# CH579 Data sheet

V1.6

## Overview

CH579 is an ARM core 32-bit microcontroller integrated with BLE wireless communication. The chip integrates rich peripheral resources such as Bluetooth low energy BLE communication module, Ethernet controller and transceiver, full-speed USB host and device controller and transceiver, segment LCD driver module, ADC, touch button detection module, RTC and so on.

## Features

- Core:
  - 32-bit ARM Cortex-M0 core
  - The highest 40MHz system frequency, the lowest 32KHz frequency
- 257K bytes of non-volatile storage FlashROM:
  - 250KB user application storage area CodeFlash
  - 2KB user non-volatile data storage area DataFlash
  - 4KB system boot program storage area BootLoader
  - 1KB system non-volatile configuration information storage area InfoFlash
  - Support ICP, ISP and IAP, support OTA wireless upgrade
- 32K bytes of volatile data storage SRAM:
  - 16KB RAM with main power supply only 16K
  - 14KB dual power supply sleep-maintaining memory area RAM 14K
  - 2KB dual power supply sleep-maintaining memory area RAM2K
- Power management and low power consumption:
  - Support 3.3V and 2.5V power supply, range 2.1V ～3.6V
  - Built-in DC-DC conversion, 0dBm transmission power current
  - 6mA Idle mode Idle: 1.2mA
  - Pause mode Halt: 420uA
  - Sleep mode: 0.6uA ～2.0uA multiple gears
  - Shutdown in power-off mode: 0.2uA ～1.3uA multiple gears
  - Battery voltage low-voltage monitoring
- Security features: AES-128 encryption and decryption, chip unique ID
- Bluetooth Low Energy BLE:
  - Integrated 2.4GHz RF transceiver and baseband and link control
  - single-ended RF interface, no external inductor is required, simplifying board-level design
  - Receiving sensitivity -93dBm, programmable +3dBm transmit power
  - BLE complies with Bluetooth Low Energy 4.2 specification. The
  - wireless communication distance is about 100 meters at 0dBm
  - transmission power. Various shapes and sizes of PCB on-board antennas are available
  - Provide optimized protocol stack and application layer API, support networking
- Ethernet Ethernet:
  - Built-in MAC controller, support frame filtering, support DMA integrated
  - 10Mbps transceiver PHY, built-in 50Ω matching resistor, communication
  - distance of 200 meters, support auto-negotiation, support energy saving
- Universal Serial Bus USB:
  - Built-in USB controller and DMA, support 64-byte data packets,
  - integrated USB 2.0 full-speed transceiver PHY, no peripherals,
  - support full/low-speed Host host and Device device mode, support
  - USB type-C master-slave/current detection
- Real-time clock RTC: Support timing and trigger two modes
- Segment LCD: Support 96 dots (24×4) LCD panel
- Analog-to-digital conversion ADC:
  - 12-bit analog-to-digital converter, supporting differential and single-ended input
  - 14 external analog signal channels and 2 internal signals
- Touch key detection module TouchKey: 14 channels
- Timer Timer and pulse width modulation PWM:
  - 4 groups of 26-bit timers, 16MHz main frequency timing up to 4.2S, 4
  - channels of capture/sampling, support rising edge/falling edge/double edge
  - 4 channels of 26-bit PWM output, 8 channels of 8-bit PWM output
- Asynchronous serial port UART:
  - 4 independent UARTs, compatible with 16C550, built-in 8-level FIFO
  - 23-bit counter, the highest communication baud rate can reach 5Mbps
  - UART0 supports Modem and hardware automatic flow control
  - UART0 supports automatic matching of slave address during multi-machine communication
- Serial peripheral interface SPI:
  - 2 groups of independent SPI, built-in FIFO
  - SCK serial clock frequency can reach half of the system frequency
  - SPI0 supports Master and Slave modes, and supports DMA
- LED dot matrix screen interface: Support 1/2/4 data lines
- clock: Built-in 32MHz and 32KHz clock, built-in PLL
- 8-bit passive parallel port
- Temperature sensor TS
- General-purpose input and output port GPIO:
  - 40 GPIOs, 4 of which support 5V signal input, 32
  - interrupt inputs, 32 wake-up inputs
- Package form: QFN48_5X5, QFN28_4X4

Chapter 1 Pin Information

## 1.1 Pinout

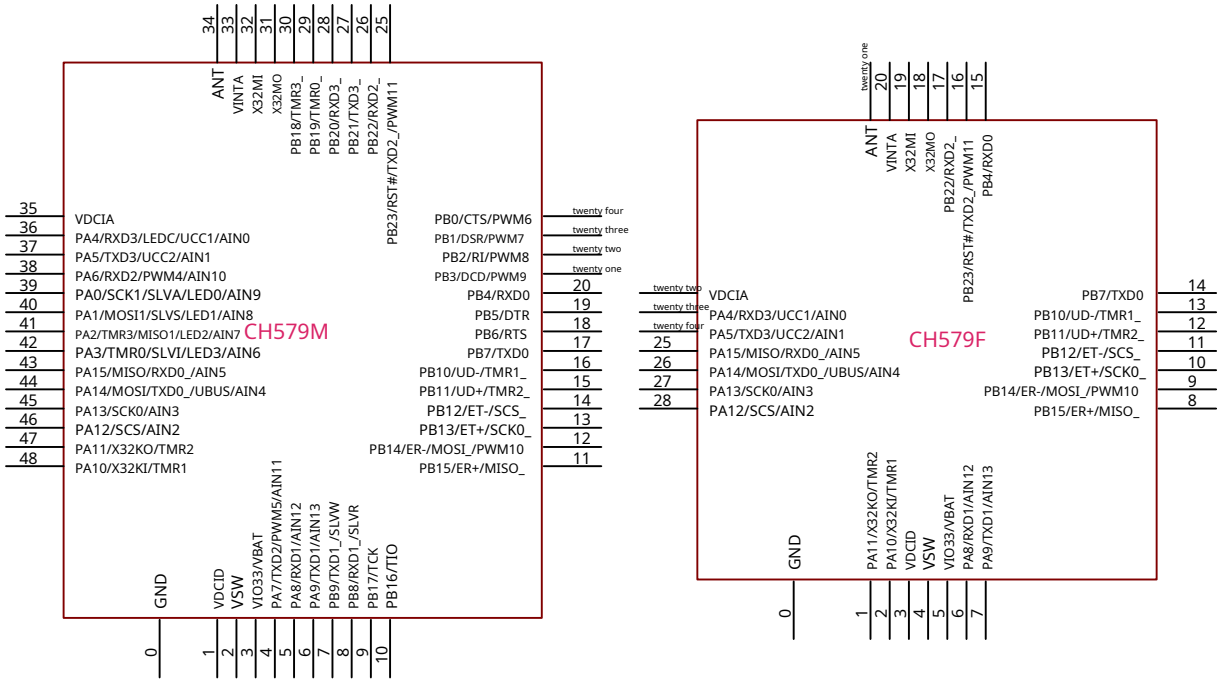



Figure 1-1 CH579M (QFN48_5X5) and CH579F (QFN28_4X4) package pin arrangement

### 1.2 Pin description

| QFN48 Pin No. | QFN28 Pin No. | Pin Name | Pin multiplexing function Type | first | Function description |
|---|---|---|---|---|---|
| 0 | 0 | GND | power supply | - | Common ground terminal, voltage 0V reference point. |
| 1 | 3 | VDCID | power supply | - | The power input of the internal digital circuit LDO regulator requires an external decoupling capacitor. 2.2uF is recommended when DC-DC is enabled, and not less than 0.1uF when not enabled. The |
| 2 | 4 | VSW | power supply | - | internal DC-DC switch output, when enabling DC-DC, it must be connected to the pin in series with 22uH or 33uH inductance connected to VDCID. When DC-DC is not enabled, VDCID can be connected directly. |
| 3 | 5 | VIO33 | power supply | VBAT | I/O and DC-DC or battery power input should be close to the pin to connect an external decoupling capacitor. When DC-DC is enabled, 2.2uF is recommended, and when DC-DC is not enabled, it is recommended not to be less th |
| 4 | no | PA7 | I/O/A | TXD2 /PWM5 /AIN11 | PA7: General purpose bidirectional digital I/0 pin. TXD2: UART2 serial data output. PWM5: Pulse width modulation output channel 5. AIN11: ADC analog signal input channel 11. |
| 5 | 6 | PA8 | I/O/A | RXD1 /AINT12 | PA8: General purpose bidirectional digital I/0 pin. RXD1: UART1 serial data input. AIN12: ADC analog signal input channel 12. |
| 6 | 7 | PA9 | I/O/A | TXD1 /AINT13 | PA9: General purpose bidirectional digital I/0 pin. TXD1: UART1 serial data output. AIN13: ADC analog signal input channel 13. |
| 7 | no | PB9 | I/O | TXD1_ /SLVW | PB9: General purpose bidirectional digital I/0 pin. TXD1_: TXD1 pin mapping of UART1. |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | SLVW: Passive parallel port write control signal input, active low. |
| 8 | no | PB8 | I/O | RXD1_ /SLVR | PB8: General purpose bidirectional digital I/0 pin.<br>RXD1_: RXD1 pin mapping of UART1.<br>SLVR: Passive parallel port read control signal input, active at low level. |
| 9 | no | PB17 | I/O | TCK | PB17: General purpose bidirectional digital I/0 pin.<br>TCK: Serial clock input of the two-wire emulation debug interface. |
| 10 | no | PB16 | I/O | TIO | PB16: General purpose bidirectional digital I/0 pin.<br>TIO: Serial data input and output of two-wire simulation debugging interface, built-in pull-up. |
| 11 | 8 | PB15 | I/O/A /5VT | ER+ /MISO_ | PB15: General purpose bidirectional digital I/0 pin.<br>ER+: Ethernet receives RX+ signal.<br>MISO_: MISO pin mapping of SPI0. |
| 12 | 9 | PB14 | I/O/A /5VT | ER- /MOSI_ /PWM10 | PB14: General purpose bidirectional digital I/0 pin.<br>ER-: Ethernet receives RX- signal.<br>MOSI_: MOSI pin mapping of SPI0. PWM10:<br>Pulse width modulation output channel 10. |
| 13 | 10 | PB13 | I/O/A /5VT | ET+ /SCK0_ | PB13: General purpose bidirectional digital I/0 pin.<br>ET+: Ethernet sends TX+ signal.<br>SCK0_: SCK pin mapping of SPI0. PB12: |
| 14 | 11 | PB12 | I/O/A /5VT | ET- /SCS_ | General purpose bidirectional digital I/0 pin.<br>ET-: Ethernet sends TX- signal.<br>SCS_: SCS pin mapping of SPI0. PB11: General |
| 15 | 12 | PB11 | I/O/A | UD+ /TMR2_ | purpose bidirectional digital I/0 pin.<br>UD+: D+ data line of USB bus.<br>TMR2_: Timer 2 TMR2 pin mapping. PB10: |
| 16 | 13 | PB10 | I/O/A | UD- /TMR1_ | General purpose bidirectional digital I/0 pin.<br>UD-: D-data line of USB bus.<br>TMR1_: TMR1 pin mapping of timer 1. PB7: |
| 17 | 14 | PB7 | I/O | TXD0 | General purpose bidirectional digital I/0 pin.<br>TXD0: UART0 serial data output. |
| 18 | no | PB6 | I/O | RTS | PB6: General purpose bidirectional digital I/0 pin.<br>RTS: MODEM output signal of UART0, request to send. PB5: |
| 19 | no | PB5 | I/O | DTR | General purpose bidirectional digital I/0 pin.<br>DTR: MODEM output signal of UART0, the data terminal is ready. PB4: |
| 20 | 15 | PB4 | I/O | RXD0 | General purpose bidirectional digital I/0 pin.<br>RXD0: UART0 serial data input. |
| twenty one | no | PB3 | I/O | DCD /PWM9 | PB3: General purpose bidirectional digital I/0 pin.<br>DCD: MODEM input signal of UART0, carrier detection. PWM9: Pulse<br>width modulation output channel 9. |
| twenty two | no | PB2 | I/O | RI /PWM8 | PB2: General purpose bidirectional digital I/0 pin.<br>RI: MODEM input signal of UART0, ringing indicator. PWM8:<br>Pulse width modulation output channel 8. |
| twenty three | no | PB1 | I/O | DSR /PWM7 | PB1: General purpose bidirectional digital I/0 pin.<br>DSR: MODEM input signal of UART0, the data device is ready. PWM7: Pulse<br>width modulation output channel 7. |
| twenty four | no | PB0 | I/O | CTS /PWM6 | PB0: General purpose bidirectional digital I/0 pin.<br>CTS: MODEM input signal of UART0, clear to send. PWM6: Pulse<br>width modulation output channel 6. |
| 25 | 16 | PB23 | I/O | RST# /TXD2_ | PB23: General purpose bidirectional digital I/0 pin.<br>RST#: External reset input, active low, built-in pull-up resistor. |

| | | | | | |
|---|---|---|---|---|---|
| | | | | /PWM11 | TXD2_: TXD2 pin mapping of UART2. PWM11:<br>Pulse width modulation output channel 11. |
| 26 | 17 | PB22 | I/O | RXD2_ | PB22: General purpose bidirectional digital I/0 pin.<br>RXD2_: RXD2 pin mapping of UART2. PB21: |
| 27 | no | PB21 | I/O | TXD3_ | General purpose bidirectional digital I/0 pin.<br>TXD3_: TXD3 pin mapping of UART3. PB20: |
| 28 | no | PB20 | I/O | RXD3_ | General purpose bidirectional digital I/0 pin.<br>RXD3_: RXD3 pin mapping of UART3. PB19: |
| 29 | no | PB19 | I/O | TMR0_ | General purpose bidirectional digital I/0 pin.<br>TMR0_: TMR0 pin mapping of Timer 0. PB18: |
| 30 | no | PB18 | I/O | TMR3_ | General purpose bidirectional digital I/0 pin.<br>TMR3_: Timer 3 TMR3 pin mapping. |
| 31 | 18 | X32MO | I/A | - | The inverted output terminal of the high frequency oscillator HSE is connected to one end of a |
| 32 | 19 | X32MI | A | - | 32MHz crystal. The input end of the high frequency oscillator HSE is connected to the other end of a |
| 33 | 20 | VINTA | power supply | - | 32MHz crystal. The power node of the internal analog circuit needs to be close to the pin and an external decoupling capacitor<br>2.2uF (1uF can be selected when DC-DC is not used, which saves power but reduces BLE sensitivity). RF |
| 34 | twenty one | ANT | A | - | signal input and output, it is recommended to connect the antenna directly. |
| 35 | twenty two | VDCIA | power supply | - | The power input of the internal analog circuit LDO regulator requires an external decoupling capacitor.<br>It is recommended not to be less than 0.1uF, and it is recommended to connect directly to the VDCID. |
| 36 | twenty three | PA4 | I/O/A | RXD3<br>/LEDC<br>/UCC1<br>/AIN0 | PA4: General purpose bidirectional digital I/0 pin.<br>RXD3: UART3 serial data input.<br>LEDC: LED screen interface serial clock output.<br>UCC1: USB type-C bidirectional configuration<br>channel 1. AIN0: ADC analog signal input channel 0. |
| 37 | twenty four | PA5 | I/O/A | TXD3<br>/UCC2<br>/AIN1 | PA5: General purpose bidirectional digital I/0 pin.<br>TXD3: UART3 serial data output.<br>UCC2: USB type-C bidirectional configuration<br>channel 2. AIN1: ADC analog signal input channel 1. |
| 38 | no | PA6 | I/O/A | RXD2<br>/PWM4<br>/AIN10 | PA6: General purpose bidirectional digital I/0 pin.<br>RXD2: UART2 serial data input.<br>PWM4: Pulse width modulation output channel 4.<br>AIN10: ADC analog signal input channel 10. |
| 39 | no | PA0 | I/O/A | SCK1<br>/SLVA<br>/LED0<br>/AIN9 | PA0: General purpose bidirectional digital I/0 pin.<br>SCK1: SPI1 serial clock output.<br>SLVA: Passive parallel port address input, high level selection command and status port,<br>Low level selects the data port.<br>LED0: LED screen interface serial data output 0.<br>AIN9: ADC analog signal input channel 9. |
| 40 | no | PA1 | I/O/A | MOSI1<br>/SLVS<br>/LED1<br>/AIN8 | PA1: General purpose bidirectional digital I/0 pin.<br>MOSI1: SPI1 serial data output.<br>SLVS: Passive parallel port chip select control input, active at low level.<br>LED1: LED screen interface serial data output 1.<br>AIN8: ADC analog signal input channel 8. |
| 41 | no | PA2 | I/O/A | TMR3<br>/MISO1<br>/LED2<br>/AIN7 | PA2: General purpose bidirectional digital I/0 pin.<br>TMR3: Capture input 3 and PWM output channel 3 of timer 3. MISO:<br>SPI1 serial data input, multiplexed for output in 2-wire mode.<br>LED2: LED screen interface serial data output 2.<br>AIN7: ADC analog signal input channel 7. |
| 42 | no | PA3 | I/O/A | TMR0 | PA3: General purpose bidirectional digital I/0 pin. |

| | | | | | |
|---|---|---|---|---|---|
| | | | | /SLVI /LED3 /AIN6 | TMR0: Capture input 0 and PWM output channel 0 of timer 0. SLVI: Passive parallel port interrupt request output, active at low level. LED3: LED screen interface serial data output 3. AIN6: ADC analog signal input channel 6. |
| 43 | 25 | PA15 | I/O/A | MISO /RXD0_ /AIN5 | PA15: General purpose bidirectional digital I/0 pin. MISO: SPI0 serial data pin, master input/slave output. RXD0_: RXD0 pin mapping of UART0. AIN5: ADC analog signal input channel 5. |
| 44 | 26 | PA14 | I/O/A | MOSI /TXD0_ /UBUS /AIN4 | PA14: General purpose bidirectional digital I/0 pin. MOSI: SPI0 serial data pin, master output/slave input. TXD0_: TXD0 pin mapping of UART0. UBUS: USB type-C bus voltage detection input. AIN4: ADC analog signal input channel 4. |
| 45 | 27 | PA13 | I/O/A | SCK0 /AIN3 | PA13: General purpose bidirectional digital I/0 pin. SCK0: SPI0 serial clock pin, master output/slave input. AIN3: ADC analog signal input channel 3. |
| 46 | 28 | PA12 | I/O/A | SCS /AIN2 | PA12: General purpose bidirectional digital I/0 pin. SCS: Chip select input in SPI0 slave mode, active at low level. AIN2: ADC analog signal input channel 2. |
| 47 | 1 | PA11 | I/O/A | X32KO /TMR2 | PA11: General purpose bidirectional digital I/0 pin. X32KO: The inverted output terminal of the low frequency oscillator, which is connected to one end of a 32KHz crystal. TMR2: Capture input 2 and PWM output channel 2 of timer 2. PA10: |
| 48 | 2 | PA10 | I/O/A | X32KI /TMR1 | General purpose bidirectional digital I/0 pin. X32KI: The input terminal of the low frequency oscillator, which is connected to the other end of the 32KHz crystal. TMR1: Capture input 1 and PWM output channel 1 of timer 1. |

Duplex for segment LCD Driven Pin description

| Pin name | Types of | Reuse function | Function description |
|---|---|---|---|
| PA0 ～PA3 | A | COM0 ～COM3 | To drive each common terminal of the segment LCD, select part or all according to the needs. |
| PB0 ～PB23 | A | SEG0 ～SEG23 | To drive each segment of the segment LCD, select part or all according to the needs. There are 6 bits RB_PIN_SEG*_IE in the R16_PIN_ANALOG_IE register. Each bit controls 4 segment pins at the same time. When this bit is 0, 4 pins are used for digital input or other non-LCD functions. When this bit is 1, 4 pins are used For LCD segment drive. |

Note :

(1), pin type:

I=TTL/CMOS level Schmitt input;

O=CMOS level tri-state output;

A=Analog signal input or output;

5VT=Support 5V signal voltage input.

(2) The multiplexing functions and mappings of pins are arranged in the table from high to bottom according to their priority, and the GPIO function is the lowest priority.

# Chapter 2 System Structure and Memory

## 2.1 System structure

The following figure shows the block diagram of CH579 chip system structure.

