

概述

CH583 是集成 BLE 无线通讯的 RISC-V MCU 微控制器。片上集成 2Mbps 低功耗蓝牙 BLE 通讯模块、2 个全速 USB 主机和设备控制器及收发器、2 个 SPI、4 个串口、14 路 ADC、触摸按键检测模块、RTC 等丰富的外设资源。

功能

● Royalty-Free 内核 Core:

- 32 位青稞 RISC-V 处理器 WCH RISC-V4A
- 支持 RV32IMAC 指令集，硬件乘法和除法
- 低功耗两级流水线，高性能 1.56DMIPS
- 多档系统主频，最低 32KHz
- 特有高速的中断响应机制

● 512K 字节非易失存储 FlashROM:

- 448KB 用户应用程序存储区 CodeFlash
- 32KB 用户非易失数据存储区 DataFlash
- 24KB 系统引导程序存储区 BootLoader
- 8KB 系统非易失配置信息存储区 InfoFlash
- 支持 ICP、ISP 和 IAP，支持 OTA 无线升级
- 20MHz 系统主频下基本零等待

● 32K 字节易失数据存储 SRAM:

- 30KB 双电源供电的睡眠保持存储区 RAM30K
- 2KB 双电源供电的睡眠保持存储区 RAM2K

● 电源管理和低功耗:

- 支持 3.3V 和 2.5V 电源，CH583M 支持 1.8V
- 内置 DC-DC 转换，可降低功耗
- 空闲模式 Idle: 1.6mA
- 暂停模式 Halt: 320uA
- 睡眠模式 Sleep: 0.7uA~2.8uA 多档
- 下电模式 Shutdown: 0.2uA~2.3uA 多档
- 可选低功耗或高精度的电池电压低压监控

● 安全特性: AES-128 加解密，芯片唯一 ID

● 低功耗蓝牙 BLE:

- 集成 2.4GHz RF 收发器和基带及链路控制
- 接收灵敏度-98dBm，可编程+6dBm 发送功率
- BLE 符合 Bluetooth Low Energy 5.0 规范
- 支持 2Mbps、1Mbps、500Kbps 和 125Kbps
- 在 6dBm 发送功率时无线通讯距离约 500 米
- 在 6dBm 发送功率 125Kbps 通讯距离约 1000 米
- 提供优化的协议栈和应用层 API，支持组网

● 实时时钟 RTC: 支持定时和触发两种模式

● 通用串行总线 USB:

- 2 组独立的 USB 控制器和收发器
- 15 个端点，支持 DMA，支持 64 字节数据包
- 集成 USB 2.0 全速收发器 PHY，无需外围器件
- 支持全/低速的 Host 主机和 Device 设备模式

● 模数转换 ADC:

- 12 位模数转换器，支持差分 and 单端输入
- 14 路外部模拟信号通道和 2 路内部信号

● 触摸按键检测模块 TouchKey: 14 路通道

● 定时器 Timer 和脉宽调制 PWM:

- 4 组 26 位定时器，16MHz 主频定时可达 4.2S
- 4 路捕捉/采样，支持上升沿/下降沿/双边沿
- 4 路 26 位 PWM 输出，8 路 8 位 PWM 输出

● 异步串口 UART:

- 4 组独立 UART，兼容 16C550，内置 8 级 FIFO
- 23 位计数器，通讯波特率可达 6Mbps
- UART0 支持部分 Modem，支持硬件自动流控
- UART0 支持多机通讯时从机地址自动匹配

● 串行外设接口 SPI:

- 2 组独立 SPI，内置 FIFO
- SCK 串行时钟频率可达系统主频的一半
- SPI0 支持 Master 和 Slave 模式，支持 DMA

● 两线串行接口 I2C:

- 支持 Master 和 Slave 模式，兼容 SMBus
- 支持 7 位或 10 位地址和总线广播
- 支持仲裁、错误检测、PEC 校验、延长时钟

● 温度传感器 TS

● 时钟: 内置 PLL，内置 32KHz 时钟

● 通用输入输出端口 GPIO:

