured via the low-speed register CAN_S_SEG and does not use the high-speed register CAN_F_SEG.

The baud rate calculation formula is as follows:

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

Where Baud is the baud rate, clock is the CAN clock frequency, SEG1 is phase segment 1 (the time before the sampling point excluding the synchronization segment), SEG2 is phase segment 2 (the time after the sampling point), and PRESC is the prescaler value. The register also requires configuring SJW, the maximum synchronization jump width. SJW does not directly affect the baud rate but adjusts SEG1 and SEG2 widths to compensate for phase differences caused by uncontrollable factors in each message bit. Therefore, SJW must be less than or equal to SEG1.

16.6.2 Filter function

The filter function can be enabled to receive only messages with specific IDs. The filter function is implemented by combining ACODE and AMASK, both configured in the CAN_ACF register. The ID bits are compared bit-by-bit with the values in ACODE; if all bits match, the filter passes the frame. Bits set to 1 in AMASK mask the corresponding ACODE bits during comparison, causing those bits to be ignored. Thus, regardless of whether the masked bits in ID and ACODE differ, the result is considered equal. For extended frames, bits ID28 to ID0 are compared; For standard frames, only bits ID10 to ID0 are compared; bits ID28 to ID11 do not affect the result. For example, with a standard frame format, if ACODE is set to 0x212 and AMASK to 0x000, only messages with ID 0x212 will pass the filter. Setting AMASK to 0x7F0 configures the filter to pass all messages with IDs ending in 2 (i.e., ID format 0xXX2).

Eight filters are available. ID comparison starts from the enabled filter with the lowest number and proceeds sequentially to higher-numbered enabled filters. If a filter matches, the message is received; otherwise, the comparison moves to the next filter. If none match, the message is discarded. Hardware reset is enabled by setting the RESET bit in the CAN_CFG_STAT register to 1. By default, only filter 0 is enabled, accepting all IDs, while other filters remain disabled. Only filter 0 is affected by the power-on reset; all other filters stay uninitialized. Filter selection and enablement are configured in the CAN_ACFCTRL register. The pointers to AMASK and ACODE are also set in CAN_ACFCTRL. When the pointer is set to AMASK, the CAN_ACF register can use the AIDE and AIDEE bits to enable reception of only standard frames or only extended frames. When the pointer is set to ACODE, configuring AIDE and AIDEE bits in the CAN_ACF register has no effect.

16.6.3 Transmit buffer

The CAN module provides two transmit buffers: PTB (Primary Transmit Buffer) and STB (Secondary Transmit Buffer). PTB has one slot, storing one frame, while STB has eight slots, storing up to eight frames.

After PTB writes a full frame of message, transmission must be enabled via TPE.

When one slot of STB is written with a message frame, set TSNEXT to 1, and STB will point to the next slot for storing the next frame. Each time STB switches to a new slot, clear the random number in the related transmission registers to avoid interference with message writing. The number of messages stored in STB can be checked via the TSSTAT bit. Whether full or not, message transmission can be triggered by enabling TSONE or TSALL. TSONE transmits one message frame per STB slot each time, while TSALL transmits all stored messages at once. TSONE generates an interrupt after transmitting each frame, while TSALL generates one interrupt after transmitting all frames in the filled slots. When transmitting data in STB, the transmission order is determined by the TSMODE bit. The mode can be set to FIFO, transmitting frames in the order they were written into STB; or based on ID priority, where frames with lower ID have higher transmission priority.

Because PTB has higher transmission priority than STB, if TPE is enabled to start PTB transmission during STB transmission, PTB will begin immediately after completing the current frame at TPE activation, delaying STB transmission until PTB finishes or TPA stops PTB transmission.

16.6.4 Automatic retransmission mask

The CAN module includes automatic retransmission mask bits for the PTB and STB transmit buffers (TPSS and TSSS bits in CAN_CFG_STAT), each controlling the retransmission function of its respective buffer. By default, the mask bits are disabled, and CAN automatically retransmits frames with errors or arbitration loss. When the automatic retransmission mask is enabled, automatic retransmission for the corresponding buffer is disabled, preventing automatic retransmission of erroneous or lost arbitration frames.

16.7 Operating mode

CAN offers four operating modes: Listen-Only, External Loopback, Internal Loopback, and Standby. The details of each mode are as follows:

- Listen-Only Mode

In Listen-Only Mode, CAN cannot transmit messages on the CAN bus but can receive messages into the receive buffer. It does not send ACKs or trigger receive Interrupts.

When CAN operates in Listen-Only Mode with External Loopback Mode enabled, the MCU can transmit messages on the bus, but only acknowledges and triggers Interrupts for its own transmitted messages; Other bus messages can be stored in the buffer but will not receive ACKs.

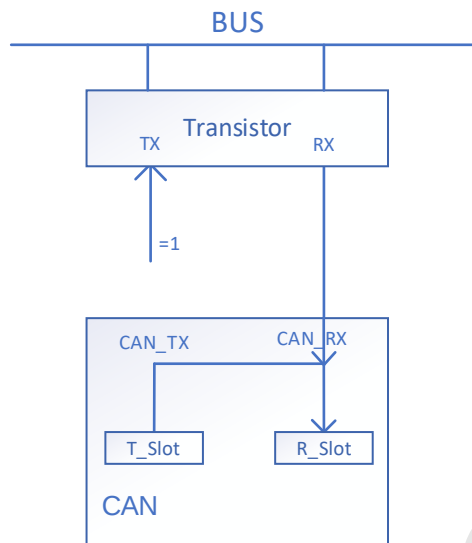


Figure 1.6.1 Listen-Only Mode

- External Loopback Mode

In External Loopback Mode, when the MCU is connected to the CAN bus, it receives both its own transmitted messages and external messages, with normal Interrupt triggering; When the MCU is not connected to the bus, connecting CAN_TX and CAN_RX and setting the SACK bit to 1 causes the MCU to receive its own transmitted message, and interrupts function normally.

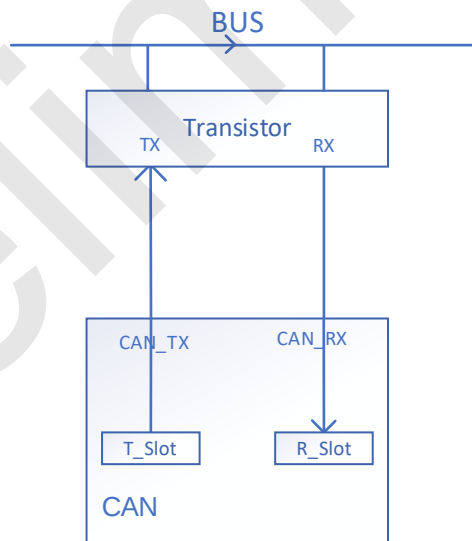


Figure 1.6.2 External Loopback Mode

- Internal Loopback Mode

In internal loopback mode, the CAN controller sends messages internally to the receiver without using the transceiver. The receiver only receives its own transmitted messages and does not observe bus traffic.

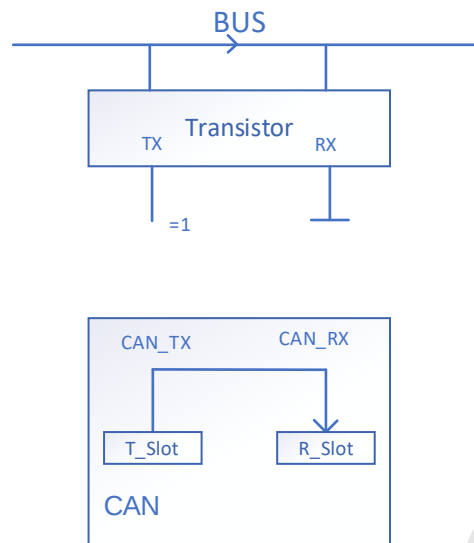


Figure 1.6.3 Internal Loopback Mode

- Standby Mode

When CAN runs in standby mode, the CAN module does not enable transmission after MCU power-on. Upon detecting a dominant level on the bus through any received message, regardless of the filter, CAN exits standby mode and can transmit normally.

16.8 CAN communication configuration

16.8.1 CAN module initialization

Before enabling CAN-related functions on the MCU, the CAN module must be initialized. Initialization steps include:

- Enabling the CAN clock on the AHB bus for clock setup
- Initializing IOs according to functional requirements
- Configuring the CAN message storage area address
- Initializing the CAN structure.

Since the transmit and receive buffers (TBUF and RBUF) are RAM-based, their contents are random after power-on or reset. Receive-related registers are read-only and cannot be cleared. Therefore, users should avoid reading the receive buffer before the CAN module receives any data to prevent reading invalid random values.

16.8.2 Transmit message configuration

To configure a transmit message, the user must clear and set the following registers:

- Configure the baud rate using the low-speed baud rate register CAN_S_SEG and the high-speed baud rate register CAN_F_SEG
- Select the PTB or STB buffer by setting the transmit buffer selection bit TBSEL
- Select CAN2.0 or CAN_FD by setting the CAN_FD mode bit FDF.
- Select a standard or extended frame ID by setting the identifier extension bit IDE
- Select a remote frame or data frame by configuring the remote transmission request bit RTR (this bit

can be ignored in CAN FD mode)

- Set the transmission data length via the data length code (DLC) field
- Load the data to be transmitted into the TBUF buffer.

16.8.3 Transmit process

1. After enabling transmit, the user writes the configured message into the buffer. For the STB buffer, TSNEXT must be enabled after each frame write to indicate the frame configuration is complete, and TBUF will advance to the next slot for writing.
2. After all messages are written, enable PTE to start transmission when PTB is selected, or enable TSONE to transmit frames one by one, or TSALL to transmit all frames simultaneously when using STB.
3. If interrupts are enabled, transmitting a message sets the corresponding interrupt flag and triggers the interrupt service routine; if interrupts are disabled, the flags remain unset.

16.8.4 Receive process

1. Enable TXEN (transmit enable) and RXEN (receive enable) in the CAN_IDE register. Enabling TXEN ensures proper ACK bit response upon reception.
2. Configure the low-speed baud rate register CAN_S_SEG and the high-speed baud rate register CAN_F_SEG.
3. RBUF contains 8 slots, each storing one frame, for a total of 8 frames. Reception supports filters that select frames by specific identifiers. Frames that fail the filter still enter the receive buffer but are overwritten by subsequent frames until a filtered frame arrives and advances RBUF to the next slot.
4. After reading data from the receive buffer RBUF, release the current slot by setting the RREL bit in the CAN_CFG_STAT register. This moves RBUF to the next slot, enabling reading of the next frame.
5. If interrupts are enabled, transmitting a message sets the corresponding interrupt flag and triggers the interrupt service routine; if interrupts are disabled, the flags remain unset.

When RBUF is full and new frames arrive, the ROM bit can be set during initialization to discard either the newest or the oldest frame.

16.8.5 Message transmission priority

Message transmission priority depends on the message identifier; a lower ID indicates a higher priority. When multiple message requests are pending on the bus simultaneously, they are processed according to their priority.

16.9 CAN interrupts

When the CAN global interrupt enable control bit and the corresponding sub-interrupt enable bit are set, an interrupt event sets the corresponding sub-interrupt flag bit and triggers the interrupt service routine. If the sub-interrupt enable bit is not set, the sub-interrupt flag bit cannot be set.

Interrupt Event	Event Flag	Global Interrupt Enable Control Bit	Module Interrupt Sub-Control Bit
Reception Complete	RIF	CAN_IDE_INTEN	CAN_RTIE->RIE
RBUF Full	ROIF		CAN_RTIE -> ROIE
RBUF Overflow	RFIF		CAN_RTIE -> RFIE
RBUF Slot Threshold Reached	RAFIF		CAN_RTIE -> RAFIE

Interrupt Event	Event Flag	Global Interrupt Enable Control Bit	Module Interrupt Sub-Control Bit
PTB Transfer Complete	TPIF		CAN_RTIE -> TPIE
STB Transfer Complete	TSIF		CAN_RTIE -> TSIE
PTB/STB Transfer Abort	AIF		\
Error Interrupt	EIF		CAN_RITE->EIE
Arbitration Lost Error	ALIF		CAN_RTIE -> ALIE
Bus Error	BEIF		CAN_RTIE -> BEIE
Error Passive Interrupt	EPIF		CAN_RTIE -> EPIE

16.10 CAN register

16.10.1 CAN related register table

16.10.1.1 Receive frame ID register CAN_RX_ID - standard frame

Register	Read/Write	Description	Reset Value
CAN_RX_ID	R	Receive Frame ID Register	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 Symbol	Description
31	ESI	Error Status Indicator Flag 0: CAN FD Node Active Error 1: CAN FD Node Passive Error
10 ~ 0	ID[10:0]	Received Standard Frame 11-bit ID Identifier
30 ~ 11	-	Reserved

16.10.1.2 Receive frame ID register CAN_RX_ID - extended frame

Register	Read/Write	Description	Reset Value
CAN_RX_ID	R	Receive Frame ID Register	0xnnnn_nnnn

31	30	29	28	27	26	25	24
ESI	-	-	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 Symbol	Description
31	ESI	Error Status Indicator Flag 0: CAN FD Node Active Error 1: CAN FD Node Passive Error

Bit Number	Bit Symbol	Description
28 ~ 0	ID[28:0]	Received Extended Frame 29-bit ID Identifier
30 ~ 29	-	Reserved

16.10.1.3 Receive frame control and status register CAN_RX_CTRL

Register	Read/Write	Description	Reset Value
CAN_RX_CTRL	R	Receive Frame Control and Status Register	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 Symbol	Description
15 ~ 13	KOER[2:0]	Error Code 000: No Error 001: Bit Error 010: Frame Format Error 011: Bit Stuffing Error 100: Acknowledge Error 101: CRC Error 110: Other errors (dominant bits after the own error flag, active error flag duration excessive, dominant bits during passive error flag after an ACK error) 111: KOER does not update with each new error. It remains unchanged during successful frame transmit or receive.
12	TX	Receive status bit in loopback mode 0: Not received 1: In loopback mode, self-transmitted frame received
7	IDE	Identifier extension 0: Standard mode ID [10:0] 1: Extended mode ID [28:0]
6	RTR	Remote transmission request 0: Data frame 1: Remote frame
5	FDF	CAN FD frame flag 0: CAN 2.0 Frame 1: CAN FD frame
4	BRS	Bit rate switch enable bit 0: Disable bit rate switching 1: Enable bit rate switching

Bit Number	Bit Symbol	Description																														
3 ~ 0	DLC[3:0]	Received frame data byte length The data byte length of a CAN 2.0 received frame is a maximum of 8 bytes and a minimum of 0 bytes. CAN FD receive frames support a maximum data byte length of 64 bytes and a minimum of 0 bytes. The data byte length correspondence between 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>	DLC	Frame Type	Payload in Bytes	0000 to 1000	CAN 2.0 and CAN FD	0 to 8	1001 to 1111	CAN 2.0	8	1001	CAN FD	12	1010	CAN FD	16	1011	CAN FD	20	1100	CAN FD	24	1101	CAN FD	32	1110	CAN FD	48	1111	CAN FD	64
		DLC	Frame Type	Payload in Bytes																												
		0000 to 1000	CAN 2.0 and CAN FD	0 to 8																												
		1001 to 1111	CAN 2.0	8																												
		1001	CAN FD	12																												
		1010	CAN FD	16																												
		1011	CAN FD	20																												
		1100	CAN FD	24																												
		1101	CAN FD	32																												
		1110	CAN FD	48																												
		1111	CAN FD	64																												
31 ~ 16 11 ~ 8	-	Reserved																														

16.10.1.4 Receive frame data registers CAN_RBUF0 to CAN_RBUF15

Register	Read/Write	Description	Reset Value
CAN_RBUF0 ~ CAN_RBUF15	R	Receive frame data registers total 64 bytes	0xnnnnn_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 Symbol	Description
31 ~ 24	Datax[7:0] (x=N*4+4 N=0 to 15)	4th byte of the Nth word in received frame data
23 ~ 16	Datax[7:0] (x=N*4+3 N=0 to 15)	3rd byte of the Nth word in received frame data
15 ~ 8	Datax[7:0] (x=N*4+2 N=0 to 15)	2nd byte of the Nth word in received frame data
7 ~ 0	Datax[7:0] (x=N*4+1 N=0 to 15)	1st byte of the Nth word in received frame data

16.10.1.5 Receive frame time-stamp storage register low (CAN_RTSL)

Register	Read/Write	Description	Reset Value
CAN_RTSL	R	Receive Frame Time-Stamp Storage Register Low	0xnnnnn_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 Symbol	Description
31 ~ 0	RTS[31:0]	Lower 32 bits of the receive frame time-stamp storage. On receiving each frame, the CAN module hardware stores the current receive frame time-stamp timer value into the RTS register per CiA 603 standard. The time-stamp position is set by the TIMEPOS field in CAN_ACFCTRL. Time-stamps can be 32 or 64 bits; unused bits are cleared to zero. The CAN module offers 8 receive frame slots, each with dedicated RTS storage.