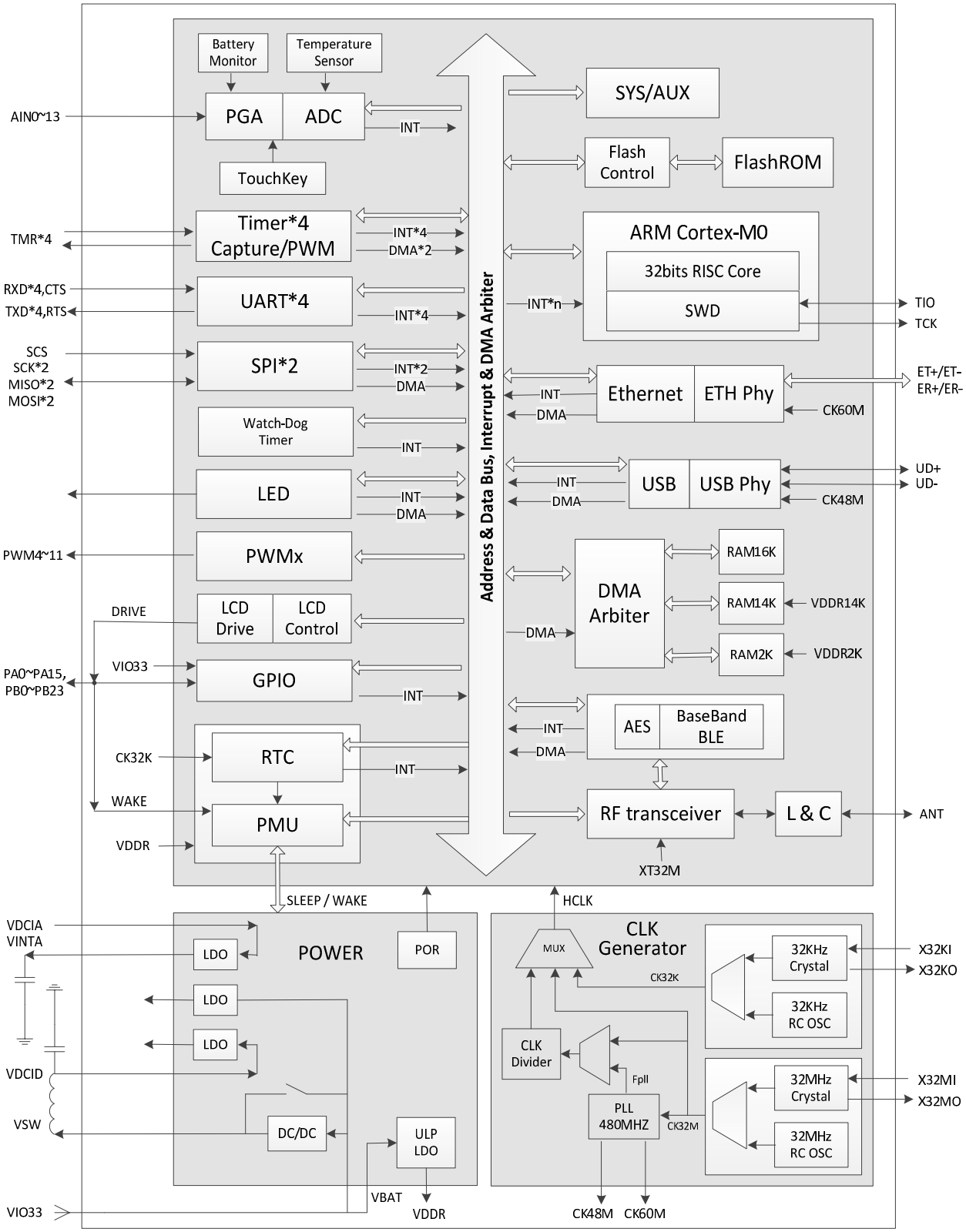



Figure 2-1 CH579 internal structure block diagram

## 2.2 Memory map

The addressing space of CH579 mainly includes several different areas such as CODE area/FlashROM, DATA area/SRAM, peripherals, etc., as shown in the figure below for details.

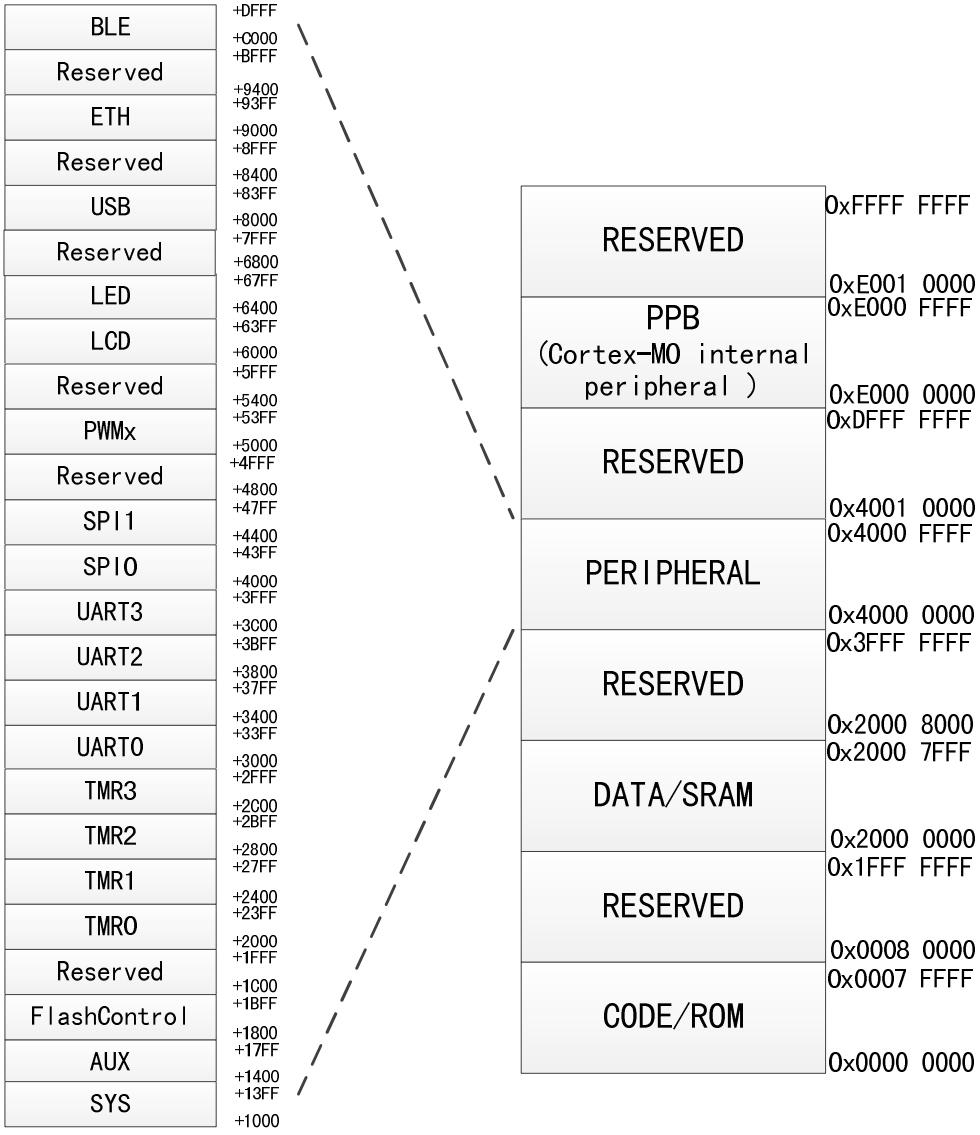



Figure 2-2 Memory map

## 2.3 Memory Map

The address range of each memory map area is shown in the following table:

Table 2-1 Memory Map Area Address

| Address range | use | description |
|---|---|---|
| 0x00000000-0x0007FFFF | On-chip CODE area, non-volatile memory | 257KB, FlashROM |
| 0x00080000-0x1FFFFFFF | Keep | - |
| 0x20000000-0x20007FFF | On-chip DATA area, volatile memory | 32KB, SRAM |
| 0x20008000-0x3FFFFFFF | Keep | - |
| 0x40000000-0x4000FFFF | Various peripherals | Multiple peripheral modules |
| 0x40010000-0xDFFFFFFF | Keep | - |
| 0xE0000000-0xE000FFFF | PPB | Internal peripheral bus area |
| 0xE0010000-0xFFFFFFFF | Keep | - |

2.3.1 On-chip CODE area mapping table

Table 2-2 CODE area address

| Address range | use | description |
|---|---|---|
| 0x00000000-0x0003E7FF | User application storage area CodeFlash | 250KB |
| 0x0003E800-0x0003EFFF | User non-volatile data storage area DataFlash | 2KB |
| 0x0003F000-0x0003FFFF | System boot program storage area BootLoader | 4KB |
| 0x00040000-0x000403FF | System non-volatile configuration information storage area InfoFlash | 1KB |
| 0x00040400-0x0007FFFF | Keep | - |

The configuration information of address 0x00040010 can be set by the user through the tool.

Table 2-3 User-level non-volatile configuration information description

| Bit address | name | use | Defaults |
|---|---|---|---|
| Bit 2 to bit 0 | RESERVED | Keep | 000b |
| Bit 3 | CFG_RESET_EN | RST# External manual reset input pin enable | 0 |
| Bit 4 | CFG_DEBUG_EN | Two-wire simulation debugging interface enable | 1 |
| Bit 5 | RESERVED | Keep | 0 |
| Bit 6 | CFG_BOOT_EN | BootLoader enable | 1 |
| Bit 7 | CFG_ROM_READ | Code and data protection mode in FlashROM:<br>0-Prohibit the programmer to read out, the program is confidential; 1-Allow to read out | 1 |
| Bit 27 to bit 8 | RESERVED | Keep | 0FFFFh |
| Bit 31 to bit 28 | VALID_SIG | Configuration information valid flag, fixed value | 0101b |

2.3.2 On-chip DATA area mapping table

Table 2-4 DATA area address

| Address range | use | description |
|---|---|---|
| 0x20000000-0x20003FFF | RAM 16K with main power supply only | 16KB |
| 0x20004000-0x200077FF | Main + auxiliary dual power supply, independently maintainable storage area RAM14K | 14KB |
| 0x20007800-0x20007FFF | Main + auxiliary dual power supply, independently maintainable storage area RAM2K | 2KB |

2.3.3 Peripheral address allocation

CH579 mainly contains the following peripherals. Each peripheral occupies a certain address space. The actual access address of the peripheral register is: Base Address + offset address. In the subsequent chapters, the address of the register is described in detail. The following table is the allocation table of the base address of each peripheral.

Table 2-5 Peripheral base address allocation table

| Peripheral number | Peripheral name | Peripheral base address |
|---|---|---|
| 1 | SYS (PMU/RTC/GPIO, etc.) | 0x4000 1000 |
| 2 | AUX (ADC/TKEY/PLL, etc.) | 0x4000 1400 |
| 3 | FlashROM-Control | 0x4000 1800 |
| 4 | TMR0 | 0x4000 2000 |
| 5 | TMR1 | 0x4000 2400 |
| 6 | TMR2 | 0x4000 2800 |
| 7 | TMR3 | 0x4000 2C00 |
| 8 | UART0 | 0x4000 3000 |
| 9 | UART1 | 0x4000 3400 |

| | | |
|---|---|---|
| 10 | UART2 | 0x4000 3800 |
| 11 | UART3 | 0x4000 3C00 |
| 12 | SPI0 | 0x4000 4000 |
| 13 | SPI1 | 0x4000 4400 |
| 14 | PWMx (PWM4 ～PWM11) | 0x4000 5000 |
| 15 | LCD | 0x4000 6000 |
| 16 | led | 0x4000 6400 |
| 17 | USB | 0x4000 8000 |
| 18 | ETH | 0x4000 9000 |
| 19 | Radio:BLE | 0x4000 C000<br>0x4000 D000 |

The following table is 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 a fixed value, not affected by reset. |
| RO | Read only. |
| WO | Write only, the read value is 0 or invalid. |
| RZ | Read only, automatically cleared to 0 after reading. |
| WZ | Clear to 0 when written. |
| RW | Read and write. |
| RW1 | Readable, write 1 to clear 0. |
| WA | In write-only and safety-only mode, the read value is 0 or invalid. |
| RWA | Readable, only writable in safe mode. |

The following table explains the abbreviations used in the subsequent chapters:

Table 2-7 Noun abbreviation description

| abbreviation | description |
|---|---|
| HSE | External high-frequency crystal oscillator clock source (32MHz recommended) |
| HSI | Internal high-frequency RC clock oscillator source (32MHz after factory |
| LSE | calibration) External low-frequency crystal oscillator clock source (32KHz recommended) |
| LSI | Internal low-frequency RC clock oscillator source (32KHz after calibration when the application |
| CK32M | software is running) High-frequency clock source (default 32MHz) |
| CK32K | Low frequency clock source (default 32KHz) |
| Fpll | PLL output clock (default frequency is 480MHz) |
| HCLK | System main frequency clock |
| Fsys | System main frequency clock frequency |
| Tsys | System main frequency clock cycle (1/Fsys) |
| RAM2K | 2KB SRAM with the highest address |
| RAM14K | 14KB SRAM with the next highest address |
| 0x | The data beginning with it represents a hexadecimal number |
| H | The data ending with it represents a hexadecimal number |
| B | The data ending with it represents a binary number |

# Chapter 3 Interrupts

## 3.1 Interrupt controller

The system supports up to 20 groups of interrupt signal sources, and each interrupt request has an independent trigger and mask control bit, as well as a dedicated status bit.

## 3.2 System tick SysTick calibration value

When the system tick clock is set to 32MHz and the system tick calibration value is fixed at 32000, a 1mS time base will be generated.

## 3.3 Interrupt and exception vector

The following table lists the vector table of the chip system

Table 3-1 Interrupt vector table

| position | priority level | priority Types of | name | Description | address |
|---|---|---|---|---|---|
| | - | - | - | Keep | 0x0000_0000 |
| -15 | -3 | fixed | Reset | Reset | 0x0000_0004 |
| -14 | - 2 | fixed | NMI | Non-maskable interrupt | 0x0000_0008 |
| -13 | - 1 | fixed | Hard fault | All types of failures | 0x0000_000C |
| | - | - | - | Keep | - |
| -5 | 0 | Can be set | SVCall | System service call via SWI instruction | 0x0000_002C |
| | - | - | - | Keep | - |
| - 2 | 1 | Can be set | PendSV | Suspend system services | 0x0000_0038 |
| - 1 | 2 | Can be set | SysTick | System tick timer | 0x0000_003C |
| 0 | 3 | Can be set | TMR0 | TMR0 timer 0 interrupt | 0x0000_0040 |
| 1 | 4 | Can be set | GPIO | GPIO general I/O interrupt | 0x0000_0044 |
| 2 | 5 | Can be set | SLAVE | Passive parallel interrupt | 0x0000_0048 |
| 3 | 6 | Can be set | SPI0 | SPI0 interrupt | 0x0000_004C |
| 4 | 7 | Can be set | BLEL | LLE interrupt of wireless module | 0x0000_0050 |
| 5 | 8 | Can be set | BLEB | BB interrupt of wireless module | 0x0000_0054 |
| 6 | 9 | Can be set | USB | USB interrupt | 0x0000_0058 |
| 7 | 10 | Can be set | ETH | Ethernet interrupt | 0x0000_005C |
| 8 | 11 | Can be set | TMR1 | TMR1 Timer 1 interrupt | 0x0000_0060 |
| 9 | 12 | Can be set | TMR2 | TMR2 Timer 2 interrupt | 0x0000_0064 |
| 10 | 13 | Can be set | UART0 | UART0 asynchronous serial port 0 interrupt | 0x0000_0068 |
| 11 | 14 | Can be set | UART1 | UART1 Asynchronous serial port 1 interrupt | 0x0000_006C |
| 12 | 15 | Can be set | RTC | RTC real-time clock interrupt | 0x0000_0070 |
| 13 | 16 | Can be set | ADC | ADC interrupt | 0x0000_0074 |
| 14 | 17 | Can be set | SPI1 | SPI1 interrupt | 0x0000_0078 |
| 15 | 18 | Can be set | led | LED screen control interrupted | 0x0000_007C |
| 16 | 19 | Can be set | TMR3 | TMR3 timer 3 interrupt | 0x0000_0080 |
| 17 | 20 | Can be set | UART2 | UART2 Asynchronous serial port 2 interrupt | 0x0000_0084 |
| 18 | twenty one | Can be set | UART3 | UART3 Asynchronous serial port 3 interrupt | 0x0000_0088 |
| 19 | twenty two | Can be set | WDOG_BAT | Watchdog timer interrupt / low battery interrupt | 0x0000_008C |

# Chapter 4 System Control

## 4.1 Reset control

The system supports 6 types of reset, including power-on reset RPOR (real power on reset), external manual reset MR (manual reset), internal software reset SR (software reset), and watch-dog timeout reset WTR (watch-dog time reset). -out reset), global reset GRWSM (global reset by waking under shutdown mode) caused by wake-up in power-down mode, local reset by waking LRW (local reset by waking) caused by regular wake-up.

The registers R8_GLOB_RESET_KEEP and RB_ROM_CODE_OFS are only reset when RPOR and GRWSM are reset, and are not affected by other reset forms.

For the timing parameters and reset characteristic parameters in the following figure, please refer to the timing parameter table in Section 20.5.

### 4.1.1 Power-on reset RPOR

When the power is turned on, the chip will generate a power-on reset and delay to wait for the power supply to stabilize. The following figure shows the power-on reset process.

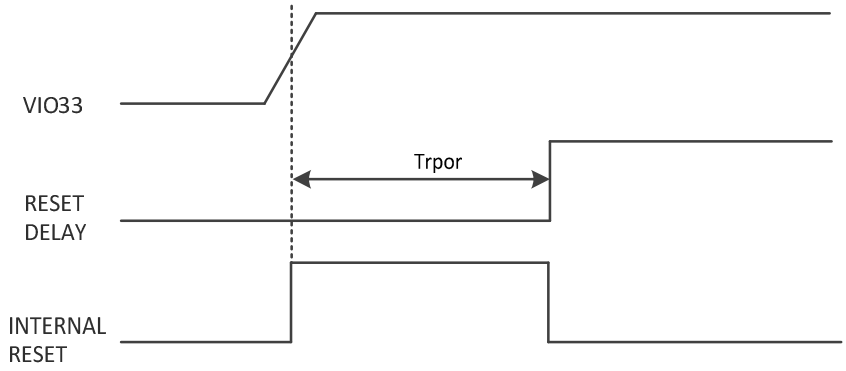



Figure 4-1 Power-on reset

### 4.1.2 External manual reset MR

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

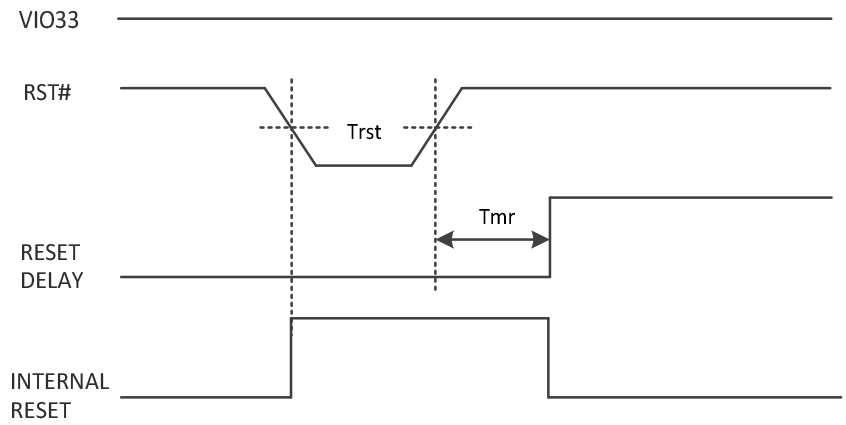



Figure 4-2 External reset

### 4.1.3 Internal software reset SR

Internal software reset is used to reset itself without external intervention. Set the global reset configuration register (R8_RST_WDOG_CTRL) If the bit RB_SOFTWARE_RESET is 1, the software reset can be realized. This bit will be automatically cleared to 0.

#### 4.1.4 Watchdog timeout reset WTR

The watchdog function is based on an 8-bit up counter with a counting clock cycle of 131072/Fsys. When the watchdog timeout reset function is enabled, the entire system will be reset once this counter overflows.

#### 4.1.5 Reset GRWSM after waking up from power-down mode

Once the system enters the power-down mode (see the power management chapter for details), the system will execute the wake-up operation in an orderly manner under the acti
After waking up, the system will perform a global reset. The effect of this reset is similar to a power-on reset.