- 40 个 GPIO，其中 2 个支持 5V 信号输入
- 可选上拉或下拉电阻，可选输出驱动能力
- 32 个 GPIO 支持电平或边沿中断输入
- 32 个 GPIO 支持电平或边沿唤醒输入

● 封装形式: QFN48_5X5、QFN28_4X4

CH583 相比 CH582 多了 SPI1 主机，支持最低 1.7V 电源电压。CH581 基于 CH582 简化，去掉了一组 USB、触摸按键检测模块、I2C 模块和两个串口，FlashROM 总容量仅 256KB，ADC 不支持 DMA。除了

上述差异，其它基本与 CH582 相同，可直接参考 CH582 手册和资料。

| 芯片型号 | 用户闪存+引导闪存+数据闪存 | 数据存储RAM | RTC时钟 | 定时器 | 捕捉 | PWM | 串口 | SPI | I2C主从 | BLE蓝牙 | ADC及TS | 电容触摸按键 | USB主机 | USB设备 | DC-DC | 最低电源电压 | 通用I/O | 封装形式 |
|--------|----------------|---------|-------|-----|----|-----|----|-----|-------|-------|--------|--------|-------|-------|-------|--------|-------|-------|
| CH583M | 448+24+32K | 30+2K | √ | 4组 | 4 | 4+8 | 4组 | 主从 | √ | √ | 14+1 | 14路 | 2 | 2 | √ | 1.7V | 40 | QFN48 |
| CH582M | 448+24+32K | 30+2K | | | 4 | 4+8 | | | √ | | 14+1 | 14路 | 2 | 2 | √ | 2.3V | 40 | QFN48 |
| CH582F | 448+24+32K | 30+2K | | | 4 | 4+6 | | | √ | | 8+1 | 8路 | 2 | 2 | √ | 2.3V | 20 | QFN28 |
| CH581F | 192+24+32K | 30+2K | | | 4 | 4+6 | | | × | | 6+1 | × | × | √ | √ | 2.3V | 20 | QFN28 |

