CH572/CH570 Datasheet

V1.1

https://wch-ic.com

Overview

CH572 is a RISC-V MCU microcontroller with integrated 2.4G wireless communication. It integrates 2Mbps Bluetooth Low Energy BLE communication module, USB full-speed controller and transceiver, voltage comparator CMP, key detection module, SPI, UART, I2C and other rich peripheral resources. CH572 has a built-in LDO5V voltage regulator to generate 3.3V, supports single 3.3V or single 5V power supply, supports 1-wire or 2-wire simulation debugging, and is suitable for 2.4G wireless communication applications and simple Bluetooth applications with low pin counts.

CH570 is a low-power 2.4G wireless communication MCU, which does not support Bluetooth. Refer to CH572 for other modules.

Feature

- **Core**
- QingKe 32-bit RISC-V3C core
- Support RV32IMBC instruction set and self-extending instructions
- Low-power 3-stage pipeline
- Multi-speed system main frequency, maximum 100MHz
- Unique high-speed interrupt response mechanism
- **256K-byte non-volatile memory FlashROM:**
- 240KB user application program memory area CodeFlash
- 8KB system boot program memory area BootLoader
- 8KB system non-volatile configuration information memory area InfoFlash
- Support ICP, ISP and IAP, support OTA wireless upgrade
- **12K-byte volatile data storage SRAM:**
- Sleep retention memory area using 12KB dual power supply
- **Power management and low-power:**
- Built-in 5V to 3.3V regulator LDO5V
- Single 5V supply rated voltage: 5V
- Or single VDD33 power supply rated voltage: 3.3V
- Idle mode: 1.7mA
- Halt mode: 1.3mA (PLL/HSE not stopped)
  420uA (PLL/HSE stopped)
- Sleep mode: 0.46uA~1.2uA multiple gears
- Shutdown mode: 0.3uA~0.9uA multiple gears
- Optional low-power battery voltage and low voltage monitoring
- **Security properties:**
- AES-128 encryption and decryption, unique chip ID
- **2.4G and Bluetooth Low Energy**
- Integrated with 2.4GHz RF transceiver and baseband and link control
- Support GFSK digital modulation and demodulation
- Receive sensitivity -95dBm, programmable +7.5dBm transmission power
- BLE complies with Bluetooth Low Energy Specification 5.0
- Support 2Mbps, 1Mbps
- Support the highest 8KHz reporting rate in 2.4G mode
- Provide optimized protocol stack and application layer API, and support networking
- **Real-time clock (RTC):**
- 2 modes of timing and triggering
- **Clock:** Built-in PLL, built-in low frequency RC oscillator
- **Key detection module:**

- Support 20-channel key detection, including 10-channel matrix area keys and 10-channel independent area keys.

- **Analog voltage comparator CMP**

- Built-in 16-speed reference voltage, equivalent to 4-bit ADC.

- **Analog voltage comparator CMP**

- Built-in 16-speed reference voltage, equivalent to 4-bit ADC.

- **1 set of full-speed USB 2.0 controller and PHY:**

- 15 endpoints, supporting 64-byte packets and DMA
- Support full/low speed Host and Device modes.
- Support 1-wire USB communication

- **Timer and pulse width modulation PWM:**

- 1 set of 26-bit timers, the main frequency timing of 16MHz can reach 4.2S
- 1-channel capture/sampling, supporting rising edge/falling edge/dual edge.
- Support ENC MODE, and 2 channels are provided to capture the rotary encoder signal for encoding.
- 1-channel 26-bit PWM output
- 5-channel 16-bit PWM output
- 2 watchdog timers: independent and window
- SysTick: 32-bit counter

- **Universal asynchronous receiver/transmitter (UART):**

- 1 set of independent UART, compatible with 16C550, built-in 8-level FIFO
- 23-bit counter, up to 12.5Mbps communication baud rate

- **Serial peripheral interface (SPI):**

- Support Master and Slave modes
- Built-in FIFO, Support DMA

- **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

- **General-purpose input/output (GPIO):**

- 12 GPIOs, 1 of them support 5V signal input
- Optional pull-up or pull-down resistor, optional output drive capability
- 12 GPIOs support level or edge interrupt input

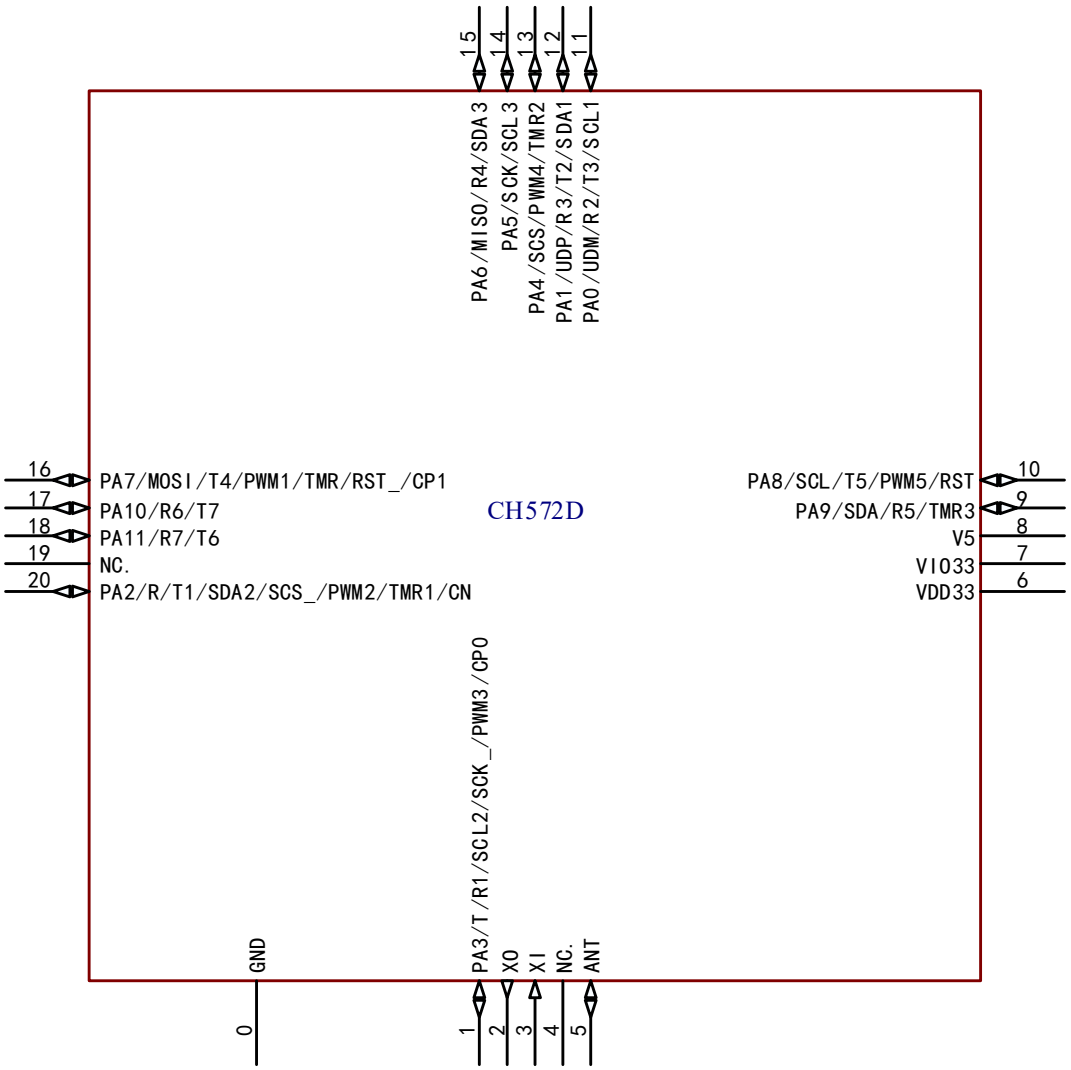
- **1-wire/2-wire serial debug interface**
- **Package:** QFN, DFN, SOP

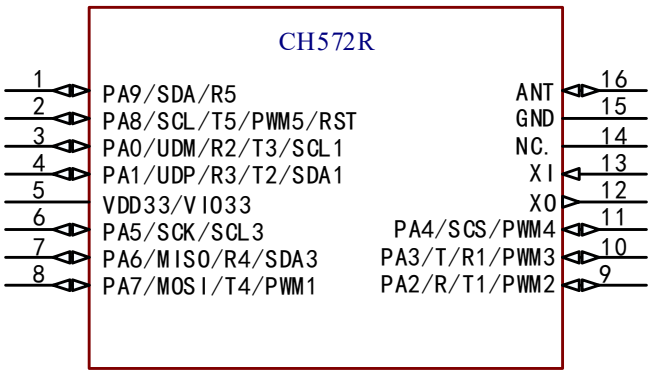
Part No.	CodeFlash +BootLoader	RAM	RTC	Timer	Capture	PWM	UART	SPI	I2C	CMP	Touch-key detection	USB host	USB device	2.4G wireless	BLE	Power voltage	GPIO	Package form
CH572D	240+8K	12K	√	1	1+2	1+5	1	1	1	1	20	√	√	√	√	2.0~3.6V or	12	QFN20
CH572Q	240+8K	12K		1	1+2	1+3	1	-	1	1	5	√	√		√	4.5~5.3V	5	DFN10X3
CH572R	240+8K	12K		1	1+2	1+5	1	1	1	1	9	√	√		√	2.0~3.6V	10	TSSOP16
CH570D	240+8K	12K		1	1+2	1+5	1	1	1	1	20	√	√		-	2.0~3.6V or	12	QFN20
CH570Q	240+8K	12K		1	1+2	1	1	-	1	-	5	-	√		-	4.5~5.3V	5	DFN10X3
CH570E	240+8K	12K		1	-	1	1	-	1	-	-	-	-		-	2.0~3.6V	3	SOP8

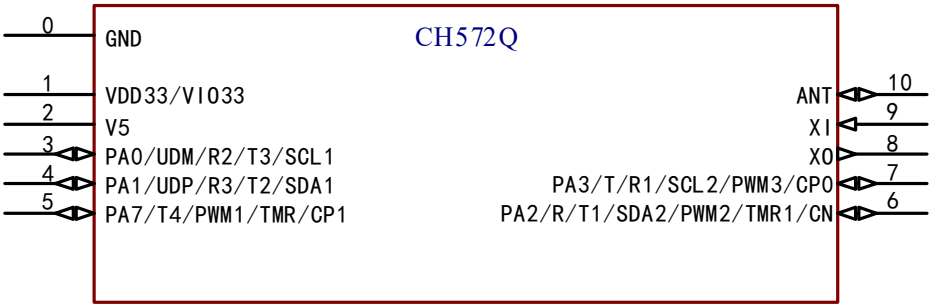
Chapter 1 Pinouts and Pin Definitions

1.1 Pinouts

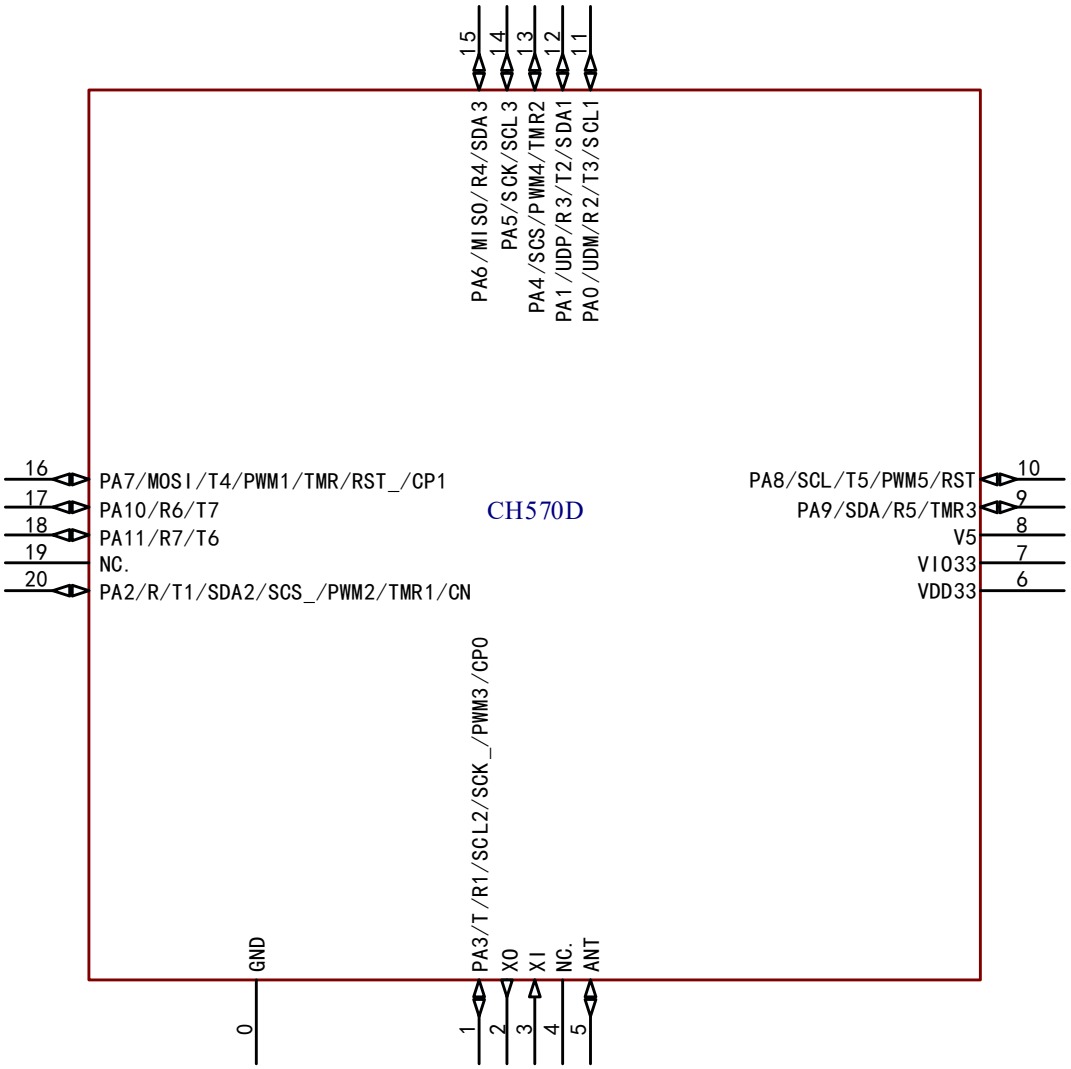
1.1.1 CH572 Pinouts

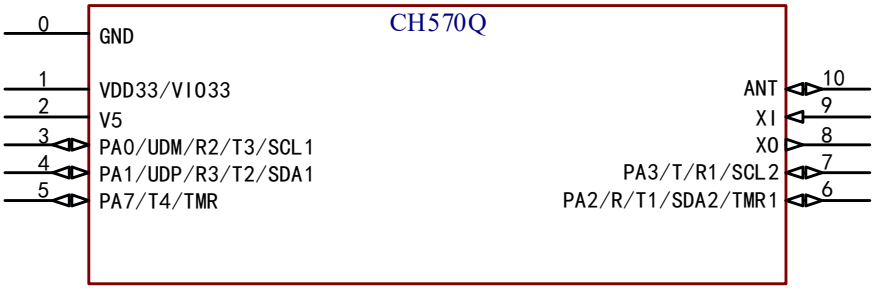


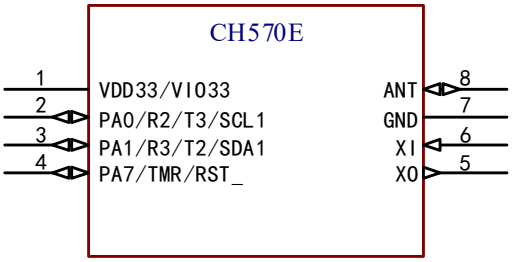




1.1.2 CH570 Pinouts




CH570Q
0
GND
1
VDD33/VIO33
2
V5
3
PA0/UDM/R2/T3/SCL1
4
PA1/UDP/R3/T2/SDA1
5
PA7/T4/TMR
ANT
10
XI
9
XO
8
PA3/T/R1/SCL2
7
PA2/R/T1/SDA2/TMR1
6


CH570E
1
VDD33/VIO33
2
PA0/R2/T3/SCL1
3
PA1/R3/T2/SDA1
4
PA7/TMR/RST_
ANT
8
GND
7
XI
6
XO
5

1.2 Pin Definitions

Note that the pin function descriptions in the following table are for all functions and do not relate to specific models. There are differences in peripheral resources between different models, please check whether this function is available according to the product model resource table before checking.

Table 1-1 CH572 pin definitions

Pin No.			Pin name	Pin type(1)	Alternate function takes precedence(2)	Function Description
CH572Q	CH572R	CH572D				
0	-	0	GND	P	-	EPAD, common ground, voltage 0V reference point.
-	15	-	GND	P	-	Common ground, voltage 0V reference point.
7	10	1	PA3	I/O/A	/CMP_P0 /TXD /SCL_2 /SCK_ /PWM3 /RXD_1 /KEYSCAN1	PA3: General-purpose bidirectional digital I/O pin. CMP_P0: Positive input terminal of the comparator. TXD: UART serial data output. SCL_2: I2C serial clock pin mapping. SCK_: SCK pin mapping of SPI. PWM3: Pulse width modulation output channel 3. RXD_1: RXD pin mapping of UART. KEYSCAN1: Key scan input 1.
8	12	2	XO	O/A	-	The inverting output terminal of the high-frequency oscillator HSE is connected to one end of the 32MHz crystal.
9	13	3	XI	A	-	The input terminal of the high-frequency oscillator HSE is connected to the other end of the 32MHz crystal.
-	14	4,19	NC.	NC.	-	Reserved
10	16	5	ANT	A	-	RF signal input and output, it is recommended to connect directly to the antenna.
1	5	6	VDD33	P	-	To simulate the power supply input, 2.2uF parallel 0.1uF capacitors must be connected close to the pin. If powered by a single 3.3V supply, the VDD33 pin input is rated 3.3V supply, the V5 pin is floating or shorted to the VDD33 pin.
		7	VIO33	P	-	The I/O power input needs to be shorted to the VDD33 pin.
2	-	8	V5	P	-	When a single 5V power supply is used, the rated 5V power supply is input to the V5 pin to supply power to the internal LDO5V, and the LDO5V generates 3.3V at the VDD33 pin. A 2.2uF capacitor should be connected close to the pin, and a rated 1.5KΩ resistor should be connected in series between the V5 and VDD33 pins. When a single 3.3V power supply is used, the V5 pin is suspended or shorted to the VDD33 pin.
-	1	9	PA9	I/O/5VT	SDA /TMR_3 /RXD_5	PA9: Universal bidirectional digital I/O pin. SDA: I2C serial data pin, open-drain output and input. TMR_3: TMR pin mapping of timer.

Pin No.			Pin name	Pin type[1]	Alternate function takes precedence[2]	Function Description
CH572Q	CH572R	CH572D				
					/CAP_IN1_3 /CAP_IN2_2	RXD_5: RXD pin mapping of UART. CAP_IN1_3: The timer captures the mapping of input channel 1. CAP_IN2_2: The timer captures the mapping of input channel 2.
-	2	10	PA8	I/O	TXD_5 /SCL /PWM5 /RST /KEYSCAN2	PA8: General bidirectional digital I/O pin. TXD_5: TXD pin mapping of UART SCL: I2C serial clock pin, master output and input/slave input. PWM5: Pulse width modulation output channel 5. RST: External reset input, active low, built-in pull-up resistor. KEYSCAN2: Key scan input 2.
3	3	11	PA0	I/O/A	UDM /SWDIO /TXD_3 /SCL_1 /RXD_2	PA0: General bidirectional digital I/O pin. UDM: D- data line of full-speed USB 2.0. SWDIO[3]: Simulate serial data input and output of debugging interface. TXD_3: TXD pin mapping of UART. SCL_1: SCL pin mapping of I2C. RXD_2: RXD pin mapping of UART.
4	4	12	PA1	I/O/A	UDP /SWCLK /TXD_2 /SDA_1 /RXD_3	PA1: General bidirectional digital I/O pin. UDP: D+ data line of full-speed USB 2.0 SWCLK[3]: Simulate serial clock input of debugging interface. TXD_2: TXD pin mapping of UART. SDA_1: SDA pin mapping of I2C. RXD_3: RXD pin mapping of UART.
-	11	13	PA4	I/O	X25MO /PWM4 /TMR_2 /SCS /CAP_IN1_2 /CAP_IN2_3	PA4: General bidirectional digital I/O pin. X25MO: 25MHz clock output. PWM4: Pulse width modulation output channel 4. TMR_2: TMR pin mapping of timer. SCS: CS input of SPI slave mode, active low. CAP_IN1_2: Mapping of capture input channel 1 of timer. CAP_IN2_3: Mapping of capture input channel 2 of timer.
-	6	14	PA5	I/O	SCL_3/SCK	PA5: General bidirectional digital I/O pin. SCL_3: SCL pin mapping of I2C. SCK: SPI serial clock pin, host output/slave input.
-	7	15	PA6	I/O	SDA_3 /MISO /RXD_4	PA6: General bidirectional digital I/O pin. SDA_3: SDA pin mapping of I2C. MISO: SPI serial data pin, host input.

Pin No.			Pin name	Pin type(1)	Alternate function takes precedence(2)	Function Description
CH572Q	CH572R	CH572D				
						RXD_4: RXD pin mapping of UART.
5	8	16	PA7	I/O/A	CMP_P1 /TXD_4 /MOSI /PWM1 /TMR /RST_ /CAP_IN1 /CAP_IN2_1	PA7: General bidirectional digital I/O pin. CMP_P1: Input positive terminal of comparator. TXD_4: TXD pin mapping of UART. MOSI: SPI serial data pin, host output. PWM1: Pulse width modulation output channel 1. TMR: Capture input 1 of timer 1 and PWM output channel 1. RST_: RST pin mapping. CAP_IN1: Capture input channel 1 of timer. CAP_IN2_1: Mapping of capture input channel 2 of timer.
-	-	17	PA10	I/O	TXD_7 /RXD_6 /KEYSCAN3	PA10: General bidirectional digital I/O pin. TXD_7: TXD pin mapping of UART. RXD_6: RXD pin mapping of UART. KEYSCAN3: Key scan input 3.
-	-	18	PA11	I/O	TXD_6/RXD_7 /KEYSCAN4	PA11: General bidirectional digital I/O pin. TXD_6: TXD pin mapping of UART. RXD_7: RXD pin mapping of UART. KEYSCAN4: Key scan input 4.
6	9	20	PA2	I/O/A	CMP_N /TXD_1 /SDA_2 /PWM2 /TMR_1 /RXD /SCS_ /KEYSCAN0 /CAP_IN1_1 /CAP_IN2	PA2: General bidirectional digital I/O pin. CMP_N: Negative input terminal of the comparator. TXD_1: TXD pin mapping of UART. SDA_2: SDA pin mapping of I2C. PWM2: Pulse width modulation output channel 2. TMR_1: TMR pin mapping of timer. RXD: Serial data input of UART. SCS_: SCS pin mapping of SPI. KEYSCAN0: Key scan input 0. CAP_IN1_1: Mapping of capture input channel 1 of timer. CAP_IN2: Capture input channel 2 of timer.

Table 1-2 CH570 pin definitions

Pin No.			Pin name	Pin type(1)	Alternate function takes precedence(2)	Function Description
CH570E	CH570Q	CH570D				
7	0	0	GND	P	-	EPAD, common ground terminal, voltage 0V reference point.
-	7	1	PA3	I/O/A	/CMP_P0 /TXD /SCL_2 /SCK_ /PWM3 /RXD_1 /KEYSCAN1	PA3: General bidirectional digital I/0 pin. CMP_P0: Positive input terminal of the comparator. TXD: UART serial data output. SCL_2: I2C serial clock pin mapping. SCK_: SCK pin mapping of SPI. PWM3: Pulse width modulation output channel 3. RXD_1: RXD pin mapping of UART. KEYSCAN1: Key scan input 1.
5	8	2	XO	O/A	-	The inverting output terminal of the high-frequency oscillator HSE is connected to one end of the 32MHz crystal.
6	9	3	XI	A	-	The input terminal of the high-frequency oscillator HSE is connected to the other end of the 32MHz crystal.
-	-	4,19	NC.	NC.	-	Reserved.
8	10	5	ANT	A	-	RF signal input and output, it is recommended to connect directly to the antenna.
1	1	6	VDD33	P	-	Analog power input, need to be close to the pin external 2.2uF parallel 0.1uF capacitor. If it is powered by a single 3.3V power supply, input the rated 3.3V power supply to the pin VDD33, and the pin V5 is suspended or shorted to the pin VDD33.
		7	VIO33	P	-	I/O power supply input, which needs to be shorted with VDD33 pin.
-	2	8	V5	P	-	When a single 5V power supply is used, the rated 5V power supply is input to the V5 pin to supply power to the internal LDO5V, and the LDO5V generates 3.3V at the VDD33 pin. A 2.2uF capacitor should be connected close to the pin, and a rated 1.5KΩ resistor should be connected in series between the V5 and VDD33 pins. When a single 3.3V power supply is used, the V5 pin is suspended or shorted to the VDD33 pin.
-	-	9	PA9	I/O/5VT	SDA /TMR_3 /RXD_5 /CAP_IN1_3 /CAP_IN2_2	PA9: General bidirectional digital I/O pin. SDA: I2C serial data pin, open-drain output and input. TMR_3: TMR pin mapping of timer. RXD_5: RXD pin mapping of UART. CAP_IN1_3: Mapping of capture input channel 1 of

Pin No.			Pin name	Pin type(1)	Alternate function takes precedence(2)	Function Description
CH570E	CH570Q	CH570D				
						timer. CAP_IN2_2: Mapping of capture input channel 2 of timer.
-	-	10	PA8	I/O	TXD_5 /SCL /PWM5 /RST /KEYSCAN2	PA8: General bidirectional digital I/O pin. TXD_5: TXD pin mapping of UART. SCL: I2C serial clock pin, host output and input/slave input. PWM5: Pulse width modulation output channel 5. RST: External reset input, active low, built-in pull-up resistor. KEYSCAN2: Key scan input 2.
2	3	11	PA0	I/O/A	UDM /SWDIO /TXD_3 /SCL_1 /RXD_2	PA0: General bidirectional digital I/O pin. UDM: D- data line of full-speed USB 2.0. SWDIO(3): Serial data input and output of simulate debug interface. TXD_3: TXD pin mapping of UART. SCL_1: SCL pin mapping of I2C. RXD_2: RXD pin mapping of UART.
3	4	12	PA1	I/O/A	UDP /SWCLK /TXD_2 /SDA_1 /RXD_3	PA1: General bidirectional digital I/O pin. UDP: D+ data line of full-speed USB 2.0. SWCLK(3): Serial clock input of simulate debug interface. TXD_2: TXD pin mapping of UART. SDA_1: SDA pin mapping of I2C. RXD_3: RXD pin mapping of UART.
-	-	13	PA4	I/O	X25MO /PWM4 /TMR_2 /SCS /CAP_IN1_2 /CAP_IN2_3	PA4: General bidirectional digital I/O pin. X25MO: 25MHz clock output. PWM4: Pulse width modulation output channel 4. TMR_2: TMR pin mapping of timer. SCS: CS input of SPI slave mode, active low. CAP_IN1_2: Mapping of timer capture input channel 1. CAP_IN2_3: Mapping of timer capture input channel 2.
-	-	14	PA5	I/O	SCL_3 /SCK	PA5: General bidirectional digital I/O pin. SCL_3: SCL pin mapping of I2C. SCK: SPI serial clock pin, host output/slave input.
-	-	15	PA6	I/O	SDA_3 /MISO /RXD_4	PA6: General bidirectional digital I/O pin. SDA_3: SDA pin mapping of I2C. MISO: SPI serial data pin, host input. RXD_4: RXD pin mapping of UART.
4	5	16	PA7	I/O/A	CMP_P1	PA7: General bidirectional digital I/O pin.

Pin No.			Pin name	Pin type(1)	Alternate function takes precedence(2)	Function Description
CH570E	CH570Q	CH570D				
					/TXD_4 /MOSI /PWM1 /TMR /RST_ /CAP_IN1 /CAP_IN2_1	CMP_P1: Positive input terminal of the comparator. TXD_4: TXD pin mapping of UART. MOSI: SPI serial data pin, host output. PWM1: Pulse width modulation output channel 1. TMR: Capture input 1 of timer 1 and PWM output channel 1. RST_: RST pin mapping. CAP_IN1: Capture input channel 1 of timer. CAP_IN2_1: Mapping of timer capture input channel 2.
-	-	17	PA10	I/O	RXD_6 /TXD_7 /KEYSCAN3	PA10: General bidirectional digital I/O pin. RXD_6: RXD pin mapping of UART. TXD_7: TXD pin mapping of UART.。 KEYSCAN3: Key scan input 3.
-	-	18	PA11	I/O	TXD_6 /RXD_7 /KEYSCAN4	PA11: General bidirectional digital I/O pin. TXD_6: TXD pin mapping of UART. RXD_7: RXD pin mapping of UART. KEYSCAN4: Key scan input 4.
-	6	20	PA2	I/O/A	CMP_N /TXD_1 /SDA_2 /PWM2 /TMR_1 /RXD /SCS_ /KEYSCAN0 /CAP_IN1_1 /CAP_IN2	PA2: General bidirectional digital I/O pin. CMP_N: Input negative terminal of the comparator. TXD_1: TXD pin mapping of UART. SDA_2: SDA pin mapping of I2C. PWM2: Pulse width modulation output channel 2. TMR_1: TMR pin mapping of timer. RXD: Serial data input of UART. SCS_: SCS pin mapping of SPI. KEYSCAN0: Key scan input 0. CAP_IN1_1: Mapping of timer capture input channel 1. CAP_IN2: Capture input channel 2 of the timer.