#### 4.1.6 Reset LRW caused by regular wake-up operation

If the system is awakened from sleep mode, a reset will be generated after the relevant power supply is ready. This reset is a partial reset.
Perform 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 three categories:

The first is the key register belonging to the functional module that requires data retention (such as configuration/mode, etc.), which is relayed by the auxiliary power supply du

Continuous power supply, no data loss, sleep and wake-up have no effect on its data;

The second type is the regenerable register (such as counter, FIFO, etc.) belonging to the functional module that needs data retention, and the power is turned off during sleep.

After waking up, the data is a random number (such as FIFO storage unit) or reset (such as FIFO counter) ;

The third type is a register belonging to a functional module that does not require data retention. The power is off during sleep, and the data is a random number after wakeup

FIFO storage unit) or reset (such as FIFO counter, configuration/mode register) .

LRW is used for the latter two registers to be reset.

### 4.2 Secure access

The attribute of some registers in the system is "RWA" or "WA", indicating that the current register is a safe access register and can be read directly
But writing needs to enter the secure access mode.

Write R8_SAFE_ACCESS_SIG register 0x57 first;

then write R8_SAFE_ACCESS_SIG register 0xA8;

Then you can enter the safe access mode. At this time, you can operate the register with the attribute of "RWA/WA", and then about 16 system clock cycles (Tsys)
Are in the safe mode, one or more safety registers can be rewritten within the validity period, and the security will be automatically terminated after the validity period is exceeded.
Full mode. Or you can write 0x00 to the R8_SAFE_ACCESS_SIG register in advance to terminate the safety mode early.

### 4.3 Register description

Table 4-1 System control related register list

| name | address | description | Reset value |
|---|---|---|---|
| R8_SAFE_ACCESS_SIG | 0x40001040 | Security Access Flag Register | 0x00 |
| R8_CHIP_ID | 0x40001041 | Chip ID register | 0x79 |
| R8_SAFE_ACCESS_ID | 0x40001042 | Secure access ID register | 0x04 |
| R8_WDOG_COUNT | 0x40001043 | Watchdog counter register | 0x00 |
| R8_RESET_STATUS | 0x40001044 | Reset status register | 0x01 |
| R8_GLOB_CFG_INFO | 0x40001045 | Global configuration information status register | 0xEX |
| R8_RST_WDOG_CTRL | 0x40001046 | Watchdog and reset configuration register | 0x00 |
| R8_GLOB_RESET_KEEP | 0x40001047 | Reset holding register | 0x00 |
| R8_CFG_FLASH | 0x4000104A | FlashROM configuration register | 0x0X |
| R32_FLASH_DATA | 0x40001800 | FlashROM data register | 0xXXXXXXXX |
| R32_FLASH_ADDR | 0x40001804 | FlashROM address register | 0xXXXXXXXX |
| R8_FLASH_COMMAND | 0x40001808 | FlashROM command register | 0x00 |
| R8_FLASH_PROTECT | 0x40001809 | FlashROM protection control register | 0x00 |
| R16_FLASH_STATUS | 0x4000180A | FlashROM status register | 0x0X00 |

An Full access mark Remember Register (R8_SAFE_ACCES S_SIG)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_SAFE_ACCESS_SIG | WO | Security access flag register.<br>Some registers (access attribute is RWA) are protected registers<br>Memory, you must enter the secure access mode to proceed<br>Write operation. Write 0x57 to this register first, then write<br>Enter 0xA8 to enter the secure access mode, and<br>The time limit is about 16 main clock cycles (Tsys), if more than<br>Automatic protection. You can write any other value to force straight<br>Then exit the safe access mode and return to the protected state. | 00h |
| 7 | Reserved | R0 | Reserved. | 0 |
| [6:4] | RB_SAFE_ACC_TIMER | RO | Current security access time count. | 000b |
| 3 | RB_SAFE_ACC_ACT | RO | Current security access mode status:<br>1: Writable in unlocked/safe access mode;<br>0: Locked, RWA attribute register cannot be rewritten. | 0 |
| 2 | Reserved | R0 | Reserved. | 0 |
| [1:0] | RB_SAFE_ACC_MODE | RO | Current security access mode status:<br>11: Safe mode, can write attribute RWA register;<br>Other: non-safe mode. | 00b |

Chip ID register (R8_CHIP_ID)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_CHIP_ID | RF | The fixed value is 79h, which is used to identify the chip. | 79h |

Security access ID Register (R8_SAFE_ACCESS_ ID)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_SAFE_ACCESS_ID | RF | The fixed value is 04h. | 04h |

Look Door dog count Device Register (R8_WDOG_COUNT)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_WDOG_COUNT | RW | Watchdog counter with preset initial value, always automatic<br>Increment, can cycle from 0xFF to 0x00 and then<br>continue. Counting period = 131072/Fsys. | 00h |

complex Bit status Save (R8_RESET_STATUS)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [7:5] | Reserved | R0 | Reserved. | 000b |
| 4 | RB_ROM_CODE_OFS | RWA | Select the beginning of the user program code in FlashROM<br>Offset address, this value is not affected by MR, SR or WTR,<br>and can be cleared only when RPOR and GRWSM are valid: 0:<br>0x000000;<br>1: 0x008000 (skip the first 32KB in ROM). | 0 |
| 3 | Reserved | R0 | Reserved. | 0 |
| [2:0] | RB_RESET_FLAG | RO | Last reset status:<br>000: Software reset SR (RB_WDOG_RST_EN=0<br>When the software reset can produce this state, otherwise it can be reset<br>But this state is not generated); | 001b |

| | | | | |
|---|---|---|---|---|
| | | | 001: Power-on reset RPOR;<br>010: Watchdog timeout reset WTR;<br>011: External manual reset MR;<br>101: Reset GRWSM when waking up from power-down mode;<br>100/110/111: wake up reset LRW, and the previous<br>The last reset was SR/WTR/MR. | |

all Bureau Configuration Letter interest Status register (R8_GLOB_C F G_INFO )

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [7:6] Reserved | | R0 | Reserved. | 11b |
| 5 | RB_BOOT_LOADER | RO | Bootloader status:<br>1: Currently in the Bootloader state;<br>0: Currently in the user program state. | 1/0 |
| 4 | RB_CFG_DEBUG_EN | RO | Two-wire simulation debugging interface enable state:<br>1: It can be simulated and debugged, and FlashROM can be read;<br>0: Prohibit simulation debugging. | 0 |
| 3 | RB_CFG_BOOT_EN | RO | BootLoader enable state of the system boot program:<br>1: Enabled; 0: Not enabled. | 1 |
| 2 | RB_CFG_RESET_EN | RO | RST# External manual reset input enable state:<br>1: Enabled; 0: Not enabled. | 0 |
| 1 | Reserved | RO | Reserved. | 0 |
| 0 | RB_CFG_ROM_READ | RO | FlashROM code and data area protection status:<br>1: Readable by external programmer;<br>0: Protection, inaccessible from outside, and the program is confidential. | 0 |

Look Door dog and complex Bit Configuration register (R8_RST_WD O G_CTRL )

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [7:5] | Reserved | RO | Reserved. | 000b |
| 4 | RB_WDOG_INT_FLAG | RW1 | Watchdog timer interrupt flag:<br>1: A watchdog count overflow has occurred, that is, it is detected R8_WDOG_COUNT increments from 0xFF to 0x00; 0: The watchdog count has not overflowed.<br>Write 1 to the flag and clear it to 0, or reload the watchdog counter value (R8_WDOG_COUNT) and clear it to 0, or execute Line __SEV() is cleared to 0. | 0 |
| 3 | Reserved | RO | Reserved. | 0 |
| 2 | RB_WDOG_INT_EN | RWA | Watchdog timer interrupt enable bit:<br>1: Enable, an interrupt will be generated after the watchdog count overflows;<br>0: Disable the watchdog timer interrupt. | 0 |
| 1 | RB_WDOG_RST_EN | RWA | Watchdog timeout reset enable bit:<br>1: Enable, the system resets after the watchdog count overflows;<br>0: Only as a watchdog timer.<br>Note: After this bit is set to 1, the software reset operation will not affect RB_RESET_FLAG status. | 0 |
| 0 | RB_SOFTWARE_RESET | WA/ WZ | System software reset control, it will be cleared automatically after reset:<br>1: Perform system software reset;<br>0: Idle, no action. | 0 |

Reset holding register (R8_GLOB_RESET_KEEP)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_GLOB_RESET_KEEP | RW | Reset the holding register, the value of this register is not subject to manual Reset, software reset, watchdog reset or normal Impact of wake-up reset. | 00h |

FlashROM configuration Register (R8_CFG_FLASH)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| 7 | RB_FLASH_BUSY_EN | RWA | Busy enable bit during FlashROM burst access: 1: There is waiting (recommended); 0: No waiting. | 0 |
| [6:4] | Reserved | RO | Reserved. | 000b |
| [3:0] | RB_CFG_FLASH_X | RWA | FlashROM configuration data, must keep the original when writing The value does not change. | XXXXb |

FlashROM data Register (R32_FLASH_DAT A)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [31:0] | R32_FLASH_DATA | RW | FlashROM data register. | XXXXXXXXh |

FlashROM address Register (R32_FLASH_ADD R)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [31:0] | R32_FLASH_ADDR | RW | FlashROM address register. | XXXXXXXXh |

FlashROM command Register (R8_FLASH_COMM AND)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_FLASH_COMMAND | WO | 0x9A: FlashROM programming operation command; 0xA6: FlashROM sector erase operation command. | 00h |

FlashROM protection Control register (R8_FLASH_PROTEC T)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [7:6] | RB_ROM_WE_MUST_10 | WO | Must write 10b, otherwise it cannot be written. | 10b |
| [5:4] | Reserved | RO | Reserved. | 00b |
| 3 | RB_ROM_CODE_WE | RW | Erase/program enable bit of CodeFlash in FlashROM program storage area: 1: Allow erasing/programming; 0: Erase and write protection. | 0 |
| 2 | RB_ROM_DATA_WE | RW | The erase/program enable bit of FlashROM data storage area DataFlash: 1: Allow erasing/programming; 0: Erase and write protection. | 0 |
| [1:0] | Reserved | RO | Reserved. | 00b |

FlashROM status Register (R16_FLASH_STA TUS)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [15:11] | Reserved | RO | Reserved. | 00000b |
| [10:9] | Reserved | RO | Reserved. | 01b |
| 8 | RB_ROM_READ_FREE | RO | FlashROM code and data area protection status: 1: Readable by external programmer; 0: Protection, inaccessible from outside, and the program is confidential. | 0 |
| 7 | Reserved | RO | Reserved. | 0 |
| 6 | RB_ROM_ADDR_OK | RO | FlashROM erase/program operation address valid flag, | 0 |

| | | | | |
|---|---|---|---|---|
| | | | It can be confirmed before and after the operation:<br>1: The address is valid; 0: The address is invalid. | |
| [5:2] | Reserved | RO | Reserved. | 0000b |
| 1 | RB_ROM_CMD_ERR | RO | FlashROM command response error:<br>1: Unknown command; 0: The command is accepted. | 0 |
| 0 | RB_ROM_CMD_TOUT | RO | FlashROM operation result:<br>1: Timeout; 0: Success. | 0 |

## 4.4 Flash-ROM operation steps

1. Erase Flash-ROM and change all data bits in the target sector to 1:

(1) Set the R8_FLASH_PROTECT register, turn on the erase/program enable RB_ROM_DATA_WE or RB_ROM_CODE_WE bit, corresponding to DataFlash or CodeFlash, InfoFlash requires both RB_ROM_DATA_WE and RB_ROM_CODE_WE to be turned on;

(2) , Set the address register R32_FLASH_ADDR, write a 32-bit address, the lower 8 bits are invalid, erase the whole block of 512 bytes at a time; (3) Set the command register R8_FLASH_COMMAND, write 0A6H, execute the sector erase operation, the MCU will automatically Suspension of operation;

(4) After the operation is completed, the program resumes operation. At this time, query the status register R16_FLASH_STATUS to view the operation status; If you want to erase multiple sectors, cycle (2) ~ (4) steps;

(5) Set the R8_FLASH_PROTECT register again and turn off the erase/program enable control bit (RB_ROM_DATA_WE=0 or RB_ROM_CODE_WE=0) .

2. Write to Flash-ROM and change some data bits in the target word from 1 to 0 (the bit data cannot be changed from 0 to 1):

(1) Set the R8_FLASH_PROTECT register, turn on the erase/program enable RB_ROM_DATA_WE or RB_ROM_CODE_WE bit, corresponding to DataFlash or CodeFlash, InfoFlash requires both RB_ROM_DATA_WE and RB_ROM_CODE_WE to be turned on;

(2) Set the address register R32_FLASH_ADDR, and write a 32-bit address, which requires 4-byte alignment; (3) Set the data register R32_FLASH_DATA to the 4-byte data to be written;

(4) Set the command register R8_FLASH_COMMAND, write 09AH, execute programming/write operation, and the MCU will automatically suspend operation during the operation;

(5) After the operation is completed, the program resumes operation. At this time, check the status register R16_FLASH_STATUS to view the operation status; If you want to write multiple data, cycle (2) ~ (4) steps;

(5) Set the R8_FLASH_PROTECT register again and turn off the erase/program enable control bit (RB_ROM_DATA_WE=0 or RB_ROM_CODE_WE=0) .

3. Read Flash-ROM:

Read the code or data of the target address through the pointer to the program memory space.

## 4.5 Chip unique ID number

Each chip has a unique ID number when it leaves the factory, that is, the chip identification number. The ID data and its checksum total 8 bytes, which are stored in the read-only area inside the chip. For specific operations, please refer to the example program.

# Chapter 5 Power Control

## 5.1 Power Management

CH579 has a built-in power management unit PMU. The system power is input from VIO33, and through the built-in multiple LDO voltage regulators, the system's FlashROM, system's digital circuits (including core, USB, Ethernet, LED, etc.) and system's analog circuits (including high-frequency oscillator, PLL, ADC and RF transceiver, etc.) provide the required power.

The power supply during normal operation is divided into two ways: direct power supply and DC-DC conversion. In addition to normal operation, CH579 provides 4 low power consumption modes: idle mode, suspend mode, sleep mode, and power-down mode.

DC-DC is not enabled by default after power-on, but a through power supply is provided, and its voltage ripple is small. In order to reduce the system function during normal operat You can choose to enable DC-DC to increase the power consumption utilization rate. The working current will usually drop to about 60% of the direct mode.

In order to reduce the power consumption of the system 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 th Provide auxiliary power. When the system enters sleep or power-down mode, in addition to the regular power supply units such as power management and RTC registers, the The 2KB and 14KB high SRAM, core and all peripherals can choose whether to maintain power supply, and LSE/LSI can choose whether to turn on.

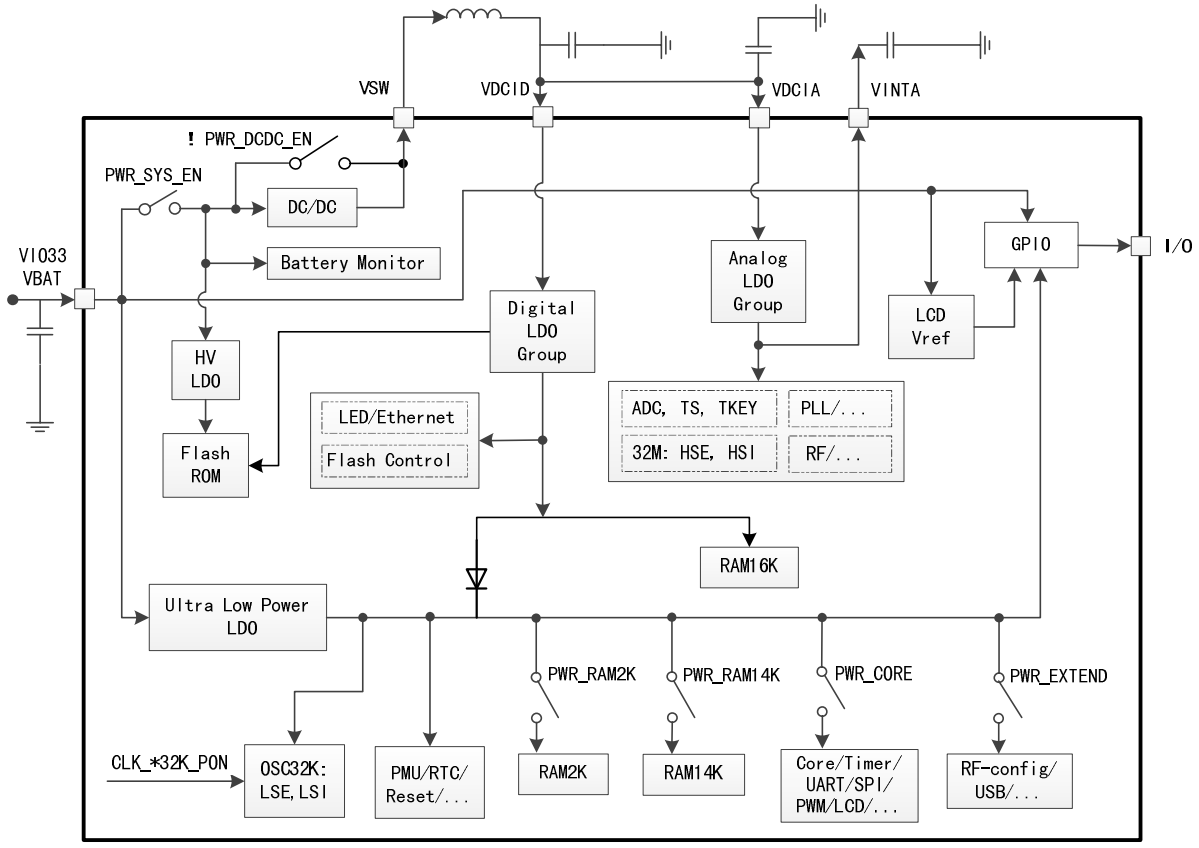



Figure 5-1 Power system

## 5.2 Register description

Table 5-1 List of registers related to power management

| name | 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 | Wake-up event configuration register | 0x20 |
| R8_SLP_POWER_CTRL | 0x4000100F | Peripheral sleep power control register | 0x08 |

| | | | |
|---|---|---|---|
| R16_POWER_PLAN | 0x40001020 | Sleep Power Management Register | 0x01DF |
| R8_AUX_POWER_ADJ | 0x40001022 | Auxiliary power adjustment control register | 0xXX |
| R8_BAT_DET_CTRL | 0x40001024 | Battery voltage detection control register | 0x00 |
| R8_BAT_DET_CFG | 0x40001025 | Battery voltage detection configuration register | 0x01 |
| R8_BAT_STATUS | 0x40001026 | Battery status register | 0x00 |

sleep Sleep clock Control register 0 (R8_SLP_CLK_OFF0)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| 7 | RB_SLP_CLK_UART3 | RWA | Serial port 3 clock source:<br>1: Closed; 0: Turn on. | 0 |
| 6 | RB_SLP_CLK_UART2 | RWA | Serial port 2 clock source:<br>1: Closed; 0: Turn on. | 0 |
| 5 | RB_SLP_CLK_UART1 | RWA | Clock source of serial port 1:<br>1: Closed; 0: Turn on. | 0 |
| 4 | RB_SLP_CLK_UART0 | RWA | Clock source of serial port 0:<br>1: Closed; 0: Turn on. | 0 |
| 3 | RB_SLP_CLK_TMR3 | RWA | Timer 3 clock source:<br>1: Closed; 0: Turn on. | 0 |
| 2 | RB_SLP_CLK_TMR2 | RWA | Timer 2 clock source:<br>1: Closed; 0: Turn on. | 0 |
| 1 | RB_SLP_CLK_TMR1 | RWA | Timer 1 clock source:<br>1: Closed; 0: Turn on. | 0 |
| 0 | RB_SLP_CLK_TMR0 | RWA | Timer 0 clock source:<br>1: Closed; 0: Turn on. | 0 |

sleep Sleep clock Control register 1 (R8_SLP_CLK_OFF1)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| 7 | RB_SLP_CLK_BLE | RWA | BLE controller clock source:<br>1: Closed; 0: Turn on. | 0 |
| 6 | RB_SLP_CLK_LED | RWA | LED controller clock source:<br>1: Closed; 0: Turn on. | 0 |
| 5 | RB_SLP_CLK_ETH | RWA | ETH controller clock source:<br>1: Closed; 0: Turn on. | 0 |
| 4 | RB_SLP_CLK_USB | RWA | USB controller clock source:<br>1: Closed; 0: Turn on. | 0 |
| 3 | RB_SLP_CLK_LCD | RWA | LCD controller clock source:<br>1: Closed; 0: Turn on. | 0 |
| 2 | RB_SLP_CLK_PWMX | RWA | PWMx clock source:<br>1: Closed; 0: Turn on. | 0 |
| 1 | RB_SLP_CLK_SPI1 | RWA | SPI1 clock source:<br>1: Closed; 0: Turn on. | 0 |
| 0 | RB_SLP_CLK_SPI0 | RWA | SPI0 clock source:<br>1: Closed; 0: Turn on. | 0 |