第 1 章 引脚信息

1.1 引脚排列

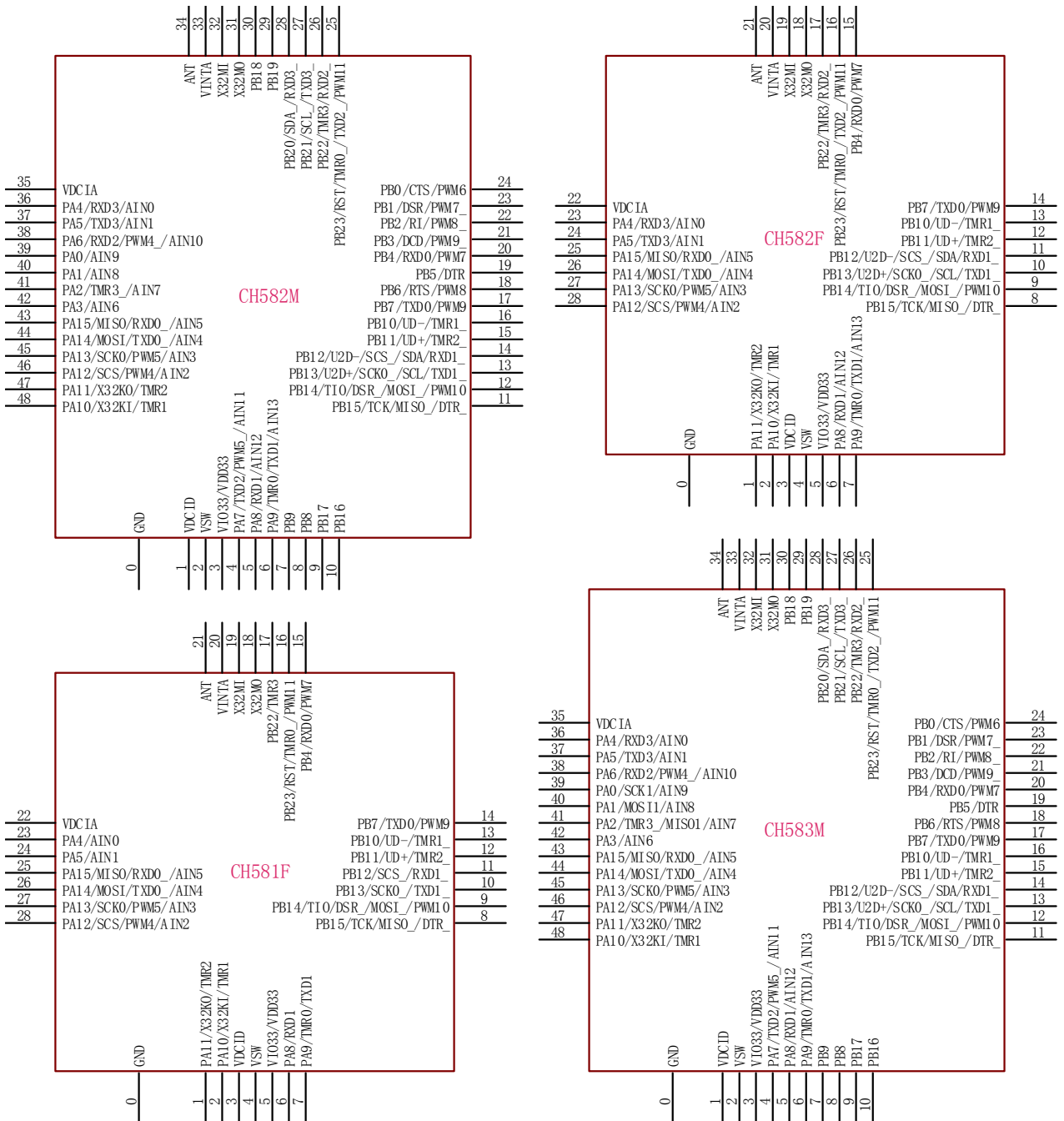


图 1-1 CH582M/CH583M(QFN48) 和 CH582F/CH581F(QFN28) 封装引脚排列

1.2 引脚描述

| QFN48 引脚号 | QFN28 引脚号 | 引脚 名称 | 引脚 类型 | 复用功能 在前优先 | 功能描述 |
|--------------|--------------|----------|-------------|----------------------------------|---|
| 0 | 0 | GND | 电源 | - | 芯片底板，公共接地端，电压 0V 参考点。 |
| 1 | 3 | VDC1D | 电源 | - | 内部数字电路 LDO 调整器的电源输入，需外接退耦电容。启用 DC-DC 时建议 2.2uF（支持 0.47uF~4.7uF，容值小略省电但降低 BLE 灵敏度 2dBm），不启用时建议不小于 0.1uF。 |
| 2 | 4 | VSW | 电源 | - | 内部 DC-DC 开关输出，启用 DC-DC 时必须贴近引脚串接电感连接到 VDC1D，建议用 10uH 电感（支持 3.3uH~33uH），不启用 DC-DC 时可以直连 VDC1D。 |
| 3 | 5 | VDD33 | 电源 | VBAT | DC-DC 或电池电源输入，需贴近引脚外接退耦电容。启用 DC-DC 时建议 2.2uF 或 1uF，不启用 DC-DC 时 0.1uF 即可。 |
| | | VI033 | 电源 | - | I/O 电源输入，需贴近引脚外接退耦电容，可与 VDD33 共用同一电容。 |
| 4 | 无 | PA7 | I/O/A | TXD2 /PWM5_ /AIN11 | PA7：通用双向数字 I/O 引脚。 TXD2：UART2 串行数据输出。 PWM5_：脉宽调制输出通道 5 引脚映射。 AIN11：ADC 模拟信号输入通道 11。 |
| 5 | 6 | PA8 | I/O/A | RXD1 /AIN12 | PA8：通用双向数字 I/O 引脚。 RXD1：UART1 串行数据输入。 AIN12：ADC 模拟信号输入通道 12。 |