Note:

1. Pin Type: P=Power; I=TTL/CMOS level Schmitt input; O=CMOS level 3-state output;
A= Analog signal input or output; 5VT= Support 5V signal voltage input.

2. 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.

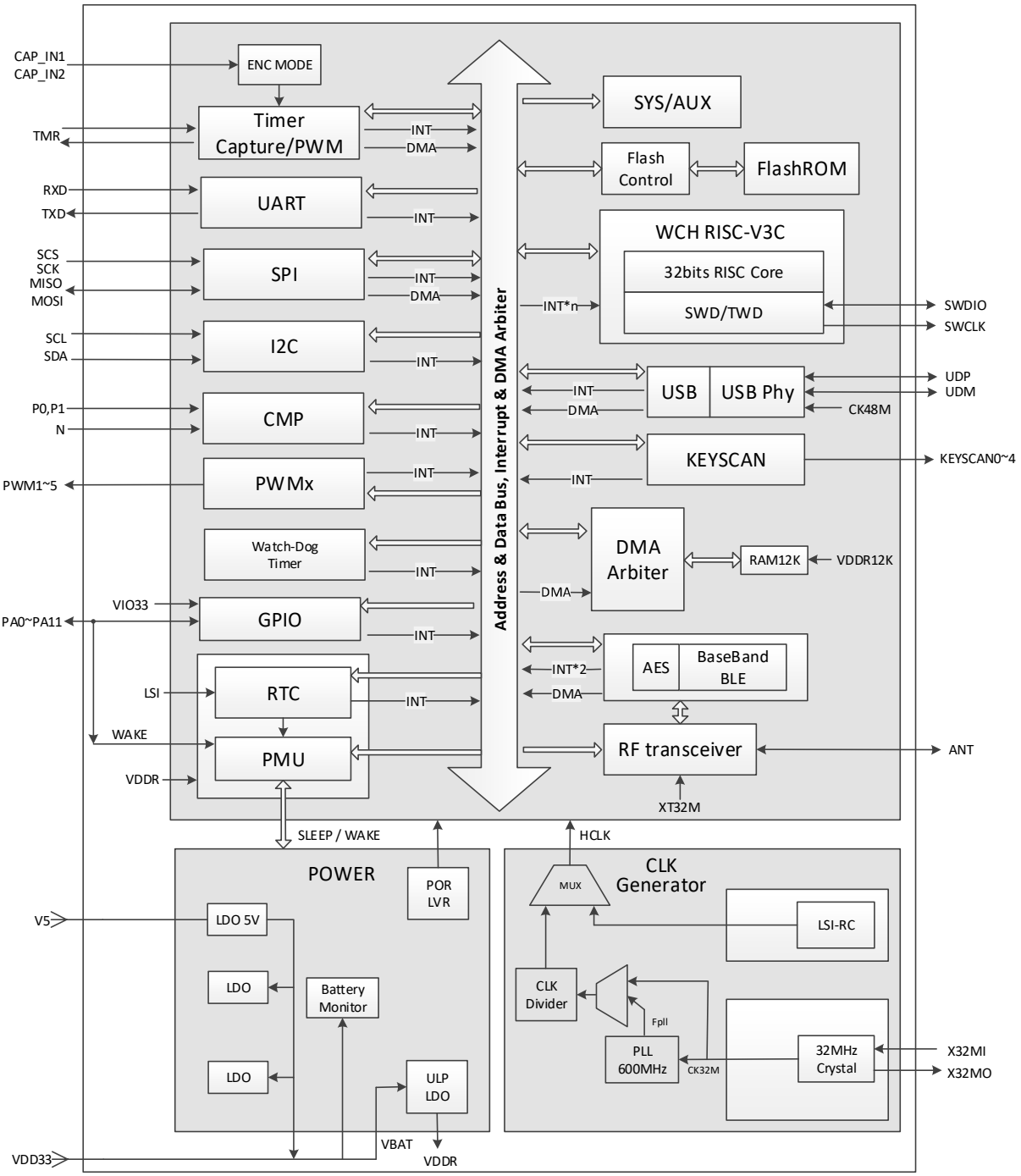
3. After the system is powered on or reset, the default debug interface pin function is enabled. The debug interface can be turned on by configuring the register as needed. After the emulation debug interface is enabled, PA0 and PA1 are used only as SWDIO and SWCLK and are no longer used for GPIO or peripheral multiplexing function pins. After the simulation debugging interface is turned off, PA0 and PA1 can be used for GPIO and peripheral multiplexing function pins.

Chapter 2 System Architecture and Memory

2.1 System Architecture

The following figure shows the system architecture block diagram of CH572. Its core is QingKe RISC-V microprocessor, please refer to *QingKeV3 Processor Manual* for details.

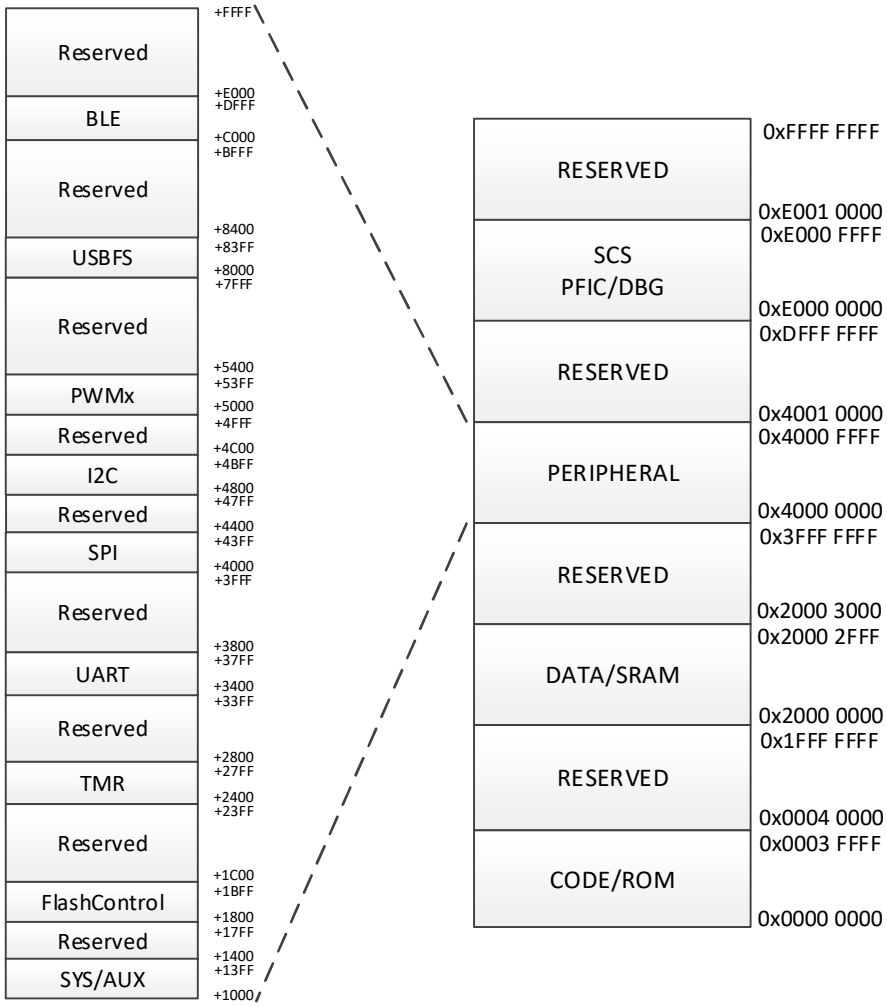
Figure 2-1 CH572 internal structure block diagram



2.2 Memory Mapping

The addressing space of CH572 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	-
0x20000000-0x20002FFF	On-chip DATA area, volatile memory	12KB, SRAM
0x20003000-0x3FFFFFFF	Reserved	-
0x40000000-0x4000FFFF	Various peripherals	Multiple peripheral modules
0x40010000-0xEFFFFFFF	Reserved	-
0xE0000000-0xE000FFFF	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
0x0003C000-0x0003DFFF	User non-volatile data memory, DataFlash	8KB
0x0003E000-0x0003FFFF	System non-volatile configuration information memory, InfoFlash	8KB

The configuration information of the addresses 0x0003E000-0x0003EEFF can be set by the user through tools.

Table 2-3 Description of user non-volatile configuration information

Bit address	Name	Application	Default Value
Bit 2~Bit 0	RESERVED	Reserved	0
Bit 3	CFG_RESET_EN	RST external manual reset input pin enable.	0
Bit 4	CFG_RST_PIN_EN	External reset pin selection: 1: PA7; 0: PA8.	0
Bit 5	CFG_IWDG_EN	Independent watchdog enable: 1: Enable IWDG; 0: Disable IWDG.	0
Bit 6	CFG_BOOT_EN	BootLoader enable.	1b
Bit 15~Bit 7	RESERVED	Reserved	0
Bit 23~ Bit 16	CFG_ROM_READ	Code and data protection mode in FlashROM: 3A: Enable readout; ！3A: Disable 2-wire debug to read out, and keep the program secret;	3Ah
Bit 27~Bit 24	RESERVED	Reserved	0
Bit 31~Bit 28	VALID_SIG	Configuration information valid flag, fixed value	0100b

Note: Disable FlashROM code and data protection mode: completely erase the user area, and write CFG_ROM_READ(bit[23:16]) of the configuration information as 0x3A.

2.3.2 On-chip DATA Area Mapping Table

Table 2-4 DATA area address

Address range	Application	Description
0x20000000-0x20002FFF	Independently maintainable memory area supplied by the main + auxiliary dual power, RAM12K	12KB

2.3.3 Peripheral Address Assignment

CH572 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 2-5 Peripheral base address assignment table

Peripheral No.	Peripheral name	Peripheral base address
1	SYS (PMU/RTC/GPIO/KEY SCAN, etc.) AUX (PLL/CMP, etc.)	0x4000 1000
2	FlashROM-Control	0x4000 1800
3	TMR	0x4000 2400
4	UART	0x4000 3400
5	SPI	0x4000 4000
6	I2C	0x4000 4800
7	PWMx (PWM1~PWM5)	0x4000 5000
8	USBFS	0x4000 8000
9	Radio: BLE	0x4000 C000 0x4000 D000

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

Table 2-6 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 2-7 Description of noun abbreviation

Abbreviation	Description
HSE	External high-frequency crystal oscillator clock source (32MHz recommended)
LSI	Internal low-frequency RC clock oscillator source
CK32M	High-frequency clock source (32MHz by default)
T_{CK32M}	High-frequency clock cycles (1/CK32M)
Fpll	PLL output clock (600MHz by default)
HCLK	System clock
Fsys	System clock frequency
Tsys	System clock cycle (1/Fsys)
RAM12K	12KB SRAM
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 20 peripheral interrupt channels and 8 core interrupt channels, and other interrupt sources are reserved.

3.1 Interrupt Controller

20 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	-	-	-	Reserved	-
17	2	Settable	GPIO_A	GPIO port PA interrupt	0x0000_0044
18	-	-	-	Reserved	-
19	3	Settable	SPI	SPI interrupt	0x0000_004C
20	4	Settable	BLEB	BB interrupt of wireless module	0x0000_0050
21	5	Settable	BLEL	LLE interrupt of wireless module	0x0000_0054
22	6	Settable	USB	USB interrupt	0x0000_0058
23	-	-	-	Reserved	0x0000_005C
24	7	Settable	TMR	TMR timer interrupt	0x0000_0060
25	-	-	-	Reserved	-
26	-	-	-	Reserved	-
27	8	Settable	UART	UART interrupt	0x0000_006C
28	9	Settable	RTC	RTC and oscillator capture completion interrupt.	0x0000_0070
29	10	Settable	CMP	CMP interrupt	0x0000_0074
30	11	Settable	I2C	I2C interrupt	0x0000_0078
31	12	Settable	PWMx	PWMx interrupt	0x0000_007C
32	-	-	-	Reserved	-
33	13	Settable	KEYSCAN	Key scan interrupt	0x0000_0080
34	14	Settable	ENCODER	Encoder terminal	0x0000_0084
35	15	Settable	WDOG_BAT	Watchdog timer interrupt/battery low voltage interrupt	0x0000_008C

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	0x0000032C
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_RESTSYS	0x48	PFIC soft reset 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_FIADDRR0	0x60	PFIC fast interrupt 0 address register	0x00000000
R32_PFIC_FIADDRR1	0x64	PFIC fast interrupt 1 address register	0x00000000
R32_PFIC_FIADDRR2	0x68	PFIC fast interrupt 2 address register	0x00000000
R32_PFIC_FIADDRR3	0x6C	PFIC fast interrupt 3 address register	0x00000000
R32_PFIC_IENR1	0x100	PFIC interrupt enable set register 1	0x00000000

R32_PFIC_IENR2	0x104	PFIC interrupt enable set register 2	0x00000000
R32_PFIC_IRER1	0x180	PFIC interrupt enable clear register 1	0x00000000
R32_PFIC_IRER2	0x184	PFIC interrupt enable clear 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 clear register 1	0x00000000
R32_PFIC_IPRR2	0x284	PFIC interrupt pending clear register 2	0x00000000
R32_PFIC_IACTR1	0x300	PFIC interrupt activation status register 1	0x00000000
R32_PFIC_IACTR2	0x304	PFIC interrupt activation status register 2	0x00000000
R32_PFIC_IPRIORx	0x400	PFIC interrupt priority configuration 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 (R32_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.	00000h
[11:0]	Reserved	RO	Reserved. Reset, NMI, EXC, ECALL and other interrupts bit, the same below.	32Ch

PFIC Interrupt Enable Status Register 2 (R32_PFIC_ISR2)

Bit	Name	Access	Description	Reset value
[31:4]	Reserved	RO	Reserved	0000000h
[3:0]	INTENSTA	RO	32# and above interrupts current enable status. 1: Enable the current number interrupt; 0: Disable the current number interrupt.	0h

PFIC Interrupt Pending Status Register 1 (R32_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: Current number interrupt is not pending.	00000h
[11:0]	Reserved	RO	Reserved	000h

PFIC Interrupt Pending Status Register 2 (R32_PFIC_IPR2)

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved	0000000h
[7:0]	PENDSTA	RO	32# and above interrupts current pending status. 1: Current number interrupt has been pending; 0: Current number interrupt is not pending.	0h

PFIC Interrupt Priority Threshold Configuration Register (R32_PFIC_ITHRESDR)

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved	000000h
[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. [7:5]: Priority threshold. [4:0]: Reserved; 0 constantly; invalid if writing.	00h

PFIC Soft Reset Register (R32_PFIC_RESTSYS)

Bit	Name	Access	Description	Reset value
[31:16]	KEYCODE	WO	Reset the keyword bit. When the keyword matches, the operation configured by bit RESETSYS is allowed, with KEY=0xBEEF, otherwise there is no response.	0000h
[15:8]	Reserved	RO	Reserved	00h
7	RESETSYS	WO	System reset. Write 1 to perform system reset when keywords match, otherwise it will be invalid.	0
[6:0]	Reserved	RO	Reserved	00h

PFIC Interrupt Global Status Register (R32_PFIC_GISR)

Bit	Name	Access	Description	Reset value
[31:14]	Reserved	RO	Reserved	0
13	CPU_LOCK_STA	RO	The current core is locked: 1: Locked; 0: Not locked.	0
12	CPU_DBG_MODE	RO	The current core is in the debug state: 1: Debugging; 0: Non-debugging.	0
11	CPU_GL0BL_IE	RO	Global interrupt enable: 1: Enable interrupt; 0: Disable interrupt.	0
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 (R32_PFIC_IDCFGR)

Bit	Name	Access	Description	Reset value
[31:24]	FIID3	RW	Configure interrupt number of fast interrupt 3.	00h
[23:16]	FIID2	RW	Configure interrupt number of fast interrupt 2.	00h
[15:8]	FIID1	RW	Configure interrupt number of fast interrupt 1.	00h
[7:0]	FIID0	RW	Configure interrupt number of fast interrupt 0.	00h

PFIC Fast Interrupt 0 Address Register (R32_PFIC_FIADDRR0)

Bit	Name	Access	Description	Reset value
[31:1]	ADDR0	RW	Fast interrupt 0 service program address bit[31:1], bit0 is 0.	00000000h
0	FI0EN	RW	Fast interrupt 0 channel enable bit: 1: Enable fast interrupt 0 channel; 0: Disable.	0

PFIC Fast Interrupt 1 Address Register (R32_PFIC_FIADDRR1)

Bit	Name	Access	Description	Reset value
[31:1]	ADDR1	RW	Fast interrupt 1 service program address bit[31:1], bit0 is 0.	00000000h
0	FI1EN	RW	Fast interrupt 1 channel enable bit: 1: Enable fast interrupt 1 channel; 0: Disable.	0

PFIC Fast Interrupt 2 Address Register (R32_PFIC_FIADDRR2)

Bit	Name	Access	Description	Reset value
[31:1]	ADDR2	RW	Fast interrupt 2 service program address bit[31:1], bit0 is 0.	00000000h
0	FI2EN	RW	Fast interrupt 2 channel enable bit: 1: Enable fast interrupt 2 channel; 0: Disable.	0

PFIC Fast Interrupt 3 Address Register (R32_PFIC_FIADDRR3)

Bit	Name	Access	Description	Reset value
[31:1]	ADDR3	RW	Fast interrupt 3 service program address bit[31:1], bit0 is 0.	00000000h
0	FI3EN	RW	Fast interrupt 3 channel enable bit: 1: Enable fast interrupt 3 channel; 0: Disable.	0

PFIC Interrupt Enable Set Register 1 (R32_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.	00000h
[11:0]	Reserved	RO	Reserved	000h

PFIC Interrupt Enable Set Register 2 (R32_PFIC_IENR2)

Bit	Name	Access	Description	Reset value
[31:4]	Reserved	RO	Reserved	0000000h
[3:0]	INTEN	WO	32# and above interrupts enable. 1: Enable the current number interrupt;	0h

			0: No effect.	

PFIC Interrupt Enable Clear Register 1 (R32_PFIC_IRER1)

Bit	Name	Access	Description	Reset value
[31:12]	INTRESET	WO	31# and below interrupts enable. 1: Enable the current number interrupt; 0: No effect.	00000h
[11:0]	Reserved	RO	Reserved	000h

PFIC Interrupt Enable Clear Register 2 (R32_PFIC_IRER2)

Bit	Name	Access	Description	Reset value
[31:4]	Reserved	RO	Reserved	0000000h
[3:0]	INTEN	WO	32# and above interrupts enable. 1: Enable the current number interrupt; 0: No effect.	0h

PFIC Interrupt Pending Set Register 1 (R32_PFIC_IPSR1)

Bit	Name	Access	Description	Reset value
[31:12]	PENDSET	WO	31# and below interrupts pending set. 1: Current number interrupt is pending; 0: No effect.	00000h
[11:0]	Reserved	RO	Reserved	000h

PFIC Interrupt Pending Set Register 2 (R32_PFIC_IPSR2)

Bit	Name	Access	Description	Reset value
[31:4]	Reserved	RO	Reserved	0000000h
[3:0]	PENDSET	WO	32# and above interrupts pending set. 1: Current number interrupt is pending; 0: No effect.	0h

PFIC Interrupt Pending Clear Register 1 (R32_PFIC_IPRR1)

Bit	Name	Access	Description	Reset value
[31:12]	PENDSET	WO	31# and below interrupts pending clear. 1: Current number interrupt is pending; 0: No effect.	00000h
[11:0]	Reserved	RO	Reserved	000h

PFIC Interrupt Pending Clear Register 2 (R32_PFIC_IPRR2)

Bit	Name	Access	Description	Reset value
[31:4]	Reserved	RO	Reserved	0000000h
[3:0]	PENDSET	WO	32# and above interrupts pending clear. 1: Current number interrupt is pending; 0: No effect.	0h

PFIC Interrupt Activation Status Register 1 (R32_PFIC_IACTR1)

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.	00000h
[11:0]	Reserved	RO	Reserved	000h

PFIC Interrupt Activation Status Register 2 (PFIC_IACTR2)

Bit	Name	Access	Description	Reset value
[31:4]	Reserved	RO	Reserved	0000000h
[3:0]	IACTS	RW1	32# and above interrupts activation status. 1: Executing the current number interrupt; 0: The current number interrupt is not executed.	0h

PFIC Interrupt Priority Configuration Register (R32_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.	00h
…	…	…	…	…
[31:24]	IP_3	RW	Same as IP_0 description.	00h
[23:16]	IP_2	RW	Same as IP_0 description.	00h
[15:8]	IP_1	RW	Same as IP_0 description.	00h
[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.	00h

PFIC System Control Register (R32_PFIC_SCTLR)

Bit	Name	Access	Description	Reset value
31	SYSRESET	WO	System reset. Cleared to 0 automatically. Valid	0

			when writing 1, while invalid when writing 0.	
[30:6]	Reserved	RO	Reserved	0
5	SETEVENT	WO	Set event to wake up the WFE.	0
4	SEVONPEND	RW	Setting a new interrupt into the pending state as a wakeup event allows you to wake up the system from after a WFE instruction, and if the WFE instruction is not executed, it will wake up the system immediately after the next execution of the instruction. 1: New interrupt enters the pending state as a wake-up event 0: New interrupt entering the pending state is not a wake-up event.	0
3	WFITOWFE	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 apace. 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: Some CSR registers need to be accessed by the system in machine mode; accessing them in non-machine mode will cause the chip to enter an exception. Labeled as "MRW", they need to be accessed by the system in machine mode; labeled as "MRO", they are read-only in machine mode.

Interrupt System Control Register (INTSYSCR)

CSR address: 0x804

Bit	Name	Access	Description	Reset value
31	LOCK	MRW	User mode locking flag, after locking, user mode cannot open this group of register configuration. 1: Locked, only configurable in machine mode; 0: Unlocked, configurable in machine/user mode.	0
[30:6]	Reserved	RO	Reserved	0
[5]	HW_POP_OFF	MRW	One-time hardware out of stack off, automatic	0

			reset after exiting interrupts. 1: Mask hardware out stack next time you exit interrupt; 0: Do not mask hardware out stack.	
4	Reserved	RO	Reserved	0
[3:2]	PREEMPT	MRW	The preemption priority bit width configuration register is used for configure that preemption priority bit width in the interrupt priority. 00: The preemption bit width is 0, and interrupts with any priority cannot be nested; 01: The preemption bit width is 1, that is, bit [7] of the interrupt priority register is the preemption priority; 10: The preemption bit width is 2, that is, bits [7:6] of the interrupt priority register are preemption priorities; 11: The preemption bit width is 3, that is, bits [7:5] of the interrupt priority register are preemption priorities.	01b
1	INESTEN	MRW	Interrupt nesting function enable. 1: Enable; 0: Disable.	0
0	HWSTKEN	MRW	Hardware stack protection enable. 1: Hardware stack protection is used when entering an interrupt; 0: hardware stack protection is not used when entering an interrupt.	0

Note: The interrupt nesting feature is turned on and off, controlled by bit NEST_LVL of register INESTCR.

User access to Machine Status Register (USER_ACCESS_MSTATUS)

CSR address: 0x800

Register function setting is the same as MSTATUS register.

Bit	Name	Access	Description	Reset value
[31:13]	Reserved	RO	Reserved	0
[12:11]	MPP	RO	Machine status at exit exception: 00: Machine status set to U mode when exiting an exception; 11: Machine status set to M mode when exiting an exception; 01: Reserved; 10: Reserved.	0
[10:8]	Reserved	RO	Reserved	0
7	MPIE	RW	When bit IE_REMAP_EN of register CORECFGR is enabled, this bit can be read and written in user	0

			mode.	
[6:4]	Reserved	RO	Reserved	0
3	MIE	RW	When bit IE_REMAP_EN of register CORECFGR is enabled, this bit can be read and written in user mode.	0
[2:0]	Reserved	RO	Reserved	0

Machine Status Register (MSTATUS)

CSR address: 0x300

Bit	Name	Access	Description	Reset value
[31:13]	Reserved	MRO	Reserved	0
[12:11]	MPP	MRW	Machine status at exit exception: 00: Machine status set to U mode when exiting an exception; 11: Machine status set to M mode when exiting an exception; 01: Reserved; 10: reserved.	0
[10:8]	Reserved	MRO	Reserved	0
7	MPIE	MRW	Global interrupt enabled after exiting interrupt (Updated to entry MIE value on entry interrupt): 1: Enable global interrupt after exiting interrupt; 0: Disable global interrupt after exiting interrupt.	0
[6:4]	Reserved	MRO	Reserved	0
3	MIE	MRW	Global interrupt enabled on entry to interrupt, (Updated to MPIE value on exit from interrupt): 1: Enable global interrupt; 0: Disable global interrupt.	0
[2:0]	Reserved	MRO	Reserved	0

Machine Trap-vector Base-address Register (MTVEC)

CSR address: 0x305

Bit	Name	Access	Description	Reset value
[31:2]	BASEADD	RO	Interrupt vector table base address.	00000000h
1	MODE1	RW	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	RW	Interrupt or exception entry address mode select: 1: Offset address based on number*4; 0: Unified entry address.	0

Microprocessor Configuration Register (CORECFGR)

CSR address: 0XBC0

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	MRO	Reserved	0
7	HF_NMI	MRW	Hardware error generates NMI: 1: When there is a hardware error, an NMI interrupt is generated 0: When there is a hardware error, an abnormal interrupt is generated; Note: Current hardware errors include only nested overflows.	0
6	Reserved	MRO	Reserved	0
5	IE_REMAP_EN	MRW	MIE register mapping enable: 1: Bits 3 and 7 of CSR address 0x800 are mapped to bit MIE of the MSTATUS register and bit MPIE of the MSTATUS register, respectively; 0: CSR address 0x800 is a read-only register and the return value is the value of the MSTATUS.	0
4	Reserved	MRO	Reserved	0
3	ROM_CACHE_EN	MRW	ROM area instruction cache enable: 1: Enable ROM area instruction cache, which can cache up to 128 bytes of instructions; 0: Disable ROM instruction cache.	0
2	ROM_JUMP_ACC	MRW	ROM area instruction jump acceleration enable: 1. Enable ROM area instruction jump acceleration; 0: Disable ROM area instruction jump acceleration.	0
[1:0]	FETCH_MODE	MRW	Fetching mode: 00: Prefetch disabled. The instruction prefetch function is turned off to avoid invalid instruction fetching operation, and there is at most one valid instruction on the CPU pipeline. This mode has the lowest power consumption, and its performance drops by about 2 ~ 3 times. 01: Prefetch enabled. When the instruction prefetch function is turned on, the CPU will continue to access the instruction memory until the number of instructions to be executed in the internal instruction buffer exceeds a certain number, or the instruction buffer is full. This mode has high power consumption and strong performance. (Failure of CPU prediction will lead to redundant fetch operation, and in some cases, the execution unit will introduce 0 ~ 2 cycles of bubbles, and the performance of most programs	01b

			will not decrease obviously); Other: Reserved.	

Interrupt Nesting Control Register (INESTCR)

CSR address: 0XBC1

Bit	Name	Access	Description	Reset value
31	Reserved	MRO	Reserved	0
30	NEST_OVR	MRW	Hardware error interrupts nested overflow flag bit, and clears it by writing 1: 1: Interrupt overflow flag; 0: Interrupt did not overflow. *Note: Interrupt overflow will only occur when executing secondary interrupt service function to generate instruction exception and NMI interrupt. At this time, the exception and NMI interrupt enter normally, but the CPU stack overflows, so you cannot exit from this exception and NMI interrupt.*	0
[29:12]	Reserved	MRO	Reserved	0
[11:8]	NEST_STA	MRO	Nesting status flag bit: 0000: Non-interruption; 0001: 1-level interrupt; 0011: 2-level interrupt, (1-level nesting); 0111: 3-level interrupt, (2-level nesting, overflow)	0
[7:2]	Reserved	MRO	Reserved	0
[1:0]	NEST_LVL	MRW	Nesting level: 00: Nesting disabled, (Turn off nesting function); 01: 1-level nesting, (Turn on nesting function); 10: Unable to write; 11: Unable to write. *Note 1: Writing 10 or 11 to this field sets the register to 01;* *Note 2: When writing 11, read this register for the highest nesting level of the chip.*	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

<table>
<tr><th>Bit</th><th>Name</th><th>Access</th><th>Description</th><th>Reset value</th></tr>
<tr><td>[31:24]</td><td>pmp3cfg</td><td>MRW</td><td>See pmp0cfg.</td><td>00h</td></tr>
<tr><td>[23:16]</td><td>pmp2cfg</td><td>MRW</td><td>See pmp0cfg.</td><td>00h</td></tr>
<tr><td>[15:8]</td><td>pmp1cfg</td><td>MRW</td><td>See pmp0cfg.</td><td>00h</td></tr>
<tr><td>[7:0]</td><td>pmp0cfg</td><td>MRW</td><td>
<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>7</td><td>L</td><td>Lock enabled, can unlock only in machine mode:
1. Lock related registers; 0: Not locked.</td></tr>
<tr><td>[6:5]</td><td>-</td><td>Reserved</td></tr>
<tr><td>[4:3]</td><td>A</td><td>Address alignment and protection area range selection.</td></tr>
<tr><td>2</td><td>X</td><td>Execute.</td></tr>
<tr><td>1</td><td>W</td><td>Write.</td></tr>
<tr><td>0</td><td>R</td><td>Read.</td></tr>
</table>
</td><td>00h</td></tr>
</table>

Address alignment and protection region range select, memory protection for the region, A_ADDR ≤ 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 ≥ 1, and A_ADDR is $2^{(G+2)}$ aligned, select NAPOT;
3. Otherwise select TOR.