Call Wake up event Set Register (R8_SLP_WAKE_C TRL)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [7:6] | Reserved | RO | Reserved. | 00b |
| 5 | RB_SLP_BAT_WAKE | RWA | Enable battery low-voltage event to wake up the system: | 1 |

| | | | | |
|---|---|---|---|---|
| | | | 1: enable; 0: Closed. | |
| 4 | RB_SLP_GPIO_WAKE | RWA | Enable GPIO event to wake up the system:<br>1: enable; 0: Closed. | 0 |
| 3 | RB_SLP_RTC_WAKE | RWA | Enable RTC event to wake up the system:<br>1: enable; 0: Closed. | 0 |
| 2 | Reserved | RO | Reserved. | 0 |
| 1 | RB_SLP_ETH_WAKE | RWA | Enable Ethernet event to wake up the system:<br>1: enable; 0: Closed. | 0 |
| 0 | RB_SLP_USB_WAKE | RWA | Enable USB event to wake up the system:<br>1: enable; 0: Closed. | 0 |

outer Set sleep source Control register (R8_SLP_PO W ER_CTR L)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [7:6] | Reserved | RO | Reserved. | 00b |
| 5 | RB_SLP_CLK_RAM2K | RWA | SRAM clock control of RAM2K:<br>1: Close; 0: Turn on. | 0 |
| 4 | RB_SLP_CLK_RAMX | RWA | Clock control of main SRAM (RAM16K+RAM14K):<br>1: Closed; 0: Turn on. | 0 |
| 3 | RB_SLP_ROM_PWR_DN | RWA | In suspend mode, FlashROM power-down enable:<br>1: Make the ROM enter the disabled state in the pause mode;<br>0: Keep ROM in standby state in suspend mode. | 1 |
| 2 | Reserved | RO | Reserved. | 0 |
| 1 | RB_SLP_ETH_PWR_DN | RWA | Ethernet transceiver ETH PHY power-down enable:<br>1: power-down/disable; 0: keep power supply. | 0 |
| 0 | Reserved | RO | Reserved. | 0 |

sleep Sleep power management send Register (R16_POWER_PLA N)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| 15 | RB_PWR_PLAN_EN | RWA/ WZ | Sleep power planning control enable:<br>1: Open planning; 0: Close or end planning.<br>Turn on the power plan for going to sleep or downloading later<br>It is executed in power mode, and the bit is automatically cleared after execution. | 0 |
| [14:11] | RB_PWR_MUST_0010 | RWA | Reserved, 0010b must be written. | 0000b |
| 10 | RB_PWR_DCDC_PRE | RWA | DC-DC bias circuit enable (immediate effect):<br>1: enable; 0: Prohibited. | 0 |
| 9 | RB_PWR_DCDC_EN | RWA | DC-DC enable bit (effective immediately) :<br>1: DC-DC is enabled, the pass-through power is off;<br>0: DC-DC forbidden, the direct power is turned on. | 0 |
| 8 | RB_PWR_LDO_EN | RWA | Internal LDO control (sleep planning):<br>1: Turn on LDO; 0: Plan to turn off LDO, which is more economical. | 1 |
| 7 | RB_PWR_SYS_EN | RWA | System power control (sleep planning) :<br>1: Provide system power (at VSW pin);<br>0: Turn off the system power, the plan will enter sleep mode<br>Or power down mode. | 1 |
| 6 | Reserved | RWA | Reserved, 0 must be written. | 1 |
| 5 | Reserved | RO | Reserved. | 0 |
| 4 | RB_PWR_RAM14K | RWA | SRAM power supply for RAM14K (sleep planning): 1: Dual power supply; 0: No auxiliary power supply. | 1 |

| | | | | |
|---|---|---|---|---|
| 3 | RB_PWR_EXTEND | RWA | USB and RF configuration power supply (sleep planning)<br>1: Dual power supply; 0: No auxiliary power supply. | 1 |
| 2 | RB_PWR_CORE | RWA | Power supply for the core and basic peripherals (sleep planning):<br>1: Dual power supply; 0: No auxiliary power supply. | 1 |
| 1 | RB_PWR_RAM2K | RWA | SRAM power supply for RAM2K (sleep planning):<br>1: Dual power supply; 0: No auxiliary power supply. | 1 |
| 0 | Reserved | RO | Reserved. | 1 |

Except for RB_PWR_DCDC_PRE and RB_PWR_DCDC_EN, this register is preset for sleep planning, and its power configuration takes effect after entering the low-power sleep mode and power-down mode.

auxiliary Power adjustment whole Control register (R8_AUX_PO W ER_ADJ )

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [7:6] | Reserved | RO | Reserved. | 00b |
| [5:3] | Reserved | RWA | Reserved, the original value must be kept unchanged when writing. | 1XXb |
| [2:0] | RB_ULPLDO_ADJ | RWA | Auxiliary power output voltage regulation of ultra-low power LDO<br>Value (the value is for reference only, it is not recommended to modify):<br>000: 0.908V; 001: 0.931V;<br>010: 0.954V; 011: 0.977V;<br>100: 1.000V; 101: 1.023V;<br>110: 1.046V; 111: 1.069V. | XXXb<br>(100b) |

Electricity Cell voltage check Measurement Control register (R8_BAT_DE T_CTRL)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [7:4] | Reserved | RO | Reserved. | 0000b |
| 3 | RB_BAT_LOW_IE | RWA | Battery low voltage interrupt enable:<br>1: enable; 0: Closed. | 0 |
| 2 | RB_BAT_LOWER_IE | RWA | Battery ultra-low voltage interrupt enable:<br>1: enable; 0: Closed. | 0 |
| 1 | Reserved | RO | Reserved. | 0 |
| 0 | RB_BAT_DET_EN | RWA | Battery voltage detection function is enabled:<br>1: Enable low voltage detection and enable reference voltage at the same time<br>Module, current 210uA in sleep mode;<br>0: Closed. | 0 |

Note: If the battery voltage reaches the ultra-low voltage detection threshold, and RB_BAT_LOWER_IE and RB_BAT_LOW_IE are both enabled (only one of them is enabled under normal circumstances), an NMI non-maskable interrupt will be generated, which is equivalent to increasing the interrupt priority.

Electricity Cell voltage check Measurement Configuration register (R8_BAT_DE T_CFG)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [7:2] | Reserved | RO | Reserved. | 000000b |
| [1:0] | RB_BAT_LOW_VTH | RWA | Set the detection threshold of low voltage and ultra-low voltage:<br>(Ultra-low voltage reference threshold, low voltage reference threshold)<br>00: 1.97V , 2.25V;<br>01: 2.05V , 2.33V;<br>10: 2.13V , 2.41V;<br>11: 2.21V , 2.49V. | 01b |

Electricity Pool status Register (R8_BAT_STATUS)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|

| [7:2] | Reserved | RO | Reserved. | 000000b |
|---|---|---|---|---|
| 1 | RB_BAT_STAT_LOW | RO | The battery voltage is in a low voltage state:<br>1: Lower than the low voltage threshold; 0: No. | 0 |
| 0 | RB_BAT_STAT_LOWER | RO | The battery voltage is in an ultra-low voltage state:<br>1: Lower than the ultra-low voltage threshold; 0: No. | 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, you can select the appropriate low-power mode To save power consumption. The user needs to select an appropriate low power consumption based on the lowest power consumption, fastest startup time, and available wake-up events. Power consumption mode.

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

- Idle mode (Idle)

  All peripherals maintain normal power supply, the core stops running, and the clock system runs. After a wake-up event is detected, it can be awakened immediately.

- Pause Mode (Halt)

  On the basis of the idle mode, the clock system is stopped, and the FlashROM can be put into shutdown mode to further reduce power consumption. detected After a wake-up event, the clock runs first, and then the core runs.

- Sleep mode (Sleep):

  The main LDO is turned off, and the ultra-low power ULP-LDO maintains the power supply of the PMU, core and basic peripherals. The LSE or LSI can choose whether to turn on, and the RAM2K, RAM14K, USB and RF configurations can choose whether to maintain power. After a wake-up event is detected, first the main LDO is turned on, then the clock runs, and finally the core is woken up, the program continues to run, and can be reset to a higher frequency if necessary.

- Power-off mode (Shutdown)：

  On the basis of sleep mode, the kernel and basic peripherals, USB and RF configuration are turned off, LSE or LSI can choose whether to turn on, RAM2K, RAM14K can choose whether to maintain power supply. After a wake-up event is detected, the PMU will perform a GRWSM reset, and the software can distinguish it from RPOR based on the reset flag RB_RESET_FLAG and optional RAM retention data.

The following table details the characteristics and wake-up ways of several low-power modes:

Table 5-2 Low power consumption mode

| mode | feature | Entry conditions | Wake up event | Power consumption |
|---|---|---|---|---|
| Idle mode<br>Idle | The peripherals are powered normally, the kernel stops running,<br>When the clock system is running, but the peripherals can be used<br>The clock control bit selects to turn off the clock of each peripheral | Set the core control bit SCR=0,<br>the wake-up condition and executes<br>__WFI() or __WFE(). | I/O or RTC sets<br>or BAT or<br>USB or ETH | 1.15mA ～<br>1.5mA |
| Pause mode<br>Halt | The peripherals are powered normally, the kernel stops running,<br>Clock system stop (PLL/HSE/HSI stop) . Execute after setting the wake-up condition<br>You can choose FlashROM standby or disable. | Set the core control bit SCR=1, I/O or RTC<br>or BAT or<br>__WFI() or __WFE(). | | 420uA ～<br>470uA |
| Sleep mode<br>Sleep | The main LDO is turned off, and the ultra-low power<br>ULP-LDO maintains the power supply of the PMU, core and basic peripherals<br>LSE or LSI can choose whether to turn on, RAM2K,<br>RAM14K, USB and RF configuration can choose<br>whether to maintain power supply. | Set the core control bit SCR=1, I/O or RTC<br>set POWER_PLAN,<br>After setting the wake-up condition, execute the chip<br>__WFI() or __WFE(). | Or BAT.<br>wake-up<br>Will continue to run | 0.6uA ～<br>2.0uA |
| You can choose whether to turn on LSE or LSI in power-off mode,<br>Shutdown RAM2K, RAM14K can choose whether to maintain the<br>Power supply, used to maintain data. | The ultra-low power LDO maintains the power supply of the PMU, | Set the core control bit SCR=1, I/O or RTC<br>set POWER_PLAN,<br>set wake-up conditions and execute the chip<br>__WFI() or __WFE(). | Or BAT.<br>wake-up<br>Will automatically reset | 0.2uA ～<br>1.3uA |

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

Table 5-3 Low Power consumption mode details Configuration example

| Planning configuration | SYS_EN | RAM2K RAM14K | CK32K | CORE | EXTEND power consumption (for reference only) |
|---|---|---|---|---|---|
| Maintain the data area of | the power supply data area of | | the power supply system | LSE/LSI choose one M0 core | and USB and PMU and RTC registers |

| Function of | VSW | 2KB | 14KB | RTC wake up | Basic peripherals | RF configuration | often power supply, about 0.2uA |
|---|---|---|---|---|---|---|---|
| Power down mode Common configuration | 0 | 0 | 0 | 0 | 0 | 0 | 0.2uA |
| | 0 | 1 | 0 | 0 | 0 | 0 | 0.5uA |
| | 0 | 0 | 0 | 1 | 0 | 0 | 0.5uA ～0.6uA |
| | 0 | 1 | 0 | 1 | 0 | 0 | 0.8uA ～0.9uA |
| Sleep mode Common configuration | 0 | 1 | 0 | 0 | 1 | 0 | 0.6uA |
| | 0 | 1 | 0 | 1 | 1 | 0 | 0.9uA ～1.0uA |
| | 0 | 0 | 1 | 0 | 1 | 0 | 1.1uA |
| | 0 | 0 | 1 | 1 | 1 | 0 | 1.4uA ～1.5uA |
| | 0 | 1 | 1 | 1 | 1 | 1 | 1.8uA ～2.0uA |

## 5.4 DC-DC operation steps

Enable the DC-DC power mode (before enabling the external hardware circuit to confirm the inductance and capacitance required for DC-DC) (1), enter the safe access mode: register R8_SAFE_ACCESS_SIG first write 0x57, then write 0xA8; (2) 、 Open the DC-DC bias circuit: set the RB_PWR_DCDC_PRE of the register R16_POWER_PLAN to 1; (3) The delay is at least 10uS;

(4) Enter the safe access mode again: register R8_SAFE_ACCESS_SIG first write 0x57, and then write 0xA8; (5) Turn on the DC-DC power supply: set the RB_PWR_DCDC_EN of the register R16_POWER_PLAN to 1 to enable DC-DC.

Turn off DC-DC and switch to direct power supply mode

(1) Enter the safe access mode: Register R8_SAFE_ACCESS_SIG first write 0x57, and then write 0xA8; (2) Clear the RB_PWR_DCDC_EN and RB_PWR_DCDC_PRE control bits of the register R16_POWER_PLAN.

# 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)

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

Any clock source can be started or shut down independently, thereby optimizing system power consumption.

### 6.1.1 Clock structure

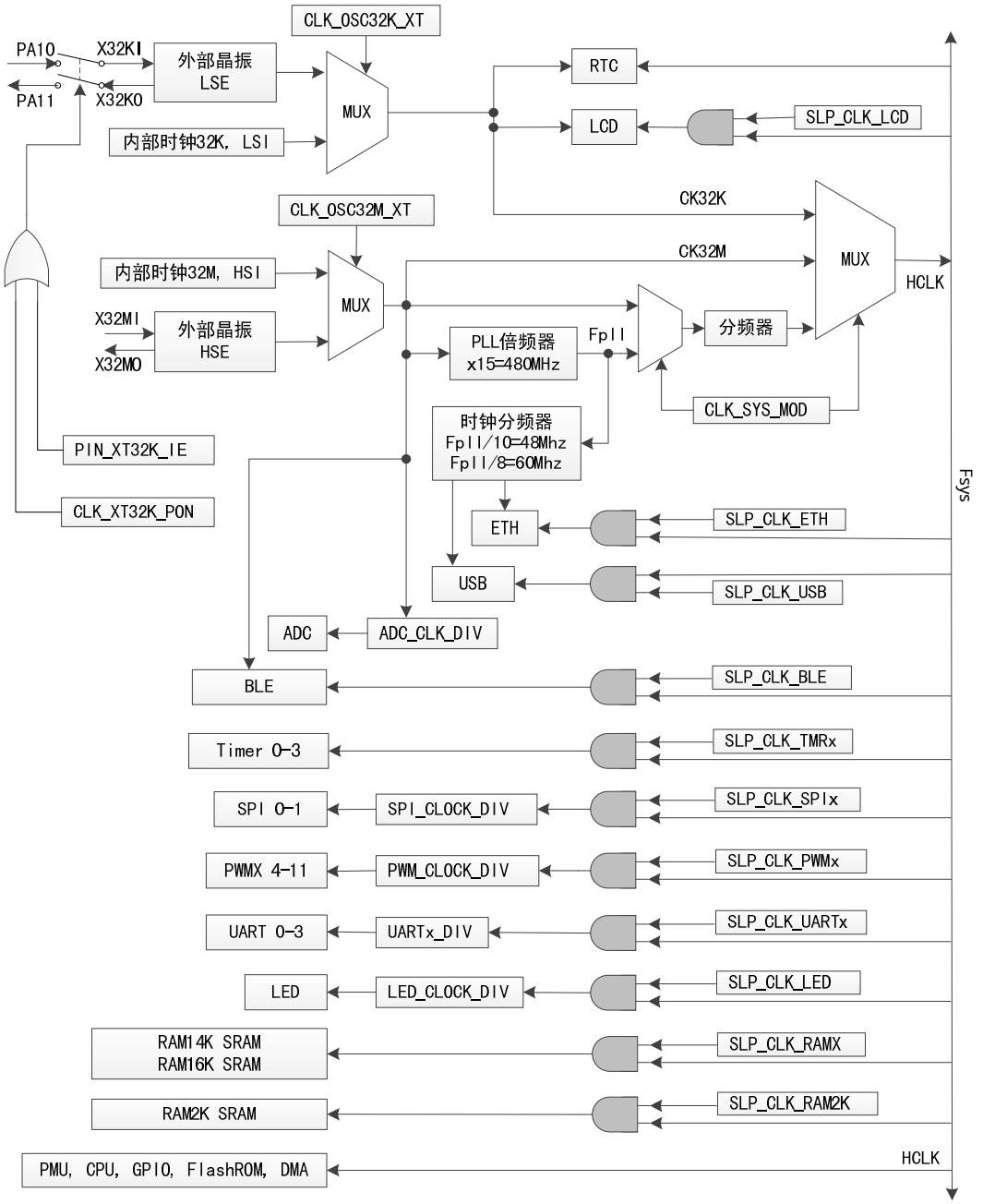



Figure 6-1 Clock tree block diagram

The above figure is the internal clock tree structure of the system. The RTC and LCD functions use the 32KHz clock source CK32K, so the low-frequency clock must be turned on to use these functions; the data transmission of USB and Ethernet depends on the clock source generated after PLL frequency division; Other peripheral drive clocks and digital control logic are driven by the system clock or sub-frequency division.

## 6.2 Introduction to RTC

The real-time clock (RTC) is an independent timer that contains a set of counters that count continuously. Under the corresponding software configuration, you can mention Provide simple calendar function. Modifying the counter value can reset the current time and date.

The RTC register is always supplied with power as the PMU. After the system is reset or wakes up from the low-power mode, the RTC setting and time remain unchanged.

### 6.2.1 Main features

l Two modes can be configured:

- — Timing mode: The software can select a fixed cycle time (timing) to generate interrupt notifications.
- — Trigger mode: match a target alarm time preset by the software to generate an interrupt notification.

l Three groups of 16-bit counters provide CK32K primitive cycle, 2-second cycle, and 1-day cycle counting.

