

Overview

The CH596 is a RISC-V MCU microcontroller with integrated BLE wireless communication. It features an on-chip 2Mbps Bluetooth Low Energy BLE communication module, along with SPI, 2 serial ports, 8-channel ADC, touch button detection module, and other rich peripheral resources. The CH596 supports single-cell 1.5V battery power supply.

Feature

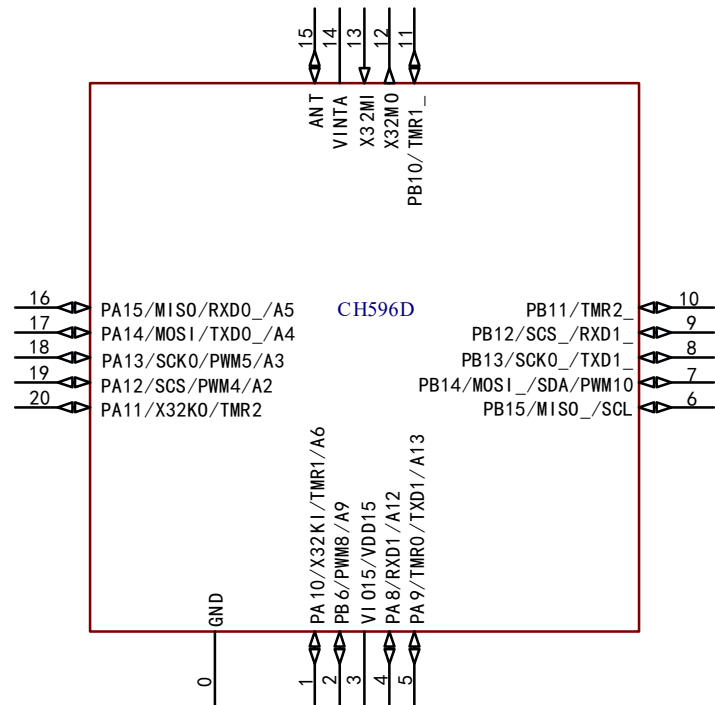
- **Core:**
 - QingKe 32-bit RISC-V3C core
 - Support RV32IMBC-X instruction set
 - Fast programmable interrupt controller + hardware interrupt stack
 - Branch prediction, conflict handling mechanism
 - Single-cycle multiplication, hardware division
 - Multi-speed system main frequency, minimum 32KHz
- **Non-volatile memory FlashROM:**
 - 240KB user application program memory area CodeFlash
 - 16KB system boot program memory area BootLoader
 - 256B user-defined information storage area
 - 256B system non-volatile configuration information memory area InfoFlash
 - Support ICP, ISP and IAP, support OTA wireless upgrade
 - Basically, zero-wait at the system frequency of 80MHz
- **32K-byte volatile data storage SRAM:**
 - Sleep retention memory area RAM24K using 24KB dual power supply
 - Sleep retention memory area RAM8K using 8KB dual power supply
- **Power management and low-power:**
 - Support 1.2V~1.8V power supply
 - Idle mode: 1.5mA
 - Halt mode: 1.02mA (PLL/HSE not stopped)
180uA (PLL/HSE stopped)
- Sleep mode: Multiple gears from 1.8uA to 3.8uA
- Shutdown mode: Multiple gears from 0.85uA to 2.6uA
- Low power consumption battery voltage low voltage monitoring
- **Security properties:**
 - AES-128 encryption and decryption, unique chip ID
- **Bluetooth Low Energy (BLE):**
 - Integrated with 2.4GHz RF transceiver and baseband and link control
 - -95dBm RX sensitivity, programmable +4dBm TX power
 - BLE complies with Bluetooth Low Energy Specification 5.4
 - Support 2Mbps, 1Mbps
 - Provide optimized protocol stack and application layer API, and support networking
- **Analog-to-digital conversion ADC:**
 - 12-bit ADC with single-ended input
 - 8 external analog signal channels and 1 internal signal.
- **Real-time clock (RTC):**
 - 2 modes of timing and triggering
- **TouchKey detection module:**
 - 8-channel
- **Timer and pulse width modulation (PWM):**
 - 3 sets of 26-bit timers, which can reach 4.2S at the frequency of 16MHz
 - 3-channel capture/sample, support rising edge/falling edge /double edge

- 3-channel 26-bit PWM output
- 4-channel 8-bit PWM output, the first 3 channels support 16-bit PWM output
- **Encoder ENC:** 2-channel interface for capturing rotary encoder signals for encoding
- **Universal asynchronous receiver/transmitter (UART):**
 - 2 independent UARTs, built-in 8-level FIFO
 - 23-bit counter, up to 6Mbps communication baud rate
 - UART0 supports some Modem
 - UART0 supports automatic matching of slave address during multi-device communication
- **Serial peripheral interface (SPI):**
 - Support Master and Slave modes
 - Built-in FIFO, Support DMA
 - SCK serial clock frequency can reach half of system clock frequency
- **LED dot matrix screen interface: supports 1/2/4 data lines**
- **2-wire serial interface (I2C):**
 - Support Master and Slave modes, compatible with SMBus
 - Support 7-bit or 10-bit address and bus broadcast
 - Support arbitration, error detection, PEC check and clock extension
- **Clock:** Built-in PLL, built-in 16MHz and 32KHz clock
- **General-purpose input/output (GPIO):**
 - 15 GPIOs, 4 of them support 5V signal input
 - Optional pull-up or pull-down resistor, optional output drive capability
 - 15 GPIOs support level or edge interrupt, wake-up input
- **Package:** QFN

Part No	CodeFlash +BootLoader	DataFlash RAM	RTC	Timer	Capture	PWM	UART	SPI	I2C	BLE	ADC	Capacitive Touch-key	Min. supply voltage	GPIO	Package
CH596D	240+16K	24+8K	√	3	3	3+4	2	√	√	√	8	8-channel	1.2V	15	QFN20

Chapter 1 Pinouts and Pin Definitions

1.1 Pinouts



1.2 Pin Definitions

Note that the pin function descriptions in the table below are for all functions and do not relate to specific model products. Peripheral resources may vary between models, so please check the availability of this function according to the product model resource table before viewing.

Pin No.	Pin name	Pin type ⁽¹⁾	Alternate function takes precedence ⁽²⁾	Function Description
CH596D				
0	GND	P	-	Chip EPAD, common ground, voltage 0V reference point.
1	PA10	I/O/A	X32KI/TMR1 /ENCCAP_IN1_ /AIN6	PA10: General-purpose bidirectional digital I/O pin. X32KI: Input for the low-frequency oscillator, connected to the other end of an external 32 kHz crystal. TMR1: Capture input 1 for Timer 1 and PWM output channel 1. ENCCAP_IN1_: Encoder capture channel 1 mapping. AIN6: ADC analog signal input channel 6.
2	PB6	I/O/A	PWM8 /AIN9	PB6: General-purpose bidirectional digital I/O pin. PWM8: PWM output channel 8. AIN9: ADC analog signal input channel 9.
3	VDD15	P	VBAT	Battery power input, close to the pin external 2.2uF parallel 0.1uF decoupling capacitor.
	VIO15	P	-	I/O power input.
4	PA8	I/O/A	RXD1 /AIN12	PA8: General-purpose bidirectional digital I/O pin. RXD1: UART1 serial data input. AIN12: ADC analog signal input channel 12.
5	PA9	I/O/A	TMR0 /TXD1 /AIN13	PA9: General-purpose bidirectional digital I/O pin. TMR0: Capture input 0 of Timer 0 and PWM output channel 0. TXD1: UART1 serial data output channel. AIN13: ADC analog signal input channel 13.
6	PB15	I/O/5VT	MISO_ /SCL /DTR /LED_CLK_	PB15: General-purpose bidirectional digital I/O pin. MISO: SPI's MISO pin mapping. SCL: I2C serial clock pin, host output and input/slave input. DTR: UART0's MODEM output signal, data terminal ready. LED_CLK_: LED output clock pin mapping.
7	PB14	I/O/5VT	MOSI_ /SDA /PWM10 /LED[0] /DSR	PB14: General-purpose bidirectional digital I/O pin. MOSI_: MOSI pin mapping of SPI. SDA: SDA pin of I2C. PWM10: PWM output channel 10. LED[0]: LED output bit0 pin mapping. DSR: DSR pin of UART0.
8	PB13	I/O/5VT	SCK_ /TXD1_	PB13: General-purpose bidirectional digital I/O pin. SCK_: SCK pin mapping of SPI.

Pin No.	Pin name	Pin type ⁽¹⁾	Alternate function takes precedence ⁽²⁾	Function Description
CH596D				
			/LED[1] /DCD_	TXD1_: TXD1 pin mapping of UART1. LED[1]_: LED output bit1 pin mapping. DCD_: DCD pin mapping of UART0.
9	PB12	I/O/5VT	LED[2]_ /RXD1_ /SCS_ /RI_	PB12: General-purpose bidirectional digital I/O pin. LED[2]_: LED output bit2 pin mapping. RXD1_: RXD1 pin mapping of UART1. SCS_: SCS pin mapping of SPI. RI_: RI pin mapping of UART0.
10	PB11	I/O/A	LED[3]_ /ENCCAP_IN2 /TMR2_	PB11: General-purpose bidirectional digital I/O pin. LED[3]_: LED output bit3 pin mapping. ENCCAP_IN2: Encoder capture channel 2. TMR2_: TMR2, CAP2 pin mapping of Timer 2.
11	PB10	I/O/A	ENCCAP_IN1 /TMR1_	PB10: General-purpose bidirectional digital I/O pin. ENCCAP_IN1: Encoder capture channel 1. TMR1_: TMR1, CAP1 pin mapping of Timer 1.
12	X32MO	I/A	-	The inverting output terminal of the high-frequency oscillator HSE is externally connected with one end of a 32MHz crystal.
13	X32MI	A	-	The input end of the high frequency oscillator HSE is externally connected with the other end of the 32MHz crystal.
14	VINTA	P	-	The power supply node of the internal analog circuit needs to be close to the pin external decoupling capacitor. It is recommended to 1uF (support 0.1uF~2.2uF, large capacitance and slightly larger power consumption).
15	ANT	A	-	RF signal input and output, it is recommended to directly connect the antenna.
16	PA15	I/O/A	MISO /LED_CLK /RXD0_ /AIN5	PA15: General-purpose bidirectional digital I/O pin. MISO: MISO pin of SPI, host input/slave output. LED_CLK: LED output clock. RXD0_: RXD0 pin mapping of UART0. AIN5: ADC analog signal input channel 5.
17	PA14	I/O/A	MOSI /TXD0_ /LED[0] /AIN4	PA14: General-purpose bidirectional digital I/O pin. MOSI: MOSI pin of SPI, host input/slave output. TXD0_: TXD0 pin mapping of UART0. LED[0]: LED output bit0. AIN4: ADC analog signal input channel 4.
18	PA13	I/O/A	SCK /PWM5 /LED[1] /SWCLK	PA13: General-purpose bidirectional digital I/O pin. SCK: Serial clock pin of SPI, host input/slave output. PWM5: PWM output channel 5. LED[1]: LED output bit1.

Pin No.	Pin name	Pin type ⁽¹⁾	Alternate function takes precedence ⁽²⁾	Function Description
CH596D				
			/AIN3	SWCLK ⁽³⁾ : Serial clock input of simulation debugging interface. AIN3: ADC analog signal input channel 3.
19	PA12	I/O/A	SWDIO /PWM4 /LED[2] /SCS/AIN2	PA12: General-purpose bidirectional digital I/O pin. SWDIO ⁽³⁾ : Serial clock input/output of simulation debugging interface. PWM4: PWM output channel 4. LED[2]: LED output bit2. SCS: SCS chip selection input of SPI, active low. AIN2: ADC analog signal input channel 2.
20	PA11	I/O/A	TMR2 /LED[3] /ENCCAP_IN2_ /X32KO	PA11 ⁽⁴⁾ : General-purpose bidirectional digital I/O pin. TMR2: Capture input 2 of Timer 2 and PWM output channel 2. LED[3]: LED output bit3. ENCCAP_IN2_: Encoder capture channel 2 mapping. X32KO: The inverting output end of the low-frequency oscillator is externally connected with one end of the 32KHz crystal.

Note: 1. Pin Type: I=TTL/CMOS level Schmitt input; O=CMOS level 3-state output;

A= Analog signal input or output; 5VT= Support 5V signal voltage input.

- The alternate functions and mappings of pins are arranged in the table based on their priorities from high to low, and the GPIO function is with the lowest priority.
- 2-wire simulation debug interface is configured by ISP tool. After the simulation debug interface is enabled, PA13 and PA12 are only used as SWCLK and SWDIO, and will not be used as GPIO or peripheral alternate function pins. After the simulation debug interface is disabled, PA13 and PA12 can be used as GPIO and peripheral alternate function pins.
- Pull-up is not supported, and push-pull output is not supported in sleep mode.

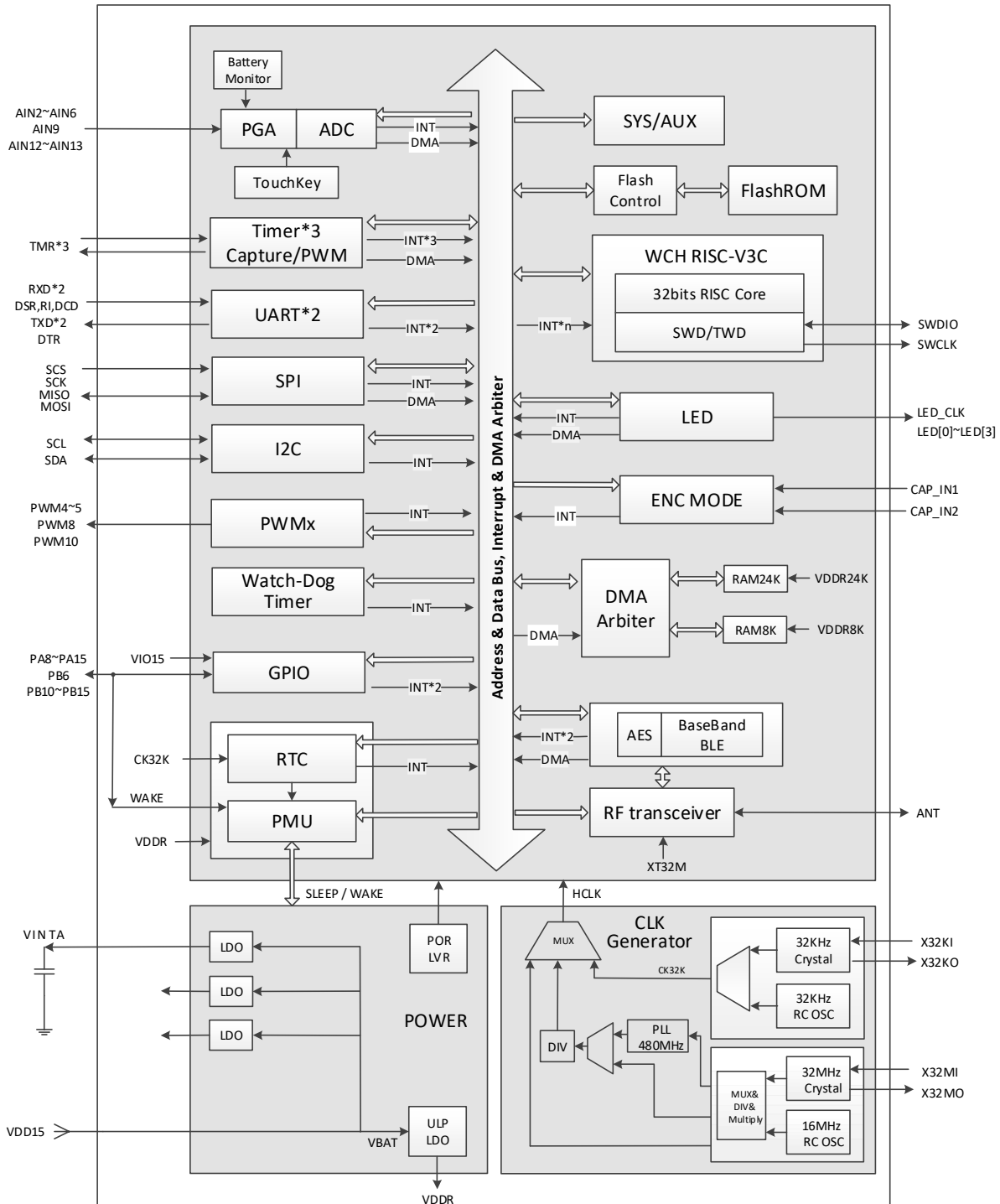
5.

Chapter 2 System Architecture and Memory

2.1 System Architecture

The following figure shows the system architecture block diagram. Its core is QingKe RISC-V microprocessor, please refer to QingKe Processor Manual for details.

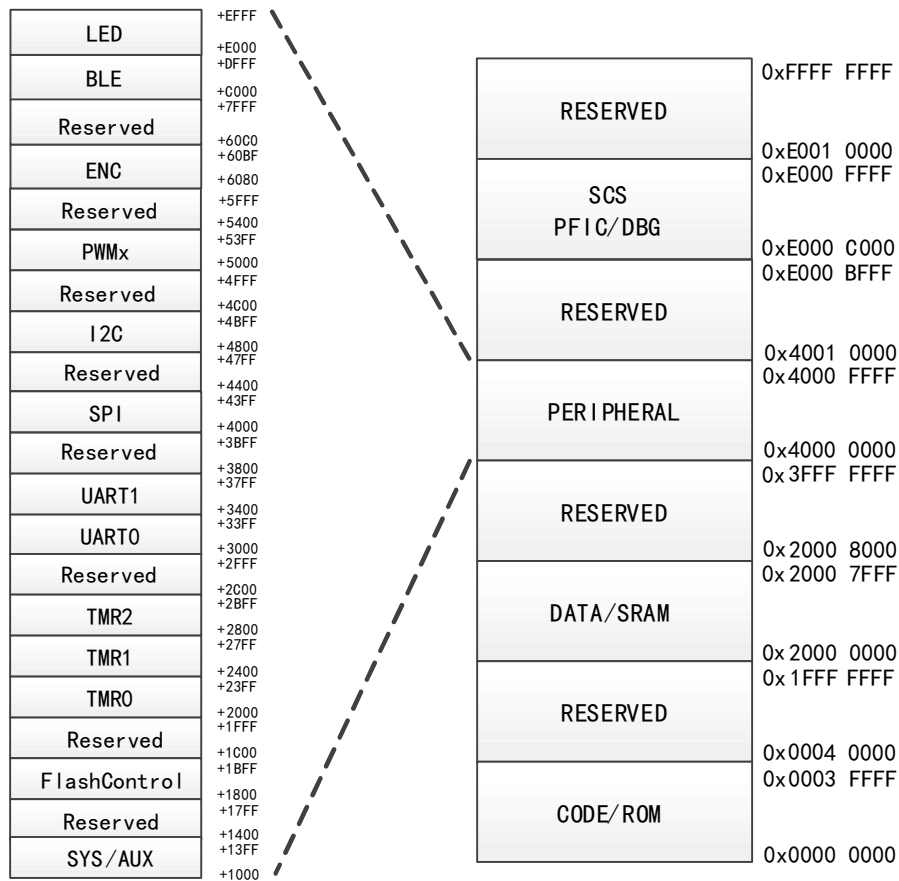
Figure 2-1 Internal structure block diagram



2.2 Memory Mapping

The addressing space mainly includes several different areas, CODE area/FlashROM, DATA area/SRAM and peripherals, as shown in the figure below.

Figure 2-2 Memory mapping diagram



2.3 Memory Mapping Table

The address range of each memory mapping area is shown in the table below:

Table 2-1 Memory mapping area address

Address range	Application	Description
0x00000000-0x0003FFFF	On-chip CODE area, non-volatile memory	256KB, FlashROM
0x00040000-0x1FFFFFFF	Reserved	-
0x20000000-0x20007FFF	On-chip DATA area, volatile memory	32KB, SRAM
0x20008000-0x3FFFFFFF	Reserved	-
0x40000000-0x4000FFFF	Various peripherals	Multiple peripheral modules
0x40010000-0xE000BFFF	Reserved	-
0xE000C000-0xE000FFFF	Various peripherals in system	System Control Space (SCS)
0xE0010000-0xFFFFFFFF	Reserved	-

2.3.1 On-chip CODE Area Mapping Table

Table 2-2 CODE area address

Address range	Application	Description
0x00000000-0x0003BFFF	User application program memory, CodeFlash	240KB
0x0003C000-0x0003FFFF	User non-volatile data memory, DataFlash	16KB
0x00040100-0x000401FF	System bootloader memory, BootLoader	256B
0x00040200-0x000402FF	System non-volatile configuration information memory, InfoFlash	256B

The configuration information of the addresses 0x00040100-0x000401FF can be set by the user through tools.

Note: The system bootloader storage area can also be used to store user applications. For specific operations, please refer to the EVT example program.

Table 2-3 Boot description

Chip model	Boot pin	Download interface	Enter condition
CH596D	PB15 low level	UART1_(PB12/PB13) Baud rate: 115200/1M/2M	1. Serial port supports keyless operation (detects serial communication packets within 40ms of power-up). 2. Power-up detection verifies boot pin meets requirements. 3. Power-up determines user area 16 bytes [0-15] are in erased state.

Note: Upon entering Boot mode via any method, the device will exit Boot mode if no data communication occurs within 10 seconds. After a successful handshake, the device will exit Boot mode if no data communication occurs within 60 seconds.

Table 2-4 Description of user non-volatile configuration information

Address	Bit address	Name	Application	Default value
0x40100	Bit 4	RESERVED	Reserved	0
	Bit 5	CFG_IWDG_EN	Independent watchdog enable: 1: Enable independent watchdog; 0: Disable independent watchdog.	0
	Bit 6	CFG_BOOT_EN	BootLoader enable	1
	Bit 15~Bit 7	RESERVED	Reserved	0
	Bit 23~Bit 16	CFG_ROM_READ	Code and data protection modes in FlashROM: Value 0x3A: Read access permitted; Non-0x3A value: 2-wire debug read access prohibited; program confidentiality maintained.	0x3A
	Bit 31~Bit 24	RESERVED	Reserved	0
0x4010C	Bit 31~Bit 0	VALID_SIG	Configuration information valid flag, fixed value	0x235654AD

Note: To disable FlashROM code and data protection mode: Perform a full erase of the user area and write the configuration information CFG_ROM_READ (bits [23:16]) to 0x3A.

2.3.2 On-chip DATA Area Mapping Table

Table 2-5 DATA area address

Address range	Application	Description
0x20000000-0x20005FFF	Independently maintainable memory area supplied by the main + auxiliary dual power, RAM24K	24KB
0x20006000-0x20007FFF	Independently maintainable memory area supplied by the main + auxiliary dual power, RAM8K	8KB

2.3.3 Peripheral Address Assignment

The chip mainly contains the following peripherals. Each peripheral occupies a certain address space, and the actual access address of peripheral register is: base address + offset address. In the following chapters, the address of the register is described in detail. The following table shows the assignment of base address of each peripheral.

Table 2-6 Peripheral base address assignment table

Peripheral No.	Peripheral name	Peripheral base address
1	SYS(PMU/RTC/GPIO, etc.) AUX(ADC/TKEY/PLL, etc.)	0x4000 1000
2	FlashROM-Control	0x4000 1800
3	TMR0	0x4000 2000
4	TMR1	0x4000 2400
5	TMR2	0x4000 2800
6	UART0	0x4000 3000
7	UART1	0x4000 3400
8	SPI	0x4000 4000
9	I2C	0x4000 4800
10	PWMx (PWM8, PWM10, PWM4~PWM5)	0x4000 5000
11	ENC	0x4000 6080
12	Radio: BLE	0x4000 C000 0x4000 D000
13	LED	0x4000 E000

The following table shows the explanation of "Access" in the register description in the subsequent chapters:

Table 2-7 Description of access attributes

Abbreviation	Description
RF	Read only and the read value is fixed and not affected by reset.
RO	Read-only
WO	Write only, read value is 0 or invalid.
RZ	Read only, clear 0 automatically after reading.
WZ	Write then clear 0.

RW	Readable and writeable.
RW1	Readable, clear 0 if write 1.
WA	Write only and only in safe mode, read value is 0 or invalid.
RWA	Readable, write only in safe mode.

The following table explains the abbreviations used in subsequent chapters:

Table 2-8 Description of Abbreviations

Abbreviation	Description
HSE	External high-frequency crystal oscillator clock source (32MHz recommended)
HSI	Internal high-frequency RC clock oscillator (calibrated to 16MHz during application software operation)
LSE	External low-frequency crystal oscillator clock source (32KHz recommended)
LSI	Internal low-frequency RC clock oscillator (calibrated to 16MHz during application software operation)
CK32M	High-frequency clock source (default 32MHz)
T _{CK32M}	High-frequency clock cycle (1/CK32M)
CK16M	High-frequency clock source (default 16MHz)
T _{CK16M}	High-frequency clock cycle (1/CK16M)
CK32K	Low-frequency clock source (default 32 kHz)
Fpll	PLL output clock (default frequency: 480MHz)
HCLK	System clock
Fsys	System clock frequency
Tsys	System clock cycle (1/Fsys)
RAM8K	8KB SRAM of high address
RAM24K	24KB SRAM of low address
0x	The data starting with it indicates a hexadecimal number
h	The data ending with it indicates a hexadecimal number
b	The data ending with it indicates a binary number

Chapter 3 Interrupt

The system has a built-in Programmable Fast Interrupt Controller (PFIC), which supports up to 255 interrupt vectors. The current system manages 18 peripheral interrupt channels and 7 core interrupt channels, and other interrupt sources are reserved.

3.1 Interrupt Controller

18 peripheral interrupts; each interrupt request has an independent trigger and maskable control bit, as well as a dedicated status bit.

1 non-maskable interrupt NMI.

Unique fast interrupt entry and exit mechanism, hardware automatic stacking and recovery, without instruction overhead.

Unique fast interrupt response mechanism, 4 channels programmable directly access interrupt vector addresses.

3.2 System Timer (SysTick)

The core provides a 32-bit counter (SysTick), supports HCLK or HCLK/8 as the time base, with higher priority.

3.3 Interrupt and Exception Vector

The following is the vector table.

Table 3-1 Interrupt vector table

No.	Priority	Priority Type	Name	Description	Address
0	-3	Fixed	Reset	Reset	0x0000_0000
1	-	-	-	Reserved	0x0000_0004
2	-2	Fixed	NMI	Non-maskable interrupt	0x0000_0008
3	-1	Fixed	EXC	Failures and exception interrupts of all types	0x0000_000C
4	-	-	-	Reserved	-
5	-1	Fixed	ECALL-M	Machine mode callback interrupt	0x0000_0014
6-7	-	-	-	Reserved	-
8	-1	Fixed	ECALL-U	User mode callback interrupt	0x0000_0020
9	-1	Fixed	BREAKPOINT	Breakpoint callback interrupt	0x0000_0024
10-11	-	-	-	Reserved	-
12	0	Settable	SysTick	SysTick timer	0x0000_0030
13	-	-	-	Reserved	-
14	1	Settable	SWI	Software interrupt	0x0000_0038
15	-	-	-	Reserved	0x0000_003C
16	2	Settable	TMR0	TMR0 interrupt	0x0000_0040
17	3	Settable	GPIO_A	GPIO port PA interrupt	0x0000_0044
18	4	Settable	GPIO_B	GPIO port PB interrupt	0x0000_0048

19	5	Settable	SPI0	SPI0 interrupt	0x0000_004C
20	6	Settable	BLEB	BB interrupt of wireless module	0x0000_0050
21	7	Settable	BLEL	LLE interrupt of wireless module	0x0000_0054
22	-	-	-	Reserved	0x0000_0058
23	-	-	-	Reserved	0x0000_005C
24	8	Settable	TMR1	TMR1 interrupt	0x0000_0060
25	9	Settable	TMR2	TMR2 interrupt	0x0000_0064
26	10	Settable	UART0	UART0 interrupt	0x0000_0068
27	11	Settable	UART1	UART1 interrupt	0x0000_006C
28	12	Settable	RTC	RTC and oscillator calibration interrupts	0x0000_0070
29	13	Settable	ADC	ADC and TouchKey interrupts	0x0000_0074
30	14	Settable	I2C	I2C interrupt	0x0000_0078
31	15	Settable	PWMX	PWMX interrupt	0x0000_007C
32	-	-	-	Reserved	0x0000_0080
33	-	-	-	Reserved	0x0000_0084
34	-	-	-	Reserved	0x0000_0088
35	16	Settable	WDOG_BAT	Watchdog timer interrupt/battery low voltage interrupt	0x0000_008C
36	17	Settable	ENCODER	Encoder interrupt	0x0000_0090
37	-	-	-	Reserved	0x0000_0094
38	18	Settable	LED	LED interrupt	0x0000_0098

3.4 Registers

3.4.1 PFIC Register Description

PFIC register base address: 0xE000E000

Table 3-2 PFIC registers

Name	Offset address	Description	Reset Value
R32_PFIC_ISR1	0x00	PFIC interrupt enable status register 1	0x0000000C
R32_PFIC_ISR2	0x04	PFIC interrupt enable status register 2	0x00000000
R32_PFIC_IPR1	0x20	PFIC interrupt pending status register 1	0x00000000
R32_PFIC_IPR2	0x24	PFIC interrupt pending status register 2	0x00000000
R32_PFIC_ITHRESDR	0x40	PFIC interrupt priority threshold configuration register	0x00000000
R32_PFIC_CFGR	0x48	PFIC interrupt configuration register	0x00000000
R32_PFIC_GISR	0x4C	PFIC interrupt global status register	0x00000000
R32_PFIC_IDCFGR	0x50	PFIC fast interrupt ID configuration register	0x00000000
R32_PFIC_FIADDR0	0x60	PFIC fast interrupt 0 address register	0x00000000
R32_PFIC_FIADDR1	0x64	PFIC fast interrupt 1 address register	0x00000000
R32_PFIC_FIADDR2	0x68	PFIC fast interrupt 2 address register	0x00000000
R32_PFIC_FIADDR3	0x6C	PFIC fast interrupt 3 address register	0x00000000
R32_PFIC_IENR1	0x100	PFIC interrupt enable register 1	0x00000000
R32_PFIC_IENR2	0x104	PFIC interrupt enable register 2	0x00000000

R32_PFIC_IRER1	0x180	PFIC interrupt reset enable register 1	0x00000000
R32_PFIC_IRER2	0x184	PFIC interrupt reset enable register 2	0x00000000
R32_PFIC_IPSR1	0x200	PFIC interrupt pending set register 1	0x00000000
R32_PFIC_IPSR2	0x204	PFIC interrupt pending set register 2	0x00000000
R32_PFIC_IPRR1	0x280	PFIC interrupt pending reset register 1	0x00000000
R32_PFIC_IPRR2	0x284	PFIC interrupt pending reset register 2	0x00000000
R32_PFIC_IACR1	0x300	PFIC interrupt activation status register 1	0x00000000
R32_PFIC_IACR2	0x304	PFIC interrupt activation status register 2	0x00000000
R32_PFIC_IPRIORx	0x400	PFIC interrupt priority configuration register	0x00000000
R32_PFIC_EENR	0xC80	PFIC event enable register	0xFFFFFFFF
R32_PFIC_EPR	0xC84	PFIC event pending register	0x00000000
R32_PFIC_EWUPR	0xC88	PFIC event wakeup register	0x00000000
R32_PFIC_SCTLR	0xD10	PFIC system control register	0x00000000

In user mode, global interrupt control is supported, please refer to the examples provided in evaluation board documentation.

Description about core interrupt control bit:

1. Reset, NMI, EXC, ECALL-M, ECALL-U and BREAKPOINT interrupts are always enabled by default.
2. NMI and EXC support interrupt suspend clear and set control (controlled by PFIC_IPSR1 and PFIC_IPRR1), but do not support interrupt enable set and clear control.
3. Reset, ECALL-M, ECALL-U and BREAKPOINT do not support interrupt suspend clear and set control, interrupt enable set and clear control.

PFIC Interrupt Enable Status Register 1 (PFIC_ISR1)

Bit	Name	Access	Description	Reset value
[31:12]	INTENSTA	RO	31# and below interrupts current enable status. 1: Enable the current number interrupt; 0: Disable the current number interrupt.	0
[11:0]	Reserved	RO	Reserved. Reset, NMI, EXC, ECALL and other interrupts bit, the same below.	32Ch

PFIC Interrupt Enable Status Register 2 (PFIC_ISR2)

Bit	Name	Access	Description	Reset value
[31:4]	Reserved	RO	Reserved	0
[3:0]	INTENSTA	RO	32# and above interrupts current enable status. 1: Enable the current number interrupt; 0: Disable the current number interrupt.	0

PFIC Interrupt Pending Status Register 1 (PFIC_IPR1)

Bit	Name	Access	Description	Reset value
[31:12]	PENDSTA	RO	31# and below interrupts current pending status. 1: Current number interrupt has been pending;	0

			0: Current number interrupt is not pending.	
[11:0]	Reserved	RO	Reserved	0

PFIC Interrupt Pending Status Register 2 (PFIC_IPR2)

Bit	Name	Access	Description	Reset value
[31:4]	Reserved	RO	Reserved	0
[3:0]	PENDSTA	RO	32# and above interrupts current pending status. 1: Current number interrupt has been pending; 0: Current number interrupt is not pending.	0

PFIC Interrupt Priority Threshold Configuration Register (PFIC_ITHRESDR)

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved	0
[7:0]	THRESHOLD	RW	Interrupt priority threshold setting value. If the interrupt priority value is lower than the current setting value, interrupt service will not be performed when suspended. When this register is 0, the threshold register function is invalid. [7:4]: Priority threshold. [3:0]: Reserved; 0 constantly; invalid if writing.	0

PFIC Interrupt Configuration Register (PFIC_CFGR)

Bit	Name	Access	Description	Reset value
[31:16]	KEYCODE	WO	For different target control bits, the corresponding security access identification data needs to be written synchronously for modification, and the read value is always 0: KEY1 = 0xFA05. KEY2 = 0xBCAF. KEY3 = 0xBEEF.	0
[15:8]	Reserved	RO	Reserved	0
7	RESETSYS	WO	System reset (Write into KEY3 synchronously). Cleared to 0 automatically. Valid when writing 1; invalid when writing 0. The function of this bit is the same as that of SYSRESET bit of PFIC_SCTLR register.	0
[6:0]	Reserved	RO	Reserved	0

PFIC Interrupt Global Status Register (PFIC_GISR)

Bit	Name	Access	Description	Reset value
[31:10]	Reserved	RO	Reserved	0

9	GPENDSTA	RO	Whether there is interrupt pending currently: 1: Yes; 0: No.	0
8	GACTSTA	RO	Whether the interrupt is executed currently: 1: Yes; 0: No.	0
[7:0]	NESTSTA	RO	Current interrupt nesting status, support 2-level nesting currently, [1:0] valid. 3: Level 2 interrupt in process; 1: Level 1 interrupt in process; 0: No interrupt occurs; Others: Impossible condition.	0

PFIC Fast Interrupt ID Configuration Register (PFIC_IDCFGR)

Bit	Name	Access	Description	Reset value
[31:24]	FIID3	RW	Configure interrupt number of fast interrupt 3.	0
[23:16]	FIID2	RW	Configure interrupt number of fast interrupt 2.	0
[15:8]	FIID1	RW	Configure interrupt number of fast interrupt 1.	0
[7:0]	FIID0	RW	Configure interrupt number of fast interrupt 0.	0

PFIC Fast Interrupt 0 Address Register (PFIC_FIADDR0)

Bit	Name	Access	Description	Reset value
[31:1]	ADDR0	RW	Fast interrupt 0 service program address bit[31:1], bit0 is 0.	0
0	FI0EN	RW	Fast interrupt 0 channel enable bit: 1: Enable fast interrupt 0 channel; 0: Disable.	0

PFIC Fast Interrupt 1 Address Register (PFIC_FIADDR1)

Bit	Name	Access	Description	Reset value
[31:1]	ADDR1	RW	Fast interrupt 1 service program address bit[31:1], bit0 is 0.	0
0	FI1EN	RW	Fast interrupt 1 channel enable bit: 1: Enable fast interrupt 1 channel; 0: Disable.	0

PFIC Fast Interrupt 2 Address Register (PFIC_FIADDR2)

Bit	Name	Access	Description	Reset value
[31:1]	ADDR2	RW	Fast interrupt 2 service program address bit[31:1], bit0 is 0.	0
0	FI2EN	RW	Fast interrupt 2 channel enable bit: 1: Enable fast interrupt 2 channel; 0: Disable.	0

PFIC Fast Interrupt 3 Address Register (PFIC_FIADDR3)

Bit	Name	Access	Description	Reset value
[31:1]	ADDR3	RW	Fast interrupt 3 service program address bit[31:1], bit0 is 0.	0

0	FI3EN	RW	Fast interrupt 3 channel enable bit: 1: Enable fast interrupt 3 channel; 0: Disable.	0
---	-------	----	---	---

PFIC Interrupt Enable Register 1 (PFIC_IENR1)

Bit	Name	Access	Description	Reset value
[31:12]	INTEN	WO	31# and below interrupts enable. 1: Enable the current number interrupt; 0: No effect.	0
[11:0]	Reserved	RO	Reserved	0

PFIC Interrupt Enable Register 2 (PFIC_IENR2)

Bit	Name	Access	Description	Reset value
[31:4]	Reserved	RO	Reserved	0
[3:0]	INTEN	WO	32# and above interrupts enable. 1: Enable the current number interrupt; 0: No effect.	0

PFIC Interrupt Reset Enable Register 1 (PFIC_IRER1)

Bit	Name	Access	Description	Reset value
[31:12]	INTRESET	WO	31# and below interrupts disable. 1: Disable the current number interrupt; 0: No effect.	0
[11:0]	Reserved	RO	Reserved	0

PFIC Interrupt Reset Enable Register 2 (PFIC_IRER2)

Bit	Name	Access	Description	Reset value
[31:4]	Reserved	RO	Reserved	0
[3:0]	INTRESET	WO	32# and above interrupts disable. 1: Disable the current number interrupt; 0: No effect.	0

PFIC Interrupt Pending Set Register 1 (PFIC_IPSR1)

Bit	Name	Access	Description	Reset value
[31:12]	PENDSET	WO	31# and below interrupts pending sett. 1: Current number interrupt is pending; 0: No effect.	0
[11:0]	Reserved	RO	Reserved	0

PFIC Interrupt Pending Set Register 2 (PFIC_IPSR2)

Bit	Name	Access	Description	Reset value
[31:4]	Reserved	RO	Reserved	0
[3:0]	PENDSET	WO	32# and above interrupts pending set. 1: Current number interrupt is pending;	0

			0: No effect.	
--	--	--	---------------	--

PFIC Interrupt Pending Reset Register 1 (PFIC_IPRR1)

Bit	Name	Access	Description	Reset value
[31:12]	PENDRESET	WO	31# and below interrupts pending reset. 1: Current number interrupt reset pending status; 0: No effect.	0
[11:0]	Reserved	RO	Reserved	0

PFIC Interrupt Pending Reset Register 2 (PFIC_IPRR2)

Bit	Name	Access	Description	Reset value
[31:4]	Reserved	RO	Reserved	0
[3:0]	PENDRESET	WO	32# and above interrupts pending reset. 1: Current number interrupt reset pending status; 0: No effect.	0

PFIC Interrupt Activation Status Register 1 (PFIC_IACR1)

Bit	Name	Access	Description	Reset value
[31:12]	IACTS	RW1	31# and below interrupts activation status. 1: Executing the current number interrupt; 0: The current number interrupt is not executed.	0
[11:0]	Reserved	RO	Reserved	0

PFIC Interrupt Activation Status Register 2 (PFIC_IACR2)

Bit	Name	Access	Description	Reset value
[31:4]	Reserved	RO	Reserved	0
[3:0]	IACTS	RW1	32# and above interrupts activation status. 1: Executing the current number interrupt; 0: The current number interrupt is not executed.	0

PFIC Interrupt Priority Configuration Register (PFIC_IPRIORx) (x=0-63)

The controller supports 256 interrupts (0-255), and 8 bits are used to set the control priority for each interrupt.