16.10.1.6 Receive frame time-stamp storage register high (CAN_RTSH)

Register	Read/Write	Description	Reset Value
CAN_RTSH	R	High bits of the receive frame time-stamping storage register	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 Symbol	Description
31 ~ 0	RTS[63:32]	High 32 bits of the receive frame time-stamping storage register. On receiving each frame, the CAN module hardware stores the current receive frame time-stamp timer value into the RTS register per CiA 603 standard. The time-stamp position is set by the TIMEPOS field in CAN_ACFCTRL. Time-stamps can be 32 or 64 bits; unused bits are cleared to zero. The CAN module provides 8 receive frame slots, each with independent RTS memory.

16.10.1.7 Transmit frame ID register CAN_TX_ID - standard frame

Register	Read/Write	Description	Reset Value
CAN_TX_ID	R/W	Transmit frame ID register	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 Symbol	Description
31	TTSEN	Transmit frame timestamp generation enable bit When enabled, hardware records the transmit frame's timestamp (TTS) following CiA 603. 0: Do not capture transmit timestamp for this frame 1: Enable TTS update
10 ~ 0	ID[10:0]	11-bit ID identifier of the transmitted standard frame
30 ~ 11	-	Reserved

16.10.1.8 Transmit frame ID register CAN_TX_ID - extended frame

Register	Read/Write	Description	Reset Value
CAN_TX_ID	R/W	Transmit frame ID register	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 Symbol	Description
31	TTSEN	Transmit frame timestamp generation enable bit When enabled, hardware records the transmit frame's timestamp (TTS) following CiA 603. 0: Do not capture transmit timestamp for this frame 1: Enable TTS update
28 ~ 0	ID[28:0]	29-bit ID identifier of the transmitted extended frame
30 ~ 29	-	Reserved

16.10.1.9 Transmit frame control and status register CAN_TX_CTRL

Register	Read/Write	Description	Reset Value
CAN_TX_CTRL	R/W	Transmit frame control and status register	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 Symbol	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

Bit Number	Bit Symbol	Description																														
		Only CAN 2.0 frames can be remote frames. CAN FD has no remote frames.																														
5	FDF	CAN 2.0 / CAN FD frame mode bit 0: CAN 2.0 frame (data field up to 8 bytes) 1: CAN FD frame (data field up to 64 bytes)																														
4	BRS	Bit rate switch enable bit: 0: Disable bit rate switching; data and control segments use the same bit rate, operating in standard or slow mode 1: Enable bit rate switching; data segment (including data and CRC fields) transmits at a higher bit rate than the control segment																														
3 ~ 0	DLC[3:0]	Transmit frame data length field The data field length of CAN 2.0 transmit frames has a maximum of 8 bytes and a minimum of 0 bytes. The data length of CAN FD transmit frames ranges from 0 to 64 bytes. The relationship between the data field lengths of 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>	DLC	Frame Type	Payload in Bytes	0000 to 1000	CAN 2.0 and CAN FD	0 to 8	1001 to 1111	CAN 2.0	8	1001	CAN FD	12	1010	CAN FD	16	1011	CAN FD	20	1100	CAN FD	24	1101	CAN FD	32	1110	CAN FD	48	1111	CAN FD	64
DLC	Frame Type	Payload in Bytes																														
0000 to 1000	CAN 2.0 and CAN FD	0 to 8																														
1001 to 1111	CAN 2.0	8																														
1001	CAN FD	12																														
1010	CAN FD	16																														
1011	CAN FD	20																														
1100	CAN FD	24																														
1101	CAN FD	32																														
1110	CAN FD	48																														
1111	CAN FD	64																														
31 ~ 8	-	Reserved																														

16.10.1.10 Transmit frame data registers CAN_TBUF0 to CAN_TBUF15

Register	Read/Write	Description	Reset Value
CAN_TBUF0 ~ CAN_TBUF15	R/W	Total transmit frame data registers: 64 Bytes	0xnnnnn_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 Symbol	Description
31 ~ 24	Datax[7:0] (x=N*4+4 N=0 to 15)	4th byte of the Nth word in transmit frame data
23 ~ 16	Datax[7:0] (x=N*4+3 N=0 to 15)	3rd byte of the Nth word in transmit frame data
15 ~ 8	Datax[7:0] (x=N*4+2 N=0 to 15)	2nd byte of the Nth word in transmit frame data
7 ~ 0	Datax[7:0] (x=N*4+1 N=0 to 15)	1st byte of the Nth word in transmit frame data

16.10.1.11 Transmit frame time-stamp storage register low CAN_TTSL

Register	Read/Write	Description	Reset Value
CAN_TTSL	R	Transmit frame time-stamp storage register low	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 Symbol	Description
31 ~ 0	TTS[31:0]	Receive frame time-stamp storage unit low 32 bits. Transmit frame time-stamp storage unit low 32 bits. Stores the transmit frame time-stamp (TTS). The time master obtains synchronization information (SYNC message) via TTS. When transmit frame time-stamp generation is enabled (TTSEN=1), TTS updates with each transmitted frame. Time-stamps can be 32-bit or 64-bit; unused bits are zeroed.

16.10.1.12 Transmit frame time-stamp storage register high 32 bits CAN_TTSH

Register	Read/Write	Description	Reset Value
CAN_TTSH	R	Transmit frame time-stamp storage register high 32 bits	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 Symbol	Description
31 ~ 0	TTS[63:32]	Receive frame time-stamp storage unit high 32 bits. Transmit frame time-stamp storage unit high 32 bits. Stores the transmit frame time-stamp (TTS). The time master obtains synchronization information (SYNC message) via TTS. When transmit frame time-stamp generation is enabled (TTSEN=1), TTS updates with each transmitted frame. Time-stamps can be 32-bit or 64-bit; unused bits are zeroed.

16.10.1.13 CAN status bits and configuration register CAN_CFG_STAT

Register	Read/Write	Description	Reset Value
CAN_CFG_STAT	R/W	CAN status bits and configuration register	0x0090_0080

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 Symbol	Description
31	SACK	Auto Acknowledge Enable Bit 0: Disable Auto Acknowledge 1: Auto Acknowledge required in loopback mode (LBME=1) This bit is writable only when RESET=0
30	ROM	Receive Buffer (R_slot) Overflow Mode Selection Bit When all eight receive R_slots are full and a new message arrives, two handling methods are available: 0: Discard the earliest message 1: Discard the newly received message
29	ROV	Receive Buffer (R_slot) Overflow Flag (read-only) 0: No overflow 1: R_slot overflow occurred This bit clears to 0 after writing 1 to RREL
28	RREL	Receive Buffer (R_slot) Release Control Bit After reading RBUF, write 1 to this bit to release the current R_slot and point to the next R_slot. RSTAT (1:0) will update accordingly. 0: Current R_slot space not yet released 1: Current R_slot content read; RBUF space released
27	RBALL	Receive buffer (R_slot) storage mode configuration bit 0: Normal mode: Data frame stored in R_slot only after verification 1: Data frame stored in R_slot regardless of validity This bit is writable only when RESET=0
25 ~ 24	RSTAT[1:0]	Receive buffer status 00: R_slot empty 01: R_slot not empty, not full, no overflow; stored frames below AFWL programming threshold 10: R_slot not full, no overflow; stored frames at or above AFWL programming threshold 11: R_slot full; ROV=1 indicates RBUF overflow
23	FD_ISO	CAN FD ISO mode 0: Bosch CAN FD (non-ISO) mode 1: ISO CAN FD mode (ISO 11898-1:2015)
22	TSNEXT	Secondary Transmit Buffer (STB slot) switch control bit After each TBUF write completes, set this bit to 1 to mark the current STB as full and advance TBUF to the next STB. TSSTAT (1:0) updates accordingly. This bit is cleared automatically after CAN module reset. 0: No Operation 1: Current STB slot full; next slot selected
21	TSMODE	Secondary transmit buffer send mode select bit 0: FIFO mode 1: Arbitration mode
17 ~ 16	TSSTAT[1:0]	Secondary transmit status bit Indicates the number of messages stored in STB: 00: STB empty 01: STB stored messages ≤ 4 10: STB stored messages ≥ 5 11: STB stored messages = 8 TSSTAT fixed at 00 when STB is disabled

Bit Number	Bit Symbol	Description
15	TBSEL	Transmit buffer selection Select the transmit buffer for loading the message. Access via TBUF register. 0: Message write area is PTB (high-priority buffer) 1: Message write area is STB
14	LOM	Listen-only mode enable 0: Disabled 1: Enabled This bit is writable only when RESET=0
13	STBY	Transceiver standby mode 0: Disabled 1: Enabled This bit is writable only when RESET=0
12	TPE	PTB (high-priority) transmit mode enable 0: PTB disabled 1: Transmit PTB message
11	TPA	Abort PTB transmission 0: No abort 1: Abort PTB transfers requested by TPE=1 but not yet started (data remains in PTB). Enabling this bit clears TPE.
10	TSONE	STB single frame transmission enable bit 0: STB transmission disabled 1: Enable transmission of a single frame in STB
9	TSALL	STB (sub-priority) all frames transmission enable bit 0: STB transmission disabled 1: Enable transmission of all frames in STB
8	TSA	STB transfer abort 0: No abort 1: Abort STB transfers requested but not yet started
7	RESET	Reset request bit 0: No reset 1: Reset Some control bits are modifiable only when RESET=1, while others only when RESET=0 (e.g., LBME and other mode-related bits).
6	LBME	Loopback Mode 0: Disabled 1: Enabled LBME must not be enabled while transmission is active. This bit is writable only when RESET = 0.
5	LBMI	Internal Loopback Mode 0: Disabled 1: Enabled LBMI must not be enabled while transmission is active. This bit is writable only when RESET = 0.
4	TPSS	PTB automatic retransmission mask bit 0: PTB operates in automatic retransmission mode. Retransmission is triggered automatically on errors or arbitration loss. 1: PTB automatic retransmission is disabled. Retransmission is not performed on errors or arbitration loss.
3	TSSS	STB automatic retransmission mask bit 0: STB operates in automatic retransmission mode. Retransmission is triggered automatically on errors or arbitration loss. 1: STB automatic retransmission is disabled. Retransmission is not performed on errors or arbitration loss.
2	RACTIVE	Receive Status Bit 0: No receive activity 1: Receiving frame
1	TACTIVE	Transmit Status Bit 0: No transmit activity

Bit Number	Bit Symbol	Description
		1: Transmitting frame
0	BUSOFF	CAN Module Bus-Off Control 0: Bus on 1: Bus off Writing 1 to this bit resets the receive and transmit error counters RECNT and TECNT. It is recommended to modify this bit only during debugging.
26 20 ~ 18	-	Reserved

16.10.1.14 CAN interrupt control and interrupt flag register CAN_RTIE

Register	Read/Write	Description	Reset Value
CAN_RTIE	R/W	CAN Interrupt Control	0x1B00_00FE

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 Symbol	Description
31 ~ 28	AFWL[3:0]	Receive Buffer Threshold When the number of filled RB slots reaches the value set by AFWL (3:0), the RB slots threshold warning flag RAFIF is set. If RAFIE=1 at this moment, an interrupt is triggered.
27 ~ 24	EWL[3:0]	Error count threshold = (EWL[3:0] + 1) × 8, configurable range 8 to 128 RECNT and TECNT count errors during transmission and reception. When either counter reaches the error count threshold, EWARN and EIF are set. If EIE=1, an interrupt is triggered.
23	EWARN	Error count threshold warning flag 0: Both RECNT and TECNT counters are below the threshold (EWL[3:0]+1)*8 1: Either RECNT or TECNT counter has reached the threshold (EWL[3:0]+1)*8
22	EPASS	Error passive mode start bit 0: Not started (node is error active) 1: Started (node is error passive)
21	EPIE	Error passive interrupt enable bit 0: Disabled 1: Enabled
20	EPIF	Error passive interrupt flagIf the error state changes between error active and error passive, and this interrupt is enabled, EPIF is set.
19	ALIE	Arbitration Lost Interrupt Enable 0: Disabled 1: Enabled
18	ALIF	Arbitration Lost Interrupt Flag
17	BEIE	Bus Error Interrupt Enable 0: Disabled 1: Enabled
16	BEIF	Bus Error Interrupt Flag
15	RIF	Receive Interrupt Flag

Bit Number	Bit Symbol	Description
		0: No Frame Received 1: Data or Remote Frame Fully Received and Readable from Receive Buffer
14	ROIF	RB Overflow Flag 0: RB Slots Not Overflowed 1: RB Slots Full and At Least One Received Message Overwritten When Receive Overflow Occurs, ROIF and RFIF are Set. If ROIE=1, an Interrupt is Triggered
13	RFIF	RB Full Flag 0: RB Slots Not Full 1: RB Slots Full Clear RB Slots Before Next Message Arrival; Otherwise, the Oldest Message Will Be Discarded
12	RAFIF	RB slots threshold warning flag 0: Filled RB slots < AFWL 1: Filled RB slots ≥ AFWL
11	TPIF	PTB transfer complete flag 0: PTB transfer not completed 1: PTB transfer request acknowledged, transfer completed
10	TSIF	STB transfer complete flag 0: STB transfer not completed 1: STB transfer request acknowledged, transfer completed
9	EIF	Error interrupt flag This bit is set when any of the following conditions are met: a) a) RECNT error count reaches error count threshold ($EWL[3:0] + 1) \times 8$ b) TECNT error count reaches error count threshold ($EWL[3:0] + 1) \times 8$ c) BUSOFF bit is overwritten
8	AIF	Transmission abort flag 0: Abort operation not executed 1: This bit is set after TPA or TSA writes 1, aborting the related transmit request.
7	RIE	Receive interrupt enable 0: Disabled 1: Enabled
6	ROIE	RB overflow interrupt enable 0: Disabled 1: Enabled
5	RFIE	RB full interrupt enable 0: Disabled 1: Enabled
4	RAFIE	RB slots threshold warning interrupt enable 0: Disabled 1: Enabled
3	TPIE	PTB transmission complete interrupt enable 0: Disabled 1: Enabled
2	TSIE	STB transmission complete interrupt enable 0: Disabled 1: Enabled
1	EIE	Error interrupt enable 0: Disabled 1: Enabled
0	TSFF	Transmit area full flag Secondary transmit buffer full flag 0: STB has not reached the maximum number of messages 1: STB has reached the maximum number of messages If STB_DISABLE disables the STB, TSFF is 0.

16.10.1.15 Low-speed clock configuration register CAN_S_SEG

Register	Read/Write	Description	Reset Value
CAN_S_SEG	R/W	Low-speed clock configuration register	0x0102_0203

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 Symbol	Description
31 ~ 24	S_PRESC[7:0]	Low-speed prescaler The prescaler divides the system clock to generate the time quantum clock (tq_clk). Valid range PRESC = [0x00, 0xFF], prescaler value from 1 to 256 Prescaler value = S_PRESC + 1
22 ~ 16	S_SJW[6:0]	Low-speed synchronization jump width $t_{SJW} = (SJW + 1) \cdot TQ$ Synchronization jump width defines the maximum time by which the resynchronization bit time can be shortened or extended, where TQ represents the time quantum.
14 ~ 8	S_Seg_2[6:0]	Low-speed phase segment 2 Phase segment 2 configures the time after the sampling point; the sampling point is set to $t_{Seg_2} = (Seg_2 + 1) \cdot TQ$
7 ~ 0	S_Seg_1[7:0]	Low-speed phase segment 1 Phase segment 1 configures the time before the sampling point minus the synchronization segment time; the sampling point is set to $t_{Seg_1} = (Seg_1 + 2) \cdot TQ$
23 15	-	Reserved

16.10.1.16 High-speed clock configuration register CAN_F_SEG

Register	Read/Write	Description	Reset Value
CAN_F_SEG	R/W	High-speed clock configuration register	0x0102_0203

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 Symbol	Description
31 ~ 24	F_PRESC[7:0]	High-speed prescaler The prescaler divides the system clock to generate the time quantum clock (tq_clk). Valid range PRESC = [0x00, 0xFF], prescaler value from 1 to 256 Division value = F_PRESC + 1

Bit Number	Bit Symbol	Description
19 ~ 16	F_SJW[3:0]	High-speed synchronous jump width $t_{SJW} = (SJW + 1) \cdot TQ$ Synchronous jump width is the maximum time to shorten or extend the resynchronization bit time, where TQ is the time quantum
11 ~ 8	F_Seg_2[3:0]	High-speed phase segment 2 Phase segment 2 configures the time after the sampling point; the sampling point is set to $t_{Seg_2} = (Seg_2 + 1) \cdot TQ$
4 ~ 0	F_Seg_1[4:0]	High-speed phase segment 1 Phase segment 1 configures the time before the sampling point minus the synchronization segment time; the sampling point is set to $t_{Seg_1} = (Seg_1 + 2) \cdot TQ$
23 ~ 20 15 ~ 12 7 ~ 5	-	Reserved

16.10.1.17 Delay compensation and transmit/receive error register CAN_EALCAP

Register	Read/Write	Description	Reset Value
CAN_EALCAP	R/W	Delay compensation and transmit/receive error register	0x0000_0000

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 Symbol	Description
31 ~ 24	TECNT[7:0]	Transmit error counter, representing the number of transmission errors
23 ~ 16	RECNT[7:0]	Receive error counter, representing the number of reception errors
15	TDCEN	Transmit delay compensation enable bit 0: Disable transmitter delay compensation 1: Enable transmitter delay compensation
14 ~ 8	SSPOFF	Second sampling point offset
7 ~ 5	KOER[2:0]	Error type Error type (error code) 000: No Error 001: Bit Error 010: Format error 011: Bit Stuffing Error 100: Acknowledge Error 101: CRC Error 110: Other errors (dominant bits after the own error flag, active error flag duration excessive, dominant bits during passive error flag after an ACK error) 111: KOER does not update with each new error. It remains unchanged during successful frame transmit or receive.
4 ~ 0	ALC[4:0]	Arbitration lost capture (bit position of arbitration lost within the frame) The ALC values are defined as follows: The frame starts with the SOF bit, followed by the first bit of the ID. The ALC value for the first ID bit is 0, for the second ID bit is 1, and

Bit Number	Bit Symbol	Description
		so forth.