## 6.3 Register description

Table 6-1 Clock and oscillator control related register list

| name | address | description | Reset value |
|---|---|---|---|
| R16_CLK_SYS_CFG | 0x40001008 | System clock configuration register | 0x05 |
| R8_HFCK_PWR_CTRL | 0x4000100A | High frequency clock module power control register | 0x0C |
| R16_INT32K_TUNE | 0x4000102C | Internal 32KHz clock calibration register | 0x0200 |
| R8_XT32K_TUNE | 0x4000102E | External 32KHz clock resonance control register | 0xC3 |
| R8_CK32K_CONFIG | 0x4000102F | 32KHz oscillator configuration register | 0xX2 |
| R8_INT32M_CALIB | 0x4000104C | Internal 32MHz clock calibration register | 0xXX |
| R8_XT32M_TUNE | 0x4000104E | External 32MHz clock resonance control register | 0x31 |
| R16_OSC_CAL_CNT | 0x40001050 | Oscillator frequency calibration count value register | 0xXXXX |
| R8_OSC_CAL_CTRL | 0x40001052 | Oscillator frequency calibration control register | 0x02 |
| R8_PLL_CONFIG | 0x4000104B | PLL configuration register | 0xX0 |
| R8_RTC_FLAG_CTRL | 0x40001030 | RTC flag and control register | 0x30 |
| R8_RTC_MODE_CTRL | 0x40001031 | RTC mode configuration register | 0x02 |
| R32_RTC_TRIG | 0x40001034 | RTC trigger value register | 0x00000000 |
| R16_RTC_CNT_32K | 0x40001038 | RTC is based on 32768Hz count value register | 0xXXXXXXXX |
| R16_RTC_CNT_2S | 0x4000103A | RTC count value register with 2S as the unit | 0xXXXXXXXX |
| R32_RTC_CNT_DAY | 0x4000103C | RTC count value register in days | 0x0000XXXX |

system System clock configuration send Register (R16_CLK_SYS_C FG)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| 15 | RB_XO_DI | RO | The sampling value of the input state of the X32MO pin. | 0 |
| [14:10] | Reserved | RO | Reserved. | 00000b |
| 9 | RB_CLK_OSC32M_XT | RWA | CK32M (32MHz) clock source selection bit:<br>1: External 32MHz oscillator;<br>0: Internal 32MHz oscillator. | 0 |
| 8 | Reserved | RO | Reserved. | 0 |
| [7:6] | RB_CLK_SYS_MOD | RWA | HCLK system clock source mode selection:<br>00: CK32M (default 32MHz) for frequency division; | 00b |

| | | | | |
|---|---|---|---|---|
| | | | 01: PLL (default 480MHz) for frequency division;<br>10: CK32M (default 32MHz) as HCLK;<br>11: CK32K (default 32KHz) is used as HCLK. | |
| 5 | Reserved | RO | Reserved. | 0 |
| [4:0] | RB_CLK_PLL_DIV | RWA | HCLK output clock frequency division coefficient, the minimum value is 2, 0<br>Represents the maximum value of 32, writing 1 will turn off HCLK. | 00101b |

Calculation:

Fck32m = RB_CLK_OSC32M_XT? XT_32MHz: RC_32MHz;

Fck32k = RB_CLK_OSC32K_XT? XT_32KHz: RC_32KHz; Fpll =

Fck32m * 15 = 480MHz;

Fsys = RB_CLK_SYS_MOD[1]? (RB_CLK_SYS_MOD[0]? Fck32k: Fck32m):

((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: 32KHz, 1MHz ～16MHz, 32MHz, 15MHz～44MHz

high Frequency clock mode Piece Power control register (R8_HF C K_PWR_ CTRL)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [7:6] | Reserved | RWA | Reserved, 00 must be written. | 00b |
| 5 | Reserved | RO | Reserved. | 0 |
| 4 | RB_CLK_PLL_PON | RWA | PLL power control bit:<br>1: Power on; 0: Power down. | 0 |
| 3 | RB_CLK_INT32M_PON | RWA | Internal 32MHz oscillator power control bit:<br>1: Power on; 0: Power down. | 1 |
| 2 | RB_CLK_XT32M_PON | RWA | External 32MHz oscillator power control bit:<br>1: Power on; 0: Power down. | 1 |
| [1:0] | Reserved | RO | Reserved. | 00b |

Internal 32KHz clock Calibration register (R16_INT 3 2K_TUN E)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [15:10] | Reserved | RO | Reserved. | 000000b |
| [9:0] | RB_INT32K_TUNE | RWA | Internal RC 32KHz clock frequency calibration value. | 1000000000b |

External 32KHz clock Resonance control register (R8_ X T32K_T UNE)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [7:4] | RB_XT32K_C_LOAD | RWA | Choose the built-in load that matches the external 32KHz crystal Capacitance (may affect RTC clock accuracy):<br>0000: The built-in load capacity is 2pF;<br>other:<br>Capacity = RB_XT32K_C_LOAD + 12pF, 0001b ～1111b correspond to about 13pF～27pF respectively.<br>Choose according to the crystal parameters used. | 1100b |
| [3:2] | Reserved | RO | Reserved. | 00b |
| [1:0] | RB_XT32K_I_TUNE | RWA | External 32KHz oscillator bias current selection:<br>00: 70% rated current; 01: rated current;<br>10: 140% of rated current; 11: 200% of rated current. | 11b |

32KHz oscillator with Set register (R8_CK32K_CONFIG)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| 7 | RB_32K_CLK_PIN | RO | 32KHz clock pin status (asynchronous signal). | X |
| [6:3] | Reserved | RO | Reserved. | 0000b |
| 2 | RB_CLK_OSC32K_XT | RWA | CK32K (32KHz) clock source selection bit:<br>1: External 32KHz oscillator;<br>0: Internal 32KHz oscillator. | 0 |
| 1 | RB_CLK_INT32K_PON | RWA | Internal 32KHz oscillator power control bit:<br>1: Power on; 0: Power down. | 1 |
| 0 | RB_CLK_XT32K_PON | RWA | External 32KHz oscillator power control bit:<br>1: Power on; 0: Power down. | 0 |

At internal 32MHz Clock calibration register (R8_INT32M_CALIB)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_INT32M_CALIB | RWA | Internal RC 32MHz clock frequency calibration value. | XXXXXXXXb |

External 32MHz clock Resonance control register (R8_XT32M_TUNE)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| 7 | Reserved | RO | Reserved. | 0 |
| [6:4] | RB_XT32M_C_LOAD | RWA | Select the built-in load that matches the external 32MHz crystal<br>Capacitance (may affect wireless communication):<br>Capacity = RB_XT32M_C_LOAD * 2 + 10pF, 000b ～111b correspond to about 10pF～24pF respectively.<br>According to the crystal parameters used, it is commonly used as 111b. | 011b |
| [3:2] | Reserved | RO | Reserved. | 00b |
| [1:0] | RB_XT32M_I_BIAS | RWA | External 32MHz oscillator bias current selection:<br>00: 75% rated current; 01: rated current;<br>10: 125% rated current; 11: 150% rated current. | 01b |

Vibrate Oscillator frequency calibration quasi-Count value register (R16_OSC_CAL_CNT)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [15:12] | Reserved | RO | Reserved. | 0000b |
| [11:0] | RB_OSC_CAL_CNT | RO | For 5 CK32K cycles, the count value based on the main<br>frequency of the system is used to calibrate the frequency of the internal 32KHz oscillator. | XXXh |

Vibrate Oscillator frequency calibration quasi-Control register (R8_OSC_CAL_CTRL)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [7:2] | Reserved | RO | Reserved. | 000000b |
| 1 | RB_OSC_CNT_HALT | RO | Oscillator frequency calibration counter counting status bit:<br>1: Counting is paused; 0: Counting is in progress. | 1 |
| 0 | RB_OSC_CNT_EN | RWA | Oscillator frequency calibration counter enable bit:<br>1: Enable counting; 0: Counting is prohibited. | 0 |

PLL Configuration hosting (R8_PLL_CONFIG)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| 7 | RB_PLL_LOCKED | RO | PLL lock status:<br>1: Locking; 0: Currently not locked. | X |
| [6:2] | Reserved | RO | Reserved. | 00000b |
| [1:0] | RB_PLL_CFG_DAT | RWA | PLL configuration parameters. | 00b |

RTC Sign and control system 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 timer mode activation flag. | 0 |
| 5 | RB_RTC_TRIG_CLR | RW | When the trigger mode is disabled, this bit is fixed at 1.<br>When the trigger mode is enabled, write 1 to clear the trigger mode activation<br>The live flag RB_RTC_TRIG_FLAG is automatically cleared to 0. | 1 |
| 4 | RB_RTC_TMR_CLR | RW | When the timer mode is disabled, this bit is fixed at 1.<br>When the timer mode is enabled, write 1 to clear the timer mode activation<br>The live flag RB_RTC_TMR_FLAG is automatically cleared to 0. | 1 |
| [3:0] | Reserved | RO | Reserved. | 0000b |

RTC Mode configuration send Register (R8_RTC_MODE_CT RL)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| 7 | RB_RTC_LOAD_HI | RWA | Writing 1 will load the high word of the RTC counter, and will automatically clear it to 0 after loading. Load R32_RTC_TRIG (actually only the lower 14 bits) into R32_RTC_CNT_DAY. | 0 |
| 6 | RB_RTC_LOAD_LO | RWA | Writing 1 will load the low word of the RTC counter, and will automatically clear it to 0 after loading. Load the high 16 bits of R32_RTC_TRIG to R16_RTC_CNT_2S; load R32_RTC_TRIG The lower 16 bits are loaded into R16_RTC_CNT_32K. | 0 |
| 5 | RB_RTC_TRIG_EN | RWA | RTC trigger mode enable bit:<br>1: enable; 0: Disabled. | 0 |
| 4 | RB_RTC_TMR_EN | RWA | RTC timing mode enable bit:<br>1: enable; 0: Disabled. | 0 |
| 3 | RB_RTC_IGNORE_B0 | RWA | Ignore the lowest bit of the comparison match value in trigger mode:<br>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:<br>000: 0.125S; 001: 0.25S;<br>010: 0.5S; 011: 1S;<br>100: 2S; 101: 4S;<br>110: 8S; 111: 16S. | 010b |

RTC Trigger value send Register (R32_RTC_TRIG)

| Bit | *name* | access | description | Reset value |
|---|---|---|---|---|
| [31:0] | R32_RTC_TRIG | RWA | The preset matching value in RTC trigger mode, its high The 16 bits and the lower 16 bits are matched with R16_RTC_CNT_2S and R16_RTC_CNT_32K respectively. Cooperate with RB_RTC_LOAD_LO and RB_RTC_LOAD_HI to update the current value of the RTC counter. | 0000h |

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

RTC is based on the 32768Hz count value register (R16_ RTC_CN T_32K)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_RTC_CNT_32K | RO | The RTC is based on the 32768Hz count value register. | XXXXh |

RTC count value register with 2S as the unit (R16_RTC_CNT_2S)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_RTC_CNT_2S | RO | RTC is the current count value in 2S units. | XXXXh |

RTC Take the sky as a single Bit Count value register (R32_RTC_CN T_DAY)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [31:16] | Reserved | RO | Reserved. | 0000h |
| [15:14] | Reserved | RO | Reserved. | 00b |
| [13:0] | R32_RTC_CNT_DAY | RO | The current count value of RTC in days. | XXXXXXXXXXXXXXb |

## 6.4 Function description and configuration

### 6.4.1 RTC counter initialization

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

### 6.4.2 Switch RTC clock source to LSE crystal oscillator

(1) Confirm that the GPIO pins where X32KI and X32KO are located are not set as outputs, and there are no pull-up and pull-down resistors, only crystals; (2) Configure the R8_CK32K_CONFIG register, set RB_CLK_XT32K_PON to 1, and turn on the external 32KHz crystal oscillator; ( 3) It is recommended to set RB_XT32K_I_TUNE to the maximum first, and wait for the crystal oscillator to stabilize (about hundreds of mS) before changing to the rated current; (4) Configure the R8_CK32K_CONFIG register and set RB_CLK_OSC32K_XT to 1, requiring the clock source to switch to the crystal oscillator; (5) Wait for at least half a 32KHz clock cycle, usually 16uS, to complete the switching of the clock source.

### 6.4.3 RTC timing function

(1) Configure the 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 the RTC timing function;
(2) After the timing period is reached, the RTC timing activation flag RB_RTC_TMR_FLAG and interrupt will be generated. Check the R8_RTC_FLAG_CTRL register and set the RB_RTC_TMR_CLR to clear the flag.

### 6.4.4 RTC trigger function

(1). Set the target matching value in the R32_RTC_TRIG register, calculation and operation steps:

Add the interval time DelayTime (unit S) to the current time R32_RTC_CNT_32K (high 16 bits R16_RTC_CNT_2S and low 16 bits R16_RTC_CNT_32K) to calculate the target time value, T32 = R32_RTC_CNT_32K + DelayTime * 32768,

If (T32 & 0xFFFF)! = 0, then T32 = T32 + 0x10000, write T32 into the

R32_RTC_TRIG register to complete the matching value setting;

(2) Configure the R8_RTC_MODE_CTRL register, set RB_RTC_TRIG_EN to 1, and turn on the RTC trigger function;
(3) When the current RTC count values R16_RTC_CNT_2S and R16_RTC_CNT_32K match the preset high and low 16 bits of R32_RTC_TRIG respectively, the RTC trigger activation flag RB_RTC_TRIG_FLAG and interrupt are generated, and RB_RTC_TRIG_CLR can be cleared. (4) If the RTC has been time-calibrated, it can support the target absolute time trigger, which is calculated by the target year/month/day/hour/minute/second/millisecond Get the target time value T32, the other steps are the same as above. For details, refer to the evaluation board example program.

### 6.4.5 Use HSE to calibrate internal 32K clock LSI

Reference evaluation board example program

# Chapter 7 General I/O and Multiplex Functions

## 7.1 Introduction to GPIO

The chip provides 2 groups of GPIO ports PA and PB, a total of 40 general-purpose input and output pins, some pins have interrupt, multiplexing and mapping functions.

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

In the PA port, the PA[0] ～PA[15] bits are valid, corresponding to the 16 GPIO pins on the chip; in the PB port, the PB[0]～PB[23] bits are valid, corresponding to the 24 GPIO pins on the chip; Among them, 32 I/O pins PA[0]～PA[15], PB[0]～PB[15] have interrupt function and can realize sleep wake-up function.

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

The following figure is a block diagram of the internal structure of GPIO:

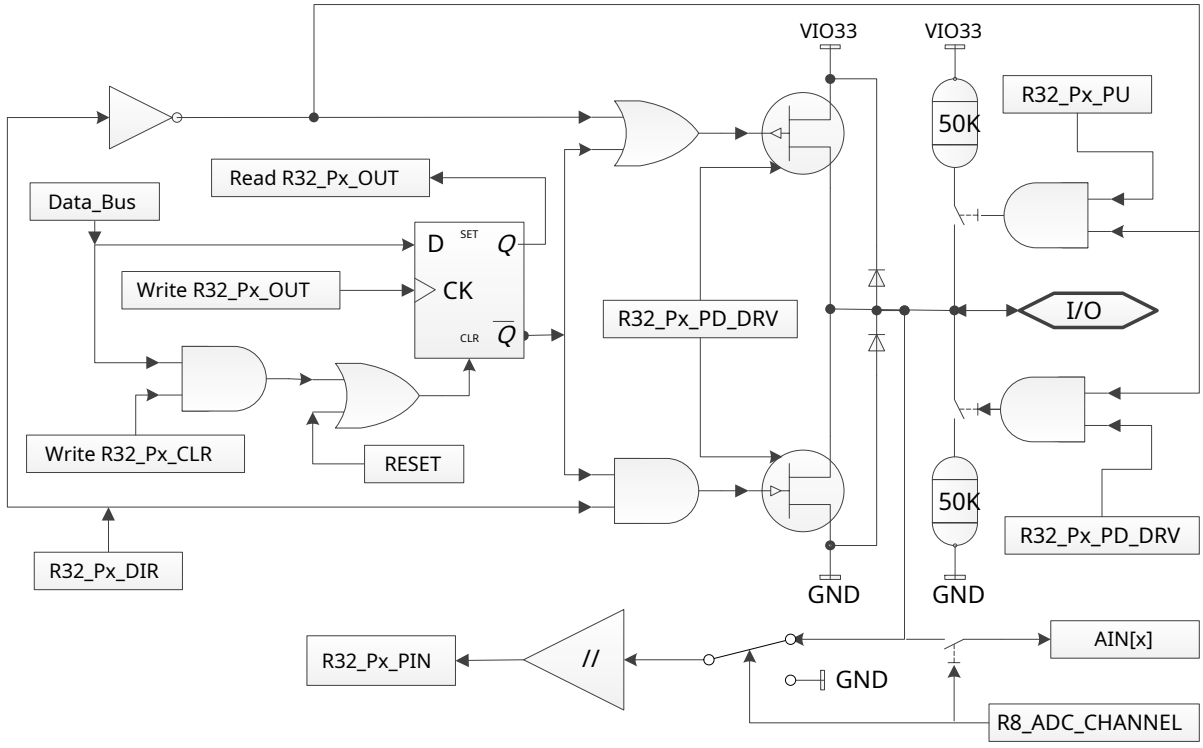



Figure 7-1 Block diagram of I/O internal structure

## 7.2 External interrupt/wake-up

The 32 I/O pins of the chip PA[0] ～PA[15], PB[0]～PB[15]) have interrupt function and can realize sleep wakeup. In order to use external interrupts, the port bits must be configured in input mode. And provide 4 trigger modes: high level, low level, rising Edge, falling edge.

The wake-up function needs to turn on the interrupt enable R16_Px_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 Multiplexing and remapping of GPIO

7.3.1 Multiplexing function

Some I/O pins have multiplexing functions. After power-on, all I/O pins are general-purpose I/O functions by default. After each functional module is enabled, the corresponding original GPIO pins are configured to the corresponding functions of their respective functional modules. Pin.

If a pin is multiplexed with multiple functions, and multiple functions are turned on, please refer to section 1.2 for the priority order of multiplexing functions The order of functions in the "Alternative Function" list of the pin description

For example: PA0 pin is multiplexed with SCK1/SLVA/LED0, the clock function of SPI1 has priority, and the LED screen interface serial data output 0 has the lowest function. In this way, among multiple multiplexed functions, the unused pins with the lowest function priority can be enabled with relatively higher priority. Multiplexing functions.

The following table lists some of the GPIO configurations used for the function pins of the peripheral modules.

Table 7-1 Timer x

| TMR0/1/2/3 pins | Function configuration | GPIO configuration |
|---|---|---|
| TMRx | Input capture channel x | Input (floating input/pull-up input/pull-down input) |
| | Output PWM channel x | Push-pull output |

Table 7-2 UARTx

| UART0/1/2/3 pins | Function configuration | GPIO configuration |
|---|---|---|
| TXDx | Serial port send x | Push-pull output |
| RXDx | Serial port x | Pull up input (recommended) or floating input |
| RTS, DTR | MODEM signal output or RS485 control | Push-pull output |
| CTS, DSR, RI, DCD | MODEM signal input | Pull up input (recommended) or floating input |

Table 7-3 SPIx

| SPI0/1 pins | Function configuration | GPIO configuration |
|---|---|---|
| SCKx | Master mode clock output | Push-pull output |
| | Slave mode clock input | Input (floating input/pull-up input/pull-down input) |
| MOSIx | Full duplex mode-main mode | Push-pull output |
| | Full duplex mode-slave mode | Input (floating input/pull-up input/pull-down input) |
| | Half-duplex mode-main mode | Not used, can be used as general I/O |
| | Half-duplex mode-slave mode | Not used, can be used as general I/O |
| MISOx | Full duplex mode-main mode | Input (floating input/pull-up input/pull-down input) |
| | Full duplex mode-slave mode | Input (pull-up is recommended, automatically cut to push-pull output after chip selection) Or push-pull output (forbidden to be used in bus connection mode) |
| | Half-duplex mode-main mode | Input or push-pull output, manual switching |
| | Half-duplex mode-slave mode | Input (pull-up is recommended, automatically cut to push-pull output after chip selection) |
| SCS | Main mode chip select output | Push-pull output (other pins can be used instead) |
| | Input from mode chip select | Pull up input (recommended) or floating input |

Table 7-4 ADC

| ADC sampling channel pin | Function configuration | GPIO configuration |
|---|---|---|
| AINx | Analog-to-digital conversion input channel | Floating input |

Table 7-5 USB

| USB signal pin | Function configuration | GPIO configuration |
|---|---|---|
| USB_DM | Connect to internal USB transceiver | Floating input |
| USB_DP | Connect to internal USB transceiver | Floating input |

Table 7-6 Ethernet

| Ethernet signal pin | Function configuration | GPIO configuration |
|---|---|---|
| ET+/ET- | Connect to internal Ethernet transceiver | Floating input |
| ER+/ER- | Connect to internal Ethernet transceiver | Floating input |

### 7.3.2 Function pin remapping

In order to optimize the utilization of peripheral functions at the same time, you can set the R16_PIN_ALTERNATE function pin to remap the register

The device remaps some functional pins to other pins.