	31	24	23	16	15	8	7	0
IPRIOR63	PRIO_255		PRIO_254		PRIO_253		PRIO_252	
...	...		...		...		...	
IPRIORx	PRIO_(4x+3)		PRIO_(4x+2)		PRIO_(4x+1)		PRIO_(4x)	
...	...		...		...		...	
IPRIOR0	PRIO_3		PRIO_2		PRIO_1		PRIO_0	

Bit	Name	Access	Description	Reset value
-----	------	--------	-------------	-------------

[2047:2040]	IP_255	RW	Same as IP_0 description.	0
...	...	...	...	...
[31:24]	IP_3	RW	Same as IP_0 description.	0
[23:16]	IP_2	RW	Same as IP_0 description.	0
[15:8]	IP_1	RW	Same as IP_0 description.	0
[7:0]	IP_0	RW	Number 0 interrupt priority configuration: [7:5]: Priority control bit. [4:0]: Reserved. Always 0. Invalid if writing. The smaller priority value means higher priority. Only 2-level nested interrupts, i.e., it can be only preempted once.	0

PFIC Event Enable Register (PFIC_EENR)

Bit	Name	Access	Description	Reset value
[31:0]	EVENTEN	RW	31#-0# event wakeup enable: 1: Allow the corresponding event to wake the kernel from sleep; 0: Mask the corresponding event; do not allow the core to be woken when this event occurs. Supports up to 32 event sources, where 0-7 are level-triggered (active high) and 8-31 are pulse-triggered: 0#: Core suspend request; 1#: Core interrupt request; 7#-2#: Reserved; 8#: Core enter interrupt; 9#: Core exit interrupt; 10#: New interrupt suspension; (For shared interrupts, regardless of whether the interrupt is routed to the local core); 11#: Core send event; (Send event from the local core); 12#: Core receive event; (Send event from other cores); 13#: System receive event; (System-configured events, such as GPIO flip events...); 31#-14#: Reserved.	0xFFFFFFFF FF

PFIC Event Pending Register (PFIC_EPR)

Bit	Name	Access	Description	Reset value
[31:8]	EVENT_PEND	RW1Z	31#-8# event pending status, write 1 to clear: 1: The corresponding event occurred but was not processed; 0: The corresponding event did not occur.	0

[7:0]	EVENT_PEND	RO	7#-0# event pending status, cannot be cleared: 1: The corresponding event occurred but was not processed; 0: The corresponding event did not occur.	0
-------	------------	----	---	---

PFIC Event Wakeup Register (PFIC_EWUPR)

Bit	Name	Access	Description	Reset value
[31:0]	EVENT_WUP	RO	31#-0# event wakeup register: 1: The corresponding event wakes the core from sleep; 0: The corresponding event is not the event that woke the core this time.	0

PFIC System Control Register (PFIC_SCTLR)

Bit	Name	Access	Description	Reset value
31	SYSRESET	WO	System reset. Cleared to 0 automatically. Valid when writing 1, while invalid when writing 0. The same effect as PFIC_CFGR register.	0
[30:7]	Reserved	RO	Reserved	0
6	LOCKUP	RW	Deadlock enable. 1: Disable; 0: Enable.	0
5	SETEVENT	WO	Set event to wake up the WFE.	0
4	SEVONPEND	RW	When an event or interrupt suspending status occurs, the system can be woken up by the WFE command. If the WFE command is not executed, the system will be woken up immediately after the next execution of the command. 1: Enable events and all interrupts (including disabled interrupts) can wake up the system; 0: Only enabled events and enabled interrupts can wake up the system.	0
3	WFIOWFE	RW	The WFI command is executed as WFE. 1: The subsequent WFI command is deemed as WFE command; 0: No action.	0
2	SLEEPDEEP	RW	Low-power mode of control system: 1: deepsleep; 0: sleep.	0
1	SLEEPONEXIT	RW	The system status after controlled to exit the interrupt service program: 1: The system gets into low-power mode; 0: The system gets into the main program.	0

0	Reserved	RO	Reserved	0
---	----------	----	----------	---

3.4.2 CSR Registers Defined by WCH

In RISC-V, some control and status registers (CSR) are defined, which are used to configure, mark and record run status. CSR registers belong to internal registers of the core, and have a dedicated 12-bit address space. WCH devices not only provide standard registers defined in RISC-V architecture documentation, but also provide some registers defined by manufactures and the csr instruction is needed to access. Note: These registers with “MRW” attribute can only be accessed when the system is in machine mode.

Machine Trap-vector Base-address Register (MTVEC)

CSR address: 0x305

Bit	Name	Access	Description	Reset value
[31:2]	BASEADD	MRW	Interrupt vector table base address.	0
1	MODE1	MRW	Interrupt vector table recognize mode. 1: Recognize based on absolute address, support full range, but must jump; 0: Recognize based on jump instruction, limited range, support non-jump instruction.	0
0	MODE0	MRW	Interrupt or exception entry address mode select: 1: Offset address based on number*4; 0: Unified entry address.	0

User Mode Global Interrupt Enable Register (GINTENR)

This register controls the enable and disable of global interrupts. In machine mode, global interrupt enable and disable are managed via the MIE and MPIE bits in the mstatus register, which is inaccessible in user mode. The global interrupt enable register gintenr maps to the MIE and MPIE bits in mstatus. In user mode, gintenr can be manipulated to set or clear MIE and MPIE.

CSR address: 0x800

Bit	Name	Access	Description	Reset value
[31:13]	Reserved	URO	Reserved	0
[12:11]	MPP[1:0]	URO	Enter privileged mode before interrupt.	0
[10:8]	Reserved	URO	Reserved	0
7	MPIE	URW	When bit 5 of 0xBC0 (CSR) is enabled, this bit is readable and writable in user mode.	0
[6:4]	Reserved	URO	Reserved	0
3	MIE	URW	When bit 5 of 0xBC0 (CSR) is enabled, this bit is readable and writable in user mode.	0
[2:0]	Reserved	URO	Reserved	0

Interrupt System Control Register (INTSYSCR)

CSR address: 0x804

Bit	Name	Access	Description	Reset value
31	LOCK	URO	1: This register is readable and writable only in	0

			machine mode; 0: This register is readable and writable in user mode.	
[30:6]	Reserved	URO	Reserved	0x380
5	GIHWSTKNEN	URW1	Global interrupts and hardware stack push/pop disabled. <i>Note: This bit is commonly used in real-time operating systems. When setting this bit during interrupt context switching, it disables global interrupts and hardware stack operations. Upon completion of context switching and return from the interrupt, the hardware automatically clears this bit.</i>	0
4	Reserved	URO	Reserved	0
[3:2]	PMTCFG	RW	Preemption Priority Bit Width Configuration Register, used to configure the preemption priority bit width within the interrupt priority: 00: Preemption bit width is 0, interrupts of any priority cannot be nested; 01: Preemption width is 1 bit, meaning interrupt priority register bit [7] represents the preemption priority; 10: Preemption width is 2 bits, meaning interrupt priority register bits [7:6] represent the preemption priority; 11: Preemption width is 3 bits, meaning interrupt priority register bits [7:5] represent the preemption priority.	0x1
1	INESTEN	URO	Interrupt nesting function enable, fixed value set to 1: 1: Enable; 0: Disable. <i>Note: The actual nesting level is controlled by NEST_LVL in CSR 0xBC1.</i>	1
0	HWSTKEN	URW	Hardware stack enable: 1: Hardware stack function enabled; 0: Hardware stack function disabled.	0

Interrupt Nesting Control Register (INESTCR)

CSR address: 0xBC1

Bit	Name	Access	Description	Reset value
31	Reserved	MRO	Reserved	0
30	NEST_OV	MRW	Interrupt/Exception Nesting Overflow Flag, Write 1 to Clear:	0

			1: Interrupt overflow flag set; 0: Interrupt overflow not set. <i>Note: Interrupt overflow occurs only when an instruction exception is generated during execution of a secondary interrupt service function or during an NMI interrupt. In such cases, the exception and NMI interrupt proceed normally, but the CPU stack overflows, preventing exit from these exceptions or the NMI interrupt.</i>	
[29:12]	Reserved	MRO	Reserved	0
[11:8]	NEST_STA[3:0]	MRO	Nested status flag: 0000: No interrupt; 0001: Level 1 interrupt; 0011: Level 2 interrupt (Level 1 nested); 0111: Level 3 interrupt (overflow); 1111: Level 4 interrupt (overflow).	0
[7:2]	Reserved	MRO	Reserved	0
[1:0]	NEST_LVL[1:0]	MRW	Nesting Level: 00: Nesting disabled; nesting function turned off; 01: Level 1 nesting enabled; nesting function turned on; Others: Invalid. <i>Note: Writing 10 or 11 to this field sets it to 01. When 11 is written to this bit, reading the register returns the chip's maximum nesting level.</i>	0

3.4.3 Physical Memory Protection (PMP)

In order to improve system security, a set of physical address access privileges are defined in the RISC-V architecture, which can be used to set read/write/execute attribute of physical memory in the space and support regions as small as 4 bytes. PMP unit is always effective in user mode, and is optionally effective in machine mode. If the current memory restriction is violated, it may cause system exception interrupt (EXC).

PMP unit contains 4 sets of 8-bit configuration registers (in total 32 bits) and 4 sets of address registers, which need csr instruction to access and the registers must be in machine mode.

PMP Configuration Register (PMPCFG0)

CSR address: 0x3A0

Bit	Name	Access	Description	Reset value
[31:24]	pmp3cfg	MRW	See pmp0cfg.	0
[23:16]	pmp2cfg	MRW	See pmp0cfg.	0
[15:8]	pmp1cfg	MRW	See pmp0cfg.	0

[7:0]	pmp0cfg	MRW	Bit	Name	Description	0
			7	L	Lock enabled. Disable lock in machine mode: 1: Lock related register; 0: Not lock.	
			[6:5]	-	Reserved	
			[4:3]	A	Address alignment and protection region range selection.	
			2	X	Execute.	
			1	W	Write.	
			0	R	Read.	

Address alignment and protection region range select, memory protection for the region, $A_ADDR \leq \text{region} < B_ADDR$ (A_ADDR and B_ADDR are required to be 4 bytes aligned):

1. If $B_ADDR - A_ADDR = 2^2$, select NA4;
2. If $B_ADDR - A_ADDR = 2^{(G+2)}$, $G \geq 1$, and A_ADDR is $2^{(G+2)}$ aligned, select NAPOT;
3. Otherwise select TOR.

A	Name	Description
0	OFF	No region to be protected
1	TOR	Top alignment region protection: pmp0cfg, $0 \leq \text{region} < \text{pmpaddr0}$; pmp1cfg, $\text{pmpaddr0} \leq \text{region} < \text{pmpaddr1}$; pmp2cfg, $\text{pmpaddr1} \leq \text{region} < \text{pmpaddr2}$; pmp3cfg, $\text{pmpaddr2} \leq \text{region} < \text{pmpaddr3}$. $\text{pmpaddr}_{i-1} = A_ADDR \gg 2$; $\text{pmpaddr}_i = B_ADDR \gg 2$.
2	NA4	Fixed 4-byte region protection. pmp0cfg~pmp3cfg correspond to pmpaddr0~pmpaddr3 as start address. $\text{pmpaddr}_i = A_ADDR \gg 2$.
3	NAPOT	$2^{(G+2)}$ region protection, $G \geq 1$, and A_ADDR is $2^{(G+2)}$ aligned. $\text{pmpaddr}_i = (A_ADDR \gg 2) (2^{(G-1)} - 1)$.

PMP Address 0 Register (PMPADDR0)

CSR address: 0x3B0

Bit	Name	Access	Description	Reset value
[31:0]	ADDR0	MRW	Bit[33:2] of PMP address 0. Actually, the higher 2 bits are not used.	0

PMP Address 1 Register (PMPADDR1)

CSR address: 0x3B1

Bit	Name	Access	Description	Reset value
[31:0]	ADDR1	MRW	Bit[33:2] of PMP set address 1. Actually, the higher 2 bits are not used.	0

PMP Address 2 Register (PMPADDR2)

CSR address: 0x3B2

Bit	Name	Access	Description	Reset value
[31:0]	ADDR2	MRW	Bit[33:2] of PMP set address 2. Actually, the higher 2 bits are not used.	0

PMP Address 3 Register (PMPADDR3)

CSR address: 0x3B3

Bit	Name	Access	Description	Reset value
[31:0]	ADDR3	MRW	Bit[33:2] of PMP set address 3. Actually, the higher 2 bits are not used.	0

3.4.4 SysTick Register Description

STK register base address: 0xE000F000

Table 3-3 STK registers

Name	Offset address	Description	Reset value
R32_STK_CTRL	0x00	System count control register	0x00000000
R32_STK_SR	0x04	System count status register	0x00000000
R32_STK_CNTL	0xE000F008	System counter low register	0xFFFFFFFF
R32_STK_CMPLR	0xE000F010	Count compare low register	0xFFFFFFFF

System Count Control Register (STK_CTRL)

Bit	Name	Access	Description	Reset value
31	SWIE	RW	Software interrupt trigger enable (SWI): 1: Trigger software interrupt; 0: Disable trigger. This bit must be cleared after going into software interrupt, otherwise trigger will be always enabled.	0
[30:6]	Reserved	RO	Reserved	0
5	INIT	W1	Counter initial value update: 1: Upcount from 0, downcount from comparison value; 0: No effect.	0
4	MODE	RW	Count mode: 1: Downcount; 0: Upcount.	0
3	STRE	RW	Auto reload count enable: 1: Count from 0 after upcounting to comparison value, downcount from comparison value after downcounting to 0; 0: Continue upcounting/downcounting.	0
2	STCLK	RW	Counter clock source select: 1: HCLK as timebase; 0: HCLK/8 as timebase.	0
1	STIE	RW	Counter interrupt enable control bit:	0

			1: Enable counter interrupt; 0: No counter interrupt.	
0	STE	RW	System counter enable control bit: 1: Enable system counter STK; 0: Disable system counter STK; the counter stops counting.	0

System Count Status Register (STK_SR)

Bit	Name	Access	Description	Reset value
31	SWIE	RW	Software interrupt trigger enable: 1: Enable software interrupt triggering; 0: Disable triggering. After entering a software interrupt, this bit must be cleared to 0; otherwise, triggering will persist indefinitely.	0
[30:1]	Reserved	RO	Reserved	0
0	CNTIF	RW0	Count value compare flag, cleared when writing 0, while invalid when writing 1: 1: Upcount to comparison value, downcount to 0; 0: Not reach the comparison value.	0

System Counter Low Register (STK_CNTL)

Bit	Name	Access	Description	Reset value
[31:0]	CNT[31:0]	RW	Lower 32 bits of STK counter count value.	X

Count Compare Low Register (STK_CMPLR)

Bit	Name	Access	Description	Reset value
[31:0]	CMP[31:0]	RW	Set reload counter value higher 32 bits.	X

Chapter 4 System Control

4.1 Reset Control

The system supports 5 types of resets, including Real Power on Reset (RPOR), internal Software Reset (SR), Watch-dog Reset, Global Reset by Waking under Shutdown Mode (GRWSM) and Local Reset by Waking (LRW).

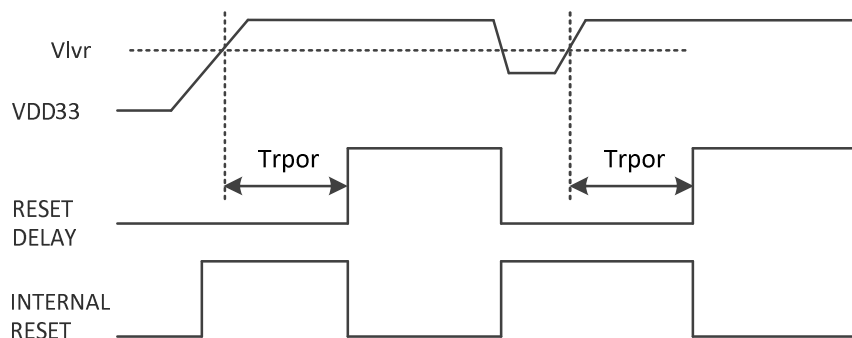
The R8_GLOB_RESET_KEEP register and the RB_ROM_CODE_OFS register are reset only when RPOR or GRWSM occurs, and are not affected by other types of resets.

Please refer to the timing parameter table in Section 20.5 for the timing parameters and reset property parameters in the figure below.

4.1.1 Real Power on Reset (RPOR)

When the power is on, the POR module inside the chip will generate a power-on reset and delay to wait for the power to stabilize. In addition, during operation, when the power voltage is lower than V_{lvr} , the internal LVR module of the chip will generate a low voltage reset until the voltage rises, and delay to wait for the power to stabilize. The figure below shows the power-on reset process and the low-voltage reset process.

Figure 4-1 Real power on reset



4.1.2 Internal Software Reset (SR)

Internal software reset is automatically carried out without external intervention. Set the bit RB_SOFTWARE_RESET of global reset configuration register (R8_RST_WDOG_CTRL) to 1, to realize software reset. This bit will be automatically cleared to 0.

4.1.3 Watchdog Reset (WTR/IWDG)

Watchdog Reset includes both Watchdog Timeout Reset and Independent Watchdog Reset. Watchdog Timeout Reset (WTR) is based on an 8-bit incremental counter with a count clock period of $131072/F_{sys}$. When Watchdog Timeout Reset is enabled, the entire system will be reset once this counter overflows.

Independent Watchdog (IWDG) is internally a decrementing 12-bit counter that will generate a system reset when the counter value is decremented to 0.

4.1.4 Global Reset by Waking under Shutdown Mode (GRWSM)

Once the system enters the shutdown mode (see the power management chapter for details), the system will perform the wake-up operation in an orderly manner under the action of the wake-up signal, and the system will perform a global reset after wake-up. This reset effect is similar to that of power-on reset.

4.1.5 Local Reset by Waking (LRW)

If the system is awakened from sleep mode, a reset will be generated after the associated power is ready. It is a partial reset, with a selective reset of the registers that are powered down in sleep mode as needed.

In sleep mode, the registers of each functional module are divided into 3 categories:

The first type is the key registers belonging to the functional module that requires data retention (such as configuration/mode, etc.). At the time of sleep, the auxiliary power continues to supply power, and the data is not lost, both sleep and wake-up have no effect on data;

The second type is the regenerative registers belonging to the functional module that requires data retention (such as counters, FIFOs, etc.). The power is turned off during sleep, and the data is a random number (such as FIFO memory cell) or reset (such as FIFO counter) after waking up;

The third type is the register belonging to the functional module that does not require data retention. The power is turned off during sleep, and the data is a random number (such as FIFO memory cell) or reset (such as FIFO counter, configuration/mode register) after waking up.

LRW is used for the latter 2 reset registers above.

4.2 Safe Access

The attributes of some registers of the system are "RWA" or "WA", indicating that the current register can be safely accessed and can be read directly, but needs to enter the safe access mode when write-in:

First write 0x57 to the R8_SAFE_ACCESS_SIG register;

Then write 0xA8 to the R8_SAFE_ACCESS_SIG register;

At this time, you can enter the safe access mode and operate the registers with the attribute "RWA/WA". After that, about 112 system frequency cycles (Tsys) are in safe mode, and one or more secure registers can be rewritten within the valid period. The safe mode will be automatically terminated after the above validity period is exceeded. Or you can write 0x00 in the R8_SAFE_ACCESS_SIG register in advance to terminate the safe mode.

4.3 Register Description

Table 4-1 System control registers

Name	Access address	Description	Reset value
R8_SAFE_ACCESS_SIG	0x40001040	Safe access flag register	0x00
R8_CHIP_ID	0x40001041	Chip ID register	0x96
R8_SAFE_ACCESS_ID	0x40001042	Safe access ID register	0x0C
R8_WDOG_COUNT	0x40001043	Watchdog counter register	0x00
R8_RESET_STATUS	0x40001044	Reset status register	0x01
R8_GLOB_ROM_CFG	0x40001044	FlashROM application configuration register	0x01

R8_GLOB_CFG_INFO	0x40001045	Global configuration information status register	0xX9
R8_RST_WDOG_CTRL	0x40001046	Watchdog and reset configuration register	0x00
R8_GLOB_RESET_KEEP	0x40001047	Reset keep register	0x00
R8_PREPG_CTRL	0x40001800	FlashROM preprogramm control register	0x00
R8_BUF_CTRL	0x40001801	FlashROM page latch control register	0x00
R8_ERASE_CTRL	0x40001802	FlashROM erase control register	0x00
R8_PROGRAM_CTRL	0x40001803	FlashROM program control register	0x00
R8_READ_CTRL	0x40001807	FlashROM read control register	0x00
R8_RDPRCTDIS_CTRL	0x40001808	FlashROM read protection control register	0x00
R8_CHIP_STATUS	0x40001809	Chip status register	0x04
R32_FLASH_ADDR	0x4000180C	FlashROM address register	0x00000000
R32_FLASH_DATA	0x40001810	FlashROM data register	0x00000000
R32_FLASH_DATA1	0x40001814	FlashROM data register 1	0x00000000
R32_FLASH_DATA2	0x40001818	FlashROM data register 2	0x00000000
R32_FLASH_DATA3	0x4000181C	FlashROM data register 3	0x00000000
R32_FLASH_STATUS	0x40001820	FlashROM status register	0x000DXXXX

Safe Access Flag Register (R8_SAFE_ACCESS_SIG)

Bit	Name	Access	Description	Reset value
[7:0]	R8_SAFE_ACCESS_SIG	WO	Safe access flag register. Some registers (access attribute is RWA) are protection registers and write operation can be conducted only after entering the safe access mode. Write 0x57 first and then 0xA8 to this register, to enter the safe access mode. The time is limited to about 16 main clock cycles (Tsys), and automatic protection is enabled if it is exceeded. You can write any other value to force direct exit from the safe access mode and return to the protected state.	0
7	Reserved	RO	Reserved	0
[6:4]	RB_SAFE_ACC_TIMER	RO	Count the current safe access time.	0
3	RB_SAFE_ACC_ACT	RO	Current safe access mode status: 1: Writable in unlocked/safe access mode; 0: Locked, register with RWA attribute cannot be rewritten.	0
2	Reserved	RO	Reserved	0
[1:0]	RB_SAFE_ACC_MODE	RO	Current safe access mode status. 11: Safe mode, register with RWA attribute can be written; Others: Non-safe mode.	0

Chip ID Register (R8_CHIP_ID)

Bit	Name	Access	Description	Reset value
[7:0]	R8_CHIP_ID	RF	Always 96h, to identify the chip.	96h

Safe Access ID Register (R8_SAFE_ACCESS_ID)

Bit	Name	Access	Description	Reset value
[7:0]	R8_SAFE_ACCESS_ID	RF	Fixed value 0Ch.	0Ch

Watchdog Counter Register (R8_WDOG_COUNT)

Bit	Name	Access	Description	Reset value
[7:0]	R8_WDOG_COUNT	RW	Watchdog counter, whose initial value can be preset, is always automatically incremented, and can be cycled from 0xFF to 0x00 and then continue. Count cycle = 131072/Fsys.	0

Reset Status Register (R8_RESET_STATUS) / FlashROM Application Configuration Register (R8_GLOB_ROM_CFG)

Bit	Name	Access	Description	Reset value
7	RB_ROM_CODE_WE	RWA	Erase/program enable bit of CodeFlash in FlashROM program memory area: 1: Erase/program is enabled; 0: Erase/program protection for this area.	0
6	RB_ROM_DATA_WE	RWA	Erase/program enable bit of DataFlash in FlashROM data memory area: 1: Erase/program is enabled; 0: Erase/program protection for all.	0
5	RB_ROM_CTRL_EN	RWA	FlashROM access control interface enable: 1: Control enabled; 0: Access disabled.	0
4	RB_ROM_CODE_OFS	RWA	Select the start offset address of user program code in FlashROM. This value is not affected by SR, WTR or GRWSM, and can be cleared only when RPOR is valid: 1: 0x040000 (skip the first 256KB in ROM). 0: 0x000000;	0
3	Reserved	RO	Reserved	0
[2:0]	RB_RESET_FLAG	RO	Last reset state: 000: Software reset SR (generated when RB_WDOG_RST_EN=0 and software reset, otherwise, it can be reset but this status is not generated); 001: Real Power-on reset (RPOR); 010: Watchdog timeout reset (WTR);	001b

			011: Reserved; 101: Global Reset by Waking under Shutdown Mode (GRWSM) 100/110/111: Local Reset by waking (LRW), and the last reset is SR/WTR.	
--	--	--	--	--

Global Configuration Information Status Register (R8_GLOB_CFG_INFO)

Bit	Name	Access	Description	Reset value
[7:6]	Reserved	RO	Reserved	11b
5	RB_BOOT_LOADER	RO	Bootloader status: 1: Currently in Bootloader status; 0: Currently in application program status.	1/0
4	Reserved	RO	Reserved	Reserved
3	RB_CFG_BOOT_EN	RO	System BootLoader enable status: 1: Enable; 0: Disable.	1
[2:0]	Reserved	RO	Reserved	001b

Watchdog and Reset Configuration Register (R8_RST_WDOG_CTRL)

Bit	Name	Access	Description	Reset value
[7:5]	Reserved	RO	Reserved	0
4	RB_WDOG_INT_FLAG	RW1	Watchdog timer interrupt flag: 1: Watchdog count overflows, that is, R8_WDOG_COUNT is detected to progressively increase from 0xFF to 0x00; 0: Watchdog count has not overflowed. Write 1 to clear, or reload the watchdog counter value (R8_WDOG_COUNT) to clear, or execute SEV() to clear.	0
3	Reserved	RO	Reserved	0
2	RB_WDOG_INT_EN	RWA	Watchdog timer interrupt enable bit: 1: Enable, an interrupt will be generated after the watchdog count overflows; 0: Disable the watchdog timer interrupt.	0
1	RB_WDOG_RST_EN	RWA	Watchdog timeout reset enable bit: 1: Enable, system is reset after the watchdog count overflows; 0: Only used as watchdog timer. <i>Note: After this bit is set to 1, the software reset operation will not affect the RB_RESET_FLAG status.</i>	0
0	RB_BOOT_LOADER_MAN	RO	Manual BOOT LOADER entry flag: 1: Manually enter BOOT; 0: Idle, no action.	0

			(When RB_WDOG_INT_EN and RB_ROM_CODE_WE are set to 1, RB_ROM_DATA_WE is 0, and $128 \leq \text{R8_WDOG_COUNT} < 192$, bit RB_BOOT_LOAD_MAN is 1 and boot starts from BOOT; otherwise, bit RB_BOOT_LOAD_MAN is 0).	
	RB_SOFTWARE_RESET	WA/ WZ	System software reset control; automatically cleared after reset: 1: Perform system software reset; 0: Idle, no action.	0

Reset Keep Register (R8_GLOB_RESET_KEEP)

Bit	Name	Access	Description	Reset value
[7:0]	R8_GLOB_RESET_KEEP	RW	Reset keep register. The value of this register is not affected by manual reset, software reset, watchdog reset or ordinary wake-up reset.	0

FlashROM Preprogramm Control Register (R8_GLOB_RESET_KEEP)

Bit	Name	Access	Description	Reset value
[7:5]	Reserved	RO	Reserved	0
4	RB_PREPG_START	RWA	Preprogramming start. Remains high throughout the preprogramming process. Automatically resets to 0 upon completion. If RRB_PREG_AUTO_EN is set to 1, it will not reset to 0 until the corresponding erase process completes. 1: Initiate preprogramming.	0
3	Reserved	RO	Reserved	0
[2:1]	RB_PREG_MODE	RWA	Preprogramming Modes: 00: Page preprogramming (256 BYTE); 01: Sector preprogramming (2K); 10: BANK area preprogramming (64K), determined by BIT[8] of address R32_FLASH_ADDR whether odd or even pages. 1: Odd pages; 0: Even pages; 11: Bank area preprogramming (64K).	0
0	RB_PREG_AUTO_EN	RWA	Preprogrammed auto mode: 1: After preprogramming completes, an automatic erase operation will occur (requires prior configuration of the erase command RB_ERASE_MODE); 0: After preprogramming completes, no	0

			automatic erase operation will occur.	
--	--	--	---------------------------------------	--

FlashROM Page Latch Control Register (R8_BUF_CTRL)

Bit	Name	Access	Description	Reset value
[7:6]	Reserved	RO	Reserved	0
5	RB_BUF_LOAD_AUTO	RWA	1: Page buffer auto-load mode enabled. When enabled, writing data to R32_FLASH_DATA triggers RB_BUF_LOAD_START to go high and initiates loading. Upon completion, RB_BUF_LOAD_START automatically resets to low. 0: Page buffer auto-load mode disabled.	0
4	RB_BUF_LOAD_START	RWA	Page register loading is initiated. The page register remains high throughout the loading process and automatically clears to zero upon completion. 1: Initiate page register loading.	0
[3:1]	Reserved	RO	Reserved	0
0	RB_BUF_CLR_START	RWA	Page latch clear initiated The page latch remains high throughout the clear process and automatically resets to zero upon completion. 1: Initiate page latch clear.	0

FlashROM Erase Control Register (R8_ERASE_CTRL)

Bit	Name	Access	Description	Reset value
[7:5]	Reserved	RO	Reserved	0
4	RB_ERASE_START	RWA	Erase start. Remains high throughout the erase process. Automatically resets to 0 upon completion. 1: Initiate FlashROM erase.	0
[3:2]	Reserved	RO	Reserved	0
[1:0]	RB_ERASE_MODE	RWA	00: Page erase; 01: Sector erase; 10/11: BANK erase.	0

FlashROM Program Control Register (R8_PROGRAM_CTRL)

Bit	Name	Access	Description	Reset value
[7:1]	Reserved	RO	Reserved	0
0	RB_PROGRAM_START	RWA	Page programming starts, remains high throughout the programming process, and automatically resets to 0 upon completion; 1: Initiate FlashROM page programming.	0

FlashROM Read Control Register (R8_READ_CTRL)

Bit	Name	Access	Description	Reset value
[7:6]	Reserved	RO	Reserved	0
[5:4]	RB_FLASH_CLK	RWA	Clock adjustment for FLASH pre-programming/erasing/page latch loading/page latch clearing/programming/reading: 00: High level is half of the system clock cycle; 01/10: High level is one system clock cycle; 11: The high level is two system clock cycles. <i>Note: The high level must be greater than 10ns.</i>	0
3	Reserved	RO	Reserved	0
2	RB_WRRD_ADDR_INC	RWA	1: After reading and writing once, the address will be automatically increased by 1; 0: The address remains unchanged after reading and writing once. <i>Note: "plus 1" is 4 words.</i>	0
1	RB_RD_START	RO	The read operation is started, and cleared automatically after the read operation is completed: 1: Start the read operation.	0
0	RB_CFG_RD_EN	RWA	1: R32_FLASH_DATA will update the read value during the read operation, but data cannot be written at this address; 0: Write enables R32_FLASH_DATA to write programming data.	0

FlashROM Read Protection Control Register (R8_RDPRCTDIS_CTRL)

Bit	Name	Access	Description	Reset value
[7:0]	R8_RDPRCTDIS_CTRL	WO	Read protection control register: The address is sequentially configured with 0x3C/0x9A/0x6D/0x73 to enable the read protection to be lifted; The address is configured with 0x3C/0x9A/0x6D/0x99 clear read protection release logic in turn.	0
[7:6]	RB_RD_PROTECTDIS_START	RO	Current read protection control status: 11: In R8_RDPRCTDIS_CTRL, configure 0x73 to enable the start of read protection release, or configure 0x99 to clear the read	

			protection release logic.	
[5:4]	Reserved	RO	Reserved	0
[3:1]	RB_RDPROTECT_DISS TATE	RO	000: Read protection cancellation is not started; 001~011: Reading protection is being released; 100: Read protection is released; 101: Read protection failed.	0
0	RB_RD_PROTECTDIS_ FLAG	RO	1: In the process of removing read protection; 0: Not in the process of removing read protection.	0

Chip Status Register (R8_CHIP_STATUS)

Bit	Name	Access	Description	Reset value
[7:4]	Reserved	RO	Reserved	0
3	RB_FLASH_HALTED	RO	In DEBUG mode: 1: FLASH prohibits operation; 0: FLASH allows operation.	0
2	Reserved	RO	Reserved	1
1	RB_DBG_MODE	RO	1: Enter the debugging mode; 0: Debugging mode is not entered.	0
0	RB_RD_PROTECT	RO	1: Read protection is on; 0: Read protection removed.	0

FlashROM Address Register (R32_FLASH_ADDR)

Bit	Name	Access	Description	Reset value
[31:19]	Reserved	RO	Reserved	0
[18:2]	R32_FLASH_ADDR	RWA	When RB_CFG_RD_EN = 0, R32_FLASH_ADDR [7:2] represents the loading address of the page latch, and R32_FLASH_ADDR[18:2] represents the programming address (the loading and programming operations of the page latch are independent, so they do not conflict, and are time-shared); When RB_CFG_RD_EN = 1, R32_FLASH_ADDR [18:2] represents the read data address.	0
[1:0]	Reserved	RO	Reserved	0

FlashROM Data Register (R32_FLASH_DATA)

Bit	Name	Access	Description	Reset value
-----	------	--------	-------------	-------------

[31:0]	R32_FLASH_DATA	RWA	RB_CFG_RD_EN = 0 indicates programming data; RB_CFG_RD_EN = 1 indicates read data; In 4-word alignment mode, R32_FLASH_DATA represents the first word; R32_FLASH_DATA1 represents the second word; R32_FLASH_DATA2 represents the third word; R32_FLASH_DATA3 represents the fourth word; For example: If the read address is R32_FLASH_ADDR[18:2] = 0x00000, then after the read operation completes, R32_FLASH_DATA will hold the value of R32_FLASH_ADDR[18:2] = 0x00000. R32_FLASH_DATA1 holds the value at R32_FLASH_ADDR[18:2] = 0x00001. R32_FLASH_DATA2 holds the value at R32_FLASH_ADDR[18:2] = 0x00002. R32_FLASH_DATA3 holds the value at R32_FLASH_ADDR [18:2] = 0x00003. That is, four words can be read at once.	0
--------	----------------	-----	--	---

FlashROM Data Register 1 (R32_FLASH_DATA1)

Bit	Name	Access	Description	Reset value
[31:0]	R32_FLASH_DATA1	RO	Flash read data 1.	0

FlashROM Data Register 2 (R32_FLASH_DATA2)

Bit	Name	Access	Description	Reset value
[31:0]	R32_FLASH_DATA2	RO	Flash read data 2.	0

FlashROM Data Register 3 (R32_FLASH_DATA3)

Bit	Name	Access	Description	Reset value
[31:0]	R32_FLASH_DATA3	RO	Flash read data 3.	0

FlashROM Status Register (R32_FLASH_STATUS)

Bit	Name	Access	Description	Reset value
31	RB_CTRL_BUSY	RO	1: Indicates FLASH is performing pre-programming/erase/page register clear/page register load/program/read.	0
[30:20]	Reserved	RO	Reserved	0

19	RB_FLASH_READY	RO	1: Preprogramming/erase/page latch clear/page latch load/program/read operations are permitted. 0: Preprogramming/erase/page latch clear/page latch load/program/read operations are not permitted.	1
[18:0]	Reserved	RO	Reserved	5XXXXh

4.4 Flash-ROM Operation Steps

4.4.1 Program

Before programming begins, write R8_RESET_STATUS (0x40001044) to 0xe1 to enable the erase programming channel and open the security register channel.

Programming operations require five steps:

- (1) Pre-program the target programming area;
- (2) Erase the target programming area;
- (3) Clear the page latch;
- (4) Write the programming data into the page latch;
- (5) Program the contents of the page latch into the specified page address;
- (6) If the pre-programmed and erased area exceeds the page programming size, repeat steps (3) to (5) to continue programming. For example, if both pre-programming and erasing are 2K, steps (3) to (5) can be performed 8 times.

Note: The erase operation must execute both Step (1) and Step (2).

Below are the specific operations for each step (in conjunction with FLASH-related registers).

Preprogramming:

- (1) Address R32_FLASH_ADDR (0x4000180C): Write the address required for preprogramming;
- (2) Address R8_PREPG_CTRL (0x40001800): Write the preprogramming command. If bit[0] is also set to 1, the erase mode must be written to R8_ERASE_CTRL (0x40001802) bit[1:0] beforehand. Simultaneously set RB_PREPG_START to 1 (i.e., command and enable can be written in one operation) to start preprogramming;
- (3) Sequentially monitor until RB_CTRL_BUSY clears and RB_FLASH_READY sets to 1, indicating preprogramming completion.

Erase:

- (1) Address R32_FLASH_ADDR (0x4000180C): Write the address to be erased;
- (2) Address R8_ERASE_CTRL (0x40001802): Write the erase command (RB_ERASE_MODE) and enable erase start (RB_ERASE_START);
- (3) Monitor RB_CTRL_BUSY clearing to 0 and RB_FLASH_READY setting to 1 sequentially to indicate erase completion.

Clearing the page latch:

- (1) Write RB_BUF_CLR_START to 1 at address R8_BUF_CTRL (0x40001801) to initiate clearing the page latch;
- (2) After sequentially polling RB_CTRL_BUSY (reset to 0) and RB_FLASH_READY (set to 1), the page latch clearing is complete.