16.10.1.18 Filter control register CAN_ACFCTRL

Register	Read/Write	Description	Reset Value
CAN_ACFCTRL	R/W	Filter control register	0x00010200

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 Symbol	Description
23 ~ 16	AE_X X=(0 to 7)	Receive filter X enable 0: Disable receive filter 1: Enable receive filter
9	TIMEPOS	Timestamp position 0:SOF 1:EOF TIMEPOS can only be modified when TIMEEN=0; however, TIMEPOS can be updated via write access when TIMEEN=1.
8	TIMEEN	Time-Stamping Enable 0: Disabled 1: Enabled
5	SELMASK	Select Receive MASK 0: Direct ACF_x to receive code 1: Direct ACF_x to receive mask
3 ~ 0	ACFADR[3:0]	Receive Filter Address ACFADR points to a specific receive filter, accessible via the ACF_x register. SELMASK bit selects between receive code and receive mask for the filter.
31 ~ 24 15 ~ 10 7 ~ 6 4	-	Reserved

16.10.1.19 Filter data register CAN_ACF

Register	Read/Write	Description	Reset Value
CAN_ACF	R/W	Filter Data Register	0x0000_0000

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 Symbol	Description
30	AIDEE	Receive Mask IDE Bit Check Enable 0: Receive filter ignores AIDE restriction; both typical and extended frames are accepted 1: Receive filter applies AIDE to accept only typical or only extended frames
29	AIDE	Receive mask IDE bit; valid only when AIDEE = 1 0: Receive filter accepts standard frames only 1: Receive filter accepts extended frames only
28 ~ 0	AMASK /ACODE[28:0]	Receive MASK 0: Enable reception check on these bits of the receive identifier 1: Disable reception check on these bits of the receive identifier Receive CODE 0: ACC bit compared with receive message ID bit 1: ACC bit compared with receive message ID bit
31	-	Reserved

16.10.1.20 CAN interrupt and control enable register (CAN_IDE)

Register	Read/Write	Description	Reset Value
CAN_IDE	R/W	CAN Interrupt and Control Enable Register	0x8000_0000

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 Symbol	Description
31	FDEN	FD frame format enable bit 0: Disabled 1: Enabled
15	TXEN	CAN transmit enable bit 0: Disabled 1: Enabled
14	RXEN	CAN receive enable bit 0: Disabled 1: Enabled
11	TIM_EN	Time-Stamping Counter Enable Bit 0: Disabled 1: Enabled
0	INTEN	Interrupt Request CPU Mask Bit 0: Interrupt Request Masked 1: Interrupt Request Enabled
30 ~ 16 13 ~ 12 10 ~ 1	-	Reserved

16.10.1.21 CAN time-stamping timer low register CAN_TIML

Register	Read/Write	Description	Reset Value
CAN_TIML	R/W	CAN Time-Stamping Timer Low R/W Register	0x0000_0000

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 Symbol	Description
31 ~ 0	CAN_TIML[31:0]	64-bit Time-Stamping Timer Lower 32 bits Note: Data can only be written when TIM_EN is 0.

16.10.1.22 CAN time-stamping timer high register CAN_TIMH

Register	Read/Write	Description	Reset Value
CAN_TIMH	R/W	CAN Time-Stamping Timer High R/W Register	0x0000_0000

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 Symbol	Description
31 ~ 0	CAN_TIMH[31:0]	64-bit Time-Stamping Timer Upper 32 bits Note: Data can only be written when TIM_EN is 0.

16.10.1.23 CAN address configuration register CAN_SADDR

Register	Read/Write	Description	Reset Value
CAN_SADDR	R/W	CAN Address Configuration Register	0x0000_0000

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

Bit Number	Bit Symbol	Description
31 ~ 0	SADDR[31:0]	CAN Message RAM Address Configuration Register Once the CAN module is enabled, its Message RAM occupies a contiguous RAM area dedicated to CAN message storage.

Bit Number	Bit Symbol	Description
		This region uses the value in the SADDR[31:0] register as the base address. The address range is from base address to base address + 0x597, totaling 1432 bytes. Specific Address distribution visible at 16.5 message RAM

16.10.2 CAN register mapping

Register	Offset Address	Read/Write	Description	Reset Value
CAN Base Address: 0x40010C00				
CAN_RX_ID	0x00	R	Receive Frame ID Register	0xnnnn_nnnn
CAN_RX_CTRL	0x04	R	Receive Frame Control and Status Register	0xnnnn_nnnn
CAN_RBUF0	0x08	R	Receive Frame Data Register 0	0xnnnn_nnnn
CAN_RBUF1	0x0C	R	Receive Frame Data Register 1	0xnnnn_nnnn
CAN_RBUF2	0x10	R	Receive Frame Data Register 2	0xnnnn_nnnn
CAN_RBUF3	0x14	R	Receive Frame Data Register 3	0xnnnn_nnnn
CAN_RBUF4	0x18	R	Receive Frame Data Register 4	0xnnnn_nnnn
CAN_RBUF5	0x1C	R	Receive Frame Data Register 5	0xnnnn_nnnn
CAN_RBUF6	0x20	R	Receive Frame Data Register 6	0xnnnn_nnnn
CAN_RBUF7	0x24	R	Receive Frame Data Register 7	0xnnnn_nnnn
CAN_RBUF8	0x28	R	Receive Frame Data Register 8	0xnnnn_nnnn
CAN_RBUF9	0x2C	R	Receive Frame Data Register 9	0xnnnn_nnnn
CAN_RBUF10	0x30	R	Receive Frame Data Register 10	0xnnnn_nnnn
CAN_RBUF11	0x34	R	Receive Frame Data Register 11	0xnnnn_nnnn
CAN_RBUF12	0x38	R	Receive Frame Data Register 12	0xnnnn_nnnn
CAN_RBUF13	0x3C	R	Receive Frame Data Register 13	0xnnnn_nnnn
CAN_RBUF14	0x40	R	Receive Frame Data Register 14	0xnnnn_nnnn
CAN_RBUF15	0x44	R	Receive Frame Data Register 15	0xnnnn_nnnn
CAN_RTSL	0x48	R	Receive Frame Time-Stamp Storage Register Low	0xnnnn_nnnn
CAN_RTSH	0x4C	R	High bits of the receive frame time-stamping storage register	0xnnnn_nnnn
CAN_TX_ID	0x50	R/W	Transmit frame ID register	0xnnnn_nnnn

Register	Offset Address	Read/Write	Description	Reset Value
CAN_TX_CTRL	0x54	R/W	Transmit frame control and status register	0xnnnn_nnnn
CAN_TBUF0	0x58	R/W	Transmit Frame Data Register 0	0xnnnn_nnnn
CAN_TBUF1	0x5C	R/W	Transmit Frame Data Register 1	0xnnnn_nnnn
CAN_TBUF2	0x60	R/W	Transmit Frame Data Register 2	0xnnnn_nnnn
CAN_TBUF3	0x64	R/W	Transmit Frame Data Register 3	0xnnnn_nnnn
CAN_TBUF4	0x68	R/W	Transmit Frame Data Register 4	0xnnnn_nnnn
CAN_TBUF5	0x6C	R/W	Transmit Frame Data Register 5	0xnnnn_nnnn
CAN_TBUF6	0x70	R/W	Transmit Frame Data Register 6	0xnnnn_nnnn
CAN_TBUF7	0x74	R/W	Transmit Frame Data Register 7	0xnnnn_nnnn
CAN_TBUF8	0x78	R/W	Transmit Frame Data Register 8	0xnnnn_nnnn
CAN_TBUF9	0x7C	R/W	Transmit Frame Data Register 9	0xnnnn_nnnn
CAN_TBUF10	0x80	R/W	Transmit Frame Data Register 10	0xnnnn_nnnn
CAN_TBUF11	0x84	R/W	Transmit Frame Data Register 11	0xnnnn_nnnn
CAN_TBUF12	0x88	R/W	Transmit Frame Data Register 12	0xnnnn_nnnn
CAN_TBUF13	0x8C	R/W	Transmit Frame Data Register 13	0xnnnn_nnnn
CAN_TBUF14	0x90	R/W	Transmit Frame Data Register 14	0xnnnn_nnnn
CAN_TBUF15	0x94	R/W	Transmit Frame Data Register 15	0xnnnn_nnnn
CAN_TTSL	0x98	R	Transmit frame time-stamp storage register low	0xnnnn_nnnn
CAN_TTSH	0x9C	R	Transmit frame time-stamp storage register high 32 bits	0xnnnn_nnnn
CAN_CFG_STAT	0xA0	R/W	CAN status bits and configuration register	0x0090_0080
CAN_RTIE	0xA4	R/W	CAN Interrupt Control and Interrupt Flag Register	0x1B00_00FE
CAN_S_SEG	0xA8	R/W	Low-speed clock configuration register	0x0102_0203
CAN_F_SEG	0xAC	R/W	High-speed clock configuration register	0x0102_0203
CAN_EALCAP	0xB0	R/W	Delay compensation and transmit/receive error register	0x0000_0000
CAN_ACFCTRL	0xB4	R/W	Filter control register	0x0001_0200
CAN_ACF	0xB8	R/W	Filter Data Register	0x0000_0000

Register	Offset Address	Read/Write	Description	Reset Value
CAN_IDE	0xCC	R/W	CAN Interrupt and Control Enable Register	0x8000_0000
CAN_TIML	0xD0	R/W	CAN Time-Stamping Timer Low Register	0x0000_0000
CAN_TIMH	0xD4	R/W	CAN Time-Stamping Timer High Register	0x0000_0000
CAN_SADDR	0xD8	R/W	CAN Address Configuration Register	0x0000_0000

17 Hardware watchdog WDT

The SC32F11T/11G Series includes an independent hardware Watchdog Timer (WDT) with an internal 37 kHz oscillator LIRC as its clock source. Users can enable or disable the Watchdog Reset function via the ENWDT control bit in the Customer Option of the programmer.

The hardware Watchdog WDT offers high security, precise timing, and flexible usage. This Watchdog peripheral can detect faults caused by software errors and triggers a System Reset when the counter reaches the set overflow time.

The WDT is driven by its internal low-frequency oscillator, ensuring it remains operational even if the main clock fails.

17.1 Clock source

The clock source of the WDT in the SC32F11T/11G Series is fixed to LIRC. Once the WDT is enabled, LIRC automatically starts and remains oscillating throughout the WDT operation; users cannot disable it.

17.2 WDT registers

17.2.1 WDT related register table

17.2.1.1 WDT control register WDTCON

Register	Read/Write	Description	Reset Value
WDTCON	R/W	WDT 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
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	CLRWDT

Bit Number	Bit Symbol	Description
0	CLRWDT	WDT Counter Clear Bit This bit is set to 1 by software and cleared to 0 automatically by hardware. 0: No effect 1: WDT counter starts counting from 0
31 ~ 1	-	Reserved

17.2.1.2 WDT configuration register WDTCFG

Register	Read/Write	Description	Reset Value
WDTCFG	R/W	WDT 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
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	-	WDTCKS[2:0]		

Bit Number	Bit Symbol	Description
2 ~ 0	WDTCKS[2:0]	Watchdog Clock Selection:
		WDTCKS[2:0]
		WDT Timeout Period
		000 500ms
		001 250ms
		010 125ms
		011 62.5ms
		100 31.5ms
		101 15.75ms
		110 7.88ms
		111 3.94ms
31 ~ 3	-	Reserved

17.2.2 WDT register mapping

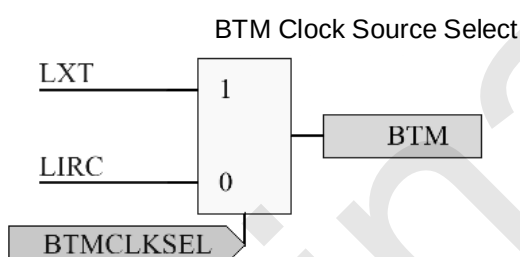
Register	Offset Address	Read/Write	Description	Reset Value	Access Restriction
WDT Base Address: 0x4000_0330					
WDTCON	0x0C	R/W	WDT Control Register	0x0000_0000	Byte/Half-word Access Not Supported
WDTCFG	0x10	R/W	WDT Configuration Register	0x0000_0000	Byte/Half-word Access Not Supported

18 Base timer (BTM)

The SC32F11T/11G Series has a built-in Base Timer (BTM) that can generate Interrupts at intervals ranging from 15.625 ms to 32 s. Both the 37 kHz LIRC and the external 32.768 kHz Crystal Oscillator LXT can be used as the Clock Source for the BTM. Interrupts generated by the BTM can wake the CPU from STOP mode.

18.1 Clock source

- The BTM clock source options for the SC32F11T/11G Series include LXT and LIRC.



18.2 Feature

- Interrupt frequency interval selectable between 15.625 ms and 32 s
- Wakeup STOP Mode

18.3 BTM interrupt

In the SC32F11T/11G Series, when the BTM counter meets the BTMFS conditions, BTMIF is set. If BTM_CON.INTEN = 1, an interrupt is generated.

Interrupt Event	Event Flag	Interrupt Enable Control Bit
BTM Interrupt Request	BTMIF	BTM_CON->INTEN

18.4 BTM register

18.4.1 BTM-related register table

18.4.1.1 Low-frequency timer control register (BTM_CON)

Register	Read/Write	Description	Reset Value
BTM_CON	R/W	Low-Frequency Timer 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
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
ENBTM	INTEN	-	-	BTMFS[3:0]			

Bit Number	Bit Symbol	Description
7	ENBTM	Base Timer Start Control Bit 0: Base Timer stopped 1: Base Timer started
6	INTEN	CPU Interrupt Request Enable Control Bit 0: Disable interrupt request 1: Enable interrupt request
3 ~ 0	BTMFS[3:0]	Low-Frequency Clock Interrupt Frequency Selection 0000: Generates an Interrupt every 15.625 ms 0001: Generates an Interrupt every 31.25 ms 0010: Generates an Interrupt every 62.5 ms 0011: Generates an Interrupt every 125 ms 0100: Generates an Interrupt every 0.25 s 0101: Generates an Interrupt every 0.5 s 0110: Generates an Interrupt every 1.0 s 0111: Generates an Interrupt every 2.0 s 1000: Generates an Interrupt every 4.0 s 1001: Generates an Interrupt every 8.0 s 1010: Generates an Interrupt every 16.0 s 1011: Generates an Interrupt every 32.0 s 1100 to 1111: Reserved
31 ~ 8 5 ~ 4	-	Reserved

18.4.1.2 BTM flag register BTM_STS

Register	Read/Write	Description	Reset Value
BTM_STS	R/W	BTM Flag 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
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	BTMIF

Bit Number	Bit Symbol	Description
0	BTMIF	Base Timer Interrupt Flag This bit is set by hardware and cleared by writing 1 in software. This bit is set automatically when the BTM count meets the BTMFS configuration conditions.
31 ~ 1	-	Reserved

18.4.2 BTM register mapping

Register	Offset Address	Read/Write	Description	Reset Value
BTM Base Address: 0x4002_2100				
BTM_CON	0x00	R/W	Low-Frequency Timer Control Register	0x0000_0000
BTM_STS	0x04	R/W	BTM Flag Register	0x0000_0000

19 Built-in CRC verification module

The SC32F11T/11G Series features a built-in CRC Verification Module that uses a polynomial generator to produce CRC codes from 8-bit, 16-bit, or 32-bit data Words.

19.1 Clock source

- The CRC clock source of the SC32F11T/11G Series is derived from HCLK

19.2 Feature

- Built-in hardware CRC module.
- Initial value configurable, default is 0xFFFF_FFFF.
- Supports 8-bit, 16-bit, and 32-bit data units.
- Polynomial is programmable, default is 0x04C1_1DB7.
- Only supports software mode for CRC calculation.
- Supports DMA: CRC_DR can be used as a DMA target address and accessed directly via Register.
- CRC calculation for a single byte requires 1 system clock.