Table 7-7 Multiplexed function remapping pins

| Peripheral function pins | Default GPIO pin | GPIO pin to be remapped |
|---|---|---|
| SPI0 | PA[12]/PA[13]/PA[14]/PA[15] | PB[12]/PB[13]/PB[14]/PB[15] |
| RXD3/TXD3 | PA[4]/PA[5] | PB[20]/PB[21] |
| RXD2/TXD2 | PA[6]/PA[7] | PB[22]/PB[23] |
| RXD1/TXD1 | PA[8]/PA[9] | PB[8]/PB[9] |
| RXD0/TXD0 | PB[4]/PB[7] | PA[15]/PA[14] |
| TMR3/PWM3/CAP3 | PA[2] | PB[18] |
| TMR2/PWM2/CAP2 | PA[11] | PB[11] |
| TMR1/PWM1/CAP1 | PA[10] | PB[10] |
| TMR0/PWM0/CAP0 | PA[3] | PB[19] |

## 7.4 Register description

Table 7-8 List of GPIO related registers

| name | address | description | Reset value |
|---|---|---|---|
| R16_PIN_ALTERNATE | 0x40001018 | Function pin remapping register | 0x0000 |
| R16_PIN_ANALOG_IE | 0x4000101A | Peripheral analog pin configuration register | 0x0000 |
| R16_PA_INT_EN | 0x40001090 | PA port interrupt enable register | 0x0000 |
| R16_PB_INT_EN | 0x40001092 | PB port interrupt enable register | 0x0000 |
| R16_PA_INT_MODE | 0x40001094 | PA port interrupt mode configuration register | 0x0000 |
| R16_PB_INT_MODE | 0x40001096 | PB port interrupt mode configuration register | 0x0000 |
| R16_PA_INT_IF | 0x4000109C | PA port interrupt flag register | 0x0000 |
| R16_PB_INT_IF | 0x4000109E | PB port interrupt flag register | 0x0000 |
| R32_PA_DIR | 0x400010A0 | PA port direction configuration register | 0x00000000 |
| R32_PA_PIN | 0x400010A4 | PA port pin input register | 0x0000XXXX |
| R32_PA_OUT | 0x400010A8 | PA port data output register | 0x00000000 |
| R32_PA_CLR | 0x400010AC | PA port data reset register | 0x00000000 |
| R32_PA_PU | 0x400010B0 | PA port pull-up resistor configuration register | 0x00000000 |
| R32_PA_PD_DRV | 0x400010B4 | PA port pull-down/drive configuration register | 0x00000000 |
| R32_PB_DIR | 0x400010C0 | PB port direction configuration register | 0x00000000 |
| R32_PB_PIN | 0x400010C4 | PB port pin input register | 0x00XXXXXX |
| R32_PB_OUT | 0x400010C8 | PB port data output register | 0x00000000 |
| R32_PB_CLR | 0x400010CC | PB port data reset register | 0x00000000 |
| R32_PB_PU | 0x400010D0 | PB port pull-up resistor configuration register | 0x00000000 |
| R32_PB_PD_DRV | 0x400010D4 | PB port pull-down/drive configuration register | 0x00000000 |

Work Capable of pin remapping Shoot Register (R16_PIN_ALT E RNATE)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| [15:12] | Reserved | RO | Reserved. | 0000b |
| [11:9] | Reserved | RO | Reserved. | 000b |
| 8 | RB_PIN_SPI0 | RW | SPI0 function pin mapping selection bit:<br>1: SCK0_/SCS_/MOSI_/MISO_ is mapped to PB[12]/PB[13]/PB[14]/PB[15];<br>0: SCK0/SCS/MOSI/MISO is mapped to PA[12]/PA[13]/PA[14]/PA[15]. | 0 |
| 7 | RB_PIN_UART3 | RW | UART3 function pin mapping selection bit:<br>1: RXD3_/TXD3_ is mapped to PB[20]/PB[21];<br>0: RXD3/TXD3 is mapped to PA[4]/PA[5]. | 0 |
| 6 | RB_PIN_UART2 | RW | UART2 function pin mapping selection bit:<br>1: RXD2_/TXD2_ is mapped to PB[22]/PB[23];<br>0: RXD2/TXD2 is mapped to PA[6]/PA[7]. | 0 |
| 5 | RB_PIN_UART1 | RW | UART1 function pin mapping selection bit:<br>1: RXD1_/TXD1_ is mapped to PB[8]/PB[9];<br>0: RXD1/TXD1 is mapped to PA[8]/PA[9]. | 0 |
| 4 | RB_PIN_UART0 | RW | UART0 function pin mapping selection bit:<br>1: RXD0_/TXD0_ is mapped to PA[15]/PA[14];<br>0: RXD0/TXD0 is mapped to PB[4]/PB[7]. | 0 |
| 3 | RB_PIN_TMR3 | RW | TMR3 function pin mapping selection bit:<br>1: TMR3_/PWM3_/CAP3_ is mapped to PB[18];<br>0: TMR3/PWM3/CAP3 is mapped to PA[2]. | 0 |
| 2 | RB_PIN_TMR2 | RW | TMR2 function pin mapping selection bit:<br>1: TMR2_/PWM2_/CAP2_ is mapped to PB[11];<br>0: TMR2/PWM2/CAP2 is mapped to PA[11]. | 0 |
| 1 | RB_PIN_TMR1 | RW | TMR1 function pin mapping selection bit:<br>1: TMR1_/PWM1_/CAP1_ is mapped to PB[10];<br>0: TMR1/PWM1/CAP1 is mapped to PA[10]. | 0 |
| 0 | RB_PIN_TMR0 | RW | TMR0 function pin mapping selection bit:<br>1: TMR0_/PWM0_/CAP0_ is mapped to PB[19];<br>0: TMR0/PWM0/CAP0 is mapped to PA[3]. | 0 |

outer Set analog pin Match Set the register (R16_PIN_A N ALOG_I E)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| 15 | RB_PIN_ADC4_5_IE | RW | ADC/TKEY 4/5 channel pin digital input disabled:<br>1: Turn off PA14-15 digital input to save power consumption;<br>0: Turn on digital input. | 0 |
| 14 | RB_PIN_ADC2_3_IE | RW | ADC/TKEY 2/3 channel pin digital input disabled:<br>1: Turn off PA12-13 digital input to save power consumption;<br>0: Turn on digital input. | 0 |
| 13 | RB_PIN_XT32K_IE | RW | 32KHz crystal oscillator LSE pin digital input is disabled: 1: Turn off PA10-11 digital input to save power consumption;<br>0: Turn on digital input. | 0 |
| 12 | RB_PIN_ADC12_13_IE | RW | ADC/TKEY 12/13 channel pin digital input disabled:<br>1: Turn off PA8-9 digital input to save power consumption;<br>0: Turn on digital input. | 0 |
| 11 | RB_PIN_ADC10_11_IE | RW | ADC/TKEY 10/11 channel pin digital input disabled:<br>1: Turn off PA6-7 digital input to save power consumption;<br>0: Turn on digital input. | 0 |

| | | | | |
|---|---|---|---|---|
| 10 | RB_PIN_ADC0_1_IE | RW | ADC/TKEY 0/1 channel pin digital input disabled:<br>1: Turn off PA4-5 digital input to save power consumption;<br>0: Turn on digital input. | 0 |
| 9 | RB_PIN_ADC6_7_IE | RW | ADC/TKEY 7/6 channel pin digital input disabled:<br>1: Turn off PA2-3 digital input to save power consumption;<br>0: Turn on digital input. | 0 |
| 8 | RB_PIN_ADC8_9_IE | RW | ADC/TKEY 9/8 channel pin digital input disabled:<br>1: Turn off PA0-1 digital input to save power consumption;<br>0: Turn on digital input. | 0 |
| 7 | RB_PIN_USB_IE | RW | USB pin enable:<br>1: PB10-11 is the USB communication pin; 0:<br>PB10-11 is not used for USB communication. | 0 |
| 6 | RB_PIN_ETH_IE | RW | ETH pin enable:<br>1: PB12-15 is the ETH communication pin. If the LCD is not enabled at all, you can set RB_PIN_SEG12_15_IE<br>Set to 1 to turn off the digital input, which can save power consumption;<br>0: PB12-15 is not used for ETH communication. | 0 |
| 5 | RB_PIN_SEG20_23_IE | RW | LCD 20-23 segment pin enable/digital input disable:<br>1: PB20-23 is segment drive, close the digital input;<br>0: Turn on the digital input and turn off the LCD segment drive. | 0 |
| 4 | RB_PIN_SEG16_19_IE | RW | LCD 16-19 segment pin enable/digital input disable:<br>1: PB16-19 is segment drive, close the digital input;<br>0: Turn on the digital input and turn off the LCD segment drive. | 0 |
| 3 | RB_PIN_SEG12_15_IE | RW | LCD 12-15 segment pin enable/digital input disable:<br>1: PB12-15 is segment drive, close the digital input;<br>0: Turn on the digital input and turn off the LCD segment drive. | 0 |
| 2 | RB_PIN_SEG8_11_IE | RW | LCD 8-11 segment pin enable/digital input disable:<br>1: PB8-11 is segment drive, close the digital input;<br>0: Turn on the digital input and turn off the LCD segment drive. | 0 |
| 1 | RB_PIN_SEG4_7_IE | RW | LCD 4-7 segment pin enable/digital input disable:<br>1: PB4-7 is segment drive, close the digital input;<br>0: Turn on the digital input and turn off the LCD segment drive. | 0 |
| 0 | RB_PIN_SEG0_3_IE | RW | LCD 0-3 segment pin enable/digital input disable:<br>1: PB0-3 is segment drive, close the digital input;<br>0: Turn on the digital input and turn off the LCD segment drive. | 0 |

Note: If the pin is used for analog functions (ADC/TouchKey/ETH), it is recommended to turn off the digital input function of this pin, that is, set the digital The input is disabled, which can reduce power consumption and help reduce interference.

PA Port interrupt Make Capability register (R16_PA_INT_ EN)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_PA_INT_EN | RW | PA pin interrupt enable bit:<br>1: Enable the corresponding interrupt;<br>0: Disable the corresponding interrupt. | 0000h |

PB Port interrupt Make Function register (R16_PB_INT_ EN)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_PB_INT_EN | RW | PB pin interrupt enable bit:<br>1: Enable the corresponding interrupt;<br>0: Disable the corresponding interrupt. | 0000h |

PA Port interrupt mold Configuration register (R16_PA_INT_MO DE)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_PA_INT_MODE | RW | PA pin interrupt mode selection bit:<br>1: Edge trigger; 0: Level trigger. | 0000h |

PB Port interrupt mold Configuration register (R16_PB_INT_MO DE)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_PB_INT_MODE | RW | PB pin interrupt mode selection bit:<br>1: Edge trigger; 0: Level trigger. | 0000h |

PA Port interrupt Mark Log 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: there is an interrupt; 0: No interruption. | 0000h |

PB Port interrupt Mark Log register (R16_PB_INT_ IF)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_PB_INT_IF | RW1 | PB pin interrupt flag bit, write 1 to clear: 1: there is an interrupt; 0: No interruption. | 0000h |

PA Port direction Match Set register (R32_PA_DIR)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [31:16] | Reserved | RO | Reserved. | 0000h |
| [15:8] | R8_PA_DIR_1 | RW | Current input and output direction configuration of PA pin:<br>1: The pin is in output mode;<br>0: The pin is in input mode. | 00h |
| [7:0] | R8_PA_DIR_0 | RW | | 00h |

PA Port pin lose Into the register (R32_PA_PIN)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [31:16] | Reserved | RO | Reserved. | 0000h |
| [15:8] | R8_PA_PIN_1 | RO | PA pin current level state (only in R32_PA_DIR When the corresponding bit is 0, the bit value is valid):<br>1: Pin input high level;<br>0: Pin input low level. | XXh |
| [7:0] | R8_PA_PIN_0 | RO | | XXh |

PA Port data lose Out register (R32_PA_OUT)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [31:16] | Reserved | RO | Reserved. | 0000h |
| [15:8] | R8_PA_OUT_1 | RW | When the corresponding bit of the direction register R32_PA_DIR is 1: Control the PA pin output level status:<br>1: output high level; 0: output low level. | 00h |
| [7:0] | R8_PA_OUT_0 | RW | When the corresponding bit of the direction register R32_PA_DIR is 0: Control the PA pin interrupt polarity selection:<br>1: High level/rising edge; 0: Low level/falling edge. | 00h |

PA port data reset register (R32_PA_CLR)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [31:16] | Reserved | RO | Reserved. | 0000h |
| [15:8] | R8_PA_CLR_1 | WZ | PA data register reset control: | 00h |
| [7:0] | R8_PA_CLR_0 | WZ | 1: R32_PA_OUT corresponding bit data is cleared to 0;<br>0: No effect. | 00h |

PA Port pull up Electricity Resistance configuration register (R32_PA_ PU)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [31:16] | Reserved | RO | Reserved. | 0000h |
| [15:8] | R8_PA_PU_1 | RW | PA pin pull-up resistor enable control: | 00h |
| [7:0] | R8_PA_PU_0 | RW | 1: Enable the pull-up resistor; 0: Disable the pull-up resistor. | 00h |

PA Port pull down/ drive Dynamic configuration register (R32_PA_PD_DR V)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [31:16] | Reserved | RO | Reserved. | 0000h |
| [15:8] | R8_PA_PD_DRV_1 | RW | When the corresponding bit of the direction register R32_PA_DIR is 0: PA pin pull-down resistor enable control:<br>1: Enable the pull-down resistor; 0: Turn off the pull-down resistor | 00h |
| [7:0] | R8_PA_PD_DRV_0 | RW | When the corresponding bit of the direction register R32_PA_DIR is 1: PA pin current drive capability selection:<br>1: 20mA level; 0: 5mA level. | 00h |

PB Port direction Match Set register (R32_PB_DIR)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [31:24] | Reserved | RO | Reserved. | 00h |
| [23:16] | R8_PB_DIR_2 | RW | PB pin current input and output direction configuration: | 00h |
| [15:8] | R8_PB_DIR_1 | RW | 1: The pin is in output mode; | 00h |
| [7:0] | R8_PB_DIR_0 | RW | 0: The pin is in input mode. | 00h |

PB Port pin lose Into the register (R32_PB_PIN)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [31:24] | Reserved | RO | Reserved. | 00h |
| [23:16] | R8_PB_PIN_2 | RO | PB pin current level state (only in R32_PB_DIR | XXh |
| [15:8] | R8_PB_PIN_1 | RO | When the corresponding bit is 0, the bit value is valid): | XXh |
| [7:0] | R8_PB_PIN_0 | RO | 1: Pin input high level;<br>0: Pin input low level. | XXh |

PB Port data lose Out register (R32_PB_OUT)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [31:24] | Reserved | RO | Reserved. | 00h |
| [23:16] | R8_PB_OUT_2 | RW | When the corresponding bit of the direction register R32_PB_DIR is 1: Control the output level status of the PB pin:<br>1: output high level; 0: output low level. | 00h |
| [15:8] | R8_PB_OUT_1 | RW | When the corresponding bit of the direction register R32_PB_DIR is 1: Control the output level status of the PB pin:<br>1: output high level; 0: output low level. | 00h |
| [7:0] | R8_PB_OUT_0 | RW | When the corresponding bit of the direction register R32_PB_DIR is 0: | 00h |

| | | | | |
|---|---|---|---|---|
| | | | Control PB pin interrupt polarity selection:<br>1: High level/rising edge; 0: Low level/falling edge. | |

PB Port data complex Bit register (R32_PB_CLR)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [31:24] | Reserved | RO | Reserved. | 00h |
| [23:16] | R8_PB_CLR_2 | WZ | PB data register reset control:<br>1: R32_PB_OUT corresponding bit data is cleared to 0;<br>0: No effect. | 00h |
| [15:8] | R8_PB_CLR_1 | WZ | | 00h |
| [7:0] | R8_PB_CLR_0 | WZ | | 00h |

PB Port pull up Electricity Resistance configuration register (R32_PB_ PU)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [31:24] | Reserved | RO | Reserved. | 00h |
| [23:16] | R8_PB_PU_2 | RW | PB pin pull-up resistor enable control:<br>1: Enable the pull-up resistor; 0: Disable the pull-up resistor. | 00h |
| [15:8] | R8_PB_PU_1 | RW | | 00h |
| [7:0] | R8_PB_PU_0 | RW | | 00h |

PB Port pull down/ drive Dynamic configuration register (R32_PB_PD_DRV)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [31:24] | Reserved | RO | Reserved. | 00h |
| [23:16] | R8_PB_PD_DRV_2 | RW | When the corresponding bit of the direction register R32_PB_DIR is 0: PB pin pull-down resistor enable control:<br>1: Enable the pull-down resistor; 0: Turn off the pull-down resistor<br>When the corresponding bit of the direction register R32_PB_DIR is 1: PB pin current drive capability selection:<br>1: 20mA level; 0: 5mA level. | 00h |
| [15:8] | R8_PB_PD_DRV_1 | RW | | 00h |
| [7:0] | R8_PB_PD_DRV_0 | RW | | 00h |

## 7.5 GPIO pin mode configuration

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

Table 7-9 Port configuration table

| mode | R32_Px_DIR | R32_Px_PU | R32_Px_PD_DRV |
|---|---|---|---|
| Floating input/high impedance input/analog input | 0 | 0 | 0 |
| Input with pull-up resistor | 0 | 1 | 0 |
| Input with pull-down resistor | 0 | 0 | 1 |
| Push-pull output, driving capability 5mA level | 1 | X | 0 |
| Push-pull output, drive capability 20mA level | 1 | X | 1 |

# Chapter 8 General Timer TMRx

## 8.1 Introduction to TMRx

The chip provides 4 26-bit timers, TMR0, TMR1, TMR2 and TMR3, the longest timing time is 2^26 clock cycles. It is suitable for many occasions, including measuring the pulse length of the input signal (input capture) or generating the output waveform (PWM) , And TMR1 And TMR2 supports DMA function. Each timer is completely independent and can operate simultaneously.

### 8.1.1 Main features

l Four 26-bit timers, each timer has a maximum timing time of 2^26 clock cycles.

l Support timer interrupts, among which TMR1 and TMR2 support DMA and interrupts.

l Support the capture function to measure the length or period of the input pulse.

l The capture function can be set as level change capture and high or low level hold time capture function.

l Supports 26-bit PWM function, which can dynamically adjust the PWM duty cycle setting.