Write the data to be programmed into the page latch:

- (1) Address R8_FLASH_CTRL2 (0x40001807): Write RB_CFG_RD_EN to 0 to ensure R32_FLASH_DATA is valid. (Note: Throughout the entire programming process, only ensure RB_CFG_RD_EN is 0; it is not necessary to write it each time. It can be written once at the start of programming.)
- (2) Address R8_FLASH_CTRL2 (0x40001807): Set RB_WRRD_ADDR_INC to 1 (After each page latch data write completes, address R32_FLASH_ADDR[7:2] will automatically increment by 1; the initial address R32_FLASH_ADDR[7:2] requires configuration). Address R8_BUF_CTRL (0x40001801), set RB_BUF_LOAD_AUTO to 1 (RB_BUF_LOAD_START is automatically triggered after each write to R32_FLASH_DATA).
- (3) Address R32_FLASH_ADDR (0x4000180C): Configure the initial address R32_FLASH_ADDR[7:2];
- (4) Address R32_FLASH_DATA (0x40001810): Write R32_FLASH_DATA;
- (5) Sequentially check until RB_CTRL_BUSY is set to 0 and RB_FLASH_READY is set to 1, indicating the data write is complete;
- (6) Repeat steps (4) and (5).

Programming:

- (1) Address R32_FLASH_ADDR (0x4000180C): Write the address required for programming;
- (2) Address R8_PROGRAM_CTRL (0x40001803): Set RB_PROGRAM_START to 1 to initiate page programming;
- (3) Page programming completes when RB_CTRL_BUSY is cleared and RB_FLASH_READY is set to 1.

4.4.2 Read Operation

Read operation:

- (1) Read address R8_FLASH_CTRL2 (0x40001807), write RB_CFG_RD_EN to 1 to ensure R32_FLASH_DATA is read data. Configure read address R32_FLASH_ADDR[18:2], set RB_WRRD_ADDR_INC to 0;
- (2) Read address R32_FLASH_DATA (0x40001810). After sequentially verifying that RB_CTRL_BUSY is 0 and RB_FLASH_READY is 1, proceed to the next read operation.

For example, if the read address is R32_FLASH_ADDR[18:2] = 0x00000; After the read operation completes, R32_FLASH_DATA holds the value at R32_FLASH_ADDR[18:2] = 0x00000, and R32_FLASH_DATA1 holds the value at R32_FLASH_ADDR[18:2] = 0x00001. R32_FLASH_DATA2 holds the value at R32_FLASH_ADDR[18:2] = 0x00002. R32_FLASH_DATA3 holds the value at R32_FLASH_ADDR[18:2] = 0x00003. This means 4 words can be read in a single operation.

Automatic Address Increment Read Operation:

- (1) Configure the initial address to read, i.e., R32_FLASH_ADDR (0x4000180C);
- (2) Configure read enable and read/write address increment enable, i.e., write R8_FLASH_CTRL2 (0x40001807) (by byte) to 0x05;
- (3) Read the data register, R32_FLASH_DATA (0x40001810), to trigger a read operation. This updates R32_FLASH_DATA, R32_FLASH_DATA1, R32_FLASH_DATA2, and R32_FLASH_DATA3, automatically incrementing the address to the next 128-bit block;
- (4) Follow the read sequence: R32_FLASH_DATA3, R32_FLASH_DATA2, R32_FLASH_DATA1, R32_FLASH_DATA. Reading R32_FLASH_DATA will trigger another read operation;
- (5) Repeat step 4.

Example: Initial address is 0x00000:

Step 3 updates the address to 0x00010, updating the data in R32_FLASH_DATA, R32_FLASH_DATA1, R32_FLASH_DATA2, and R32_FLASH_DATA3 to 0x00000, 0x00004, 0x00008, and 0x0000C respectively. Step 4 reads R32_FLASH_DATA3, R32_FLASH_DATA2, R32_FLASH_DATA1, and R32_FLASH_DATA in sequence. Upon reading R32_FLASH_DATA, it triggers a read at address 0x00010 and updates DATA to the next 128-bit block. This process repeats iteratively.

4.5 Unique ID

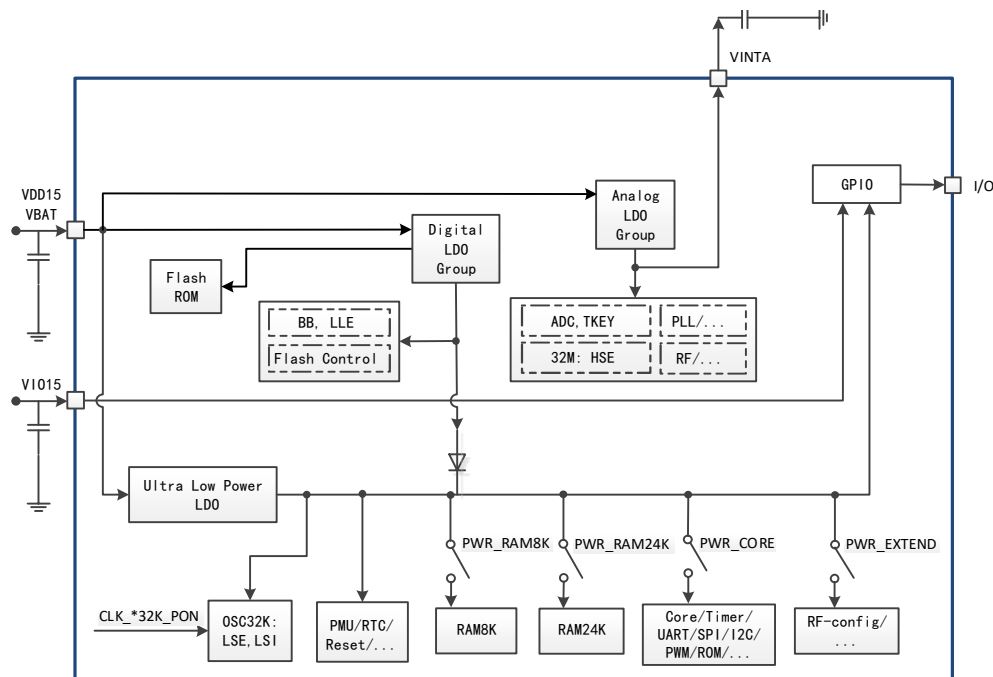
Each chip has a unique ID (Chip identification number) when it is delivered from the factory. The ID data and its checksum are 8 bytes in total, stored in the read-only area of chip. Please refer to the routines for details.

Chapter 5 Power Control

5.1 Power Management

The chip incorporates a built-in Power Management Unit (PMU). System power is supplied from VDD15 input. Through multiple integrated LDO voltage regulators, it provides the required power for the system's Flash ROM, digital circuits (including the core), and analog circuits (including high-frequency oscillators, PLL, ADC, and RF transceivers). GPIO power is sourced from VDD15 input.

Figure 5-1 Power system



The chip offers 4 low-power modes: idle mode, suspend mode, sleep mode, and power-down mode.

To reduce system power consumption during sleep, the system's main LDO can be disabled and switched to auxiliary power supplied by the built-in ultra-low-power ULP-LDO. When the system enters sleep or power-down mode, except for continuously powered units like the power management and RTC registers, the system's upper 8KB and lower 24KB SRAM, the core, and all peripherals can be selectively powered. The LSE/LSI can also be optionally enabled.

5.2 Register Description

Table 5-1 Power management registers

Name	Access address	Description	Reset value
R8_SLP_CLK_OFF0	0x4000100C	Sleep clock control register 0	0x00
R8_SLP_CLK_OFF1	0x4000100D	Sleep clock control register 1	0x00
R8_SLP_WAKE_CTRL	0x4000100E	Wakeup event configuration register	0x20
R8_SLP_POWER_CTRL	0x4000100F	Peripheral sleep power control register	0x00
R8_LONG_RST_CFG	0x40001014	Long reset configuration register	0x00

R8_SLP_CLK_OFF2	0x40001015	Sleep clock control register 2	0x00
R16_SLP_WAKE_CFG	0x40001016	Sleep configuration register	0x0900
R16_POWER_PLAN	0x40001020	Sleep power management register	0x11DF
R16_AUX_POWER_ADJ	0x40001022	Auxiliary power adjustment control register	0xFFFF
R8_BAT_DET_CTRL	0x40001024	Battery voltage detection control register	0x00
R8_BAT_DET_CFG	0x40001025	Battery voltage detection configuration register	0x00
R8_BAT_STATUS	0x40001026	Battery status register	0x00

Sleep Clock Control Register 0 (R8_SLP_CLK_OFF0)

Bit	Name	Access	Description	Reset value
[7:6]	Reserved	RO	Reserved	0
5	RB_SLP_CLK_UART1	RWA	Clock source of UART1: 1: Disable; 0: Enable.	0
4	RB_SLP_CLK_UART0	RWA	Clock source of UART0: 1: Disable; 0: Enable.	0
3	Reserved	RO	Reserved	0
2	RB_SLP_CLK_TMR2	RWA	Clock source of timer 2: 1: Disable; 0: Enable.	0
1	RB_SLP_CLK_TMR1	RWA	Clock source of timer 1: 1: Disable; 0: Enable.	0
0	RB_SLP_CLK_TMR0	RWA	Clock source of timer 0: 1: Disable; 0: Enable.	0

Sleep Clock Control Register 1 (R8_SLP_CLK_OFF1)

Bit	Name	Access	Description	Reset value
7	RB_SLP_CLK_BLE	RWA	Clock source of BLE controller: 1: Disable; 0: Enable.	0
6	Reserved	RO	Reserved	0
5	RB_SLP_CLK_ADC	RWA	ADC clock source: 1: Disable; 0: Enable.	0
4	Reserved	RO	Reserved	0
3	RB_SLP_CLK_I2C	RWA	I2C clock source: 1: Disable; 0: Enable.	0
2	RB_SLP_CLK_PWMX	RWA	PWMx clock source: 1: Disable; 0: Enable.	0
1	RB_SLP_CLK_SPI1	RWA	SPI1 clock source: 1: Disable; 0: Enable.	0
0	RB_SLP_CLK_SPI0	RWA	SPI0 clock source: 1: Disable; 0: Enable.	0

Wakeup Event Configuration Register (R8_SLP_WAKE_CTRL)

Bit	Name	Access	Description	Reset value
7	RB_GPIO_WAKE_MODE	RWA	GPIO wakeup mode enable:	0

			1: RB_SLP_GPIO_EDGE_MODE=1, Edge-triggered wake-up; RB_SLP_GPIO_EDGE_MODE=0, Rising edge wake-up. 0: Level-triggered wake-up.	
6	RB_WAKE_EV_MODE	RWA	Wake-up event internal memory mode enable: 1: Enable memory, support short-pulse event wake-up; 0: Disable memory, the event should remain valid until wake up.	0
5	RB_SLP_BAT_WAKE	RWA	Enable battery low voltage event wake up system: 1: Enable; 0: Disable.	1
4	RB_SLP_GPIO_WAKE	RWA	Enable GPIO event wake up system: 1: Enable; 0: Disable.	0
3	RB_SLP_RTC_WAKE	RWA	Enable RTC event wake up system: 1: Enable; 0: Disable.	0
2	RB_SLP_GPIO_EDGE_MODE	RWA	When RB_GPIO_WAKE_MODE = 1, GPIO event wake-up mode selection: 1: Non-polarity; any edge can trigger wake-up; 0: Edge-triggered wake-up.	0
1	RB_SLP_UART_WAKE	RWA	Enable UART event wakeup system: 1: Enable; 0: Disable.	0
0	Reserved	RO	Reserved	0

Peripheral Sleep Power Control Register (R8_SLP_POWER_CTRL)

Bit	Name	Access	Description	Reset value
[7:6]	RB_RAM_RET_LV	RWA	SRAM sleep auxiliary power low-voltage Enable: 00: Disabled; 01: Low-Power Mode 1; 10: Low-Power Mode 2; 11: Low-Power Mode 3.	0
5	RB_SLP_CLK_RAM8K	RWA	SRAM clock control of RAM8K: 1: Disable; 0: Enable.	0
4	RB_SLP_CLK_RAMX	RWA	Clock control of main SRAM (RAM24K): 1: Disable; 0: Enable.	0
[3:2]	Reserved	RO	Reserved	0
[1:0]	RB_WAKE_DLY_MOD	RWA	Delay cycle select after wake-up: 11: No delay, 8 cycles +T _{SUCLK} , disabled; 10: Ultra short delay, 70 cycles +T _{SUCLK} ; 01: Short delay, 520 cycles +T _{SUCLK} ,	00b

			recommended; 00: Long delay, 3590 cycles +T _{SUCLK} . Where T _{SUCLK} depends on the sleep mode and clock configuration and may contain the start-up time of T _{SUHSE} or PLL or both combined, refer to the example in the evaluation board for the exact combination.	
--	--	--	--	--

Long Reset Configuration Register (R8_LONG_RST_CFG)

Bit	Name	Access	Description	Reset value
[7:3]	Reserved	RO	Reserved	0
[2:1]	RB_LONG_TIM_SEL	RWA	Long reset duration selection. When using HSE clock: 11: 32.768ms; 10: 25.000ms; 01: 20.000ms; 00: 15.000ms. When using HSI clock: 11: 65.536ms; 10: 50.000ms; 01: 40.000ms; 00: 30.000ms.	0
0	RB_LONG_RST_EN	RWA	Long reset enable: 1: Enable long reset; 0: Disable long reset.	0

Sleep Clock Control Register 2 (R8_SLP_CLK_OFF2)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved	0
6	RB_SLP_ENC_WAKE	RWA	Enable encoder events to wake the system: 1: On; 0: Off.	0
5	Reserved	RO	Reserved	0
4	RB_CLK_OFF_ENC	RWA	1: Disable encoder clock; 0: Enable encoder clock.	0
[3:2]	Reserved	RO	Reserved	0
1	RB_CLK_OFF_DEBU G	RWA	1: Disable 32 MHz clock for two-wire debugging; 0: Enable 32 MHz clock for two-wire debugging.	0
0	Reserved	RO	Reserved	0

Sleep Configuration Register (R16_SLP_WAKE_CFG)

Bit	Name	Access	Description	Reset value
[15:13]	Reserved	RO	Reserved	0
12	RB_DEBUG_EN	RWA	1: Debugging enabled; 0: Debugging disabled.	0
11	RB_UART_DLY_IDLE	RO	After enabling RB_UART_DLY_EN, the UART receive status is: 1: No data received; 0: Data received.	1
10	RB_UART_WAKE_INT	RW1	UART wake-up interrupt flag: 1: Interrupt generated; 0: No interrupt generated.	0
9	RB_UART_DLY_EN	RWA	UART receive delay enable: 1: Enable delay; 0: Disable delay.	0
8	RB_ACAUTO_ENABLE	RW	1: Safety register channel automatically closes; 0: Safety register channel remains open after activation.	1
7	Reserved	RO	Reserved	0
[6:5]	RB_PRECLK_CNT_SEL	RWA	Sleep time configuration, sleep duration = number of cycles $\times$ Fsys: 11: 256; 10: 512; 01: 1024; 00: 2048.	0
4	RB_PRECLK_CNT_EN	RWA	1: After sleep wake-up, wait for a specified period before releasing fclk; 0: Release fclk immediately after sleep wake-up.	0
[3:0]	Reserved	RO	Reserved	0

Sleep Power Management Register (R16_POWER_PLAN)

Bit	Name	Access	Description	Reset value
15	RB_PWR_PLAN_EN	RWA/WZ	Sleep power planning control enable: 1: Enable planning; 0: Disable or end planning. The power planning is enabled for execution when entering sleep or power-off mode later, and this bit is automatically cleared after execution.	0
14	RB_XT_PRE_EN	RWA	External 32MHz clock (HSE) pre-wake enable: 1: Enabled. Counting must be performed using LSI or LSE. 0: Disabled.	0
13	Reserved	RWA	Reserved, 1 must be written.	1

[12:11]	RB_XT_PRE_CFG	RWA	External 32MHz clock (HSE) pre-wake time: 00: 0.531ms; 01: 0.968ms; 10: 1.406ms; 11: 1.781ms.	10b
[10:9]	Reserved	RO	Reserved	0
8	RB_PWR_LDO_EN	RWA	Internal LDO control (sleep planning): 1: Turn on LDO; 0: Plan to turn off LDO, saving more power.	1
7	RB_PWR_SYS_EN	RWA	System power control (sleep planning): 1: Provide system power; 0: Turn off the system power, plan to enter sleep mode or power-off mode.	1
6	RB_MAIN_ACT	RWA	Main power supply selection 1: ULL_LDO; 0: LDO.	1
5	Reserved	RO	Reserved	0
4	RB_PWR_RAM30K	RWA	SRAM power supply of RAM30K (sleep planning): 1: Dual power; 0: No auxiliary power.	1
3	RB_PWR_EXTEND	RWA	RF configuration power supply (sleep planning): 1: Dual power; 0: No auxiliary power.	1
2	RB_PWR_CORE	RWA	Power of the core and basic peripherals (sleep planning): 1: Dual power; 0: No auxiliary power.	1
1	RB_PWR_RAM2K	RWA	SRAM power supply of RAM2K (sleep planning): 1: Dual power; 0: No auxiliary power.	1
0	RB_PWR_XROM	RWA	FlashROM power supply (sleep planning): 1: Continuous power; 0: Power off during sleep.	1

Note: This register is preset for sleep planning, and its power configuration will take effect after entering the low-power sleep mode and power-down mode.

Auxiliary Power Adjustment Control Register (R16_AUX_POWER_ADJ)

Bit	Name	Access	Description	Reset value
[15:12]	Reserved	RO	Reserved	00XXb
[11:8]	RB_CFG_IVREF	RWA	Reserved, read-only. Write operation has no effect.	XXXXb
[7:5]	Reserved	RO	Reserved	0
[4:3]	Reserved	RWA	Reserved, the original value must be kept unchanged when writing.	XXb
[2:0]	RB_ULPLDO_ADJ	RWA	The auxiliary power output voltage adjustment value of ultra-low-power LDO	XXXb

			(the value is for reference only, and it is not recommended to modify): 000: 0.77V; 001: 0.80V; 010: 0.83V; 011: 0.85V; 100: 0.88V; 101: 0.91V; 110: 0.95V; 111: 0.98V.	
--	--	--	---	--

Battery Voltage Detection Control Register (R8_BAT_DET_CTRL)

Bit	Name	Access	Description	Reset value
[7:4]	Reserved	RO	Reserved	0
3	RB_BAT_LOW_IE	RWA	Battery low voltage interrupt enable: 1: Enable; 0: Disable.	0
2	RB_BAT_LOWER_IE	RWA	Battery ultra-low voltage interrupt enable: 1: Enable; 0: Disable.	0
1	RB_BAT_MON_EN	RWA	Low-power battery voltage monitor function enable: 1: Enable, increasing about 1uA current; 0: Disable.	0
0	Reserved	RO	Reserved	0

Battery Voltage Detection Configuration Register (R8_BAT_DET_CFG)

Bit	Name	Access	Description	Reset value
[7:3]	Reserved	RO	Reserved	0
[2:0]	RB_BAT_LOW_VTH	RWA	When RB_BAT_MON_EN=1, set the low-power low-voltage monitoring threshold: 000: 1.06V; 001: 1.11V; 010: 1.16V; 011: 1.21V; 100: 1.26V; 101: 1.31V; 110: 1.36V; 111: 1.41V.	0

Battery Status Register (R8_BAT_STATUS)

Bit	Name	Access	Description	Reset value
[7:2]	Reserved	RO	Reserved	0
1	RB_BAT_STAT_LOW	RO	The result of battery low-voltage detection or low-voltage monitoring, indicating that the battery voltage is in low voltage status: 1: Below the low voltage threshold; 0: No.	0
0	Reserved	RO	Reserved	0

5.3 Low-power Mode

After the system is reset, the microcontroller is in normal operation. When the MCU does not need to run, an appropriate low-power mode can be selected to save power. The user needs to select an appropriate low-power

mode based on conditions such as the lowest power consumption, the fastest startup time and available wake-up events.

The chip provides the following 4 main low-power modes:

- Idle mode

All peripherals remain powered, the core stops running, and the clock system is running. After a wake-up event is detected, it can be woken up immediately.

- Halt mode

On the basis of idle mode, the clock system stops. After a wake-up event is detected, the clock will run first, and then the core will be woken up to run.

- Sleep mode:

The main LDO is turned off, and the ultra-low-power ULP-LDO maintains the power supply of PMU, core and basic peripherals. You can select whether to turn on LSE or LSI, and to maintain power supply of RAM8K, RAM24K and RF configurations. After a wake-up event is detected, first the main LDO is turned on, then the clock will run, and finally the core will be woken up, the program will continue to run, and a higher frequency can be reset when needed.

- Shutdown mode:

Based on the sleep mode, the core and basic peripherals, RF configurations are turned off, and you can select whether to turn on LSE or LSI, and to maintain power supply of RAM8K and RAM24K. After detecting a wake-up event, PMU will perform a GRWSM reset, and the software can distinguish RPOR based on the reset flag RB_RESET_FLAG and the data retained in optional RAM.

The following table describes in detail the characteristics and wake-up means in several low-power modes:

Table 5-2 Low-power modes

Mode	Feature	Entry condition	Wakeup event	Test condition	Power consumption ⁽¹⁾
Idle	The peripherals are powered normally, the core stops running, the clock system is running, but the clocks of each peripheral can be selected to turn off by the peripheral clock control bit.	Set SLEEPDEEP=0, execute _WFI() or _WFE() after setting the wake-up conditions.	I/O or RTC or BAT or I2C or SysTick or SPI or TMR or UART	Main frequency HSE=16MHz, FLASH chip select disabled, peripheral clock disabled	1.5mA
				Main frequency HSI=16MHz, FLASH chip select disabled, peripheral clock disabled	1.4mA
Halt	The peripherals are powered normally, the core stops running, the clock system stops (PLL/HSE).	Set SLEEPDEEP=1, execute _WFI() or _WFE() after setting the wake-up conditions.	I/O or RTC or BAT	Main frequency HSE=16MHz, XT_FORCE_EN=1	1.02mA
				Main frequency HSI=16MHz, XT_FORCE_EN	0.83mA

				=1	
				XT_FORCE_EN =0	180uA
Sleep	The main LDO is off, and the ultra-low-power ULP-LDO maintains power supply of PMU, core and basic peripherals, you can select whether to turn on LSE or LSI, and to maintain power supply of RAM8K, RAM24K and RF configurations.	Set SLEEPDEEP=1, Configure POWER_PLAN, Execute __WFI() or __WFE() after setting wake-up conditions.	I/O or RTC or BAT. The chip will continue to run after waking up.	See Table 5-3 for details	1.8uA~3.8uA
Shut down	Ultra-low-power consumption LDO maintains power supply of PMU, you can select whether to turn on LSE or LSI, and to maintain power supply of RAM8K and RAM24K for data retention.	Set SLEEPDEEP=1, Configure POWER_PLAN, Execute __WFI() or __WFE() after setting wake-up conditions.	I/O or RTC or BAT. The chip will automatically reset after wake-up.	See Table 5-3 for details	0.85uA~2.6uA

The following table describes the detailed configurations of several low-power modes:

Table 5-3 Detailed configuration example of low-power mode

Planning configuration	SYS_EN	RAM2K	RAM24K	CK32K	CORE	EXTEND	Power consumption ⁽¹⁾ (for reference only)
Maintaining supply function	System power	Data area 8KB	Data area 24KB	LSE/LSI RTC wake-up	CPU core and basic peripherals	RF configurations	PMU and RTC registers are always powered
Common configurations in shutdown mode	0	0	0	0	0	0	0.85uA
	0	1	0	0	0	0	1.6uA
	0	0	1	0	0	0	2.6uA
	0	0	0	1	0	0	1.1uA (LSI) 1.2uA (LSE)
	0	1	0	1	0	0	1.8uA (LSI) 1.9uA (LSE)
Common configurations in sleep mode	0	1	0	0	1	0	1.8uA
	0	1	0	1	1	0	2.1uA (LSI) 2.2uA (LSE)
	0	0	1	0	1	0	2.8uA
	0	0	1	1	1	0	3.1uA (LSI) 3.2uA (LSE)
	0	1	1	1	1	1	3.7uA (LSI) 3.8uA (LSE)

Note: 1. Current parameters are measured at room temperature, note that the temperature changes lead to current changes;

Chapter 6 System Clock and RTC

6.1 Introduction to System Clock

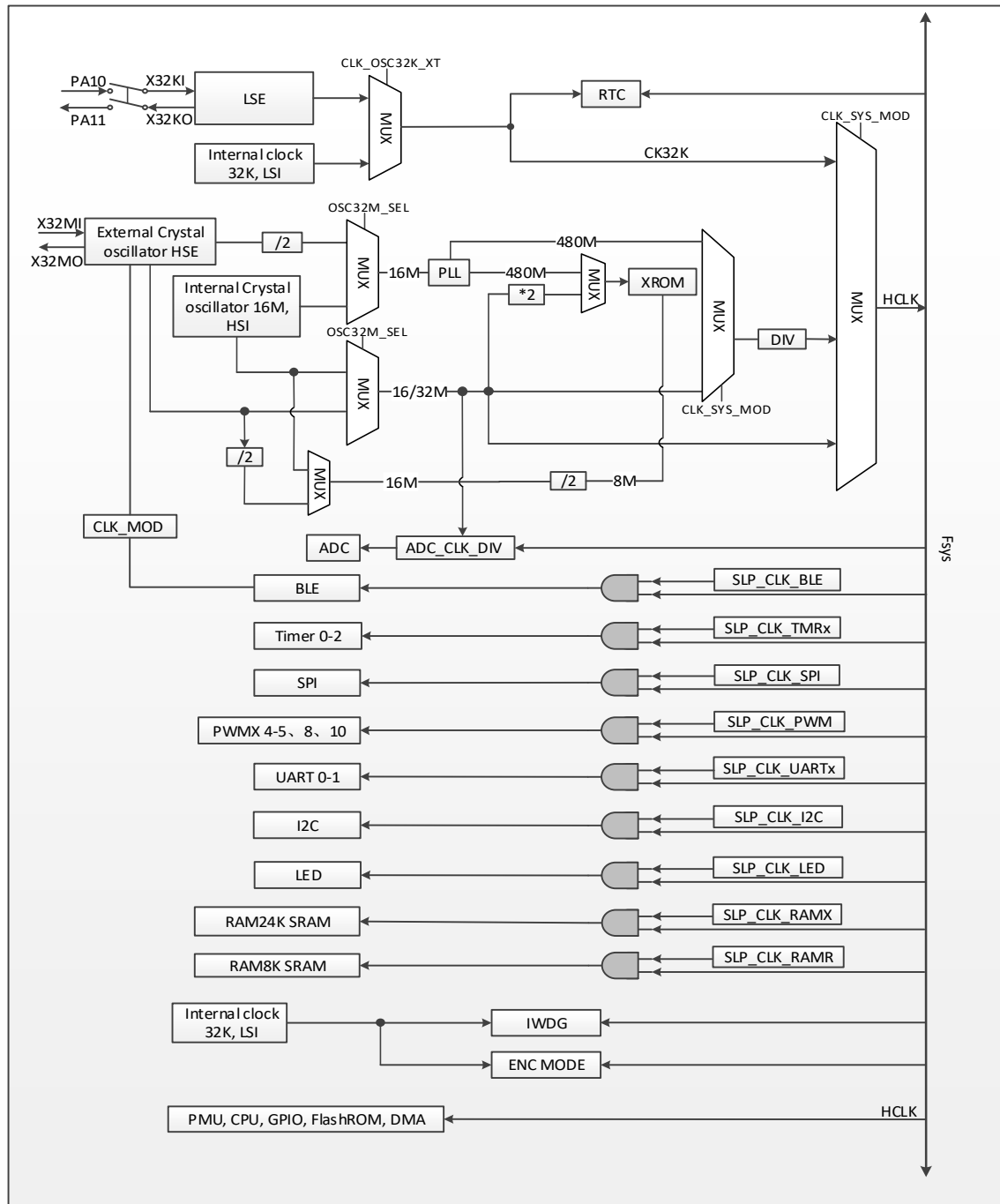
The following different clock sources can be selected to drive the system clock HCLK (Fsys)

- Frequency division of HSE.
- Internal PLL (480MHz by default) frequency division.
- HSE primary clock CK32M or HSI primary clock CK16M
- LSE or LSI original clock CK32K.

Any clock source can be turned on/off independently, thereby optimizing system power consumption.

6.1.1 Clock Architecture

Figure 6-1 Clock tree block diagram



The figure above is the internal clock tree architecture of the system. The 32KHz clock source (CK32K) is selected for RTC function, so the low-frequency clock must be turned on when using these functions.

6.2 Introduction to RTC

The Real-time Clock (RTC) is an independent timer that contains a set of counters that can count continuously. Under the corresponding software configuration, a simple calendar function is available. Reset the current time and date by modifying the value of counter.

The RTC register is powered as often as the PMU. After the system is reset or woken up from low-power mode, RTC setting and time remain unchanged.

6.2.1 Main Features

- 2 modes that can be configured:
 - Timing mode: A fixed cycle time (timing) can be selected for the software to generate interrupt notifications.
 - Trigger mode: Match a target alarm clock time preset by the software to generate an interrupt notification.
- 3 groups of 16-bit counters that provide count of CK32K primitive cycle, 2s cycle and 1-day cycle.

6.3 Register Description

Table 6-1 Clock and oscillator control registers

Name	Access address	Description	Reset value
R32_CLK_SYS_CFG	0x40001008	System clock configuration register	0x0803
R8_HFCK_PWR_CTRL	0x4000100A	High frequency clock module power control register	0x16
R16_INT32K_TUNE	0x4000102C	Internal 32KHz clock tune register	0x1011
R8_XT32K_TUNE	0x4000102E	32KHz clock resonance tune register	0xC7
R8_CK32K_CONFIG	0x4000102F	32KHz oscillator configuration register	0xX2
R8_XT32M_TUNE	0x4000104E	External 32MHz clock resonance tune register	0x03
R16_OSC_CAL_CNT	0x40001050	Oscillator frequency calibration count register	0xFFFF
R8_OSC_CAL_OV_CNT	0x40001052	Oscillator frequency calibration overflow count register	0x00
R8_OSC_CAL_CTRL	0x40001053	Oscillator frequency calibration control register	0x09
R8_PLL_CONFIG	0x4000104B	PLL configuration register	0x0C
R8_RTC_FLAG_CTRL	0x40001030	RTC flag and control register	0x30
R8_RTC_MODE_CTRL	0x40001031	RTC mode control register	0xC2
R32_RTC_TRIG	0x40001034	RTC trigger value register	0x00000000
R16_RTC_CNT_32K	0x40001038	RTC based 32768Hz count value register	0xFFFF
R16_RTC_CNT_2S	0x4000103A	RTC count value register in the unit of 2S	0xFFFF
R32_RTC_CNT_DAY	0x4000103C	RTC count value register in the unit of day	0x0000FFFF

System Clock Configuration Register (R16_CLK_SYS_CFG)

Bit	Name	Access	Description	Reset value
[15:14]	Reserved	RO	Reserved	0
13	RB_PLL_GATE_TIME	RWA	When switching the PLL clock source, the time to turn off the PLL clock is selected: 1: 40us; 0: 30us.	0
12	RB_PLL_GATE_DIS	RWA	When switching the PLL clock source, should the PLL clock be disabled: 1: No;	0

			0: Yes.	
11	RB_PLL_GATE_ST ATUS	RO	PLL clock enable indicator: 1: PLL clock is enabled; 0: PLL clock is disabled.	1
10	Reserved	RO	Reserved	0
9	RB_OSC32M_SEL	RWA	PLL clock source selection: 1: External 32MHz oscillator HSE; 0: Internal 16MHz oscillator HSI. <i>Note: Switching completion requires 1μs.</i>	0
8	Reserved	RO	Reserved	0
[7:6]	RB_CLK_SYS_MOD	RWA	HCLK system clock source mode selection: 00/10: CK32M (32MHz by default) for frequency division; 01: PLL (480MHz by default) for frequency division; 11: CK32K (32KHz by default), used as HCLK.	0
5	Reserved	RO	Reserved	0
[4:0]	RB_CLK_PLL_DIV	RWA	HCLK output clock frequency division factor, the minimum value is 2. 0 means the maximum value 32. Write 1 to disable HCLK.	00011b

Calculation:

$F_{ck32m} = XT_32MHz;$

$F_{ck16m} = RC_16MHz;$

$F_{ck32k} = RB_CLK_OSC32K_XT ? XT_32KHz : RC_32KHz;$

$F_{pll} = F_{ck32m} * 15 = 480MHz;$

$F_{sys} = RB_CLK_SYS_MOD[1] ? (RB_CLK_SYS_MOD[0] ? F_{ck32k} : (RB_OSC32M_SEL ? F_{ck32m} : F_{ck16m})) : ((RB_CLK_SYS_MOD[0] ? F_{pll} / RB_CLK_PLL_DIV : (RB_OSC32M_SEL ? F_{ck32m} : F_{ck16m})) / RB_CLK_PLL_DIV);$

Power-on default $F_{sys} = F_{ck16m} / RB_CLK_PLL_DIV = 16MHz / 3 = 5.33MHz;$

F_{sys} range:

Bit RB_CLK_SYS_MOD[1:0]	Clock source mode of HCLK system	F_{sys} range
11	CK32K	32KHz (run in RAM)
00/10	CK32M or CK16M divide frequency/ CK32M or CK16M	1MHz~32MHz
01	PLL divide frequency	15MHz~80MHz

High Frequency Clock Module Power Control Register (R8_HFCK_PWR_CTRL)

Bit	Name	Access	Description	Reset value
[7:5]	Reserved	RO	Reserved	0
4	RB_CLK_PLL_PON	RWA	PLL power control bit: 1: Power-on; 0: Power-down.	1

3	RB_CLK_XT32M_KEEP	RWA	For controlling the halt of the clock system in suspend mode: 1: HSE and PLL do not automatically halt in Halt mode; 0: HSE and PLL automatically halt in Halt mode.	0
2	RB_CLK_XT32M_PON	RWA	External 32MHz oscillator HSE power control bit: 1: Power-on; 0: Power-down.	1
1	RB_CLK_RC16M_PON	RWA	Enable for internal 16MHz oscillator HSI: 1: On; 0: Off.	1
0	Reserved	RO	Reserved	0

Internal 32KHz Clock Calibration Register (R16_INT32K_TUNE)

Bit	Name	Access	Description	Reset value
[15:13]	Reserved	RO	Reserved	0
[12:0]	RB_INT32K_TUNE	RWA	Internal RC 32KHz clock frequency calibration value.	1011h

32KHz Clock Resonance Tune Register (R8_XT32K_TUNE)

Bit	Name	Access	Description	Reset value
[7:4]	RB_XT32K_C_LOAD	RWA	Select the built-in load capacitor matching the external 32KHz crystal (which may affect the RTC clock accuracy). Capacitance = RB_XT32K_C_LOAD + 12pF. 0000b to 1111b correspond to approximately 12pF to 27pF, respectively. Select according to the crystal parameters used.	1100b
[3:2]	RB_RC32K_I_TUNE	RWA	RC32K adjustment bit, internal 32KHz oscillator bias current selection: 00: 120% rated current; 01: 160% rated current; 10: 230% rated current; 11: Rated current. <i>Note: Default setting is 11.</i>	01b
[1:0]	RB_XT32K_I_TUNE	RWA	32KHz oscillator bias current select: 00: 70% of rated current; 01: Rated current; 10: 140% of rated current; 11: 200% of rated current. <i>Note: The current can be changed to the rated current after the crystal oscillator is stable.</i>	11b

32KHz Oscillator Configuration Register (R8_CK32K_CONFIG)

Bit	Name	Access	Description	Reset value
7	RB_32K_CLK_PIN	RO	32KHz clock pin status (asynchronous signal).	X
[6:4]	Reserved	RO	Reserved	0
3	RB_CLK_OSC32K_FILT	RWA	Internal 32KHz oscillator noise filter mode: 1: Disable; 0: Enable.	0
2	RB_CLK_OSC32K_XT	RWA	CK32K (32KHz) clock source select bit: 1: External 32KHz oscillator; 0: Internal 32KHz oscillator.	0
1	RB_CLK_INT32K_PON	RWA	Internal 32KHz oscillator power control bit: 1: Power on; 0: Power off.	1
0	RB_CLK_XT32K_PON	RWA	External 32KHz oscillator power control bit: 1: Power on; 0: Power off.	0

External 32MHz Clock Resonance Control Register (R8_XT32M_TUNE)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved	0
[6:4]	RB_XT32M_C_LOAD	RWA	Select the built-in load capacitor that matches the external 32MHz crystal (Which may affect wireless communication): Capacity=RB_XT32M_C_LOAD*2+6pF, 000b~111b correspond to approximately 6pF~20pF respectively. Select according to the parameters of crystal used; the common value is 111b.	0
[3:2]	Reserved	RO	Reserved	0
[1:0]	RB_XT32M_I_BIAS	RWA	External 32MHz oscillator bias current select: 00: 75% of rated current; 01: Rated current; 10: 125% of rated current; 11: 150% of rated current.	11b

Oscillator Frequency Calibration Count Register (R16_OSC_CAL_CNT)

Bit	Name	Access	Description	Reset value
15	RB_OSC_CAL_IF	RW1	Oscillator capture complete interrupt flag bit, write 1 to clear it: 1: Interrupt; 0: No interrupt.	0
14	RB_OSC_CAL_OV_CLR	RW1	R8_OSC_CAL_OV_CNT register value non-zero indicator, write 1 to clear R8_OSC_CAL_OV_CNT.	0
[13:0]	RB_OSC_CAL_CNT	RO	Count value based on system clock frequency of multiple CK32K cycles, used to calibrate internal 32KHz oscillator frequency.	XXXXh

Oscillator Frequency Calibration Overflow Count Register (R8_OSC_CAL_OV_CNT)

Bit	Name	Access	Description	Reset value
[7:0]	RB_OSC_CAL_OV_CNT	RO	Oscillator frequency calibration count overflow times, write 1 to RB_OSC_CAL_OV_CLR to clear this register.	0

Oscillator Frequency Calibration Control Register (R8_OSC_CAL_CTRL)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved	0
6	RB_OSC_CNT_END	RWA	Oscillator capture end-point selection: 1: 2 additional cycles; 0: No.	0
5	RB_OSC_CNT_EN	RWA	Oscillator frequency calibration counter enable: 1: Enable counting; 0: Disable counting.	0
4	RB_OSC_CAL_IE	RWA	Oscillator capture complete interrupt enable: 1: Enable; 0: Disable.	0
3	RB_OSC_CNT_HALT	RO	Oscillator frequency calibration counter count status: 1: Counting is being paused; 0: Counting is in progress.	1
[2:0]	RB_OSC_CNT_TOTAL	RWA	Oscillator capture total cycle selection: 000: 1 cycle; 001: 2 cycles; 010: 4 cycles; 011: 32 cycles; 100: 64 cycles; 101: 128 cycles; 110: 1024 cycles; 111: 2047 cycles.	001b

PLL Configuration Register (R8_PLL_CONFIG)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved	0
[6:0]	RB_PLL_CFG_DAT	RWA	PLL configuration parameters.	0Ch

RTC Flag and Control Register (R8_RTC_FLAG_CTRL)

Bit	Name	Access	Description	Reset value
7	RB_RTC_TRIG_FLAG	RO	RTC trigger mode activation flag.	0
6	RB_RTC_TMR_FLAG	RO	RTC timing mode activation flag.	0
5	RB_RTC_TRIG_CLR	RW	This bit is always 1 when the trigger mode is disabled. When the trigger mode is enabled, write 1, clear the trigger mode activation flag RB_RTC_TRIG_FLAG and automatically cleared to 0.	1
4	RB_RTC_TMR_CLR	RW	When the timing mode is disabled, this bit is fixed as 1.	1

			When the timing mode is enabled, write 1, clear the timing mode activation flag RB_RTC_TMR_FLAG and automatically cleared to 0.	
[3:0]	Reserved	RO	Reserved	0

RTC Mode Control Register (R8_RTC_MODE_CTRL)

Bit	Name	Access	Description	Reset value
7	RB_RTC_LOAD_HI	RWA	Write 1 to load the high word of RTC counter, and automatically cleared to 0 after loading. Load R32_RTC_TRIG (actually only the low 14 bits) to R32_RTC_CNT_DAY.	1
6	RB_RTC_LOAD_LO	RWA	Write 1 to load the low word of RTC counter, and automatically cleared after loading. Load the high 16 bits of R32_RTC_TRIG to R16_RTC_CNT_2S; load the low 16 bits of R32_RTC_TRIG to R16_RTC_CNT_32K.	1
5	RB_RTC_TRIG_EN	RWA	RTC trigger mode enable: 1: Enable; 0: Disable.	0
4	RB_RTC_TMR_EN	RWA	RTC timing mode enable: 1: Enable; 0: Disable.	0
3	RB_RTC_IGNORE_B0	RWA	Ignore and compare the lowest bit of matching value in trigger mode: 1: Ignore the lowest bit; 0: Compare the lowest bit.	0
[2:0]	RB_RTC_TMR_MODE	RWA	RTC timing mode fixed cycle (timing) selection: 000: 0.125S; 001: 0.25S; 010: 0.5S; 011: 1S; 100: 2S; 101: 4S; 110: 8S; 111: 16S.	010b

RTC Trigger Value Register (R32_RTC_TRIG)

Bit	Name	Access	Description	Reset value
[31:0]	R32_RTC_TRIG	RWA	The preset matching value in RTC trigger mode, and the high 16 bits and low 16 bits are matched with R16_RTC_CNT_2S and R16_RTC_CNT_32K respectively. Cooperate with RB_RTC_LOAD_LO and RB_RTC_LOAD_HI to update the current value of RTC counter.	0

Note: The preset matching value is not directly written into the target time, and it involves simple calculations. Please refer to the following instructions.