CRC algorithm name	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
Initial value	0xFFFF_FFFF
Result XOR value	0x0000_0000
Input value inversion	false
Output Inversion	false
LSB/MSB	MSB

Note: The data written to and read from CRCDR are not the same.

19.3 CRC register

19.3.1 CRC related register table

19.3.1.1 CRC data register (CRC_DR)

Register	Read/Write	Description	Reset Value
CRC_DR	R/W	CRC Data Register (Input Data / Result)	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 Symbol	Description
31 ~ 0	CRCDR[31:0]	CRC Data Register Bits This register is used to write new data to the CRC calculator. Reading this register returns the previous CRC calculation result. For data sizes less than 32 bits, the least significant bits are used for correct write/read operations. The following conditions apply when accessing this register: 1. First, set CRC_CON.CRCRST to 1 to reset the CRCDR. 2. Each write to the CRC data register triggers automatic CRC calculation; the result is stored in CRCDR. CRC calculation results can be read at any time.

19.3.1.2 CRC control register (CRC_CON)

Register	Read/Write	Description	Reset Value
CRC_CON	R/W	CRC 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
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
POLYSIZE[1:0]		-	-	-	-	-	CRCRST

Bit Number	Bit Symbol	Description
7 ~ 6	POLYSIZE[1:0]	CRC Polynomial Size Setting Bits 00: 32-bit Polynomial 01: 16-bit Polynomial 10: 8-bit Polynomial 11: 7-bit Polynomial
0	CRCRST	CRCDR Register Reset (Q31 to Q0) This bit is set to 1 by software and cleared to 0 automatically by hardware. 0: No effect 1: Reset CRCDR; the reset value is the value written by the user to the CRC_INIT register.
31 ~ 8 5 ~ 1	-	Reserved

19.3.1.3 CRC initial value register CRC_INIT

Register	Read/Write	Description	Reset Value
CRC_INIT	R/W	CRC Initial Value Register	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 Symbol	Description
31 ~ 0	CRC_INIT[31:0]	Programmable CRC initial value; reset value: 0xFFFFFFFF This register is used to write the CRC initial value.

19.3.1.4 CRC polynomial setting register CRC_POL

Register	Read/Write	Description	Reset Value
CRC_POL	R/W	CRC Polynomial Setting Register	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 Symbol	Description
31 ~ 0	POL[31:0]	Programmable polynomial; reset value: 0x04C11DB7 This register is used to write the polynomial coefficients for CRC calculation. If the polynomial value is less than 32 bits, the correct value must be written starting from the least significant bit.

19.3.2 CRC register mapping

Register	Offset Address	Read/Write	Description	Reset Value	Access Restriction
CRC Base Address: 0x4000_2000					
CRC_DR	0x00	R/W	CRC Data Register	0x0000_0000	-
CRC_CON	0x04	R/W	CRC Control Register	0x0000_0000	-

Register	Offset Address	Read/Write	Description	Reset Value	Access Restriction
CRC_INT	0x08	R/W	CRC Initial Value Register	0x0000_0000	Byte/Half-word Access Not Supported
CRC_POL	0x0C	R/W	CRC Polynomial Setting Register	0x0000_0000	Byte/Half-word Access Not Supported

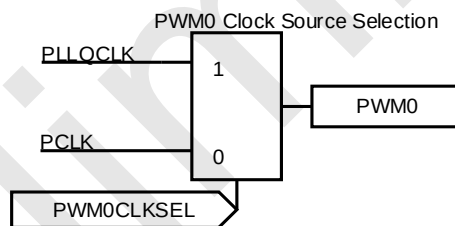
20 PWM0: 8-channel 16-bit multifunction PWM

20.1 Overview

PWM0 of the SC32F11T/11G Series is an 8-channel 16-bit common-period multifunction PWM. PWM0 offers comprehensive functions: enabling adjustment of period and duty cycle; selection of output waveform types including center-aligned or edge-aligned modes; output modes configurable as independent or complementary; support for dead time; and a Fault Detection mechanism. Registers PWM0_CON and PWM0_STS control the PWM status and period. Each PWM channel's activation, output waveform, waveform inversion, and duty cycle can be configured individually.

20.2 Clock source

- The clock source of the SC32F11T/11G Series PWM0 can be selected between PLLQCLK and PCLK
- PWM0 output frequency can reach the frequency of the selected clock source
- PWM0 clock prescaler range: /1 to /128



20.3 Feature

- Supports 8-stage PWM0 clock prescaling.
- 8-channel 16-bit common period multifunction PWM.
- Output waveform can be inverted.
- Waveform type: configurable as center-aligned or edge-aligned.
- PWM Mode: can be set to independent mode or complementary mode
 - In independent mode, all eight PWM channels have the same period, but each channel's duty cycle can be adjusted individually.
 - In complementary Mode, four groups of complementary PWM waveforms with Dead Time can be output simultaneously.
- Provides one PWM overflow Interrupt
- Supports Fault Detection mechanism
- Has independent Interrupt request Flags

20.5.2 Alignment type

20.5.2.1 Edge-aligned

The PWM counter counts up from 0. When the counter value matches the duty cycle register PDT0x[15:0], the PWM output switches between high and low. The counter continues counting until it matches PWMPD[15:0] + 1 (end of the PWM period), then resets to 0. If PWM interrupt is enabled, an interrupt is generated at this point. The PWM output waveform is left edge-aligned.

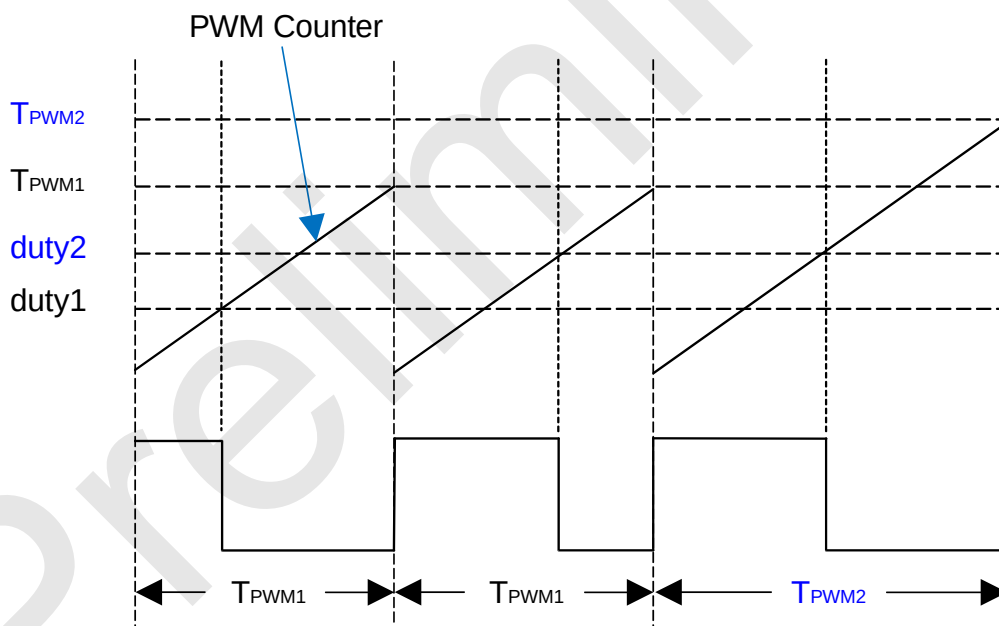
Edge-aligned PWM period (TPWM) calculation formula:

$$T_{pwm} = \frac{PWMPD[15:0] + 1}{PWM\text{Clock Frequency}}$$

Edge-aligned PWM duty cycle calculation formula:

$$\text{duty} = \frac{PDT0x[15:0]}{PWMPD[15:0] + 1}$$

The edge-aligned waveform diagram is shown below:



Edge-aligned PWM

20.5.2.2 Center-aligned

The PWM counter counts up from 0. When the count matches the duty cycle setting PDT0x[15:0], the PWM output toggles its level. The counter then continues counting up until it reaches PWMPD[15:0] + 1 (the PWM period midpoint), at which point it switches to counting down. When the count matches PDT0x[15:0] again during the down count, the PWM output toggles level once more. Counting continues down until an overflow occurs, marking the end of one PWM period. If PWM interrupts are enabled, an interrupt is generated at this stage.

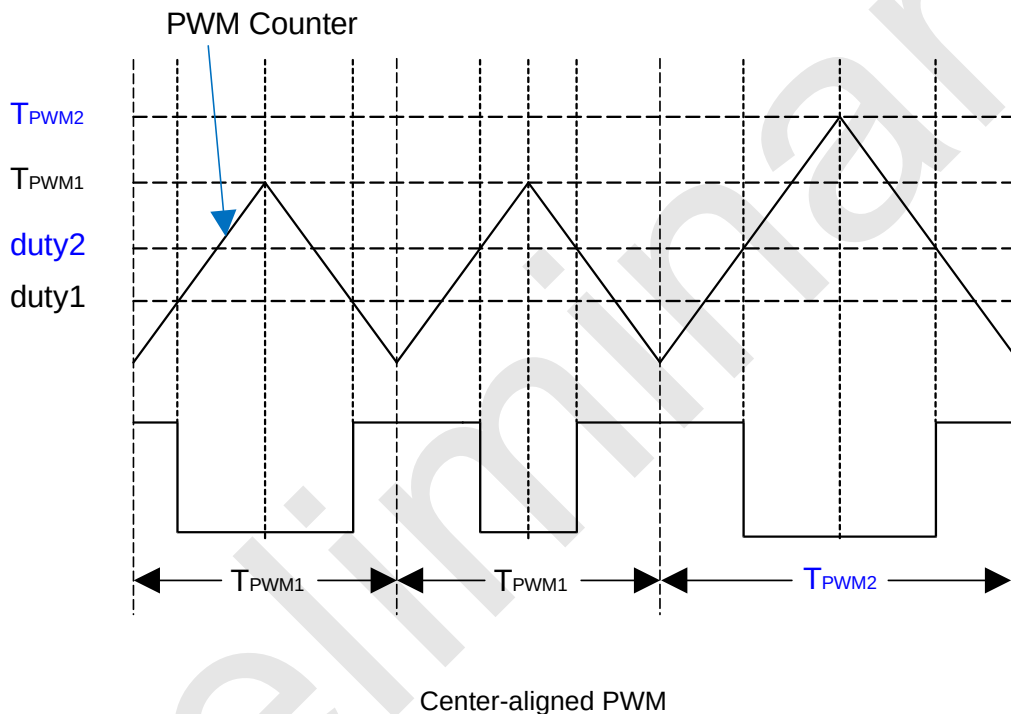
Center-aligned period T_{PWM} calculation formula:

$$T_{pwm} = 2 * \frac{PWMPD[15:0] + 1}{PWM \text{ Clock Frequency}}$$

Center-aligned duty cycle calculation formula:

$$duty = \frac{PDT0x[15:0]}{PWMPD[15:0] + 1}$$

Center-aligned waveform diagram:

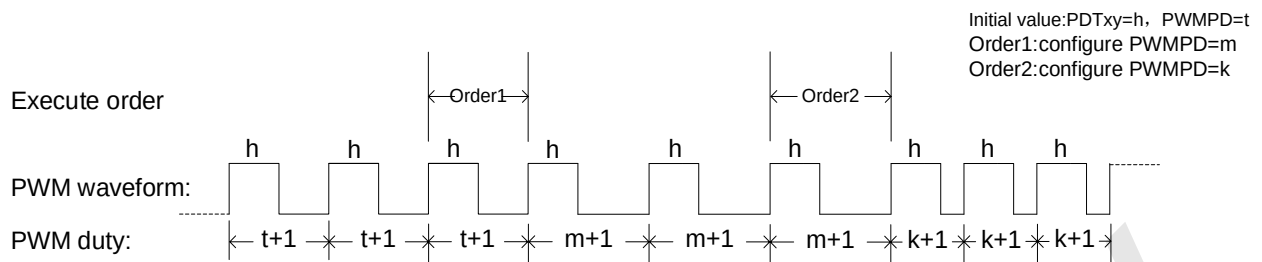


20.5.3 Duty cycle variation feature

When PWM0n outputs a waveform, the Duty Cycle can be changed by modifying the value of the high-level setting Register (PDT0x). Note that when PDT0x is changed, the duty cycle does not update immediately. It takes effect when the PWM counter either reaches 0 or counts up to PWMPD[15:0] + 1, which defines the PWM period.

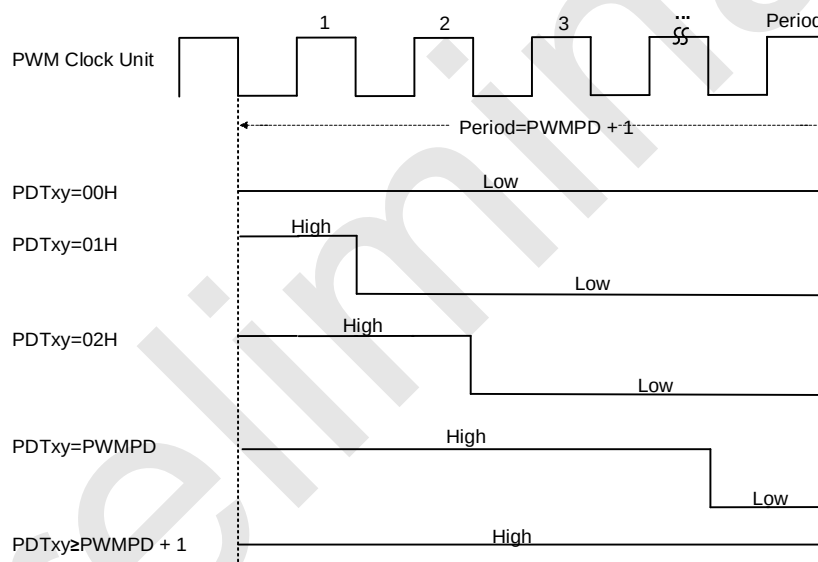
20.5.4 Period change feature

When the PWM output waveform is active, you can change the period by modifying the value of the period setting register PWMPD. Changing the PWMPD value does not immediately update the Period. The change takes effect when the PWM counter reaches zero or counts up to PWMPD[15:0] + 1, as illustrated below.



20.5.5 Relationship between period and duty cycle

The relationship between Period and Duty Cycle is shown in the following diagram. This behavior assumes PWM output inversion control (INVx, x=0 to 7) is initially cleared (0). To reverse the output polarity, set INVx to 1.



Period and Duty Cycle Relationship Diagram

20.6 PWM0 independent mode

In independent mode (PWM0_CON.5 = 0), the duty cycle for each of the 8 PWM channels can be set independently. After configuring the PWM output state and Period, set the duty cycle register for each PWM channel to output PWM waveforms with a fixed duty cycle.



20.7 PWM0 complementary mode

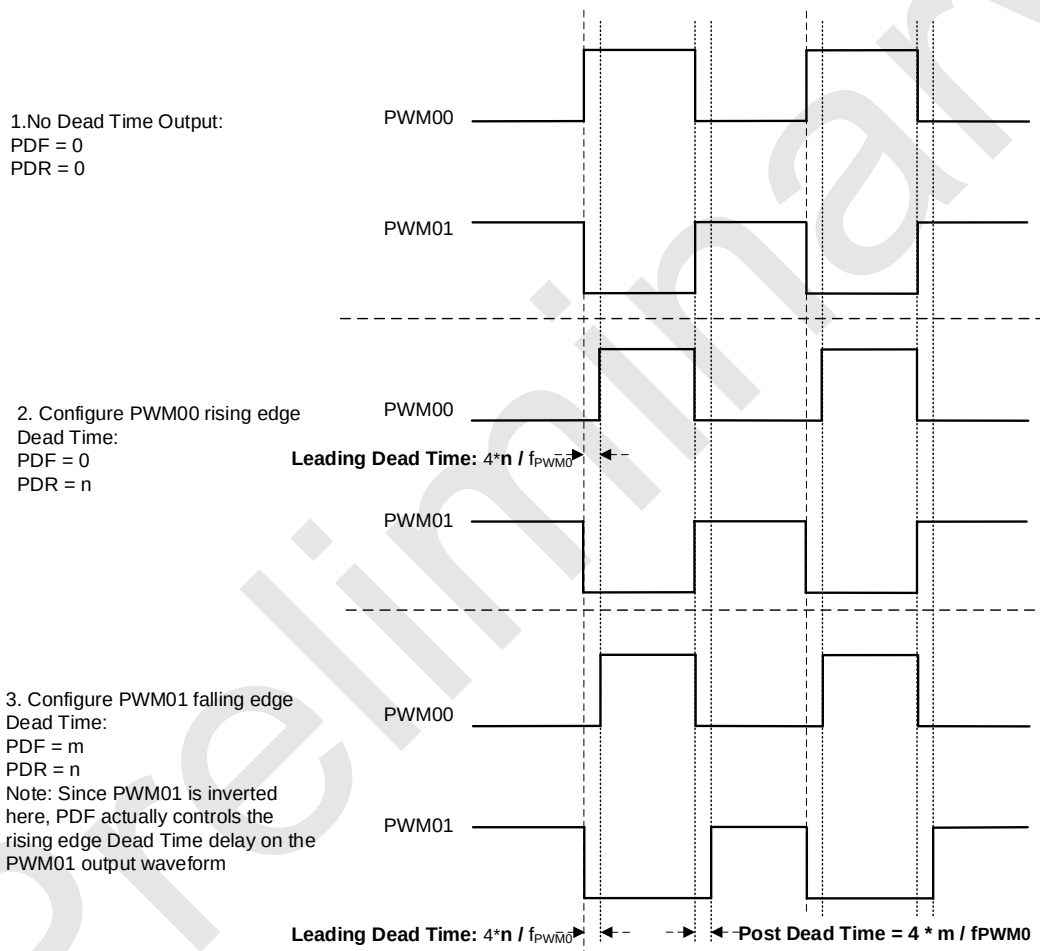


20.7.1 PWM0 complementary mode dead time configuration

When PWM0 on the SC32F11T/11G operates in complementary mode, the dead time control module prevents overlap during the active periods of the two complementary PWM signals. This ensures that the complementary power switches driven by these PWM signals never conduct simultaneously.