A value	Method	Description
0	OFF	No region to be protected
1	TOR	Top alignment region protection: Under pmp0cfg, 0≤region＜pmpaddr0; Under pmp1cfg, pmpaddr0≤region＜pmpaddr1; Under pmp2cfg, pmpaddr1≤region＜pmpaddr2; Under pmp3cfg, pmpaddr2≤region＜pmpaddr3. $pmpaddr_{i-1}$ = A_ADDR >> 2; $pmpaddr_i$ = B_ADDR >> 2.
2	NA4	Fixed 4-byte region protection. pmp0cfg~pmp3cfg correspond to pmpaddr0~pmpaddr3 as start address. $pmpaddr_i$ = A_ADDR >> 2.
3	NAPOT	$2^{(G+2)}$ region protection, G≥1, and A_ADDR is $2^{(G+2)}$ aligned. $pmpaddr_i$ = (A_ADDR >> 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	00000000h

			2 bits are not used.	

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.	00000000h

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.	00000000h

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.	00000000h

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	0x08	System counter low register	0x00000000
R32_STK_CMPLR	0x10	Count reloaded low register	0x00000000

System Count Control Register (R32_STK_CTRL)

Bit	Name	Access	Description	Reset value
31	SWIE	RW	Software interrupt trigger enable (SWI), same as STK_SR: 1: Trigger software interrupt; 0: Turn off the trigger. Note: This bit must be cleared to 0 after entering software interrupt, otherwise it will be triggered all the time.	0
[30:5]	Reserved	RO	Reserved	0000000h
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

			0: Continue upcounting/downcounting.	
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: 1: Enable counter interrupt; 0: No counter interrupt.	0
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 (R32_STK_SR)

Bit	Name	Access	Description	Reset value
31	Reserved	RW	Software interrupt (SWI) trigger enable: 1: Trigger software interrupt; 0: Turn off the trigger. *Note: This bit must be cleared after entering the software interrupt, otherwise it will always trigger.*	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 (R32_STK_CNTL)

Bit	Name	Access	Description	Reset value
[31:0]	CNTL	RW	STK counter count value lower 32 bits.	00000000h

Count Reloaded Low Register (R32_STK_CMPLR)

Bit	Name	Access	Description	Reset value
[31:0]	CMPL	RW	Set reloaded counter value lower 32 bits.	00000000h

3.4.5 Hardware Breakpoint Setting

The processor supports 4-channel instruction address and data address breakpoints, in which the fixed value of bit TYPE in the TDATA1 register is 2, while other bits in the TDATA1 register conform to the definition of mcontrol in the debugging standard, and support four-channel breakpoints at most. The lower 2 bits of the TSELECT trigger are valid, and the breakpoint channel is selected by the value of the TSELECT register, and then the breakpoint address and control information are configured.

Breakpoint Channel Selection Register (TSELECT)

CSR address: 0x7A0

Bit	Name	Access	Description	Reset value
[31:2]	Reserved	MRO	Reserved	0

[1:0]	TSELECT	MRW	The breakpoint channel selection register, selects the corresponding channel after configuration, and the TDATA1 and TDATA2 registers can be operated to configure breakpoint information.	X

Breakpoint Channel Control Register (TDATA1) (Power-on reset)

CSR address: 0x7A1

Bit	Name	Access	Description	Reset value
[31:28]	TYPE	MRO	Breakpoint type definition, mcontrol type.	0010b
[27]	DMODE	MRO	Debug mode and machine mode can modify the relevant registers of the trigger.	0
[26:21]	MASKMAX	MRO	When MATCH=1, the maximum exponential power range of allowable matching, that is, the maximum allowable matching range of 2^{31} bytes.	011111b
[20:13]	Reserved	MRO	Reserved	0
12	ACTION	MRW	Set the processing mode to be adopted when a breakpoint is triggered: 1: Enter debugging mode when triggered; 0: Enter the breakpoint callback interrupt when triggered.	0
[11:8]	Reserved	MRO	Reserved	0
7	MATCH	MRW	Matching policy configuration: 1: Match when the trigger value is equal to the high m bit of TDATA2, where m = 31–n, where n is the index of the first 0 of TDATA2 (Starting from the low bit); 0: Match when the trigger value is equal to TDATA2.	0
6	M	MRW	Trigger enable in M mode: 1: Enable; 0: Disable.	0
3	U	MRW	Trigger enable in U mode: 1: Enable; 0: Disable.	0
2	EXECUTE	MRW	Instruction read address trigger enable: 1: Enable; 0: Disable.	0
1	STORE	MRW	Data write address trigger enable: 1: Enable; 0: Disable.	0
0	LOAD	MRW	Data read address trigger enable: 1: Enable; 0: Disable.	0

Breakpoint Channel Address Register (TDATA2)

CSR address: 0x7A2

Bit	Name	Access	Description	Reset value
[31:0]	TDATA2	MRW	Used to save matching values.	X

Chapter 4 System Control

4.1 Reset Control

The system supports 6 types of resets, including Real Power on Reset (RPOR), external Manual Reset (MR), internal Software Reset (SR), Watch-dog Timeout Reset (WTR), Global Reset by Waking under Shutdown Mode (GRWSM) and Local Reset by Waking (LRW).

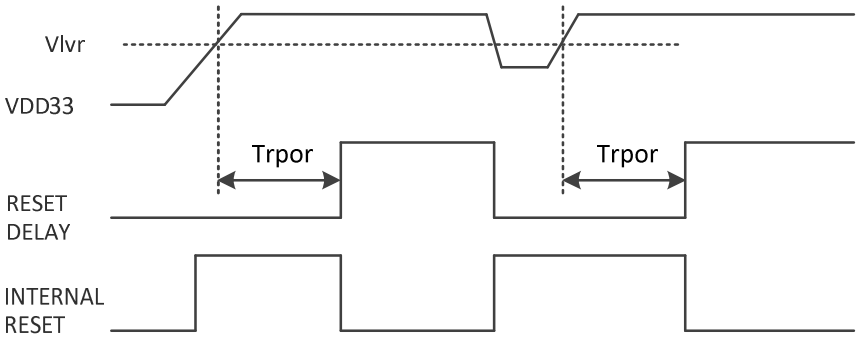
The R8_GLOB_RESET_KEEP register and the RB_ROM_CODE_OFS register are reset only when RPOR or GRWSM occurs, and are not affected by other types of resets.

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 Vlvr, 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.

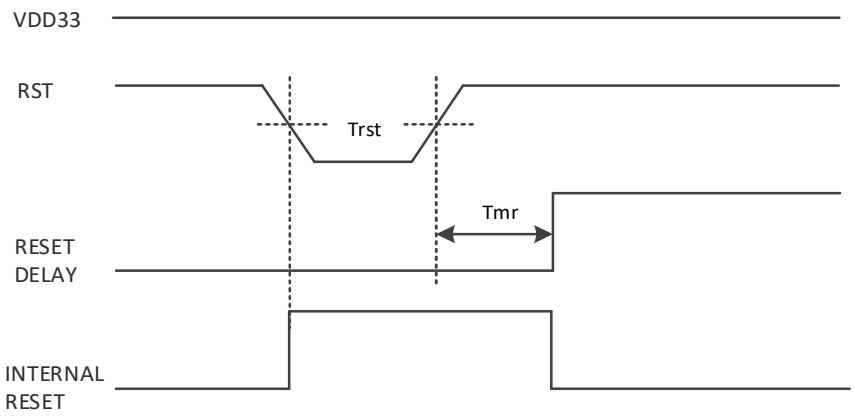
Figure 4-1 Real power on reset



4.1.2 External Manual Reset (MR)

The external manual reset is triggered by a low level externally added to the RST pin. When the duration of reset low level is greater than the minimum reset pulse width (Trst), the system is triggered to reset.

Figure 4-2 External manual reset



4.1.3 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.

4.1.4 Watchdog Timeout Reset (WTR, IWDG)

Watchdog reset includes two methods: Watchdog timeout reset and independent watchdog reset. Watchdog timeout reset function (WTR) is based on an 8-bit incremental counter with a counting clock period of 131072/Fsys. When the watchdog timeout reset function is enabled, the entire system will be reset once this counter overflows.

Inside the Independent Watchdog (IWDG) is a 12-bit counter that runs downward. When the counter value is reduced to 0, a system reset will be generated.

4.1.5 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.

4.1.6 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 16 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 sign register	0x00
R8_CHIP_ID	0x40001041	Chip ID register	0x72/0x70
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	0xX8
R8_RST_WDOG_CTRL	0x40001046	Watchdog and reset configuration register	0x00
R8_GLOB_RESET_KEEP	0x40001047	Reset keep register	0x00
R32_SAFE_ACCESS_SIG2	0x40001058	Safe access sign 2 register	0x07000000
R32_FLASH_DATA	0x40001800	FlashROM word data register	0xXXXXXXXX
R32_FLASH_CONTROL	0x40001804	FlashROM control register	0x074000XX
R8_FLASH_DATA	0x40001804	FlashROM byte data register	0xXX
R8_FLASH_CTRL	0x40001806	FlashROM access control register	0x40
R8_FLASH_CFG	0x40001807	FlashROM access configuration register	0x07

Safe Access Sign Register (R8_SAFE_ACCESS_SIG)

Bit	Name	Access	Description	Reset value

[7:0]	R8_SAFE_ACCESS_SIG	WO	Secure access sign register.。 Some of the registers (Access attribute is RWA) are protected registers and must enter the secure access mode to perform write operations. Write 0x57 and then 0xA8 to this register to enter the secure access mode, and the time limit is about 112 (7*16) main clock cycles (Tsys), beyond which it is automatically protected. Any other value can be written to force the device to exit the secure access mode directly and return to the protected state.	00h
[7:3]	RB_SAFE_ACC_TIMER	RO	Current safe access time count.	000b
2	RB_SAFE_ACC_ACT	RO	Current secure access mode status: 1: Unlocked/writable in secure access mode; 0: Locked, RWA attribute register cannot be rewritten.	0
[1:0]	RB_SAFE_ACC_MODE	RO	Current safe access mode status: 11: Safe mode, which can be written into the RWA register; Other: Non-secure mode.	00b

Chip ID Register (R8_CHIP_ID)

Bit	Name	Access	Description	Reset value
[7:0]	R8_CHIP_ID	RF	CH572: Fixed value 72h, for chip identification. CH572: Fixed value 70h, for chip identification.	72h 70h

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 with preset initial value, which increases automatically all the time, can cycle from 0xFF to 0x00 and then continue. Counting period = 131072/Fsys.	00h

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

Bit	Name	Access	Description	Reset value
[7:6]	RB_ROM_CODE_WE	RWA	FlashROM program memory area CodeFlash erase, programming enable bits: X0: all erase and write protection; 01: 129~240K allow erasing and writing; 11: 0~240K allow erasing and writing.	0
5	RB_ROM_CTRL_EN	RWA	FlashROM access control interface enabled: 1: Allow control; 0: No access	0
4	RB_ROM_CODE_OFS	RWA	Select the address of the user program code's start position offset in FlashROM: 1: 0x008000; 0: 0x000000.	0
3	Reserved	R0	Reserved	0
[2:0]	RB_RESET_FLAG	RO	Last reset status: 000: Software reset SR (This state can be generated by software reset when RB_WDOG_RST_EN=0, otherwise it can be reset without generating this state); 001: Power-on reset RPOR; 010: Watchdog timeout reset; 011: External manual reset MR; 101: Reset GRWSM when waking up from power-down mode; 100/110/111: Wake-up reset LRW, and the previous last reset was SR/WTR/MR respectively.	001b

Global Configuration Information Status Register (R8_GLOB_CFG_INFO)

Bit	Name	Access	Description	Reset value
[7:6]	Reserved	R0	Reserved	11b
5	RB_BOOT_LOADER	RO	Bootloader status: 1: Currently in Bootloader status; 0: Currently in user program status.	1/0
4	RB_CFG_RST_PIN	RO	Reset pin selection: 1: PA7; 0: PA8.	0
3	RB_CFG_BOOT_EN	RO	System BootLoader enable status: 1: Enable; 0: Disable.	1
2	RB_CFG_RESET_EN	RO	RST external manual reset enable status: 1: Enable; 0: Disable.	0

1	Reserved	RO	Reserved	0
0	Reserved	RO	Reserved	0

Watchdog and Reset Configuration Register (R8_RST_WDOG_CTRL)

Bit	Name	Access	Description	Reset value
[7:5]	Reserved	RO	Reserved	000b
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. *Note: After this bit is set to 1, the software reset operation will not affect the RB_RESET_FLAG status.*	0
0	RB_BOOT_LOAD_MAN	RO	Manual entry of the BOOT LOADER flag: 1: Manual BOOT entry; 0: Idle, no action.	0
	RB_SOFTWARE_RESET	WA/WZ	System software reset control; automatically cleared after reset: 1: Perform system software reset; (Bit RB_BOOT_LOAD_MAN is 1 and boot from BOOT when RB_WDOG_INT_EN and RB_ROM_CODE_WE are configured as 1, RB_ROM_DATA_WE is 0, and 128 ≤ R8_WDOG_COUNT < 192, otherwise bit RB_BOOT_LOAD_MAN is 0.)	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	00h

			wake-up reset.	

FlashROM Safe Access Sign 2 Register (R32_SAFE_ACCESS_SIG2)

Bit	Name	Access	Description	Reset value
[31:27]	Reserved	RO	Reserved	0
[26:24]	RB_FUN_MODE	RO	Function enable register.	111b
23	RB_FLASH_HALTED	RO	2-wire Flash operation prohibition sign: 1: No operation; 0: Operation allowed.	0
22	RB_MANU_CFG_LOCK	RO	Vendor configuration word lock flag: 1: Locked; 0: Not locked.	0
21	RB_RD_PROTECT	RO	Read the protection sign: 1: On; 0: Off.	0
20	RB_SAFE_AC_DIS	RO	Safety register automatic closing flag: 1: Automatic shutdown mode is closed; 0: Automatic shutdown mode is still in effect.	0
[19:0]	Reserved	RO	Reserved	0

For the operation or setting of FlashROM, please refer to related subprograms. This datasheet does not provide the introductions to FlashROM word data registers and FlashROM control registers.

4.4 Flash-ROM Operation Steps

1. Erase Flash-ROM, please refer to and call related subprograms.
2. Write Flash-ROM, please refer to and call the related subprograms.
3. Read DataFlash, please refer to and call related subroutines.
4. Read CodeFlash, through a pointer to the program storage space.

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

CH572 has a built-in Power Management Unit (PMU).

When a single 5V power supply is used, a rated 1.5KΩ resistor should be connected in series between V5 and VDD33 pins. The V5 pin inputs a rated 5V power supply, which supplies power for the internal LDO5V, and the LDO5V outputs 3.3V Then, multiple built-in LDO voltage regulators provide the required power supply for the system's FlashROM, digital circuit (including the kernel and USB) and analog circuit (including high-frequency oscillator, PLL, CMP and RF transceiver). 3 low-power consumption modes, such as pause, sleep or power down, are not supported when the 5V power supply is used.

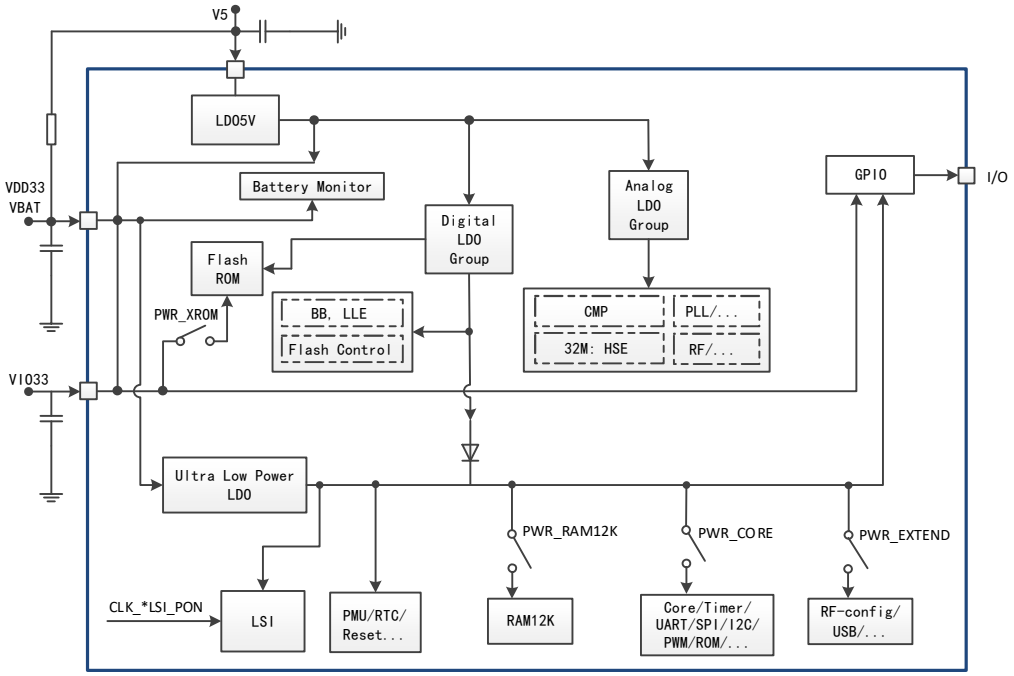
When a single 3.3V power supply is used, the V5 pin is suspended or shorted to VDD33. The VDD33 pin is input with a rated power supply of 3.3V, and multiple built-in LDO voltage regulators provide the required power supply for each module of the system.

The power supply of GPIO and FlashROM is input from VIO33.

The power supply in normal operation is direct power supply. In addition to normal operation, CH572 provides four low-power modes: idle mode, pause mode, sleep mode and power-down mode.

In order to reduce the system power consumption during sleep, you can choose to turn off the main LDO of the system and switch to the ultra-low power ULP-LDO built in the system to provide auxiliary power. When the system enters the sleep or power-down mode, except the constant power supply units such as power management and RTC registers, the 12KB SRAM, core and all peripherals of the system can choose whether to maintain power supply or not, and the LSI can choose whether to turn on or not.

Figure 5-1 Power system



5.2 Register Description

Table 5-1 Power management registers

Name	Access address	Description	Reset value
R16_SLP_CLK_OFF	0x4000100C	Sleep clock control register	0x0000
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	0x4000101C	Long reset configuration register	0x00
R8_SLP_CLK_OFF2	0x4000101D	Sleep clock control register 2	0x00
R16_SLP_WAKE_CFG	0x4000101E	Sleep configuration register	0x0100
R16_POWER_PLAN	0x40001020	Sleep power management register	0x11CF
R16_AUX_POWER_ADJ	0x40001022	Auxiliary power adjustment control register	0x0XXX
R8_BAT_DET_CTRL	0x40001024	Battery voltage detection control register	0x00
R8_BAT_DET_CFG	0x40001025	Battery voltage detection configuration register	0x02
R8_BAT_STATUS	0x40001026	Battery status register	0x00

Sleep Clock Control Register 0 (R8_SLP_CLK_OFF0)

Bit	Name	Access	Description	Reset value
7	RB_SLP_KEYSCAN_WAKE	RWA	Enable key scanning events to wake up the system: 1: On; 0: Off.	0
[6:5]	Reserved	RO	Reserved	0
4	RB_SLP_CLK_UART	RWA	Serial clock source: 1: Off; 0: On.	0
[3:2]	Reserved	RO	Reserved	0
1	RB_SLP_CLK_CMP	RWA	Comparator clock source: 1: Off; 0: On.	0
0	RB_SLP_CLK_TMR	RWA	Timer clock source: 1: Off; 0: On.	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
[6:5]	Reserved	RO	Reserved	0
4	RB_SLP_CLK_USB	RWA	Clock source of USBFS controller: 1: Disable; 0: Enable.	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_CLK_OFF_AESCCM	RWA	AES_CCM clock source: 1: Disable; 0: Enable.	0
0	RB_SLP_CLK_SPI	RWA	SPI 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 Wake-on-Edge Mode Enable: 1: RB_SLP_GPIO_EDGE_MODE=1, Wake-on-Edge; RB_SLP_GPIO_EDGE_MODE=0, Wake-on-Rising Edge; 0: Level wake-up.	0
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	GPIO event wakeup event selection when RB_GPIO_WAKE_MODE = 1: 1: Wake-up on any edge regardless of polarity; 0: Wake-up on edge.	0
1	RB_SLP_ENC_WAKE	RWA	Enable encoder event wake-up system: 1: Enable; 0: Disable.	0
0	RB_SLP_USB_WAKE	RWA	Enable USB event wake up system: 1: Enable; 0: Disable.	0

Peripheral Sleep Power Control Register (R8_SLP_POWER_CTRL)

Bit	Name	Access	Description	Reset value
[7:6]	RB_RAM_RET_LV	RWA	Auxiliary power low voltage enabled during SRAM sleep:	0

			00: Disable; 01: Low-power mode 1; 10: Low-power mode 2; 11: Low-power mode 3.	
5	Reserved	RO	Reserved	0
4	RB_SLP_CLK_RAMX	RWA	Clock control of main SRAM (RAM12K): 1: Disable; 0: Enable.	0
3	Reserved	RO	Reserved	0
[2:0]	RB_WAKE_DLY_MOD	RWA	Delay cycle select after wake-up (Fsys): 000: 3584; 001: 512; 010: 64; 011: 1; 100: 8191; 101: 7168; 110: 6144; 111: 4096. *Note: When bit[2] = 1, it is a long delay; when bit[2] = 0, it is a short delay.*	0

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: 11: 32.768ms; 10: 25.000ms; 01: 20.000ms; 00: 15.000ms.	0
0	RB_LONG_RST_EN	RWA	Long reset enable: 1: Enable; 0: Disable.	0

Sleep Clock Control Register 2 (R8_SLP_CLK_OFF2)

Bit	Name	Access	Description	Reset value
[7:5]	Reserved	RO	Reserved	0
4	RB_CLK_OFF_HCLK	RWA	1: Disable the HCLK clock of the FLASH controller; 0: Enable the HCLK clock of the FLASH controller.	0
[3:2]	Reserved	RO	Reserved	0
1	RB_CLK_OFF_DEBUG	RWA	1: Disable the 2-wire debug 32M clock; 0: Enable the 2-wire debug 32M clock.	0
0	RB_CLK_OFF_XROM	RWA	1: Disable the 64M or 600M clock of the	0

			FLASH controller; 0: Enable the 64M or 600M clock of the FLASH controller.	

Sleep Configuration Register (R16_SLP_WAKE_CFG)

Bit	Name	Access	Description	Reset value
[15:9]	Reserved	RO	Reserved	0
8	RB_ACAUTO_ENABLE	RWA	1: The safety register channel is automatically closed; 0: The safety register channel is not automatically closed after it is opened.	1
7	Reserved	RO	Reserved	0
[6:5]	RB_PRECLK_CNT_SEL	RWA	Sleep time configuration, sleep duration = number of cycles *Fsys: 11: 2048; 10: 1024; 01: 512; 00: 256.	0
4	RB_PRECLK_CNT_EN	RWA	1: Wait for a certain amount of time before releasing HCLK when waking up from sleep; 0: HCLK is released immediately after sleep wakeup.	0
[3:2]	Reserved	RO	Reserved	0
0	RB_OSCCLK_RDY_KEEP	RWA	1: The peripheral clock remains on during Halt sleep; 0: The peripheral clock remains off during Halt sleep.	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	Reserved	RWA	Reserved	0
13	Reserved	RWA	Reserved	0
[12:9]	Reserved	RWA	Reserved	1000b
8	RB_PWR_LDO5V_EN	RWA	Internal LDO5V control (sleep planning), pre-sleep configuration: 1: Single V5 supply;	1

			0: Single VDD33 supply.	
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 selection 1: ULL_LDO; 0: LDO.	1
[5:4]	Reserved	RO	Reserved	0
3	RB_PWR_EXTEND	RWA	USBFS and 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_RAM12K	RWA	SRAM power supply of RAM12K (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

Auxiliary Power Adjustment Control Register (R16_AUX_POWER_ADJ)

Bit	Name	Access	Description	Reset value
[15:12]	Reserved	RO	Reserved	0
[11:8]	RB_CFG_IVREF	RO	Reserved. The original value must be kept unchanged when writing.	XXXXb
[7:3]	Reserved	RO	Reserved	00h
[2:0]	RB_ULPLDO_ADJ	RWA	The auxiliary power output voltage adjustment value of ultra-low power LDO (The value is for reference only, and it is not recommended to modify): 000: 0.756V; 001: 0.789V; 010: 0.823V; 011: 0.855V; 100: 0.889V; 101: 0.928V; 110: 0.966V; 111: 1.0V.	100b

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	Reserved	RO	Reserved	0
1	RB_BAT_MON_EN	RWA	Low-power battery voltage monitor	0

			function enable: 1: Enable, increasing about 1uA current; 0: Disable.	
0	RB_PWR_LDO_EN	RWA	Reserved	0

Note: If the battery voltage reaches the voltage detection threshold and both RB_BAT_MON_EN and RB_BAT_LOW_IE are enabled, then an NMI non-maskable interrupt will be generated.

Battery Voltage Detection Configuration Register (R8_BAT_DET_CFG)

Bit	Name	Access	Description	Reset value
[7:2]	Reserved	RO	Reserved	0
[1:0]	RB_BAT_LOW_VTH	RWA	Low voltage detection threshold: 00: 1.8V; 01: 2.0V; 10: 2.2V; 11: 2.4V.	10b

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 indicates that the battery voltage is in a low voltage state: 1: Lower than the low voltage threshold; 0: Not lower than the low voltage threshold.	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 LSI, and to maintain power supply of RAM12K, USB 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, USB and RF configurations are turned off, and you can select whether to turn on LSI, and to maintain power supply of RAM12K. 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 kernel stops running, and the clock system is running, but the clock of each peripheral can be turned off through 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 USB or I2C or SysTick or SPI or TMR or UART or KEYSCAN	System frequency 16MHz, FLASH chip select off, peripheral clock off	1.7mA
Halt	The peripherals are powered normally, the kernel stops running, and the clock system (PLL/HSE) can be configured on or off.	Set SLEEPDEEP=1, execute WFI() or WFE() after setting the wake-up conditions.	I/O or RTC or BAT or USB or KEYSCAN	XT_FORCE_EN=1	1.3mA
				XT_FORCE_EN=0	420uA
Sleep	The main LDO is turned off, the ultra-low power ULP-LDO maintains the power supply of PMU, kernel and basic peripherals, LSI can choose whether to turn on or not, and RAM12K, USB and RF configurations can choose whether to maintain the power supply.	Set SLEEPDEEP=1, execute WFI() or WFE() after setting the wake-up conditions.	I/O or RTC or BAT or KEYSCAN. The chip will continue to run after woken up.	See Table 5-3 for details	0.46uA~1.2uA
Shut down	Ultra-low power LDO maintains PMU power supply, LSI can choose whether to turn on or not, and RAM12K can choose whether to maintain power supply to keep data.	Set SLEEPDEEP=1, set POWER_PLAN, execute WFI() or WFE() after setting the wake-up conditions.	I/O or RTC or BAT or KEYSCAN. The chip will automatically reset after woken up	See Table 5-3 for details	0.3uA～0.9uA

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	RAM12K	LSI	CORE	EXTEND	Power consumption(1) (for reference only)
Maintaining supply function	System power	Data area 12KB	LSI RTC wake-up	CPU core and basic peripherals	USB and RF configurations	PMU and RTC registers are always powered, about 0.3uA
Common configurations in shutdown mode	0	0	0	0	0	0.3uA
	0	1	0	0	0	0.6uA
	0	0	1	0	0	0.56uA
	0	1	1	0	0	0.9uA
Common configurations in sleep mode	0	0	0	1	0	0.46uA
	0	0	0	1	1	0.5uA
	0	1	0	1	0	0.8uA
	0	1	1	1	0	1.1uA
	0	0	1	1	0	0.8uA
	0	1	1	1	1	1.2uA

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

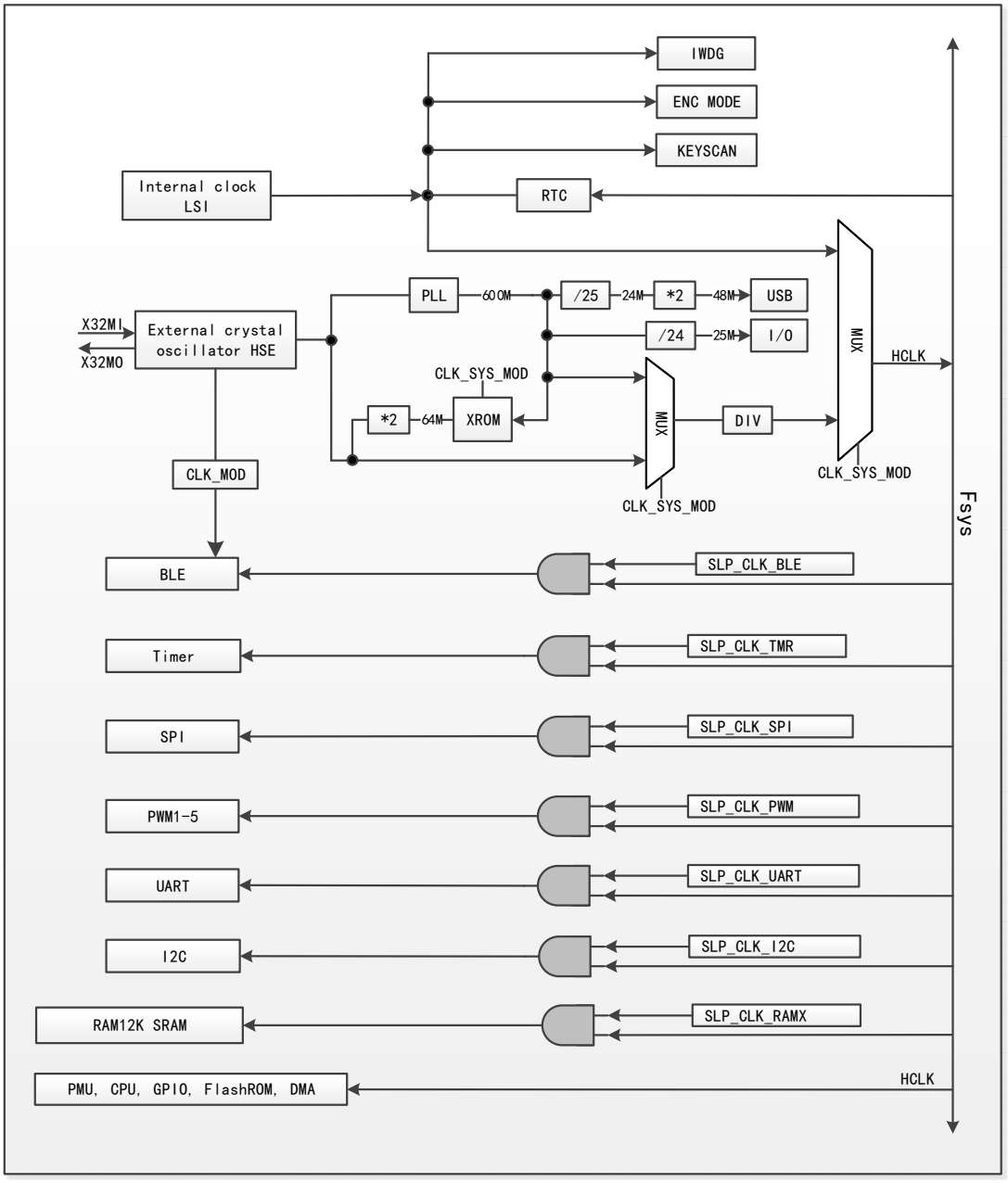
The following different clock sources can be selected to drive the system clock HCLK (Fsys)

- Frequency division of HSE.
- Internal PLL (default 600MHz) frequency division.
- LSI original clock.