| 6 | 7 | PA9 | I/O/A | TMRO /TXD1 /AIN13 | PA9：通用双向数字 I/O 引脚。 TMRO：定时器 0 的捕捉输入 0 和 PWM 输出通道 0。 TXD1：UART1 串行数据输出。 AIN13：ADC 模拟信号输入通道 13。 |
| 7 | 无 | PB9 | I/O | - | PB9：通用双向数字 I/O 引脚。 |
| 8 | 无 | PB8 | I/O | - | PB8：通用双向数字 I/O 引脚。 |
| 9 | 无 | PB17 | I/O | - | PB17：通用双向数字 I/O 引脚。 |
| 10 | 无 | PB16 | I/O | - | PB16：通用双向数字 I/O 引脚。 |
| 11 | 8 | PB15 | I/O /5VT | TCK /MISO_ /DTR_ | PB15：通用双向数字 I/O 引脚。 TCK：两线仿真调试接口的串行时钟输入。注 3 MISO_：SPI0 的 MISO 引脚映射。 DTR_：UART0 的 DTR 引脚映射。 |
| 12 | 9 | PB14 | I/O /5VT | TIO /MOSI_ /PWM10 /DSR_ | PB14：通用双向数字 I/O 引脚。 TIO：仿真调试接口的串行数据输入输出，内置上拉。注 3 MOSI_：SPI0 的 MOSI 引脚映射。 PWM10：脉宽调制输出通道 10。 DSR_：UART0 的 DSR 引脚映射。 |
| 13 | 10 | PB13 | I/O | U2D+ /SCK0_ /SCL /TXD1_ | PB13：通用双向数字 I/O 引脚。 U2D+：USB2 总线的 D+数据线。 SCK0_：SPI0 的 SCK 引脚映射。 SCL：I2C 串行时钟引脚，主机输出和输入/从机输入。 TXD1_：UART1 的 TXD1 引脚映射。 |
| 14 | 11 | PB12 | I/O | U2D- /SCS_ /SDA /RXD1_ | PB12：通用双向数字 I/O 引脚。 U2D-：USB2 总线的 D-数据线。 SCS_：SPI0 的 SCS 引脚映射。 SDA：I2C 串行数据引脚，开漏输出和输入。 RXD1_：UART1 的 RXD1 引脚映射。 |