20.7.2 PWM0 dead time output waveform

The diagram below shows the dead time adjustment waveform for PWM0 and PWM1 in complementary mode. To distinguish them, PWM01 is inverted (INV1=1).



PWM0 Dead Time Output Waveform

20.8 PWM0 fault detection function configuration

Fault detection is commonly used for protection in motor control systems. When the fault detection function is enabled and FLTEN (PWM0_FLT.7) is set to 1, the fault detection signal input pin (FLT) is enabled. When the signal on the FLT pin meets fault conditions, the FLTSTA flag is set by hardware, PWM output stops, the PWM counter continues counting, and PWM interrupts remain unaffected. Fault detection modes include latch mode and immediate mode.

In immediate mode, when the fault signal on the FLT pin meets disable conditions, the FLTSTA flag is cleared

by hardware. PWM output resumes after the PWM counter reaches zero.

In latch mode, when the fault signal on the FLT pin meets the disable condition, the FLTSTA flag remains unchanged. Users can clear it via software. Once cleared, the PWM counter resumes counting, and PWM output resumes after the counter reaches zero.

20.9 PWM0 interrupt

In the SC32F11T/11G Series, after PWM completes one output period, PWMIF is set. If PWM0_CON.INTEN = 1, an interrupt is triggered.

Interrupt Event	Event Flag	Interrupt Enable Control Bit
PWM0 Interrupt Request	PWMIF	PWM0_CON->INTEN

20.10 PWM0 registers

20.10.1 PWM0 related register table

20.10.1.1 PWM0 control register PWM0_CON

Register	Read/Write	Description	Reset Value
PWM0_CON	R/W	PWM0 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
-	-	-	-	-	-	-	INTEN
7	6	5	4	3	2	1	0
ENPWM	PWMMD0	PWMMD1	-	-	PWMCK[2:0]		

Bit Number	Bit Symbol	Description
8	INTEN	CPU Interrupt Request Enable Control Bit 0: Disable interrupt request 1: Enable interrupt request
7	ENPWM	PWM Module Enable Control Bit 1: Enable clock to the PWM unit; PWM is active. The PWM output state is controlled by the ENPWMx register (x=0 to 7). 0: PWM unit stops working, PWM counter resets to 0, all PWM outputs are set to GPIO mode.
6	PWMMD0	PWM waveform alignment mode selection bit 0: Edge-aligned mode 1: Center-aligned mode
5	PWMMD1	PWM waveform complementary mode setting bit 0: Independent mode 1: Complementary mode

Bit Number	Bit Symbol	Description
2 ~ 0	PWMCK[2:0]	PWM clock frequency level control bit Used to set PWM clock frequency f_{PWM0} 000: $f_{source}/1$ 001: $f_{source}/2$ 010: $f_{source}/4$ 011: $f_{source}/8$ 100: $f_{source}/16$ 101: $f_{source}/32$ 110: $f_{source}/64$ 111: $f_{source}/128$ $f_{source} = f_{PCLK}$ or f_{PLLQ}
31 ~ 9 4 ~ 3	-	Reserved

20.10.1.2 PWM0 channel configuration register PWM0_CHN

Register	Read/Write	Description	Reset Value
PWM0_CHN	R/W	PWM0 channel 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
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
ENPWM7	ENPWM6	ENPWM5	ENPWM4	ENPWM3	ENPWM2	ENPWM1	ENPWM0

Bit Number	Bit Symbol	Description
7 ~ 0	ENPWMx (x=0 to 7)	PWM0x waveform output selection 0: PWM0x output off and set as GPIO 1: When ENPWM = 1, the PWM0x pin outputs the waveform Description: If ENPWM is set to 1, the PWM module is enabled. However, if ENPWMx = 0, the PWM output is disabled and the pin functions as a GPIO. In this state, the PWM module can be used as a 16-bit Timer. If PWM0_CON.INTEN = 1, interrupts will still be generated.
31 ~ 8	-	Reserved

20.10.1.3 PWM0 status flag register (PWM0_STS)

Register	Read/Write	Description	Reset Value
PWM0_STS	R/W	PWM0 Status Flag 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
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	-	-	FLTSTA	PWMIF

Bit Number	Bit Symbol	Description
1	FLTSTA	PWM Fault Detection Flag 0: PWM operates normally 1: Fault detection is active, PWM output is in high-impedance state. In latch mode, this bit can be cleared by writing 1 in software.
0	PWMIF	PWM Interrupt Request Flag This bit is set by hardware and cleared by writing 1 in software. This bit is set by hardware when the PWM counter overflows (count exceeds PWMPD). If PWM0_CON.INTEN = 1, a PWM0 interrupt is triggered.
31 ~ 2	-	Reserved

20.10.1.4 PWM0 waveform output reverse control register PWM0_INV

Register	Read/Write	Description	Reset Value
PWM0_INV	R/W	PWM0 Waveform Output Reverse 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
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
INV7	INV6	INV5	INV4	INV3	INV2	INV1	INV0

Bit Number	Bit Symbol	Description
7 ~ 0	INVx (x=0 to 7)	PWM0x Waveform Output Reverse Control 1: PWM0x Waveform Output Reversed 0: PWM0x Waveform Output Not Reversed
31 ~ 8	-	Reserved

20.10.1.5 PWM0 dead time setting register PWM0_DFR

Register	Read/Write	Description	Reset Value
PWM0_DFR	R/W	PWM0 Dead Time Setting 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
-	-	-	-	PDF[3:0]			
7	6	5	4	3	2	1	0
-	-	-	-	PDR[3:0]			

Bit Number	Bit Symbol	Description
11 ~ 8	PDF[3:0]	Falling Edge Dead Time Bit This bit is valid only in complementary mode: PWM Falling Edge Dead Time = $4 \times \text{PDF}[3:0] / f_{\text{PWM0}}$
3 ~ 0	PDR[3:0]	Rising Edge Dead Time Bit This bit is valid only in complementary mode: PWM Rising Edge Dead Time = $4 \times \text{PDR}[3:0] / f_{\text{PWM0}}$
31 ~ 12 7 ~ 4	-	Reserved

20.10.1.6 PWM0 fault detection setting register PWM0_FLT

Register	Read/Write	Description	Reset Value
PWM0_FLT	R/W	PWM0 Fault Detection Setting 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
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
FLTEN	-	FLTMD	FLTLV	-	-	FLTDT[1:0]	

Bit Number	Bit Symbol	Description
7	FLTEN	PWM Fault Detection Function Control Bit 0: Fault detection function disabled 1: Fault detection function enabled
5	FLTMD	PWM Fault Detection Mode Setting Bit 0: Latch mode. When fault input is active, the FLTSTA flag is set by hardware and PWM output stops. When fault input is inactive, FLTSTA remains unchanged. 1: Immediate mode. When fault input is active, the FLTSTA flag is set by hardware and PWM output stops. When fault input is inactive, FLTSTA is immediately cleared by hardware, and PWM output resumes when the counter reaches zero.
4	FLTLV	PWM fault detection level selection 0: Fault detection active low 1: Fault detection active high
1 ~ 0	FLTDT[1:0]	PWM fault detection input signal filter time setting 00: Filter time = 0 01: Filter time = 1 μs 10: Filter time = 4 μs

Bit Number	Bit Symbol	Description
		11: Filter time = 16 μ s
31 ~ 8 6,3 ~ 2	-	Reserved

20.10.1.7 PWM0 period register PWM0_CYCLE

Register	Read/Write	Description	Reset Value
PWM0_CYCLE	R/W	PWM0 period 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
PWMPD[15:8]							
7	6	5	4	3	2	1	0
PWMPD[7:0]							

Bit Number	Bit Symbol	Description
15 ~ 0	PWMPD[15:0]	PWM0 period setting bits The value represents (period - 1) of the PWM output waveform; PWM output period = (PWMPD[15:0] + 1) $\times$ f PWM0
31 ~ 16	-	Reserved

20.10.1.8 PWM0 channel duty cycle adjustment register PWM0_DT_x (x = 0 to 7)

Register	Read/Write	Description	Reset Value
PWM0_DT _x (x = 0 ~ 7)	R/W	PWM0 Channel n Duty 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
PDT[15:8]							
7	6	5	4	3	2	1	0
PDT[7:0]							

Bit Number	Bit Symbol	Description
15 ~ 0	PDT[15:0]	PWM0 _x waveform duty cycle setting, x = 0 to 7 Independent mode: high-level pulse width of PWM0_x waveform is PDT_x[15:0] PWM clocks.

Bit Number	Bit Symbol	Description
		Complementary mode: complementary channels PWM0x and PWM0y, where $y = x + 1$, share a high-level pulse width of PDTx[15:0] PWM clocks.
31 ~ 16	-	Reserved

20.10.2 PWM0 register mapping

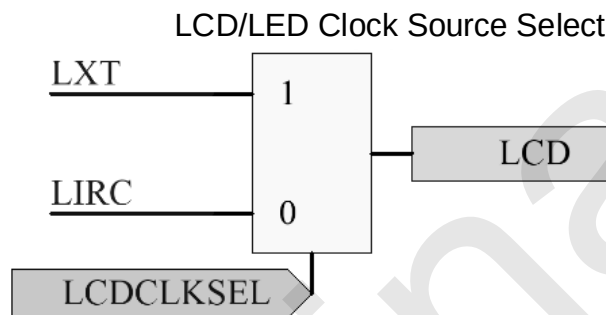
Register	Offset Address	Read/Write	Description	Reset Value
PWM0 Base Address: 0x4002_0200				
PWM0_CON	0x00	R/W	PWM0 Control Register	0x0000_0000
PWM0_CHN	0x04	R/W	PWM0 channel configuration register	0x0000_0000
PWM0_STS	0x08	R/W	PWM0 Status Flag Register	0x0000_0000
PWM0_INV	0x0C	R/W	PWM0 Waveform Output Reverse Control Register	0x0000_0000
PWM0_DFR	0x10	R/W	PWM0 Dead Time Setting Register	0x0000_0000
PWM0_FLT	0x14	R/W	PWM0 Fault Detection Setting Register	0x0000_0000
PWM0_CYCLE	0x18	R/W	PWM0 period register	0x0000_0000
PWM0_DT _x (x = 0 ~ 7) Base Address: 0x4002_0230				
PWM0_DT0	0x00	R/W	PWM0 Channel 0 Duty Register	0x0000_0000
PWM0_DT1	0x04	R/W	PWM0 Channel 1 Duty Register	0x0000_0000
PWM0_DT2	0x08	R/W	PWM0 Channel 2 Duty Register	0x0000_0000
PWM0_DT3	0x0C	R/W	PWM0 Channel 3 duty register	0x0000_0000
PWM0_DT4	0x10	R/W	PWM0 Channel 4 duty register	0x0000_0000
PWM0_DT5	0x14	R/W	PWM0 Channel 5 duty register	0x0000_0000
PWM0_DT6	0x18	R/W	PWM0 Channel 6 duty register	0x0000_0000
PWM0_DT7	0x1C	R/W	PWM0 Channel 7 duty register	0x0000_0000

21 LCD/LED driver

- LCD or LED selectable, sharing registers and ports

21.1 Clock source

- The SC32F11T/11G Series LCD/LED supports two clock sources: LXT and LIRC.



21.2 Built-in 8 COM x 31 SEG LED driver

- Voltage driving method with 1/1 to 1/8 duty cycle
- Software-supportable switching between common cathode and common anode modes

21.3 Built-in 8 COM x 31 SEG LCD driver

- Selectable Type A / Type B waveform
- 8 x 31, 6 x 33, 5 x 34, or 4 x 35
- Selectable voltage division resistor at LCD voltage output port
- LCD display driving bias voltage
 - 1/4 bias voltage
 - 1/3 bias voltage
- Three selectable frame frequencies:
 - 32/64/128 Hz in Type A mode
 - 64/128/256 Hz in Type B mode

21.4 LCD/LED registers

21.4.1 LCD/LED related register table

21.4.1.1 Display drive control register (DDR_CON)

Register	Read/Write	Description	Reset Value
DDR_CON	R/W	Display Drive 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
TRIMODE	TRICOM	-	-	-	-	DDRCK[1:0]	
7	6	5	4	3	2	1	0
DDRON	-	-	TPYE	VOIRSIF	-	BIAS	DMOD

Bit Number	Bit Symbol	Description
15	TRIMODE	Custom Frame Frequency Mode Control Bit 0: Disable Custom Frame Frequency Mode 1: Enable Custom Frame Frequency Mode Note: Custom frame frequency mode requires synchronization with TIM or other peripheral interrupts to control frame frequency. When selecting LXT as the clock source with custom frame frequency mode enabled, ensure the crystal oscillator is fully stabilized before writing '1' to TRICOM to switch the COM scanning port.
14	TRICOM	After enabling custom frame frequency mode, each write of '1' to this bit toggles the COM output once. If the duty cycle is set to 1/8, By default, output starts at COM7 and ends at COM0. If the duty cycle is set to 1/4 By default, output starts at COM7 and ends at COM3.
9 ~ 8	DDRCK[1:0]	LCD/LED frame frequency divider setting bit 00: B waveform frame frequency 64 Hz, A waveform frame frequency 32 Hz 01: B waveform frame frequency 128 Hz, A waveform frame frequency 64 Hz 10: B waveform frame frequency 256 Hz, A waveform frame frequency 128 Hz 11: Reserved
7	DDRON	LCD/LED display driver enable control bit 0: Display driver scan disabled 1: Display driver scan enabled
4	TPYE	LCD driver waveform selection bit 0: B waveform 1: A waveform

Bit Number	Bit Symbol	Description
		Note: In LED mode, changing this bit also affects the LED waveform frequency.
3	VOIRSIF	LCD fast charging enable bit 0: Disable fast charging 1: Enable fast charging using a 33k resistor for 5 periods, then switch to the resistor value selected by VOIRS
1	BIAS	LCD display driving bias voltage setting 0: 1/4 bias voltage 1: 1/3 bias voltage
0	DMOD	LCD/LED display driving mode selection bit 0: LCD mode 1: LED mode
31 ~ 16 13 ~ 10 6 ~ 5 2	-	Reserved

21.4.1.2 Display driver configuration register DDR_CFG

Register	Read/Write	Description	Reset Value
DDR_CFG	R/W	Display driver 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
-	-	-	-	VLCD[3:0]			
7	6	5	4	3	2	1	0
SCS	-	DUTY[1:0]		-	-	VOIRS[1:0]	

Bit Number	Bit Symbol	Description
11 ~ 8	VLCD[3:0]	LCD voltage adjustment setting bit LCD output voltage: $V_LCD = V_DD * (17 + VLCD[3:0]) / 32$
7	SCS	LCD/LED segment/common multiplex pin selection bit 0: When configured for 1/4 duty cycle, S0 to S34 serve as segment pins, C4 to C7 serve as common pins 1: When setting a 1/4 duty cycle, S4 to S34 serve as segments and C0 to C3 as commons
5 ~ 4	DUTY[1:0]	LCD/LED Duty Cycle Configuration Bits 00: 1/8 duty cycle, S4 to S34 as segments, C0 to C7 as commons 01: 1/6 duty cycle, S2 to S34 as segments, C2 to C7 as commons 10: 1/5 duty cycle, S1 to S34 as segments, C3 to C7 as commons 11: 1/4 duty cycle, S0 to S34 as segments, C4 to C7 as commons, or S4 to S34 as segments, C0 to C3 as commons
1 ~ 0	VOIRS[1:0]	LCD Voltage Output Port Voltage Division Resistor Selection Bits 00: Internal voltage division resistor total resistance set to 33 kΩ 01: Internal voltage division resistor total resistance set to 100 kΩ 10: Set internal voltage divider total resistance to 300k 11: Set internal voltage divider total resistance to 800k