## 8.2 Register description

Table 8-1 TMR0 related register list

| name | address | description | Reset value |
|---|---|---|---|
| R8_TMR0_CTRL_MOD | 0x40002000 | Mode setting register | 0x02 |
| R8_TMR0_INTER_EN | 0x40002002 | Interrupt enable register | 0x00 |
| R8_TMR0_INT_FLAG | 0x40002006 | Interrupt flag register | 0x00 |
| R8_TMR0_FIFO_COUNT | 0x40002007 | FIFO count register | 0x0X |
| R32_TMR0_COUNT | 0x40002008 | Current count value register | 0x0XXXXXXX |
| R32_TMR0_CNT_END | 0x4000200C | Counting end value setting register | 0x0XXXXXXX |
| R32_TMR0_FIFO | 0x40002010 | FIFO register | 0x0XXXXXXX |

Table 8-2 TMR1 related register list

| name | address | description | Reset value |
|---|---|---|---|
| R8_TMR1_CTRL_MOD | 0x40002400 | Mode setting register | 0x02 |
| R8_TMR1_CTRL_DMA | 0x40002401 | DMA control register | 0x00 |
| R8_TMR1_INTER_EN | 0x40002402 | Interrupt enable register | 0x00 |
| R8_TMR1_INT_FLAG | 0x40002406 | Interrupt flag register | 0x00 |
| R8_TMR1_FIFO_COUNT | 0x40002407 | FIFO count register | 0x0X |
| R32_TMR1_COUNT | 0x40002408 | Current count value register | 0x0XXXXXXX |
| R32_TMR1_CNT_END | 0x4000240C | Counting end value register | 0x0XXXXXXX |
| R32_TMR1_FIFO | 0x40002410 | FIFO register | 0x0XXXXXXX |
| R16_TMR1_DMA_NOW | 0x40002414 | DMA current buffer address | 0x0000XXXX |
| R16_TMR1_DMA_BEG | 0x40002418 | DMA start buffer address | 0x0000XXXX |
| R16_TMR1_DMA_END | 0x4000241C | DMA end buffer address | 0x0000XXXX |

Table 8-3 TMR2 related register list

| name | address | description | Reset value |
|---|---|---|---|
| R8_TMR2_CTRL_MOD | 0x40002800 | Mode setting register | 0x02 |
| R8_TMR2_CTRL_DMA | 0x40002801 | DMA control register | 0x00 |
| R8_TMR2_INTER_EN | 0x40002802 | Interrupt enable register | 0x00 |
| R8_TMR2_INT_FLAG | 0x40002806 | Interrupt flag register | 0x00 |

| | | | |
|---|---|---|---|
| R8_TMR2_FIFO_COUNT | 0x40002807 | FIFO count register | 0x0X |
| R32_TMR2_COUNT | 0x40002808 | Current count value register | 0x0XXXXXXX |
| R32_TMR2_CNT_END | 0x4000280C | Counting end value register | 0x0XXXXXXX |
| R32_TMR2_FIFO | 0x40002810 | FIFO register | 0x0XXXXXXX |
| R16_TMR2_DMA_NOW | 0x40002814 | DMA current buffer address | 0x0000XXXX |
| R16_TMR2_DMA_BEG | 0x40002818 | DMA start buffer address | 0x0000XXXX |
| R16_TMR2_DMA_END | 0x4000281C | DMA end buffer address | 0x0000XXXX |

Table 8-4 TMR3 related register list

| name | address | description | Reset value |
|---|---|---|---|
| R8_TMR3_CTRL_MOD | 0x40002C00 | Mode setting register | 0x02 |
| R8_TMR3_INTER_EN | 0x40002C02 | Interrupt enable register | 0x00 |
| R8_TMR3_INT_FLAG | 0x40002C06 | Interrupt flag register | 0x00 |
| R8_TMR3_FIFO_COUNT | 0x40002C07 | FIFO count register | 0x0X |
| R32_TMR3_COUNT | 0x40002C08 | Current count value register | 0x0XXXXXXX |
| R32_TMR3_CNT_END | 0x40002C0C | Counting end value setting register | 0x0XXXXXXX |
| R32_TMR3_FIFO | 0x40002C10 | FIFO register | 0x0XXXXXXX |

mold Style setting Save (R8_TMRx_CTRL_MOD) (x=0/1 /2/3)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [7:6] | RB_TMR_CAP_EDGE | RW | In the capture mode, select the capture trigger mode:<br>00: do not trigger;<br>01: Capture the time between any edge changes;<br>10: Capture the time between the falling edge and the falling edge;<br>11: Capture the time between rising edge and rising edge.<br>In the counting mode, select the counting edge:<br>00: no sampling count;<br>01: Sampling to any edge count;<br>10: Count from sampling to falling edge;<br>11: Sampling to rising edge count. | 00b |
| [7:6] | RB_TMR_PWM_REPEAT | RW | In PWM mode, select the data repetition method:<br>00: Repeat 1 time; 01: Repeat 4 times; 10:<br>Repeat 8 times; 11: Repeat 16 times. | 00b |
| 5 | Reserved | RO | Reserved. | 0 |
| 4 | RB_TMR_CAP_COUNT | RW | RB_TMR_MODE_IN=1 Input mode sub-mode:<br>1: Counting mode; 0: Capture mode. | 0 |
| 4 | RB_TMR_OUT_POLAR | RW | In PWM mode, the output polarity setting bit:<br>1: The default high level, low level is valid;<br>0: Default low level, high level valid. | 0 |
| 3 | RB_TMR_OUT_EN | RW | Timer output enable bit:<br>1: Output enable; 0: Output disable. | 0 |
| 2 | RB_TMR_COUNT_EN | RW | Timer counting enable bit:<br>1: Enable counting; 0: Stop counting. | 0 |
| 1 | RB_TMR_ALL_CLEAR | RW | Clear the timer's FIFO/counter/interrupt flag:<br>1: Forced clearing and clearing; 0: Uncleared. | 1 |
| 0 | RB_TMR_MODE_IN | RW | Timer mode setting bit:<br>1: Input mode (capture mode or counting mode) ;<br>0: Timing mode or PWM mode. | 0 |

in Disable enable Save (R8_TMRx_INTER_EN) (x=0/1 /2/3)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [7:5] | Reserved | RO | Reserved. | 000b |
| 4 | RB_TMR_IE_FIFO_OV | RW | FIFO overflow (capture mode FIFO full or PWM mode FIFO empty) interrupt enable bit:<br>1: Enable interrupt; 0: Disable interrupts. | 0 |
| 3 | RB_TMR_IE_DMA_END | RW | DMA end interrupt enable bit (only supported by TMR1/2): 1: enable interrupt; 0: Disable interrupts. | 0 |
| 2 | RB_TMR_IE_FIFO_HF | RW | FIFO used more than half (capture mode FIFO>=4 or PWM Mode FIFO<4) Interrupt enable bit:<br>1: Enable interrupt; 0: Disable interrupts. | 0 |
| 1 | RB_TMR_IE_DATA_ACT | RW | Data activation (capture mode means each time a new data is captured According to the data, the PWM mode index value trigger causes the effective End) Interrupt enable bit:<br>1: Enable interrupt; 0: Disable interrupts. | 0 |
| 0 | RB_TMR_IE_CYC_END | RW | End of period (capture mode refers to timeout, PWM mode And timing mode refers to the end of the period) interrupt enable bit:<br>1: Enable interrupt; 0: Disable interrupts. | 0 |

in Broken sign post Save (R8_TMRx_INT_FLAG) (x=0/1 /2/3)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [7:5] | Reserved | RO | Reserved. | 000b |
| 4 | RB_TMR_IF_FIFO_OV | RW1 | FIFO overflow (capture mode FIFO full or PWM mode FIFO empty) flag bit, write 1 to clear: 1: overflow; 0: No overflow. | 0 |
| 3 | RB_TMR_IF_DMA_END | RW1 | DMA complete flag bit, write 1 to clear: 1: Completed; 0: Not completed. | 0 |
| 2 | RB_TMR_IF_FIFO_HF | RW1 | FIFO used more than half (capture mode FIFO>=4 or PWM Mode FIFO<4) Flag bit, write 1 to clear: 1: FIFO has been used more than half;<br>0: FIFO is not more than half used. | 0 |
| 1 | RB_TMR_IF_DATA_ACT | RW1 | Data activation (capture mode means each time a new data is captured According to the data, the PWM mode index value trigger causes the effective End) flag bit, write 1 to clear:<br>1: Data generated/used; 0: Not generated/unused. | 0 |
| 0 | RB_TMR_IF_CYC_END | RW1 | End of period (capture mode refers to timeout, PWM mode And timing mode refers to the end of the period, counting mode refers to the counter Number overflow) flag bit, write 1 to clear:<br>1: Timeout/period end; 0: Not timeout/not ended. | 0 |

FIFO count register (R8_TMRx_FIFO_COUN T) (x=0/1/2/3)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_TMRx_FIFO_COUNT | RO | The data count in the FIFO, the maximum value is 8. | 0x0X |

when Previous count send Register (R32_TMRx_COUNT) (x=0/1 /2/3)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [31:0] | R32_TMRx_COUNT | RO | The current count value of the counter. | 0XXXXXXXh |

meter Set the final value of the number Set Register (R32_TMRx_CNT_ END) (x=0/1/2/3)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [31:0] | R32_TMRx_CNT_END | RW | In timer mode, the number of clocks in a timing cycle;<br>In PWM mode, the total number of clocks per PWM cycle;<br>In capture mode, the number of capture timeout clocks.<br>Only the lower 26 bits are valid, the maximum value is 67108863. In the counting mode, the final value of the count value is -2 (overflow).<br>Note: Write operations to this register will be automatically cleared<br>The value in the R32_TMRx_COUNT register. | 0XXXXXXXh |

FIFO register (R32_TMRx_FIFO) (x=0/1 /2/3)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [31:0] | R32_TMRx_FIFO | RO/ WO | FIFO data register, only the lower 26 bits are valid. | 0XXXXXXXh |

DMA Control post Save (R8_TMRx_CTRL_DMA) (x=1/2 ) (Only supported by TMR1/2)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [7:3] | Reserved | RO | Reserved. | 00000b |
| 2 | RB_TMR_DMA_LOOP | RW | DMA address cycle function enable bit:<br>1: Enable address loop; 0: Disable address loop.<br>If the DMA address loop is enabled, when the DMA address increases to the end address set, it will automatically loop to Set the first address. | 0 |
| 1 | Reserved | RO | Reserved. | 0 |
| 0 | RB_TMR_DMA_ENABLE | RW | DMA function enable bit:<br>1: Enable DMA; 0: Disable DMA. | 0 |

DMA Current delay Rush Area address (R16_TMRx_DMA_N OW) (x=1/2)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_TMRx_DMA_NOW | RO | The current address of the DMA data buffer.<br>Can be used to calculate the number of conversions, calculation method: COUNT=(TMR_DMA_NOW-TMR_DMA_BEG)/4. | XXXXh |

DMA Start slow Rush Area address (R16_TMRx_DMA_B EG) (x=1/2)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_TMRx_DMA_BEG | RW | DMA data buffer start address, only the lower 15 bits are valid, and the address must be aligned with 4 bytes. | XXXXh |

DMA End slow Rush Area address (R16_TMRx_DMA_E ND) (x=1/2)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_TMRx_DMA_END | RW | DMA data buffer end address (not included) , Only low 15 bits are valid, and the address must be aligned with 4 bytes. | XXXXh |

## 8.3 Function description and configuration

### 8.3.1 Timing and counting functions

Each timer of the chip supports the longest timing time 2^26 clock cycles, and executes the incremental counting mode. If the system clock week

If the period is 32MHz, the longest timing time is: 31.25nS*2^26≈2S. Each timer has an independent interrupt.

The timing function operation steps are as follows:

(1) Set RB_TMR_ALL_CLEAR, clear R32_TMRx_COUNT and interrupt flags, etc.; (2) Set the register R32_TMRx_CNT_END to the time value that needs timing;

The specific calculation method is: Time = Tsys * R32_TMRx_CNT_END;

(3), clear RB_TMR_ALL_CLEAR, clear RB_TMR_MODE_IN corresponding to the timing mode;

(4), optional steps, set the R8_TMRx_INTER_EN register, set RB_TMR_IE_CYC_END to turn on the timer period interrupt;

(5), set the RB_TMR_COUNT_EN of the R8_TMRx_CTRL_MOD register to start the timer count;

(6) When the count reaches R32_TMRx_COUNT equal to R32_TMRx_CNT_END, the timing is completed. At this time, the RB_TMR_IF_CYC_END of R8_TMRx_INT_FLAG will be set to 1, which can be cleared by writing 1.

The operation steps of the counting function are as follows:

(1) Set the direction of the I/O pin corresponding to the count as input;

(2) Set the final value of counting overflow in R32_TMRx_CNT_END;

(3) Configure R8_TMRx_CTRL_MOD, set RB_TMR_MODE_IN and RB_TMR_CAP_COUNT corresponding to the counting mode, clear the RB_TMR_ALL_CLEAR bit, select the sampling edge mode through RB_TMR_CAP_EDGE, set the RB_TMR_COUNT_EN function of R8_TMRx_CTRL_MOD to 1, enable;

(4), optional steps, if you need to enable interrupts, set the corresponding interrupt enable register bit;

(5) R32_TMRx_COUNT stores the current count value. Every time the count reaches the final count value, RB_TMR_IE_CYC_END will be set to 1 and R32_TMRx_COUNT will be cleared to 0. If the interrupt is turned on, a hardware interrupt will be triggered.

### 8.3.2 PWM function

Each timer of the chip has PWM function. Among them, the PWM function of TMR1 and TMR2 supports DMA data loading. The default output polarity of PWM can be set to high level or low level. The number of repeated output of the same data can be selected as 1, 4, 8 or 16 times. This repeat 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, and the duty ratio of PWM can be dynamically modified to imitate special waveforms.

The PWM operation steps are as follows:

(1) Set RB_TMR_ALL_CLEAR, clear and clear R32_TMRx_FIFO and interrupt flags, etc.;

(2) Set the PWM total period register R32_TMRx_CNT_END, the value should not be less than the value in the R32_TMRx_FIFO register; (3) Configure R8_TMRx_CTRL_MOD, clear RB_TMR_ALL_CLEAR, clear RB_TMR_MODE_IN corresponding to the PWM mode, and select the output polarity through RB_TMR_OUT_POLAR according to your needs. RB_TMR_PWM_REPEAT selects the number of repetitions of the same data;

(4) Set the data register R32_TMRx_FIFO, the minimum value is 0, corresponding to the duty cycle 0%, the maximum value is the same as R32_TMR_CNT_END, corresponding to the duty cycle 100%, the duty cycle calculation: R32_TMRx_FIFO/R32_TMRx_CNT_END. TMR1 and TMR2 can load continuous dynamic data through DMA, combined with the repeated output times of the same data, can imitate special waveforms;

(5) Configure R8_TMRx_CTRL_MOD, set RB_TMR_COUNT_EN to start counting and RB_TMR_OUT_EN to allow PWM output; (6) Set the corresponding I/O pin of PWM as output;

(7), optional steps, if you need to enable interrupts, set the corresponding interrupt enable register bit;

(8) After a PWM cycle is completed, if the interrupt is turned on, the hardware interrupt will be triggered when RB_TMR_IF_DATA_ACT or RB_TMR_IF_CYC_END is set;

(9). Updating the data in R32_TMRx_FIFO can dynamically change the duty cycle of PWM. It is recommended to load it through DMA.

For example: set the RB_TMR_OUT_POLAR bit to 0, R32_TMRx_FIFO to 6, R32_TMRx_CNT_END to 18, the basic timing diagram of PWM generation is shown below, and its duty cycle is: R32_TMRx_FIFO / R32_TMRx_CNT_END = 1/3

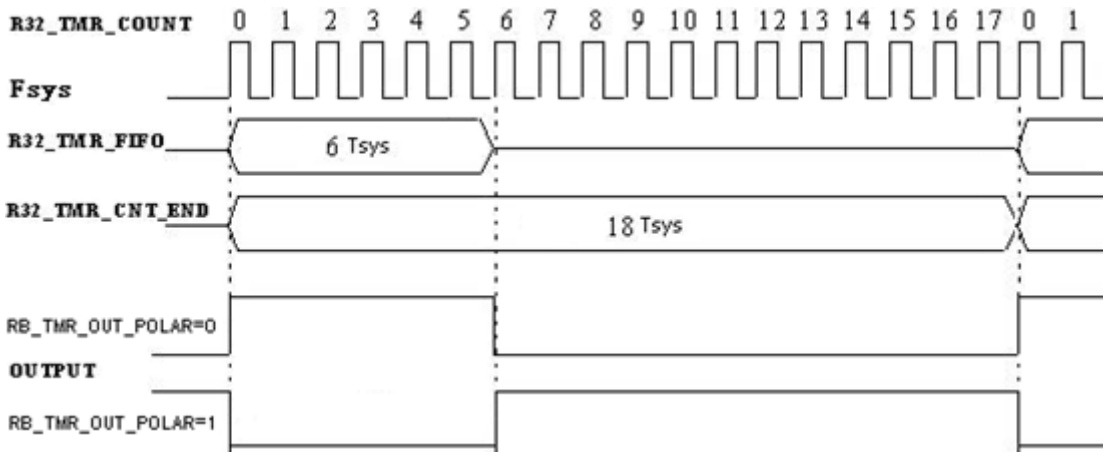


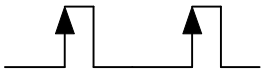

Figure 8-1 PWM output timing diagram

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

### 8.3.3 Capture function

Each timer of the chip has a capture function, and the capture functions of TMR1 and TMR2 support DMA data storage. The capture mode can select any edge trigger start to any edge trigger end, rising edge trigger start to rising edge trigger end, falling edge trigger on
There are three modes from 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<br>RB_TMR_CATCH_EDGE | Trigger method | Icon |
|---|---|---|
| 00 | Don't catch | no |
| 01 | Edge trigger<br>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. The most significant bit (bit 25) of the valid data in the data register R32_TMRx_FIFO is 1 to indicate that the captured is high, and 0 indicates that the captured is low. If the bit 25 of multiple groups of data is 1 (or 0) , It means that the width of the high (or low) level exceeds the timeout value, and it needs to be combined and accumulated.

In the falling edge to falling edge and rising edge to rising edge trigger mode, an input change cycle can be captured. Data register
The most significant bit (bit 25) of the valid data of R32_TMRx_FIFO is 0, which means normal sampling to one cycle, and 1 means that the input change cycle exceeds the timeout value R32_TMRx_CNT_END, and the latter group of data must be added to accumulate a single input change cycle.
The specific instructions are shown in the figure below:

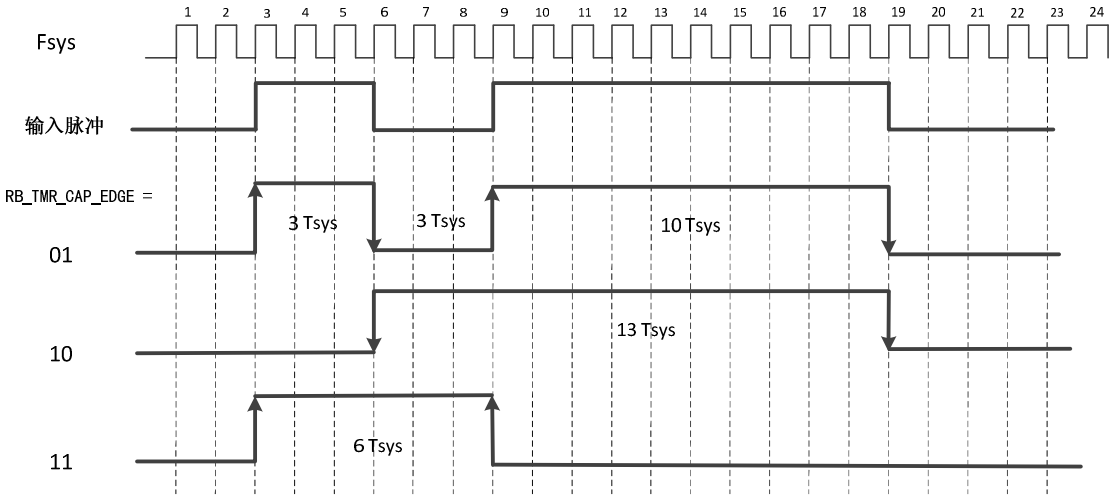



Figure 8-2 Count the capture period with the system clock

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

When RB_TMR_CATCH_EDGE=01b, it is set to edge-triggered sampling, and the sampling time width is 3, 3, and 10; when RB_TMR_CATCH_EDGE=10b, it is set to falling edge to falling edge sampling, and the sampling time width is 13; For 11b, set the rising edge to rising edge sampling, and the sampling time width is 6.

Capture mode operation steps:

(1) Set RB_TMR_ALL_CLEAR, clear and clear R32_TMRx_FIFO and interrupt flags, etc.; (2) Set the direction of the corresponding I/O pin to capture as input;

(3) Set a reasonable capture timeout time in R32_TMRx_CNT_END, which can be used to generate a timeout interrupt when the input signal does not change for a long time, or it can generate timeout data after the input signal does not change overtime (bit 25 of the data is 1, The lower 25 bits can be accumulated backwards);

(4) Configure R8_TMRx_CTRL_MOD, set RB_TMR_MODE_IN corresponding to the capture mode, select the edge mode of capture through RB_TMR_CAP_EDGE, set RB_TMR_COUNT_EN of R8_TMRx_CTRL_MOD to 1, enable counting;

(5), optional steps, if you need to enable interrupts, set the corresponding interrupt enable register bit;

(6) If you need to save the captured data in DMA mode, you need to set the register R16_TMRx_DMA_BEG to the first address of the buffer for storing the captured data, and set the register R16_TMRx_DMA_END to the end address (not included) of the buffer for storing the captured data, and set RB_TMR_DMA_ENABLE of R8_TMRx_CTRL_DMA is 1 to enable the DMA function; (7) Clear RB_TMR_ALL_CLEAR of R8_TMRx_CTRL_MOD to zero to start the capture function;

(8) Every time data is captured, RB_TMR_IF_DATA_ACT will be set to 1, if the interrupt is turned on, a hardware interrupt will be triggered. The captured data is stored in R32_TMRx_FIFO by default. If DMA is turned on, the captured data will be automatically stored in the DMA setting. In the data buffer.

# Chapter 9 Universal Asynchronous Receiver Transmitter UART

## 9.1 Introduction to UART

The chip provides 4 sets of full-duplex asynchronous serial ports, UART0/1/2/3. Supports full-duplex and half-duplex serial communication. UART0 provides the sending status pin for switching RS485, and supports MODEM modem signals CTS, DSR, RI, DCD, DTR, RTS.

### 9.1.1 Main features

- Compatible with 16C550 asynchronous serial port and enhanced.
- Supports 5, 6, 7 or 8 data bits and 1 or 2 stop bits. Support odd, even,
- no check, blank 0, mark 1 and other check methods. Programmable
- communication baud rate, up to 5Mbps baud rate.
- Built-in 8-byte FIFO first-in first-out buffer, supports 4 FIFO trigger levels.
- UART0 supports MODEM modem signals CTS, DSR, RI, DCD, DTR, RTS.
- UART0 supports hardware flow control signals CTS and RTS automatic handshake and automatic transmission rate control, compatible with
- TL16C550C. Support serial frame error detection, support Break line interval detection.
- Supports full-duplex and half-duplex serial communication, UART0 provides a sending status pin for switching RS485.