| | | | | | |
|----|----|-------|-------|------------------------------------|---|
| 15 | 12 | PB11 | I/O/A | UD+ /TMR2_ | PB11: 通用双向数字 I/O 引脚。 UD+: USB 总线的 D+数据线。 TMR2_: 定时器 2 的 TMR2 引脚映射。 |
| 16 | 13 | PB10 | I/O/A | UD- /TMR1_ | PB10: 通用双向数字 I/O 引脚。 UD-: USB 总线的 D-数据线。 TMR1_: 定时器 1 的 TMR1 引脚映射。 |
| 17 | 14 | PB7 | I/O | TXD0 /PWM9 | PB7: 通用双向数字 I/O 引脚。 TXD0: UART0 串行数据输出。 PWM9: 脉宽调制输出通道 9。 |
| 18 | 无 | PB6 | I/O | RTS /PWM8 | PB6: 通用双向数字 I/O 引脚。 RTS: UART0 的 MODEM 输出信号, 请求发送。 PWM8: 脉宽调制输出通道 8。 |
| 19 | 无 | PB5 | I/O | DTR | PB5: 通用双向数字 I/O 引脚。 DTR: UART0 的 MODEM 输出信号, 数据终端就绪。 |
| 20 | 15 | PB4 | I/O | RXD0 /PWM7 | PB4: 通用双向数字 I/O 引脚。 RXD0: UART0 串行数据输入。 PWM7: 脉宽调制输出通道 7。 |
| 21 | 无 | PB3 | I/O | DCD /PWM9_ | PB3: 通用双向数字 I/O 引脚。 DCD: UART0 的 MODEM 输入信号, 载波检测。 PWM9_: 脉宽调制输出通道 9 引脚映射。 |
| 22 | 无 | PB2 | I/O | R1 /PWM8_ | PB2: 通用双向数字 I/O 引脚。 R1: UART0 的 MODEM 输入信号, 振铃指示。 PWM8_: 脉宽调制输出通道引脚映射。 |
| 23 | 无 | PB1 | I/O | DSR /PWM7_ | PB1: 通用双向数字 I/O 引脚。 DSR: UART0 的 MODEM 输入信号, 数据装置就绪。 PWM7_: 脉宽调制输出通道 7 引脚映射。 |
| 24 | 无 | PB0 | I/O | CTS /PWM6 | PB0: 通用双向数字 I/O 引脚。 CTS: UART0 的 MODEM 输入信号, 清除发送。 PWM6: 脉宽调制输出通道 6。 |
| 25 | 16 | PB23 | I/O | RST# /TMRO_ /TXD2_ /PWM11 | PB23: 通用双向数字 I/O 引脚。 RST#: 外部复位输入, 低电平有效, 内置上拉电阻。 TMRO_: 定时器 0 的 TMRO 引脚映射。 TXD2_: UART2 的 TXD2 引脚映射。 PWM11: 脉宽调制输出通道 11。 |
| 26 | 17 | PB22 | I/O | TMR3 /RXD2_ | PB22: 通用双向数字 I/O 引脚。 TMR3: 定时器 3 的捕捉输入 3 和 PWM 输出通道 3。 RXD2_: UART2 的 RXD2 引脚映射。 |
| 27 | 无 | PB21 | I/O | SCL_ /TXD3_ | PB21: 通用双向数字 I/O 引脚。 SCL_: I2C 串行时钟引脚映射。 TXD3_: UART3 的 TXD3 引脚映射。 |
| 28 | 无 | PB20 | I/O | SDA_ /RXD3_ | PB20: 通用双向数字 I/O 引脚。 SDA_: I2C 串行数据引脚映射。 RXD3_: UART3 的 RXD3 引脚映射。 |
| 29 | 无 | PB19 | I/O | - | PB19: 通用双向数字 I/O 引脚。 |
| 30 | 无 | PB18 | I/O | - | PB18: 通用双向数字 I/O 引脚。 |
| 31 | 18 | X32M0 | I/A | - | 高频振荡器 HSE 的反相输出端, 外接 32MHz 晶体的一端。 |
| 32 | 19 | X32M1 | A | - | 高频振荡器 HSE 的输入端, 外接 32MHz 晶体的另一端。 |
| 33 | 20 | VINTA | 电源 | - | 内部模拟电路的电源节点, 需贴近引脚外接退耦电容。不启用 DC-DC 时建议 0.47uF 或更大; 启用 DC-DC 时可用 1uF, |