Bit Number	Bit Symbol	Description
31 ~ 12 6 3 ~ 2	-	Reserved

21.4.1.3 SEG port enable register SEG_EN0

Register	Read/Write	Description	Reset Value
SEG_EN0	R/W	SEG port enable register 0	0x0000_0000

31	30	29	28	27	26	25	24
SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24
23	22	21	20	19	18	17	16
SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16
15	14	13	12	11	10	9	8
SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8
7	6	5	4	3	2	1	0
SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0

Bit Number	Bit Symbol	Description
31 ~ 0	SEGx (x=0 to 34)	SEGx port display drive output control bit, x = 0 to 34 0: Disable display drive output on SEGx port 1: Enable display drive output on SEGx port

21.4.1.4 SEG port enable register SEG_EN1

Register	Read/Write	Description	Reset Value
SEG_EN1	R/W	SEG port enable register 1	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
-	-	-	-	-	SEG34	SEG33	SEG32

Bit Number	Bit Symbol	Description
2 ~ 0	SEGx (x=0 to 34)	SEGx port display drive output control bit, x = 0 to 34 0: Disable display drive output on SEGx port 1: Enable display drive output on SEGx port
31 ~ 3	-	Reserved

21.4.1.5 COM port enable register COM_EN

Register	Read/Write	Description	Reset Value
COM_EN	R/W	COM port enable 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
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0

Bit Number	Bit Symbol	Description
7 ~ 0	COM _x (x=0 to 7)	COM _x port display drive output control bit, x = 0 to 7 0: Disable display drive output on COM _x port 1: Enable display drive output on COM _x port
31 ~ 8	-	Reserved

21.4.1.6 SEGR_n display register SEGR_n

Register	Read/Write	Description	Reset Value
SEGR _n (n=0 ~ 34)	R/W	SEGR _n Display RAM 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
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0

Bit Number	Bit Symbol	Description
7 ~ 0	-	Control bit for SEGR _n display driver output corresponding to COM _m , where n = 0-34 and m = 0-7 Configures SEGR _n display driver output corresponding to COM _m 0: Off 1: On
31 ~ 8	-	Reserved

21.4.2 LCD/LED register mapping

Register	Offset Address	Read/Write	Description	Reset Value
LCD/LED Base Address: 0x4002_2280				
DDR_CON	0x00	R/W	Display Drive Control Register	0x0000_0000
DDR_CFG	0x04	R/W	Display driver configuration register	0x0000_0000
SEG_EN0	0x08	R/W	SEG Port Enable Register	0x0000_0000
SEG_EN1	0x0C	R/W	SEG Port Enable Register	0x0000_0000
COM_EN	0x10	R/W	COM port enable register	0x0000_0000
SEGR Base Address: 0x4002_2330				
SEGR0	0x00	R/W	SEG0 Display Register	0x0000_0000
SEGR1	0x04	R/W	SEG1 Display Register	0x0000_0000
SEGR2	0x08	R/W	SEG2 Display Register	0x0000_0000
SEGR3	0x0C	R/W	SEG3 Display Register	0x0000_0000
SEGR4	0x10	R/W	SEG4 Display Register	0x0000_0000
SEGR5	0x14	R/W	SEG5 Display Register	0x0000_0000
SEGR6	0x18	R/W	SEG6 Display Register	0x0000_0000
SEGR7	0x1C	R/W	SEG7 Display Register	0x0000_0000
SEGR8	0x20	R/W	SEG8 display register	0x0000_0000
SEGR9	0x24	R/W	SEG9 display register	0x0000_0000
SEGR10	0x28	R/W	SEG10 display register	0x0000_0000
SEGR11	0x2C	R/W	SEG11 display register	0x0000_0000
SEGR12	0x30	R/W	SEG12 display register	0x0000_0000
SEGR13	0x34	R/W	SEG13 display register	0x0000_0000
SEGR14	0x38	R/W	SEG14 display register	0x0000_0000
SEGR15	0x3C	R/W	SEG15 display register	0x0000_0000
SEGR16	0x40	R/W	SEG16 display register	0x0000_0000
SEGR17	0x44	R/W	SEG17 display register	0x0000_0000

Register	Offset Address	Read/Write	Description	Reset Value
SEGR18	0x48	R/W	SEG18 display register	0x0000_0000
SEGR19	0x4C	R/W	SEG19 display register	0x0000_0000
SEGR20	0x50	R/W	SEG20 display register	0x0000_0000
SEGR21	0x54	R/W	SEG21 display register	0x0000_0000
SEGR22	0x58	R/W	SEG22 display register	0x0000_0000
SEGR23	0x5C	R/W	SEG23 display register	0x0000_0000
SEGR24	0x60	R/W	SEG24 display register	0x0000_0000
SEGR25	0x64	R/W	SEG25 display register	0x0000_0000
SEGR26	0x68	R/W	SEG26 Segment Display Register	0x0000_0000
SEGR27	0x6C	R/W	SEG27 Segment Display Register	0x0000_0000
SEGR28	0x70	R/W	SEG28 Segment Display Register	0x0000_0000
SEGR29	0x74	R/W	SEG29 Segment Display Register	0x0000_0000
SEGR30	0x78	R/W	SEG30 Segment Display Register	0x0000_0000
SEGR31	0x7C	R/W	SEG31 Segment Display Register	0x0000_0000
SEGR32	0x80	R/W	SEG32 Segment Display Register	0x0000_0000
SEGR33	0x84	R/W	SEG33 Segment Display Register	0x0000_0000
SEGR34	0x88	R/W	SEG34 Segment Display Register	0x0000_0000

22 42-channel high sensitivity touch circuit (TK)

- High sensitivity mode
- Suitable for capacitive proximity keys, proximity sensing, and other touch applications requiring high sensitivity
- Channels can be scanned in parallel
- CMOD pin requires a 103 capacitor connected to ground
- Supports self-capacitance and mutual-capacitance modes
- TK's internal reference voltage can be set up to 4 V at the highest level to achieve higher sensitivity
- Supports low power mode
- Supports fast wake up from STOP mode
- Complete development support: highly flexible touch software library and intelligent debug software

Note: Only the SC32F11T series supports the TK function.

23 16-bit timers/counters (TIM) Timer0 to Timer7

23.1 Clock source

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

23.2 Feature

- Eight independent 16-bit auto-reload counters Timer0 to Timer7
- 16-bit upcounting, downcounting, and up/down auto-reload counters
- Supports rising edge/falling edge capture, enabling PWM duty cycle and period capture
- 16-bit programmable prescaler, with division factor of the counter clock frequency adjustable between 1 and 65535
- Overflow and capture events of TIM1/2/6 can generate DMA requests
- TIM2 to 7 Tn/TnCAP/TnPWMA pins support remapping and can be mapped to another IO group.

23.3 Counting method

23.3.1 TIM counting method in timer mode

- Count up: counts up from the set value to 0xFFFF before overflowing.
- Count down: counts down from 0xFFFF to the set value.

23.3.2 TIM counting method in PWM output mode

Only count up is allowed in PWM output mode: counting starts from 0 and counts up; when reaching the duty cycle setting PDT, the PWM output waveform switches between high and low levels, then continues counting up to the set reload value RLD, at which point it overflows and restarts counting from 0.

PWM period T of TIM output is calculated as follows:

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

Duty cycle calculation formula:

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

23.4 Timer-related signal pins

- TnCAP/Tn, n=0 to 7
 - Tn clock input/output
 - TnCAP can capture on both Rising Edge and Falling Edge
 - Note: Tn and TnCAP are multiplexed functions and cannot be used simultaneously
- TnEX, n=0 to 7

- In reload mode, the external event input on the TnEX pin (Falling Edge) is used to control reload enable or disable.
- In capture mode, when FSEL = 1, the TnEX pin acts as a Falling Edge capture signal input. Upon detecting a falling edge on TnEX, a capture event occurs, EXIF is set, and the value in the TnCNT register is stored in the FCAP register.
- TnPWM,n=0 to 7
 - Each TIM channel can provide two PWM signals with the same Period via the Tn and TnEX ports, each having independently adjustable duty cycles: TnPWMA and TnPWMB.
 - TnPWMA and TnPWMB share the same Period, and their clock source follows the TIM.

Note: The PWM Capture and PWM Output functions of TIM cannot be enabled simultaneously.

23.5 TIM interrupts and their corresponding flags

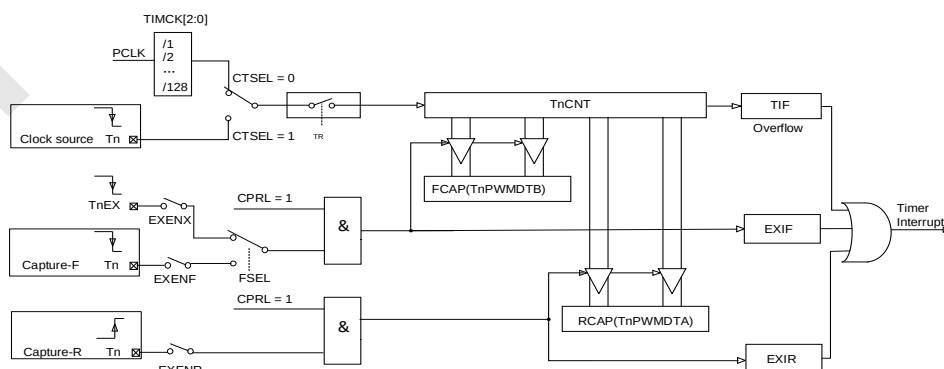
- Counter overflow/underflow share the interrupt flag TIF.
- Capture Status Flag
 - EXIF flag indicates a falling edge has been detected on the external event input.
 - EXIR flag indicates a rising edge has been detected on the external event input.
- Interrupt and Priority Configuration Control Bits are integrated into the NVIC module.

23.6 TIM operating modes

23.6.1 Mode 0: 16-bit capture PWM both-edge capture mode

TnEX is the PWM capture input. Internally, measurement is divided into separate channels that sample the rising and falling edges respectively:

- Dedicated 16-bit rising edge capture counter register - multiplexed with PWMA duty cycle setting register TIMn_PDTA
- Dedicated 16-bit falling edge capture counter register - multiplexed with PWMB duty cycle setting register TIMn_PDTB
- Capture flag bits:
 - Separate enable bits for rising and falling edge capture
 - Separate capture flags for rising and falling edges



Both-edge capture block diagram

23.6.2 Mode 1: 16-bit auto-reload timer/counter mode

In 16-bit auto-reload mode, the timer can operate in upcounting or downcounting mode. This is enabled by setting the DEC bit to 1 in TIMn_CON (n=0 to 7), with the counting direction controlled by the TnEX pin. After system reset, the DEC bit resets to 0, so timer n defaults to upcounting. When DEC is set to 1, timer n counts up or down based on the level of the TnEX pin.

When DEC = 0, two options are selectable via the EXENX bit in TIMn_CON:

If EXENX = 0, when TIMn_CNT reaches 0xFFFF, the timer overflow flag TIF is set, and the timer automatically reloads TIMn_CNT with the 16-bit value from the auto-reload register TIMn_RLD.

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

When DCEN = 1, the TnEX pin controls the counting direction, and EXENX is ignored.

If TnEX = 1, TIMn counts up. When TIMn_CNT increments to 0xFFFF, the timer overflow bit TIF is set. The timer then automatically loads the 16-bit value from the reload register TIMn_RLD into TIMn_CNT.

If TnEX = 0, TIMn counts down. When TIMn_CNT decrements from 0xFFFF down to TIMn_RLD, the timer overflows, setting the TIF overflow flag, and reloads 0xFFFF into TIMn_CNT.

In this mode, EXFIF is not used as an interrupt flag regardless of TIMn overflow.

23.6.3 Mode 3: programmable clock output mode

In this mode, TIMn (n=0 to 7) can be programmed to output a 50% duty cycle clock: enabling TIMn as a clock generator when CTSEL=0 and TnOE=1.

The clock frequency output on Tn in this mode is:

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

23.6.4 Mode 4: PWM output mode

- PWM Duty Cycle Variation Feature

Modifying PDTx[15:0] does not immediately change the duty cycle; the update takes effect from the next period after the current one ends.

- PWM Period Change Feature

Implemented by modifying the values in the period setting registers [RCAPXL / RCAPXH]. Let Tn be the current period count, Tm the timer value when writing the period register, and Tx the updated period count, then:

- If $T_m \leq T_x$: the period updates immediately to Tx
- If $T_m > T_x$: the period change proceeds in two stages. Stage one: after writing the period register, the period counter increments from the current value until it overflows and resets. Stage two: the period updates to Tx.

23.7 TIM interrupt

In timer or counter mode, when CNT reaches the TIMn count value, TIF is set. If TIMn_IDE.INTEN = 1, an interrupt is triggered.

In external event input mode, detecting a valid edge transition sets the EXIR/EXIF flag. If TIMn_IDE.INTEN=1, an interrupt is generated.

Interrupt Event	Event Flag	Global Interrupt Enable Control Bit	Module Interrupt Sub-Control Bit
Timer overflow	TIF	TIMn_IDE->INTEN (n=0 to 7)	TIMn_IDE->TIE
External event input rising edge interrupt	EXIR		TIMn_IDE->EXRIE
External event input falling edge interrupt	EXIF		TIMn_IDE->EXFIE

23.8 TIM registers

23.8.1 TIM-related register table

23.8.1.1 Timer control register TIMn_CON

Register	Read/Write	Description	Reset Value
TIMn_CON (n=0 ~ 7)	R/W	Timer control register	0x0000_0000

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

Bit Number	Bit Symbol	Description
21	SPOS	TIM2 pin mapping control bits @ TIM2_CON
		Signal SPOS Value
		T2CAP/T2/T2PWMA
		SPOS=0
		PB6
		SPOS=1
		PB8
		TIM3 pin mapping control bits @ TIM3_CON
		Signal SPOS Value
		T3CAP/T3/T3PWMA
		SPOS=0
		PA4
		SPOS=1
		PB9
		TIM4 pin mapping control bits @ TIM4_CON

Bit Number	Bit Symbol	Description																								
		<table><tr><td>Signal SPOS Value</td><td>T4CAP/T4/T4PWMA</td></tr><tr><td>SPOS=0</td><td>PD2</td></tr><tr><td>SPOS=1</td><td>PB10</td></tr></table> TIM5 pin mapping control bits @ TIM5_CON <table><tr><td>Signal SPOS Value</td><td>T5CAP/T5/T5PWMA</td></tr><tr><td>SPOS=0</td><td>PA14</td></tr><tr><td>SPOS=1</td><td>PB11</td></tr></table> TIM6 pin mapping control bits @ TIM6_CON <table><tr><td>Signal SPOS Value</td><td>T6CAP/T6/T6PWMA</td></tr><tr><td>SPOS=0</td><td>PA13</td></tr><tr><td>SPOS=1</td><td>PB12</td></tr></table> TIM7 Pin Mapping Control Bit @ TIM7_CON <table><tr><td>Signal SPOS Value</td><td>T7CAP/T7/T7PWMA</td></tr><tr><td>SPOS=0</td><td>PD3</td></tr><tr><td>SPOS=1</td><td>PB13</td></tr></table>	Signal SPOS Value	T4CAP/T4/T4PWMA	SPOS=0	PD2	SPOS=1	PB10	Signal SPOS Value	T5CAP/T5/T5PWMA	SPOS=0	PA14	SPOS=1	PB11	Signal SPOS Value	T6CAP/T6/T6PWMA	SPOS=0	PA13	SPOS=1	PB12	Signal SPOS Value	T7CAP/T7/T7PWMA	SPOS=0	PD3	SPOS=1	PB13
Signal SPOS Value	T4CAP/T4/T4PWMA																									
SPOS=0	PD2																									
SPOS=1	PB10																									
Signal SPOS Value	T5CAP/T5/T5PWMA																									
SPOS=0	PA14																									
SPOS=1	PB11																									
Signal SPOS Value	T6CAP/T6/T6PWMA																									
SPOS=0	PA13																									
SPOS=1	PB12																									
Signal SPOS Value	T7CAP/T7/T7PWMA																									
SPOS=0	PD3																									
SPOS=1	PB13																									
15	TXOE	Tn Pin Signal Direction Control Bit 0: Tn configured as Clock Input or I/O Pin 1: Tn configured for Programmable Clock Output																								
14	EPWMNA	PWM Output Enable Bit for Pin with Tn_PWMA 0: Off 1: Enable																								
13	EPWMNB	PWM Output Enable Bit for Pin with TnEX_PWMB 0: Off 1: Enable																								
12	INVNA	TPWMnA Waveform Output Inversion Control Bit 0: Normal 1: Output Inverted																								
11	INVNB	TPWMnB Waveform Output Inversion Control Bit 0: Normal 1: Output Inverted																								
10 ~ 8	TIMCK[2:0]	TIM Clock Frequency Range Control Bit 000:f _{TIM} /1 001:f _{TIM} /2 010:f _{TIM} /4 011:f _{TIM} /8 100:f _{TIM} /16 101:f _{TIM} /32 110:f _{TIM} /64 111:f _{TIM} /128 f _{TIM} clock source may be either f _{PCLK} or Tn input																								
7	TR	TIMn Start/Stop Control Bit, n=0 to 7 0: Stop TIMn / TPWMn Counter 1: Start TIMn / TPWMn timer/counter																								
6	DEC	Increment/Decrement direction control bit 0: TIMn operates as an incrementing timer/counter																								