RTC Count Value Register Based on 32768Hz (R16_RTC_CNT_32K)

Bit	Name	Access	Description	Reset value
[15:0]	R16_RTC_CNT_32K	RO	RTC count value register based on 32768Hz.	XXXXh

RTC Count Value Register in the Unit of 2S (R16_RTC_CNT_2S)

Bit	Name	Access	Description	Reset value
[15:0]	R16_RTC_CNT_2S	RO	The current count value of the RTC in 2S units.	XXXXh

RTC Count Value Register in the Unit of Day (R32_RTC_CNT_DAY)

Bit	Name	Access	Description	Reset value
[31:14]	Reserved	RO	Reserved	0
[13:0]	R32_RTC_CNT_DAY	RO	RTC current count value in the unit of day.	XXXXb

6.4 Functional Description and Configuration

6.4.1 RTC Counter Initialization

- (1) Set the value of the R32_RTC_TRIG register and set RB_RTC_LOAD_HI, to load the value of the R32_RTC_TRIG register into the R32_RTC_CNT_DAY register;
- (2) Set the value of the R32_RTC_TRIG register and set RB_RTC_LOAD_LO, to load the value of the high 16 bits and the low 16 bits of the R32_RTC_TRIG register into the R16_RTC_CNT_2S register and the R16_RTC_CNT_32K register respectively.

6.4.2 Switch RTC Clock Source to LSE Crystal

- (1) Confirm that the GPIO pins where X32KI and X32KO are located are not set as outputs, with no pull-up and pull-down resistors, and only crystals are provided;
- (2) Configure the R8_CK32K_CONFIG register, set RB_CLK_XT32K_PON to 1, to enable the external 32KHz crystal oscillator;
- (3) It is recommended to set RB_XT32K_I_TUNE to the maximum first, and wait for the crystal oscillator to stabilize (about several hundreds of mS) and then change to the rated current;
- (4) Configure the R8_CK32K_CONFIG register, set RB_CLK_OSC32K_XT to 1, and switch clock source to the crystal oscillator;
- (5) Wait for at least half of the 32KHz clock cycle, usually 16μs, to actually finish the switch of clock source.

6.4.3 RTC Timing Function

- (1) Configure R8_RTC_MODE_CTRL register, set RB_RTC_TMR_MODE to select the appropriate timing period, set RB_RTC_TMR_EN to 1, and turn on RTC timing function;
- (2) After reaching the timing period, RTC timing activation flag RB_RTC_TMR_FLAG and interrupt will be generated; check R8_RTC_FLAG_CTRL register and set RB_RTC_TMR_CLR to clear the flag.

6.4.4 RTC Trigger Function

- (1) Set the target matching value in R32_RTC_TRIG register, and see the calculation and operation steps:
Calculate the target time value by taking the current time R32_RTC_CNT_32K (high 16 bits R16_RTC_CNT_2S and low 16 bits R16_RTC_CNT_32K) plus the interval time DelayTime (in the unit of S), $T32 = R32_RTC_CNT_32K + DelayTime * 32768$,
Write T32 into the R32_RTC_TRIG register to complete the matching value setting;
- (2) Configure R8_RTC_MODE_CTRL, set RB_RTC_TRIG_EN to 1, and turn on RTC trigger function;
- (3) When the current RTC count values R16_RTC_CNT_2S and R16_RTC_CNT_32K respectively match the preset high and low 16 bits of R32_RTC_TRIG, RTC trigger activation flag RB_RTC_TRIG_FLAG and interrupt are generated, and the flag can be cleared by setting RB_RTC_TRIG_CLR.

Please refer to the evaluation board example program for details.

6.4.5 Calibrate Internal 32K Clock LSI with HSE

Refer to evaluation board example program.

Chapter 7 General-purpose I/O and Alternate Functions

7.1 Introduction to GPIO

The chip provides 2 sets of GPIO ports PA and PB, with a total of 15 general-purpose input and output pins, 15 of which have interrupt and wake-up functions, and some of which have alternate and mapping functions.

Each GPIO port has a 32-bit direction configuration register R32_Px_DIR, a 32-bit pin input register R32_Px_PIN, a 32-bit data output register R32_Px_OUT, a 32-bit data reset register R32_Px_CLR, a 32-bit pull-up resistor configuration register R32_Px_PU, a 32 pull-down resistor/driver capability configuration register R32_Px_PD_DRV.

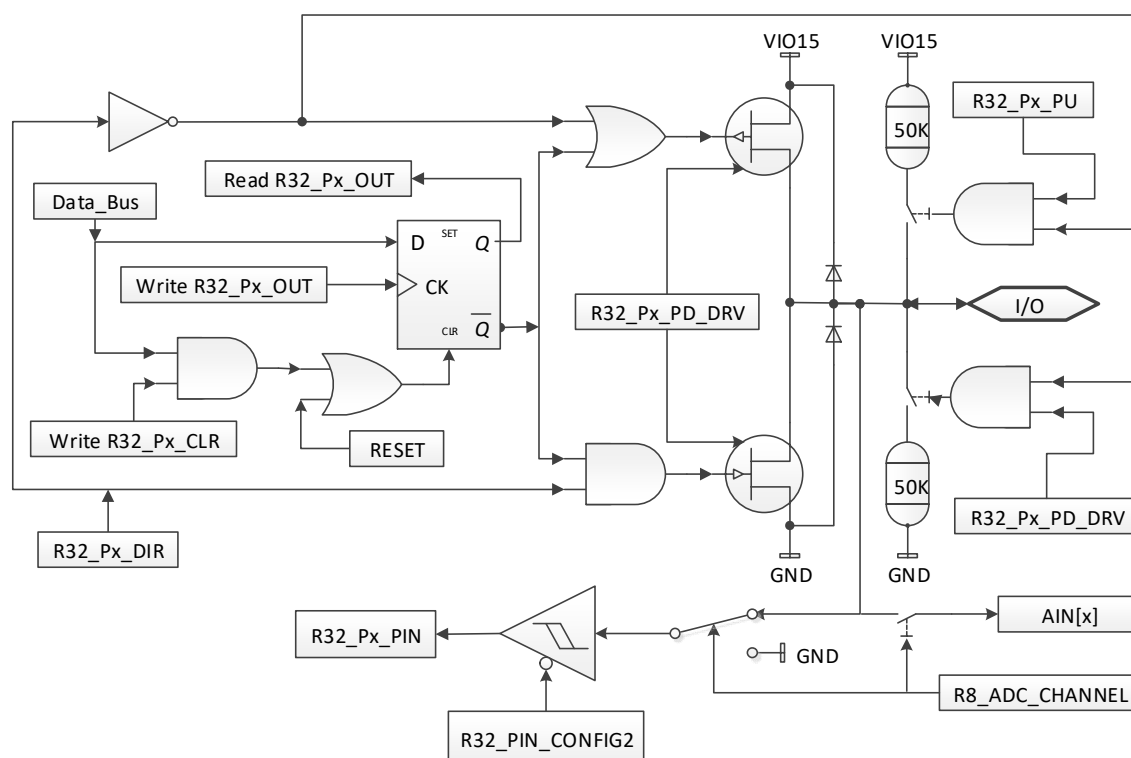
The PA[8]~A[15] bits in the PA port are valid, corresponding to the 8 GPIO pins on the chip.

The PB[6], PB[10]~PB[15] bits are active in the PB port, corresponding to 7 GPIO pins on the chip.

Each I/O port bit can be freely programmed, but the I/O port register must be accessed by 8-bit, 16-bit or 32-bit words. If the alternate function of pin is not enabled, it will be used as a general-purpose I/O port by default.

The following figure is the block diagram of GPIO internal architecture:

Figure 7-1 I/O internal architecture block diagram



7.2 External Interrupt/Wakeup

Parts of I/O pins of the chip have interrupt function and can realize sleep and wakeup.

In order to use external interrupts, the port bits must be configured in input mode. And provide 4 kinds of trigger modes: high level, low level, rising edge, falling edge.

The wake-up function needs to enable the interrupt R16_Px_INT_EN of the port bit, and turn on the GPIO wake-up control bit RB_SLP_GPIO_WAKE in the R8_SLP_WAKE_CTRL register.

7.3 GPIO Alternate and Remapping

7.3.1 Alternate Functions

Some I/O pins have alternate functions. After power on, all I/O pins are as GPIO by default. After enabling various functional modules, the corresponding original GPIO pins are configured as corresponding functional pins of each functional module.

If a pin has multiple alternate functions, and multiple functions are enabled, please refer to the function order in the "Alternate Function" list in the pin description in section 1.2 for the priority order of alternate functions.

For example: If the PA9 pin is alternate as TMR0/TXD1/AIN13, TMR0 function has a high priority, and AIN13 function has the lowest priority. In this way, the alternate functions with the relatively higher priority of the pin whose functions with the lowest priority need not to be used can be enabled among multiple alternate functions.

The following tables list some GPIO configurations of some functional pins which are used for peripheral modules.

Table 7-1 Timer x

TMR0/1/2 pin	Functional configuration	GPIO configuration
TMRx	Input capture channel x	Input (floating input/pull-up input/pull-down input)
	Output PWM channel x	Push-pull output

Table 7-2 UARTx

UART0/1/2 pin	Functional configuration	GPIO configuration
TXDx	UART transmit x	Push-pull output
RXDx	UART receive x	Pull-up input (recommended) or floating input
DTR	MODEM signal output or RS485 control	Push-pull output
DSR/RI/DCD	MODEM signal input	Pull-up input (recommended) or floating input

Table 7-3 SPI

SPI pin	Functional configuration	GPIO configuration
SCKx	Clock output in master mode	Push-pull output
	Clock input in slave mode	Input (floating input/pull-up input/pull-down input)
MOSIx	Full-duplex mode-master mode	Push-pull output
	Full-duplex mode-slave mode	Input (floating input/pull-up input/pull-down input)
	Half-duplex mode-master mode	Not used, can be used as general purpose I/O
	Half-duplex mode-slave mode	Not used, can be used as general purpose I/O
MISOx	Full-duplex mode-master mode	Input (floating input/pull-up input/pull-down input)

	Full duplex mode-slave mode	Input (pull-up is recommended, automatically switched to push-pull output after chip select) or push-pull output (it is forbidden to be used for bus connection)
	Half duplex mode-master mode	Input or push-pull output, manual switching
	Half duplex mode-slave mode	Input (pull-up is recommended, automatically switched to push-pull output after chip select)
SCS	Chip select output in master mode	Push-pull output (can be replaced with other pins)
	Chip select input in slave mode	Pull-up input (recommended) or floating input

Table 7-4 I2C

I2C pin	Functional configuration	GPIO configuration
SCL	Serial clock output – master mode	Push-pull output (multi-master is not supported in this mode)
	Serial clock output/input – multi-master mode	Input (pull-up is recommended, automatically open-drain output when needed)
	Serial clock input – slave mode	Pull-up input (recommended) or floating input
SDA	Serial data input/output	Input (pull-up is recommended, automatically open-drain output when needed)

Table 7-5 ADC

ADC sample channel pin	Functional configuration	GPIO configuration
AINx	ADC input channel	Only analog input

Table 7-6 LED

LED signal pin	Functional configuration	GPIO configuration
LED[3:0]	LED output bit3~bit0	Push-pull output
LED_CLK	LED output clock	Push-pull output

Table 7-7 ENCODER

ENCODER signal pin	Functional configuration	GPIO configuration
ENCCAP_IN1	Encoder capture channel 1	Input (Floating input/pull-up input/pull-down input)
ENCCAP_IN2	Encoder capture channel 2	Input (Floating input/pull-up input/pull-down input)

7.3.2 Remapping of Function Pins

To optimize the simultaneous utilization rate of peripheral functions, certain functional pins can be remapped to other pins by configuring the R16_PIN_ALTERNATE function pin remapping register and the R16_PIN_ANALOG_IE peripheral analog pin configuration register.

Table 7-9 Alternate function remapping pins

Peripheral function pins	Default GPIO pins	Remapped GPIO pins
SCS/SCK/MOSI/MISO	PA12/PA13/PA14/PA15	PB12/PB13/PB14/PB15
RXD1/TXD1	PA8/PA9	PB12/PB13
RXD0/TXD0/DSR/DTR/RI/DCD	-/-/PB14/PB15/-/-	PA15/PA14/-/-/PB12/PB13
TMR2/PWM2/CAP2	PA11	PB11
TMR1/PWM1/CAP1	PA10	PB10
TMR0/PWM0/CAP0	PA9	-
PWM4/PWM5	PA12/PA13	-
PWM8/PWM10	PB6/PB14	-
SCL/SDA (I2C)	PB15/PB14	-
LED[3]/LED[2]/LED[1]/LED[0]/LED_CLK	PA11/PA12/PA13/PA14/PA15	PB11/PB12/PB13/PB14/PB15
ENCCAP_IN1/ENCCAP_IN2	PB10/PB11	PA10/PA11

7.4 Register Description

Table 7-10 GPIO registers

Name	Access address	Description	Reset value
R32_PIN_CONFIG1	0x40001018	Pin configuration register 1	0x00000000
R16_PIN_ALTERNATE	0x40001018	Functional pin remapping register	0x0000
R16_PIN_ANALOG_IE	0x4000101A	Peripheral analog pin configuration register	0x0000
R32_PIN_CONFIG2	0x4000101C	Pin configuration register 2	0x00000000
R16_PA_INT_EN	0x40001090	PA port interrupt enable register	0x0000
R16_PB_INT_EN	0x40001092	PB port interrupt enable register	0x0000
R16_PA_INT_MODE	0x40001094	PA port interrupt mode configuration register	0x0000
R16_PB_INT_MODE	0x40001096	PB port interrupt mode configuration register	0x0000
R16_PA_INT_EDGE_TYPE	0x40001098	PA port interrupt edge type configuration register	0x0000
R16_PB_INT_EDGE_TYPE	0x4000109A	PB port interrupt edge type configuration register	0x0000
R16_PA_INT_IF	0x4000109C	PA port interrupt flag register	0x0000
R16_PB_INT_IF	0x4000109E	PB port interrupt flag register	0x0000
R32_PA_DIR	0x400010A0	PA port direction configuration register	0x00000000
R32_PA_PIN	0x400010A4	PA port pin input register	0x0000XXXX
R32_PA_OUT	0x400010A8	PA port data output register	0x00000000
R32_PA_CLR	0x400010AC	PA port data reset register	0x00000000
R32_PA_PU	0x400010B0	PA port pull-up resistor configuration register	0x00000000
R32_PA_PD_DRV	0x400010B4	PA port pull-down/drive configuration register	0x00000000
R32_PA_SET	0x400010B8	PA port output setting register	0x00000000

R32_PB_DIR	0x400010C0	PB port direction configuration register	0x00000000
R32_PB_PIN	0x400010C4	PB port pin input register	0x00XXXXXX
R32_PB_OUT	0x400010C8	PB port data output register	0x00000000
R32_PB_CLR	0x400010CC	PB port data reset register	0x00000000
R32_PB_PU	0x400010D0	PB port pull-up resistor configuration register	0x00000000
R32_PB_PD_DRV	0x400010D4	PB port pull-down/drive configuration register	0x00000000
R32_PB_SET	0x400010D8	PB port output setting register	0x00000000

Functional Pin Remapping Register (R16_PIN_ALTERNATE)

Bit	Name	Access	Description	Reset value
[15:13]	Reserved	RO	Reserved	0
12	RB_PIN_MODEM	RW	UART0 MODEM functional pin mapping select bit: 1: R1/DCD is mapped to PB[12]/PB[13]; 0: Reserved.	0
[11:9]	Reserved	RO	Reserved	0
8	RB_PIN_SPI0	RW	SPI0 functional pin mapping select bit: 1: SCK0_/SCS_/MOSI_/MISO_ is mapped to PB[13]/PB[12]/PB[14]/PB[15]; 0: SCK0/SCS/MOSI/MISO is mapped to PA[13]/PA[12]/PA[14]/PA[15].	0
[7:6]	Reserved	RO	Reserved	0
5	RB_PIN_UART1	RW	UART1 functional pin mapping select bit: 1: RXD1_/TXD1_ is mapped to PB[12]/PB[13]; 0: RXD1/TXD1 is mapped to PA[8]/PA[9].	0
4	RB_PIN_UART0	RW	UART0 functional pin mapping select bit: 1: RXD0_/TXD0_ is mapped to PA[15]/PA[14]; 0: Reserved.	0
3	Reserved	RO	Reserved	0
2	RB_PIN_TMR2	RW	TMR2 functional pin mapping select bit: 1: TMR2_/PWM2_/CAP2_ is mapped to PB[11]; 0: TMR2/PWM2/CAP2 is mapped to PA[11].	0
1	RB_PIN_TMR1	RW	TMR1 functional pin mapping select bit: 1 : TMR1_/PWM1_/CAP1_ is mapped to PB[10]; 0: TMR1/PWM1/CAP1 is mapped to PA[10].	0
0	Reserved	RO	Reserved	0

Peripheral Analog Pin Configuration Register (R16_PIN_ANALOG_IE)

Bit	Name	Access	Description	Reset value
[15:6]	Reserved	RO	Reserved	00h
5	RB_PIN_LED	RW	LED pin mapping: 1: LED[3]/LED[2]/LED[1]/LED[0]/LED_CLK	0

			mapped to PB[11]/PB[12]/PB[13]/PB[14]/PB[15]; 0: LED[3]/LED[2]/LED[1]/LED[0]/LED_CLK mapped to PA[11]/PA[12]/PA[13]/PA[14]/PA[15].	
4	RB_PIN_ENC	RW	Encoder function pin mapping selection bit: 1: ENCCAP_IN1/ENCCAP_IN2 mapped to PA[10]/PA[11]; 0: CAP_IN1/CAP_IN2 mapped to PB[10]/PB[11].	0
[3:0]	Reserved	R0	Reserved	0

Note: If the pin is used for analog function, it is recommended to turn off the digital input function of the pin, namely to set the digital input as disabled, thus reducing power consumption and helping reduce interference.

Pin Configuration Register 2 (R32_PIN_CONFIG2)

Bit	Name	Access	Description	Reset value
[31:26]	RB_PIN_PB_DIS[10:15]	RW	PB10-PB15 channel pins digital input enable: 1: Disable the digital input to save power; 0: Enable the digital input.	0
[25:23]	Reserved	RO	Reserved	0
22	RB_PIN_PB_DIS[6]	RW	PB6 channel pins digital input enable: 1: Disable the digital input to save power; 0: Enable the digital input.	0
[21:16]	Reserved	RO	Reserved	0
[15:8]	RB_PIN_PA_DIS[8:15]	RW	PA8-PA15 channel pins digital input enable: 1: Disable the digital input to save power; 0: Enable the digital input.	0
[7:0]	Reserved	RO	Reserved	0

PA Port Interrupt Enable Register (R16_PA_INT_EN)

Bit	Name	Access	Description	Reset value
[15:0]	R16_PA_INT_EN	RW	PA pin interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0

PB Port Interrupt Enable Register (R16_PB_INT_EN)

Bit	Name	Access	Description	Reset value
[15:0]	R16_PB_INT_EN	RW	PB pin interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0

PA Port Interrupt Mode Configuration Register (R16_PA_INT_MODE)

Bit	Name	Access	Description	Reset value
[15:0]	R16_PA_INT_MODE	RW	PA pin interrupt mode select bit: 1: Edge trigger; 0: Level trigger.	0

PB Port Interrupt Mode Configuration Register (R16_PB_INT_MODE)

Bit	Name	Access	Description	Reset value
[15:0]	R16_PB_INT_MODE	RW	PB pin interrupt mode select bit: 1: Edge trigger; 0: Level trigger.	0

PA Port Interrupt Edge Type Configuration Register (R16_PA_INT_EDGE_TYPE)

Bit	Name	Access	Description	Reset value
[15:0]	R16_PA_INT_EDGE_TYPE	RW	PA pin interrupt edge type configuration: 1: Non-polarity sensitive; 0: Polarity sensitive.	0

PB Port Interrupt Edge Type Configuration Register (R16_PB_INT_EDGE_TYPE)

Bit	Name	Access	Description	Reset value
[15:0]	R16_PB_INT_EDGE_TYPE	RW	PB pin interrupt edge type configuration: 1: Non-polarity sensitive; 0: Polarity sensitive.	0

PA Port Interrupt Flag Register (R16_PA_INT_IF)

Bit	Name	Access	Description	Reset value
[15:0]	R16_PA_INT_IF	RW1	PA pin interrupt flag bit, write 1 to clear: 1: Interrupt; 0: No interrupt.	0

PB Port Interrupt Flag Register (R16_PB_INT_IF)

Bit	Name	Access	Description	Reset value
[15:0]	R16_PB_INT_IF	RW1	PB pin interrupt flag bit, write 1 to clear: 1: Interrupt; 0: No interrupt.	0

PA Port Direction Configuration Register (R32_PA_DIR)

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:8]	R8_PA_DIR_1	RW	Current input/output direction configure of PA pin: 1: The pin is in output mode; 0: The pin is in input mode.	0
[7:0]	R8_PA_DIR_0	RW		0

PA Port Pin Input Register (R32_PA_PIN)

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:8]	R8_PA_PIN_1	RO	Current level status of PA pin (valid only when R32_PA_DIR corresponding bit is 0): 1: Pin input is at high level; 0: Pin input is at low level.	XXh
[7:0]	R8_PA_PIN_0	RO		XXh

PA Port Data Output Register (R32_PA_OUT)

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:8]	R8_PA_OUT_1	RW	When the corresponding bit of direction register R32_PA_DIR is 1: Control PA pin output level status: 1: Output high level; 0: Output low level.	0
[7:0]	R8_PA_OUT_0	RW	When the corresponding bit of direction register R32_PA_DIR is 0: Control PA pin interrupt polarity select: 1: High level/rising edge; 0: Low level/falling edge.	0

PA Port Data Reset Register (R32_PA_CLR)

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:8]	R8_PA_CLR_1	WZ	PA data register reset control: 1: The corresponding bit data of R32_PA_OUT is cleared to 0; 0: No effect.	0
[7:0]	R8_PA_CLR_0	WZ		0

PA Port Pull-up Resistor Configuration Register (R32_PA_PU)

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:8]	R8_PA_PU_1	RW	PA pin pull-up resistor enable control: 1: Enable the pull-up resistor; 0: Disable the pull-up resistor.	0
[7:0]	R8_PA_PU_0	RW		0

PA Port Pull-down/Drive Configuration Register (R32_PA_PD_DRV)

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:8]	R8_PA_PD_DRV_1	RW	When the corresponding bit of direction register R32_PA_DIR is 0: PA pin pull-down resistor enable control: 1: Enable the pull-down resistor; 0: Disable the pull-down resistor.	0
[7:0]	R8_PA_PD_DRV_0	RW	When the corresponding bit of direction register R32_PA_DIR is 1: PA pin current drive capability select: 1: 20mA level; 0: 5mA level.	0

PA Port Output Setting Register (R32_PA_SET)

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:8]	R8_PA_SET_1	WZ	When the corresponding bit in the set register R32_PA_SET is 0, the PA pin output is held; when it is 1, the PA pin outputs a high level.	0
[7:0]	R8_PA_SET_0	WZ		0

PB Port Direction Configuration Register (R32_PB_DIR)

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RO	Reserved	0
[23:16]	R8_PB_DIR_2	RW	Current input/output direction configure of PB pin: 1: The pin is in output mode; 0: The pin is in input mode.	0
[15:8]	R8_PB_DIR_1	RW		0
[7:0]	R8_PB_DIR_0	RW		0

PB Port Pin Input Register (R32_PB_PIN)

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RO	Reserved	0
[23:16]	R8_PB_PIN_2	RO	Current level state of PB pin (only when the corresponding bit of R32_PB_DIR is 0, the bit value is valid): 1: Pin input is at high level; 0: Pin input is at low level.	XXh
[15:8]	R8_PB_PIN_1	RO		XXh
[7:0]	R8_PB_PIN_0	RO		XXh

PB Port Data Output Register (R32_PB_OUT)

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RO	Reserved	0
[23:16]	R8_PB_OUT_2	RW	When the corresponding bit of direction register R32_PB_DIR is 1: Control PB pin output level status: 1: Output high level; 0: Output low level.	0
[15:8]	R8_PB_OUT_1	RW	When the corresponding bit of direction register R32_PB_DIR is 1: Control PB pin output level status: 1: Output high level; 0: Output low level.	0
[7:0]	R8_PB_OUT_0	RW	When the corresponding bit of direction register R32_PB_DIR is 0: Control PB pin interrupt polarity select: 1: High level/rising edge; 0: Low level/falling edge.	0

PB Port Data Reset Register (R32_PB_CLR)

Bit	Name	Access	Description	Reset value
-----	------	--------	-------------	-------------

[31:24]	Reserved	RO	Reserved	0
[23:16]	R8_PB_CLR_2	WZ	PB data register reset control:	0
[15:8]	R8_PB_CLR_1	WZ	1: The corresponding bit data of	0
[7:0]	R8_PB_CLR_0	WZ	R32_PB_OUT is cleared to 0; 0: No effect.	0

PB Port Pull-up Resistor Configuration Register (R32_PB_PU)

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RO	Reserved	0
[23:16]	R8_PB_PU_2	RW	PB pin pull-up resistor enable control:	0
[15:8]	R8_PB_PU_1	RW	1: Enable the pull-up resistor;	0
[7:0]	R8_PB_PU_0	RW	0: Disable the pull-up resistor.	0

PB Port Pull-down/Drive Configuration Register (R32_PB_PD_DRV)

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RO	Reserved	0
[23:16]	R8_PB_PD_DRV_2	RW	When the corresponding bit of direction register R32_PB_DIR is 0:	0
[15:8]	R8_PB_PD_DRV_1	RW	PB pin pull-down resistor enable control: 1: Enable the pull-down resistor;	0
[7:0]	R8_PB_PD_DRV_0	RW	0: Disable the pull-down resistor. When the corresponding bit of direction register R32_PB_DIR is 1: PB pin current drive capability select: 1: 20mA level; 0: 5mA level.	0

PB Port Output Setting Register (R32_PB_SET)

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RO	Reserved	0
[23:16]	R8_PB_SET_2	WZ	When the corresponding bit in the set register R32_PB_SET is 0, the PB pin output is held; when it is 1, the PB pin outputs a high level.	0
[15:8]	R8_PB_SET_1	WZ		0
[7:0]	R8_PB_SET_0	WZ		0

7.5 Mode Configuration of GPIO Pins

Each GPIO can be configured to 5 modes, as shown in the table below:

Table 7-11 Port configuration

Mode	R32_Px_DIR	R32_Px_PU	R32_PIN_CONFIG2	R32_Px_PD_DRV
Floating input/high impedance input/analog input	0	0	0	0
Only analog input (Reduce I/O leakage)	0	0	1	0

Input with pull-up resistor	0	1	0	0
Input with pull-down resistor	0	0	0	1
Push-pull output, 5mA level drive capability:	1	X	X	0
Push-pull output, 20mA level drive capability:	1	X	X	1

Chapter 8 General-purpose Timer (TMRx)

8.1 Introduction to TMRx

The chip is equipped with three 26-bit timers, TMR0, TMR1 and TMR2, and the longest timing interval is 2^{26} clock cycles. It is applicable to various occasions, including measuring the length of input signal pulse (input capture) or generating output waveform (PWM). In addition, TMR1 and TMR2 support DMA function. Each timer is completely independent and can be operated simultaneously.

8.1.1 Main Features

- 3 × 26-bit timers, and the longest timing interval is 2^{26} clock cycles.
- Timer interrupt is supported, and among them TMR1 and TMR2 support DMA and interrupt.
- Support capture function to measure input pulse length or cycle.
- The capture function can be set to be capture of level change and hold time of high or low level.
- 26-bit PWM function is supported, which can dynamically adjust the PWM duty cycle setting.

8.2 Introduction to TMRx

Table 8-1 TMR0 registers

Name	Access address	Description	Reset value
R8_TMR0_CTRL_MOD	0x40002000	Mode set register	0x02
R8_TMR0_INTER_EN	0x40002002	Interrupt enable register	0x00
R8_TMR0_INT_FLAG	0x40002006	Interrupt flag register	0x00
R8_TMR0_FIFO_COUNT	0x40002007	FIFO count register	0x0X
R32_TMR0_COUNT	0x40002008	Current count value register	0x0XXXXXXXX
R32_TMR0_CNT_END	0x4000200C	Final count value set register	0x0XXXXXXXX
R32_TMR0_FIFO	0x40002010	FIFO register	0x0XXXXXXXX

Table 8-2 TMR1 registers

Name	Access address	Description	Reset value
R8_TMR1_CTRL_MOD	0x40002400	Mode set register	0x02
R8_TMR1_CTRL_DMA	0x40002401	DMA control register	0x00
R8_TMR1_INTER_EN	0x40002402	Interrupt enable register	0x00
R8_TMR1_INT_FLAG	0x40002406	Interrupt flag register	0x00
R8_TMR1_FIFO_COUNT	0x40002407	FIFO count register	0x0X
R32_TMR1_COUNT	0x40002408	Current count value register	0x0XXXXXXXX
R32_TMR1_CNT_END	0x4000240C	Final count value register	0x0XXXXXXXX
R32_TMR1_FIFO	0x40002410	FIFO register	0x0XXXXXXXX
R16_TMR1_DMA_NOW	0x40002414	Current buffer address of DMA	0xXXXX
R16_TMR1_DMA_BEG	0x40002418	Start buffer address of DMA	0xXXXX
R16_TMR1_DMA_END	0x4000241C	End buffer address of DMA	0xXXXX

Table 8-3 TMR2 registers

Name	Access address	Description	Reset value
R8_TMR2_CTRL_MOD	0x40002800	Mode set register	0x02
R8_TMR2_CTRL_DMA	0x40002801	DMA control register	0x00
R8_TMR2_INTER_EN	0x40002802	Interrupt enable register	0x00
R8_TMR2_INT_FLAG	0x40002806	Interrupt flag register	0x00
R8_TMR2_FIFO_COUNT	0x40002807	FIFO count register	0x0X
R32_TMR2_COUNT	0x40002808	Current count value register	0x0XXXXXXX
R32_TMR2_CNT_END	0x4000280C	Final count value register	0x0XXXXXXX
R32_TMR2_FIFO	0x40002810	FIFO register	0x0XXXXXXX
R16_TMR2_DMA_NOW	0x40002814	Current buffer address of DMA	0xXXXX
R16_TMR2_DMA_BEG	0x40002818	Start buffer address of DMA	0xXXXX
R16_TMR2_DMA_END	0x4000281C	End buffer address of DMA	0xXXXX

Mode Set Register (R8_TMRx_CTRL_MOD) (x=0/1/2)

Bit	Name	Access	Description	Reset value
[7:6]	RB_TMR_CAP_EDGE	RW	Capture trigger mode selection in capture mode: 00: Not triggered; 01: Capture the time between any edge changes; 10: Capture the time between falling edges; 11: Capture the time between rising edges. In the count mode, select the edge of count: 00: Not sample count; 01: Count when sampling to any edge; 10: Count when sampling to falling edge; 11: Count when sampling to rising edge.	0
[7:6]	RB_TMR_PWM_REPEAT	RW	Data repetition selection in PWM mode: 00: Repeat once; 01: Repeat 4 times; 10: Repeat 8 times; 11: Repeat 16 times.	0
5	RB_TMR_PWM_REPEAT_EN	RW	When repeating transmission in PWM mode, the method for switching data when the FIFO is empty and new data is written: 1: Switch to new data after completing the current repetition count; 0: Switch to new data immediately.	0
4	RB_TMR_CAP_COUNT	RW	Sub-mode of RB_TMR_MODE_IN=1 input mode: 1: Count mode; 0: Capture mode.	0
4	RB_TMR_OUT_POLAR	RW	In PWM mode, output polarity set:	0

			1: Default at high level, active low; 0: Default at low level, active high;	
3	RB_TMR_OUT_EN	RW	Timer output enable: 1: Output enabled; 0: Output disabled.	0
2	RB_TMR_COUNT_EN	RW	Timer count enable: 1: Enable counting; 0: Disable counting.	0
1	RB_TMR_ALL_CLEAR	RW	Clear the FIFO/counter/interrupt flag of timer: 1: Force to empty and clear; 0: Not clear.	1
0	RB_TMR_MODE_IN	RW	Timer mode set: 1: Input mode (capture mode or count mode); 0: Timing mode or PWM mode.	0

Interrupt Enable Register (R8_TMRx_INTER_EN) (x=0/1/2)

Bit	Name	Access	Description	Reset value
[7:5]	Reserved	RO	Reserved	0
4	RB_TMR_IE_FIFO_OV	RW	FIFO overflow (FIFO is full in capture mode or FIFO is empty in PWM mode) interrupt enable: 1: Enable interrupt; 0: Disable interrupt.	0
3	RB_TMR_IE_DMA_END	RW	DMA end interrupt enable (only TMR1/2 support): 1: Enable interrupt; 0: Disable interrupt.	0
2	RB_TMR_IE_FIFO_HF	RW	FIFO used more than half (FIFO>=4 in capture mode or FIFO<4 in PWM mode) interrupt enable: 1: Enable interrupt; 0: Disable interrupt.	0
1	RB_TMR_IE_DATA_ACT	RW	Data activation (In capture mode, it means that every time new data is captured. In PWM mode, it means that value triggers the effective level to end) interrupt enable: 1: Enable interrupt; 0: Disable interrupt.	0
0	RB_TMR_IE_CYC_END	RW	Cycle end (it refers to timeout in capture mode, and it refers to the end of cycle in PWM mode and timing mode) interrupt enable: 1: Enable interrupt; 0: Disable interrupt.	0

Interrupt Flag Register (R8_TMRx_INT_FLAG) (x=0/1/2)

Bit	Name	Access	Description	Reset value
[7:5]	Reserved	RO	Reserved	0
4	RB_TMR_IF_FIFO_OV	RW1	FIFO overflow (FIFO is full in capture	0

			mode or FIFO is empty in PWM mode) flag. Write 1 to reset: 1: Has overflowed; 0: Not overflowed.	
3	RB_TMR_IF_DMA_END	RW1	DMA end. Write 1 to reset: 1: Has completed; 0: Not completed.	0
2	RB_TMR_IF_FIFO_HF	RW1	FIFO used more than half (FIFO>=4 in capture mode or FIFO<4 in PWM mode) flag. Write 1 to reset: 1: FIFO has been used more than half; 0: FIFO has not been used more than half.	0
1	RB_TMR_IF_DATA_ACT	RW1	Data activation (it means that every time new data is captured in capture mode, and it means that value triggers the effective level to end in PWM mode) flag. Write 1 to reset: 1: Data generated/used; 0: Not generated/not used.	0
0	RB_TMR_IF_CYC_END	RW1	Cycle end (it refers to timeout in capture mode, and it refer to the end of cycle in PWM mode and timing mode) flag. Write 1 to reset: 1: Timeout/end of cycle; 0: No timeout/ not end.	0

FIFO Count Register (R8_TMRx_FIFO_COUNT) (x=0/1/2)

Bit	Name	Access	Description	Reset value
[7:4]	Reserved	RO	Reserved	0
[3:0]	R8_TMRx_FIFO_COUNT	RO	Data count in FIFO, the maximum value is 8.	0Xh

Current Count Value Register (R32_TMRx_COUNT) (x=0/1/2)

Bit	Name	Access	Description	Reset value
[31:26]	Reserved	RO	Reserved	0Xh
[25:0]	R32_TMRx_COUNT	RO	Current count value of counter.	XXXXXXXh

Final Count Value Set Register (R32_TMRx_CNT_END) (x=0/1/2)

Bit	Name	Access	Description	Reset value
[31:0]	R32_TMRx_CNT_END	RW	In timer mode, the number of clocks in a timing cycle; In PWM mode, the total number of clocks in a PWM cycle;	0XXXXXXXXh

			Capture the number of timeout clocks in capture mode. Only the lower 26 bits are valid, and the maximum value is 67108863. In counting mode, final count value -2 (overflow). <i>Note: With write operation on this register, the value of R32_TMRx_COUNT will be automatically cleared to 0.</i>	
--	--	--	---	--

FIFO Register (R32_TMRx_FIFO) (x=0/1/2)

Bit	Name	Access	Description	Reset value
[31:0]	R32_TMRx_FIFO	RO/WO	FIFO data register, only the lower 26 bits are valid.	0XXXXXXXXh

DMA Control Register (R8_TMRx_CTRL_DMA) (x=1/2) (only TMR1/2 support)

Bit	Name	Access	Description	Reset value
[7:5]	Reserved	RO	Reserved	0
4	RB_HI_PWM_EN	RW	160M high-frequency PWM mode enable: 1: Enable high-frequency PWM mode; 0: Disable high-frequency PWM mode.	0
3	Reserved	RO	Reserved	0
2	RB_TMR_DMA_LOOP	RW	DMA address loop enable bit: 1: Enable address loop; 0: Disable address loop. If the DMA address loop is enabled, when the DMA address is added to the set end address, it will automatically loop to the start address set.	0
1	Reserved	RO	Reserved	0
0	RB_TMR_DMA_ENABLE	RW	DMA function enable bit: 1: DMA enabled. 0: DMA disabled.	0

DMA Current Buffer Address (R16_TMRx_DMA_NOW) (x=1/2)

Bit	Name	Access	Description	Reset value
15	Reserved	RO	Reserved	0
[14:2]	R16_TMRx_DMA_NOW	RO	Current address of DMA data buffer. It can be used to calculate the number of conversions, and the calculation method is: COUNT = (TMR_DMA_NOW - TMR_D	XXXXh

			MA_BEG)/4.	
[1:0]	Reserved	RO	Reserved	0

DMA Start Buffer Address (R16_TMRx_DMA_BEG) (x=1/2)

Bit	Name	Access	Description	Reset value
15	Reserved	RO	Reserved	0
[14:2]	R16_TMRx_DMA_BEG	RW	DMA data buffer start address, the address must be 4-byte aligned.	XXXXh
[1:0]	Reserved	RO	Reserved	0

DMA End Buffer Address (R16_TMRx_DMA_END) (x=1/2)

Bit	Name	Access	Description	Reset value
15	Reserved	RO	Reserved	0
[14:2]	R16_TMRx_DMA_END	RW	DMA data buffer end address (not included), address must be 4-byte aligned.	XXXXh
[1:0]	Reserved	RO	Reserved	0

8.3 Functional Description and Configuration

8.3.1 Timing and Counting Functions

Each timer of the chip supports the longest time interval of 2^{26} clock cycles and performs an incremental count mode. If the system clock cycle is 32MHz, the longest time interval is: $31.25\text{nS} \times 2^{26} \approx 2\text{S}$. Each timer has an independent interrupt.