| | | | | | |
|----|----|-------|-------|--------------------------|--|
| | | | | | 支持 0.47 μ F~2.2 μ F，容值大则信号质量好但功耗略大。 |
| 34 | 21 | ANT | A | - | RF 射频信号输入输出，建议直连天线。 |
| 35 | 22 | VDC1A | 电源 | - | 内部模拟电路 LDO 调整器的电源输入，需外接退耦电容。建议不小于 0.1 μ F，建议直连 VDC1D。 |
| 36 | 23 | PA4 | I/O/A | RXD3 /AIN0 | PA4: 通用双向数字 I/O 引脚。 RXD3: UART3 串行数据输入。 AIN0: ADC 模拟信号输入通道 0。 |
| 37 | 24 | PA5 | I/O/A | TXD3 /AIN1 | PA5: 通用双向数字 I/O 引脚。 TXD3: UART3 串行数据输出。 AIN1: ADC 模拟信号输入通道 1。 |
| 38 | 无 | PA6 | I/O/A | RXD2 /PWM4_ /AIN10 | PA6: 通用双向数字 I/O 引脚。 RXD2: UART2 串行数据输入。 PWM4_: 脉宽调制输出通道 4 引脚映射。 AIN10: ADC 模拟信号输入通道 10。 |
| 39 | 无 | PA0 | I/O/A | SCK1 /AIN9 | PA0: 通用双向数字 I/O 引脚。 SCK1: SPI1 串行时钟引脚，主机输出。 AIN9: ADC 模拟信号输入通道 9。 |
| 40 | 无 | PA1 | I/O/A | MOSI1 /AIN8 | PA0: 通用双向数字 I/O 引脚。 MOSI1: SPI1 串行数据引脚，主机输出。 AIN8: ADC 模拟信号输入通道 8。 |
| 41 | 无 | PA2 | I/O/A | TMR3_ /MISO1 /AIN7 | PA0: 通用双向数字 I/O 引脚。 TMR3_: 定时器 3 的 TMR3 引脚映射。 MISO1: SPI1 串行数据引脚，主机输入。 AIN7: ADC 模拟信号输入通道 7。 |
| 42 | 无 | PA3 | I/O/A | AIN6 | PA3: 通用双向数字 I/O 引脚。 AIN6: ADC 模拟信号输入通道 6。 |
| 43 | 25 | PA15 | I/O/A | MISO /RXD0_ /AIN5 | PA15: 通用双向数字 I/O 引脚。 MISO: SPI0 串行数据引脚，主机输入/从机输出。 RXD0_: UART0 的 RXD0 引脚映射。 AIN5: ADC 模拟信号输入通道 5。 |
| 44 | 26 | PA14 | I/O/A | MOSI /TXD0_ /AIN4 | PA14: 通用双向数字 I/O 引脚。 MOSI: SPI0 串行数据引脚，主机输出/从机输入。 TXD0_: UART0 的 TXD0 引脚映射。 AIN4: ADC 模拟信号输入通道 4。 |
| 45 | 27 | PA13 | I/O/A | SCK0 /PWM5 /AIN3 | PA13: 通用双向数字 I/O 引脚。 SCK0: SPI0 串行时钟引脚，主机输出/从机输入。 PWM5: 脉宽调制输出通道 5。 AIN3: ADC 模拟信号输入通道 3。 |
| 46 | 28 | PA12 | I/O/A | SCS /PWM4 /AIN2 | PA12: 通用双向数字 I/O 引脚。 SCS: SPI0 从机模式下的片选输入，低电平有效。 PWM4: 脉宽调制输出通道 4。 AIN2: ADC 模拟信号输入通道 2。 |
| 47 | 1 | PA11 | I/O/A | X32K0 /TMR2 | PA11: 通用双向数字 I/O 引脚。 X32K0: 低频振荡器的反相输出端，外接 32KHz 晶体的一端。 TMR2: 定时器 2 的捕捉输入 2 和 PWM 输出通道 2。 |
| 48 | 2 | PA10 | I/O/A | X32K1 /TMR1 | PA10: 通用双向数字 I/O 引脚。 X32K1: 低频振荡器的输入端，外接 32KHz 晶体的另一端。 TMR1: 定时器 1 的捕捉输入 1 和 PWM 输出通道 1。 |

注:

(1)、引脚类型:

I=TTL/CMOS 电平斯密特输入;

O=CMOS 电平三态输出;

A=模拟信号输入或输出;

5VT=支持 5V 信号电压输入。

(2)、引脚的复用功能及映射根据其优先级在表中按从高到底排列，其中 GPIO 功能为最低优先级。

(3)、两线仿真调试接口通过 ISP 工具配置。仿真调试接口启用后，PB15 和 PB14 仅用作 TCK 和 TIO，不再用于 GPIO 或外设复用功能引脚。关闭仿真调试接口后，PB15 和 PB14 才可用于 GPIO 和外设复用功能引脚。