Bit Number	Bit Symbol	Description
		1: TIMn operates as an incrementing/decrementing timer/counter, with TnEX selecting the count direction
5	EXENX	TnEX control bit, n=0 to 7 This bit has different functions depending on the mode: ● Reload mode (CPRL = 0) Controls external event input (falling edge) on the TnEX pin to enable or disable reload: 0: Ignore events on the TnEX pin 1: Detect a falling edge on the TnEX pin to trigger a reload ● Capture Mode (CPRL = 1) This bit serves as the gating enable bit for TnEX falling edge signal capture: 0: Ignore events on the TnEX pin 1: When FSEL = 1, detecting a falling edge on the TnEX pin triggers a capture, sets the EXIF flag, and captures the TnCNT register value into the FCAP register.
4	FSEL	Falling edge signal select bit This bit is valid only in Capture Mode (CPRL = 1) 0: A falling edge detected on the Tn pin triggers a capture; events on the TnEX pin are ignored. 1: A falling edge detected on the TnEX pin triggers a capture; events on the Tn pin are ignored.
3	EXENF	Falling edge capture enable bit: 0: Ignore events on the Tn pin 1: Trigger a capture on a falling edge at the Tn pin, set EXIF, and store TnCNT value in the FCAP register
2	EXENR	Rising edge capture enable bit on the Tn pin 0: Ignore events on the Tn pin 1: Trigger a capture on a rising edge at the Tn pin, set EXIR, and store TnCNT value in the RCAP register
1	CTSEL	Timer/Counter selection bit 0: Timer mode 1: Counter mode
0	CPRL	Capture/Reload mode setting bit 0: Reload function 1: Capture function
31 ~ 22 20 ~ 16	-	Reserved

23.8.1.2 Timer count value register TIMn_CNT

Register	Read/Write	Description	Reset Value
TIMn_CNT (n=0 ~ 7)	R/W	Timer count value 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
CNT[15:8]							
7	6	5	4	3	2	1	0
CNT[7:0]							

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

23.8.1.3 Timer reload register TIMn_RLD

Register	Read/Write	Description	Reset Value
TIMn_RLD (n=0 ~ 7)	R/W	Timer reload 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
RLD [15:8]							
7	6	5	4	3	2	1	0
RLD [7:0]							

Bit Number	Bit Symbol	Description
15 ~ 0	RLD[15:0]	A 16-bit reload is triggered by either timer overflow or a falling edge on the external input TnEX. Upon reload, the timer automatically loads the RLD[15:0] value written by user software into the TnCNT register.
31 ~ 16	-	Reserved

23.8.1.4 Timer status register TIMn_STS

Register	Read/Write	Description	Reset Value
TIMn_STS (n=0 ~ 7)	R/W	Timer status 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
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	-	EXIF	EXIR	TIF

Bit Number	Bit Symbol	Description
2	EXIF	Flag indicating detection of a falling edge on the external event input. This bit is set by hardware and cleared by writing 1 in software. 0: No external event input

Bit Number	Bit Symbol	Description
		1: External input detected (set by hardware if EXENF = 1) Note: In capture mode, the TnFCAP value cannot be updated until this bit is cleared by software.
1	EXIR	Flag indicating a rising edge detected on the external event input of the Tn pin. This bit is set by hardware and cleared by writing 1 in software. 0: No external event input 1: External input detected (set by hardware if EXENR = 1). Note: In capture mode, the TnRCAP value cannot be updated until this bit is cleared by software.
0	TIF	Timer overflow flag. This bit is set by hardware and cleared by writing 1 in software. 0: No overflow (must be cleared by software). 1: Overflow occurred (set by hardware if RCLK = 0 and TCLK = 0).
31 ~ 3	-	Reserved

23.8.1.5 TnPWMA duty cycle setting register TIMn_PDTA (@CPRL = 0) .

Register	Read/Write	Description	Reset Value
TIMn_PDTA (n=0 ~ 7)	R/W	TnPWMA duty cycle setting register TIMn_PDTA (@CPRL = 0).	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 Symbol	Description
15 ~ 0	PDT[15:0]	TPWMnA duty cycle register, n = 0 to 7. The high-level pulse width of the TPWMnA waveform is PDT[15:0] TIM clock cycles
31 ~ 16	-	Reserved

23.8.1.6 TIMn_PDTB duty cycle register for TnPWMB (@CPRL = 0)

Register	Read/Write	Description	Reset Value
TIMn_PDTB (n=0 ~ 7)	R/W	TIMn_PDTB duty cycle register for TnPWMB (@CPRL = 0)	0x0000_0000

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

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
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 Symbol	Description
15 ~ 0	PDT[15:0]	TPWMnB duty cycle register, n = 0 to 7 The high-level pulse width of the TPWMnB waveform is PDT[15:0] TIM clock cycles
31 ~ 16	-	Reserved

23.8.1.7 Rising edge capture data register TIMn_RCAP (@CPRL = 1)

Register	Read/Write	Description	Reset Value
TIMn_RCAP (n=0 ~ 7)	R/W	Rising edge capture data register TIMn_RCAP (@CPRL = 1)	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 Symbol	Description
15 ~ 0	RCAP [15:0]	In TIMn PWM capture mode, when a rising edge capture condition occurs, the CNT counter value is stored in this register.
31 ~ 16	-	Reserved

23.8.1.8 Falling edge capture data register TIMn_FCAP (@CPRL = 1)

Register	Read/Write	Description	Reset Value
TIMn_FCAP (n=0 ~ 7)	R/W	Falling edge capture data register TIMn_FCAP (@CPRL = 1)	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 Symbol	Description
15 ~ 0	FCAP [15:0]	In TIMn PWM capture mode, when a falling edge capture condition occurs, the CNT counter value is stored in this register.
31 ~ 16	-	Reserved

23.8.1.9 TIMn interrupt enable and DMA control register (TIMn_IDE)

Register	Read/Write	Description	Reset Value
TIMn_IDE n=0 ~ 7	R/W	TIMn 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
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	CAPFDE	CAPRDE	TIDE	EXFIE	EXRIE	TIE	INTEN

Bit Number	Bit Symbol	Description
6	CAPFDE	DMA Request Enable Bit for Falling Edge Capture Event 0: Disable DMA request on falling edge capture event 1: Trigger DMA request on new falling edge capture
5	CAPRDE	DMA Request Enable Bit for Rising Edge Capture Event 0: Disable DMA request on rising edge capture event 1: Trigger DMA request on new rising edge capture
4	TIDE	DMA Request Enable Bit for Timer Overflow Event 0: Disable DMA request on timer overflow 1: Enable DMA request on timer overflow
3	EXFIE	External event input falling edge interrupt enable 0: Disable falling edge interrupt 1: Enable falling edge interrupt
2	EXRIE	External event input rising edge interrupt enable 0: Disable rising edge interrupt 1: Enable rising edge interrupt
1	TIE	Timer overflow interrupt enable 0: Disable overflow interrupt 1: Enable overflow interrupt
0	INTEN	CPU Interrupt Request Enable Control Bit 0: Disable interrupt request 1: Enable interrupt request
31 ~ 7	-	Reserved

23.8.2 TIM register mapping

Register	Offset Address	Read/Write	Description	Reset Value
TIM0 base address: 0x4002_0100				
TIM0_CON	0x00	R/W	Timer control register	0x0000_0000
TIM0_CNT	0x04	R/W	Timer count value register	0x0000_0000
TIM0_RLD	0x08	R/W	Timer reload register	0x0000_0000
TIM0_STS	0x0C	R/W	Timer flag register	0x0000_0000
TIM0_PDTA	0x10	R/W	TnPWMA duty cycle setting register (@CPRL = 0)	0x0000_0000
TIM0_RCAP	0x10	R/W	Rising edge data capture register (@CPRL = 1)	0x0000_0000
TIM0_PDTB	0x14	R/W	TnPWMB duty cycle setting register (@CPRL = 0)	0x0000_0000
TIM0_FCAP	0x14	R/W	Falling edge data capture register (@CPRL = 1)	0x0000_0000
TIM0_IDE	0x18	R/W	TIMn Interrupt Enable and DMA Control Register	0x0000_0000
TIM1 Base Address: 0x4002_0140				
TIM1_CON	0x00	R/W	Timer control register	0x0000_0000
TIM1_CNT	0x04	R/W	Timer count value register	0x0000_0000
TIM1_RLD	0x08	R/W	Timer reload register	0x0000_0000
TIM1_STS	0x0C	R/W	Timer Flag Register	0x0000_0000
TIM1_PDTA	0x10	R/W	TnPWMA duty cycle setting register (@CPRL = 0)	0x0000_0000
TIM1_RCAP	0x10	R/W	Rising edge data capture register (@CPRL = 1)	0x0000_0000
TIM1_PDTB	0x14	R/W	TnPWMB duty cycle setting register (@CPRL = 0)	0x0000_0000
TIM1_FCAP	0x14	R/W	Falling edge data capture register (@CPRL = 1)	0x0000_0000
TIM1_IDE	0x18	R/W	TIMn Interrupt Enable and DMA Control Register	0x0000_0000
TIM2 Base Address: 0x4002_0180				
TIM2_CON	0x00	R/W	Timer control register	0x0000_0000

Register	Offset Address	Read/Write	Description	Reset Value
TIM2_CNT	0x04	R/W	Timer count value register	0x0000_0000
TIM2_RLD	0x08	R/W	Timer reload register	0x0000_0000
TIM2_STS	0x0C	R/W	Timer Flag Register	0x0000_0000
TIM2_PDTA	0x10	R/W	TnPWMA duty cycle setting register (@CPRL = 0)	0x0000_0000
TIM2_RCAP	0x10	R/W	Rising edge data capture register (@CPRL = 1)	0x0000_0000
TIM2_PDTB	0x14	R/W	TnPWMB duty cycle setting register (@CPRL = 0)	0x0000_0000
TIM2_FCAP	0x14	R/W	Falling edge data capture register (@CPRL = 1)	0x0000_0000
TIM2_IDE	0x18	R/W	TIMn Interrupt Enable and DMA Control Register	0x0000_0000
TIM3 Base Address: 0x4002_01C0				
TIM3_CON	0x00	R/W	Timer control register	0x0000_0000
TIM3_CNT	0x04	R/W	Timer count value register	0x0000_0000
TIM3_RLD	0x08	R/W	Timer reload register	0x0000_0000
TIM3_STS	0x0C	R/W	Timer Flag Register	0x0000_0000
TIM3_PDTA	0x10	R/W	TnPWMA duty cycle setting register (@CPRL = 0)	0x0000_0000
TIM3_RCAP	0x10	R/W	Rising edge data capture register (@CPRL = 1)	0x0000_0000
TIM3_PDTB	0x14	R/W	TnPWMB duty cycle setting register (@CPRL = 0)	0x0000_0000
TIM3_FCAP	0x14	R/W	Falling edge data capture register (@CPRL = 1)	0x0000_0000
TIM3_IDE	0x18	R/W	TIMn Interrupt Enable and DMA Control Register	0x0000_0000
TIM4 Base Address: 0x4002_1100				
TIM4_CON	0x00	R/W	Timer control register	0x0000_0000
TIM4_CNT	0x04	R/W	Timer count value register	0x0000_0000
TIM4_RLD	0x08	R/W	Timer reload register	0x0000_0000
TIM4_STS	0x0C	R/W	Timer Flag Register	0x0000_0000

Register	Offset Address	Read/Write	Description	Reset Value
TIM4_PDTA	0x10	R/W	TnPWMA duty cycle setting register (@CPRL = 0)	0x0000_0000
TIM4_RCAP	0x10	R/W	Rising edge data capture register (@CPRL = 1)	0x0000_0000
TIM4_PDTB	0x14	R/W	TnPWMB duty cycle setting register (@CPRL = 0)	0x0000_0000
TIM4_FCAP	0x14	R/W	Falling edge data capture register (@CPRL = 1)	0x0000_0000
TIM4_IDE	0x18	R/W	TIMn Interrupt Enable and DMA Control Register	0x0000_0000
TIM5 Base Address: 0x4002_1140				
TIM5_CON	0x00	R/W	Timer control register	0x0000_0000
TIM5_CNT	0x04	R/W	Timer count value register	0x0000_0000
TIM5_RLD	0x08	R/W	Timer reload register	0x0000_0000
TIM5_STS	0x0C	R/W	Timer Flag Register	0x0000_0000
TIM5_PDTA	0x10	R/W	TnPWMA duty cycle setting register (@CPRL = 0)	0x0000_0000
TIM5_RCAP	0x10	R/W	Rising edge data capture register (@CPRL = 1)	0x0000_0000
TIM5_PDTB	0x14	R/W	TnPWMB duty cycle setting register (@CPRL = 0)	0x0000_0000
TIM5_FCAP	0x14	R/W	Falling edge data capture register (@CPRL = 1)	0x0000_0000
TIM5_IDE	0x18	R/W	TIMn Interrupt Enable and DMA Control Register	0x0000_0000
TIM6 Base Address: 0x4002_1180				
TIM6_CON	0x00	R/W	Timer control register	0x0000_0000
TIM6_CNT	0x04	R/W	Timer count value register	0x0000_0000
TIM6_RLD	0x08	R/W	Timer reload register	0x0000_0000
TIM6_STS	0x0C	R/W	Timer Flag Register	0x0000_0000
TIM6_PDTA	0x10	R/W	TnPWMA duty cycle setting register (@CPRL = 0)	0x0000_0000
TIM6_RCAP	0x10	R/W	Rising edge data capture register (@CPRL = 1)	0x0000_0000

Register	Offset Address	Read/Write	Description	Reset Value
TIM6_PDTB	0x14	R/W	TnPWMB duty cycle setting register (@CPRL = 0)	0x0000_0000
TIM6_FCAP	0x14	R/W	Falling edge data capture register (@CPRL = 1)	0x0000_0000
TIM6_IDE	0x18	R/W	TIMn Interrupt Enable and DMA Control Register	0x0000_0000
TIM7 Base Address: 0x4002_11C0				
TIM7_CON	0x00	R/W	Timer control register	0x0000_0000
TIM7_CNT	0x04	R/W	Timer count value register	0x0000_0000
TIM7_RLD	0x08	R/W	Timer reload register	0x0000_0000
TIM7_STS	0x0C	R/W	Timer Flag Register	0x0000_0000
TIM7_PDTA	0x10	R/W	TnPWMA duty cycle setting register (@CPRL = 0)	0x0000_0000
TIM7_RCAP	0x10	R/W	Rising edge data capture register (@CPRL = 1)	0x0000_0000
TIM7_PDTB	0x14	R/W	TnPWMB duty cycle setting register (@CPRL = 0)	0x0000_0000
TIM7_FCAP	0x14	R/W	Falling edge data capture register (@CPRL = 1)	0x0000_0000
TIM7_IDE	0x18	R/W	TIMn Interrupt Enable and DMA Control Register	0x0000_0000

24 DMA controller

24.1 Overview

The Direct Memory Access (DMA) controller is used for high-speed data transfer. The DMA controller can transfer data from one address to another without CPU intervention. Data transfer via DMA reduces the CPU workload, allowing the saved CPU resources to be used for other applications. The DMA controller contains 4 channels, each directly connected to dedicated hardware DMA requests, and each channel also supports software triggering. The DMA controller supports 4 levels of channel priority for managing DMA request priorities, ensuring that only one DMA channel operates at any given time. The DMA controller also supports single and burst transfers; the request source can be either a software request or an interface request. Data transfer between memories uses software requests.

Note: For bidirectional data transfer applications, two DMA channels are required to separately perform transmitting and receiving.

24.2 Clock source

The DMA clock source is HCLK; DMA peripheral clock is enabled via AHB_CFG.DMAEN

24.3 Feature

- Supports four independently configurable channels
- Supports four request priorities
- Supports 8-bit, 16-bit, and 32-bit data transfers
- Supports auto-increment or fixed source and destination addresses; data width supports Byte, Half-word, and Word
- Supports single and burst transfer modes

24.4 Function description

24.4.1 Transfer direction

Memory to Memory	Memory to Peripheral	Peripheral to Memory	Peripheral to Peripheral
Unlimited	Unlimited	Unlimited	Unlimited

24.4.2 DMA access region restriction

When users operate DMA, writing to Flash is prohibited, and operating the Core via DMA is also not allowed; otherwise, unpredictable exceptions may occur.

24.4.3 Channel priority

Four priority levels can be set via register PL[1:0]:

- 00: Low
- 01: Medium
- 10: High
- 11: Very High

24.4.4 Single transfer and burst transfer

The DMA controller supports single and burst transfer modes. Request sources can be software or interface requests. Data transfers between Memory regions use software requests. Single transfer means the software or interface prepares to transfer one data unit (each data requires one request). Burst transfer means multiple data units are transferred with only one request.