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 above picture shows the clock tree structure inside the system, in which the RTC function uses LSI, so the low-frequency clock must be turned on to use these functions; The data transmission of USB depends on the clock source generated by PLL frequency division; Other peripheral drive clocks and digital control logic are driven by system clock or frequency division.

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, the counting of LSI original cycle, 65536 frequency division cycle and 2764800000 frequency division cycle is provided.

6.3 Register Description

Table 6-1 Clock and oscillator control registers

Name	Access address	Description	Reset value
R8_CLK_SYS_CFG	0x40001008	System clock configuration register	0x0005
R8_HFCK_PWR_CTRL	0x4000100A	High frequency clock module power control register	0x14
R8_LSI_CONFIG	0x4000102F	Internal low frequency oscillator LSI configuration register	0xX2
R8_XT32M_TUNE	0x4000104E	External 32MHz clock resonance tune register	0x53
R16_OSC_CAL_CNT	0x40001050	Oscillator capture counter count register	0xXXXX
R8_OSC_CAL_OV_CNT	0x40001052	Oscillator capture counter overflow number register	0x00
R8_OSC_CAL_CTRL	0x40001053	Oscillator capture control register	0x01
R8_PLL_CONFIG	0x4000104B	PLL configuration register	0x0A
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_LSI	0x40001038	RTC count register based on LSI clock cycle	0xXXXX
R16_RTC_CNT_DIV1	0x4000103A	RTC with 65536 LSI clock cycles count registers	0xXXXX

R32_RTC_CNT_DIV2	0x4000103C	RTC with 2831155200 LSI clock cycles count registers	0x0000XXXX

System Clock Configuration Register (R8_CLK_SYS_CFG)

Bit	Name	Access	Description	Reset value
[7:6]	RB_CLK_SYS_MOD	RWA	HCLK system clock source mode selection: 00/10: CK32M (default 32MHz) for frequency division; 01: PLL (default 600MHz) for frequency division; 11: LSI as HCLK.	00b
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.	00101b

Calculation:

Fck32m = XT_32 MHz;

FckLSI = 24~42KHz;

Fpll = Fck32m * 18.75 = 600MHz;

Fsys = RB_CLK_SYS_MOD == 3 ? FckLSI : (RB_CLK_SYS_MOD[0] ? Fpll : Fck32m) / RB_CLK_PLL_DIV;

Power-on default value Fsys = Fck32m / RB_CLK_PLL_DIV = 32MHz / 5 = 6.4MHz;

Fsys range:

Bit RB_CLK_SYS_MOD[1:0]	HCLK system clock source mode	Fsys range
11	LSI	24~42KHz (Run in RAM)
00/10	CK32M divides frequency	1MHz~16MHz
01	PLL divides frequency	18.75MHz~100MHz

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	Used to control the stopping of the clock system in Halt mode: 1: HSE and PLL are not automatically stopped in Halt mode; 0: HSE and PLL are automatically stopped in Halt mode.	0
2	RB_CLK_XT32M_PON	RWA	External 32MHz oscillator HSE power control bits:	1

			1: Power-on; 0: Power-down.	
[1:0]	Reserved	RO	Reserved	0

32KHz Oscillator Configuration Register (R8_CK32K_CONFIG)

Bit	Name	Access	Description	Reset value
7	RB_LSI_CLK_PIN	RO	LSI clock pin status (Asynchronous signal).	X
[6:2]	Reserved	RO	Reserved	0
1	RB_CLK_LSI_PON	RWA	Internal low-frequency RC oscillator LSI power control bits: 1: Power on; 0: Power down.	1
0	Reserved	RO	Reserved	0

External 32MHz Clock Resonance Tune 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 matching with the external 32MHz crystal (which may affect the wireless communication): capacitance = RB_XT32M_C_LOAD * 2+6pF, and 000b~111b respectively correspond to about 6pF~20pF. According to the crystal parameters used, it is usually 111b.	101b
[3:2]	Reserved	RO	Reserved	0
[1:0]	RB_XT32M_I_BIAS	RWA	External 32MHz oscillator bias current selected: 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	Indicates that the RB_OSC_CAL_CNT counter overflows, and write 1 to clear R8_OSC_CAL_OV_CNT.	0
[13:0]	RB_OSC_CAL_CNT	RO	Capture count value based on system main frequency for multiple LSI cycles.	XXXXh

Oscillator Capture Counter Overflow Count Register (R8_OSC_CAL_OV_CNT)

Bit	Name	Access	Description	Reset value
[7:0]	RB_OSC_CAL_OV_CNT	RO	RB_OSC_CAL_CNT count overflow times,	00h

			write 1 to RB_OSC_CAL_OV_CLR to clear this register.	

Oscillator Capture Control Register (R8_OSC_CAL_CTRL)

Bit	Name	Access	Description	Reset value
7	RB_CNT_CLR	RWA	Software reset RB_OSC_CNT_TOTAL: 1: Reset; 0: No operation.	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 capture 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 capture counter count status: 1: Counting is being paused; 0: Counting is in progress.	0
[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
[7:6]	Reserved	RO	Reserved	0
[5:0]	RB_PLL_CFG_DAT	RWA	PLL configuration parameters.	0Ah

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. When the timing mode is enabled, write 1, clear the timing mode activation flag RB_RTC_TMR_FLAG and automatically	1

			cleared to 0.	
[3:0]	Reserved	RO	Reserved	0h

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_DIV2.	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_DUV1; load the low 16 bits of R32_RTC_TRIG to R16_RTC_CNT_LSI.	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 period (timing) selection, unit is one period of LSI: 000: 4096; 001: 8192; 010: 16384; 011: 32768; 100: 65536; 101: 131072; 110: 262144; 111: 524288.	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, in which the upper 16 bits are matched with R16_RTC_CNT_DIV1 and the lower 16 bits are matched with R16_RTC_CNT_LSI. Cooperate with RB_RTC_LOAD_LO and RB_RTC_LOAD_HI to update the current value of RTC counter.	00000000h

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 Register Based on LSI Clock Cycle (R16_RTC_CNT_LSI)

Bit	Name	Access	Description	Reset value
[15:0]	R16_RTC_CNT_LSI	RO	RTC's count register based on LSI clock cycle.	XXXXh

RTC Count Value Register in Units of 65536 LSI Clock Cycles (R16_RTC_CNT_DIV1)

Bit	Name	Access	Description	Reset value
[15:0]	R16_RTC_CNT_DIV 1	RO	The current count value of RTC in units of 65536 LSI clock cycles.	XXXXh

RTC Count Value Register in Units of 2831155200 LSI Clock Cycles (R32_RTC_CNT_DAY)

Bit	Name	Access	Description	Reset value
[31:14]	Reserved	RO	Reserved	0
[13:0]	R32_RTC_CNT_DIV2	RO	RTC takes 2831155200 LSI clock cycles as the single-bit current count value.	XXXXh

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_DIV2 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_DIV1 register and the R16_RTC_CNT_LSI register respectively.

6.4.2 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.

6.4.3 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_LSI (High 16 bits R16_RTC_CNT_DIV1 and low 16 bits R16_RTC_CNT_LSI) plus the interval time DelayTime (in the unit of S), T32 = R32_RTC_CNT_LSI + DelayTime * FckLSI, write T32 into R32_RTC_TRIG register to complete the matching value setting. Example: Assuming that FckLSI is 42KHz and the trigger time is set to 1S, calculate the target time value, T32 = R32 _ RTC _ CNT _ LSI+1 * 42000;

(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_DIV1 and R16_RTC_CNT_LSI 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.

Chapter 7 General-purpose I/O and Alternate Functions

7.1 Introduction to GPIO

The chip provides a group of GPIO ports PA with 12 general input and output pins, all of which have interrupt and wake-up functions, and some pins have multiplexing and mapping functions.

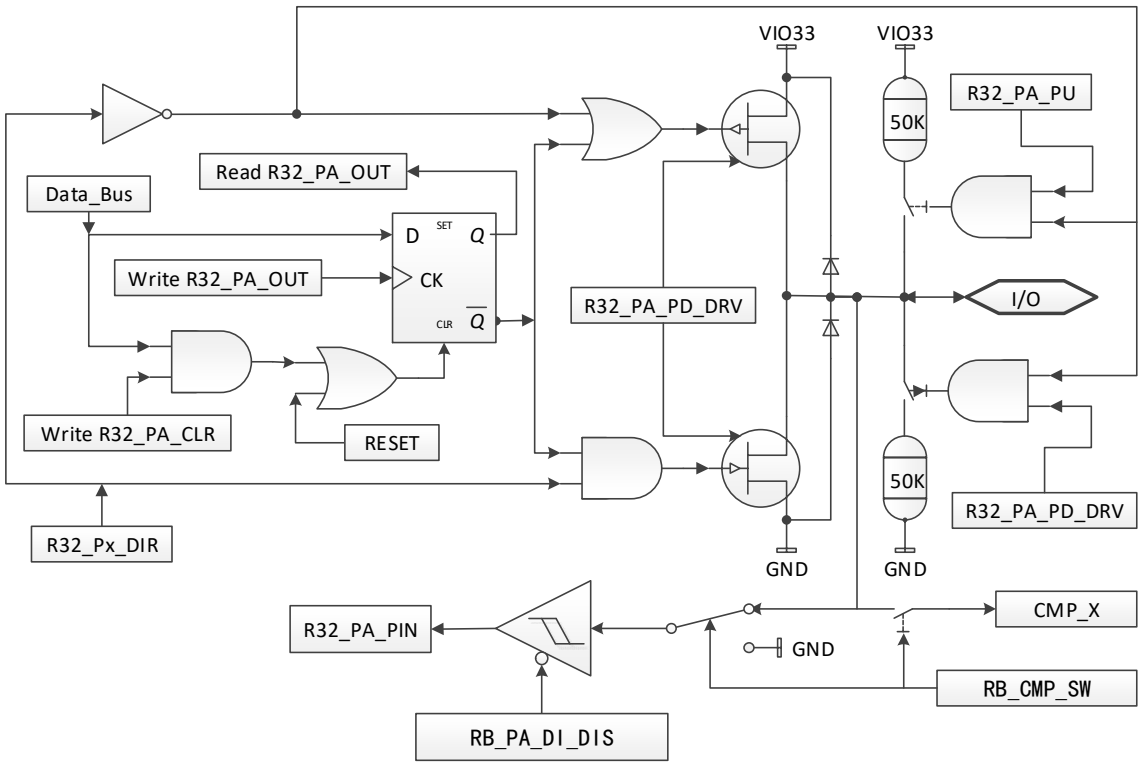
Each GPIO port has a 32-bit direction configuration register R32_PA_DIR, a 32-bit pin input register R32_PA_PIN, a 32-bit data output register R32_PA_OUT, a 32-bit data reset register R32_PA_CLR, a 32-bit pull-up resistor configuration register R32_PA_PU and a 32-bit pull-down resistor/driving capability configuration register R32.

The PA[4]~PA[11] bits in the PA port are valid, corresponding to the 12 GPIO pins on the chip.

Each I/O port bit can be programmed independently, but I/O port registers must be accessed as 8-bit, 16-bit or 32-bit words. If the pin's multiplexing function is not turned on, it is used as a general 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

All I/O pins of the chip have interrupt function and can realize sleep wake-up.

In order to use external interrupts, the port bits must be configured in input mode. Four trigger modes are provided: high level, low level, rising edge and falling edge.

The wake-up function needs to turn on the interrupt enable R16_PA_INT_EN of the port bit and turn on the GPIO wake-up control bit RB_SLP_GPIO_WAKE in the register R8_SLP_WAKE_CTRL.

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.

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

Table 7-1 Timer x

TMR pin	Functional configuration	GPIO configuration
TMR	Input capture channel x	Input (Floating input/pull-up input/pull-down input)
	Output PWM channel x	Push-pull output

Table 7-2 UART

UART pin	Functional configuration	GPIO configuration
TXD	UART transmit x	Push-pull output
RXD	UART receive x	Pull-up input (recommended) or floating input

Table 7-3 SPI

SPI pin	Functional configuration	GPIO configuration
SCK	Clock output in master mode	Push-pull output
	Clock input in slave mode	Input (floating input/pull-up input/pull-down input)
MOSI	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
MISO	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 USB

USB signal pin	Functional configuration	GPIO configuration
UDP	Connected to internal full-speed USB transceiver	Floating input
UDM	Connected to internal full-speed USB transceiver	Floating input

Table 7-6 CMP

CMP signal pin	Functional configuration	GPIO configuration
CMP_P0, CMP_P1	CMP analog input + terminal	Analog input
CMP_N	CMP analog input - terminal	Analog input

7.3.2 Remapping of Function Pins

In order to enable the peripheral functions and optimize the utilization rate at the same time, some function pins can be remapped to other pins by setting the function pin remapping register (R16_PIN_ALTERNATE).

Table 7-7 Alternate function remapping pins

Peripheral function pins	**Default GPIO pins**	**Remapped GPIO pins**
SCS/SCK/MOSI/MISO	PA4/PA5/PA7/PA6	PA2/PA3/-/-
RXD	PA2	PA3/PA0/PA1/PA4/PA9/PA10/PA11
TXD	PA3	PA2/PA1/PA0/PA7/PA8/PA11/PA10
TMR/PWM0	PA7	PA2/PA4/PA9
CAP_IN1	PA7	PA2/PA4/PA9
CAP_IN2	PA2	PA7/PA9/PA4
PWM1	PA7	-
PWM2	PA2	-

PWM3	PA3	-
PWM4	PA4	-
PWM5	PA8	-
SCL	PA8	PA0/PA3/PA5
SDA	PA9	PA1/PA2/PA6

7.4 Register Description

Table 7-8 GPIO registers

Name	Access address	Description	Reset value
R16_PIN_ALTERNATE	0x40001018	Function pin remapping register	0x0000
R16_PIN_ALTERNATE_H	0x4000101A	Function pin high remapping register	0x0000
R16_PA_INT_EN	0x40001090	PA port interrupt enable register	0x0000
R16_PA_INT_MODE	0x40001094	PA port interrupt mode configuration register	0x0000
R16_PA_INT_EDGE_TYPE	0x40001096	PA port interrupt edge type configuration register	0x0000
R16_PA_INT_IF	0x4000109C	PA 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 set register	0x00000000

Function Pin Remapping Register (R16_PIN_ALTERNATE)

Bit	Name	Access	Description	Reset value
15	Reserved	RO	Reserved	0
14	RB_PIN_DEBUG_EN	RW	Debugging interface control bits: 1: Enable debugging interface; 0: Disable debugging interface.	1
13	RB_PIN_USB_EN	RW	Full-speed USB pin enable: 1. Enable full-speed USB communication pin; 0: Disable full-speed USB communication pin.	0
12	RB_UDP_PU_EN	RW	UDP pin Internal pull-up resistor of full-speed USB is enabled: 1: Pull-up is forced to be enabled (RB_UC_DEV_PU_EN does not work in sleep or power-down mode, so it is replaced); 0: RB_UC_DEV_PU_EN controls whether to pull up or not.	0
[11:0]	RB_PA_DI_DIS	RW	PA0-PA11 channel pin digital input enable: 1. Turn off the digital input to save power	0

			consumption; 0: Turn on digital input.	

Function Pin High Remapping Register (R16_PIN_ALTERNATE_H)

Bit	Name	Access	Description	Reset value
[15:13]	Reserved	RO	Reserved	0
12	RB_25M_EN	RW	25MHz clock output enable: 1: PA4 output 25MHz clock; 0: No output.	0
11	RB_SPI_CLK	RO	SPI's CLK function pin mapping selection bit: 1: Remapping (SCK/PA3); 0: Default mapping (SCK/PA5).	0
[10:9]	RB_I2C_PIN	RW	I2C function pin mapping select bit 00: Default mapping (SCL/PA8, SDA/PA9); 01: Remapping (SCL/PA0, SDA/PA1); 10: Remapping (SCL/PA3, SDA/PA2); 11: Remapping (SCL/PA5, SDA/PA6).	0
8	RB_SPI_CS	RW	SPI's CS function pin mapping selection bit: 1: Remapping (SCS/PA2); 0: Default mapping (SCS/PA4).	0
[7:6]	RB_TMR_PIN	RW	Timer function pin mapping select bit: 00: Default mapping (PWM0/PA7, CAP_IN1/PA7, CAP_IN2/PA2); 01: Remapping (PWM0/PA2, CAP_IN1/PA2, CAP_IN2/PA7); 10: Remapping (PWM0/PA4, CAP_IN1/PA4 CAP_IN2/PA9); 11: Remapping (PWM0/PA9, CAP_IN1/PA9, CAP_IN2/PA4).	0
[5:3]	RB_UART_TXD	RW	TXD function pin mapping select bit of UART: 000: Default mapping (TXD/PA3); 001: Remapping (TXD/PA2); 010: Remapping (TXD/PA1); 011: Remapping (TXD/PA0); 100: Remapping (TXD/PA7); 101: Remapping (TXD/PA8); 110: Remapping (TXD/PA11); 111: Remapping (TXD/PA10).	0
[2:0]	RB_UART_RXD	RW	RXD function pin mapping select bit of UART: 000: Default mapping (RXD/PA2); 001: Remapping (RXD/PA3);	0

			010: Remapping (RXD/PA0); 011: Remapping (RXD/PA1); 100: Remapping (RXD/PA4); 101: Remapping (RXD/PA9); 110: Remapping (RXD/PA10); 111: Remapping (RXD/PA11).	

Note:

1 If the pin is used for analog functions, it is recommended to turn off the digital input function of the pin, i.e., set the digital input disabled, which can reduce power consumption and help reduce interference.

2. This register is cleared only during power-on reset and Shutdown sleep.

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.	0000h

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.	0000h

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: Do not follow polarity; 0: Follow polarity.	0000h

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.	0000h

PA Port Direction Configuration Register (R32_PA_DIR)

Bit	Name	Access	Description	Reset value
[31:12]	Reserved	RO	Reserved	0
[11:8]	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

[31:12]	Reserved	RO	Reserved	0
[11:8]	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.	Xh
[7:0]	R8_PA_PIN_0	RO		XXh

PA Port Data Output Register (R32_PA_OUT)

Bit	Name	Access	Description	Reset value
[31:12]	Reserved	RO	Reserved	0000h
[11:8]	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. 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
[7:0]	R8_PA_OUT_0	RW		0

PA Port Data Reset Register (R32_PA_CLR)

Bit	Name	Access	Description	Reset value
[31:12]	Reserved	RO	Reserved	0
[11:8]	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
[31:12]	Reserved	RO	Reserved	0000h
[11:8]	R8_PA_PU_1	RW	PA pin pull-up resistor enable control: 1: Enable the pull-up resistor; 0: Disable the pull-up resistor.	00h
[7:0]	R8_PA_PU_0	RW		00h

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

Bit	Name	Access	Description	Reset value
[31:12]	Reserved	RO	Reserved	0
[11:8]	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
[7:0]	R8_PA_PD_DRV_0	RW		0

			0: Disable the pull-down resistor. When the corresponding bit of direction register R32_PA_DIR is 1: PA pin current drive capability select: 1: 20mA level; 0: 5mA level.	

PA Port Output Set Register (R32_PA_SET)

Bit	Name	Access	Description	Reset value
[31:12]	Reserved	RO	Reserved	0
[11:8]	R8_PA_SET_1	WZ	When the corresponding bit of the setting register R32_PA_SET is 0, the output of the PA pin remains; When it is 1, the PA pin outputs a high level.	0
[7:0]	R8_PA_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 7-9 Port configuration

Mode	R32_PA_DIR	R32_PA_PU	RB_PA_DI_DIS	R32_PA_PD_DRV
Floating input/high impedance input/analog input	0	0	0	0
Analog inputs only (reduces 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 (TMR)

8.1 Introduction to TMR

The chip provides a 26-bit timer with the longest timing time of 2^26 clock cycles. It is suitable for many occasions, including measuring the pulse length of input signal (input capture) or generating output waveform (PWM); It also supports the encoder interface mode and provides two channels to capture the rotary encoder signal for encoding. Support DMA function.

8.1.1 Main Features

- 26-bit timers, and the longest timing interval is 2^26 clock cycles.
- Support timer interrupt, DMA and interrupt.
- Support capture function, provide 1 channel capture/sampling, and measure the input pulse length or period.
- Support ENCMODE. 2-channel capture two rotary encoder signals from I/O port for encoding. The high and low levels of the other signal can be compared at the edge of one signal to determine the encoding direction. The clock source is LSI.
- 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 TMR registers

Name	Access address	Description	Reset value
R8_TMR_CTRL_MOD	0x40002400	Mode set register	0x02
R8_TMR_CTRL_DMA	0x40002401	DMA control register	0x00
R8_TMR_INTER_EN	0x40002402	Interrupt enable register	0x00
R8_TMR_INT_FLAG	0x40002406	Interrupt flag register	0x00
R8_TMR_FIFO_COUNT	0x40002407	FIFO count register	0x0X
R32_TMR_COUNT	0x40002408	Current count value register	0x0XXXXXXX
R32_TMR_CNT_END	0x4000240C	Final count value set register	0x0XXXXXXX
R32_TMR_FIFO	0x40002410	FIFO register	0x0XXXXXXX
R16_TMR_DMA_NOW	0x40002414	DMA count buffer address	0x0000XXXX
R16_TMR_DMA_BEG	0x40002418	DMA begin buffer address	0x0000XXXX
R16_TMR_DMA_END	0x4000241C	DMA end buffer address	0x0000XXXX
R8_ENC_REG_CTRL	0x40002420	Encoder mode control register	0x00
R8_ENC_INTER_EN	0x40002421	Encoder mode interrupt enable register	0x00
R8_ENC_INT_FLAG	0x40002422	Encoder mode interrupt flag register	0x00
R32_ENC_REG_CEND	0x40002424	Encoder mode final value configuration register	0x00000000
R32_ENC_REG_CCNT	0x40002428	Encoder mode current value configuration register	0x00000000

Mode Set Register (R8_TMR_CTRL_MOD)

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.	00b
[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.	00b
5	Reserved	RO	Reserved	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: 1: Default at high level, active low; 0: Default at low level, active high;	0
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_TMR_INTER_EN)

Bit	Name	Access	Description	Reset value
[7:5]	Reserved	RO	Reserved	000b
4	RB_TMR_IE_FIFO_OV	RW	FIFO overflow (FIFO is full in capture mode or FIFO is empty in PWM mode) interrupt enable:	0

			1: Enable interrupt; 0: Disable interrupt.	
3	RB_TMR_IE_DMA_END	RW	DMA end interrupt enable: 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_TMR_INT_FLAG)

Bit	Name	Access	Description	Reset value
[7:5]	Reserved	RO	Reserved	000b
4	RB_TMR_IF_FIFO_OV	RW1	FIFO overflow (FIFO is full in capture mode or FIFO is empty in PWM mode) flag. Write 1 to reset: 1: Has overflowed; 0: Not overflowed.	0
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.	0

			Write 1 to reset: 1: Timeout/end of cycle; 0: No timeout/ not end.	

FIFO Count Register (R8_TMR_FIFO_COUNT)

Bit	Name	Access	Description	Reset value
[7:4]	Reserved	RO	Reserved	0h
[3:0]	R8_TMR_FIFO_COUNT	RO	Data count in FIFO, the maximum value is 8.	0Xh

Current Count Value Register (R32_TMR_COUNT)

Bit	Name	Access	Description	Reset value
[31:26]	Reserved	RO	Reserved	0Xh
[25:0]	R32_TMR_COUNT	RO	Current count value of counter.	XXXXXXXh

Final Count Value Set Register (R32_TMR_CNT_END)

Bit	Name	Access	Description	Reset value
[31:0]	R32_TMR_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; 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). *Note: With write operation on this register, the value of R32_TMR_COUNT will be automatically cleared to 0.*	0XXXXXXXh

FIFO Register (R32_TMR_FIFO)

Bit	Name	Access	Description	Reset value
[31:0]	R32_TMR_FIFO	RO/WO	FIFO data register, only the lower 26 bits are valid.	0XXXXXXXh

DMA Control Register (R8_TMR_CTRL_DMA)

Bit	Name	Access	Description	Reset value
[7:3]	Reserved	RO	Reserved	00000b
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,	0

			when the DMA address is added to the set end address, it will automatically loop to the start address set.	
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_TMR_DMA_NOW)

Bit	Name	Access	Description	Reset value
15	Reserved	RO	Reserved	0
[14:2]	R16_TMR_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_DMA_BEG)/4.	XXXXh
[1:0]	Reserved	RO	Reserved	00b

DMA Begin Buffer Address (R16_TMR_DMA_BEG)

Bit	Name	Access	Description	Reset value
15	Reserved	RO	Reserved	0
[14:2]	R16_TMR_DMA_BEG	RW	DMA data buffer begin address, the address must be 4-byte aligned.	XXXXh
[1:0]	Reserved	RO	Reserved	00b

DMA End Buffer Address (R16_TMR_DMA_END)

Bit	Name	Access	Description	Reset value
15	Reserved	RO	Reserved	0
[14:2]	R16_TMR_DMA_END	RW	DMA data buffer end address (not included), address must be 4-byte aligned.	XXXXh
[1:0]	Reserved	RO	Reserved	00b

Encoder Mode Control Register (R8_ENC_REG_CTRL)

Bit	Name	Access	Description	Reset value
[7:6]	Reserved	RO	Reserved	0
5	RB_ENC_DIR	RO	Encoder current direction: 0: Forward; 1: Backward.	0
4	Reserved	RO	Reserved	0
3	RB_RD_CLR_EN	RW	Encoder mode reads count and clears.	0
[2:1]	RB_SMS_MODE	RW	Encoder mode edge operation mode: 00 = IDLE;	0

			10 = Counting on T1 edge; 01 = Counting on T2 edge; 11 = Counting on T1 AND T2 edge.	
0	RB_START_ENC_EN	RW	Encoder mode operation enables. 1: Enable (must open LSI); 0:Disable.	0

Encoder Mode 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 mode back-interrupt enable. 1: Enable interrupt; 0: Disable interrupt.	0
0	RB_IE_DIR_INC	RW	Encoder mode advance interrupt enable. 1: Enable interrupt; 0: Disable interrupt.	0

Encoder Mode 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 mode backward interrupt flag, write 1 to clear 0. 1: Encoder mode backward; 0: Encoder mode remains unchanged.	0
0	RB_IF_DIR_INC	RW	Encoder mode advance interrupt flag, write 1 to clear 0. 1: Encoder mode advance; 0: Encoder mode remains unchanged.	0

Encoder Mode Final Value Configuration Register (R32_ENC_REG_CEND)

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

Encoder Mode Current Value Configuration Register (R32_ENC_REG_CEND)

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

8.3 Functional Description and Configuration

8.3.1 Timing and Counting Functions

Timer 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: 31.25nS*2^26 ≈ 2S. Timer also support independent interrupt.

The operation steps for timing function are as follows:

(1) Set RB_TMR_ALL_CLEAR, clear R32_TMR_COUNT and interrupt flag, etc.
(2) Set the R32_TMR_CNT_END register to the time value that needs timing;
Time = Tsys*R32_TMR_CNT_END;
(3) Clear RB_TMR_ALL_CLEAR, clear the timing mode corresponding to RB_TMR_MODE_IN;
(4) Optional steps, set R8_TMR_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_TMR_CTRL_MOD register, and start the timer counting;
(6) When count value of R32_TMR_COUNT is equal to that of R32_TMR_CNT_END, the timing is completed. In this case, RB_TMR_IF_CYC_END in R8_TMR_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_TMR_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_TMR_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_TMR_COUNT. Every time the count value reaches final count value, RB_TMR_IE_CYC_END will be set to 1 and R32_TMR_COUNT will be cleared to 0. Hardware interrupt is triggered if enabling interrupt.