The operation steps for timing function are as follows:

- (1) Set RB_TMR_ALL_CLEAR, clear R32_TMRx_COUNT and interrupt flag, etc.
- (2) Set the R32_TMRx_CNT_END register to the time value that needs timing; $\text{Time} = \text{Ts}_{\text{sys}} \times \text{R32_TMRx_CNT_END}$;
- (3) Clear RB_TMR_ALL_CLEAR, clear the timing mode corresponding to RB_TMR_MODE_IN;
- (4) Optional steps, set R8_TMRx_INTER_EN register, set RB_TMR_IE_CYC_END to open the timing cycle interrupt;
- (5) Set the RB_TMR_COUNT_EN in the R8_TMRx_CTRL_MOD register, and start the timer counting;
- (6) When count value of R32_TMRx_COUNT is equal to that of R32_TMRx_CNT_END, the timing is completed. In this case, RB_TMR_IF_CYC_END in R8_TMRx_INT_FLAG is set to 1, which can be cleared by writing 1.

The operation steps for counting function are as follows:

- (1) Set the corresponding I/O pin direction of counting as input;
- (2) Set the count overflow final value in R32_TMRx_CNT_END;
- (3) Configure R8_TMRx_CTRL_MOD, set the corresponding count mode of RB_TMR_MODE_IN and RM_TMR_CAP_COUNT, clear RB_TMR_ALL_CLEAR, select sample edge method by RB_TMR_CAP_EDGE, set RB_TMR_COUNT_EN in R8_TMRx_CTRL_MOD to 1, enable counting function;
- (4) Optional, set the corresponding interrupt enable register bit if it is needed to enable interrupt;

- (5) Save current count value in R32_TMRx_COUNT. Every time the count value reaches final count value, RB_TMR_IE_CYC_END will be set to 1 and R32_TMRx_COUNT will be cleared to 0. Hardware interrupt is triggered if enabling interrupt.

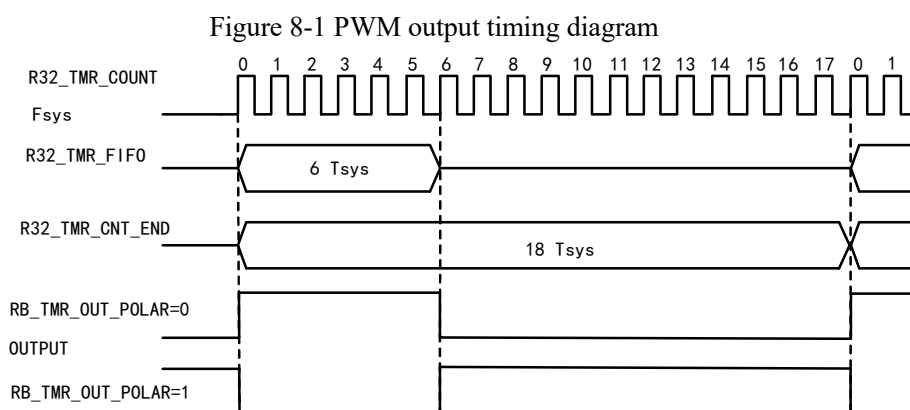
8.3.2 PWM Function

Each timer of the chip has PWM function. The PWM functions of TMR1 and TMR2 support DMA data loading. The default output polarity of PWM can be set to high level or low level. The repeated output times of the same data can be selected as 1, 4, 8 or 16. This repeat function is combined with DMA to simulate the effect of DAC. The shortest time unit for PWM to output valid level is 1 system clock cycle, and the duty cycle of PWM can be dynamically modified to simulate special waveforms.

PWM operation steps are as follows:

- (1) Set RB_TMR_ALL_CLEAR, empty and clear R32_TMRx_FIFO and interrupt flags, etc.
- (2) Set the PWM total cycle register R32_TMRx_CNT_END, the value shall not be less than the value in R32_TMRx_FIFO register;
- (3) Configure R8_TMRx_CTRL_MOD, clear RB_TMR_ALL_CLEAR, clear PWM mode corresponding to RB_TMR_MODE_IN, select the output polarity through RB_TMR_OUT_POLAR, and select the repetition times of the same data through RB_TMR_PWM_REPEAT as needed;
- (4) Set the data register R32_TMRx_FIFO, the minimum value is 0, with the corresponding duty cycle of 0%; the maximum value is the same as that of R32_TMR_CNT_END, with the corresponding duty cycle of 100%; the calculation of duty cycle: $R32_TMRx_FIFO/R32_TMRx_CNT_END$. TMR1 and TMR2 can load continuous dynamic data through DMA, and simulate special waveforms combined with the repeated output times of the same data;
- (5) Configure R8_TMRx_CTRL_MOD, set RB_TMR_COUNT_EN to start counting and RB_TMR_OUT_EN to allow PWM output;
- (6) Set the I/O pin corresponding to PWM as output;
- (7) Optional. If it is needed to enable interrupts, set the corresponding interrupt enable register bit;
- (8) After a PWM cycle is completed, if an interrupt is enabled, the hardware interrupt will be triggered after RB_TMR_IF_DATA_ACT or RB_TMR_IF_CYC_END is set;
- (9) The duty cycle of PWM can be dynamically changed by updating the data in R32_TMRx_FIFO. It is recommended to load it through DMA.

For example: Set the RB_TMR_OUT_POLAR bit to 0, R32_TMRx_FIFO to 6, R32_TMRx_CNT_END to 18, the basic timing diagram of PWM generation is as follows, and its duty cycle is: $R32_TMRx_FIFO/R32_TMRx_CNT_END = 1/3$.



If `RB_TMR_PWM_REPEAT` is set to 00, it means that the above process is repeated once, 01 means repeating 4 times, 10 means that repeating 8 times, and 11 means repeating 16 times. After repeating, load the next data in FIFO and then continue.

8.3.3 Capture Function

Each timer of chip has a capture function, among which the capture functions of TMR1 and TMR2 support DMA data storage. 3 capture modes can be selected: start from any edge trigger and end at any edge trigger, start from rising edge trigger and end at rising edge trigger, and start from falling edge trigger and end at falling edge trigger. The following table shows the description of capture trigger mode:

Table 8-5 Description of capture trigger mode

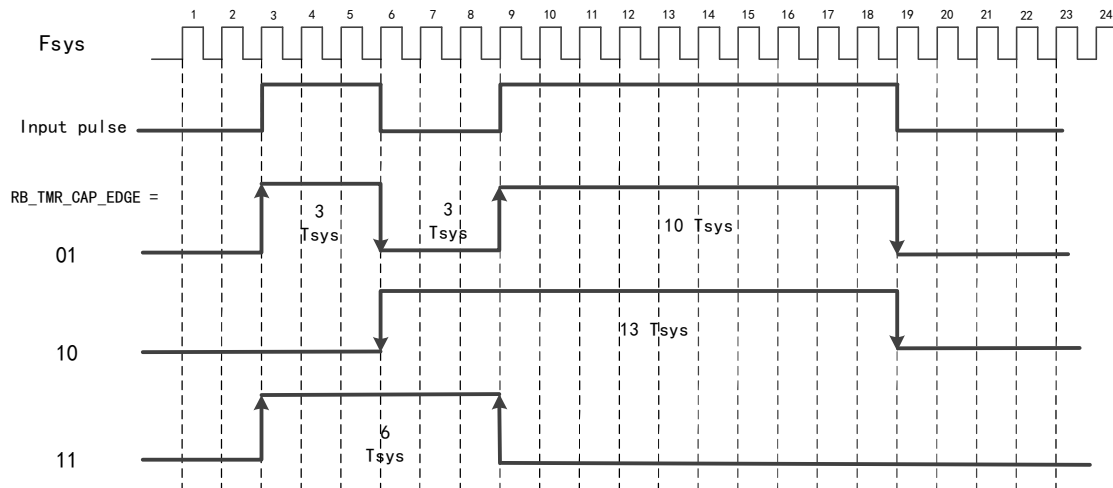
Capture mode select bit <code>RB_TMR_CATCH_EDGE</code>	Trigger Mode	Icon
00	Not capture	None
01	Edge trigger edge to edge	
10	Falling edge to falling edge	
11	Rising edge to rising edge	

There are 2 trigger states in edge trigger mode, which can capture high level width or low-level width. When the highest bit (bit 25) of the valid data in data register `R32_TMRx_FIFO` is 1, high level is captured; when it is 0, low level is captured. If the bit 25 of consecutive sets of data is 1 (or 0), the width of the high (or low) level exceeds the timeout value, and needs to be combined and accumulated.

In the trigger modes from falling edge to falling edge and from rising edge to rising edge, an input change cycle can be captured. When the highest bit (bit 25) of the valid data in data register `R32_TMRx_FIFO` is 0, one cycle is normally sampled; when it is 1, the input change period exceeds the timeout value `R32_TMRx_CNT_END`, and the latter set of data needs to be added and accumulated as a single input change period.

The specific description is shown in the figure below:

Figure 8-2 Count the capture cycle with system clock



As shown in the figure above, sample once in each clock cycle:

When RB_TMR_CATCH_EDGE=01b, sampling is set to the mode of edge trigger, and the time width sampled is 3, 3, 10;

When RB_TMR_CATCH_EDGE=10b, sampling is set to the mode of falling edge to falling edge, and the time width sampled is 13;

When RB_TMR_CATCH_EDGE=11b, sampling is set to the mode of rising edge to rising edge, and the time width sampled is 6.

Operation steps for capture mode:

- (1) Set RB_TMR_ALL_CLEAR, empty and clear R32_TMRx_FIFO and interrupt flags, etc.
- (2) Set the direction of the I/O pin corresponding to capture as input;
- (3) Set a reasonable capture timeout time in R32_TMRx_CNT_END, which can be used to generate a timeout interrupt when the input signal remains unchanged for a long time, and generate timeout data after the input signal does not change overtime (bit 25 of data is 1, and the lower 25 bits can be accumulated backward);
- (4) Configure R8_TMRx_CTRL_MOD, set the capture mode corresponding to RB_TMR_MODE_IN, select the edge mode of capture through RB_TMR_CAP_EDGE, set RB_TMR_COUNT_EN of R8_TMRx_CTRL_MOD as 1, and enable counting;
- (5) Optional step: If it is needed to enable interrupts, set the corresponding interrupt enable register bit;
- (6) To save the captured data in the way of DMA, you need to set the register R16_TMRx_DMA_BEG as the first address of buffer which stores the data captured, set the register R16_TMRx_DMA_END as the end address of buffer which stores the data captured (not included), and set the RB_TMR_DMA_ENABLE of R8_TMRx_CTRL_DMA as 1, and enable DMA function;
- (7) Clear RB_TMR_ALL_CLEAR of R8_TMRx_CTRL_MOD, and start the capture function;
- (8) Every time data is captured, RB_TMR_IF_DATA_ACT will be set as 1; if the interrupt is enabled, a hardware interrupt will be triggered; the captured data is stored in R32_TMRx_FIFO by default; if DMA is enabled, the captured data will be automatically stored in the data buffer set by DMA.

8.3.4 DMA Function

The chip's timers TMR1 and TMR2 support DMA functionality. When using DMA to complete interrupts, pay attention to the configuration sequence of the relevant registers.

The steps to enable the DMA completion interrupt are as follows:

- (1) Read `R16_TMR1_DMA_NOW` and set `R16_TMR1_DMA_END` to any value not equal to `R16_TMR1_DMA_NOW` (e.g., set it to `R16_TMR1_DMA_NOW + 0x100`);
- (2) Clear bit `RB_TMR_IF_DMA_END` in `R8_TMR1_INT_FLAG` to 0;
- (3) Set bit `RB_TMR_IE_DMA_END` in `R8_TMR1_INTER_EN` to 1.

The steps to clear the DMA completion interrupt in non-cyclic mode are as follows:

- (1) Read `R16_TMR1_DMA_NOW` and set `R16_TMR1_DMA_END` to any value not equal to `R16_TMR1_DMA_NOW` (e.g., set it to `R16_TMR1_DMA_NOW + 0x100`);
- (2) Clear bit `RB_TMR_IF_DMA_END` in `R8_TMR1_INT_FLAG` by writing a 1.

The steps to clear the DMA completion interrupt in the cyclic mode are as follows:

- (1) Clear bit `RB_TMR_IF_DMA_END` in the `R8_TMR1_INT_FLAG` register by writing a value of 1.

Chapter 9 Universal Asynchronous Receiver-Transmitter (UART)

9.1 Introduction to UART

The chip provides 4 sets of full-duplex UART0 and UART1. Full-duplex and half-duplex serial communication are supported. Among them, UART0 provides the transmit status pin for switching RS485, and supports MODEM signals DSR, RI, DCD and DTR.

9.1.1 Main Features

- 5/6/7/8 data bits, 1/2 stop bits.
- Support the verification modes of odd, even, no parity, blank 0 and flag 1, etc.
- Programmable communication baud rate, up to 6Mbps.
- Built-in 8-byte FIFO buffer, support 4 FIFO trigger stages.
- UART0 supports MODEM signals DSR, RI, DCD and DTR.
- Support serial frame error detection and Break circuit interval detection.
- Full-duplex and half-duplex serial communication are supported, and UART0 provides the transmit status pin for switching RS485.

9.2 Register Description

Table 9-1 UART0 registers

Name	Access address	Description	Reset value
R8_UART0_MCR	0x40003000	MODEM control register	0x00
R8_UART0_IER	0x40003001	Interrupt enable register	0x00
R8_UART0_FCR	0x40003002	FIFO control register	0x00
R8_UART0_LCR	0x40003003	Line control register	0x00
R8_UART0_IIR	0x40003004	Interrupt identification register	0x01
R8_UART0_LSR	0x40003005	Line status register	0x60
R8_UART0_MSR	0x40003006	MODEM status register	0xX0
R8_UART0_TXE	0x40003007	Transmit complete register	0x00
R8_UART0_RBR	0x40003008	Receive buffer register	0xXX
R8_UART0_THR	0x40003008	Transmit hold register	0xXX
R8_UART0_RFC	0x4000300A	Receive FIFO count register	0x00
R8_UART0_TFC	0x4000300B	Transmit FIFO count register	0x00
R16_UART0_DL	0x4000300C	Baud rate divisor latch	0xFFFF
R8_UART0_DIV	0x4000300E	Prescaler divisor register	0xFF
R8_UART0_ADR	0x4000300F	Slave address register	0xFF

Table 9-2 UART1 registers

Name	Access address	Description	Reset value
R8_UART1_MCR	0x40003400	MODEM control register	0x00
R8_UART1_IER	0x40003401	Interrupt enable register	0x00
R8_UART1_FCR	0x40003402	FIFO control register	0x00
R8_UART1_LCR	0x40003403	Line control register	0x00
R8_UART1_IIR	0x40003404	Interrupt identification register	0x01
R8_UART1_LSR	0x40003405	Line status register	0x60
R8_UART1_TXE	0x40003407	Transmit complete register	0x00
R8_UART1_RBR	0x40003408	Receive buffer register	0xXX
R8_UART1_THR	0x40003408	Transmit hold register	0xXX
R8_UART1_RFC	0x4000340A	Receive FIFO count register	0x00
R8_UART1_TFC	0x4000340B	Transmit FIFO count register	0x00
R16_UART1_DL	0x4000340C	Baud rate divisor latch	0xFFFF
R8_UART1_DIV	0x4000340E	Prescaler divisor register	0xXX

MODEM Control Register (R8_UARTx_MCR) (x=0/1)

Bit	Name	Access	Description	Reset value
7	RB_MCR_HALF	RW	Half-duplex transceiver mode control (only supported by UART0): 1: Enter half-duplex transceiver mode, transmit with priority, and receive when not transmitting; 0: Disable half-duplex mode.	0
6	RB_MCR_TNOW	RW	Status enable that DTR pin output is being transmitted (only supported by UART0): 1: Output the indication status of being transmitted to DTR pin, used to control the RS485 receive/transmit switch. 0: DTR pin is in normal function.	0
5	Reserved	RO	Reserved	0
4	RB_MCR_LOOP	RW	Test mode of internal loop enable (only supported by UART0): 1: Enable the test mode of internal loop; 0: Disable the test mode of internal loop. In the test mode of the internal loop, all external output pins of the serial port are invalid, TXD internally returns to RXD, DTR internally returns to DSR, OUT1 internally returns to RI and OUT2 internally returns to DCD.	0
3	RB_MCR_OUT2 RB_MCR_INT_OE	RW	UART interrupt request output control: 1: Enable to send request; 0: Disable.	0

2	RB_MCR_OUT1	RW	User-defined MODEM control (only supported by UART0), and no actual output pin is connected: 1: Set high; 0: Set low.	0
1	Reserved	RO	Reserved	0
0	RB_MCR_DTR	RW	DTR signal output level control (only supported by UART0): 1: DTR signal output is valid (low level); 0: DTR signal output high level (default).	0

Interrupt Enable Register (R8_UARTx_IER) (x=0/1)

Bit	Name	Access	Description	Reset value
7	RB_IER_RESET	WZ	UART software reset control bit, automatically cleared: 1: Software resets UART; 0: Normal operation.	0
6	RB_IER_TXD_EN	RW	UART TXD pin output enable bit: 1: Enable pin output; 0: Disable pin output.	0
5	Reserved	RO	Reserved	0
4	RB_IER_DTR_EN	RW	DTR pin output enable bit (only supported by UART0): 1: Enable output; 0: Disable output.	0
3	RB_IER_MODEM_CHG	RW	Modem input status change interrupt enable bit (only supported by UART0): 1: Enable interrupt; 0: Disable interrupt.	0
2	RB_IER_LINE_STAT	RW	Receive line status interrupt enable bit: 1: Enable interrupt; 0: Disable interrupt.	0
1	RB_IER_THR_EMPTY	RW	Transmit hold register empty interrupt enable bit: 1: Enable interrupt; 0: Disable interrupt.	0
0	RB_IER_RECV_RDY	RW	Receive data interrupt enable bit: 1: Enable interrupt; 0: Disable interrupt.	0

FIFO Control Register (R8_UARTx_FCR) (x=0/1)

Bit	Name	Access	Description	Reset value
[7:6]	RB_FCR_FIFO_TRIG	RW	Trigger points select of receiving FIFO interrupt and hardware flow control: 00: 1 byte; 01: 2 bytes; 10: 4 bytes; 11: 7 bytes. Used to set the trigger points of receiving FIFO interrupt and hardware flow control. For example: 10 corresponds to 4 bytes, that is, interrupt available for receiving	0

			data is generated when 4 bytes are received.	
[5:3]	Reserved	RO	Reserved	0
2	RB_FCR_TX_FIFO_CLR	WZ	Transmit FIFO data clear enable bit, and automatically cleared: 1: Clear the data of transmitter FIFO (excluding TSR); 0: Not clear the data of transmitter FIFO.	0
1	RB_FCR_RX_FIFO_CLR	WZ	Receive FIFO data clear enable bit, and automatically cleared: 1: Clear the data of receiver FIFO (excluding RSR); 0: Not clear the data of receiver FIFO.	0
0	RB_FCR_FIFO_EN	RW	FIFO enable bit: 1: Enable 8-byte FIFO; 0: Disable FIFO. After disabling FIFO, it is 16C450 compatible mode, which means that there is only 1 byte in FIFO (RECV_TG1=0, RECV_TG0=0, FIFO_EN=1), and it is recommended to enable.	0

Line Control Register (R8_UARTx_LCR) (x=0/1)

Bit	Name	Access	Description	Reset value
7	RB_LCR_DLAB RB_LCR_GP_BIT	RW	UART general purpose bit, user-defined.	0
6	RB_LCR_BREAK_EN	RW	Force to generate BREAK line interval enable: 1: Forced to generate; 0: Not generate.	0
[5:4]	RB_LCR_PAR_MOD	RW	Parity bit format selection: 00: Odd; 01: Even; 10: Mark (MARK, set to 1); 11: Space (SPACE, cleared). Valid only when RB_LCR_PAR_EN is 1.	0
3	RB_LCR_PAR_EN	RW	Parity bit enable: 1: Allow to generate parity bit when sending and check parity bit when receiving; 0: No parity bit.	0
2	RB_LCR_STOP_BIT	RW	Stop bit format set: 0: 1 stop bit; 1: 2 stop bits.	0
[1:0]	RB_LCR_WORD_SZ	RW	UART data length selection: 00: 5 data bits; 01: 6 data bits;	0

			10: 7 data bits; 11: 8 data bits.	
--	--	--	-----------------------------------	--

Interrupt Identification Register (R8_UARTx_IIR) (x=0/1)

Bit	Name	Access	Description	Reset value
[7:6]	RB_IIR_FIFO_ID	RO	UART FIFO enable status: 11: FIFO has been enabled; 00: FIFO is not enabled.	0
[5:4]	Reserved	RO	Reserved	0
[3:0]	RB_IIR_INT_MASK	RO	Interrupt flag: If the RB_IIR_NO_INT bit is 0, an interrupt is generated, and it is needed to judge the interrupt source after reading. Please refer to Table 9-5 for details.	0
0	RB_IIR_NO_INT	RO	UART no interrupt flag: 1: No interrupt; 0: Interrupt.	1

The meanings of bit RB_IIR_NO_INT of interrupt identification register R8_UARTx_IIR and each bit of RB_IIR_INT_MASK is shown in the following table:

Table 9-3 Meaning of RB_IIR_INT_MASK in IIR register

IIR register bit				Priority	Interrupt type	Interrupt source	Means of clearing interrupts
IID3	IID2	IID1	NOINT				
0	0	0	1	None	No interrupt	No interrupt	-
1	1	1	0	0	Bus address matching	The received one data is the UART bus address, and the address matches the preset slave value or the broadcast address. (Only supported by UART0)	Read IIR or disable multi-device mode
0	1	1	0	1	Receive line status	OVER_ERR/PAR_ERR/FRAM_ERR/BREAK_ERR	Read LSR
0	1	0	0	2	Receive data available	The number of bytes received reaches the trigger point of FIFO.	Read RBR
1	1	0	0	2	Receive data timeout	The next data is not received for more than 4 data periods.	Read RBR
0	0	1	0	3	THR register empty	Transmit hold register is empty, or RB_IER_THR_EMPTY bit is changed from 0 to 1 and triggered.	Read IIR or write THR
0	0	0	0	4	MODEM input change	Triggered by setting Δ DSR, Δ RI and Δ DCD to 1.	Read MSR

Line Status Register (R8_UARTx_LSR) (x=0/1)

Bit	Name	Access	Description	Reset value
7	RB_LSR_ERR_RX_FIFO	RO	Receive FIFO error flag: 1: At least one PAR_ERR or FRAM_ERR or BREAK_ERR error in the receiver FIFO;	0

			0: No error in receiver FIFO.	
6	RB_LSR_TX_ALL_EMP	RO	THR and TSR empty flag. 1: Both are empty; 0: Both are not empty.	1
5	RB_LSR_TX_FIFO_EMP	RO	Transmit FIFO empty flag: 1: Transmit FIFO is empty; 0: Transmit FIFO is not empty.	1
4	RB_LSR_BREAK_ERR	RZ	BREAK line interval detection flag: 1: BREAK is detected; 0: BREAK is not detected.	0
3	RB_LSR_FRAME_ERR	RZ	Data frame error flag: 1: Frame error in the data being read from the receiver FIFO, and a valid stop bit is missing. 0: No error in the currently read data frame.	0
2	RB_LSR_PAR_ERR	RZ	Receive data Parity error flag: 1: Parity error in the data being read from the receiver FIFO. 0: The currently read data parity is correct.	0
1	RB_LSR_OVER_ERR	RZ	Receiver FIFO buffer overflow flag: 1: Has overflowed; 0: Not overflowed.	0
0	RB_LSR_DATA_RDY	RO	Receiver FIFO receive data flag: 1: Data in FIFO; 0: No data. After reading all the data in the FIFO, this bit will be automatically cleared.	0

MODEM Status Register (R8_UART0_MSR) (only supported by UART0)

Bit	Name	Access	Description	Reset value
7	RB_MSR_DCD	RO	DCD pin status: 1: DCD is active (low level); 0: DCD is inactive (high level).	0
6	RB_MSR_RI	RO	RI pin status: 1: RI is active (low level); 0: RI is inactive (high level).	0
5	RB_MSR_DSR	RO	DSR pin status: 1: DSR pin is active (low level); 0: DSR pin is inactive (high level).	X
4	Reserved	RO	Reserved	0
3	RB_MSR_DCD_CHG	RZ	DCD input status change flag: 1: Has changed; 0: No change.	0
2	RB_MSR_RI_CHG	RZ	RI input status change flag: 1: Has changed; 0: No change.	0
1	RB_MSR_DSR_CHG	RZ	DSR pin input status change flag: 1: Has changed; 0: No change.	0
0	Reserved	RO	Reserved	0

Transmit Complete Register (R8_UARTx_TXE) (x=0/1)

Bit	Name	Access	Description	Reset value
7	RB_UART_TRAN_IF	RW1	Data transfer completion flag in FIFO: 1: Transfer completed; 0: Transfer not completed.	0
[6:1]	Reserved	RO	Reserved	0
0	RB_UART_TRAN_IE	RW	Interrupt enable for completed data transfer in FIFO: 1: Enable interrupt; 0: Disable interrupt.	0

Receive Buffer Register (R8_UARTx_RBR) (x=0/1)

Bit	Name	Access	Description	Reset value
[7:0]	R8_UARTx_RBR	RO	Data receive buffer register. If the DATA_RDY bit of LSR is 1, the received data can be read from this register; If FIFO_EN is 1, the data received from UART shift register RSR will be firstly stored in the receiver FIFO, and then read out through the register.	XXh

Transmit Hold Register (R8_UARTx_THR) (x=0/1)

Bit	Name	Access	Description	Reset value
[7:0]	R8_UARTx_THR	WO	Transmit hold register. Transmitter FIFO is included, used to write the data to be transmitted; if FIFO_EN is 1, the written data will be firstly stored in the transmitter FIFO, and then output one by one through the transmit shift register TSR.	XXh

Receive FIFO Count Register (R8_UARTx_RFC) (x=0/1)

Bit	Name	Access	Description	Reset value
[7:0]	R8_UARTx_RFC	RO	Data count in the current receiver FIFO.	00h

Transmit FIFO Count Register (R8_UARTx_TFC) (x=0/1)

Bit	Name	Access	Description	Reset value
[7:0]	R8_UARTx_TFC	RO	Data count in the current transmitter FIFO.	00h

Baud Rate Divisor Lock (R16_UARTx_DL) (x=0/1)

Bit	Name	Access	Description	Reset value
[15:0]	R16_UARTx_DL	RW	The 16-bit divisor is used to calculate the baud rate.	XXXXh

			Formula: Divisor = the serial internal reference clock Fuart / 16 / the required communication baud rate. For example: If the serial internal reference clock Fuart is 1.8432MHz and the required baud rate is 9600bps, then the divisor = 1843200/16/9600=12.	
--	--	--	---	--

Prescaler Divisor Register (R8_UARTx_DIV) (x=0/1)

Bit	Name	Access	Description	Reset value
[7:0]	R8_UARTx_DIV	RW	It is used to calculate the internal reference clock of UART, the lower 7 bits are valid. Formula: Divisor = Fsys*2 / internal reference clock of UART, the maximum value is 127.	XXh

Slave Address Register (R8_UART0_ADR) (only supported by UART0)

Bit	Name	Access	Description	Reset value
[7:0]	R8_UART0_ADR	RW	Slave address of UART0 during multi-device communication: FFh: Not used; Others: Slave address.	0FFh

R8_UART0_ADR presets the address when this device acts as a slave, it is used to automatically compare the received addresses during multi-device communication, and generate an interrupt when the address matches or when the broadcast address 0FFH is received. Meanwhile, it is allowed to receive subsequent data packets. Data is not received until the address matches. After it starts sending data or rewriting the R8_UART0_ADR register, stop receiving any data, until the address matches again next time or the broadcast address is received.

When R8_UART0_ADR is 0FFH or RB_LCR_PAR_EN=0, the automatic comparison function of bus address is disabled.

When R8_UART0_ADR is not 0FFH and RB_LCR_PAR_EN=1, the automatic comparison function of bus address is enabled, and the following parameters should be configured: RB_LCR_WORD_SZ is 11b to select method of 8 data bits. For the case when the address byte is MARK (that is, the bit 9 of data byte is 0), RB_LCR_PAR_MOD should be set to 10b. For the case when the address byte is SPACE (that is, the bit 9 of data byte is 1), RB_LCR_PAR_MOD should be set to 11b.

9.3 Functional Description and Configuration

UART0/1 output pins are all at 1.5V LVCMOS level. The pins in UART mode include: data transmission pins and MODEM contact signal pins (only supported by UART0). Data transmission pins include: TXD pin and RXD pin, both of which are at high level by default. MODEM contact signal pins include: DSR pin, RI pin, DCD pin, DTR pin, all of which are at high level by default.

All these MODEM contact signals can be used as general-purpose I/O pins, and the application program controls them and define their purposes.

2 sets of UARTs have built-in independent transceiver buffers and 8-byte FIFOs, support simplex, half-duplex or full-duplex UART communication. Serial data includes 1 low-level start bit, 5/6/7/8 data bits, 0/1 additional check bit or flag bit, 1/2 high-level stop bits, and supports odd/even/mark/blank checking. The baud rate error of the serial port transmitting signal is less than 0.5%, and the allowable baud rate error of the serial port receiving signal is not more than 2%.

9.3.1 Baud Rate Calculation

- 1) Calculate the internal reference clock Fuart of serial port, set the R8_UARTx_DIV register, the maximum value is 127, and usually 1 is written.
- 2) Calculate the baud rate and set R16_UARTx_DL register.

$$\text{Baud rate} = \text{Fsys} * 2 / \text{R8_UARTx_DIV} / 16 / \text{R16_UARTx_DL}$$

9.3.2 UART Transmission

"THR register empty" interrupt UART_II_THR_EMPTY sent by UART means that the current transmitter FIFO is empty. The interrupt is cleared when the IIR register is read, or cleared when the next data is written to THR. If only one byte is written to THR, it will soon generate again a request to send THR register empty interrupt as the byte is quickly transferred to the transmitter shift register (TSR) to start transmitting. At this point, the next data ready to be transmitted can be written. After all the data in TSR register is removed, UART transmission is completed. At this time, RB_LSR_TX_ALL_EMP bit of LSR register becomes active at 1.

In interrupt trigger mode, when THR empty interrupt from UART is received, if FIFO is enabled, up to 8 bytes can be written to THR and FIFO at a time, and then will be transmitted automatically by the controller in sequence; if FIFO is disabled, only one byte can be written at a time. If no data needs to be transmitted, exit directly (the interrupts have been automatically cleared when IIR is read before).

In the query mode, whether the transmitter FIFO is empty can be judged according to RB_LSR_TX_FIFO_EMP bit of LSR. When this bit is 1, the data can be written to THR and FIFO. If FIFO is enabled, up to 8 bytes can be written at a time.

R8_UARTx_TFC register can also be read to determine the number of remaining data to be sent in the current FIFO. If it is not equal to 8, continue to write the data to be sent into the FIFO, and that can save filling time.

9.3.3 UART Reception

UART receive data available interrupt UART_II_RECV_RDY means that the number of existing data bytes in the receiver FIFO has reached or exceeded the FIFO trigger points set and selected by RB_FCR_FIFO_TRIG of FCR. The interrupt is cleared when the data is read from RBR to cause the number of bytes in the FIFO less than that of the FIFO trigger points.

UART receive data timeout interrupt UART_II_RECV_TOUT means that there is at least 1-byte data in the receiver FIFO, and the waiting time is equivalent to the time of receiving 4 data starting from the last time when UART receives data and the last time when the system takes the data. The interrupt is cleared when a new data is received again or after the MCU reads RBR once. When receiver FIFO is empty, RB_LSR_DATA_RDY bit of LSR is 0; when there is data in the receiver FIFO, it is valid when RB_LSR_DATA_RDY bit is 1.

In the interrupt trigger mode, R8_UARTx_RFC register can be read to query the remaining data count in the current FIFO after receiving UART receive data timeout interrupt, and read all the data directly, or continuously query the RB_LSR_DATA_RDY of LSR. If this bit is valid, read the data until this bit becomes invalid. After

receiving UART receive data available interrupt, read the data for the number of bytes set by RB_FCR_FIFO_TRIG from RBR at one time, or read all the data in the current FIFO according to the RB_LSR_DATA_RDY bit and the R8_UARTx_RFC register.

In query mode, whether the receiver FIFO is empty can be judged according to the RB_LSR_DATA_RDY bit of LSR, or read the R8_UARTx_RFC register to get the data count in the current FIFO and get all the data received by UART.

Chapter 10 Serial Peripheral Interface (SPI)

10.1 Introduction to SPI

SPI is a full-duplex serial interface with a host and several slaves connected to the bus, and only a pair of host and slave is communicating at the same time. Usually, SPI interface consists of 4 pins: SPI chip selected pin SCS, SPI clock pin (SCK), SPI serial data pin MISO (master input/slave output pin) and SPI serial data pin MOSI (master output/slave input pin).

10.1.1 Main Features

The chip provides 1 SPI interface with the following characteristics:

- Support both master mode and slave mode.
- Compatible with Serial Peripheral Interface (SPI) specification.
- Data transfer modes: mode 0 and mode 3.
- 8-bit data transmission mode, optional data bit sequence: low bits of a byte are in front or high bits are in front.
- Clock frequency can be up to half of the system clock frequency (Fsys).
- 8-byte FIFO.
- Slave mode supports the first byte as command mode or data stream mode.
- Support DMA, so the data transmission efficiency is higher.