Single Transfer and Burst Transfer modes can be configured via the register TPTYPE (DMA_n_CFG[15]).

When the DMA controller operates in Single Transfer mode, each data movement requires one request; each time data is moved, the register DMA_n_CNT[31:0], n=0 to 3, decreases by 1 until the count in DMA_n_CNT[31:0] reaches 0, at which point the transfer completes. In this mode, BURSIZE (DMA_n_CFG[14:12]) does not control the amount of data transferred; its value is fixed at 1.

In Burst Transfer mode, the DMA controller transfers DMA_n_CNT[31:0] data items per request. After transferring BURSIZE data (DMA_n_CFG[14:12]), the value in DMA_n_CNT[31:0] decreases by BURSIZE. The data transfer is complete only when the value in DMA_n_CNT[31:0] decreases to zero.

24.5 Loop mode

Loop Mode can be used to handle circular buffers and continuous data streams (such as ADC scan mode). During transmission in Loop Mode, the number of data items to be transferred is automatically reloaded to the initial value set during channel configuration, and DMA requests continue to be serviced. To stop loop transmission, software must first stop the peripheral from generating DMA requests (for example, exit ADC scan mode) before disabling the DMA channel. Software must explicitly set the DMACNT value both before starting/enabling transmission and after stopping loop transmission.

The DMA controller of the SC32F11T/11G Series supports Normal Mode and Loop Mode:

- When CIRC=0 (the DMA channel operates in non-loop mode), no further DMA requests will be accepted after the programmed number of data items to be transferred is reached
- When CIRC=1 (DMA channel is in Loop Mode), after the transfer completes, the channel's DMACNT will automatically reload the previously set value, waiting for the next loop.

Users can choose flexibly according to their actual needs.

24.6 DMA interrupts

For each DMA channel n (n=0 to 3), interrupts are generated on transfer complete, half transfer, or transfer error events. Individual interrupt enable bits improve flexibility.

Interrupt Event	Event Flag	Interrupt Request Control Bit	Sub-event Flags	Interrupt Enable Sub-Bit
DMA Channel n Transfer Complete	GIF	DMA _n _CFG ->INTEN	TCIF	TCIE

Interrupt Event	Event Flag	Interrupt Request Control Bit	Sub-event Flags	Interrupt Enable Sub-Bit
DMA Channel n Half Transfer			HTIF	HTIE
DMA Channel n Transfer Error			TEIF	TEIE

24.7 DMA registers

24.7.1 DMA register table

24.7.1.1 DMA channel n transfer source address buffer register DMA_n_SADR

Register	Read/Write	Description	Reset Value
DMA _n _SADR n = 0 ~ 3	R/W	DMA Channel n Transfer Source Address Buffer Register	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 Symbol	Description
31 ~ 0	SADR[31:0]	DMA Transfer Source Address Buffer ● Read ■ When the channel is enabled, reading returns the internal source address working register ■ When the channel is disabled, reading returns the visible source address buffer register. ● Update ■ After each transfer, the source address working register automatically updates based on SAINC[1:0]; the update width is defined by TXWIDTH[1:0]. ■ In loop mode (SAINC == 11), the source address buffer register reloads into the source address working register. ● Write ■ Writing to the source address buffer register is allowed when CHEN=0, or when CHEN=1 but the DMA channel transfer is complete and the channel is in IDLE state.

24.7.1.2 DMA channel n transfer destination address buffer register DMA_n_DADR

Register	Read/Write	Description	Reset Value
DMA _n _DADR n = 0 ~ 3	R/W	DMA channel n destination address buffer register	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 Symbol	Description
31 ~ 0	DADR[31:0]	DMA transfer destination address buffer ● Read ■ When the channel is open, reads return the internal destination address working register ■ When the channel is disabled, reads return the target address buffer register. ● Update ■ After each transfer, the target address working register automatically updates according to DAINC[1:0], with the increment size determined by TXWIDTH[1:0]. ■ In loop mode (SAINC == 11), the target address buffer register reloads into the target address working register. ● Write ■ Writing to the target address buffer register is allowed when CHEN=0, or when CHEN=1 but the DMA channel has completed the transfer and is in IDLE state.

24.7.1.3 DMA channel n control/configuration register DMA_n_CFG

Register	Read/Write	Description	Reset Value
DMA _n _CFG n = 0 ~ 3	R/W	DMA channel n control/configuration register	0x0000_0000

31	30	29	28	27	26	25	24
-	-	REQSRC[5:0]					
23	22	21	20	19	18	17	16
CHRRQ	-	-	-	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	CHRRST	PAUSE	CIRC	TXWIDTH[1:0]		PL[1:0]	

Bit Number	Bit Symbol	Description
29 ~ 24	REQSRC[5:0]	DMA channel request source selection bits 0: Disable the peripheral request for the current DMA channel Select the configuration below. The peripheral request source for the current DMA channel will generate DMA requests according to the peripheral DMA request enable setting: 2:UART0_IDE->TXDMAEN 3:UART0_IDE->RXDMAEN 4:UART1_IDE->TXDMAEN 5:UART1_IDE->RXDMAEN 12:TWI0_QSPI0_IDE->TXDMAEN 13:TWI0_QSPI0_IDE->RXDMAEN 14:TWI1_QSPI1_IDE->TXDMAEN 15:TWI1_QSPI1_IDE->RXDMAEN 20:TWI2_SPI2_IDE->TXDMAEN 21:TWI2_SPI2_IDE->RXDMAEN 33:TIM1_IDE->TIDE

Bit Number	Bit Symbol	Description
		34:TIM1_IDE->CAPFDE 35:TIM1_IDE->CAPRDE 36:TIM2_IDE->TIDE 37:TIM2_IDE->CAPFDE 38:TIM2_IDE->CAPRDE 48:TIM6_IDE->TIDE 49:TIM6_IDE->CAPFDE 50:TIM6_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 requests
23	CHRQ	DMA request enable bit for the DMA channel 0: Disable. The current DMA channel cannot serve as a request source for other DMA channels 1: Enable. The current DMA channel can serve as a request source for other DMA channels, allowing it to generate DMA requests like peripherals. Enabling this bit allows DMA request chaining, for example: CHRQ = 1: When DMA channel n completes data transfer, it sends a DMA request to channel m. Channel m responds by updating channel n's registers with the pre-configured parameter table, enabling automatic parameter updates for channel n. Note: Once CHRQ is set, the DMA acting as the request source can transfer data, but it will neither set the flag nor trigger the corresponding interrupt. The flag is set and the interrupt triggered only after CHRQ is cleared.
19	TEIE	DMA transfer error interrupt enable bit 0: DMA transfer error interrupt disabled 1: DMA transfer error interrupt enabled
18	HTIE	DMA half transfer interrupt enable bit 0: DMA half transfer interrupt disabled 1: DMA half transfer interrupt enabled
17	TCIE	DMA transfer complete interrupt enable bit 0: DMA transfer complete interrupt disabled 1: DMA transfer complete interrupt enabled
16	INTEN	CPU Interrupt Request Enable Control Bit 0: Disable interrupt request 1: Enable interrupt request
15	TPTYPE	DMA channel transfer type selection bit 0: Single Transfer 1: Burst Transfer In burst transfer mode, the DMA controller moves DMACNT units of data with a single request. After the channel responds, data is transferred in bursts sized by BURSIZE until DMACNT reaches zero, completing the burst transfer.
14 ~ 12	BURSIZE[2:0]	During burst transfer, the burst size can be selected based on the burst transfer mode definition as follows: 000:128 001:64 010:32 011:16 100:8 101:4 110:2 111:1

Bit Number	Bit Symbol	Description
11 ~ 10	SAINC[1:0]	DMA channel source address increment/decrement mode configuration bits 00: No increment (Fixed address mode) 01: Increment mode 10: Decrement mode 11: Incremental loop mode (refer to the DMA source address buffer register) SAINC[1:0] can be freely modified and takes effect immediately when the channel is disabled; when enabled, changes take effect upon loop mode reload
9 ~ 8	DAINC[1:0]	DMA channel destination address increment/decrement mode configuration bits 00: No increment (Fixed address mode) 01: Increment mode 10: Decrement mode 11: Incremental loop mode (refer to the DMA destination address buffer register) DAINC[1:0] can be freely modified and takes effect immediately when the channel is disabled; when enabled, changes take effect upon loop mode reload
7	CHEN	DMA channel enable bit 0: DMA channel disabled 1: DMA channel enabled
6	CHRST	DMA channel reset control bit, used to reset the DMA channel. 0: None 1: Reset the current DMA channel. When set, CHEN of the current DMA channel is disabled, the interrupt flag is cleared, and other registers retain their values.
5	PAUSE	DMA channel transfer pause control bit 0: None 1: Pause the current DMA channel. When set, CHEN of the current DMA channel is disabled. The state machine returns to state=1 after completing the current read/write cycle, and internal registers retain their values (source/destination address, counter). To resume the transfer after pause, set CHEN=1 and PAUSE=0 simultaneously.
4	CIRC	DMA Channel Loop Mode Enable Bit 0: Channel not operating in loop mode. When the configured data count is reached, DMACNT remains zero for this channel 1: Channel operating in loop mode. Upon transfer completion, DMACNT automatically reloads the previously set value. Loop Mode can be used to handle circular buffers and continuous data streams (such as ADC scan mode). During transmission in Loop Mode, the number of data items to be transferred is automatically reloaded to the initial value set during channel configuration, and DMA requests continue to be serviced. To stop loop transmission, software must first stop the peripheral from generating DMA requests (for example, exit ADC scan mode) before disabling the DMA channel. Software must explicitly set the DMACNT value both before starting/enabling transmission and after stopping loop transmission.
3 ~ 2	TXWIDTH[1:0]	DMA Channel Transfer Width Selection Bit Select the data width for each transfer of the current DMA channel's source and destination addresses: 00: 8bit 01: 16bit 10: 32bit 11: 32bit TXWIDTH[1:0] can be modified at any time. Changes take effect immediately when the channel is disabled

Bit Number	Bit Symbol	Description
		When the channel is enabled, changes take effect upon reload during loop mode.
1 ~ 0	PL[1:0]	DMA Channel Priority Setting Bit If a DMA channel is active and other channels have pending requests, priority arbitration is triggered once the active channel completes. 00: Low 01: Medium 10: High 11: Very High Note: When priorities are equal, the channel with the smaller channel number has higher priority.
31 ~ 30 22 ~ 20	-	Reserved

24.7.1.4 DMA channel n counter buffer register DMA_n_CNT

Register	Read/Write	Description	Reset Value
DMA _n _CNT n = 0 ~ 3	R/W	DMA channel n counter buffer register	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 Symbol	Description
31 ~ 0	DMACNT[31:0]	DMA channel counter buffer register ● Write: ■ The value of DMACNT equals the remaining transfer count of the current DMA channel. ■ Each DMA channel contains a working counter that decrements by TXWIDTH units after each transfer. ◆ When CIRC=0 (DMA channel in non-loop mode), the working counter stops accepting DMA requests once it reaches 0. ◆ When CIRC=1 (DMA channel in loop mode), the working counter reloads DMACNT upon reaching 0 and waits for the next cycle. ● Read: ■ When the channel is disabled, reading returns the DMACNT value ■ When the channel is enabled, reading returns the internal working counter's real-time data.

24.7.1.5 DMA channel n status register DMA_n_STS

Register	Read/Write	Description	Reset Value
DMA _n _STS n = 0 ~ 3	R/W	DMA channel n status 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
-	-	-	-	-	-	-	SWREQ
7	6	5	4	3	2	1	0
STATUS[3:0]				TEIF	HTIF	TCIF	GIF

Bit Number	Bit Symbol	Description
8	SWREQ	DMA channel software request trigger bit Writing 1 to this bit keeps the DMA channel's software request pending until the channel responds; the bit is then automatically cleared by hardware.
7 ~ 4	STATUS[3:0]	DMA channel status bits 0000: Idle. 0001: Writing source address 0010: Reading source data and writing to destination 0011: Writing destination data 0100: Reserved 0101: Suspend pending (channel busy; other channels' requests suspended) 0110: Pause waiting state (after writing PAUSE=1 in burst transfer mode) 0111: Burst transfer in progress 1000: Burst transfer stopped: triggered by PAUSE enabled, DMACNT reaching 0, or bursize reaching 0
3	TEIF	DMA transfer error interrupt flag TEIF is set by hardware when DMA accesses an undefined address. This bit is cleared by writing 1.
2	HTIF	DMA half-transfer interrupt flag HTIF is set by hardware when DMACNT reaches half its initial value. This bit is cleared by writing 1.
1	TCIF	DMA transfer complete interrupt flag TCIF is set by hardware when DMACNT reaches 0. This bit is cleared by writing 1.
0	GIF	DMA channel global interrupt flag 0: No interrupt generated on the current DMA channel 1: Interrupt generated on the current DMA channel: transfer error, half-transfer, or transfer complete
31 ~ 9	-	Reserved

24.7.2 DMA register mapping

Register	Offset Address	Read/Write	Description	Reset Value
DMA0 Base Address: 0x4001_0800				
DMA0_SADR	0x00	R/W	DMA Source Address Buffer Register	0x0000_0000
DMA0_DADR	0x04	R/W	DMA Destination Address Buffer Register	0x0000_0000

Register	Offset Address	Read/Write	Description	Reset Value
DMA0_CFG	0x08	R/W	DMA Control/Configuration Register	0x0000_0000
DMA0_CNT	0x0C	R/W	DMA Counter Register	0x0000_0000
DMA0_STS	0x10	R/W	DMA Status Register	0x0000_0000
DMA1 Base Address: 0x4001_0840				
DMA1_SADR	0x00	R/W	DMA Source Address Buffer Register	0x0000_0000
DMA1_DADR	0x04	R/W	DMA Destination Address Buffer Register	0x0000_0000
DMA1_CFG	0x08	R/W	DMA Control/Configuration Register	0x0000_0000
DMA1_CNT	0x0C	R/W	DMA Counter Register	0x0000_0000
DMA1_STS	0x10	R/W	DMA Status Register	0x0000_0000
DMA2 Base Address: 0x4001_0880				
DMA2_SADR	0x00	R/W	DMA Source Address Buffer Register	0x0000_0000
DMA2_DADR	0x04	R/W	DMA Destination Address Buffer Register	0x0000_0000
DMA2_CFG	0x08	R/W	DMA Control/Configuration Register	0x0000_0000
DMA2_CNT	0x0C	R/W	DMA Counter Register	0x0000_0000
DMA2_STS	0x10	R/W	DMA Status Register	0x0000_0000
DMA3 Base Address: 0x4001_08C0				
DMA3_SADR	0x00	R/W	DMA Source Address Buffer Register	0x0000_0000
DMA3_DADR	0x04	R/W	DMA Destination Address Buffer Register	0x0000_0000
DMA3_CFG	0x08	R/W	DMA Control/Configuration Register	0x0000_0000
DMA3_CNT	0x0C	R/W	DMA Counter Register	0x0000_0000
DMA3_STS	0x10	R/W	DMA Status Register	0x0000_0000

25 SysTick

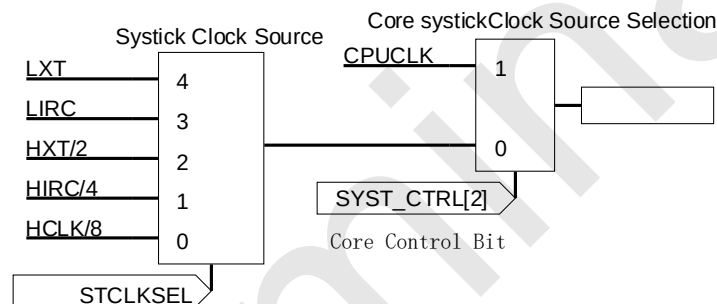
SysTick is a simple, 24-bit write-to-clear, decrementing, auto-reload counter with a flexible control mechanism. This counter can be used as the tick timer for a real-time operating system (RTOS) or as a simple counter.

25.1 Clock source

The clock sources of SysTick (Cortex®-M0+ Core SysTick) are divided into internal and external clock sources:

- Internal clock source: the CPU clock
- 5 external clock sources

The block diagram of the SysTick clock sources is as follows:



25.2 SysTick calibration register default value

The method for setting the calibration value of the SysTick Calibration Register is as follows:

- If the power-on default clock frequency is f_{HCLK}/n (MHz), where n is the default division factor at power-on and the default Clock Source is HIRC,
- then the initial SysTick calibration value is $1000 \times (f_{HCLK}/n)$, ensuring a default 1 ms time base.

26 Revision history

Version	Record	Date
V0.1	First edition	June 26, 2026

Preliminary

27 Important notice

Shenzhen SinOne Microelectronics Co., Ltd. (hereinafter SinOne) reserves the right to change, correct, enhance, modify, and improve SinOne products, documents, or services at any time without prior notice. SinOne believes the information provided is accurate and reliable. This document's information is valid from June 2026. For actual production design, please refer to the latest datasheets and other relevant materials for each product.

Preliminary