8.3.2 PWM Function

Timer TMR has PWM function, which supports DMA data loading. PWM can set the default output polarity to high level or low level, and the number of repetitions of the same data can be selected as 1, 4, 8 or 16 times. This repetition function combined with DMA can be used to imitate the effect of DAC. The shortest time unit of PWM output effective level is 1 system clock, which can dynamically modify the duty cycle of PWM and imitate a special waveform.

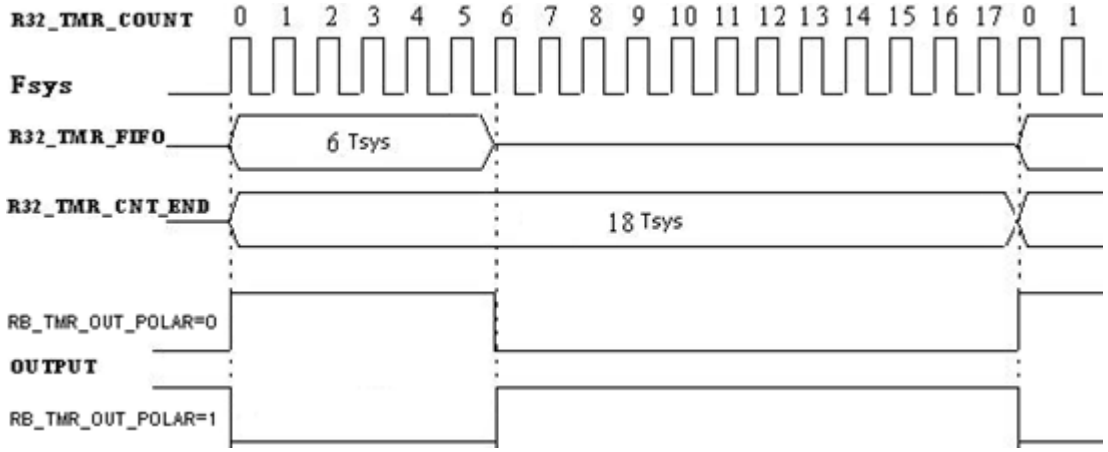
PWM operation steps are as follows:

(1) Set RB_TMR_ALL_CLEAR, empty and clear R32_TMR_FIFO and interrupt flags, etc.
(2) Set the PWM total cycle register R32_TMR_CNT_END, the value shall not be less than the value in R32_TMR_FIFO register;
(3) Configure R8_TMR_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_TMR_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_TMR_FIFO/R32_TMR_CNT_END. Can load continuous dynamic data through DMA, and simulate special waveforms combined with the repeated output times of the same data;
(5) Configure R8_TMR_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_TMR_FIFO. It is recommended to load it through DMA.

For example: Set the RB_TMR_OUT_POLAR bit to 0, R32_TMR_FIFO to 6, R32_TMR_CNT_END to 18, the basic timing diagram of PWM generation is as follows, and its duty cycle is: R32_TMR_FIFO/R32_TMR_CNT_END = 1/3.

Figure 8-1 PWM output timing diagram

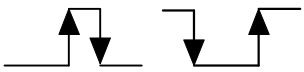
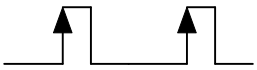


If RB_TMR_PWM_REPEAT is set to 00, it means that the above process is repeated once, 01 means repeating 4 times, 10 means that repeating 8 times, and 11 means repeating 16 times. After repeating, load the next data in FIFO and then continue.

8.3.3 Capture Function

Timer TMR has capture function, which supports DMA data storage. The capture mode can be selected from any edge trigger start to any edge trigger end, rising edge trigger start to rising edge trigger end and falling edge trigger start to falling edge trigger end. The following is the description table of capture trigger mode:

Table 8-5 Description of capture trigger mode

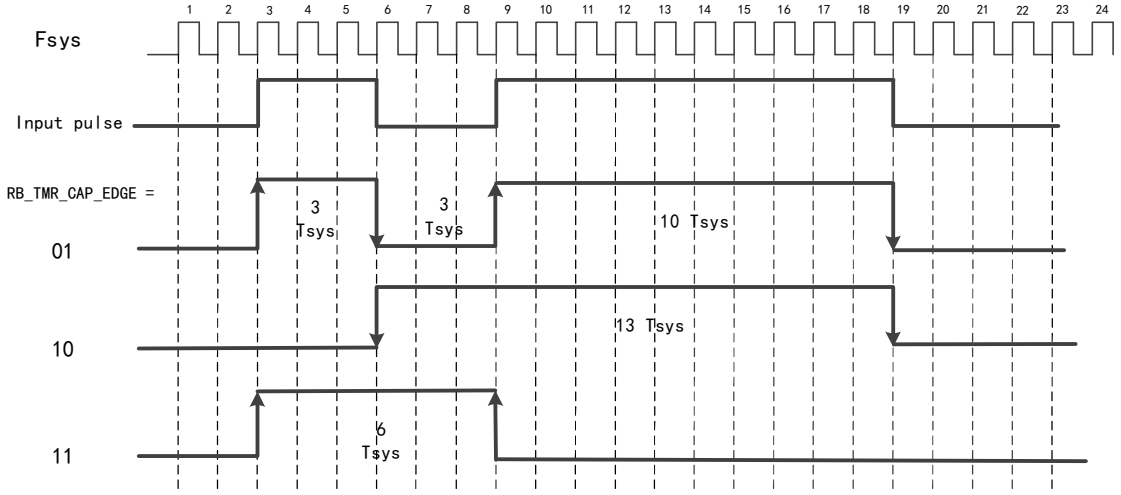
Capture mode selection bit RB_TMR_CATCH_EDGE	Trigger Mode	Icon
00	Not capture	None
01	Edge trigger edge to edge	
10	Falling edge to falling edge	
11	Rising edge to rising edge	

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_TMR_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_TMR_FIFO is 0, one cycle is normally sampled; when it is 1, the input change period exceeds the timeout value R32_TMR_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_TMR_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_TMR_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_TMR_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_TMR_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 R32_TMR_DMA_BEG as the first address of buffer which stores the data captured, set the register R32_TMR_DMA_END as the end address of buffer which stores the data captured (not included), and set the RB_TMR_DMA_ENABLE of R8_TMR_CTRL_DMA as 1, and enable DMA function;
(7) Clear RB_TMR_ALL_CLEAR of R8_TMR_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_TMR_FIFO by default; if DMA is enabled, the captured data will be automatically stored in the data buffer set by DMA.

8.3.4 Encoder Function

Timer TMR module supports encoder mode. Two input channels CAP_IN1 and CAP_IN2 are provided to capture 2 rotary encoder signals for encoding. The high and low levels of one captured signal can be compared at the edge of the other captured signal to judge the coding direction.

Select the encoder mode configuration: 2 input channels CAP_IN1 and CAP_IN2 are used as the interfaces of the rotary encoder. If the counter only counts at the edge of CAP_IN2, set Rb _ SMS _ mode = 01 in the R8_ENC_REG_CTRL register; Set Rb _ SMS _ mode = 10 if counting only at the edge of CAP_IN1; If the counter counts at the edges of CAP_IN1 and CAP_IN2 at the same time, set RB_SMS_MODE = 11.

According to the jump order of the 2 input signals, the counter counts up or down, and the hardware sets the DIR bit in the R8_ENC_REG_CTRL register accordingly. Whether the counter depends on CAP_IN1, CAP_IN2 or both CAP_IN1 and CAP_IN2. The RB_ENC_DIR bit will be recalculated when any input (CAP_IN1 or CAP_IN2) jumps.

R32_ENC_REG_CEND must be configured before starting counting.

Table 8-6 Relationship between technical direction and coded signals

Valid edge	Relative signal level	Signal corresponding to channel CAP_IN1		Signal corresponding to channel CAP_IN2	
		Rising	Falling	Rising	Falling
Count only in CAP_IN1	High	Count down	Count up	Not count	Not count
	Low	Count up	Count down	Not count	Not count
Count only in CAP_IN2	High	Not count	Not count	Count up	Count down
	Low	Not count	Not 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

Capture the rotary encoder signal for encoding operation steps:
(1) Set RB_SMS_MODE to configure the encoding edge working mode, if you need to sleep and wake up, you need to turn on the wake-up enable;
(2) Configure R32_ENC_REG_CEND, the value is not 0 before working;
(3) RB_START_ENC_EN set to 1 to start working;
(4) RB_RD_CLR_EN set to 1, then automatically clear the current encoding value after each reading of R32_ENC_REG_CCNT; automatically clear the wake-up signal after each system wake-up.

8.3.5 DMA Function

Timer TMR has DMA function. When using DMA to complete interrupt, you need to pay attention to the configuration order of related registers.

The steps to enable DMA completion interrupt are as follows:
(1) Read R16_TMR_DMA_NOW, and assign R16_TMR_DMA_END to any value that is not equal to

R16_TMR_DMA_NOW (For example, the value can be R16_TMR_DMA_Now+0x100);

(2) Write 1 to clear bit RB_TMR_IF_DMA_END in R8_TMR_INT_FLAG;

(3) Set bit RB_TMR_IE_DMA_END in R8_TMR_INTER_EN to 1.

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

(1) Read R16_TMR_DMA_NOW, and assign R16_TMR_DMA_END to any value that is not equal to R32_TMR1_DMA_NOW (Example: it can take the value of R16_TMR_DMA_NOW + 0x100);

(2) Write 1 to clear bit RB_TMR_IF_DMA_END of R8_TMR_INT_FLAG.

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

(1) Write 1 to clear bit RB_TMR_IF_DMA_END of R8_TMR_INT_FLAG.

Chapter 9 Universal Asynchronous Receiver-Transmitter (UART)

9.1 Introduction to UART

CH572 and CH570 provide 1 set of full-duplex UART. Full-duplex serial communication are supported.

9.1.1 Main Features

- Compatible with 16C550 asynchronous serial port and enhanced.
- 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 12.5Mbps.
- Built-in 8-byte FIFO buffer, support 4 FIFO trigger stages.
- Support serial frame error detection and Break circuit interval detection.
- Support full-duplex serial communication.

9.2 Register Description

Table 9-1 UART registers

Name	Access address	Description	Reset value
R8_UART_MCR	0x40003400	MODEM control register	0x00
R8_UART_IER	0x40003401	Interrupt enable register	0x00
R8_UART_FCR	0x40003402	FIFO control register	0x00
R8_UART_LCR	0x40003403	Line control register	0x00
R8_UART_IIR	0x40003404	Interrupt identification register	0x01
R8_UART_LSR	0x40003405	Line status register	0x60
R8_UART_RBR	0x40003408	Receive buffer register	0xXX
R8_UART_THR	0x40003408	Transmit hold register	0xXX
R8_UART_RFC	0x4000340A	Receive FIFO count register	0x00
R8_UART_TFC	0x4000340B	Transmit FIFO count register	0x00
R16_UART_DL	0x4000340C	Baud rate divisor latch	0xXXXX
R8_UART_DIV	0x4000340E	Prescaler divisor register	0xXX

MODEM Control Register (R8_UART_MCR)

Bit	Name	Access	Description	Reset value
[7:4]	Reserved	RO	Reserved	0
3	RB_MCR_OUT2 RB_MCR_INT_OE	RW	UART interrupt request output control: 1: Enable to send request; 0: Disable.	0
[2:0]	Reserved	RO	Reserved	0

Interrupt Enable Register (R8_UART_IER)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved	0

6	RB_IER_TXD_EN	RW	UART TXD pin output enable bit: 1: Enable pin output; 0: Disable pin output.	0
[5:3]	Reserved	RO	Reserved	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_UART_FCR)

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 data is generated when 4 bytes are received, and RTS pin is automatically invalidated when hardware flow control is enabled.	0
[5:1]	Reserved	RO	Reserved	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_UART_LCR)

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:	0

			1: Forced to generate; 0: Not generate.	
[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.	00b
3	RB_LCR_PAR_EN	RW	Parity bit enable: 1: Allow to generate parity bit when transmitting 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; 10: 7 data bits; 11: 8 data bits.	00b

Interrupt Identification Register (R8_UART_IIR)

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.	00b
[5:4]	Reserved	RO	Reserved	00b
[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.	000b
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_UART_IIR and each bit of RB_IIR_INT_MASK is shown in the following table:

Table 9-5 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	Reserved	Reserved	Reserved
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	Reserved	Reserved	Reserved

Line Status Register (R8_UART_LSR)

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: No error in receiver FIFO.	0
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

Receive Buffer Register (R8_UART_RBR)

Bit	Name	Access	Description	Reset value
[7:0]	R8_UART_RBR	RO	Data receive buffer register. If the DATA_RDY bit of LSR is 1, the received data can be read from this register;	XXh

			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.	

Transmit Hold Register (R8_UART_THR)

Bit	Name	Access	Description	Reset value
[7:0]	R8_UART_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_UART_RFC)

Bit	Name	Access	Description	Reset value
[7:0]	R8_UART_RFC	RO	Data count in the current receiver FIFO.	00h

Transmit FIFO Count Register (R8_UART_TFC)

Bit	Name	Access	Description	Reset value
[7:0]	R8_UART_TFC	RO	Data count in the current transmitter FIFO.	00h

Baud Rate Divisor Latch (R16_UART_DL)

Bit	Name	Access	Description	Reset value
[15:0]	R16_UART_DL	RW	The 16-bit divisor is used to calculate the baud rate. 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.	XXXXh

Prescaler Divisor Register (R8_UART_DIV)

Bit	Name	Access	Description	Reset value
[7:0]	R8_UART_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

9.3 Functional Description and Configuration

UART output pins are all at 3.3V LVCMOS level. The pins in UART mode include: data transmission pins. Data transmission pins include: TXD pin and RXD pin, both of which are at high level by default.

UARTs have built-in independent transceiver buffers and 8-byte FIFOs, support 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_UART_DIV register, the maximum value is 127, and usually 1 is written.
2) Calculate the baud rate and set R16_UART_DL register.
   Baud rate =Fsys * 2 / R8_UART_DIV / 16 / R16_UART_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 transmit 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_UART_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_UART_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_UART_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_UART_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 CH572 and CH570 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: mode0 and mode3.
- 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 SPI registers

Name	Access address	Description	Reset value
R8_SPI_CTRL_MOD	0x40004000	SPI0 mode control register	0x02
R8_SPI_CTRL_CFG	0x40004001	SPI0 configuration register	0x00
R8_SPI_INTER_EN	0x40004002	SPI0 interrupt enable register	0x00
R8_SPI_CLOCK_DIV R8_SPI_SLAVE_PRE	0x40004003	SPI0 clock divider register in master mode SPI0 preset data register in slave mode	0x10
R8_SPI_BUFFER	0x40004004	SPI0 data buffer	0xXX
R8_SPI_RUN_FLAG	0x40004005	SPI0 working status register	0x00
R8_SPI_INT_FLAG	0x40004006	SPI0 interrupt flag register	0x40
R8_SPI_FIFO_COUNT	0x40004007	SPI0 transceiver FIFO count register	0x00
R8_SPI_INT_TYPE	0x40004008	SPI interrupt trigger mode selection register	0x00
R8_SPI_INTER1_EN	0x40004009	SPI interrupt 1 enable register	0x00
R8_SPI_INT1_FLAG	0x4000400A	SPI interrupt 1 flag register	0x00
R16_SPI_TOTAL_CNT	0x4000400C	SPI0 transceiver data total length register	0x0000
R8_SPI_FIFO	0x40004010	SPI0 data FIFO register	0xXX
R8_SPI_FIFO_COUNT1	0x40004013	SPI0 transceiver FIFO count register	0x00
R16_SPI_DMA_NOW	0x40004014	SPI0 DMA buffer current address	0xXXXX
R16_SPI_DMA_BEG	0x40004018	SPI0 DMA buffer start address	0xXXXX

R16_SPI_DMA_END	0x4000401C	SPI0 DMA buffer end address	0xXXXX

SPI Mode Control Register (R8_SPI_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 (receive data); 0: Output (transmit data).	0
3	RB_SPI_SLV_CMD_MOD	RW	First byte mode selection in SPI 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_MOD	RW	2-wire or 3-wire SPI mode selection: 1: 2-wire mode/halfduplex (MISO in slave mode /MOSI in master mode); 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	SPI master/slave mode selection: 1: Slave mode; 0: Master mode.	0

SPI Configuration Register (R8_SPI_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	RB_SPI_MODE_SLAVE = 0 for main clock polarity selection: 1: polarity reversed; 0: polarity unchanged. RB_SPI_MODE_SLAVE = 1 is the two-wire mode slave input and output direction selection: 1: Slave input; 0: slave output.	0
0	RB_SPI_DMA_ENABLE	RW	DMA enable (only supported by SPI0): 1: Enable; 0: Disable.	0

SPI Interrupt Enable Register (R8_SPI_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	00b
4	RB_SPI_IE_FIFO_OV	RW	FIFO overflow (FIFO is full when receiving, or FIFO is empty when transmitting) interrupt enable: 1: Interrupt enabled; 0: Interrupt disabled.	0
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:	0

			1: Interrupt enabled; 0: Interrupt disabled.	
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_SPI_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, up to 254. Fsck= Fsys/frequency division factor.	10h

SPI Preset Data Register in Slave Mode (R8_SPI_SLAVE_PRE)

Bit	Name	Access	Description	Reset value
[7:0]	R8_SPI_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_SPI_BUFFER)

Bit	Name	Access	Description	Reset value
[7:0]	R8_SPI_BUFFER	RW	SPI data transmit and receive buffer.	XXh

SPI Working Status Register (R8_SPI_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_SPI_SLAVE_PRE; 0: Not yet loaded or has completed.	0
5	RB_SPI_FIFO_READY	RO	FIFO ready: 1: FIFO is ready (R16_SPI_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	0

			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.	
[3:0]	Reserved	RO	Reserved	0000b

SPI Interrupt Flag Register (R8_SPI_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	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>=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; 0: All byte transfer not end.	0

SPI Transceiver FIFO Count Register (R8_SPIx_FIFO_COUNT)

Bit	Name	Access	Description	Reset value
[7:4]	Reserved	RO	Reserved	0h
[3:0]	R8_SPI_FIFO_COUNT	RW	Current byte count in FIFO.	0h

SPI Transceiver Data Total Length Register (R16_SPI_TOTAL_CNT)

Bit	Name	Access	Description	Reset value
[15:0]	R16_SPI_TOTAL_CNT	RW	Total number of bytes of SPI data transceiver in master mode, and the lower 12 bits are valid. At most 4095 bytes can be received/transmitted at a time when using DMA. It is not supported in slave mode.	0

SPI Data FIFO Register (R8_SPI_FIFO)

Bit	Name	Access	Description	Reset value
[7:0]	R8_SPI_FIFO	RO/ WO	Data FIFO register.	XXh

The registers R8_SPI_BUFFER and R8_SPI_FIFO are both SPI data related registers, and the main differences between them are:
Reading R8_SPI_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_SPI_BUFFER in master mode means to send the byte directly, and the write operation in slave mode is not defined;

Reading R8_SPI_FIFO means to obtain the data from the earliest exchange in FIFO, which will reduce FIFO and R8_SPI_FIFO_COUNT;

Writing to R8_SPI_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_SPI_TOTAL_CNT is not 0.

SPI Transceiver FIFO Count Register (R8_SPI_FIFO_COUNT1)

Bit	Name	Access	Description	Reset value
[7:4]	Reserved	RO	Reserved	0h
[3:0]	R8_SPI_FIFO_COUNT1	RW	Current byte count in FIFO. The same as R8_SPI_FIFO_COUNT.	0h

SPI Interrupt Trigger Mode Selection Register (R8_SPI_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= pulse trigger, 0= edge 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_SPI_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: Turn on enable; 0: Turn off enable.	0
0	RB_SPI_IE_FIFO_EMPTY	RW	Current FIFO data empty interrupt enable: 1: Turn on enable; 0: Turn off enable.	0

SPI Interrupt 1 Flag Register (R8_SPI_INT1_FLAG)

Bit	Name	Access	Description	Reset value
[7:2]	Reserved	RO	Reserved	0
0	RB_SPI_IF_FIFO_FULL	RW1	Current FIFO data full flag bit, write 1 to clear: 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 not empty.	0

SPI0 DMA Buffer Current Address (R16_SPI_DMA_NOW)

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
[15:0]	R16_SPI_DMA_NOW	RW	DMA data buffer current address. It can be used to calculate the number of conversions. COUNT=SPI_DMA_NOW-SPI_DMA_BEG.	XXXXh

SPI DMA Buffer Start Address (R16_SPI_DMA_BEG)

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
[16:0]	R16_SPI_DMA_BEG	RW	DMA data buffer start address, only the lower 14 bits are valid.	XXXXh

SPI0 DMA Buffer End Address (R16_SPI_DMA_END)

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
[16:0]	R16_SPI_DMA_END	RW	DMA data buffer end address (not included) ,	XXXXh

			only the lower 14 bits are valid.	

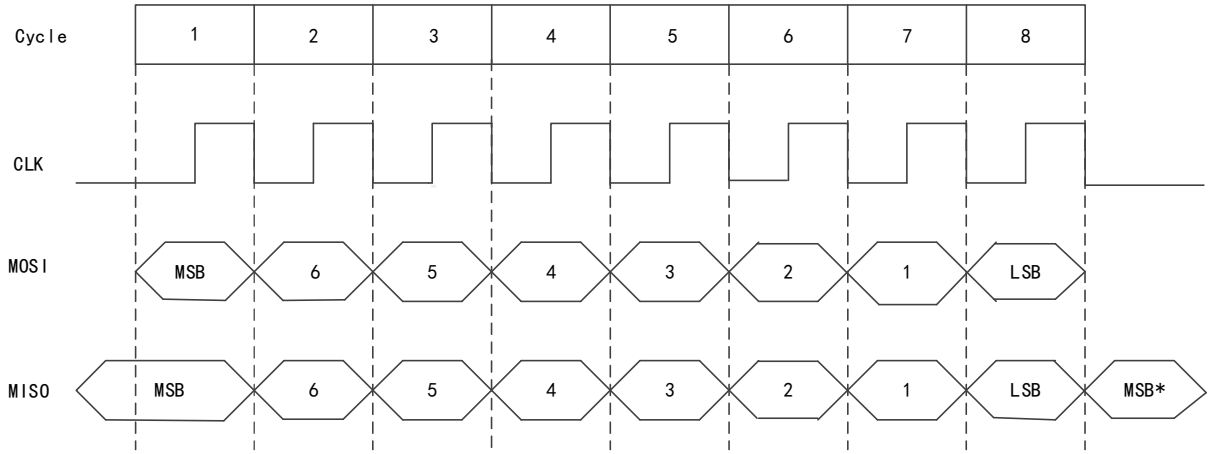
10.3 SPI Transfer Frame Formats

SPI supports 2 transfer frame formats, mode0 and mode3, which can be selected by setting RB_SPI_MST_SCK_MOD in R8_SPI_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:

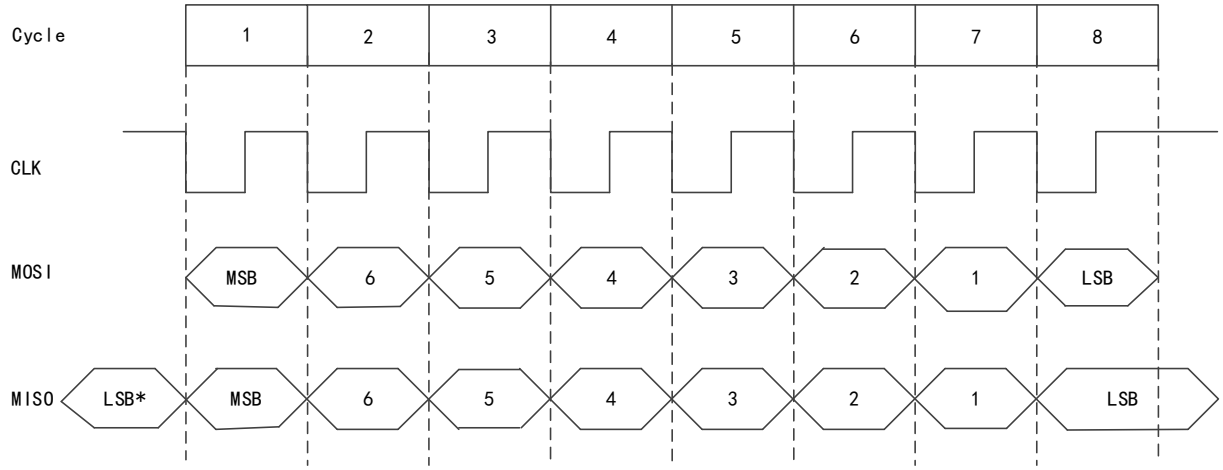
Mode0: RB_SPI_MST_SCK_MOD = 0

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_SPI_CLOCK_DIV, to configure SPI clock frequency;

(2) Set RB_SPI_MODE_SLAVE in R8_SPI_CTRL_MOD to 0, to configure SPI to master mode;

(3) Set RB_SPI_MST_SCK_MOD in R8_SPI_CTRL_MOD, to select clock idle mode0 or mode3;

(4) Set the RB_SPI_FIFO_DIR in R8_SPI_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_SPI_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) In 2-wire mode, SCK is unchanged, RB_SPI_MOSI_OE = 0, and MISO is not needed. In 3-wire mode, input (Rb_SPI_MOSI_OE = 0 and pin set as input) and output (RB_SPI_MOSI_OE = 1 and pin set as output) are realized by MOSI half-duplex, and the direction is manually switched.

(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_SPI_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_SPI_TOTAL_CNT register, and set the length of the data to be sent;

(3) Write to the R8_SPI_FIFO register and write the data to be transmitted to FIFO. If R8_SPI_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_SPI_TOTAL_CNT is not 0 and there is data in FIFO, SPI master will automatically transmit data, otherwise, it will pause;

(5) Wait until R16_SPI_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_SPI_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_SPI_TOTAL_CNT register, to set the length of the data to be received;

(3) As long as R16_SPI_TOTAL_CNT is not 0 and FIFO is not full, SPI master will automatically receive data, otherwise, it will pause;

(4) Wait until R8_SPI_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

SPI 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_SPI_CTRL_MOD to 1, to configure SPI to slave mode;

(2) Set RB_SPI_SLV_CMD_MOD in R8_SPI_CTRL_MOD as needed, to select the slave first byte mode or data stream mode;

(3) Set RB_SPI_FIFO_DIR in R8_SPI_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) In 2-wire mode, SCK is unchanged, RB_SPI_MISO_OE=0, and MOSI is not needed. In 3-wire mode, input (Rb _ SPI _ MISO _ OE = 0 and pin set as input) and output (RB_SPI_MISO_OE=1 and pin set as output) are realized by miso half-duplex, and the direction is manually switched.

(6) Optional, set the preset data register (R8_SPI_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_SPI_SLAVE_PRE through exchange. The bit7 of R8_SPI_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_SPI_SLAVE_PRE is preset, the external SPI host will obtain the preset value of bit7 of R8_SPI_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_SPI_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);

(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.

Data transmission:

(1) Set RB_SPI_FIFO_DIR in R8_SPI_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_SPI_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_SPI_FIFO_COUNT, if it is not full, continue to write data to be sent to FIFO.

Data reception:

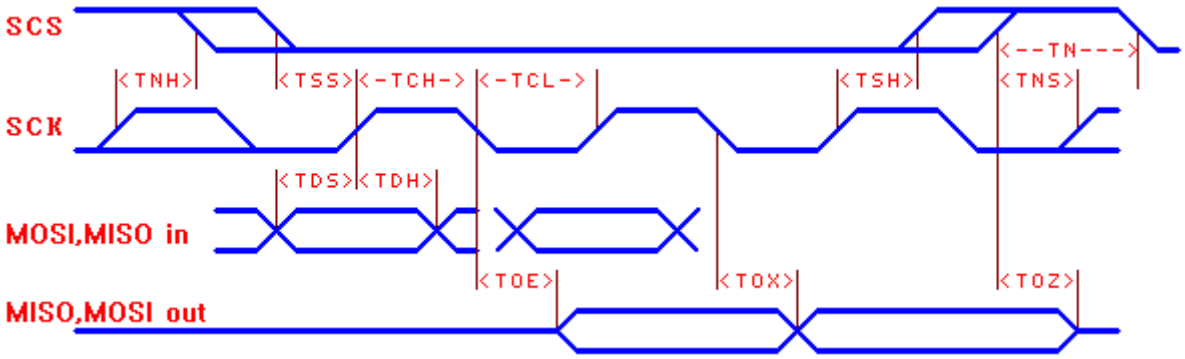
(1) Set RB_SPI_FIFO_DIR of R8_SPI_CTRL_MOD to 1, and the current FIFO direction as input;

(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_SPI_FIFO_COUNT, if it is not empty, the data has been received and the data will be taken away by reading R8_SPI_FIFO. If DMA is enabled, DMA will automatically read FIFO to complete this step;

(4) For reception of the single byte data, R8_SPI_BUFFER can be read directly without using FIFO.

10.5 SPI Timing



Name	Parameter description (TA=25°C, VIO33=3.3V)	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	Tsys*0.55			nS
TCL	Time of SCK clock at low level	Tsys*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: Tsys is the cycle of system clock frequency (1/Fsys).