10.2 Register Description

Table 10-1 SPI0 registers

Name	Access address	Description	Reset value
R8_SPI0_CTRL_MOD	0x40004000	SPI0 mode control register	0x02
R8_SPI0_CTRL_CFG	0x40004001	SPI0 configuration register	0x00
R8_SPI0_INTER_EN	0x40004002	SPI0 interrupt enable register	0x00
R8_SPI0_CLOCK_DIV R8_SPI0_SLAVE_PRE	0x40004003	SPI0 clock divider register in master mode SPI0 preset data register in slave mode	0x10
R8_SPI0_BUFFER	0x40004004	SPI0 data buffer	0xFF
R8_SPI0_RUN_FLAG	0x40004005	SPI0 working status register	0x00
R8_SPI0_INT_FLAG	0x40004006	SPI0 interrupt flag register	0x40
R8_SPI0_FIFO_COUNT	0x40004007	SPI0 interrupt trigger mode selection register	0x00
R8_SPI0_INT_TYPE	0x40004008	SPI0 interrupt 1 enable register	0x00
R8_SPI0_INTER1_EN	0x40004009	SPI0 interrupt 1 flag register	0x00
R8_SPI0_INT1_FLAG	0x4000400A	SPI0 data receive/transmit data total length register	0x00
R16_SPI0_TOTAL_CNT	0x4000400C	SPI0 data FIFO register	0x0000

R8_SPI0_FIFO	0x40004010	SPI0 transceiver FIFO count register	0xXX
R8_SPI0_FIFO_COUNT1	0x40004013	Current address of SPI0 DMA buffer	0x00
R16_SPI0_DMA_NOW	0x40004014	Start address of SPI0 DMA buffer	0xFFFF
R16_SPI0_DMA_BEG	0x40004018	End address of SPI0 DMA buffer	0xFFFF

SPI Mode Control Register (R8_SPI0_CTRL_MOD)

Bit	Name	Access	Description	Reset value
7	RB_SPI_MISO_OE	RW	MISO pin output enable (can be used at data line switching direction in 2-wire mode): 1: MISO output enabled; 0: MISO output disabled.	0
6	RB_SPI_MOSI_OE	RW	MOSI pin output enable: 1: MOSI output enabled; 0: MOSI output disabled.	0
5	RB_SPI_SCK_OE	RW	SCK pin output enable: 1: SCK output enabled; 0: SCK output disabled.	0
4	RB_SPI_FIFO_DIR	RW	FIFO direction: 1: Input mode (receive data); 0: Output mode (transmit data).	0
3	RB_SPI_SLV_CMD_MOD	RW	First byte mode selection in SPI0 slave mode: 1: First byte command mode; 0: Data stream mode. In the first byte command mode, it will be regarded as a command code when receiving the first byte of data after the SPI chip select is valid and RB_SPI_IF_FST_BYTE will be set to 1.	0
3	RB_SPI_MST_SCK_MOD	RW	Clock idle mode selection in master mode: 1: Mode3 (SCK is at high level when idle); 0: Mode0 (SCK is at low level when idle).	0
2	RB_SPI_2WIRE_MODE	RW	2-wire or 3-wire SPI mode selection in slave mode: 1: 2-wire mode/halfduplex (SCK/MISO); 0: 3-wire mode/full duplex (SCK/MOSI/MISO).	0
1	RB_SPI_ALL_CLEAR	RW	SPI FIFO/counter/interrupt flag clear: 1: Force to empty and clear; 0: Not clear.	1
0	RB_SPI_MODE_SLAVE	RW	SPI0 master/slave mode selection: 1: Slave mode; 0: Master mode.	0

SPI Configuration Register (R8_SPI0_CTRL_CFG)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved	0
6	RB_SPI_MST_DLY_EN	RW	Input delay enable in master mode: 1: Enable, used for high-speed applications such as SPI clock close to half of Fsys; 0: Disable, regular applications.	0
5	RB_SPI_BIT_ORDER	RW	SPI data bit order selection: 1: LSB first, 0: MSB first.	0
4	RB_SPI_AUTO_IF	RW	Enable the function of automatically clearing flag bit RB_SPI_IF_BYTE_END when accessing BUFFER/FIFO: 1: Enable; 0: Disable.	0
3	Reserved	RO	Reserved	0
2	RB_SPI_DMA_LOOP	RW	DMA address loop enable: 1: Enable address loop; 0: Disable address loop. If the DMA address loop is enabled, when the DMA address is added to the set end address, the auto loop points to the set first address.	0
1	RB_MST_CLK_SEL	RW	When RB_SPI_MODE_SLAVE = 0 (Master mode), Master clock polarity selection: 1: Polarity inverted; 0: Polarity unchanged. When RB_SPI_MODE_SLAVE = 1 (2-wire mode), Slave input/output direction selection: 1: Slave input; 0: Slave output.	0
0	RB_SPI_DMA_ENABLE	RW	DMA enable: 1: Enable; 0: Disable.	0

SPI Interrupt Enable Register (R8_SPI0_INTER_EN)

Bit	Name	Access	Description	Reset value
7	RB_SPI_IE_FST_BYTE	RW	In the first byte command mode of slave mode, first byte interrupt receive enable: 1: Enable receiving the first byte interrupt; 0: Disable receiving the first byte interrupt.	0
[6:5]	Reserved	RO	Reserved	0
4	RB_SPI_IE_FIFO_OV	RW	In slave mode, FIFO overflow (FIFO is full when receiving, or FIFO is empty when sending) interrupt enable:	0

			1: Interrupt enabled; 0: Interrupt disabled.	
3	RB_SPI_IE_DMA_END	RW	DMA end interrupt enable: 1: Interrupt enabled; 0: Interrupt disabled.	0
2	RB_SPI_IE_FIFO_HF	RW	More than half of FIFO used interrupt enable: 1: Interrupt enabled; 0: Interrupt disabled.	0
1	RB_SPI_IE_BYTE_END	RW	SPI single byte transmission completion interrupt enable: 1: Interrupt enabled; 0: Interrupt disabled.	0
0	RB_SPI_IE_CNT_END	RW	SPI all byte transmission completion interrupt enable: 1: Interrupt enabled; 0: Interrupt disabled.	0

SPI Clock Divider Register in Master Mode (R8_SPI0_CLOCK_DIV)

Bit	Name	Access	Description	Reset value
[7: 0]	R8_SPI_CLOCK_DIV	RW	Frequency division factor in master mode, the minimum value is 2, the maximum value is 254. F _{sck} = F _{sys} /frequency division factor.	10h

SPI Preset Data Register in Slave Mode (R8_SPI0_SLAVE_PRE)

Bit	Name	Access	Description	Reset value
[7: 0]	R8_SPI0_SLAVE_PRE	RW	Preset data first returned in slave mode. Used to receive the returned data after first byte of data.	10h

SPI Data Buffer (R8_SPI0_BUFFER)

Bit	Name	Access	Description	Reset value
[7: 0]	R8_SPIx_BUFFER	RW	SPI data transmit and receive buffer.	XXh

SPI Working Status Register (R8_SPI0_RUN_FLAG)

Bit	Name	Access	Description	Reset value
7	RB_SPI_SLV_SELECT	RO	Chip select status in slave mode: 1: Being selected; 0: No chip selected.	0
6	RB_SPI_SLV_CS_LOAD	RO	First loading status after chip select in slave mode: 1: Being loading R8_SPI0_SLAVE_PRE; 0: Not yet loaded or has completed.	0

5	RB_SPI_FIFO_READY	RO	FIFO ready: 1: FIFO is ready (R16_SPIx_TOTAL_CNT is not 0, and the FIFO is not full when receiving or the FIFO is not empty when transmitting); 0: FIFO is not ready.	0
4	RB_SPI_SLV_CMD_ACT	RO	Command received completion status in slave mode, that is, completing the exchange of first byte data: 1: That has just been exchanged is the first byte; 0: The first byte has not been exchanged or it is not the first byte.	0
[3:0]	Reserved	RO	Reserved	0

SPI Interrupt Flag Register (R8_SPI0_INT_FLAG)

Bit	Name	Access	Description	Reset value
7	RB_SPI_IF_FST_BYTE	RW1	First byte received flag in slave mode: 1: The first byte has been received; 0: The first byte is not received.	0
6	RB_SPI_FREE	RO	Current SPI free: 1: Free; 0: Not free.	1
5	Reserved	RO	Reserved	0
4	RB_SPI_IF_FIFO_OV	RW1	In slave mode, FIFO overflow (FIFO is full when receiving or FIFO is empty when transmitting) flag. Write 1 to reset: 1: Overflow; 0: Not overflow.	0
3	RB_SPI_IF_DMA_END	RW1	DMA end flag. Write 1 to reset: 1: End; 0: Not end.	0
2	RB_SPI_IF_FIFO_HF	RW1	More than half of FIFO used (FIFO $\geq$ 4 when receiving or FIFO $<$ 4 when transmitting) flag. Write 1 to reset: 1: More than half of FIFO has been used; 0: FIFO has been used not more than half.	0
1	RB_SPI_IF_BYTE_END	RW1	SPI single byte transfer end flag. Write 1 to reset: 1: End; 0: Not end.	0
0	RB_SPI_IF_CNT_END	RW1	SPI all byte transfer end flag. Write 1 to reset: 1: All byte transfer ends;	1

			0: All byte transfer not end.	
--	--	--	-------------------------------	--

SPI Transceiver FIFO Count Register (R8_SPI0_FIFO_COUNT)

Bit	Name	Access	Description	Reset value
[7:4]	Reserved	RO	Reserved	0
[3:0]	R8_SPIx_FIFO_COUNT	RW	Current byte count in FIFO.	0

SPI Interrupt Trigger Mode Selection Register (R8_SPI0_INT_TYPE)

Bit	Name	Access	Description	Reset value
[7:5]	Reserved	RO	Reserved	0
[4:0]	RB_SPI_INT_TYPE	RW	Interrupt trigger mode selection: 1 = Rising edge trigger, 0 = Level trigger. bit[4]: RB_SPI_IF_FIFO_FULL; bit[3]: RB_SPI_IF_FIFO_EMPTY; bit[2]: RB_SPI_IF_DMA_END; bit[1]: RB_SPI_IF_FIFO_HF; bit[0]: RB_SPI_IF_CNT_END.	0

SPI Interrupt 1 Enable Register (R8_SPI0_INTER1_EN)

Bit	Name	Access	Description	Reset value
[7:2]	Reserved	RO	Reserved	0
1	RB_SPI_IE_FIFO_FULL	RW	Current FIFO data full interrupt enable: 1: Enable on; 0: Disable on.	0
0	RB_SPI_IE_FIFO_EMPTY	RW	Current FIFO data empty interrupt enable: 1: Enable on; 0: Disable on.	0

SPI Interrupt 1 Flag Register (R8_SPI0_INT1_FLAG)

Bit	Name	Access	Description	Reset value
[7:2]	Reserved	RO	Reserved	0
1	RB_SPI_IF_FIFO_FULL	RW1	The current FIFO data full flag is set to 1 and cleared to 0: 1: FIFO is full; 0: FIFO is not full.	0
0	RB_SPI_IF_FIFO_EMPTY	RW1	Current FIFO data empty flag bit, write 1 to clear: 1: FIFO is empty; 0: FIFO is not empty.	0

The registers R8_SPIx_BUFFER and R8_SPIx_FIFO are both SPI data related registers, and the main differences between them are:

Reading R8_SPIx_BUFFER means to obtain the data from the last exchange of SPI, and it does not affect FIFO and R8_SPIx_FIFO_COUNT;

Writing to R8_SPIx_BUFFER in master mode means to send the byte directly, and the write operation in slave mode is not defined;

Reading R8_SPIx_FIFO means to obtain the data from the earliest exchange in FIFO, which will reduce FIFO and R8_SPIx_FIFO_COUNT;

Writing to R8_SPI0_FIFO means to temporarily store the data in FIFO. In slave mode, the external SPI host decides when to take it. In master mode, the transmission is automatically started when R16_SPIx_TOTAL_CNT is not 0.

SPI Transceiver FIFO Count Register (R8_SPI0_FIFO_COUNT1)

Bit	Name	Access	Description	Reset value
[7:4]	Reserved	RO	Reserved	0
[3:0]	R8_SPI0_FIFO_COUNT1	RW	Current byte count in FIFO. The same as R8_SPIx_FIFO_COUNT.	0

Current Address of SPI0 DMA Buffer (R16_SPI0_DMA_NOW)

Bit	Name	Access	Description	Reset value
15	Reserved	RO	Reserved	0
[14:0]	R16_SPI0_DMA_NOW	RW	Current address of DMA data buffer. It can be used to calculate the number of conversions. COUNT=SPI0_DMA_NOW-SPI0_DMA_BEG.	XXXXh

Start Address of SPI0 DMA Buffer (R16_SPI0_DMA_BEG)

Bit	Name	Access	Description	Reset value
15	Reserved	RO	Reserved	0
[14:0]	R16_SPI0_DMA_BEG	RW	Start address of DMA data buffer, only the lower 15 bits are valid.	XXXXh

End Address of SPI0 DMA Buffer (R16_SPI0_DMA_END)

Bit	Name	Access	Description	Reset value
15	Reserved	RO	Reserved	0
[14:0]	R16_SPI0_DMA_END	RW	End address of DMA data buffer (not included), only the lower 15 bits are valid.	XXXXh

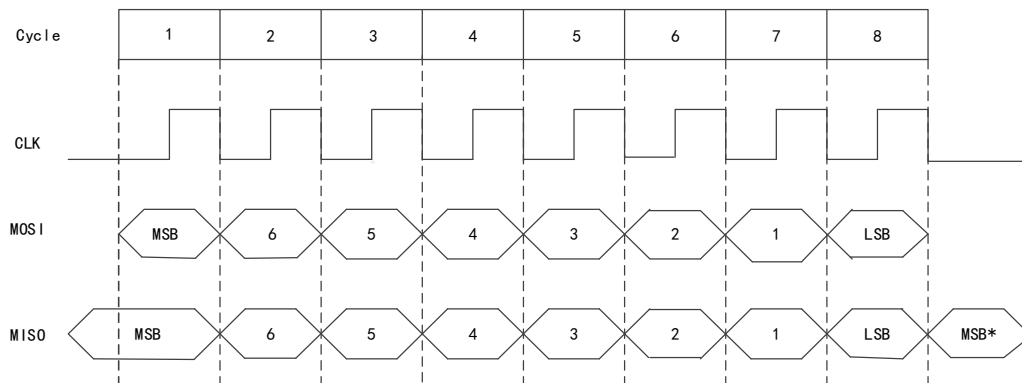
10.3 SPI Transfer Frame Formats

SPI supports 2 transfer frame formats, mode0 and mode3, which can be selected by setting RB_SPI_MST_SCK_MOD in R8_SPIx_CTRL_MOD. Always sample and input serial data at rising edge of SCK, and output serial data at falling edge.

The data transmission formats are shown in the figures below:

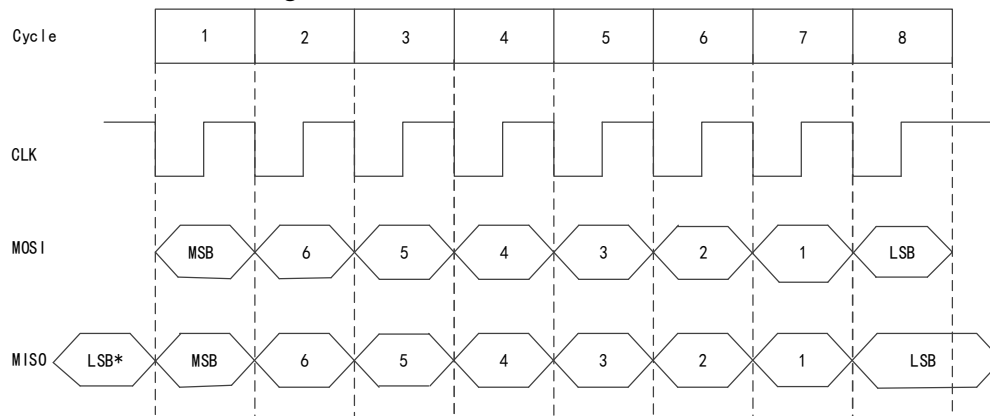
Mode0: RB_SPI_MST_SCK_MOD = 0

Figure 10-1 Transfer format of SPI mode0



Mode3: RB_SPI_MST_SCK_MOD = 1

Figure 10-2 Transfer format of SPI mode3



10.4 SPI Configuration

10.4.1 Master Mode

In SPI master mode, serial clock is generated on SCK pin, and chip select pin can be specified as any I/O pin.

Configuration procedure:

- (1) Set R8_SPIx_CLOCK_DIV, to configure SPI clock frequency;
- (2) Set RB_SPI_MODE_SLAVE in R8_SPIx_CTRL_MOD to 0, to configure SPI to master mode;
- (3) Set RB_SPI_MST_SCK_MOD in R8_SPIx_CTRL_MOD, to select clock idle mode0 or mode3;
- (4) Set the RB_SPI_FIFO_DIR in R8_SPIx_CTRL_MOD to configure the FIFO direction. If it is 1, FIFO is used to receive. If it is 0, FIFO is used to transmit;
- (5) Set RB_SPI_MOSI_OE and RB_SPI_SCK_OE in R8_SPIx_CTRL_MOD to 1, and set RB_SPI_MISO_OE to 0, and set GPIO direction configuration register (R32_PA/PB_DIR), to set the MOSI pin and SCK pin as output, and MISO pin as input;
- (6) SCK remains unchanged in 2-wire mode, RB_SPI_MOSI_OE=0, MOSI is not used. Input (same as 3-wire mode, RB_SPI_MISO_OE=0 and the pin is set as input) and output (RB_SPI_MISO_OE=1 and the pin is set as output) are realized by MISO half-duplex, and the directions are switched manually;
- (7) Optional. If DMA is enabled, it is needed to write the start address of transceiver buffer to R16_SPI_DMA_BEG and write the end address (not included) to R16_SPI_DMA_END. It is recommended

to set RB_SPI_DMA_ENABLE after setting RB_SPI_FIFO_DIR. If R16_SPIx_TOTAL_CNT is confirmed as 0, RB_SPI_DMA_ENABLE can be first set to 1 to enable DMA function.

Data transmission:

- (1) Set RB_SPI_FIFO_DIR to 0, and the current FIFO direction is output;
- (2) Write to the R16_SPIx_TOTAL_CNT register, and set the length of the data to be sent;
- (3) Write to the R8_SPIx_FIFO register and write the data to be sent to FIFO. If R8_SPIx_FIFO_COUNT is less than FIFO capacity, continue to write FIFO. If DMA is enabled, DMA will automatically load FIFO to complete this step;
- (4) As long as R16_SPIx_TOTAL_CNT is not 0 and there is data in FIFO, SPI master will automatically send data, otherwise, it will pause;
- (5) Wait until R16_SPIx_TOTAL_CNT register becomes 0, indicating that the data transmission is completed. If only one byte is sent, you can also query and wait for RB_SPI_FREE to be idle or wait for R8_SPIx_FIFO_COUNT to be 0.

Data reception:

- (1) Set RB_SPI_FIFO_DIR to 1, to set the current FIFO direction to input;
- (2) Write to the R16_SPIx_TOTAL_CNT register, to set the length of the data to be received;
- (3) As long as R16_SPIx_TOTAL_CNT is not 0 and FIFO is not full, SPI master will automatically receive data, otherwise, it will pause;
- (4) Wait until R8_SPIx_FIFO_COUNT register is not 0, indicating that the return data is received, the value read in R8_SPI0_FIFO is the received data. If DMA is enabled, DMA will automatically read FIFO to complete this step.

10.4.2 Slave Mode

SPI0 supports the slave mode. In the slave mode, SCK pin is used to receive the serial clock of SPI master connected to the external.

Configuration procedure:

- (1) Set RB_SPI_MODE_SLAVE in R8_SPI0_CTRL_MOD to 1, to configure SPI0 to slave mode;
- (2) Set RB_SPI_SLV_CMD_MOD in R8_SPI0_CTRL_MOD as needed, to select the slave first byte mode or data stream mode;
- (3) Set RB_SPI_FIFO_DIR in R8_SPI0_CTRL_MOD, to configure the FIFO direction. If it is 1, FIFO is used to receive; if it is 0, FIFO is used to transmit;
- (4) Set RB_SPI_MOSI_OE and RB_SPI_SCK_OE in R8_SPI0_CTRL_MOD to 0, and set RB_SPI_MISO_OE to 1, and set GPIO direction configuration register (R32_PA/PB_DIR) to make MOSI pin, SCK pin and SCS pin as input, MISO pin as input (support connect multiple slaves under the bus; MISO will automatically switch to output after chip select; one master with one slave is also supported) or output (only for connection of one master with one slave). In SPI slave mode, the I/O pin direction of MISO can be set as output by GPIO direction configuration register, it can also automatically switch to output during the period of valid SPI chip select. But its output data is selected by RB_SPI_MISO_OE, it outputs SPI data when it is 1, and it outputs data of GPIO data output register when it is 0. It is recommended to set the MISO pin as input, so that MISO does not output when chip select is invalid, so that SPI bus can be shared during multiple-device operation;
- (5) Optional, set the preset data register (R8_SPI0_SLAVE_PRE) in SPI0 slave mode, used to be automatically loaded into the buffer for the first time after chip select for external output. After 8 clocks (that is, the first

data byte is exchanged between the master and the slave), the controller will obtain the first data byte (command code) sent by the external SPI host, and the external SPI host obtains the preset data (status value) in R8_SPI0_SLAVE_PRE through exchange. The bit7 of R8_SPI0_SLAVE_PRE will be automatically loaded into the MISO pin during SCK low level period after the SPI chip select is valid. For SPI mode 0 (CLK is at low level by default), if the bit7 of R8_SPI0_SLAVE_PRE is preset, the external SPI host will obtain the preset value of bit7 of R8_SPI0_SLAVE_PRE by inquiring the MISO pin when the SPI chip select is valid but has no data transmission, thereby the value of bit7 of R8_SPI0_SLAVE_PRE can be obtained only by a valid SPI chip select (Usually a busy status is provided for the host, so that the host can quickly query);

- (6) Optional. If DMA is enabled, it is needed to write the start address of transceiver buffer to R16_SPI_DMA_BEG and write the end address (not included) to R16_SPI_DMA_END. It is recommended to set RB_SPI_DMA_ENABLE after setting RB_SPI_FIFO_DIR.

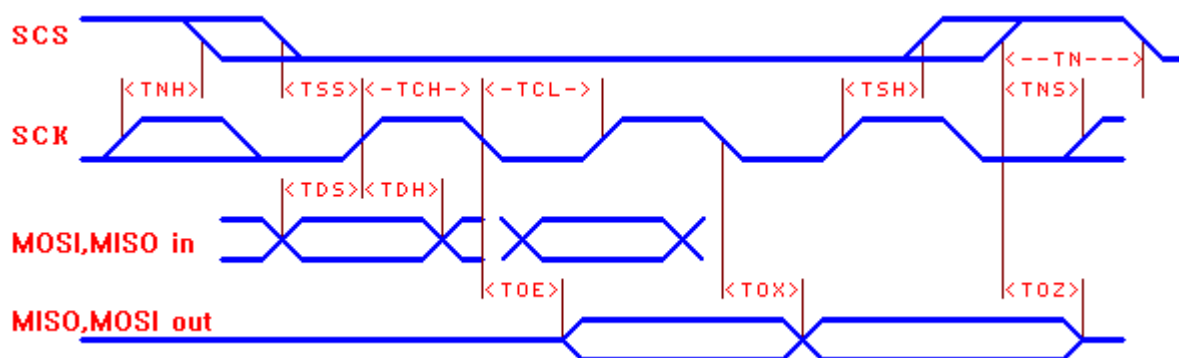
Data transmission:

- (1) Set RB_SPI_FIFO_DIR in R8_SPI0_CTRL_MOD to 0, and the current FIFO direction as output;
- (2) Optional step. If DMA is enabled, it is needed to set RB_SPI_DMA_ENABLE to 1 to enable DMA function;
- (3) Write multiple transmission data into FIFO register R8_SPI0_FIFO, and the external host determines when to take it away. If DMA is enabled, DMA will automatically load FIFO to complete this step;
- (4) Query R8_SPI0_FIFO_COUNT, if it is not full, continue to write data to be sent to FIFO.

Data reception:

- (1) Set RB_SPI_FIFO_DIR of R8_SPI0_CTRL_MOD to 1, and the current FIFO direction as input;
- (2) Optional step. If DMA is enabled, it is needed to set RB_SPI_DMA_ENABLE to 1 to enable DMA function;
- (3) Query R8_SPI0_FIFO_COUNT, if it is not empty, the data has been received and the data will be taken away by reading R8_SPI0_FIFO. If DMA is enabled, DMA will automatically read FIFO to complete this step;
- (4) For reception of the single byte data, R8_SPI0_BUFFER can be read directly without using FIFO.

10.5 SPI Timing



Name	Parameters description (TA=25°C, VIO15=1.5V)	Min.	Typ.	Max.	Unit
TSS	Setup time of valid SCS before SCK rising edge	Tsys*1.05			nS
TSH	Hold time of valid SCS before SCK rising edge	Tsys*1.05			nS
TNS	Setup time of invalid SCS before SCK rising edge	15			nS
TNH	Hold time of invalid SCS before SCK rising edge	15			nS
TN	Time of invalid SCS (interval time of SPI operation)	Tsys*2			nS

TCH	Time of SCK clock at high level	$T_{sys} \times 0.55$			nS
TCL	Time of SCK clock at low level	$T_{sys} \times 0.55$			nS
TDS	Setup time of MOSI/MISO input before SCK rising edge	8			nS
TDH	Hold time of MOSI/MISO input before SCK rising edge	5			nS
TOE	SCK falling edge to MISO/MOSI output valid	0		18	nS
TOX	SCK falling edge to MISO/MOSI output change	0	5	16	nS
TOZ	SCS invalid to MISO/MOSI output invalid	2		24	nS

Note: T_{sys} is the cycle of system clock frequency ($1/F_{sys}$).

Chapter 11 PWM

11.1 Introduction to PWM

In addition to the three 26-bit PWM outputs provided by the timer, the system offers either four 8-bit PWM outputs (PWM4~PWM5, PWM8, PWM10) or six 16-bit PWM outputs (PWM4~PWM5, PWM8), all with adjustable duty cycles. The 8-bit PWM outputs feature fixed selection among 8 preset periods, while the 16-bit PWM outputs allow arbitrary period adjustment, ensuring straightforward operation.

11.2 Register Description

Table 11-1 PWMx registers

Name	Access address	Description	Reset value
R8_PWM_OUT_EN	0x40005000	PWMx output enable register	0x00
R8_PWM_POLAR	0x40005001	PWMx output polarity configuration register	0x00
R8_PWM_CONFIG	0x40005002	PWMx configuration control register	0x0X
R8_PWM_CLOCK_DIV	0x40005003	PWMx clock divider register	0x00
R32_PWM4_7_DATA	0x40005004	PWM4/5 data hold register	0xFFFFFFFF
R8_PWM4_DATA	0x40005004	PWM4 data hold register	0xFF
R8_PWM5_DATA	0x40005005	PWM5 data hold register	0xFF
R32_PWM8_11_DATA	0x40005008	PWM8/9/10/11 data hold register	0xFFFFFFFF
R8_PWM8_DATA	0x40005008	PWM8 data hold register	0xFF
R8_PWM10_DATA	0x4000500A	PWM10 data hold register	0xFF
R8_PWM_INT_CTRL	0x4000500C	PWMx interrupt control and status register	0x00
R32_PWM_REG_DATA8	0x40005010	PWM8/9 data register	0xFFFFFFFF
R32_PWM_REG_CYCLE	0x40005014	PWMx cycle count end register	0x0000XXXX

PWMx Output Enable Register (R8_PWM_OUT_EN)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved	0
6	RB_PWM10_OUT_EN	RW	PWM10 output enable: 1: Enabled; 0: Disabled.	0
5	Reserved	RO	Reserved	0
4	RB_PWM8_OUT_EN	RW	PWM8 output enable: 1: Enabled; 0: Disabled.	0
[3:2]	Reserved	RO	Reserved	0
1	RB_PWM5_OUT_EN	RW	PWM5 output enable: 1: Enabled; 0: Disabled.	0
0	RB_PWM4_OUT_EN	RW	PWM4 output enable: 1: Enabled; 0: Disabled.	0

PWMx Output Polarity Configuration Register (R8_PWM_POLAR)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved	0
6	RB_PWM10_POLAR	RW	PWM10 output polarity control: 1: Default at high level, active low; 0: Default at low level, active high.	0
5	Reserved	RO	Reserved	0
4	RB_PWM8_POLAR	RW	PWM8 output polarity control: 1: Default at high level, active low; 0: Default at low level, active high.	0
[3:2]	Reserved	RO	Reserved	0
1	RB_PWM5_POLAR	RW	PWM5 output polarity control: 1: Default at high level, active low; 0: Default at low level, active high.	0
0	RB_PWM4_POLAR	RW	PWM4 output polarity control: 1: Default at high level, active low; 0: Default at low level, active high.	0

PWMx Configuration Control Register (R8_PWM_CONFIG)

Bit	Name	Access	Description	Reset value
[7:5]	Reserved	RO	Reserved	0
4	RB_PWM4_5_STAG_EN	RW	PWM4/5 interleaved output enable: 1: Interleaved output; 0: Independent output.	0
[3:2]	RB_PWM_CYC_MOD	RW	PWM data width selection: 00: 8-bit data width; 01: 7-bit data width; 00: 6-bit data width; 11: 5-bit data width.	0
1	RB_PWM_STAG_ST	RO	PWM interleave flag: 1: PWM5 is allowed to output; 0: PWM4 is allowed to output.	X
0	RB_PWM_CYCLE_SEL	RW	PWM period selection: 1: 8/7/6-bit data widths corresponding to 255/127/63 clock cycles; 16-bit data width corresponding to clock cycles controlled by R32_PWM_REG_CYCLE. 0: 8/7/6-bit data width corresponding to 256/128/64 clock cycles; The 16-bit data width corresponds to clock cycles controlled by R32_PWM_REG_CYCLE.	0

PWMx Clock Divider Register (R8_PWM_CLOCK_DIV)

Bit	Name	Access	Description	Reset value
[7:0]	R8_PWM_CLOCK_DIV	RW	PWM reference clock frequency division factor. $F_{pwm} = F_{sys} / R8_PWM_CLOCK_DIV$.	0

PWM Data Hold Register Group1 (R32_PWM4_7_DATA)

Bit	Name	Access	Description	Reset value
[15:8]	R8_PWM5_DATA	RW	PWM5 data hold register.	XXh
[7:0]	R8_PWM4_DATA	RW	PWM4 data hold register.	XXh
[31:16]	R16_PWM5_DATA	RW	PWM5 data hold register (16-bit width).	XXh
[15:0]	R16_PWM4_DATA	RW	PWM4 data hold register (16-bit width).	XXh

PWM Data Hold Register Group2 (R32_PWM8_11_DATA)

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RO	Reserved	0
[23:16]	R8_PWM10_DATA	RW	PWM10 data hold register.	XXh
[15:8]	Reserved	RO	Reserved	0
[7:0]	R8_PWM8_DATA	RW	PWM8 data hold register.	XXh

PWMx Interrupt Control and Status Register (R8_PWM_INT_CTRL)

Bit	Name	Access	Description	Reset value
7	RB_PWM_IF_CYC	RW1	PWM cycle end flag. Write 1 to reset: 1: End; 0: Not end.	0
[6:2]	Reserved	RO	Reserved	0
1	RB_PWM_CYC_PRE	RW	Select the interrupt time point at the end of PWM cycle: 1: Interrupt is generated 16 counts in advance (take 8-bit data width as an example, interrupt is generated when the count reaches 240); 0: Interrupt is generated 2 counts in advance (take 8-bit data width as an example, interrupt is generated when the count reaches 254).	0
0	RB_PWM_IE_CYC	RW	PWM cycle end interrupt enable: 1: Enabled; 0: Disabled.	0

PWM8 Data Register (R32_PWM_REG_DATA8)

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:0]	R16_PWM8_REG_DATA	RW	16-bit bit data of PWM8 channel.	XXXXh

PWM Cycle Count End Register (R32_PWM_REG_CYCLE)

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0

[15:0]	RB_PWM_CYC_VALUE	RW	PWM clock period at 16 data width: Ncyc=RB_PWM_CYC_VALUE+1.	XXXXh
--------	------------------	----	--	-------

11.3 PWM Configuration

- (1) Set the R8_PWM_CLOCK_DIV register, to configure the reference clock frequency of PWM;
- (2) Set the PWM output polarity configuration register (R8_PWM_POLAR), to configure the output polarity of the corresponding PWMx;
- (3) Set the PWM configuration control register (R8_PWM_CONFIG), to set the PWM mode, data width and cycle;
- (4) Set PWM output enable register (R8_PWM_OUT_EN), to enable the corresponding PWMx output;
- (5) Calculate the data according to the required duty cycle and write it into the corresponding data hold register (R8_PWMx_DATA);
- (6) Set the required PWM pin direction of PWM4~PWM5, PWM8, PWM10 as output, optional, set the drive capability of corresponding I/O;
- (7) Update the data in R8_PWMx_DATA as needed, to update the output duty cycle.

Number of clock cycles (Ncyc) calculation formula:

data width n=8/7/6: $Ncyc=2^n-RB_PWM_CYCLE_SEL$;

16-bit data width: $Ncyc=RB_PWM_CYC_VALUE+1$

PWMx duty cycle calculation formula:

duty cycle= $R8_PWMx_DATA/Ncyc$

PWMx output frequency Fpwmout calculation formula:

PWMx output frequency $Fpwmout = Fpwm/Ncyc = Fsys / R8_PWM_CLOCK_DIV/Ncyc$

Note: If a DC signal needs to be generated by PWM, then the PWMx output can be filtered using a circuit such as R/C. It is recommended to use a two-stage RC with a time constant much greater than $4/Fpwmout$, or a one-stage RC with a time constant much greater than $100/Fpwmout$.

Chapter 12 Inter Integrated Circuit (I2C) Interface

12.1 Introduction to I2C

I2C (inter-integrated circuit) bus interface is a medium-low-speed serial bus. Multiple masters and slaves can be connected on the bus. Usually, I2C interface consists of 2 pins: serial clock pin (SCL) and serial data pin (SDA).

12.1.1 Main Features

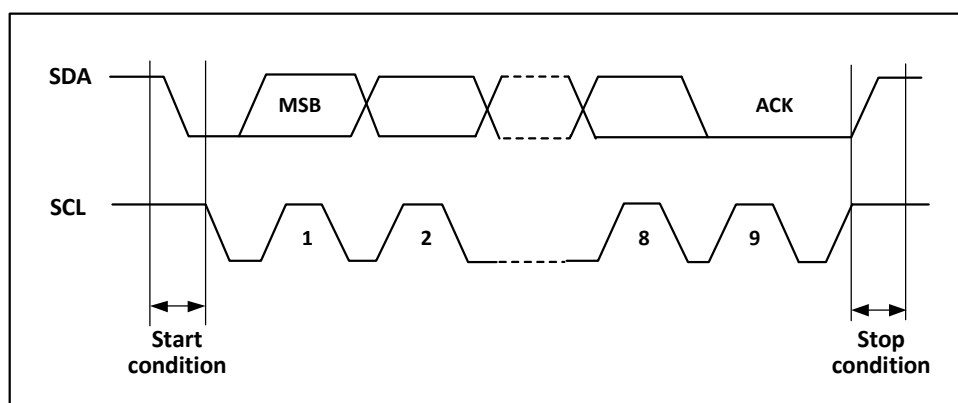
The chip provides an I2C interface with the following characteristics:

- Master mode and Slave mode, support multi-master and multi-slave.
- 2 speeds: 100KHz and 400KHz, compatible with I2C Bus Specification.
- 7-bit and 10-bit addressing.
- The slave device supports dual 7-bit addresses.
- Broadcast bus.
- Bus arbitration, error detection, PEC verification, clock extension.
- SMBus compatibility.

12.2 I2C Overview

I2C is a half-duplex bus and can operate in one of the 4 following modes: master transmitter, master receiver, slave transmitter and slave receiver. By default, I2C operates in slave mode. After I2C generates a ATART condition, it automatically switches from slave to master. After an arbitration is lost or a Stop signal generates, it switches to slave. I2C supports multi-master mode. When it operates in master mode, I2C transmits data and addresses actively. Data and addresses are transferred as 8-bit bytes, high bits first, low bits last. After a Start event, it is 1-byte (in 7-bit mode) or 2-byte (in 10-bit mode). Every time the master transmits 8-bit data or address, the slave needs to respond one ACK, that is, pull down SDA bus, as shown in Figure 12-1.

Figure 12-1 I2C timing diagram



12.3 Master Mode

In master mode, I2C dominates a data transmission and generates the clock signal. A data transmission always begins with a Start condition, and ends with a Stop condition. The following is the step sequence in master mode:

- (1) Set the correct timings in the control register2 (R16_I2C_CTRL2) and clock control register (R16_I2C_CKCFGR);
- (2) Set suitable rise time in rise time register (R16_I2C_RTR);
- (3) Set the PE bit to enable peripheral in the control register (R16_I2C_CTRL1);
- (4) Set the START bit in the control register (R16_I2C_CTRL1), to generate a Start condition.

After the START bit is set, I2C automatically switch to master mode, MSL bit is set, a Start condition is generated. After a Start condition is generated, SB bit will be set. If ITEVTEN bit (in R16_I2C_CTRL2) is set, an interrupt will be generated. At this time, it is required to read the status register1 (R16_I2C_STAR1), write slave address to the data register, SB bit will be automatically cleared;

- (5) If in 10-bit addressing mode, write the data register to send the header sequence (header sequence is 11110xx0b, where xx denotes the highest 2 bits of 10-bit address).

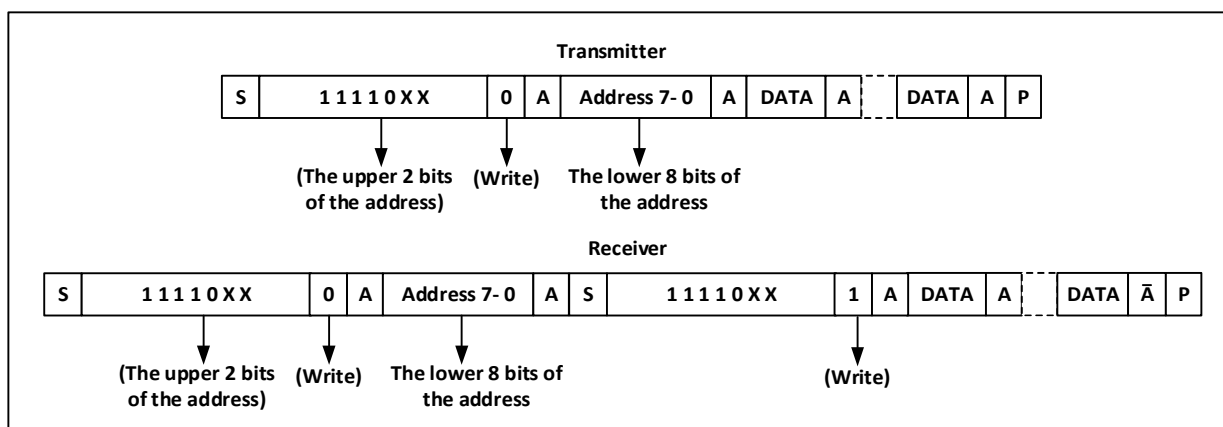
After the header sequence is sent, the ADD10 bit of the status register will be set. If ITEVTEN bit is set, an interrupt will be generated. At this time, it is required to read R16_I2C_STAR1 register, then write the second address byte to the data register, to clear ADD10 bit.

Then, write the data register to send the second address byte. After the second address byte is sent, ADDR bit of the status register will be set. If ITEVTEN bit is set, an interrupt will be generated. At this time, it is required to read R16_I2C_STAR1 register and then read R16_I2C_STAR2 register to clear ADDR bit;

If in 7-bit addressing mode, write data register to send address byte. After the address byte is sent, ADDR bit of the status register will be set. If ITEVTEN bit is set, an interrupt will be generated. At this time, it is required to read R16_I2C_STAR1 register and then read R16_I2C_STAR2 register to clear ADDR bit; In 7-bit addressing mode, the first sent byte is the address byte. The first 7 bits represent target slave device address, and the 8th bit determines the direction of the following message. 0 means that the master writes data to the slave, and 1 means that the master reads data from the slave.

In 10-bit addressing mode, as shown in Figure 12-2, when transmitting address, the first byte is 11110xx0, where xx denotes the highest 2 bits of the 10-bit address. The second byte is the lower 8 bits of the 10-bit address. If going into master transmitter mode later, continue to transmit data. If going into master receiver mode later, it is required to send a repeated Start condition, followed by the header (11110xx1), then enter master receiver mode.

Figure 12-2 Master transceiver data in 10-bit mode



- (1) For transmitter, the internal shift register of the master transmits data from the data register to SDA. When the master receives ACK, the TxE bit of the status register1 (R16_I2C_STAR1) is set. If ITEVTEN bit and ITBUFEN bit are set, an interrupt will be generated. Write data to the data register to clear TxE bit.

If TxE bit is set and no data is written to the data register before last data transmission, BTF bit will be set. SCL is kept at low level before it is cleared. Read R16_I2C_STAR1, and then write data into the data register, to clear BTF bit.