## 9.2 Register description

Table 9-1 List of UART0 related registers

| name | address | description | Reset value |
|---|---|---|---|
| R8_UART0_MCR | 0x40003000 | Modem MODEM control register | 0x00 |
| R8_UART0_IER | 0x40003001 | Interrupt enable register | 0x00 |
| R8_UART0_FCR | 0x40003002 | FIFO control register | 0x00 |
| R8_UART0_LCR | 0x40003003 | Line control register | 0x00 |
| R8_UART0_IIR | 0x40003004 | Interrupt recognition register | 0x01 |
| R8_UART0_LSR | 0x40003005 | Line status register | 0x60 |
| R8_UART0_MSR | 0x40003006 | Modem status register | 0xX0 |
| R8_UART0_RBR | 0x40003008 | Receive buffer register | 0xXX |
| R8_UART0_THR | 0x40003008 | Transmit holding register | 0xXX |
| R8_UART0_RFC | 0x4000300A | Receive FIFO count register | 0x00 |
| R8_UART0_TFC | 0x4000300B | Transmit FIFO count register | 0x00 |
| R16_UART0_DL | 0x4000300C | Baud rate divisor latch | 0xXX |
| R8_UART0_DIV | 0x4000300E | Prescaler Divisor Register | 0xXX |
| R8_UART0_ADR | 0x4000300F | Slave address register | 0xFF |

Table 9-2 UART1 related register list

| name | address | description | Reset value |
|---|---|---|---|
| R8_UART1_MCR | 0x40003400 | Modem MODEM control register | 0x00 |
| R8_UART1_IER | 0x40003401 | Interrupt enable register | 0x00 |
| R8_UART1_FCR | 0x40003402 | FIFO control register | 0x00 |
| R8_UART1_LCR | 0x40003403 | Line control register | 0x00 |
| R8_UART1_IIR | 0x40003404 | Interrupt recognition register | 0x01 |
| R8_UART1_LSR | 0x40003405 | Line status register | 0x60 |
| R8_UART1_RBR | 0x40003408 | Receive buffer register | 0xXX |
| R8_UART1_THR | 0x40003408 | Transmit holding register | 0xXX |

| | | | |
|---|---|---|---|
| R8_UART1_RFC | 0x4000340A | Receive FIFO count register | 0x00 |
| R8_UART1_TFC | 0x4000340B | Transmit FIFO count register | 0x00 |
| R16_UART1_DL | 0x4000340C | Baud rate divisor latch | 0xXX |
| R8_UART1_DIV | 0x4000340E | Prescaler Divisor Register | 0xXX |

Table 9-3 UART2 related register list

| name | address | description | Reset value |
|---|---|---|---|
| R8_UART2_MCR | 0x40003800 | Modem MODEM control register | 0x00 |
| R8_UART2_IER | 0x40003801 | Interrupt enable register | 0x00 |
| R8_UART2_FCR | 0x40003802 | FIFO control register | 0x00 |
| R8_UART2_LCR | 0x40003803 | Line control register | 0x00 |
| R8_UART2_IIR | 0x40003804 | Interrupt recognition register | 0x01 |
| R8_UART2_LSR | 0x40003805 | Line status register | 0x60 |
| R8_UART2_RBR | 0x40003808 | Receive buffer register | 0xXX |
| R8_UART2_THR | 0x40003808 | Transmit holding register | 0xXX |
| R8_UART2_RFC | 0x4000380A | Receive FIFO count register | 0x00 |
| R8_UART2_TFC | 0x4000380B | Transmit FIFO count register | 0x00 |
| R16_UART2_DL | 0x4000380C | Baud rate divisor latch | 0xXX |
| R8_UART2_DIV | 0x4000380E | Prescaler Divisor Register | 0xXX |

Table 9-4 UART3 related register list

| name | address | description | Reset value |
|---|---|---|---|
| R8_UART3_MCR | 0x40003C00 | Modem MODEM control register | 0x00 |
| R8_UART3_IER | 0x40003C01 | Interrupt enable register | 0x00 |
| R8_UART3_FCR | 0x40003C02 | FIFO control register | 0x00 |
| R8_UART3_LCR | 0x40003C03 | Line control register | 0x00 |
| R8_UART3_IIR | 0x40003C04 | Interrupt recognition register | 0x01 |
| R8_UART3_LSR | 0x40003C05 | Line status register | 0x60 |
| R8_UART3_RBR | 0x40003C08 | Receive buffer register | 0xXX |
| R8_UART3_THR | 0x40003C08 | Transmit holding register | 0xXX |
| R8_UART3_RFC | 0x40003C0A | Receive FIFO count register | 0x00 |
| R8_UART3_TFC | 0x40003C0B | Transmit FIFO count register | 0x00 |
| R16_UART3_DL | 0x40003C0C | Baud rate divisor latch | 0xXX |
| R8_UART3_DIV | 0x40003C0E | Prescaler Divisor Register | 0xXX |

Tune Modem MODEM control register (R8_UAR Tx_MCR ) (X=0/1/2/3)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| 7 | RB_MCR_HALF | RW | Half-duplex transceiver mode control (only supported by UART0):<br>1: Enter half-duplex transceiver mode, send priority, no<br>When sending, it is receiving;<br>0: Disable half-duplex mode. | 0 |
| 6 | RB_MCR_TNOW | RW | DTR pin output is sending status enable (only UART0 support):<br>1: Output the sending indication status to DTR<br>Pin, can be used to control the RS485 transceiver switch;<br>0: DTR pin is normal function. | 0 |
| 5 | RB_MCR_AU_FLOW_EN | RW | CTS and RTS hardware automatic flow control enable (only supported by UART0) | 0 |

| | | | | |
|---|---|---|---|---|
| | | | 1: Enable CTS and RTS hardware automatic flow control; 0: Disable CTS and RTS hardware automatic flow control. In flow control mode, if this bit is 1, then only CTS pin input is detected as valid (active low) When the serial port continues to send the next data, otherwise temporarily Stop serial port transmission. When this bit is 1, the CTS input status change will not generate MODEM status interrupt. Such as If this bit is 1 and RTS is 1, then when the receive FIFO is empty, the serial port will automatically validate the RTS pin (active low), Until the number of bytes received reaches the FIFO The serial port will automatically invalidate the RTS pin when the trigger point is And it can be effective again when the receive FIFO is empty. foot. Using hardware automatic flow control, you can transfer your own CTS The pin is connected to the RTS pin of the other party, and the The RTS pin is sent to the CTS pin of the other party. | |
| 4 | RB_MCR_LOOP | RW | Enable the test mode of the internal loop (only for UART0 hold): 1: Enable the test mode of the internal loop; 0: Disable the test mode of the internal loop. In the test mode of the internal loop, all pairs of the serial port The external output pins are all in an invalid state, and TXD returns Back to RXD, RTS returns to CTS internally, DTR internally returns to DSR, OUT1 internally returns to RI, OUT2 Return to DCD internally. | 0 |
| 3 | RB_MCR_OUT2<br>RB_MCR_INT_OE | RW | The interrupt request output control bit of the serial port: 1: Allow request; 0: Prohibit. | 0 |
| 2 | RB_MCR_OUT1 | RW | User-defined MODEM control bit (only supported by UART0), no actual output pin is connected: 1: Set high; 0: Set low. | 0 |
| 1 | RB_MCR_RTS | RW | RTS signal output level control (Only supported by UART0): 1: RTS signal output is valid (low level) ; 0: RTS signal output high level (default) | 0 |
| 0 | RB_MCR_DTR | RW | DTR signal output level control (Only supported by UART0): 1: DTR signal output is valid (low level) ; 0: DTR signal output high level (default) . | 0 |

in Disable enable Save Device (R8_UARTx_IER) (x= 0 /1/2/3 )

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| 7 | RB_IER_RESET | WZ | The serial port software resets the control bit and is automatically cleared: 1: The software resets the serial port; 0: Normal operation. | 0 |
| 6 | RB_IER_TXD_EN | RW | Serial port TXD pin output enable bit: 1: Enable pin output; 0: Disable pin output. | 0 |
| 5 | RB_IER_RTS_EN | RW | RTS pin output enable bit (only supported by UART0): 1: Enable pin output; 0: Disable pin output. | 0 |
| 4 | RB_IER_DTR_EN | RW | DTR pin output enable bit (only supported by UART0) : 1: Enable pin output; 0: Disable pin output. | 0 |
| 3 | RB_IER_MODEM_CHG | RW | Modem input status change interrupt enable bit (Only supported by UART0): 1: Enable interrupt; 0: Disable interrupts. | 0 |

| | | | | |
|---|---|---|---|---|
| 2 | RB_IER_LINE_STAT | RW | Receive line status interrupt enable bit:<br>1: Enable interrupt; 0: Disable interrupts. | 0 |
| 1 | RB_IER_THR_EMPTY | RW | Send holding register empty interrupt enable bit:<br>1: Enable interrupt; 0: Disable interrupts. | 0 |
| 0 | RB_IER_RECV_RDY | RW | Receive data interrupt enable bit:<br>1: Enable interrupt; 0: Disable interrupts. | 0 |

FIFO control register Device (R8_UARTx_FCR) (x=0/1/2/ 3)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [7:6] | RB_FCR_FIFO_TRIG | RW | Receive FIFO interrupt and hardware flow control trigger Point selection:<br>00: 1 byte; 01: 2 bytes;<br>10: 4 bytes; 11: 7 bytes.<br>Used to set the interrupt and hardware flow control of the receive FIFO The trigger point of the system, for example: 10 corresponds to 4 bytes, that is, when 4 bytes are received, the available data will be generated. Interrupt, and automatically invalid when hardware flow control is enabled RTS pin. | 00b |
| [5:3] | Reserved | RO | Reserved. | 000b |
| 2 | RB_FCR_TX_FIFO_CLR | WZ | Send FIFO data clear enable bit, automatically cleared:<br>1: Clear the data in the transmit FIFO (excluding TSR); 0: Do not clear the data in the transmit FIFO. | 0 |
| 1 | RB_FCR_RX_FIFO_CLR | WZ | Receiving FIFO data clear enable bit, automatically cleared:<br>1: Clear the data in the receive FIFO (excluding RSR); 0: Do not clear the data in the receive FIFO. | 0 |
| 0 | RB_FCR_FIFO_EN | RW | FIFO enable bit:<br>1: Enable 8-byte FIFO; 0: Disable FIFO. After FIFO is disabled, it is 16C450 compatible mode, which is equivalent to only one byte of FIFO (RECV_TG1=0, RECV_TG0=0, FIFO_EN=1), it is recommended to enable. | 0 |

line Route control Save Device (R8_UARTx_LCR) (x= 0 /1/2/3 )

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| 7 | RB_LCR_DLAB<br>RB_LCR_GP_BIT | RW | Serial port common bits, user-defined. | 0 |
| 6 | RB_LCR_BREAK_EN | RW | Force generation of BREAK line interval enable bit:<br>1: Forced generation; 0: Not generated. | 0 |
| [5:4] | RB_LCR_PAR_MOD | RW | Parity bit format selection:<br>00: odd parity; 01: Even parity;<br>10: Mark bit (MARK, set to 1);<br>11: Space bit (SPACE, cleared to 0).<br>Only valid when the RB_LCR_PAR_EN bit is 1. | 00b |
| 3 | RB_LCR_PAR_EN | RW | Parity bit enable bit:<br>1: Allow parity to be generated when sending and parity checking when receiving Check position<br>0: No parity bit. | 0 |
| 2 | RB_LCR_STOP_BIT | RW | Stop bit format setting bit:<br>0: One stop bit; 1: Two stop bits. | 0 |

| | | | | |
|---|---|---|---|---|
| [1:0] | RB_LCR_WORD_SZ | RW | Serial port data length selection:<br>00: 5 data bits; 01: 6 data bits; 10: 7 data bits; 11: 8 data bits. | 00b |

in Break identification Save Device (R8_UARTx_IIR) (x= 0 /1/2/3 )

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [7:6] | RB_IIR_FIFO_ID | RO | Serial port FIFO enable status bit:<br>11: FIFO is enabled; 00: FIFO is not enabled. | 00b |
| [5:4] | Reserved | R0 | Reserved. | 00b |
| [3:0] | RB_IIR_INT_MASK | RO | Interrupt flag: If the RB_IIR_NO_INT bit is 0, it means that an interrupt is generated, and it needs to be read and judged Source off. Please refer to Table 9-5 for details. | 0000b |
| 0 | RB_IIR_NO_INT | RO | Serial port no interruption flag:<br>1: No interruption; 0: There is an interrupt. | 1 |

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

Table 9-5 The meaning of RB_IIR_INT_MASK in the IIR register

| IIR mail Register Bit | | | | priority level | Type of interrupt | Interrupt source | Clear interrupt method |
|---|---|---|---|---|---|---|---|
| IID3 | IID2 | IID1 | NOINT | | | | |
| 0 | 0 | 0 | 1 | no | No interrupt is generated | No interruption | |
| 1 | 1 | 1 | 0 | 0 | bus address match | The 1 data received is the serial bus address, and the read IIR address matches the preset slave value or the broadcast address. Or disable more (Only supported by UART0) | Machine mode |
| 0 | 1 | 1 | 0 | 1 | Receive line status | OVER_ERR, PAR_ERR, FRAM_ERR, BREAK_ERR Read | LSR |
| 0 | 1 | 0 | 0 | 2 | Receive data available | The number of bytes received reaches the trigger point of the FIFO. | Read RBR |
| 1 | 1 | 0 | 0 | 2 | Receive data timeout | The next data has not been received for more than 4 data times. | Read RBR |
| 0 | 0 | 1 | 0 | 3 | THR register empty | Send holding register empty, or read IIR RB_IER_THR_EMPTY from 0 to 1 to trigger. | Or write THR |
| 0 | 0 | 0 | 0 | 4 | MODEM input changes | △CTS, △DSR, △RI, △DCD are set to 1 to trigger. | Read MSR |

line Route status Save (R8_UARTx_LSR) (x= 0 /1/2/3 )

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| 7 | RB_LSR_ERR_RX_FIFO | RO | Receive FIFO error flag:<br>1: There is at least one PAR_ERR, FRAM_ERR or BREAK_ERR error in the receive FIFO;<br>0: There is no error in the receive FIFO. | 0 |
| 6 | RB_LSR_TX_ALL_EMP | RO | Transmit holding register THR and transmit shift register TSR empty flag:<br>1: Both are empty; 0: Both are not completely empty. | 1 |
| 5 | RB_LSR_TX_FIFO_EMP | RO | Send FIFO empty flag:<br>1: Transmit FIFO is empty; 0: Transmit FIFO is not | 1 |
| 4 | RB_LSR_BREAK_ERR | RZ | empty. BREAK Line gap detection flag:<br>1: BREAK detected; 0: BREAK not detected. | 0 |
| 3 | RB_LSR_FRAME_ERR | RZ | Data frame error flag:<br>1: Indicates the data being read from the receive FIFO There is a framing error, and a valid stop bit is missing;<br>0: There is no error in the currently read data frame. | 0 |
| 2 | RB_LSR_PAR_ERR | RZ | Receive data parity error flag: | 0 |

| | | | | |
|---|---|---|---|---|
| | | | 1: Indicates the data being read from the receive FIFO There is a parity error;<br>0: The parity of the currently read data is correct. | |
| 1 | RB_LSR_OVER_ERR | RZ | Receive FIFO buffer overflow flag:<br>1: Overflow; 0; no overflow. | 0 |
| 0 | RB_LSR_DATA_RDY | RO | There are received data flag bits in the receive FIFO:<br>1: There is data in FIFO; 0: No data.<br>After reading all the data in the FIFO, This bit is automatically cleared to 0. | 0 |

Tune Modem MODEM status register (R8_UAR T0_MSR ) (Only supported by UART0)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| 7 | RB_MSR_DCD | RO | DCD pin status bit:<br>1: DCD pin is valid (low level);<br>0: DCD pin is invalid (high level). | X |
| 6 | RB_MSR_RI | RO | RI pin status bit:<br>1: RI pin is active (low level) ;<br>0: RI pin is invalid (high level) . | X |
| 5 | RB_MSR_DSR | RO | DSR pin status bit:<br>1: DSR pin is valid (low level);<br>0: DSR pin is invalid (high level). | X |
| 4 | RB_MSR_CTS | RO | CTS pin status bit:<br>1: CTS pin is valid (low level);<br>0: CTS pin is invalid (high level). | X |
| 3 | RB_MSR_DCD_CHG | RZ | DCD pin input status change flag:<br>1: There has been a change; 0: No change. | 0 |
| 2 | RB_MSR_RI_CHG | RZ | RI pin input status change flag:<br>1: There has been a change; 0: No change. | 0 |
| 1 | RB_MSR_DSR_CHG | RZ | DSR pin input status change flag:<br>1: There has been a change; 0: No change. | 0 |
| 0 | RB_MSR_CTS_CHG | RZ | CTS pin input status change flag:<br>1: There has been a change; 0: No change. | 0 |

Pick up Receive buffer Save Device (R8_UARTx_RBR) (x= 0 /1/2/3 )

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_UARTx_RBR | RO | Data receiving buffer register.<br>If the DATA_RDY bit of the LSR is 1, the received data can be read from this register;<br>If FIFO_EN is 1, the data received from the serial port shift register RSR is first stored in the receive FIFO, and then read through this register. | XXh |

hair Send and keep sent Save (R8_UARTx_THR) (x= 0 /1/2/3 )

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_UARTx_THR | WO | Send holding register.<br>Including transmit FIFO, used to write data ready to be sent Data; if FIFO_EN is 1, the written data is first stored in the transmit FIFO, and then through the transmit Send the shift register TSR to output one by one. | XXh |

Receive FIFO meter Number register (R8_UARTx_RFC ) (X=0/ 1/2/3)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_UARTx_RFC | RO | The data count in the current receive FIFO. | 00h |

Transmit FIFO meter Number register (R8_UARTx_TFC ) (X=0/ 1/2/3)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_UARTx_TFC | RO | The data count in the current transmit FIFO. | 00h |

wave Special rate divisor lock Register (R16_UARTx_DL) ( x =0/1/2 /3)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [15:0] | R16_UARTx_DL | RW | The 16-bit divisor is used to calculate the baud rate.<br>Formula: Divisor = internal reference clock of serial port Fuart / 16 / required communication baud rate.<br>Example: If the internal reference clock Fuart of the serial port is 1.8432MHz, the required baud rate is 9600bps, then Divisor=1843200/16/9600=12. | XXXXh |

Foresee Divider send Register (R8_UARTx_DIV) ( x =0/1/2 /3)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_UARTx_DIV | RW | Used to calculate the internal reference clock of the serial port, the lower 7 bits effective.<br>Formula: Divisor = Fsys * 2 / The internal reference clock of the serial port, the maximum value is 127. | XXh |

From Machine address Save (R8_UART0_ADR) (only UART0 stand by)

| Bit | name | access | description | Reset value |
|---|---|---|---|---|
| [7:0] | R8_UART0_ADR | RW | Serial port 0 slave address in multi-machine communication:<br>FFh: not used; Other: slave address. | 0FFh |

R8_UART0_ADR presets the address when this machine is used as a slave, which is used to automatically compare the received address during multi-machine communication and pl
When the address matches or when the broadcast address 0FFH is received, an interrupt is generated, and subsequent data packets are allowed to be received at the same time. Not until
Receive any data, stop receiving any data after starting to send data or rewriting R8_UART0_ADR register, until the next address
Reception is allowed when it matches again or the broadcast address is received.

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

, RB_LCR_PAR_MOD should be set to 11b.

## 9.3 Function description and configuration

UART0/1/2/3 output pins are all 3.3V LVCMOS level. The pins in asynchronous serial port mode include: data transmission pins and MODEM communication signal pins (only UART0 supports). The data transmission pins include: TXD pin and RXD pin, both of which are high by default; MODEM The contact signal pins include: CTS pin, DSR pin, RI pin, DCD pin, DTR pin, RTS pin, all of which are high by default. All these MODEM contact signals can be used as general-purpose I/O pins, which are controlled by the application program and their purpose is defined.

Each of the 4 groups of UARTs has independent transceiver buffers and 8-byte FIFOs, and supports simplex, half-duplex, or full-duplex asynchronous serial communication. Serial data includes 1 low-level start bit, 5, 6, 7 or 8 data bits, 0 or 1 additional check bit or flag bit,