Chapter 11 PWM

11.1 Introduction to PWM

In addition to the1-channel 26-bit PWM outputs provided by the timer, the system also provides 5-channel 16-bit or 8-bit PWM outputs (PWM1～PWM5) with adjustable duty cycle and a fixed selectable PWM period of 8 cycles for simple 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 register	0x00
R8_PWM_DMA_CTRL	0x40005003	PWMx DMA control register	0x00
R32_PWM1_3_DATA	0x40005004	PWM data hold register 1	0xXXXXXXXX
R16_PWM1_DATA	0x40005004	PWM1 data hold register (16-bit width)	0xXXXX
R8_PWM1_DATA	0x40005004	PWM1 data hold register (8-bit width)	0xXX
R8_PWM2_DATA	0x40005005	PWM2 data hold register (8-bit width)	0xXX
R16_PWM2_DATA	0x40005006	PWM2 data hold register (16-bit width)	0xXXXX
R8_PWM3_DATA	0x40005006	PWM3 data hold register (8-bit width)	0xXX
R16_PWM3_DATA	0x40005008	PWM data hold register 2	0xXXXX
R32_PWM4_5_DATA	0x40005010	PWM data hold register 3	0xXXXXXXXX
R16_PWM4_DATA	0x40005010	PWM4 data hold register (16-bit width)	0xXXXX
R8_PWM4_DATA	0x40005010	PWM4 data hold register (8-bit width)	0xXX
R8_PWM5_DATA	0x40005011	PWM5 data hold register (8-bit width)	0xXX
R16_PWM5_DATA	0x40005012	PWM5 data hold register (16-bit width)	0xXXXX
R8_PWM_INT_EN	0x4000500C	PWMx interrupt enable register	0x00
R8_PWM_INT_FLAG	0x4000500D	PWMx interrupt flag register	0x00
R16_PWM_CYC_VALUE	0x40005014	PWM1, 2, 3 cycle end register	0xXXXX
R16_PWM_CYC1_VALUE	0x40005015	PWM4, 5 cycle end register	0xXXXX
R16_PWM_CLOCK_DIV	0x40005018	PWMx clock divider register	0x0000
R32_PWM_DMA_NOW	0x4000501C	PWMxDMA buffer current address register	0xXXXXXXXX
R32_PWM_DMA_BEG	0x40005020	PWMxDMA buffer begin address Register	0xXXXXXXXX
R32_PWM_DMA_END	0x40005024	PWMxDMA buffer end address Register	0xXXXXXXXX

PWMx Output Enable Register (R8_PWM_OUT_EN)

Bit	Name	Access	Description	Reset value

[7:5]	Reserved	RO	Reserved	0
4	RB_PWM5_OUT_EN	RW	PWM5 output enable: 1: Enabled; 0: Disabled.	0
3	RB_PWM4_OUT_EN	RW	PWM4 output enable: 1: Enabled; 0: Disabled.	0
2	RB_PWM3_OUT_EN	RW	PWM3 output enable: 1: Enabled; 0: Disabled.	0
1	RB_PWM2_OUT_EN	RW	PWM2 output enable: 1: Enabled; 0: Disabled.	0
0	RB_PWM1_OUT_EN	RW	PWM1 output enable: 1: Enabled; 0: Disabled.	0

PWMx Output Polarity Configuration Register (R8_PWM_POLAR)

Bit	Name	Access	Description	Reset value
[7:5]	Reserved	RO	Reserved	0
4	RB_PWM5_POLAR	RW	PWM5 output polarity control: 1: Default at high level, active low; 0: Default at low level, active high.	0
3	RB_PWM4_POLAR	RW	PWM4 output polarity control: 1: Default at high level, active low; 0: Default at low level, active high.	0
2	RB_PWM3_POLAR	RW	PWM3 output polarity control: 1: Default at high level, active low; 0: Default at low level, active high.	0
1	RB_PWM2_POLAR	RW	PWM2 output polarity control: 1: Default at high level, active low; 0: Default at low level, active high.	0
0	RB_PWM1_POLAR	RW	PWM1 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	RB_PWM_SYNC_EN	RW	Synchronization enable allows: 1: Synchronization is allowed; 0: Synchronization is prohibited.	0
6	RB_PWM_SYNC_START	RW	Synchronization start: 1:Start synchronization; 0: No operation is performed.	0

5	Reserved	RO	Reserved	0
4	RB_PWM4_5_CH	R0	PWM4/5 channel output status: 1: 4 channel output; 0:5 channel output.	0
3	RB_PWM4_5_STAG_EN	RW	PWM4/5 interleaved output enable bits: 1: Interleaved output; 0: Independent output.	0
[2:1]	RB_PWM_CYC_MOD	RW	PWM data width selection: 00: 8-bit data width; 01: 7-bit data width; 10: 6-bit data width; 11: 16-bit data width.	0
0	RB_PWM_CYCLE_SEL	RW	PWM cycle selection: 1: 8/7/6-bit data width corresponds to 255/127/63 clock cycles; 16-bit data width corresponds to clock cycles controlled by R16_PWM_REG_CYCLE. 0: 8/7/6-bit data width corresponds to 256/128/64 clock cycles; 16-bit data width corresponding to clock cycles controlled by R16_PWM_REG_CYCLE.	0

PWMx DMA Control Register (R8_PWM_DMA_CTRL)

Bit	Name	Access	Description	Reset value
[7:3]	Reserved	RO	Reserved	0
2	RB_DMA_SEL	RW	DMA output channel selection when RB_PWM_SYNC_EN = 0: 1: DMA selects 1, 2, 3 channel outputs 0: DMA selects 4, 5 channel outputs *Note: When RB_PWM_SYNC_EN = 1, RB_DMA_SEL = X, DMA selects 1, 2, 3, 4, 5 channel output, and RB_PWM_SYNC_START is set to 1 to start output.*	0
1	RB_DMA_ADDR_LOOP	RW	DMA cyclic output enable: 1: DMA cyclic output; 0: DMA one-way output.	0
0	RB_DMA_ENABLE	RW	DMA enable (valid only with 16 data width): 1: Enable DMA; 0: Disable DMA.	0

PWM Data Hold Register Group1 (R32_PWM1_3_DATA)

Bit	Name	Access	Description	Reset value
[7:3]	Reserved	RO	Reserved	XXh
[23:16]	R8_PWM3_DATA	RW	PWM3 data hold register (8-bit width channel 3).	XXh
[15:8]	R8_PWM2_DATA	RW	PWM2 data hold register (8-bit width channel 2).	XXh
[7:0]	R8_PWM1_DATA	RW	PWM1 data hold register (8-bit width channel 1).	XXh
[31:16]	R16_PWM2_DATA	RW	PWM2 data hold register (16-bit width channel 2).	XXXXh
[15:0]	R16_PWM1_DATA	RW	PWM1 data hold register (16-bit width channel 1).	XXXXh

PWM Data Hold Register Group2 (R16_PWM3_DATA)

Bit	Name	Access	Description	Reset value
[15:0]	R16_PWM3_DATA	RW	PWM6 data hold register (16-bit width channel 3).	XXXXh

PWM Data Hold Register Group3 (R32_PWM4_5_DATA)

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	XXXXh
[15:8]	R8_PWM5_DATA	RW	PWM5 data hold register (8-bit width channel 5).	XXh
[7:0]	R8_PWM4_DATA	RW	PWM4 data hold register (8-bit width channel 4).	XXh
[31:16]	R16_PWM5_DATA	RW	PWM5 data hold register (16-bit width channel 5).	XXXXh
[15:0]	R16_PWM4_DATA	RW	PWM4 data hold register (16-bit width channel 4).	XXXXh

PWMx Interrupt Enable Register (R8_PWM_INT_EN)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved	0
6	RB_PWM_IE_OVER	RW	FIFO read air interrupt enable bit: 1: Enable interrupt; 0: Disable interrupt.	0
5	RB_PWM_IE_DMA	RW	DMA transmission end interrupt enable bit: 1: Enable interrupt; 0: Disable interrupt.	0
4	RB_PWM_IE_FIFO	RW	FIFO data less than 4 interrupt enable bits: 1: Enable interrupt; 0: Disable interrupt.	0
3	Reserved	RO	Reserved	0
2	RB_PWM1_IE_CYC	RW	PWM4, 5 cycle end interrupt enable bit: 1: Enable interrupt; 0: Disable interrupt.	0
1	RB_PWM_CYC_PRE	RW	Select the interrupt time point at the end of	0

			PWM cycle: 1. Generate an interrupt 16 counts in advance (take 8-bit data width as an example, generate an interrupt when counting to 240); 0: Interrupt is generated by counting 2 times in advance (taking 8-bit data width as an example, interrupt is generated when counting to 254).	
0	RB_PWM_IE_CYC	RW	PPWM1, 2 and 3 cycle end interrupt enable bits: 1: Enable interrupt; 0: Disable interrupt.	0

PWMx Interrupt Flag Register (R8_PWM_INT_FLAG)

Bit	Name	Access	Description	Reset value
[7:5]	Reserved	RO	Reserved	000b
4	RB_PWM_IF_OVER	RW1	FIFO overflow interrupt flag bit, write 1 to clear: 1: FIFO overflow; 0: FIFO does not overflow.	0
3	RB_PWM_IF_DMA	RW1	DMA transfer end interrupt flag bit, write 1 to clear: 1: DMA transfer ended; 0: DMA transfer not ended.	0
2	RB_PWM_IF_FIFO	RW1	FIFO data less than 4 interrupt flag bit, write 1 to clear: 1: FIFO data less than 4; 0: Interrupt disabled.	0
1	RB_PWM1_IF_CYC	RW1	PWM4, 5 end of cycle flag bits, write 1 to clear: 1: End of cycle; 0: Not ended.	0
0	RB_PWM_IF_CYC	RW1	PWM1, 2, 3 end of cycle flag bits, write 1 to clear: 1: End of cycle; 0: Not ended.	0

PWM1, 2, 3 Cycle End Register (R16_PWM_CYC_VALUE)

Bit	Name	Access	Description	Reset value
[15:0]	R16_PWM_CYC_VALUE	RW	PWM1, 2, 3 clock cycles when data 16-width.	XXXXh

PWM4, 5 Cycle End Register (R16_PWM_CYC1_VALUE)

Bit	Name	Access	Description	Reset value
[15:0]	R16_PWM_CYC1_VALUE	RW	PWM4, 5 clock cycles when data 16-width.	XXXXh

PWMx Clock Divider Register (R16_PWM_CLOCK_DIV)

Bit	Name	Access	Description	Reset value
[15:0]	R16_PWM_CLOCK_DIV	RW	PWM reference clock frequency division coefficient. Fpwm = Fsys/R16_PWM_CLOCK_DIV。	0000h

PWMx DMA Buffer Current Address Register (R32_PWM_DMA_NOW)

Bit	Name	Access	Description	Reset value
[31:0]	R32_PWM_DMA_NOW	RW	Current address of the DMA data buffer. Can be used to calculate the number of times it has been converted, calculation method: COUNT = PWM_DMA_NOW - PWM_DMA_BEG.	XXXXXXXXh

PWMx DMA Buffer Begin Address Register (R32_PWM_DMA_BEG)

Bit	Name	Access	Description	Reset value
[31:0]	R32_PWM_DMA_BEG	RW	DMA data buffer Begin address	XXXXXXXXh

PWMx DMA Buffer End Address Register (R32_PWM_DMA_END)

Bit	Name	Access	Description	Reset value
[31:0]	R32_PWM_DMA_END	RW	DMA data buffer end address (excluding)	XXXXXXXXh

11.3 PWM Configuration

(1) Set the R16_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 PWMx 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 PWM1-PWM5 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:
Ncyc=2^n-RB_PWM_CYCLE_SEL (data width n=8/7/6);
Ncyc=RB_PWM_CYC_VALUE (n=6)
R8_PWMx_DATA calculation formula:
R8_PWMx_DATA=PWMx duty cycle*Ncyc

PWMx output frequency Fpwmout Calculation formula:
PWMx output frequency Fpwmout=Fpwm/Ncyc=Fsys/R16_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.

11.4 DMA Function

The PWM module has DMA function (can be used only when the data width is 16 bits). When using DMA to complete interrupts, you need to pay attention to the configuration order of the relevant registers.
The steps to enable DMA to complete interrupts are as follows:

(1) Read R32_PWM_DMA_NOW and assign R32_PWM_DMA_END to any value that is not equal to R32_PWM_DMA_NOW (for example, the value can be R32 _ PWM _ DMA _ Now+0x100);
(2) Write 1 to clear bit RB_PWM_IF_DMA in R8_PWM_INT_FLAG;
(3) Set bit RB_PWM_IE_DMA in R8_PWM_INT_EN to 1.

The steps to clear the DMA in acyclic mode to complete the interrupt are as follows:

(1) Read R32_PWM_DMA_NOW and assign R32_PWM_DMA_END to any value that is not equal to R32_PWM_DMA_NOW (for example, the value can be R32 _ PWM _ DMA _ Now+0x100);
(2) Clear bit RB_PWM_IF_DMA_END of R8_PWM_INT_FLAG by writing 1.

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

(1) In the DMA completion interrupt, the DMA enable is turned off first, and bit RB_DMA_ENABLE of R8_PWM_DMA_CTRL is turned off by writing 0;
(2) Clear bit RB_PWM_IF_DMA_END of R8_PWM_INT_FLAG by writing 1.

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

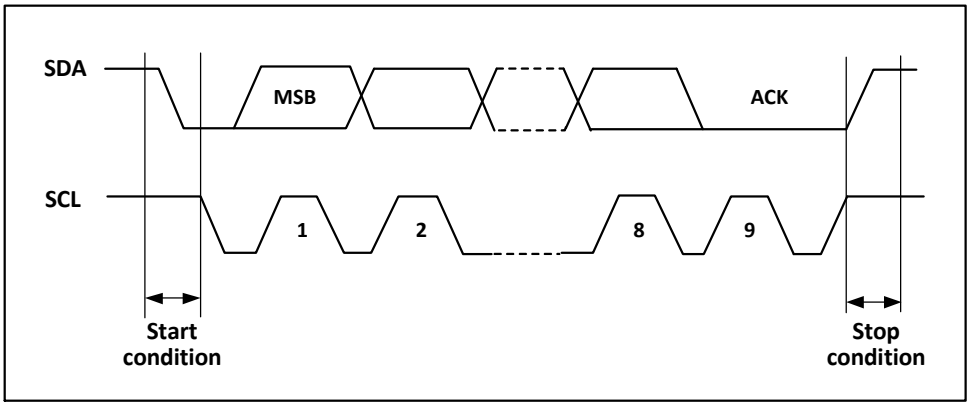
CH572 and CH570 chip provide an I2C interface with the following characteristics:

- 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 产 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).

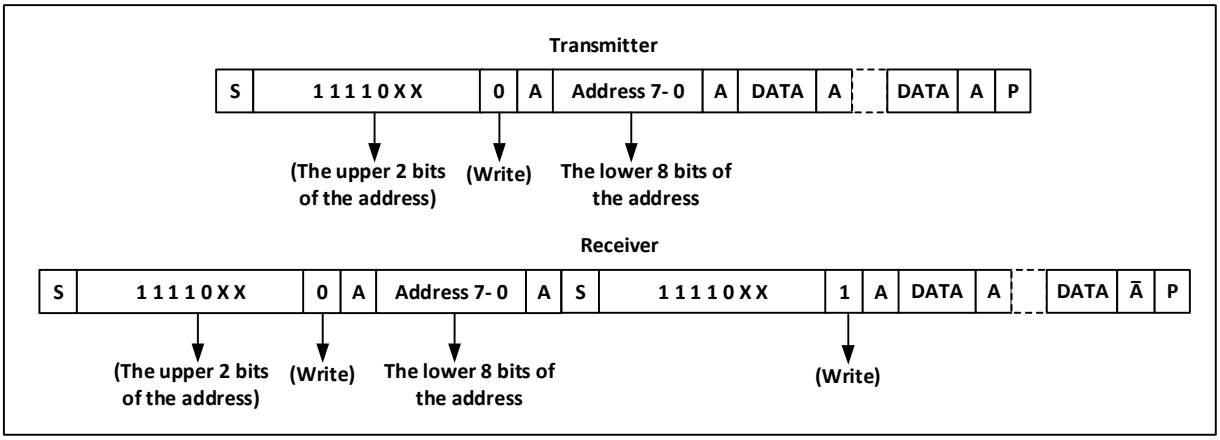
   After the header sequence is sent, the ADD10 bit of the status register will be set. If ITEVTEN bit is set, an interrupt will be generated. At this time, it is required to read R16_I2C_STAR1 register, then write the second address byte to the data register, to clear ADD10 bit.

   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 12-3, 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 12-3 Master transceiver data in 10-bit mode



6) 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.

7) 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) 2-wire communication protocol;
3) 7-bit addressing format.

Differences:

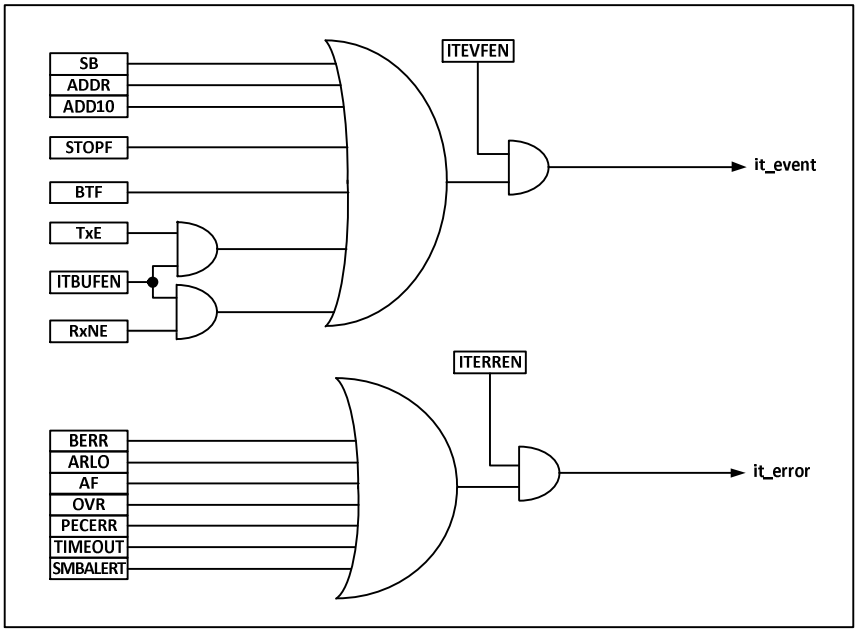
1) The speed of I2C can be up to 400KHz. The speed of SMBus can only be up to 100KHz, and the minimum speed of SMBus is 10KHz;
2) SMBus: 35 mS clock low timeout. I2C: no timeout;
3) SMBus has fixed logic levels. While I2C does not have fixed logic levels, which are determined by VDD;
4) SMBus has bus 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 12-4 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 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 (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. *Note: This bit can reset I2C when no Stop condition is detected on the bus but the busy bit is 1.*	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. *Note: PEC is corrupted when an arbitration is lost.*	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 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. *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.*	0
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_NOSTRE	RW	Clock stretching disable. This bit can be used to	0

	TCH		disable clock stretching when ADDB or BTF flag is 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 Register2 (R16_I2C_CTRL2)

Bit	Name	Access	Description	Reset value
[15:11]	Reserved	RO	Reserved.	00000b
10	RB_I2C_ITBUFEN	RW	Buffer interrupt enable. 1: When TxE or RxEN is set, event interrupt is generated; 0: When TxE 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 TxE or RxEN event; If ITBUFEN=1, TxE 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.	00b

[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.	000000b

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	00000b
[9:8]	RB_I2C_ADD9_8	RW	Bus address. Bits9-8 when using 10-bit address. Ignore when using 7-bit address.	00b
[7:1]	RB_I2C_ADD7_1	RW	Bus address, bits7-1.	0000000b
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_OADDR1)

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved.	00h
[7:1]	RB_I2C_ADD2	RW	Bus address, bits7-1 in dual addressing mode.	00h
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.	00h
[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.	00h

I2C Status Register1 (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. *Note: When this bit is set in slave mode, the slave device will reset the communication and the hardware will release the bus; when this bit is set in master mode, the hardware will issue a stop condition.*	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	Overrun/Underrun flag. 1: Overrun/Underrun: 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 overrun/underrun.	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_BTF	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 Register2 (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.	00h
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: Busy bus: SDA or SCL LOW; 0: Idle bus, and no communication.	0
0	RB_I2C_MSL	RO	Master/slave mode indication. It can be cleared by hardware when the interface is in Master mod (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)

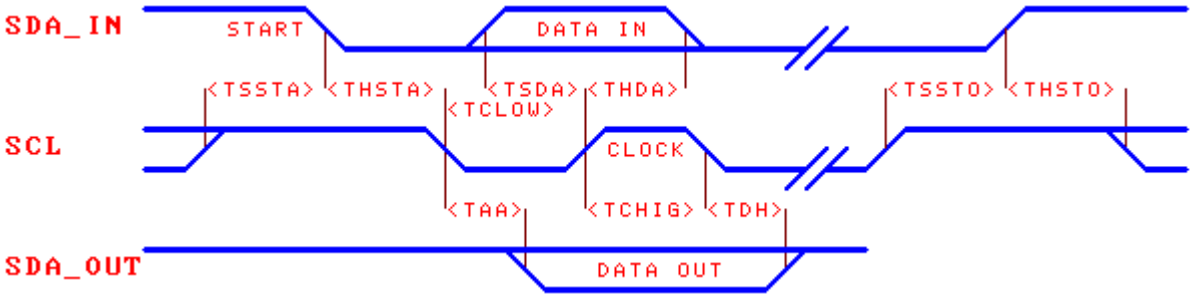
Bit	Name	Access	Description	Reset value

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.	00b
[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 * T_{HCLK} T_{low} = CCR * T_{HCLK} 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 the standard mode, the SCL frequency of 100kHz is generated: CCR = FREQR/(2*100KHz), when FREQR = 60MHz and T_{HCLK} = 1/60MHz, CCR = 60MHz/(2*100KHz) = 300, and CCR is written in 0x12C.	000h

I2C Rise Time Register (R16_I2C_RTR)

Bit	Name	Access	Description	Reset value
[15:6]	Reserved	RO	Reserved.	000h
[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℃，VIO33=3.3V)	Number of system clocks	60M system clock time (nS)
TSSTA	Setup time of SCL HIGH before SDA falling edge	5	84
THSTA	Hold time of SCL HIGH after SDA falling edge	5	84
TSDA	Setup time of SDA data before SCL rising edge	6	100
THDA	Hold time of SDA data after SCL rising edge	>TCHIG	>84
TSSTO	Setup time of SCL HIGH before SDA rising edge	5	84
THSTO	Hold time of SCL HIGH after SDA rising edge	5	84
TCLOW	Time of SCL clock LOW	6	100
TCHIG	Time of SCL clock HIGH	5	84
TAA	SCL falling edge to SDA output valid	2	34
TDH	SDA output hold time after SCL falling edge	2	34

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	0xXXXXXXXX
R32_IWDG_CFG	0x40001004	IWDG configuration register	0x4FFFXFFF

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) 0: Turn off the stop switch. 1: Turn on 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.	000b
[11:0]	RLR	RW	RL[11:0]: Watchdog counter reload (with write protection) Used to define the reload value for the watchdog counter, which is transferred to the counter whenever 0xAAAA is written to the IWDG_KR register. The counter then counts down from this value. The watchdog timeout period can be calculated from this reload value and the clock prescaler value.	FFFh

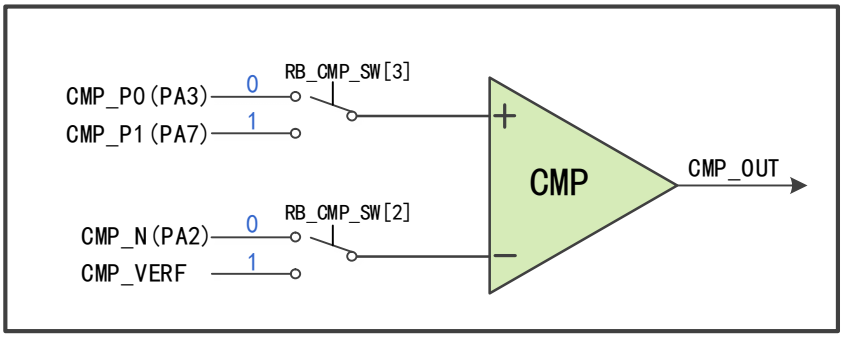
Chapter 14 Comparator (CMP)

14.1 Introduction to CMP

The CH572 provides a voltage comparator CMP with 2 inputs P and N. One of the input terminals can be selected as the reference voltage for comparison. When the input P terminal voltage is less than the input N terminal voltage, the comparator outputs a low level, and vice versa. Its structural block diagram is shown in the figure below.

The N terminal of the voltage comparator CMP can be selected from the built-in 16-speed reference voltage in the form of a binary search, and is equivalent to a 4-bit ADC through 4 comparisons.

Figure 14-1 CMP structure block diagram



14.2 Register Description

Table 14-1 CMP registers

Name	Access address	Description	Reset value
R32_CMP_CTRL	0x40001054	CMP control register	00000000h

CMP Control Register (R32_CMP_CTRL)

Bit	Name	Access	Description	Reset value
[31:26]	Reserved	RO	Reserved	0
25	RB_APR_OUT_CMP	RW	CMP outputs the signal in real time.	0
[24:17]	Reserved	RO	Reserved	0
16	RB_CMP_IF	RW1	CMP interrupt flag, write 1 to clear. 1: A comparison event has been generated; 0: No comparison event has been generated.	0
[15:12]	Reserved	RO	Reserved	0
[11:10]	RB_CMP_OUT_SEL		Comparator interrupt generation event selection: 11: Rising edge; 10: Falling edge; 01: Low level; 00: High level.	00

9	Reserved	RO	Reserved	0
8	RB_CMP_INT	RW	Comparator interrupt enable: 1: Turn on the interrupt; 0: Turn off the interrupt.	0
[7:4]	RB_CMP_NREF_LEVEL	RW	Comparator negative terminal reference level selection: 0000: 50Mv; …… 50mV per gear; …… 1110: 750mV; 1111: 800mV.	0
[3:2]	RB_CMP_SW	RW	CMP channel selection: Bit[3]: Select comparator positive input signal selection (PA3 or PA7): 1: PA7; 0: PA3. Bit[2]: Select comparator negative input signal selection (CMP_VERF or PA2): 1: CMP_VERF; 0: PA2.	00
1	RB_CMP_CAP	RW	Timer capture channel signal source selection: 1: Input comes from the comparator; 0: Input comes from the GPIO port.	0
0	RB_CMP_EN	RW	CMP enable: 1: Enable the CMP; 0: Disable the CMP.	0

14.3 CMP Configuration

(1) Configure RB_CMP_SW for channel selection;
(2) Configure RB_CMP_NREF_LEVEL to select the reference voltage;
(3) Configure interrupts as needed, and select interrupts to generate events;
(4) Set RB_CMP_EN, and the comparator starts to work.

Chapter 15 KEYSCAN

15.1 Introduction to KEYSCAN

CH572 chip provides a KEYSCAN module, which supports automatic scanning during sleep and non-sleep of the system. Its clock source is LSI.

15.2 Main Features

- Support 5 I/O ports, with a total of 20 keys, including 10 independent keys and 10 matrix keys.
- Support the interruption of scanning the same key number value for a set number of times and the interruption of single scanning end.
- Each column of the independent keypad supports simultaneous pressing of multiple keys. At this time, if a key is pressed in the independent keypad, the operation of the matrix keypad will be regarded as invalid. If there is no key press in the independent key area, the key detection module supports one or more key presses in the matrix keyboard area without electrical direct connection.