For receiver, I2C receives data from SDA, and writes data to the data register via the shift register. After each byte, if ACK bit is set, I2C will send a response low level, and RxNE bit will be set. If ITEVTEN bit and ITBUFEN bit are set, an interrupt will be generated. If RxNE is set and no original data is read out before new data reception, BTF bit will be set. Before BTF is cleared, SCL will be kept at low level. Read R16_I2C_STAR1, then read the data register, to clear BTF bit.

- (2) The master automatically sends a Stop condition when it ends sending data, that is, STOP bit is set. For receiver, the master is required to set NAK at the response bit of the last data bit. Note, after NAK is generated, I2C will automatically switch to slave mode.

12.4 Slave Mode

In slave mode, I2C can recognize its address and general call address. The recognition of general call address can be enabled or disabled by software. Once a Start condition is detected, I2C will compare the data of SDA with its own address via the shift register (bit number is determined by ENDUAL and ADDMODE) or general address (ENGC bit is set). If not matched, the interface ignores it and waits for another Start condition. If the header is matched, the ACK signal generates and waits for the address of the second byte. If the address of the second byte is matched or the full address is matched in 7-bit addressing mode, the ACK is generated firstly; the ADDR bit is set, if ITEVTEN bit has been set, the corresponding interrupt will be generated. If in dual-address mode (ENDUAL bit is set), it is required to read DUALF bit to judge which address is woken up by the master.

For slave mode, it is receiver mode by default. When the last bit of the received header is 1, or when the last bit of the 7-bit address is 1 (the first received is the header or normal 7-bit address), I2C will go into transmitter mode, TRA bit indicates that it is in receiver mode or in transmitter mode.

In transmitter mode, after ADDR bit is cleared, I2C transmits byte from the data register to SDA via the shift register. After the ACK is received, TxE bit will be set. If ITEVTEN bit and ITBUFEN bit are set, an interrupt will be generated. If TxE is set but no new data is written to the data register before the end of next data transmission, BTF bit will be set. Before BTF is cleared, SCL will be kept at low level. Read the status register1 (R16_I2C_STAR1), then write data to the data register, to clear BTF bit.

In receiver mode, after ADDR is cleared, I2C stores data in SDA to the data register via the shift register. Every time a byte is received, I2C will set ACK bit, and set RxNE bit. If ITEVTEN and ITBUFEN are set, an interrupt will be generated. If RxNE is set, and no previous data is read out before reception of new data, BTF bit will be set. Before BTF bit is cleared, SCL will be kept at low level. Read the status register1 (R16_I2C_STAR1), and read the data in the data register, to clear BTF bit.

When I2C detects a Stop condition, STOPF bit will be set. If ITEVFEN bit is set, an interrupt will be generated. Read the status register (R16_I2C_STAR1), then write to the control register (such as reset control word SWRST) to clear.

12.5 Error Conditions

12.5.1 Bus Error (BERR)

The bus error occurs when I2C interface detects an external Stop or Start condition during an address or a data transfer. When a bus error occurs, the BERR bit is set. And an interrupt is generated if the ITERREN is set. In slave mode, data are discarded, and the lines are released by hardware. If it is a Start condition, the slave considers it is a restart, and waits for an address or a Stop condition. If it is a Stop condition, the slave behaves like for a Stop condition. In master mode, lines are not released by hardware and the state of the current transmission is not affected. The user codes determine to abort or not the current transmission.

12.5.2 Acknowledge Failure (AF)

When I2C interface detects a non-acknowledge bit, the acknowledge failure occurs. When it occurs, the AF bit is set, and an interrupt is generated if the ITERREN bit is set. When AF occurs, if I2C interface works in slave mode, the lines must be released by hardware; if in master mode, a Stop condition must be generated by software.

12.5.3 Arbitration Lost (ARLO)

When I2C interface detects an arbitration lost condition, it occurs. When an arbitration lost occurs, the ARLO bit is set, and an interrupt is generated if the ITERREN bit is set. I2C switches to slave mode, and does not acknowledge its slave address in the same transfer, unless the master initiates a new Start condition. Lines are released by hardware.

12.5.4 Overload/Underload Error (OVR)

1) Overrun error:

In slave mode, when clock stretching is disabled and I2C interface is receiving data, if a byte is received and the last received data has not been read, it occurs. When it occurs, the last received byte is lost, and the transmitter should re-transmit the last received byte.

2) Underrun error:

In slave mode, when clock stretching is disabled and I2C interface is transmitting data, if no new data is written to the data register before the next byte, it occurs. When it occurs, the data in the last data register will be sent again. If it occurs, the receiver should discard the repeated received data. In order to not generate underrun error, I2C interface should write the data to the data register before the first rising edge of the next byte.

12.6 Clock Stretching

If clock stretching is disabled, overrun/underrun error may occur. However, if clock stretching is enabled:

- 1) In transmitter mode, if the TxE bit is set and the BTF bit is set, SCL is kept at low level and waits for user to read the status register, and write the data to be sent into the data register;
- 2) In receiver mode, if the RxNE bit is set and the BTF bit is set, SCL is kept at low level after data is received, until user reads the status register and read the data register;

So, enabling clock stretching helps avoid overrun/underrun error.

12.7 SMBus

The System Management Bus (SMBus) is a 2-wire interface, which is usually used between system and power management. SMBus has multiple similarities with I2C, for example, SMBus uses the same 7-bit addressing mode as I2C. The following is similarities between SMBus and I2C:

- 1) Master-slave communication, Master provides clock, multi-master multi-slave is supported;
- 2) Two-wire communication protocol;
- 3) 7-bit addressing format.

Differences:

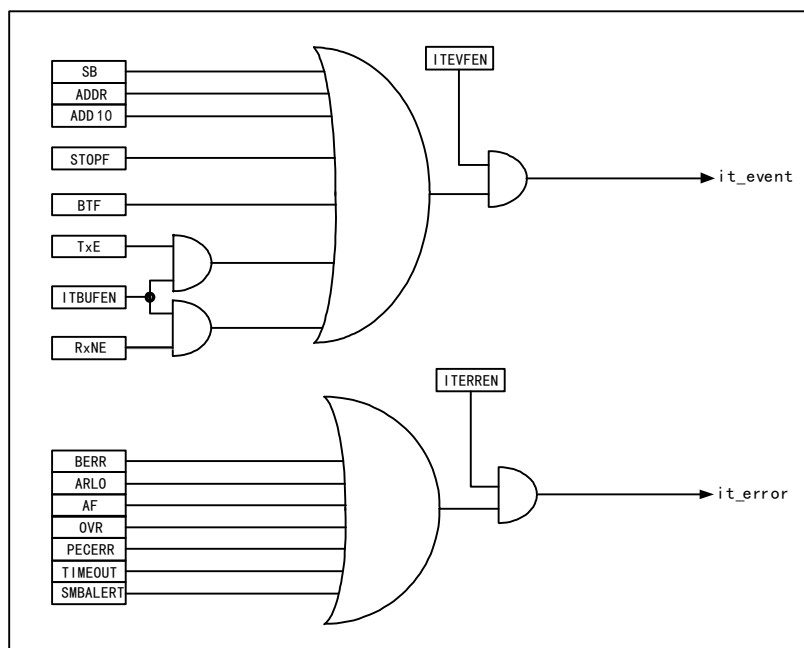
- 1) The speed of I2C can be up to 400KHz. The speed of SMBus can only be up to 100KHz, and the minimum speed of SMBus is 10KHz;
- 2) SMBus: 35 mS clock low timeout. I2C: no timeout;
- 3) SMBus has fixed logic levels. While I2C does not have fixed logic levels, which are determined by VDD;
- 4) SMBus has bus protocols, while I2C does not have bus protocols.

SMBus also contains device identification, address resolution protocol, unique device identifier, SMBus alert and various bus protocols, please refer to SMBus specification version 2.0 for details. When SMBus is used, set the SMBus bit of the control register, and configure the SMBTYPE bit and the ENAARP bit as required.

12.8 Interrupt

I2C interface provides event interrupt and error interrupt. After going into the same interrupt service program, they are processed separately after query.

Figure 12-3 I2C interrupt requests



12.9 Packet Error Checking (PEC)

Packet Error Checking (PEC) is an additional CRC8 check step to provide transmission reliability and is calculated for each bit of serial data using the following polynomial: $C=X^8+X^2+X+1$.

PEC calculation is enabled by the ENPEC bit of the control register, and calculated on all message bytes, including addressed and Read/Write bits. In transmission, if PEC is enabled, the CRC8 calculation result of the

last byte is added after the last byte data. In reception, the last byte is considered as the CRC8 calculation result. If it is not matched with the internal calculation result, a NAK is sent. In case of master receiver, a NAK follows the PEC whatever the check result.

12.10 Register Description

Table 12-1 I2C-related registers

Name	Access address	Description	Reset value
R16_I2C_CTRL1	0x40004800	I2C control register 1	0x0000
R16_I2C_CTRL2	0x40004804	I2C control register 2	0x0000
R16_I2C_OADDR1	0x40004808	I2C address register 1	0x0000
R16_I2C_OADDR2	0x4000480C	I2C address register 2	0x0000
R16_I2C_DATAR	0x40004810	I2C data register	0x0000
R16_I2C_STAR1	0x40004814	I2C status register 1	0x0000
R16_I2C_STAR2	0x40004818	I2C status register 2	0x0000
R16_I2C_CKCFGR	0x4000481C	I2C clock register	0x0000
R16_I2C_RTR	0x40004820	I2C rise time register	0x0002

I2C Control Register 1 (R16_I2C_CTRL1)

Bit	Name	Access	Description	Reset value
15	RB_I2C_SWRST	RW	Software reset. When user codes set this bit, I2C is reset. Before reset, make sure that IIC lines are released and the bus is idle. <i>Note: This bit can reset I2C when no Stop condition is detected on the bus but the busy bit is 1.</i>	0
14	Reserved	RO	Reserved	0
13	RB_I2C_ALERT	RW	SMBus alert. This bit can be set or cleared by user codes. After the PE bit is set, this bit can be cleared by hardware. 1: Drive SMBusALERT pin low, response address header followed by ACK; 0: Release SMBusALERT pin high, response address header followed by NACK.	0
12	RB_I2C_PEC	RW	Data packet error checking enable, it is set to enable data packet error checking detection. This bit can be set or cleared by user codes. After PEC is transmitted, or when a Start or Stop condition is generated, or when the PE bit is cleared to 0, this bit is cleared by hardware. 1: PEC; 0: No PEC. <i>Note: PEC is corrupted when an arbitration is lost.</i>	0
11	RB_I2C_POS	RW	ACK/PEC position. This bit can be set or cleared by user codes. Or this bit can be cleared by hardware	0

			after PE is cleared. 1: The ACK bit controls ACK/NAK of the next byte that is received in the shift register. The next byte that is received in PEC shift register is a PEC; 0: The ACK bit controls ACK/NAK of the current byte being received in the shift register. The PEC bits indicates that current byte in the shift register is a PEC. <i>Note: The POS bit is used for the reception of 2bytes data: it must be configured before reception. To NACK the second byte, the ACK bit must be cleared at once after the ADDR bit is cleared. To check the second byte as PEC, the PEC bit must be set after ADDR event occurs and the POS bit is configured.</i>	
10	RB_I2C_ACK	RW	Acknowledge enable. This bit can be set or cleared by user codes. Or this bit can be cleared by hardware after PE is set. 1: Acknowledge returned after a byte is received; 0: No acknowledge.	0
9	RB_I2C_STOP	RW	Stop condition generation. This bit can be set or cleared by user codes. Or this bit can be cleared by hardware when a Stop condition is detected. Or this bit can be set by hardware when a timeout error is detected. In master mode: 1: A Stop condition is generated during current byte transmission or after current Start condition is sent; 0: No Stop condition generated. In slave mode: 1: Release SCL and SDA after current byte transmission; 0: No Stop condition generated.	0
8	RB_I2C_START	RW	Start condition generation. This bit can be set or cleared by user codes. Or this bit can be cleared by hardware when a Start condition is sent or PE is cleared. In master mode: 1: Repeated Start condition generation; 0: No Start condition generation. In slave mode: 1: A Start condition is generated when the bus is idle; 0: No Start condition generated.	0
7	RB_I2C_NOSTRETCH	RW	Clock stretching disable. This bit can be used to disable clock stretching when ADDB or BTF flag is	0

			set, until it is cleared by software. 1: Disable clock stretching; 0: Enable clock stretching.	
6	RB_I2C_ENGC	RW	General call enable. This can be set to enable general call. Response general address 00h.	0
5	RB_I2C_ENPEC	RW	PEC enable. This bit can be set to enable PEC calculation.	0
4	RB_I2C_ENARP	RW	ARP enable. This bit can be set to enable ARP. If SMBTYPE=0, use default SMBus device address; If SMBTYPE=1, use SMBus host address.	0
3	RB_I2C_SMBTYPE	RW	SMBus device type. 1: SMBus host; 0: SMBus device.	0
2	Reserved	RO	Reserved	0
1	RB_I2C_SMBUS	RW	SMBus mode selection: 1: SMBus mode; 0: I2C mode.	0
0	RB_I2C_PE	RW	I2C peripheral enable. 1: I2C enabled; 0: I2C disabled.	0

I2C Control Register 2 (R16_I2C_CTRL2)

Bit	Name	Access	Description	Reset value
[15:11]	Reserved	RO	Reserved.	0
10	RB_I2C_ITBUFEN	RW	Buffer interrupt enable. 1: When TxEN or RxEN is set, event interrupt is generated; 0: When TxEN or RxEN is set, no interrupt is generated.	0
9	RB_I2C_ITEVTEN	RW	Event interrupt enable. This bit can be set to enable event interrupt. The interrupt is generated when: SB=1 (Master); ADDR=1 (Master/Slave); ADDR10=1 (Master); STOPF=1 (Slave); BTF=1, with no TxEN or RxEN event; If ITBUFEN=1, TxEN event to 1; If ITBUFEN=1, RxNE event to 1.	0
8	RB_I2C_ITERREN	RW	Error interrupt enable. This bit can be set to enable error interrupt. The interrupt is generated when: BERR=1; ARLO=1; AF=1; OVR=1; PECERR=1; TIMEOUT=1; SMBAlert=1.	0
[7:6]	Reserved	RO	Reserved.	0

[5:0]	RB_I2C_FREQ	RW	I2C clock frequency. The correct clock frequency must be input to generate correct timing. The allowed range is 2~36MHz. These bits must be set between 000010b and 100100b, and the unit is MHz. Recommended: Minimum 2MHz input clock in standard mode; minimum 4MHz input clock in fast mode.	0
-------	-------------	----	---	---

I2C Address Register1 (R16_I2C_OADDR1)

Bit	Name	Access	Description	Reset value
15	RB_I2C_ADDMODE	RW	Addressing mode. 1: 10-bit slave address (7-bit address not acknowledged); 0: 7-bit slave address (10-bit address not acknowledged).	0
[14:10]	Reserved	RO	Reserved	0
[9:8]	RB_I2C_ADD9_8	RW	Bus address. Bits9:8 when using 10-bit address. Ignore when using 7-bit address.	0
[7:1]	RB_I2C_ADD7_1	RW	Bus address, bits7-1.	0
0	RB_I2C_ADD0	RW	Bus address. Bit0 when using 10-bit address. Ignore when using 7-bit address.	0

I2C Address Register2 (R16_I2C_OADDR2)

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved.	0
[7:1]	RB_I2C_ADD2	RW	Bus address, bits7-1 in dual addressing mode.	0
0	RB_I2C_ENDUAL	RW	Dual addressing mode enable bit. This bit can be set to make ADD2 recognized.	0

I2C Data Register (R16_I2C_DATAR)

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved.	0
[7:0]	RB_I2C_DATAR	RW	Data register. It can be used to store the received data or store the data to be sent to the bus.	0

I2C Status Register 1 (R16_I2C_STAR1)

Bit	Name	Access	Description	Reset value
15	RB_I2C_SMBALERT	RW0	SMBus alert. This bit can be reset by user codes writing 0, or reset by hardware when PE is at low level. In SMBus host mode: 1: SMBus alert occurs; 0: No SMBus alert. In SMBus slave mode: 1: SMBAlert response address header to SMBAlert LOW received; 0: No SMBAlert response address header received.	0
14	RB_I2C_TIMEOUT	RW0	Timeout or Tlow error flag. This bit can be reset by user codes writing 0, or reset by hardware when PE is at low level. 1: SCL remained LOW for 25mS, or master low extend time more than 10mS, or slave low extend time more than 25mS; 0: No timeout error. <i>Note: When this bit is set in slave mode, slave resets communication, lines are released by hardware. When this bit is set in host mode, a Stop condition is sent by hardware.</i>	0
13	Reserved	RO	Reserved.	0
12	RB_I2C_PECERR	RW0	PEC error flag in reception. This bit can be reset by user codes writing 0, or reset by hardware when PE is at low level. 1: PEC error, NAK returned after PEC is received; 0: No PEC error.	0
11	RB_I2C_OVR	RW0	Overflow/Underflow flag. 1: Overflow/Underflow: If NOSTRETCH=1, when a new byte is received in reception and data register has not been read, the new received byte is lost. In transmission, when no new data is written into data register, the same byte is sent twice; 0: No overflow/underflow.	0
10	RB_I2C_AF	RW0	Acknowledge failure flag. This bit can be reset by user codes writing 0, or reset by hardware when PE is at low level. 1: Acknowledge failure; 0: No acknowledge failure.	0

9	RB_I2C_ARLO	RW0	Arbitration lost flag. This bit can be reset by user codes writing 0, or reset by hardware when PE is at low level. 1: Arbitration lost detected, the interface loses the control to the bus; 0: Arbitration is normal.	0
8	RB_I2C_BERR	RW0	Bus error flag. This bit can be reset by user codes writing 0, or reset by hardware when PE is at low level. 1: Start/Stop condition error; 0: Normal.	0
7	RB_I2C_TxE	RO	Data register empty flag. It can be cleared by writing data to the data register, or it can be cleared by hardware after a Start/Stop condition is generated or when PE is 0. 1: Data register empty in transmission; 0: Data register not empty.	0
6	RB_I2C_RxNE	RO	Data register not empty flag. It can be cleared by reading/writing data to the data register, or it can be cleared by hardware when PE is 0. 1: Data register not empty in reception; 0: Normal.	0
5	Reserved	RO	Reserved.	0
4	RB_I2C_STOPF	RO	Stop condition flag. It can be cleared by write operation to the control register1 after user reads the status register1, or it can be cleared by hardware when PE is 0. 1: Slave detects a Stop condition on the bus after an acknowledge; 0: No Stop condition detected.	0
3	RB_I2C_ADD10	RO	10-bit address header sent flag. It can be cleared by write operation to the control register1 after user reads the status register1, or it can be cleared by hardware when PE is 0. 1: In 10-bit addressing mode, slave has sent the first address byte; 0: None.	0

2	RB_I2C_BTFF	RO	Byte transmission finished flag. It can be cleared by read/write operation to the data register after user reads the status register1. In transmission, it can be cleared by hardware after a Start/Stop condition is generated or when PE is 0. 1: Byte transmission finished. If NOSTRETCH=0: in transmission, when a new data is sent and no new data is written into the data register; in reception, when a new byte is received but the data register has not been read; 0: None.	0
1	RB_I2C_ADDR	RW0	Address sent/matched flag. It can be cleared by read operation to the status register2 after user reads the status register1, or it can be cleared by hardware when PE is 0. In master mode: 1: Address transmission finished: in 10-bit addressing mode, this bit is set after ACK of the second address byte is received; in 7-bit addressing mode, this bit is set after ACK of the byte is received; 0: Address transmission not finished. In slave mode: 1: Received address matched; 0: Address not matched or no address received.	0
0	RB_I2C_SB	RO	Start bit sent flag. It can be cleared by write operation to the data register after user reads the status register1, or it can be cleared by hardware when PE is 0. 1: Start bit sent; 0: Start bit not sent.	0

I2C Status Register 2 (R16_I2C_STAR2)

Bit	Name	Access	Description	Reset value
[15:8]	RB_I2C_PEC	RO	Packet error checking register. When PEC is enabled (ENPEC is set), this register stores the value of PEC.	0
7	RB_I2C_DUALF	RO	Dual flag. It can be cleared by hardware when a Stop/Start bit is generated or when PE=0. 1: Received address matched with OADDR2; 0: Received address matched with OADDR1.	0

6	RB_I2C_SMBHOST	RO	SMBus host header flag. It can be cleared by hardware when a Stop/Start bit is generated or when PE=0. 1: When SMBTYPE=1 and ENARP=1, SMBus host address received; 0: SMBus host address not received.	0
5	RB_I2C_SMBDEFAULT	RO	SMBus device default address flag. It can be cleared by hardware when a Stop/Start bit is generated or when PE=0. 1: When ENARP=1, SMBus device default address received; 0: SMBus device default address not received.	0
4	RB_I2C_GENCALL	RO	General call address flag. It can be cleared by hardware when a Stop/Start bit is generated or when PE=0. 1: When ENGC=1, general call address received; 0: General call address not received.	0
3	Reserved	RO	Reserved.	0
2	RB_I2C_TRA	RO	Transmission/reception flag. It can be cleared by hardware when a Stop condition is detected (STOPF=1) or repeated Start condition is detected or an arbitration lost is detected (ARLO=1) or when PE=0. 1: Data transmitted; 0: Data received. This bit is determined by the R/W bit of the address byte.	0
1	RB_I2C_BUSY	RO	Bus busy flag. It can be cleared when a Stop condition is detected. When the interface is disabled (PE=0), the information is still updated. 1: Bus busy: SDA or SCL is low; 0: Bus idle with no communication.	0
0	RB_I2C_MSL	RO	Master/slave mode indication. It can be cleared by hardware when the interface is in Master mode (SB=1). It can be cleared by hardware when the bus detects a Stop bit or an arbitration lost or when PE=0.	0

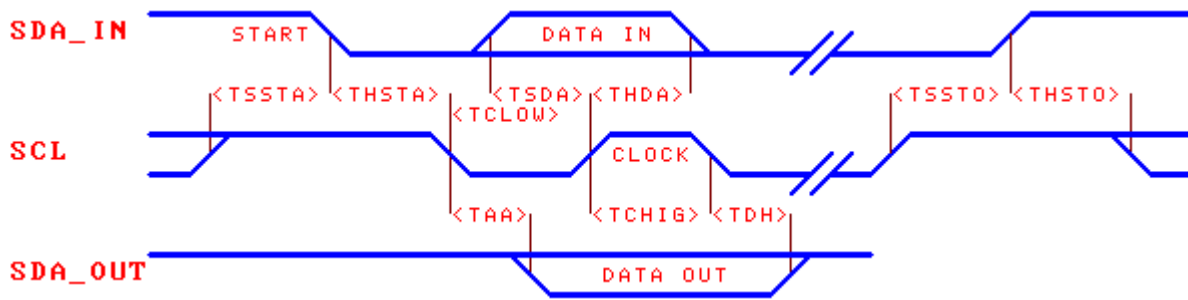
I2C Clock Register (R16_I2C_CKCFGR)

Bit	Name	Access	Description	Reset value
15	RB_I2C_F/S	RW	Master mode selection: 1: Fast mode; 0: Standard mode.	0
14	RB_I2C_DUTY	RW	Clock HIGH duty cycle in fast mode: 1: 36%; 0: 33.3%.	0
[13:12]	Reserved	RO	Reserved.	0
[11:0]	RB_I2C_CCR	RW	Clock frequency division factor register, which decides frequency wave of SCL clock. In I2C standard mode or SMBus mode: $T_{high} = CCR * THCLK$ $T_{low} = CCR * THCLK$ In I2C fast mode: If DUTY = 0: $T_{high} = CCR * T_{HCLK}$ $T_{low} = 2 * CCR * T_{HCLK}$ If DUTY = 1: (Speed reaches 400kHz) $T_{high} = 9 * CCR * T_{HCLK}$ $T_{low} = 16 * CCR * T_{HCLK}$ Example: In standard mode, to generate a 100kHz SCL frequency: $CCR = FREQR / (2 * 100kHz)$. When $FREQR = 48MHz$, $THCLK = 1/48MHz$. Thus $CCR = 48MHz / (2 * 100kHz) = 240$. Write 0xF0 to CCR.	0

I2C Rise Time Register (R16_I2C_RTR)

Bit	Name	Access	Description	Reset value
[15:6]	Reserved	RO	Reserved.	0
[5:0]	RB_I2C_TRISE	RW	Maximum rise time. These bits set the rise time of SCL in master mode. The maximum rise time=TRISE–single clock cycle. This bit can be set only when PE is cleared.	000010b

12.11 I2C Timing



Name	Parameter description (TA=25°C, VIO15=1.5V)	Min.	Typ.	Max.	Unit
TSSTA	Setup time of SCL HIGH before SDA falling edge	0.4			uS
THSTA	Hold time of SCL HIGH after SDA falling edge	0.4			uS
TSDA	Setup time of SDA data before SCL rising edge	0.05			uS
THDA	Hold time of SDA data after SCL rising edge	>TCHIG			uS
TSSTO	Setup time of SCL HIGH before SDA rising edge	0.4			uS
THSTO	Hold time of SCL HIGH after SDA rising edge	0.8			uS
TCLOW	Time of SCL clock LOW	0.6			uS
TCHIG	Time of SCL clock HIGH	0.5			uS
TAA	SCL falling edge to SDA output valid	0.006		0.4	uS
TDH	SDA output hold time after SCL falling edge	0.006			uS
TR	SCL/SDA input rising edge time			0.2	uS

Chapter 13 Independent Watchdog (IWDG)

13.1 Introduction to IWDG

Independent watchdog (IWDG) is driven by a dedicated internal low-speed clock (LSI).

13.2 Register Description

Table 13-1 IWDG registers

Name	Access address	Description	Reset value
R32_IWDG_KR	0x40001000	IWDG key register	0xFFFFFFFF
R32_IWDG_CFG	0x40001004	IWDG configuration register	0x4FFF0000

IWDG Key Register (R32_IWDG_KR)

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	WO	Reserved	XXXXh
[15:0]	IWDG_KR	WO	KEY[15:0]: key value (write register only, read out value is 0x0000) The software must write 0xAAAA at certain intervals to reload the count value, otherwise, when the counter is 0, the watchdog will generate a reset, and writing 0x5555 indicates that the protection is lifted. Write 0xCCCC to start the watchdog work (if the hardware watchdog is selected, it is not restricted by this command word)	XXXXh

IWDG Configuration Register (R32_IWDG_CFG)

Bit	Name	Access	Description	Reset value
31	IWDG_EN	RO	Watchdog start switch: 1: on; 0: off.	0
30	WR_PROTECT	RO	Write protection: 1: Prohibit the operation of the corresponding field; 0: Unprotected.	1
29	STOP_EN	RW	Watchdog stop enable (write protection exists) 1: Turn on the stop switch; 0: Turn off the stop switch.	0
28	Reserved	RO	Reserved	0
[27:16]	COUNT	RO	Watchdog decrement counter	FFFh
15	PVU	RO	Configuration register update flag bit (write protection exists)	X

			1: Register is updated; 0: Register is not updated.	
[14:12]	PR	RW	Prescaling factor (write protection exists) 000: Divided by 4; 001: Divided by 8; 010: Divided by 16; 011: Divided by 32; 100: Divided by 64; 101: Divided by 128; 110: Divided by 256; 111: Divided by 512.	0
[11:0]	RLR	RW	RL[11:0]: Watchdog counter reload (with write protection) Used to define the reload value of the watchdog counter. Whenever 0xAAAA is written to the IWDG_KR register, the reload value is transferred to the counter. The counter then counts decrementally from this value. The watchdog timeout period can be calculated from this reload value and the clock prescaler value.	FFFh

Chapter 14 Analog-to-digital Converter (ADC)

14.1 Introduction to ADC

The chip provides a 12-bit successive approximation analog to digital converter (ADC) and up to 9 channels, supports channel scanning function. Supports 8 external signal sources and 1 internal signal sources.

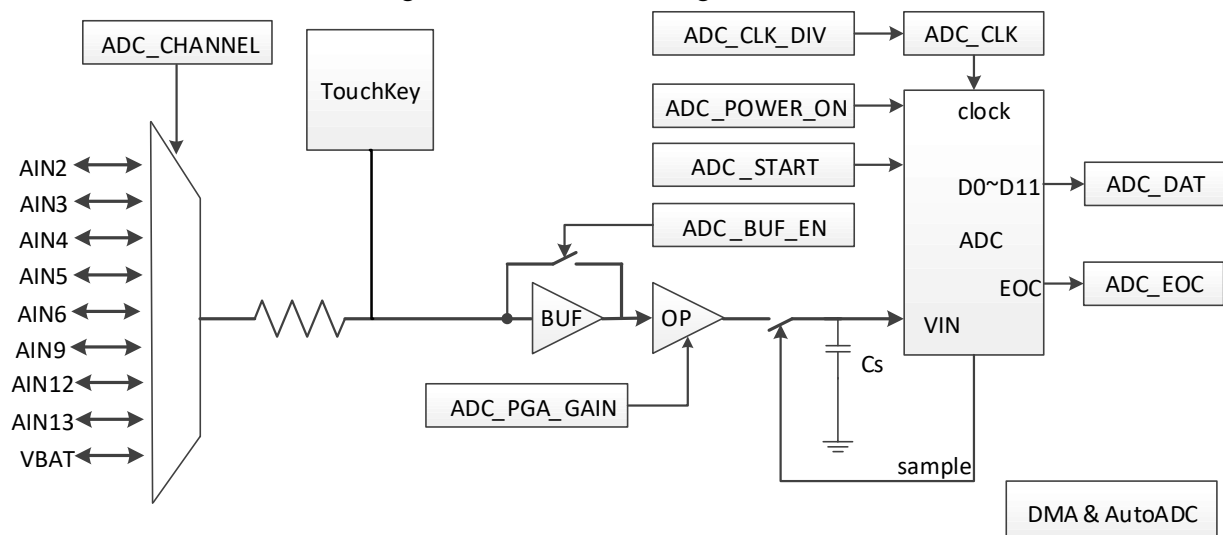
14.1.1 Main Features

- 12-bit resolution.
- 8 external voltage sampling channels and internal battery voltage detection channels.
- Single-ended input mode.
- Optional sampling clock frequency.
- ADC input voltage range: 0V~VIO15.
- Optional PGA: provide gain adjustment options.
- Optional input buffer BUF: support high resistance signal source.
- Support DMA and automatic continuous ADC sampling in timing interval, with adjustable interval.
- Support the channel scanning function, which allows you to configure the order of scanning, the number of channels to be scanned, and provides an end-of-scan interrupt.

14.1.2 Functional Description

The figure below is the block diagram of an ADC module.

Figure 14-1 Architecture diagram of ADC



14.2 Register Description

Table 14-1 ADC registers

Name	Access address	Description	Reset value
------	----------------	-------------	-------------

R8_ADC_CHANNEL	0x40001058	ADC input channel select register	0x0F
R8_ADC_CFG	0x40001059	ADC configuration register	0xA0
R8_ADC_CONVERT	0x4000105A	ADC conversion control register	0x00
R16_ADC_DATA	0x4000105C	ADC data register	0x0XXX
R8_ADC_INT_FLAG	0x4000105E	ADC interrupt flag register	0x00
R8_ADC_CTRL_DMA	0x40001061	DMA and interrupt control register	0x00
R8_ADC_DMA_IF	0x40001062	ADC and DMA interrupt flag register	0x00
R8_ADC_AUTO_CYCLE	0x40001063	Continuous ADC timing cycle register	0xXX
R16_ADC_DMA_NOW	0x40001064	Current buffer address of DMA register	0x0000XXXX
R16_ADC_DMA_BEG	0x40001068	Start buffer address of DMA	0x0000XXXX
R16_ADC_DMA_END	0x4000106C	End buffer address of DMA	0x0000XXXX
R32_ADC_SCAN_CFG1	0x40001070	Scan configuration register 1	0xFFFFFFFF
R32_ADC_SCAN_CFG2	0x40001074	Scan configuration register 2	0x00000FFF

ADC Input Channel Select Register (R8_ADC_CHANNEL)

Bit	Name	Access	Description	Reset value
[7:4]	Reserved	RO	Reserved	0
[3:0]	RB_ADC_CH_INX	RW	ADC channel index number, a total of 9 channels: 02h-06h: External signal channels AIN2 to AIN6; 09h: External signal channel AIN9; 0Ch-0Dh: External signal channels AIN12 to AIN13; CH591 only supports AIN0 ~ AIN5, VBAT and TS.	1111b

ADC Configuration Register (R8_ADC_CFG)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved	0
[7:6]	RB_ADC_CLK_DIV	RW	ADC clock frequency selection: When R16_CLK_SYS_CFG[9] = 1: 1: 4x division based on CK32M, 8MHz; 0: 2x division based on CK32M, 16MHz. When R16_CLK_SYS_CFG[9] = 0: 1: 2x division based on CK16M, 8MHz; 0: Direct feed from CK16M, 16MHz.	0
[5:4]	RB_ADC_PGA_GAIN	RW	ADC input PGA gain selection: 00: Reserved; 01: -6dB, 1/2; 10: 0dB, 1 time, no gain; 11: 6dB, 2 times.	10b
3	RB_ADC_OFS_TEST	RW	ADC offset error test mode: 1: Test/calibrate mode. The lower 12-bit data of data register R16_ADC_DATA will be inverted	0

			bit by bit in test mode (0x0573 is inverted to 0x0A8C); 0: Normal mode.	
2	Reserved	RO	Reserved	0
1	RB_ADC_BUF_EN	RW	ADC input buffer BUF enable: 1: Enable; 0: Disable.	0
0	RB_ADC_POWER_ON	RW	ADC module power enable: 1: Enable; 0: Disable.	0

The PGA gain calculation formula explains: $\Delta V * A + V_{ref} = 2 * V_{ref} * ADC / 4096$.

ΔV : Voltage input to the ADC module.

In single-ended mode $\Delta V = (VIN - 0)$, where VIN is the voltage under test.

The ADC module input voltage parameter is $V * A$, which needs to satisfy: $-V_{ref} \leq V * A \leq +V_{ref}$.

A: Gain factor, refer to Table 14-2 and Table 14-3 "PGA Gain Selection".

ADC: The digital quantity after ADC conversion, i.e. R16_ADC_DATA.

Vref: The actual voltage value of the power node VINTA of the internal analog circuit is usually $1.05V \pm 0.015$.