15.2 Register Description

Table 16-1 KEYSCAN registers

Name	Access address	Description	Reset value
R16_KEY_SCAN_CTRL	0x40001064	KEYSCAN control register	0x0000
R8_KEY_SCAN_INT_EN	0x40001066	KEYSCAN interrupt enable register	0x00
R8_KEY_SCAN_INT_FLAG	0x40001067	KEYSCAN interrupt flag register	0x00
R32_KEY_SCAN_NUMB	0x40001068	KEYSCAN key number register	0x00000000

KEYSCAN Control Register (R16_KEY_SCAN_CTRL)

Bit	Name	Access	Description	Reset value
15	Reserved	RO	Reserved	0
14	RB_CLR_WAKEUP_EN	RW	Automatically clear the wake-up signal after wake-up Enable: 1: Enable; 0: Disable.	0
13	RB_SCAN_1END_WAKE_EN	RW	The wake-up signal is triggered after a single scan: 1: Enable; 0: Disable.	0
[12:8]	RB_PIN_SCAN_EN	RW	I/O pins participating in key scanning are enabled, and bits [12] ~ [8] are PA11, PA10, PA8, PA3 and PA2 in turn: 1: Participate in key scanning; 0: Do not participate in key scanning.	0

[7:4]	RB_SCAN_CLK_DIV	RW	Scan the clock frequency division, and you can set 0000 (lowest frequency division 1) to 1111 (highest frequency division 16).	0
[3:1]	RB_SCAN_CNT_END	RW	Set the number of times to scan to the same key value. The number of times to repeat is 0 means that it is not working, and the maximum can be set to 7.	0
0	RB_SCAN_START_EN	RW	Start scanning Enable: 1: Enable; 0: Disable.	0

KEYSCAN Interrupt Enable Register (R8_KEY_SCAN_INT_EN)

Bit	Name	Access	Description	Reset value
[7:2]	Reserved	RO	Reserved	0
1	RB_SCAN_1END_IE	RW	Interrupt enabled after a single scan. 1: Turn on the interrupt; 0: Turn off the interrupt.	0
0	RB_KEY_PRESSED_IE	RW	The key press is detected and the same key press value reaches the set value (RB_SCAN_CNT_END) interrupt enable. 1: Turn on the interrupt; 0: Turn off the interrupt.	0

KEYSCAN Interrupt Flag Register (R8_KEY_SCAN_INT_FLAG)

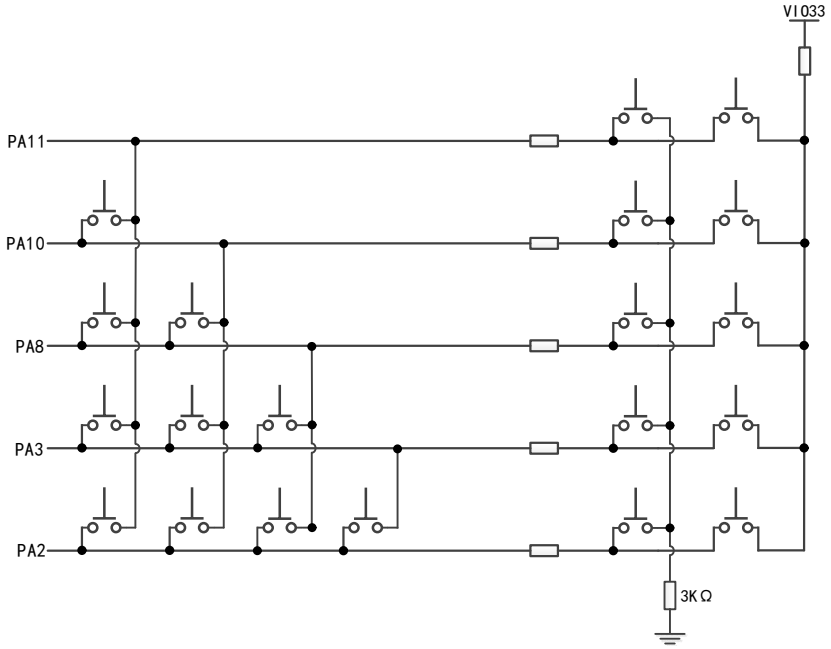
Bit	Name	Access	Description	Reset value
[7:2]	Reserved	RO	Reserved	0
1	RB_SCAN_1END_IF	RW1	Interrupt sign after word scanning, write 1 to clear 0 (RB_SCAN_1END_WAKE_EN is required). 1: The single scan is over; 0：IDLE。	0
0	RB_KEY_PRESSED_IF	RW1	When it is detected that a key is pressed, and the number of times the same key value reaches the set value (RB_SCAN_CNT_END), the interrupt flag will be written with 1 clear (0RB_KEY_SCAN_NUMB will not trigger). 1: The number of times of the same key value reaches the set value; 0：IDLE。	0

KEYSCAN Key Number Register (R32_KEY_SCAN_NUMB)

Bit	Name	Access	Description	Reset value
[31:23]	Reserved	RO	Reserved	0
[22:20]	RB_KEY_SCAN_CNT	RO	The current number of key scans.	0
[19:0]	RB_KEY_SCAN_NUMB	RO	When the current key number value, the corresponding bit of the number is 1, it means that the current key number has been detected to be pressed.	0

15.3 KEYSCAN Function Description

Figure 15-1 5 key areas where configurable I/O can wake up in low-power mode



After the KEYSCAN function is enabled, 5 I/Os (PA2/PA3/PA8/PA10/PA11) can be used for key scanning, and these I/Os can be configured to participate in key scanning respectively. Up to 20 keys are supported, including 10 keys in matrix keyboard area and 10 keys in independent keyboard area.

Table 15-2 KEYSCAN key information table

Register	Bit	Key number							
R32_KEY_SCAN_NUMB	[19:16]					PA2 PA3 (PD)	PA3 PA8 (PD)	PA2 PA8 (PD)	PA8 PA10 (PD)
	[15:8]	PA3 PA10 (PD)	PA2 PA10 (PD)	PA10 PA11 (PD)	PA8 PA11 (PD)	PA3 PA11 (PD)	PA2 PA11 (PD)	PA11 (PD)	PA10 (PD)
	[7:0]	PA8 (PD)	PA3 (PD)	PA2 (PD)	PA11 (PU)	PA10 (PU)	PA8 (PU)	PA3 (PU)	PA2 (PU)

Note: 1. PD= internal pull-down resistor; PU= internal pull-up resistor.

2. 2 lines of GPIO in the table indicate that 2 GPIO ports are connected.

The working steps of the key scanning module are as follows:
(1) First detect the pull-up key input of the independent key;
(2) Then detect the pull-down key input of the independent key;
(3) And finally detect the matrix key input;
(4) If set after a single scan to trigger the wake-up signal enable, this time to trigger the wake-up signal, otherwise wait for the detection of the same value number of times to reach the set value, trigger the wake-up signal.

15.4 KEYSCAN Configuration

(1) Set RB_SCAN_CLK_DIV, and configure the sampling clock frequency division coefficient, with the highest frequency division of 16;
(2) Set RB_PIN_SCAN_EN, and configure I/O ports participating in key scanning;
(3) Set RB_SCAN_CNT_END and configure the final value of sampling the same value. If RB_KEY_PRESSED_IE is turned on, an interrupt will be triggered when the final value of sampling the same value is reached;
(4) If RB_SCAN_1END_IE is enabled, an interrupt will be triggered after a single scan, regardless of whether a key press is detected;
(5) Enable RB_SCAN_START_EN, and turn on the wake-up signal for key scanning. If the key pressing is detected and the same value is reached for many times in succession, the wake-up signal will be triggered. If RB_SCAN_1END_WAKE_EN is turned on, the wake-up signal will be triggered after a single scanning. If RB_CLR_WAKEUP_EN is turned on, the wake-up signal will be cleared after the system wakes up.

Chapter 16 USB Controller

16.1 Introduction to USB Controller

The chip embedded with a USBFS master-slave controller and transceiver with the following features:

- USB Host function and USB Device function.
- USB2.0 full-speed 12Mbps and low-speed 1.5Mbps.
- USB control transmission, bulk transmission, interrupt transmission, synchronous/real-time transmission.
- Data packets up to 64 bytes, built-in FIFO, support interrupt and DMA.
- Optional 1-wire USB mode, support 1-wire USB communication.

16.2 Register Description

The base address of the full-speed USB controller is 0x40008000 and the USB related registers are divided into 3 parts, some of which are multiplexed in host and device mode.

(1) USB global registers.

(2) USB device control registers.

(3) USB host control registers.

16.2.1 Description of Global Registers

Table 16-1 USB registers (Those marked in grey are controlled by RB_UC_RESET_SIE reset)

Name	Access address	Description	Reset value
R8_USB_CTRL	0x40008000	USB control register	0x06
R8_USB_INT_EN	0x40008002	USB interrupt enable register	0x00
R8_USB_DEV_AD	0x40008003	USB device address register	0x00
R32_USB_STATUS	0x40008004	USB status register	0xXX20XXXX
R8_USB_MIS_ST	0x40008005	USB miscellaneous status register	0xXX
R8_USB_INT_FG	0x40008006	USB interrupt flag register	0x20
R8_USB_INT_ST	0x40008007	USB interrupt status register	0x3X
R8_USB_RX_LEN	0x40008008	USB receiving length register	0xXX

USB Control Register (R8_USB_CTRL)

Bit	Name	Access	Description	Reset value
7	RB_UC_HOST_MODE	RW	USB working mode selection: 1: HOST; 0: DEVICE.	0
6	RB_UC_LOW_SPEED	RW	USB bus signal transmission rate selection: 1: 1.5Mbps; 0: 12Mbps.	0
5	RB_UC_DEV_PU_EN	RW	USB device enable and internal pull-up resistor control in USB device mode. If it is 1, USB device transmission is enabled and the internal pull-up resistor is also enabled. RB_UDP_PU_EN can replace this bit.	0

[5:4]	MASK_UC_SYS_CTRL	RW	See the table below to configure the USB system.	0
3	RB_UC_INT_BUSY	RW	Auto pause enable before USB transmission completion interrupt flag is not cleared: 1: It will automatically pause before the interrupt flag UIF_TRANSFER is not cleared. In device mode, it will automatically respond to busy NAK. In host mode, it will automatically pause subsequent transmission; 0: Not pause.	0
2	RB_UC_RESET_SIE	RW	Software reset control of USB protocol processor: 1: Force to reset the USB protocol processor (SIE), software is needed to cleared; 0: Not reset.	1
1	RB_UC_CLR_ALL	RW	USB FIFO and interrupt flag clear: 1: Force to empty and clear; 0: Not clear.	1
0	RB_UC_DMA_EN	RW	USB DMA and DMA interrupt control: 1: Enable the DMA function and DMA interrupt; 0: Disable DMA.	0

RB_UC_HOST_MODE and MASK_UC_SYS_CTRL constitute the USB system control combination:

RB_UC_HOST_MODE	MASK_UC_SYS_CTRL	USB system control description
0	00	Disable USB device function and disable the internal pull-up resistor. *Note: If RB_UDP_PU_EN=1, the DP pull-up resistor will be enabled by force.*
0	01	Enable USB device function and disable the internal pull-up resistor, the external pull-up is needed. *Note: If RB_UDP_PU_EN=1, the DP pull-up resistor will be enabled by force.*
0	1x	Enable USB device function and enable the internal 1.5K pull-up resistor. The pull-up resistor has priority over the pull-down resistor, and can also be used in GPIO mode.
1	00	USB host mode, in normal working status.
1	01	USB host mode, DP/DM is forced to output SE0 status.
1	10	USB host mode, DP/DM is forced to output J status.
1	11	USB host mode, DP/DM is forced to output K status/wake-up.

USB Interrupt Enable Register (R8_USB_INT_EN)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved	0

6	RB_UIE_DEV_NAK	RW	In USB device mode, receive NAK interrupt: 1: Enable interrupt; 0: Disable interrupt.	0
5	RB_MOD_1_WIRE	RW	USB single wire mode enable: 1: Enable; 0: Disable.	0
4	RB_UIE_FIFO_OV	RW	FIFO overflow interrupt: 1: Enable interrupt; 0: Disable interrupt.	0
3	RB_UIE_HST_SOF	RW	In the USB host mode, SOF timing interrupt: 1: Enable interrupt; 0: Disable interrupt.	0
2	RB_UIE_SUSPEND	RW	USB bus suspend or wake-up event interrupt: 1: Enable interrupt; 0: Disable interrupt.	0
1	RB_UIE_TRANSFER	RW	USB transfer completion interrupt: 1: Enable interrupt; 0: Disable interrupt.	0
0	RB_UIE_DETECT	RW	In USB host mode, USB device connection or disconnection event interrupt: 1: Enable interrupt; 0: Disable interrupt.	0
	RB_UIE_BUS_RST	RW	In USB device mode; USB bus reset event interrupt: 1: Enable interrupt; 0: Disable interrupt.	0

USB Device Address Register (R8_USB_DEV_AD)

Bit	Name	Access	Description	Reset value
7	RB_UDA_GP_BIT	RW	USB general flag, user-defined.	0
[6:0]	MASK_USB_ADDR	RW	Host mode: address of USB device currently operated; Device mode: the address of the USB itself.	0000000b

USB Miscellaneous Status Register (R8_USB_MIS_ST)

Bit	Name	Access	Description	Reset value
7	RB_UMS_SOF_PRES	RO	SOF packet indication status in USB host mode: 1: SOF packet will be sent, and it will be automatically delayed if there are other USB data packets; 0: No SOF package is sent.	X
6	RB_UMS_SOF_ACT	RO	SOF packet transmission status in USB host mode: 1: SOF packet is being sent out; 0: The transmission is completed or idle.	X
5	RB_UMS_SIE_FREE	RO	USB protocol processor free status: 1: Free; 0: Busy, USB transmission is in progress.	1
4	RB_UMS_R_FIFO_RDY	RO	USB receiver FIFO data ready status: 1: Receiver FIFO not empty;	0

			0: Receiver FIFO empty.	
3	RB_UMS_BUS_RESET	RO	USB bus reset status: 1: The current USB bus is at the reset status; 0: The current USB bus is at the non-reset status.	X
2	RB_UMS_SUSPEND	RO	USB suspend status: 1: The USB bus is in suspended status, and there is no USB activity for a period of time; 0: USB bus is in non-suspended status.	0
1	RB_UMS_DM_LEVEL	RO	In USB host mode, the level status of the DM pin when the device is just connected to the USB port, used to determine speed: 1: High level/ low-speed; 0: Low level/ full-speed.	0
0	RB_UMS_DEV_ATTACH	RO	USB device connection status of the port in USB host mode: 1: The port has been connected to a USB device; 0: No USB device is connected to the port.	0

USB Interrupt Flag Register (R8_USB_INT_FG)

Bit	Name	Access	Description	Reset value
7	RB_U_IS_NAK	RO	In USB device mode, NAK acknowledge status: 1: NAK acknowledge during current USB transmission; 0: No NAK acknowledge.	0
6	RB_U_TOG_OK	RO	Current USB transmission DATA0/1 synchronous flag match status: 1: Synchronous; 0: Asynchronous.	0
5	RB_U_SIE_FREE	RO	USB protocol processor idle status: 1: USB idle; 0: Busy, USB transmission is in progress.	1
4	RB_UIF_FIFO_OV	RW	USB FIFO overflow interrupt flag. Write 1 to reset: 1: FIFO overflow trigger; 0: No event.	0
3	RB_UIF_HST_SOF	RW	SOF timing interrupt flag in USB host mode. Write 1 to reset: 1: SOF packet transmission completion trigger; 0: No event.	0
2	RB_UIF_SUSPEND	RW	USB bus suspend or wake-up event interrupt flag. Write 1 to reset: 1: USB suspend event or wake-up event trigger; 0: No event.	0
1	RB_UIF_TRANSFER	RW	USB transmission completion interrupt flag.	0

			Write 1 to reset: 1: USB transmission completion trigger; 0: No event.	
0	RB_UIF_DETECT	RW	In USB host mode, the USB device connection or disconnection event interrupt flag. Write 1 to reset: 1: USB device connection or disconnection trigger is detected; 0: No event.	0
	RB_UIF_BUS_RST	RW	USB bus reset event interrupt flag in USB device mode. Write 1 to reset: 1: USB bus reset event trigger; 0: No event.	0

USB Interrupt Status Register (R8_USB_INT_ST)

Bit	Name	Access	Description	Reset value
7	RB_UIS_SETUP_ACT	RO	In USB device mode, when this bit is 1, 8-byte SETUP request packet has been successfully received. SETUP token does not affect RB_UIS_TOG_OK, MASK_UIS_TOKEN, MASK_UIS_ENDP and R8_USB_RX_LEN.	0
6	RB_UIS_TOG_OK	RO	Current USB transmission DATA0/1 synchronization flag matching status. The same as RB_U_TOG_OK: 1: Synchronous; 0: Asynchronous.	0
[5:4]	MASK_UIS_TOKEN	RO	In device mode, the token PID of the current USB transfer transaction.	11b
[3:0]	MASK_UIS_ENDP	RO	In device mode, the endpoint number of the current USB transfer transaction.	XXXXb
	MASK_UIS_H_RES	RO	In host mode, the response PID identification of the current USB transfer transaction. 0000: the device has no response or timeout; Other values: respond PID.	XXXXb

MASK_UIS_TOKEN is used to identify the token PID of the current USB transfer transaction in USB device mode: 00 represents OUT packet; 10 represents IN packet; 11 represents idle; 01 undefined.

When MASK_UIS_TOKEN is not idle, and RB_UIS_SETUP_ACT is also 1, it is required to process the former first, and clear RB_UIF_TRANSFER once after the former is processed to make the former enter the idle status, and then process the latter, and finally clear RB_UIF_TRANSFER again.

MASK_UIS_H_RES is only valid in host mode. In host mode, if the host sends OUT/SETUP token packet, the PID will be the handshake packet ACK/NAK/STALL, or that device has no response/timeout. If the host sends IN token packet, the PID will the PID of the data packet (DATA0/DATA1) or the handshake packet PID.

USB Receiving Length Register (R8_USB_RX_LEN)

Bit	Name	Access	Description	Reset value
[7:0]	R8_USB_RX_LEN	RO	The number of data bytes received by the current USB endpoint.	XXh

16.2.2 Description of Device Registers

In USB device mode, the chip provides 8 sets of bidirectional endpoints, including endpoint0, endpoint1, endpoint2, endpoint3, endpoint4, endpoint5, endpoint6 and endpoint7. The maximum data packet length of each endpoint is 64 bytes.

Endpoint0 is the default endpoint and supports control transmission. The transmission and reception share a 64-byte data buffer.

Endpoint1, endpoint2 and endpoint3 each includes a transmission endpoint IN and a reception endpoint OUT.

The reception and the transmission each has a separate 64-byte or double 64-byte data buffer, supporting bulk transmission, interrupt transmission, and real-time/synchronous transmission.

Endpoint4, endpoint5, endpoint6 and endpoint7 each includes a transmission endpoint IN and a reception endpoint OUT. The reception and the transmission each has a separate 64-byte data buffer, supporting bulk transmission, interrupt transmission, and real-time/synchronous transmission.

Each set of endpoint has a control register R8_UEPn_CTRL and a transmit length register R8_UEPn_T_LEN (n=0/1/2/3/4/5/6/7), used to set the synchronization trigger bit of endpoint, the response to OUT transactions and IN transactions and the length of data to be sent.

As the necessary USB bus pull-up resistor of USB device, it can be set whether to be enabled by software at any time. When RB_UC_DEV_PU_EN of USB control register R8_USB_CTRL is set to 1, the controller will set according to the speed of RB_UD_LOW_SPEED, internally connect the pull-up resistor with the DP/DM pin of the USB bus and enable the USB device function. The above control cannot be used in sleep or power-down mode, but RB_PIN_USB_DP_PU of R16_PIN_ALTERNATE can enable the pull-up resistor of DP pin by force without being affected by sleep mode.

When a USB bus reset or USB bus suspend or wake-up event is detected, or when the USB successfully processes data sending or receiving, the USB protocol processor will set corresponding interrupt flag. If the interrupt enabling is switched on, the corresponding interrupt request will also be generated. The application program can directly query or query and analyze the interrupt flag register R8_USB_INT_FG in the USB interrupt service program, and perform corresponding processing according to RB_UIF_BUS_RST and RB_UIF_SUSPEND. In addition, if RB_UIF_TRANSFER is valid, it is required to continue to analyze the USB interrupt state register R8_USB_INT_ST, and perform the corresponding processing according to the current endpoint number MASK_UIS_ENDP and the current transaction token PID identification MASK_UIS_TOKEN. If the synchronization trigger bit RB_UEP_R_TOG of OUT transaction of each endpoint is set in advance, RB_U_TOG_OK or RB_UIS_TOG_OK can be used to judge whether the synchronization trigger bit of the data packet received matches the synchronization trigger bit of the endpoint. If the data is synchronous, the data is valid. If the data is not synchronous, the data should be discarded. Every time the USB sending or receiving interrupt is processed, the synchronization trigger bit of corresponding endpoint should be modified correctly to detect whether the data packet sent next time and the data packet received next time are synchronous. In addition, RB_UEP_AUTO_TOG can be set to automatically flip the corresponding synchronization trigger bit after sending or receiving successfully.

The data to be sent by each endpoint is in their own buffer, and the length of the data to be sent is independently set in R8_UEPn_T_LEN. The data received by each endpoint is in their own buffer, but the length of the data received is in the USB length receiving register R8_USB_RX_LEN, and it can be distinguished according to the current endpoint number when USB receives an interrupt.

Table 16-2 USB device registers (those marked in grey are controlled by RB_UC_RESET_SIE reset)

Name	Access address	Description	Reset value
R8_UDEV_CTRL	0x40008001	USB device physical port control register	0xX0
R8_UEP4_1_MOD	0x4000800c	Endpoint 1/4 mode control register	0x00
R8_UEP2_3_MOD	0x4000800d	Endpoint 2/3 mode control register	0x00
R8_UEP567_MOD	0x4000800e	Endpoint 5/6/7 mode control register	0x00
R16_UEP0_DMA	0x40008010	Start address of endpoint0 buffer	0xXXXX
R16_UEP1_DMA	0x40008014	Start address of endpoint1 buffer	0xXXXX
R16_UEP2_DMA	0x40008018	Start address of endpoint2 buffer	0xXXXX
R16_UEP3_DMA	0x4000801c	Start address of endpoint3 buffer	0xXXXX
R8_UEP0_T_LEN	0x40008020	Endpoint0 transmission length register	0xXX
R8_UEP0_CTRL	0x40008022	Endpoint0 control register	0x00
R8_UEP1_T_LEN	0x40008024	Endpoint1 transmission length register	0xXX
R8_UEP1_CTRL	0x40008026	Endpoint1 control register	0x00
R8_UEP2_T_LEN	0x40008028	Endpoint2 transmission length register	0xXX
R8_UEP2_CTRL	0x4000802a	Endpoint2 control register	0x00
R8_UEP3_T_LEN	0x4000802c	Endpoint3 transmission length register	0xXX
R8_UEP3_CTRL	0x4000802e	Endpoint3 control register	0x00
R8_UEP4_T_LEN	0x40008030	Endpoint4 transmission length register	0xXX
R8_UEP4_CTRL	0x40008032	Endpoint4 control register	0x00
R16_UEP5_DMA	0x40008054	Start address of endpoint5 buffer	0xXXXX
R16_UEP6_DMA	0x40008058	Start address of endpoint6 buffer	0xXXXX
R16_UEP7_DMA	0x4000805c	Start address of endpoint7 buffer	0xXXXX
R8_UEP5_T_LEN	0x40008064	Endpoint5 transmission length register	0xXX
R8_UEP5_CTRL	0x40008066	Endpoint5 control register	0x00
R8_UEP6_T_LEN	0x40008068	Endpoint6 transmission length register	0xXX
R8_UEP6_CTRL	0x4000806a	Endpoint6 control register	0x00
R8_UEP7_T_LEN	0x4000806c	Endpoint7 transmission length register	0xXX
R8_UEP7_CTRL	0x4000806e	Endpoint7 control register	0x00
R32_EPX_MODE	0x40008070	Endpoint8-15 control register	0x00000000

USB Device Physical Port Control Register (R8_UDEV_CTRL)

Bit	Name	Access	Description	Reset value
7	RB_UD_PD_DIS	RW	Internal pull-down resistor control of UD+/UD- pin of USB device port: 1: Disable internal pull-down; 0: Enable internal pull-down.	1

			It also can be used in GPIO mode to provide pull-down resistor.	
6	Reserved	RO	Reserved	0
5	RB_UD_DP_PIN	RO	Current UD + pin status: 1: High level; 0: Low level.	X
4	RB_UD_DM_PIN	RO	Current UD- pin status: 1: High level; 0: Low level.	X
3	Reserved	RO	Reserved	0
2	RB_UD_LOW_SPEED	RW	Current UD- pin status: 1: High level; 0: Low level.	0
1	RB_UD_HUB0_RESET	RW	USB HUB0 reset control	0
0	RB_UD_PORT_EN	RW	USB device physical port enable: 1: Enable; 0: Disable.	0

Endpoint 1/4 Mode Control Register (R8_UEP4_1_MOD)

Bit	Name	Access	Description	Reset value
7	RB_UEP1_RX_EN	RW	1: Enable endpoint 1 reception (OUT); 0: Disable endpoint 1 reception.	0
6	RB_UEP1_TX_EN	RW	1: Enable endpoint 1 transmission (IN); 0: Disable endpoint 1 transmission.	0
5	Reserved	RO	Reserved	0
4	RB_UEP1_BUF_MOD	RW	Endpoint 1 data buffer mode control bit.。	0
3	RB_UEP4_RX_EN	RW	1: Enable endpoint 4 reception (OUT); 0: Disable endpoint 4 reception.	0
2	RB_UEP4_TX_EN	RW	1: Enable endpoint 4 transmission (IN); 0: Disable endpoint 4 transmission.	0
[1:0]	Reserved	RO	Reserved	00b

The data buffer modes of USB endpoint0 and endpoint4 are configured by a combination of bUEP4_RX_EN and bUEP4_TX_EN. Refer to the following table for details:

Table 16-3 Endpoint0/4 buffer modes

bUEP4_RX_EN	bUEP4_TX_EN	Description: arrange from low to high with UEP0 DMA as the start address
0	0	Endpoint0 single 64-byte transmit/receive shared buffers (IN and OUT).
1	0	Endpoint0 single 64-byte transmit/receive shared buffers; endpoint4 single 64-byte reception buffers (OUT).
0	1	Endpoint0 single 64-byte transmit/receive shared buffers; endpoint4 single 64-byte transmission buffers (IN).
1	1	Endpoint0 single 64-byte transmit/receive shared buffers; endpoint4 single 64-byte reception buffers (OUT);

		Endpoint4 single 64-byte receive buffer (IN). All 192 bytes are arranged as follows: UEP0_DMA+0 address: 64-byte start address of endpoint0 transmit/receive shared buffer; UEP0_DMA+64 address: 64-byte start address of endpoint4 receive buffer; UEP0_DMA+128 address: 64-byte start address of endpoint4 transmit buffer.

Endpoint 2/3 Mode Control Register (R8_UEP2_3_MOD)

Bit	Name	Access	Description	Reset value
7	RB_UEP3_RX_EN	RW	1: Enable endpoint 3 reception (OUT); 0: Disable endpoint 3 reception.	0
6	RB_UEP3_TX_EN	RW	1: Enable endpoint 3 transmission (IN); 0: Disable endpoint 3 transmission.	0
5	Reserved	RO	Reserved	0
4	RB_UEP3_BUF_MOD	RW	Endpoint 3 data buffer mode control bit.	0
3	RB_UEP2_RX_EN	RW	1: Enable endpoint 2 reception (OUT); 0: Disable endpoint 2 reception.	0
2	RB_UEP2_TX_EN	RW	1: Enable endpoint 2 transmission (IN); 0: Disable endpoint 2 transmission.	0
1	Reserved	RO	Reserved	0
0	RB_UEP2_BUF_MOD	RW	Endpoint 2 data buffer mode control bit.	0

The data buffer modes of USB endpoint1/2/3 are controlled by a combination of RB_UEPn_RX_EN, RB_UEPn_TX_EN and RB_UEPn_BUF_MOD(n=1/2/3) respectively, refer to the following table for details. Among them, in the double 64-byte buffer mode, the first 64-byte buffer will be selected based on RB_UEP_*_TOG=0 and the last 64-byte buffer will be selected based on RB_UEP_*_TOG=1 during USB data transmission, and RB_UEP_AUTO_TOG=1 is set to realize automatic switch.

Table 16-4 Endpoint n buffer modes (n=1/2/3)

RB_UEPn_RX_EN	RB_UEPn_TX_EN	RB_UEPn_BUF_MOD	Description: Arrange from low to high with R16_UEPn_DMA as the start address
0	0	X	Endpoint is disabled, and R16_UEPn_DMA buffer is not used.
1	0	0	Single 64-byte receive buffer (OUT).
1	0	1	Double 64-byte receive buffer (OUT), selected by RB_UEP_R_TOG.
0	1	0	Single 64-byte transmit buffer (IN).
0	1	1	Double 64-byte transmit buffer (IN), selected by RB_UEP_T_TOG.
1	1	0	Single 64-byte receive buffer (OUT), single 64-byte transmit buffer (IN).
1	1	1	Double 64-byte receive buffer (OUT), selected by RB_UEP_R_TOG. Double 64-byte transmit buffer (IN), selected by RB_UEP_T_TOG. All 256 bytes are arranged as follows: UEPn_DMA+0 address: endpoint receive address when RB_UEP_R_TOG=0;

			UEPn_DMA+64 address: endpoint receive address when RB_UEP_R_TOG=1; UEPn_DMA+128 address: endpoint transmit address when RB_UEP_T_TOG=0; UEPn_DMA+192 address: endpoint transmit address when RB_UEP_T_TOG=1.

Endpoint 5/6/7 Mode Control Register (R8_UEP567_MOD)

Bit	Name	Access	Description	Reset value
[7:6]	Reserved	RO	Reserved	00b
5	RB_UEP7_RX_EN	RW	1: Enable endpoint 7 reception (OUT); 0: Disable endpoint 7 reception.	0
4	RB_UEP7_TX_EN	RW	1: Enable endpoint 7 transmission (IN); 0: Disable endpoint 7 transmission.	0
3	RB_UEP6_RX_EN	RW	1: Enable endpoint 6 reception (OUT); 0: Disable endpoint 6 reception.	0
2	RB_UEP6_TX_EN	RW	1: Enable endpoint 6 transmission (IN); 0: Disable endpoint 6 transmission.	0
1	RB_UEP5_RX_EN	RW	1: Enable endpoint 5 reception (OUT); 0: Disable endpoint 5 reception.	0
0	RB_UEP5_TX_EN	RW	1: Enable endpoint 5 transmission (IN); 0: Disable endpoint 5 transmission.	0

The data buffer modes of USB endpoint5, endpoint6 and endpoint7 are controlled by a combination of RB_UEPn_RX_EN and RB_UEPn_TX_EN (n=5/6/7) respectively, refer to the following table for details.

Table 16-5 Endpoint n buffer modes (n=5/6/7)

RB_UEPn_RX_EN	RB_UEPn_TX_EN	Description: Arrange from low to high with R16_UEPn_DMA as the start address
0	0	Endpoint is disabled, and R16_UEPn_DMA buffer is not used.
1	0	Single 64-byte receive buffer (OUT).
0	1	Single 64-byte transmit buffer (IN).
1	1	Single 64-byte receive buffer (OUT), single 64-byte transmit buffer (IN).

Endpoint n Buffer Start Address (R16_UEPn_DMA) (n=0/1/2/3/5/6/7)

Bit	Name	Access	Description	Reset value
[15:0]	R16_UEPn_DMA	RW	Endpoint n buffer start address. The address must be 4-byte aligned.	XXXXh

Note: The length of the buffer that receives data >= min (maximum data packet length possibly received + 2 bytes, 64 bytes).

Endpoint n Transmission Length Register (R8_UEPn_T_LEN) (n=0/1/2/3/4/5/6/7)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved	0

[6:0]	R8_UEPn_T_LEN	RW	Set the number of data bytes that USB endpoint n is ready to send.	XXh

Endpoint 2 Transmission Length Register (R8_UEPn_T_LEN)

Bit	Name	Access	Description	Reset value
7	R8_UEP2_HOST_PID3	RW	Bit 3 of the token identification PID in host mode.	0
[6:0]	R8_UEP2_T_LEN	RW	Set the number of data bytes that USB endpoint n is ready to send.	XXh

Endpoint n Control Register (R8_UEPn_CTRL) (n=0/1/2/3/4/5/6/7)

Bit	Name	Access	Description	Reset value
7	RB_UEP_R_TOG	RW	Expected synchronization trigger bit of the receiver (process OUT transactions) of USB endpoint n: 1: Expect DATA1; 0: Expect DATA0.	0
6	RB_UEP_T_TOG	RW	Synchronization trigger bit of the transmitter (process IN transactions) of USB endpoint n 1: Transmit DATA1; 0: Transmit DATA0.	0
5	Reserved	RO	Reserved	0
4	RB_UEP_AUTO_TOG	RW	Synchronization trigger bit auto flip enable control bit: 1: After the data is sent or received successfully, the corresponding synchronization trigger bit is automatically flipped; 0: Not flipped automatically, but can be switched manually. It only supports endpoint 1/2/3/5/6/7.	0
[3:2]	MASK_UEP_R_RES	RW	Control on the response to OUT transactions by the receiver of USB endpoint n: 00: Respond ACK; 01: Timeout/no response, used for real-time/synchronous transmission of non-endpoint 0; 10: Respond to NAK or busy; 11: Respond to STALL or error.	00b
[1:0]	MASK_UEP_T_RES	RW	Control on response to IN transactions of the transmitter of endpoint n: 00: DATA0/DATA1 data is ready and ACK is expected; 01: Respond to DATA0/DATA1 and expect	00b

			no response, used for real-time/synchronous transmission of non-endpoint 0; 10: Respond to NAK or busy; 11: Respond to STALL or error.	

Endpoint n Control Register (R32_EPX_MODE) (n=8/9/10/11/12/13/14/15)

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RO	Reserved	00h
[23:17]	RB_EP_T_AF	RW	1: Enables endpoint 8-15 transmit Alternate; 0: Disable endpoint 8-15 transmit Alternate.	00h
16	Reserved	RO	Reserved	0
[15:8]	RB_EP_R_EN	RW	1: Enables endpoint 8-15 reception (OUT); 0: Disable endpoint 8-15 reception.	00h
[7:0]	RB_EP_T_EN	RW	1: Enables endpoints 8-15 transmit (IN); 0: Disable endpoint 8-15 transmit.	00h

16.2.3 Description of Host Registers

In USB host mode, the chip provides 1 set of bidirectional host endpoints, including a transmission endpoint OUT and a reception endpoint IN. The maximum length of data packet is 64 bytes. It supports control transmission, interrupt transmission, bulk transmission and real-time/synchronous transmission.

Each USB transaction initiated by host endpoint always automatically sets the RB_UIF_TRANSFER interrupt flag after the processing ends. The application program can directly query or query and analyze the interrupt flag register (R8_USB_INT_FG) in the USB interrupt service program, and perform corresponding processing according to each interrupt flag. In addition, if RB_UIF_TRANSFER is valid, it is required to continue to analyze the USB interrupt status register (R8_USB_INT_ST), and perform the corresponding processing according to the response PID identification (MASK_UIS_H_RES) of the current USB transmission transaction.

If the synchronization trigger bit (RB_UH_R_TOG) of IN transaction of host reception endpoint is set in advance, whether the synchronization trigger bit of the data packet received matches the synchronization trigger bit of the endpoint can be judged through RB_U_TOG_OK or RB_UIS_TOG_OK. If the data is synchronous, the data is valid. If the data is not synchronous, the data should be discarded. Each time the USB sending or receiving interrupt is processed, the synchronization trigger bit of corresponding host endpoint should be modified correctly to synchronize the data packet sent next time and detect whether the data packet received next time is synchronous. In addition, RB_UH_T_AUTO_TOG and RB_UH_R_AUTO_TOG can be set to automatically flip the corresponding synchronization trigger bit after sending or receiving successfully.

USB host token setting register (R8_UH_EP_PID) is used to set the endpoint number of the target device being operated and the token PID packet identification of the USB transmission transaction. The data corresponding to the SETUP token and OUT token is provided by the host transmission endpoint. The data to be sent is in the R32_UH_TX_DMA buffer, and the length of the data to be sent is set in R16_UH_TX_LEN. The data corresponding to the IN token is returned by the target device to the host reception endpoint, the data received is stored in the R32_UH_RX_DMA buffer, and the length of data received is stored in R8_USB_RX_LEN.

Table 16-6 USB host registers (Those marked in grey are controlled by RB_UC_RESET_SIE reset)

Name	Access address	Description	Reset value
R8_UHOST_CTRL	0x40008001	USB host physical port control register	0xX0
R8_UH_EP_MOD	0x4000800d	USB host endpoint mode control register	0x00
R16_UH_RX_DMA	0x40008018	USB host receive buffer start address	0xXXXX
R16_UH_TX_DMA	0x4000801c	USB host transmit buffer start address	0xXXXX
R8_UH_SETUP	0x40008026	USB host auxiliary setting register	0x00
R8_UH_EP_PID	0x40008028	USB host token setting register	0x00
R8_UH_RX_CTRL	0x4000802a	USB host reception endpoint control register	0x00
R8_UH_TX_LEN	0x4000802c	USB host transmission length register	0xXX
R8_UH_TX_CTRL	0x4000802e	USB host transmission endpoint control register	0x00

USB Host Physical Port Control Register (R8_UHOST_CTRL)

Bit	Name	Access	Description	Reset value
7	RB_UH_PD_DIS	RW	Internal pull-down resistor control bit of UD+/UD- pin of USB host port: 1: Disable internal pull-down; 0: Enable internal pull-down. It can be used in GPIO mode to provide pull-down resistor.	1
6	Reserved	RO	Reserved	0
5	RB_UH_DP_PIN	RO	Current UD + pin status: 1: High level; 0: Low level.	X
4	RB_UH_DM_PIN	RO	Current UD- pin status: 1: High level; 0: Low level.	X
3	Reserved	RO	Reserved	0
2	RB_UH_LOW_SPEED	RW	USB host port low-speed mode enable bit: 1: Select 1.5Mbps low-speed mode; 0: Select 12Mbps full-speed mode.	0
1	RB_UH_BUS_RESET	RW	USB host mode bus reset: 1: Output USB bus reset by force; 0: End output.	0
0	RB_UH_PORT_EN	RW	USB host port enable: 1: Enable the host port; 0: Disable the host port. The bit is automatically cleared to 0 when the USB device is disconnected.	0

USB Host Endpoint Mode Control Register (R8_UH_EP_MOD)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved	0
6	RB_UH_EP_TX_EN	RW	Host transmission endpoint transmit (SETUP/OUT) enable:	0

			1: Enable endpoint transmission; 0: Disable endpoint transmission.	
5	Reserved	RO	Reserved	0
4	RB_UH_EP_TBUF_MOD	RW	Host transmission endpoint transmit data buffer mode control.	0
3	RB_UH_EP_RX_EN	RW	Host reception endpoint reception (IN) enable: 1: Enable endpoint reception; 0: Disable endpoint reception.	0
[2:1]	Reserved	RO	Reserved	00b
0	RB_UH_EP_RBUF_MOD	RW	USB host reception endpoint reception data buffer mode control.	0

The host transmission endpoint data buffer modes are controlled by a combination of RB_UH_EP_TX_EN and RB_UH_EP_TBUF_MOD, refer to the following table.

Table 16-7 Host transmit buffer modes

RB_UH_EP_TX_EN	RB_UH_EP_TBUF_MOD	Description: Take R16_UH_TX_DMA as start address
0	X	Endpoint is disabled, and R16_UH_TX_DMA buffer is not used.
1	0	Single 64-byte transmit buffer (SETUP/OUT).
1	1	Double 64-byte transmit buffer, selected by RB_UH_T_TOG: When RB_UH_T_TOG=0, select the first 64-byte buffer; When RB_UH_R_TOG=1, select the last 64-byte buffer.

Table 16-8 Host receive buffer modes

RB_UH_EP_RX_EN	RB_UH_EP_RBUF_MOD	Description: Take R16_UH_TX_DMA as start address
0	X	Endpoint is disabled, and the R16_UH_RX_DMA buffer is not used.
1	0	Single 64-byte receive buffer (IN).
1	1	Double 64-byte receive buffer, selected by RB_UH_R_TOG: When RB_UH_R_TOG=0, select the first 64-byte buffer; When RB_UH_R_TOG=1, select the last 64-byte buffer.

USB Host Receive Buffer Start Address (R16_UH_RX_DMA)

Bit	Name	Access	Description	Reset value
[15:0]	R16_UH_RX_DMA	RW	Host endpoint data receive buffer start address. The lower 15 bits are valid, and the address must be 4 bytes aligned.	XXXXh

USB Host Transmit Buffer Start Address (R16_UH_TX_DMA)

Bit	Name	Access	Description	Reset value
[15:0]	R32_UH_TX_DMA	RW	Host endpoint data transmit buffer start address. The lower 15 bits are valid, and the address must be 4 bytes aligned.	XXXXh

USB Host Auxiliary Setting Register (R8_UH_SETUP)

Bit	Name	Access	Description	Reset value
7	RB_UH_PRE_PID_EN	RW	Low-speed preamble packet PRE PID enable bit: 1: Enable, used to communicate with low-speed USB device through an external HUB. 0: Disable the low-speed preamble packet.	0
6	RB_UH_SOF_EN	RW	Automatically generate SOF packet enable bit: 1: The host automatically generates SOF packet; 0: The host does not automatically generate SOF packet, but can generate manually.	0
[5:0]	Reserved	RO	Reserved	000000b

USB Host Token Setting Register (R8_UH_EP_PID)

Bit	Name	Access	Description	Reset value
[7:4]	MASK_UH_TOKEN	RW	Set the token PID packet identification of this USB transmission transaction.	0000b
[3:0]	MASK_UH_ENDP	RW	Set the endpoint number of the target device being operated this time.	0000b

USB Host Reception Endpoint Control Register (R8_UH_RX_CTRL)

Bit	Name	Access	Description	Reset value
7	RB_UH_R_TOG	RW	Synchronization trigger bit expected by USB host receiver (process IN transaction): 1: Expect DATA1; 0: Expect DATA0.	0
[6:5]	Reserved	RO	Reserved	00b
4	RB_UH_R_AUTO_TOG	RW	Synchronization trigger bit auto toggle enable control bit: 1: After the data is successfully received, the corresponding expected synchronization trigger bit (RB_UH_R_TOG) is automatically toggled; 0: It is not toggled automatically, but can be switched manually.	0
3	Reserved	RO	Reserved	0

2	RB_UH_R_RES	RW	Control on response to IN transactions by host receiver: 1: No response, used for real-time/synchronous transmission of non-endpoint 0; 0: Respond to ACK.	0
[1:0]	Reserved	RO	Reserved	00b

USB Host Transmission Length Register (R8_UH_TX_LEN)

Bit	Name	Access	Description	Reset value
[7:0]	R8_UH_TX_LEN	RW	Set the number of data bytes that USB host transmission endpoint is ready to send.	XXh

USB Host Transmission Endpoint Control Register (R8_UH_TX_CTRL)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved	0
6	RB_UH_T_TOG	RW	Synchronization trigger bit prepared by USB host transmitter (process SETUP/OUT transactions): 1: Transmit DATA1; 0: Transmit DATA0.	0
5	Reserved	RO	Reserved	0
4	RB_UH_T_AUTO_TOG	RW	Synchronization trigger bit auto toggle enable control bit: 1: The corresponding synchronization trigger bit (RB_UH_T_TOG) is toggled after the data is sent successfully; 0: It is not toggled automatically, but can be switched manually.	0
[3:1]	Reserved	RO	Reserved	000b
0	RB_UH_T_RES	RW	Response control bit of USB host transmitter to SETUP/OUT transaction: 1: Expect no response, used for real-time/synchronous transmission of non-endpoint0; 0: Expect to respond to ACK.	0

Chapter 17 Reserved

Chapter 18 Wireless Communication

18.1 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. It supports GFSK (Gaussian frequency shift keying) digital modulation and demodulation, and the data transmission rate can be adjusted. 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.
- Support GFSK digital modulation and demodulation.
- Support 2Mbps, 1Mbps.
- Support the highest 8KHz reporting rate in 2.4G mode.
- -95dBm RX sensitivity.
- Programmable TX power from -25dBm to +7.5dBm, and dynamic adjustment is supported.
- Single-ended RF interface simplifies board-level design.
- Support AES-128 hardware encryption/decryption.
- Support DMA.
- Optimized protocol stack and application layer API, and support networking.

18.2 2.4GHz Module

18.2.1 2.4GHz Module Functional Characteristics

- Normal and Enhanced modes are supported.
- Address length is configurable.
- CRC is configurable: length 0 to 2 bytes, polynomial and different-or operation are configurable.
- Byte order is configurable: bit order MSB/LSB of data is configurable.
- Wireless rate support 1Mbps, 2Mbps.
- Received modulation index range: 0.3 to 0.9.

18.2.2 2.4GHz Module Data Frame Format

The data frame format of the 2.4GHz module supports normal mode and enhanced mode. When set to normal mode, the data length is fixed; when set to enhance mode, the data length is variable due to the addition of control words, and includes PID and a response flag.

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

18.3 LLE Module

LLE module supports automatic sending and receiving mode and manual sending and receiving mode. 5 groups of independent hardware timers can control the time point of sending and receiving data in any one process.

18.4 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.

18.5 BB Module

18.6 AES Module

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

Chapter 19 Parameters

19.1 Absolute Maximum Ratings

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

Table 19-1 Absolute maximum ratings

Name	Parameter description	Min.	Max.	Unit
TA	Ambient temperature during operation	-40	85	℃
TS	Ambient temperature during storage	-55	125	℃
V5	Internal voltage regulator LDO5V input power supply voltage	-0.4	5.5	V
VDD33	Operating supply voltage (VDD33 connects to power, GND connects to ground)	-0.4	4.0	V
VIO33	I/O supply voltage (VIO33 connects to power, GND connects to ground)	-0.4	4.0	V
VIO	Voltage on input/output pins	-0.4	VIO33+0.4	V
VIO5	Voltage on input/output pins that support 5V withstand voltage	-0.4	5.5	V
VXCK	Voltage on XI/XO pin	-0.3	1.4	V

19.2 Electrical Characteristics

(Test conditions: TA = 25℃, V5 = 5V or 3.3V, VDD33 = VIO33 = 3.3V, Fsys = 16MHz)

Table 19-2 Electrical characteristics

Name	Parameter description			Min.	Typ.	Max.	Unit
V5	Internal regulator LDO5V input power supply voltage @V5			4.5	5	5.3	V
VDD33	Operating power supply voltage@VDD33			2.0	3.3	3.6	V
	Internal regulator LDO5V output voltage @VDD33			3.2	3.3	3.4	V
VIO33	I/O supply voltage @VIO33			2.0	3.3	3.6	V
ICC_8	Static power supply current condition: Code runs in RAM	Fsys=8M	RAM		3.0		mA
			Flash		3.3		
ICC_{16}		Fsys=16M	RAM		3.3		mA
			Flash		3.4		
ICC_{48}		Fsys=60M	RAM		5.6		mA
			Flash		8.6		
VIL	GPIO low level input voltage			0		0.9	V
VIH	GPIO high level input voltage			2.0		VIO33	V
VIL5	GPIO low-level input voltage supporting 5V			0		0.9	V

	tolerant voltage.				
VIH5	GPIO high-level input voltage supporting 5V tolerant voltage.	2.0		5.0	V
VOL	Low level output voltage (20mA sink current)	0	0.3	0.5	V
VOH	High level output voltage (20mA output current)	VIO33-0.5	VIO33-0.3	VIO33	V
IIN	Input current of GPIO floating input terminal	-3	0	3	uA
IUP	Input current of input terminal of GPIO built-in pull-up resistor	25	60	90	uA
IDN	Input current of input terminal of GPIO built-in pull-down resistor	-90	-60	25	uA
Vlvr	Threshold voltage for LVR low voltage reset	1.5	1.7	1.9	V

19.3 Power Consumption in Low-power Modes

(Test conditions: TA=25℃, V5 = 3.3V or floating, VDD33=VIO33=3.3V, Fsys=16MHz.)

Table 19-3 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		1.7		mA
Halt mode, disable all clocks		420		uA
Sleep mode, multiple combinations, refer to Table 5-3		0.46~1.2		uA
Sleep mode, PMU+core+RAM12K, GPIO wakes up, no RTC		1.2		uA
Shutdown mode, several combinations, refer to Table 5-3		0.3~0.9		uA
Shutdown mode, only PMU, reset after GPIO wakes up, no RTC		0.3		uA
Minimum current under single 5V power supply.	1.3			mA

Table 19-4 Current on modules (for reference only, related to temperature)

Name	Parameter description		Min.	Typ.	Max.	Unit
$I_{DD(RAM12K)}$	RAM12K: 12KB SRAM			0.3		uA
$I_{DD(LSI)}$	Internal LSI oscillator			0.3		uA
$I_{DD(HSE)}$	External HSE oscillator		130	170	260	uA
$I_{DD(BM)}$	Low-power consumption battery low voltage monitoring BM module			0.9		uA
$I_{DD(PLL)}$	Internal PLL oscillator			130		uA
$I_{DD(CMP)}$	CMP module			30		uA
$I_{DD(KEYSCAN)}$	KEYSCAN module			1.3		uA
$I_{DD(USB)}$	USB module	Non-transmit status		0.6		mA
		Transmit status		2.1		mA
$I_{DD(BLE)}$	BLE	Receive		7.1		mA
		-25dBm Transmission power		3.7		mA

		0dBm Transmission power		10.6		mA
		+7.5dBm Transmission power		21.5		mA

19.4 Clock Source

Table 19-5 High-speed oscillator (HSE)

Symbol	Parameter description	Min.	Typ.	Max.	Unit
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 19-6 Low-speed oscillator (LSI and LSE)

Symbol	Parameter description	Min.	Typ.	Max.	Unit
F_{LSIR}	Internal LSI oscillator frequency	24		42	KHz
T_{SULSI}	External LSE oscillator startup to stabilization time		40	100	uS

Table 19-7 PLL characteristics

Symbol	Parameter description	Min.	Typ.	Max.	Unit
F_{PLL}	Output clock after PLL (CK32M * 18.75)		600		MHz
T_{PLLLK}	PLL lock time		20	40	uS

19.5 Timing Parameters

(Test conditions: TA = 25℃, V5 = 5V or 3.3V, VDD33 = VIO33 = 3.3V, Fsys = 6.4MHz.)

Table 19-8 Timing parameters

Symbol	Parameter description		Min.	Typ.	Max.	Unit
T_{rpor}	Reset delay after RPOR		12	13	14.5	mS
T_{V5R}	V5 supply voltage rising rate		1		10000	us/V
T_{rst}	RST# valid signal width			100		nS
T_{mr}	Reset delay after MR		2	8	18	uS
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 low-power status	0.6	1	3	uS	us
		T_{SUHSE}+1	T_{SUHSE}+80	T_{SUHSE}+150	uS	us
		T_{SUHSE}+1	T_{SUHSE}+300	T_{SUHSE}+400	uS	us
		T_{SUHSE}+0.4	T_{SUHSE}+1	T_{SUHSE}+5	mS	ms

Note: The delay parameters in the table are all based on multiples of Tsys, 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}).

19.6 CMP Parameters

(Test conditions: TA = 25℃, V5 = 5V or 3.3V, VDD33 = VIO33 = 3.3V, Fsys = 16MHz.)

Table 19-9 CMP parameters

Symbol	Parameter description	Min.	Typ.	Max.	Unit
V_{CMIR}	CMP common-mode input voltage range	0		VDD33-0.8	V
V_{OSIN}	CMP input offset voltage			±9	mV
T_s	CMP output settling time (1)		1	6	uS

Note: 1. When the voltages at the positive and negative terminals of the comparator are very close, the output stability time is close to the maximum; Under normal conditions, the output stability time is within the typical value.

19.7 Other Parameters

(Test conditions: TA = 25℃, V5 = 5V or 3.3V, VDD33 = VIO33 = 3.3V, Fsys = 16MHz.)

Table 19-10 Other parameters

Symbol	Parameter description		Min.	Typ.	Max.	Unit
T_{FRER}	Single sector erase operation time of Flash-ROM		1	3	8	mS
T_{FRPG}	Single word program operation time of Flash-ROM		0.1	1	2	mS
N_{EPCE}	Erase/program cycle endurance	5~45℃	100K	2000K (Random test)		times
		-40~85℃	50K	500K (Random test)		
T_{DR}	Data hold capability of Flash-ROM		20			years
V_{ESD}	ESD tolerant voltage on I/O input or output pins	Antenna (ANT)	2K	4K (Random test)		V
		I/O pins: PA	4K	6K (Random test)		V

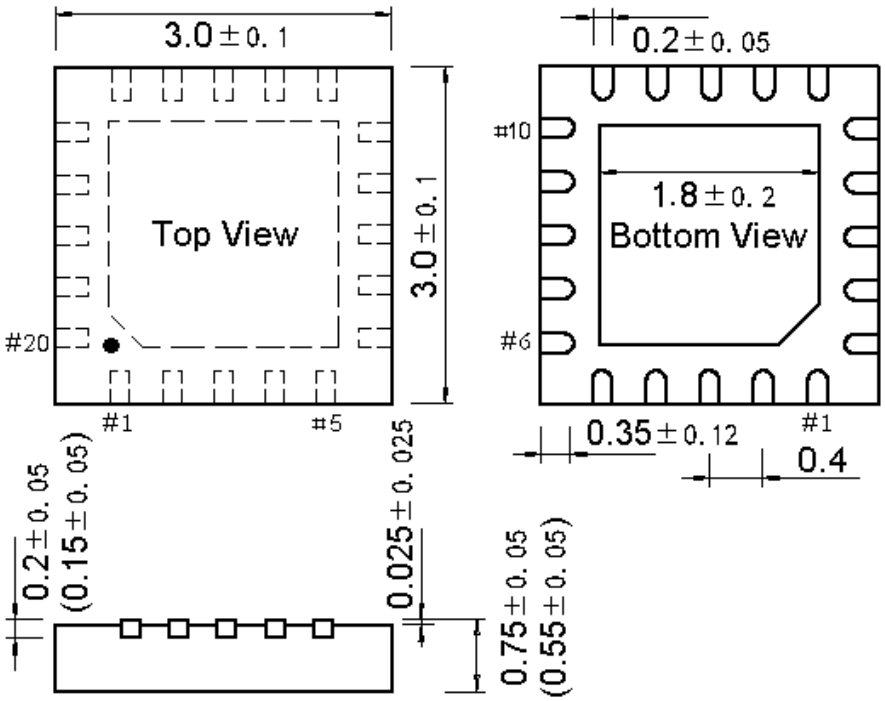
Chapter 20 Package

Chip Package

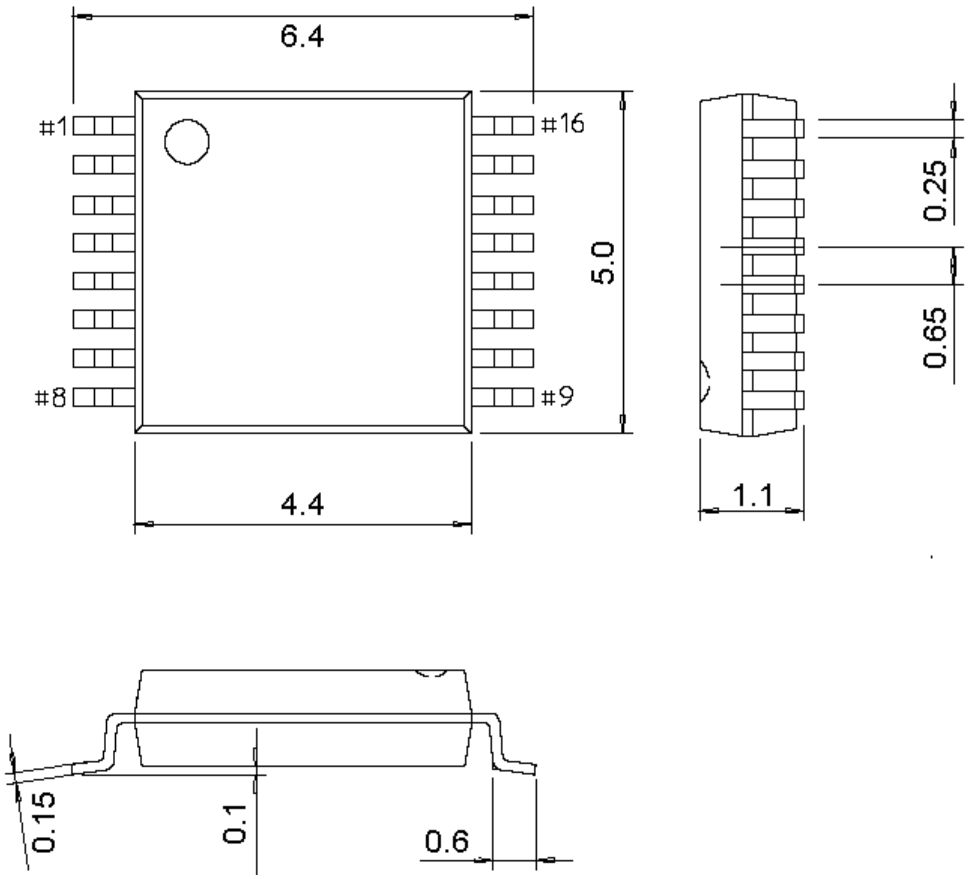
Package Form	Body Size	Pin Pitch		Package Description	Order Model
QFN20	3*3mm	0.4mm	15.7mil	Quad Flat No-lead Package	CH572D
TSSOP16	4.4*5.0mm	0.65mm	25.6mil	Thin Shrink Small Outline Package	CH572R
DFN10X3	3*3mm	0.5mm	19.7mil	Dual Flat No-lead Package	CH572Q
QFN20	3*3mm	0.4mm	15.7mil	Quad Flat No-lead Package	CH570D
DFN10X3	3*3mm	0.5mm	19.7mil	Dual Flat No-lead Package	CH570Q
SOP8	3.9*5.0mm	1.27mm	50mil	Small Out-Line Package	CH570E

Note: All dimensions are in millimeters. The pin center spacing values are nominal values, and the error of other dimensions is not more than ±0.2mm.

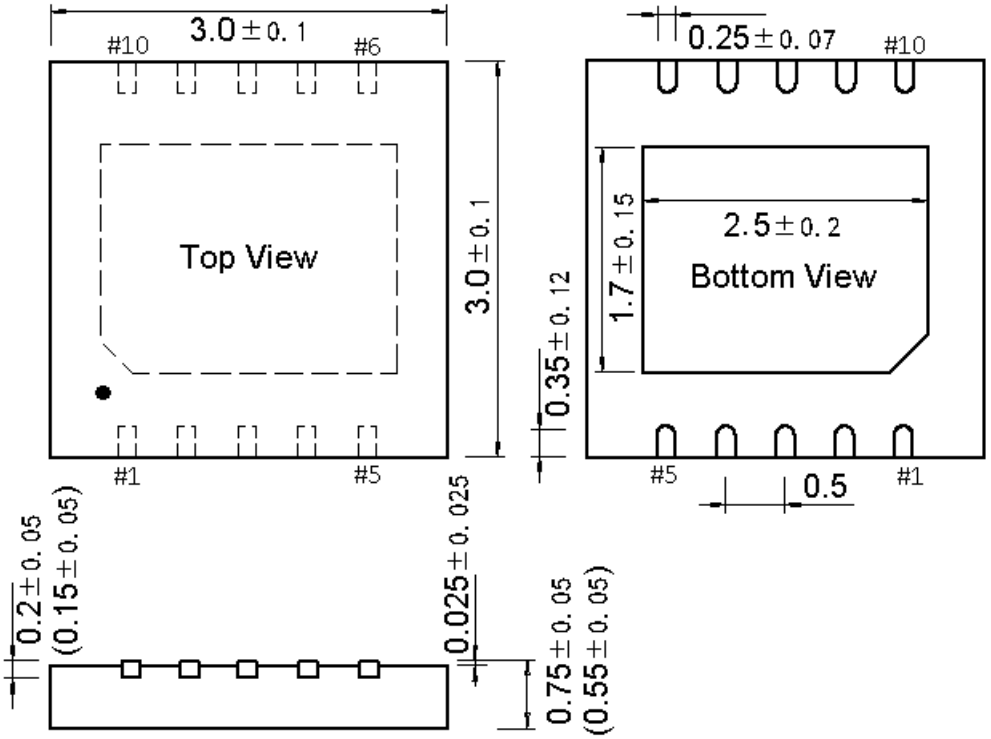
QFN20



TSSOP16



DFN10X3


3.0±0.1
#10
#6
Top View
3.0±0.1
#1
#5
0.2±0.05
(0.15±0.05)
0.025±0.025
0.75±0.05
(0.55±0.05)
0.25±0.07
#10
1.7±0.15
2.5±0.2
Bottom View
0.35±0.12
#5
0.5
#1

SOP8