Table 14-2 Table of PGA gain selection and input voltage range (Single-ended input mode)

PGA gain selection	VINn	The sampled voltage, VINp, calculated from the data converted by ADC	Upper limit of theoretically measurable voltage	Measured voltage range (assuming Vref=1.05V)	High accuracy Measured voltage range
-6dB(1/2)	0	$(ADC/1024-2)*V_{ref}$	$2*V_{ref}$	$0V \sim VIO15+0.2V$	$0.05V \sim VIO15-0.1$
0dB(1)	0	$(ADC/2048-1)*V_{ref}$	$1*V_{ref}$	$0V \sim 1.05V$	$0.05V \sim 1V$
6dB(2)	0	$(ADC/4096-0.5)*V_{ref}$	$0.5*V_{ref}$	$0V \sim 0.525V$	$0.05V \sim 0.5V$

ADC Conversion Control Register (R8_ADC_CONVERT)

Bit	Name	Access	Description	Reset value
7	RB_ADC_EOC_X	RO	ADC conversion end flag (Asynchronous signal): 1: Complete; 0: In progress.	0
6	Reserved	RO	Reserved	0
[5:4]	RB_ADC_SAMPLE_TIME	RW	ADC sampling period selection: 00: $4*T_{adc}$; 01: $6*T_{adc}$; 10: $8*T_{adc}$; 11: $10*T_{adc}$.	0
3	RB_ADC_CHAN_EN	RW	ADC channel enable: 1: Enable channel selection; 0: Disable channel selection.	1
[2:1]	Reserved	RO	Reserved	0
0	RB_ADC_START	RW	ADC conversion enable control and status.	0

			It is cleared automatically when non-continuous ADC ends or when DMA ends: 1: Start conversion/being converted; 0: Stop conversion.	
--	--	--	---	--

ADC Data Register (R16_ADC_DATA)

Bit	Name	Access	Description	Reset value
[15:12]	Reserved	RW	Reserved.	0
[11:0]	RB_ADC_DATA	RO	Data after ADC conversion.	XXXh

ADC Interrupt Flag Register (R8_ADC_INT_FLAG)

Bit	Name	Access	Description	Reset value
7	RB_ADC_IF_EOC	RO	ADC conversion finished flag. This flag can be cleared by writing to register R8_ADC_CONVERT or RB_TKEY_CTRANS_ACT.	0
[6:0]	Reserved	RW	Reserved	0

DMA and Interrupt Control Register (R8_ADC_CTRL_DMA)

Bit	Name	Access	Description	Reset value
7	RB_ADC_AUTO_EN	RW	Automatic continuous ADC sampling enable at timing interval: 1: Enable automatic ADC; 0: Disable automatic ADC.	0
6	RB_ADC_CONT_EN	RW	ADC continuous conversion mode enable: 1: Enable continuous ADC; 0: Disable continuous ADC.	0
5	Reserved	RO	Reserved	0
4	RB_ADC_IE_EOC	RW	ADC conversion completion interrupt enable: 1: Enable interrupt; 0: Disable interrupt.	0
3	RB_ADC_IE_DMA_END	RW	DMA end interrupt enable: 1: Enable interrupt; 0: Disable interrupt.	0
2	RB_ADC_DMA_LOOP	RW	DMA address loop enable: 1: Enable address loop; 0: Disable address loop. If the DMA address loop is enabled, when the DMA address is added to the set end address, it will automatically loop to the start address that is set.	0
1	Reserved	RO	Reserved	0

0	RB_ADC_DMA_ENABLER	RW	DMA function enable: 1: Enable DMA; 0: Disable DMA.	0
---	--------------------	----	---	---

ADC and DMA Interrupt Flag Register (R8_ADC_DMA_IF)

Bit	Name	Access	Description	Reset value
[7:5]	Reserved	RO	Reserved	0
4	RB_ADC_IF_END_ADC	RW1	ADC conversion completion interrupt flag, write 1 to clear or when data is taken by DMA or write to register R8_ADC_CONVERT to clear this flag: 1: ADC has been completed once; 0: Not completed.	0
3	RB_ADC_IF_DMA_END	RW1	DMA completion flag. Write 1 to reset: 1: DMA has been completed; 0: Not completed.	0
[2:0]	Reserved	RO	Reserved	0

Continuous ADC Timing Cycle Register (R8_ADC_AUTO_CYCLE)

Bit	Name	Access	Description	Reset value
[7:0]	R8_ADC_AUTO_CYCLE	RW	The starting value of continuous ADC timing cycle is counted in unit of 16 system clocks and reloaded after reaching 256. Calculation: Timing $= (256 - R8_ADC_AUTO_CYCLE) * 16 * T_{sys}$ Counts in units of 32MHz clock at the start value of continuous TKEY timing cycles. Calculation method: $Timing = R8_ADC_AUTO_CYCLE * T_{CK32M}.$	XXh

DMA Current Buffer Address (R16_ADC_DMA_NOW)

Bit	Name	Access	Description	Reset value
15	Reserved	RO	Reserved	0
[14:1]	R16_ADC_DMA_NOW	RO	Current address of DMA data buffer. It can be used to calculate the number of conversions. Calculation: $COUNT = (ADC_DMA_NOW - ADC_DMA_BEGIN) / 2.$	XXXXh
0	Reserved	RO	Reserved	0

DMA Start Buffer Address (R16_ADC_DMA_BEG)

Bit	Name	Access	Description	Reset value
15	Reserved	RO	Reserved	0

[14:1]	R16_ADC_DMA_BEG	RW	DMA data buffer start address, the address must be 2-byte aligned.	XXXXh
0	Reserved	RO	Reserved	0

DMA End Buffer Address (R16_ADC_DMA_END)

Bit	Name	Access	Description	Reset value
15	Reserved	RO	Reserved	0
[14:1]	R16_ADC_DMA_END	RW	DMA data buffer end address (not included), address must be 2-byte aligned.	XXXXh
0	Reserved	RO	Reserved	0

Scan Configuration Register 1 (R32_ADC_SCAN_CFG1)

Bit	Name	Access	Description	Reset value
[31:28]	RB_ADC_SCAN_CH8	RW	Channel 8 analog input channel select bit: 0010: AIN2; 0011: AIN3; 0100: AIN4; 0101: AIN5; 0110: AIN6; 1001: AIN9; 1010: AIN10; 1011: AIN11; 1100: AIN12; 1101: AIN13; 1110: Battery voltage VBAT; Others: Reserved.	1111b
[27:24]	RB_ADC_SCAN_CH7	RW	Channel 7 analog input channel select bit: As above.	1111b
[23:20]	RB_ADC_SCAN_CH6	RW	Channel 6 analog input channel select bit: As above.	1111b
[19:16]	RB_ADC_SCAN_CH5	RW	Channel 5 analog input channel select bit: As above.	1111b
[15:12]	RB_ADC_SCAN_CH4	RW	Channel 4 analog input channel select bit: As above.	1111b
[11:8]	RB_ADC_SCAN_CH3	RW	Channel 3 analog input channel select bit: As above.	1111b
[7:4]	RB_ADC_SCAN_CH2	RW	Channel 2 analog input channel select bit: As above.	1111b
[3:0]	RB_ADC_SCAN_CH1	RW	Channel 1 analog input channel select bit: As above.	1111b

Scan Configuration Register 2 (R32_ADC_SCAN_CFG2)

Bit	Name	Access	Description	Reset value
-----	------	--------	-------------	-------------

31	RB_ADC_IF_SCAN_END	RW	Scan end flag: A value of 1 indicates the scan has ended. Writing 1 to this flag clears it; writing 0 has no effect.	0
30	RB_ADC_SCAN_MASKDIS	RW	After the ADC sampling channel completes, disable the channel enable: 1: Disable channel shutdown; 0: Enable channel shutdown.	0
29	RB_ADC_IE_SCAN_END	RW	Scan-end interrupt enable: 1: Enable interrupt; 0: Disable interrupt.	0
28	RB_ADC_SCAN_SEL	RW	Scan selection: 1: ADC; 0: TKEY.	0
[27:24]	RB_ADC_SCAN_NUM	RW	Number of scan channels. When RB_ADC_SCAN_SEL is 1: Indicates the number of scan channels minus one; When RB_ADC_SCAN_SEL is 0: Indicates the number of scan channels.	0
23	RB_ADC_SCAN_EN	RW	Channel scan mode enable: 1: Enable mode; 0: Disable mode.	0
22	Reserved	RO	Reserved	0
[21:20]	RB_EHCH_TIM	RW	Number of ADC samples per channel during channel scanning. 00: 1 sample; 01: 2 samples; 10: 3 samples; 11: 4 samples.	0
[19:16]	RB_ADC_SCAN_CH13	RW	Channel 13 analog input channel select bit: 0010: AIN2; 0011: AIN3; 0100: AIN4; 0101: AIN5; 0110: AIN6; 1001: AIN9; 1100: AIN12; 1101: AIN13; 1110: Battery voltage VBAT; Others: Reserved.	1111b
[15:12]	RB_ADC_SCAN_CH12	RW	Channel 12 analog input channel select bit: As above.	1111b
[11:8]	RB_ADC_SCAN_CH11	RW	Channel 11 analog input channel select bit: As above.	1111b

[7:4]	RB_ADC_SCAN_CH1 0	RW	Channel 10 analog input channel select bit: As above.	1111b
[3:0]	RB_ADC_SCAN_CH9	RW	Channel 9 analog input channel select bit: As above.	1111b

14.3 ADC Configuration

14.3.1 External Channel Sampling

- 1) Set RB_ADC_POWER_ON in R8_ADC_CFG to 1 to enable ADC, select the sampling frequency through RB_ADC_CLK_DIV, and enable the input buffer and select signal gain through RB_ADC_BUF_EN and RB_ADC_PGA_GAIN;
- 2) Set R8_ADC_CHANNEL register, and select external or internal signal channel;
- 3) Set the R8_ADC_CONVERT register, set RB_ADC_START, and enable ADC conversion;
- 4) Query and wait for RB_ADC_START to be automatically cleared or wait for RB_ADC_IF_EOC to be set to 1, indicating that the conversion is completed. 12-bit ADC conversion data can be obtained by reading R16_ADC_DATA. If time is enough, it is recommended to convert again and discard the ADC data obtained first;
- 5) Repeat steps 2, 3 and 4 to continue to sample another channel or the next set of data;
- 6) Single ADC conversion cycle: ADC sampling (4 clocks) + switching 1 clock + conversion time (12 clocks) $\approx 17 T_{adc}$, and 1 time interval is added for continuous ADC. Among them, $T_{adc} = T_{CK32M}/RB_ADC_CLK_DIV$;

14.3.2 Enable DMA Automatic Continuous ADC

- 1) Set ADC parameters and select channels with reference to non-DMA mode;
- 2) Set R8_ADC_AUTO_CYCLE to select continuous ADC cycle;
- 3) Set the R16_ADC_DMA_BEG register as the start address of buffer that stores ADC data, set the R16_ADC_DMA_END register as the end address of buffer that stores ADC data (not included), and set the RB_ADC_DMA_ENABLE in R8_ADC_CTRL_DMA to 1, to enable DMA function;
- 4) Optional step. If it is needed to enable interrupts, set the corresponding interrupt enable register bit. When RB_ADC_IE_EOC=1, the ADC will be triggered by RB_ADC_IF_END_ADC to complete the interrupt. When RB_ADC_IE_EOC=0 and RB_ADC_IE_DMA_END=0, the ADC will be triggered by RB_ADC_IF_EOC to complete the interrupt;
- 5) Set RB_ADC_AUTO_EN to 1 to enable automatic continuous ADC;
- 6) RB_ADC_IF_EOC and RB_ADC_IF_END_ADC is set to 1 after ADC is completed, but RB_ADC_IF_END_ADC is automatically cleared after the data is taken by DMA, so you can query RB_ADC_IF_EOC to query ADC completion status.

14.4 Channel Scan

14.4.1 Scan Mode

Scan mode is used to scan a set of analog channels. The analog input channel corresponding to channel 0 in the scan queue is selected by [3:0] of the R32_ADC_CTRL register, while the analog input channels corresponding to channels 1 to 11 in the scan queue are configured by the R32_ADC_SCAN_CFG1 and R32_ADC_SCAN_CFG2 registers.

The scan mode can be set by setting the RB_ADC_SCAN_QUAN bit of the R32_ADC_SCAN_CFG2 register. If RB_ADC_SCAN_QUAN is not equal to 0, the input analog channels will be scanned sequentially in the configured order, and the scan mode can be set by setting the RB_ADC_SCAN_SEL bit of the R32_ADC_SCAN_CFG2 register. _SCAN_SEL bit of the R32_ADC_SCAN_CFG2 register to select whether

TKEY or ADC is scanned. Additionally, the RB_ADC_SCAN_MASK_DIS bit controls whether the channel is immediately disabled after ADC sampling completes during each channel scan. When RB_SCAN_MASK_DIS is set to 1, the channel remains enabled after ADC sampling completes during each channel scan. It is only disabled during the next channel scan, when the previous channel is closed and the next channel opened. When RB_SCAN_MASK_DIS is set to 0, the channel is immediately disabled after ADC sampling completes during each channel scan. The next channel is enabled only when the subsequent channel scan begins. This prevents the delay in disabling the previous channel from causing 2 channels to be active simultaneously.

14.4.2 Scan Interrupt

In scan mode, if the RB_ADC_IE_SCAN_END bit of the R32_ADC_SCAN_CFG2 register is set, the end-of-scan interrupt will be enabled, i.e., the interrupt will be generated at the end of the conversion of the last channel in the scan queue, at which time the RB_ADC_IF_SCAN_END bit of the R32_ADC_SCAN_CFG2 register will be set to 1 by the hardware (the bit needs to be cleared to 0 by the software) to indicate that the conversion of all channels in the scan queue is finished.

Chapter 15 Touch-Key

15.1 Introduction to Touch-Key

The chip provides a capacitance detection module, which can be used with the ADC module to realize the detection function of capacitance Touch-Key. There are a total of 8 input channels, and the supported capacitance value of Touch-Key ranges from 10pF to 100pF. The driver shielding output is provided to improve sensitivity.

15.2 Register Description

Table 15-1 Touch-Key registers

Name	Access address	Description	Reset value
R8_TKEY_CFG	0x40001057	TouchKey configuration register	0x00
R32_TKEY_CFG2	0x40001078	TouchKey configuration register 2	0x00000000
R32_TKEY_SEL	0x4000107C	TouchKey selection register	0x00000000

TouchKey Configuration Register (R8_TKEY_CFG)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved	0
6	RB_TKEY_DMA_EN	RW	TouchKey DMA enable: 1: Enable; 0: Disable.	0
[5:4]	Reserved	RO	Reserved	0
3	RB_ADC_PGA_CFG	RW	Select the operating speed of the PGA in the ADC: 1: High speed but slightly higher power consumption; 0: Normal speed.	0
2	RB_TKEY_DRV_EN	RW	TouchKey driver enable mask: 1: Enable; 0: Disable.	0
[1:0]	Reserved	RO	Reserved	0

TouchKey Configuration Register 2 (R32_TKEY_CFG2)

Bit	Name	Access	Description	Reset value
[31:26]	Reserved	RO	Reserved	0
[25:16]	RB_TRANSFER_N	RW	Number of transfer cycles.	0
[15:9]	Reserved	RO	Reserved	0
8	RB_TKEY_CTRANS_ACT	RW	TKEY module start signal and operating status, auto-zero: 1: Start detection/Operating; 0: Stopped operating.	0

7	RB_TKEY_EN	RW	TKEY module enable: 1: TKEY module enabled; 0: TKEY module disabled.	0
[6:5]	RB_TKADC_START_SEL	RW	TouchKey early ADC trigger timing selection, in units of transfer cycles: 00: 2; 01: 4; 10: 8; 11: 16.	0
[4:3]	RB_TKEY_CC	RW	TouchKey capacitor charging time: 00: 1Tsys; 01: 2Tsys; 10: 3Tsys; 11: 4Tsys.	0
[2:0]	RB_TKEY_TRANS	RW	TouchKey capacitance charge transfer time: 000: 1Tsys; 001: 4Tsys; 010: 8Tsys; 011: 12Tsys; 100: 16Tsys; 101: 24Tsys; 110: 32Tsys; 111: 48Tsys.	0

TouchKey Selection Register (R32_TKEY_SEL)

Bit	Name	Access	Description	Reset value
[31:30]	Reserved	RO	Reserved	0
[29:16]	RB_DRIVEROUT_EN	RW	TouchKey driver channel selection switch: 1: Channel open; 0: Channel closed.	0
[15:14]	Reserved	RO	Reserved	0
[13:0]	RB_SEL_CS	RW	Charge transfer channel selection: 10000000000000: ADCIN13; 01000000000000: ADCIN12; 00001000000000: ADCIN9; 00000001000000: ADCIN6; 00000000100000: ADCIN5; 00000000010000: ADCIN4; 00000000001000: ADCIN3; 00000000000100: ADCIN2;	0

Note: In RB_SEL_CS[13:0]=*****, the specific channel is enabled based on which bits are set to 1 in the 14-bit control word. Multiple channels can be enabled simultaneously.

Operation flow of single channel scanning mode:

- 1) Set RB_ADC_CLK_DIV in R8_ADC_CFG to select the sampling frequency. Set RB_ADC_BUF_EN in R8_ADC_CFG to 1. enable the input buffer. Set RB_ADC_PGA_GAIN in R8_ADC_CFG to select the signal gain. Set RB_ADC_SAMPLE_TIME in R8_ADC_CONVERT to select the sampling period, etc. (If using DMA, configure DMA as well). Set RB_TKEY_DRV_EN to enable TouchKey driver mask enable, set RB_DRIVEROUT_EN to select the driver mask channel, set RB_ADC_CH_INX for channel selection (connect to the high-capacitance channel selection pin).
- 2) Set RB_SEL_CS channel selection (transfer channel selection), configure RB_TKEY_CC charge time selection, RB_TKEY_TRANS transfer time selection, set RB_TRANSFER_N to configure single-channel transfer count, and configure RB_TKADC_START_SEL to set the time for early activation of the ADC enable signal.

Note: The ADC_PGA enable signal does not need to be configured as high. The TKEY module will enable the ADC and PGA at the appropriate time based on the configuration of RB_TKADC_START_SEL. If the ADC_PGA enable is configured first, the PGA and ADC will start operating immediately upon transfer, resulting in wasted power consumption.

- 1) Configure the RB_TKEY_EN signal to enable TKEY, then initiate charge transfer by configuring the RB_ADC_START and RB_TKEY_CTRANS_ACT signals;
- 2) Upon completion of the current channel transfer process and detection of RB_ADC_START at position 0, read R16_ADC_DATA to obtain the 12-bit touch key conversion data;
- 3) Repeat steps (2), (3), and (4) to continue sampling another channel or the next set of data.

Operation flow of continuous channel scanning mode:

- 1) Set RB_ADC_CLK_DIV in R8_ADC_CFG to select the sampling frequency. Set RB_ADC_BUF_EN in R8_ADC_CFG to 1. enable the input buffer. Set RB_ADC_PGA_GAIN in R8_ADC_CFG to select the signal gain. Set RB_ADC_SAMPLE_TIME in R8_ADC_CONVERT to select the sampling period, etc. Set RB_TKEY_DRV_EN to enable the TouchKey driver disable.
- 2) Configure DMA-related settings, such as setting the ADC DMA start address and end address (exclusive), with the data buffer length no less than the number of scans configured by RB_ADC_SCAN_NUM; configure R8_ADC_CTRL_DMA, enable the DMA function bit RB_ADC_DMA_ENABLE, and enable the DMA end interrupt bit RB_ADC_IE_DMA_END;
- 3) Configure the RB_ADC_CH_INX bit. This channel is configured for connection to a large capacitor Cx and does not participate in the channel scan cycle.
- 4) The analog input channels corresponding to channels 1 to 13 in the scan queue are configured via the R32_ADC_SCAN_CFG1 and R32_ADC_SCAN_CFG2 registers (by sequentially populating the scan queue with channels to be scanned). Scanning proceeds in the order of the channel names entered. Since this is a TKEY channel scan, RB_ADC_SCAN_SEL is set to 0. RB_ADC_SCAN_NUM configures the number of channels to scan, followed by enabling the scan-end interrupt flag (RB_ADC_IE_SCAN_END).
- 5) Configure the usage settings for a continuous channel scan charge migration TKEY. The channel selected by the RB_ADC_CH_INX bit set in (3) and (4) is the channel connected to the large capacitor Cx. The channels entered into the R32_ADC_SCAN_CFG1 and R32_ADC_SCAN_CFG2 registers are the channels that sequentially perform charge migration scans. Configure RB_TRANSFER_N to specify the number N of charge transfer cycles per channel; configure RB_TKEY_CC for charge time selection and RB_TKEY_TRANS for transfer time selection. RB_TKADC_START_SEL confirms the power-up timing for the PGA and ADC; after setting RB_TKEY_EN high and configuring the RB_ADC_START and

RB_TKEY_CTRANS_ACT signals, the charge transfer process for the current single channel begins. After ADC sampling completes, the above TKEY functional configuration remains unchanged. Functional scanning for all channels can then proceed sequentially according to the channel order set in the R32_ADC_SCAN_CFG1 and R32_ADC_SCAN_CFG2 registers.

- 6) Check the scan end flag RB_ADC_IF_SCAN_END to determine scan completion; retrieve conversion data from the DMA data buffer;
- 7) When enabling DMA to save sampled values after consecutive channel scans, configure the RB_EHCH_TIM bit to select the number of sampled values per channel. Each channel can store up to four consecutive sampled values into DMA.

15.3 Touch-Key Configuration

Please refer to and call related subprograms.

Chapter 16 Reserved

Chapter 17 Wireless Communication

17.1 Introduction

The chip integrates low-power 2.4-GHz wireless communication modules, including RF transceiver, baseband, link control and antenna matching network. Bluetooth Low Energy (BLE) is supported. Supports GFSK (Gaussian Frequency Shift Keying) digital modulation and demodulation, with adjustable data transmission rates. More than 100 registers are provided internally to adjust parameters, control process and status. This datasheet does not provide a detailed description of registers. The underlying operations of wireless communication mainly provide application support with subroutine libraries.

Main features:

- Integrated with 2.4GHz RF transceiver, BaseBand and LLE link control.
- Support Bluetooth Low Energy (BLE), compatible with Bluetooth Low Energy 5.0 specification.
- Single-ended RF interface and simplified board design.
- 2Mbps, 1Mbps.
- Support up to 8 kHz reporting rate in 2.4 GHz mode.
- -95dBm RX sensitivity.
- Programmable TX power from -24dBm to +4dBm, and dynamic adjustment is supported.
- Single-ended RF interface simplifies board-level design.
- Support AES-128 encryption/decryption.
- Support DMA.
- Optimized protocol stack and application layer API, and support networking.

17.2 2.4GHz Module

17.2.1 2.4GHz Module Functional Characteristics

- Supports standard mode and enhanced mode.
- Address length is configurable.
- CRC is configurable: length 0~2 bytes, polynomial and XOR operation configurable.
- Byte order is configurable: data bit order (MSB/LSB) configurable.
- Wireless rate support: 1Mbps, 2Mbps.
- Received modulation index range: 0.3~0.9.

17.2.2 2.4GHz Module Data Frame Format

The 2.4GHz module's data frame format supports both standard mode and enhanced mode. When configured for standard mode, the data length is fixed. When configured for enhanced mode, the data length becomes variable due to the addition of a control word, and includes a PID and an acknowledgment flag.

For specific applications, please refer to the provided RF application examples.

17.3 LLE Module

LLE module supports automatic transmitting and receiving mode and manual sending and receiving mode. 4 groups of independent hardware timers can control the time point of transmitting and receiving data in any one process.

17.4 DMA Module

The controller has 1 group of DMAs, 2 channels. 2 channels of DMAs are used for transmitting data and receiving data respectively in auto transmit mode, the address of the transmit DMA and the address of the receive DMA can be configured at the same time, so that no further configuration is required during the frame interval.

17.5 BB Module

17.6 AES Module

Please conduct specific applications based on BLE protocol stack library and refer to the BLE application examples provided.

Chapter 18 LED Display Controller

18.1 Introduce to LED Controller

The chip provides an interface for LED display control cards, featuring a built-in 2-byte FIFO. It supports DMA and interrupts to conserve CPU processing time, and accommodates 1/2/4-channel data line interfaces.

18.2 Register Description

Table 18-1 LED-related registers

Name	Access address	Description	Reset value
R8_LED_CTRL_MOD	0x4000E000	LED mode configuration register	0x02
R8_LED_CLOCK_DIV	0x4000E001	LED serial clock frequency division register	0x10
R8_LED_STATUS	0x4000E004	LED status register	0xX0
R16_LED_FIFO	0x4000E008	LED data FIFO register	0xFFFF
R16_LED_DMA_CNT	0x4000E010	LED DMA remaining count register	0x0000
R16_LED_DMA_MAIN	0x4000E014	LED main buffer DMA address	0xFFFF
R16_LED_DMA_AUX	0x4000E018	LED auxiliary buffer DMA address	0xFFFF

LED Mode Configuration Register (R8_LED_CTRL_MOD)

Bit	Name	Access	Description	Reset value
[7:6]	RB_LED_CHAN_MOD	RW	LED channel mode selection: 00: LED0, single-channel output; 01: LED0/1, dual-channel output; 10: LED0–3, 4-channel output; 11: LED0–3, 4-channel output, with LED2/3 channel data sourced from auxiliary buffer.	0
5	RB_LED_IE_FIFO	RW	FIFO count half-full interrupt enable: 1: FIFO count ≤ 2 triggers interrupt; 0: Disables interrupt.	0
4	RB_LED_DMA_EN	RW	LED DMA function and DMA interrupt enable: 1: Enable; 0: Disable.	0
3	RB_LED_OUT_EN	RW	LED signal output enable: 1: Enable; 0: Disable.	0
2	RB_LED_OUT_POLAR	RW	LED data output polarity control bit: 1: Inverted, outputs 1 for data 0, outputs 0 for data 1; 0: Non-inverted, outputs 0 for data 0, outputs 1 for data 1.	0

			0: Pass-through, outputs 0 for data 0, outputs 1 for data 1.	
1	RB_LED_ALL_CLEAR	RW	LED FIFO/counter/interrupt flag clear: 1: Force clear and reset; 0: Do not clear.	1
0	RB_LED_BIT_ORDER	RW	LED serial data bit order selection: 1: High-order bit first; 0: Low-order bit first.	0

LED Serial Clock Frequency Division Register (R8_LED_CLOCK_DIV)

Bit	Name	Access	Description	Reset value
[7:0]	R8_LED_CLOCK_DIV	RW	LED output clock division factor. LEDC frequency = Fsys / R8_LED_CLOCK_DIV.	10h

LED Status Register (R8_LED_STATUS)

Bit	Name	Access	Description	Reset value
7	RB_LED_IF_DMA_END	RW1	DMA completion flag: Write 1 to clear or write R16_LED_DMA_CNT to clear: 1: Completed; 0: Not completed.	0
6	RB_LED_FIFO_EMPTY	RO	FIFO empty status bit: 1: FIFO is empty; 0: Data is present in the FIFO.	1
5	RB_LED_IF_FIFO	RW1	FIFO count overflow interrupt flag: Write 1 to clear or write R16_LED_FIFO to clear: 1: FIFO count <= 2; 0: FIFO count > 2.	0
4	RB_LED_CLOCK	RO	Current LED clock signal level status: 1: High level; 0: Low level.	X
3	Reserved	RO	Reserved	0
[2:0]	RB_LED_FIFO_COUNT	RO	The current byte count in the FIFO must be an even number.	0

LED Data FIFO Register (R16_LED_FIFO)

Bit	Name	Access	Description	Reset value
[15:0]	R16_LED_FIFO	WO	LED data FIFO input, 16-bit write.	XXXXb

LED DMA Remaining Count Register (R16_LED_DMA_CNT)

Bit	Name	Access	Description	Reset value
[15:0]	R16_LED_DMA_CNT	RW	LED_DMA_MAIN Main buffer's current	0

			DMA remaining words (16-bit) count. Decreases automatically after DMA initiation. Only the lower 12 bits are valid. Does not include auxiliary buffers.	
--	--	--	---	--

LED Main Buffer DMA Address (R16_LED_DMA_MAIN)

Bit	Name	Access	Description	Reset value
[15:0]	R16_LED_DMA_MAIN	RW	Main buffer DMA start address/current address. After initialization with preset values, it automatically increments and must be 2-byte aligned.	XXXXb

LED Auxiliary Buffer DMA Address (R16_LED_DMA_AUX)

Bit	Name	Access	Description	Reset value
[15:0]	R16_LED_DMA_AUX	RW	Auxiliary Buffer DMA Start Address/Current Address: Preset initial value, auto-increments thereafter, must be 2-byte aligned.	XXXXb

18.3 LED Configuration

- (1) Set R8_LED_CLOCK_DIV to select the LED output clock frequency, and set RB_LED_ALL_CLEAR to 0;
- (2) Set R16_LED_DMA_MAIN to point to the buffer containing the data ready for output, i.e., the main buffer;
- (3) If LED channel mode 3 is selected, also set R16_LED_DMA_AUX to point to the auxiliary buffer;
- (4) Set R8_LED_CTRL_MOD to select the channel mode, output polarity, bit order, enable interrupts and DMA functions, etc.
- (5) Configure LEDC and the necessary LED0 to LED3 pins as outputs. Optionally, set the drive capability for the corresponding I/O pins.
- (6) Set the DMA counter register R16_LED_DMA_CNT to initiate DMA transmission, or send data using the write FIFO method.

Chapter 19 Encoder

19.1 Introduce to Encoder

The chip supports encoders (ENC). 2 channels capture signals from 2 rotary encoders via I/O ports for encoding. It compares the high/low levels of one signal against the edge of the other signal to determine the encoding direction. The clock source is derived from the LSI.

19.2 Register Description

The chip supports encoders, providing 2 input channels CAP_IN1 and CAP_IN2 to capture signals from two rotary encoders for encoding. It can compare the high/low levels of the signals captured on one channel against the edge of the signal captured on the other channel to determine the encoding direction.

Selecting Encoder Configuration: Both input channels CAP_IN1 and CAP_IN2 serve as interfaces for rotary encoders. If the counter counts only on the edge of CAP_IN2, set RB_SMS_MODE = 01 in the R8_ENC_REG_CTRL register. If counting only on the edge of CAP_IN1, set RB_SMS_MODE = 10. If the counter counts on both CAP_IN1 and CAP_IN2 edges, set RB_SMS_MODE = 11.

Based on the sequence of transitions in the 2 input signals, the counter increments or decrements, while the hardware simultaneously sets the DIR bit in the R8_ENC_REG_CTRL register accordingly. Regardless of whether the counter counts based on CAP_IN1, CAP_IN2, or both CAP_IN1 and CAP_IN2 simultaneously, a transition on either input (CAP_IN1 or CAP_IN2) will recalculate the RB_ENC_DIR bit.

The R32_ENC_REG_CEND bit must be configured before starting the count.

Table 19-1 Relationship between count direction and coded signal

Effective edge	Relative signal levels (CAP_IN1 relative to CAP_IN2, CAP_IN2 relative to CAP_IN1)	Signal corresponding to channel CAP_IN1		Signal corresponding to channel CAP_IN2	
		Rise	Fall	Rise	Fall
Only count in CAP_IN1	High	Count down	Count up	No count	No count
	Low	Count up	Count down	No count	No count
Only count in CAP_IN2	High	No count	No count	Count up	Count down
	Low	No count	No count	Count down	Count up
Count in CAP_IN1 and CAP_IN2	High	Count down	Count up	Count up	Count down
	Low	Count up	Count down	Count down	Count up

The operation steps of capturing the rotary encoder signal for encoding are as follows:

- (1) Set RB_SMS_MODE to configure the encoding edge mode. Enable the wake-up enable if sleep wake-up is required;
- (2) Configure R32_ENC_REG_CEND. The device operates only when its value is non-zero;
- (3) Set RB_START_ENC_EN to 1 to initiate operation;

(4) Set RB_RD_CLR_EN to 1 to automatically clear the current encoding value after each read of R32_ENC_REG_CCNT. This also automatically clears the wake-up signal after each system wake-up.

Table 19-2 Encoder-related register list

Name	Access address	Description	Reset value
R8_ENC_REG_CTRL	0x40006080	Encoder control register	0x00
R8_ENC_INTER_EN	0x40006081	Encoder interrupt enable register	0x00
R8_ENC_INT_FLAG	0x40006082	Encoder interrupt flag register	0x00
R32_ENC_REG_CEND	0x40006084	Encoder end configuration register	0x00000000
R32_ENC_REG_CCNT	0x40006088	Encoder current value configuration register	0x00000000

Encoder Control Register (R8_ENC_REG_CTRL)

Bit	Name	Access	Description	Reset value
7	RB_ENC_IN_CH2	RO	The value of encoder capture channel 2.	0
6	RB_ENC_IN_CH1	RO	The value of encoder capture channel 1.	0
5	RB_ENC_DIR	RO	Encoder current direction: 1: Reverse; 0: Forward.	0
4	Reserved	RO	Reserved	0
3	RB_RD_CLR_EN	RW	The encoder reads the count and clears it.	0
[2:1]	RB_SMS_MODE	RW	Encoder edge mode: 00 = IDLE; 10 = Count on CAP_IN1 edge; 01 = Count on CAP_IN2 edge; 11 = Count on both CAP_IN1 and CAP_IN2 edges.	0
0	RB_START_ENC_EN	RW	Encoder enable: 1: Enable (LSI must be turned on); 0: Disable.	0

Encoder Interrupt Enable Register (R8_ENC_INTER_EN)

Bit	Name	Access	Description	Reset value
[7:2]	Reserved	RO	Reserved	0
1	RB_IE_DIR_DEC	RW	Encoder backward interrupt enable: 1: Enable interrupt; 0: Disable interrupt.	0
0	RB_IE_DIR_INC	RW	Enable encoder forward interrupt: 1: Enable interrupt; 0: Disable interrupt.	0

Encoder Interrupt Flag Register (R8_ENC_INT_FLAG)

Bit	Name	Access	Description	Reset value
[7:2]	Reserved	RO	Reserved	0

1	RB_IF_DIR_DEC	RW	Encoder reverse interrupt flag: Write 1 to clear, 0 to set. 1: Encoder is reversing; 0: Encoder remains stationary.	0
0	RB_IF_DIR_INC	RW	Encoder forward interrupt flag: Write 1 to clear, 0 to set. 1: Encoder is advancing; 0: Encoder remains stationary.	0

Encoder End Value Configuration Register (R32_ENC_REG_CEND)

Bit	Name	Access	Description	Reset value
[31:0]	R32_ENC_REG_CEND	RW	Preset encoder end value.	0

Encoder Current Value Configuration Register (R32_ENC_REG_CCNT)

Bit	Name	Access	Description	Reset value
[31:0]	R32_ENC_REG_CCNT	RO	Current encoder value.	0

Chapter 20 Parameters

20.1 Absolute Maximum Ratings

(Critical or exceeding the absolute maximum value will probably cause the chip to work improperly or even be damaged)

Table 20-1 Absolute maximum ratings

Name	Parameter description	Min.	Max.	Unit
TA	Ambient temperature during operation	-40	85	°C
TS	Ambient temperature during storage	-55	125	°C
VDD15	System supply voltage (VDD15 connects to power, GND connects to ground)	-0.4	2.2	V
VIO15	I/O supply voltage (VIO15 connects to power, GND connects to ground)	-0.4	2.2	V
VIO	Voltage on input/output pins	-0.4	VIO15+0.4	V
VIO5	Voltage on input/output pins that support 5V withstand voltage	-0.4	5.5	V
VXCK	Voltage on PA10/PA11 after X32MI/X32MO/ enabling LSE	-0.3	1.4	V

20.2 Electrical Characteristics

Test conditions: TA=25°C, VDD15=VIO15=1.5V, Fsys=16MHz。

Table 20-2 Electrical characteristics

Name	Parameter description		Min.	Typ.	Max.	Unit
VDD15	System supply voltage @VDD33		1.2	1.5	1.8	V
VIO15	I/O supply voltage @VIO33		1.2	1.5	1.8	V
ICC ₈	Straight-through static supply current Condition: codes run in FLASH.	Fsys=8M		2.9		mA
ICC ₁₆		Fsys=16M		3.4		mA
ICC ₄₈		Fsys=48M		6.4		mA
VIL	GPIO low level input voltage		0		0.5	V
VIH	GPIO high level input voltage		1.0		VIO15	V
VIL5	GPIO that supports 5V tolerant voltage low level input voltage		0		0.5	V
VIH5	GPIO that supports 5V tolerant voltage high level input voltage		1.0		5	V
VOL	Output low level voltage (2mA/8mA input current)		0	0.3	0.4	V
VOH	Output high level voltage (2mA/8mA output current)		VIO15-0.4	VIO15-0.3	VIO15	V

IIN	Input current of GPIO floating input	-2	0	2	uA
IUP	Input current of GPIO with built-in pull-up resistor	3	9	18	uA
IDN	Input current of GPIO with built-in pull-down resistor	-18	-9	-3	uA
Vref	Voltage on VINTA pin (ADC reference voltage)	1.035	1.05	1.065	V
VLvr	LVR low-voltage reset threshold	1.03	1.1	1.16	V

20.3 Power Consumption in Low-power Modes

Test conditions: TA=25°C, VDD15=VIO15=3.3V, Fsys=16MHz.

Table 20-3 Low-power parameters (for reference only, related to temperature)

Low-power mode		Min.	Typ.	Max.	Unit
Idle mode, enable the clock combination of each module			1.4		mA
Halt mode, disable all clocks			180		uA
Sleep mode, multiple combinations, refer to Table 5-3			1.8~3.8		uA
Sleep mode, PMU+core+RAM2K, GPIO wakes up, no RTC			1.8		uA
Shutdown mode, several combinations, refer to Table 5-3			0.85~2.6		uA
Shutdown mode, only PMU, reset after GPIO wakes up, no RTC			0.85		uA

Table 20-4 Current on modules (for reference only, related to temperature)

Name	Parameter description		Min.	Typ.	Max.	Unit
I _{DD} (RAM8K)	RAM8K: 8KB SRAM			0.7		uA
I _{DD} (RAM8K)	RAM24K: 24KB SRAM			1.75		uA
I _{DD} (LSI)	Internal LSI oscillator	High precision		0.4		uA
		Low precision		0.26		uA
I _{DD} (HSI)	Internal HSI oscillator			90		uA
I _{DD} (LSE)	External LSE oscillator			0.4		uA
I _{DD} (HSE)	External HSE oscillator		110	190	300	uA
I _{DD} (BM)	Low-power battery low-voltage monitor module (BM)			1		uA
I _{DD} (PLL)	Internal PLL oscillator			170		uA
I _{DD} (ADC)	ADC module			0.45		mA
I _{DD} (TKEY)	Touch-Key module			0.2		mA
I _{DD} (BLE)	BLE	Receiving	Direct power supply	5		mA
		-24dBm transmission power	Direct power supply	3.3		mA
		0dBm transmission power	Direct power supply	8.6		mA
		+4dBm transmission	Direct power	13		mA

		power	supply				
--	--	-------	--------	--	--	--	--

20.4 Clock Source

Table 20-5 High-speed oscillator (HSI and HSE)

Symbol	Parameter description		Min.	Typ.	Max.	Unit
F _{HSI}	Internal HSI oscillator frequency			16		MHz
A _{HSI}	HSI oscillator accuracy	TA = -40°C~85°C		±1	±2.5	%
T _{SUHSI}	Internal HSI oscillator startup to available time				1	us
F _{HSE}	External HSE oscillator frequency			32		MHz
T _{SUHSE}	External HSE oscillator startup to available time		80	200	500	uS
T _{STHSE}	External HSE oscillator startup to stabilization time		200	500	3000	uS

Table 20-6 Low-speed oscillator (LSI and LSE)

Symbol	Parameter description		Min.	Typ.	Max.	Unit
F _{LSIR}	Internal LSI oscillator frequency (before calibration)		20K	32K	48K	Hz
F _{LSI}	Internal LSI frequency (after application software runtime calibration)		32726	32768	32810	Hz
A _{LSI}	LSI oscillator accuracy (after software calibration)	TA=-40°C~85°C		0.1	0.5	%
		TA=-20°C~60°C		0.04	0.2	%
T _{SULSI}	Internal LSI oscillator startup to stabilization time			40	100	uS
T _{SULSE}	External LSE oscillator startup to available time		100	300	1500	mS
T _{STLSE}	External LSE oscillator startup to stabilization time		500	1500	5000	mS

Table 20-7 PLL characteristics

Symbol	Parameter description	Min.	Typ.	Max.	Unit
F _{PLL}	Output clock after PLL (CK32M * 15)		480		MHz
T _{PLLK}	PLL lock time		20	40	uS

20.5 Timing Parameters

Test conditions: TA=25°C, VDD15=VIO15=3.3V, Fsys=6.4MHz.

Table 20-8 Timing parameters

Symbol	Parameter description		Min.	Typ.	Max.	Unit
T _{tpor}	Reset delay after RPOR		3	5	7	mS
T _{sr}	Reset delay after SR		2	8	18	uS
T _{wtr}	Reset delay after WTR		10	12	18	uS
T _{WAK}	Wakeup time to exit from	Idle Mode	0.6	1	3	uS
		Halt Mode	T _{SUHSE} +1	T _{SUHSE} +80	T _{SUHSE} +150	uS

	low-power status	Sleep Mode	$T_{SUHSE}+1$	$T_{SUHSE}+300$	$T_{SUHSE}+400$	μS
		Shutdown Mode	$T_{SUHSE}+0.4$	$T_{SUHSE}+1$	$T_{SUHSE}+5$	mS

Note: The delay parameters in the table are all based on multiples of T_{sys} , and the delay will be increased when the clock frequency is reduced.

The delay parameters in the table are based on using an external HSE clock source. If an external HSE clock source is used during sleep, the delay parameter T_{WAK} in Halt mode/Sleep mode/Shutdown mode in the table will be increased by about 0.2~1mS (activated to available T_{SUHSE}).

20.6 Other Parameters

Test conditions: $T_A=25^{\circ}C$, $V_{DD33}=V_{IO33}=3.3V$, $F_{sys}=16MHz$.

Table 23-9 Other parameters

Symbol	Parameter description		Min.	Typ.	Max.	Unit
T_{FRER}	Single sector erase operation time of Flash-ROM		6.0	6.4	7.0	mS
T_{FRPG}	Single word program operation time of Flash-ROM		2.0	2.2	2.6	mS
N_{EPCE}	Erase/program cycle endurance	5~45°C	100K	800K (Random test)		times
		-40~85°C	30K	100K (Random test)		
T_{DR}	Data hold capability of Flash-ROM		10			years
V_{ESD}	ESD tolerant voltage on I/O pins	Antenna (ANT)	2K	4K (Random test)		V
		I/O pins: PA and PB	4K	6K (Random test)		V

Chapter 21 Package

Chip Package

Package form	Body size	Pin pitch		Package description	Order model
QFN20	3*3mm	0.4mm	15.7mil	Quad Flat No-lead Package	CH596D

Note: All dimensions are in millimeters. The pin center spacing values are nominal values, and the error of other dimensions is not more than $\pm 0.2\text{mm}$.

QFN20