(4)、其它封装的引脚参考上述同名引脚的说明。

第 2 章 系统结构及存储器

2.1 系统结构

下图为 CH583 芯片系统结构框图。内核是青稞 RISC-V 微处理器，详细说明参考青稞内核手册。

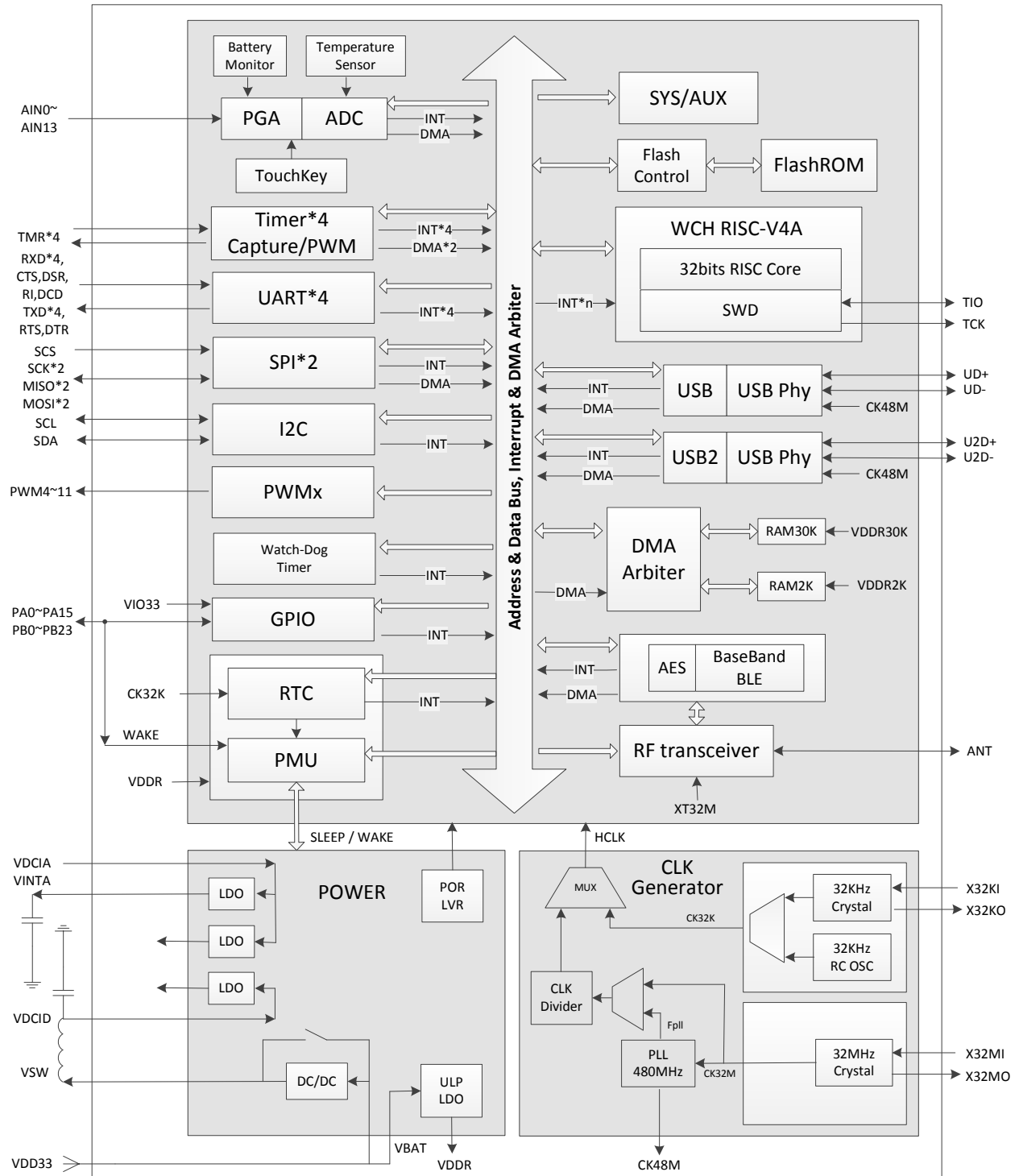


图 2-1 CH583 内部结构框图

2.2 存储器映射

CH583 的寻址空间主要包括 CODE 区/FlashROM、DATA 区/SRAM、外设等几个不同区域，详见下图所示。

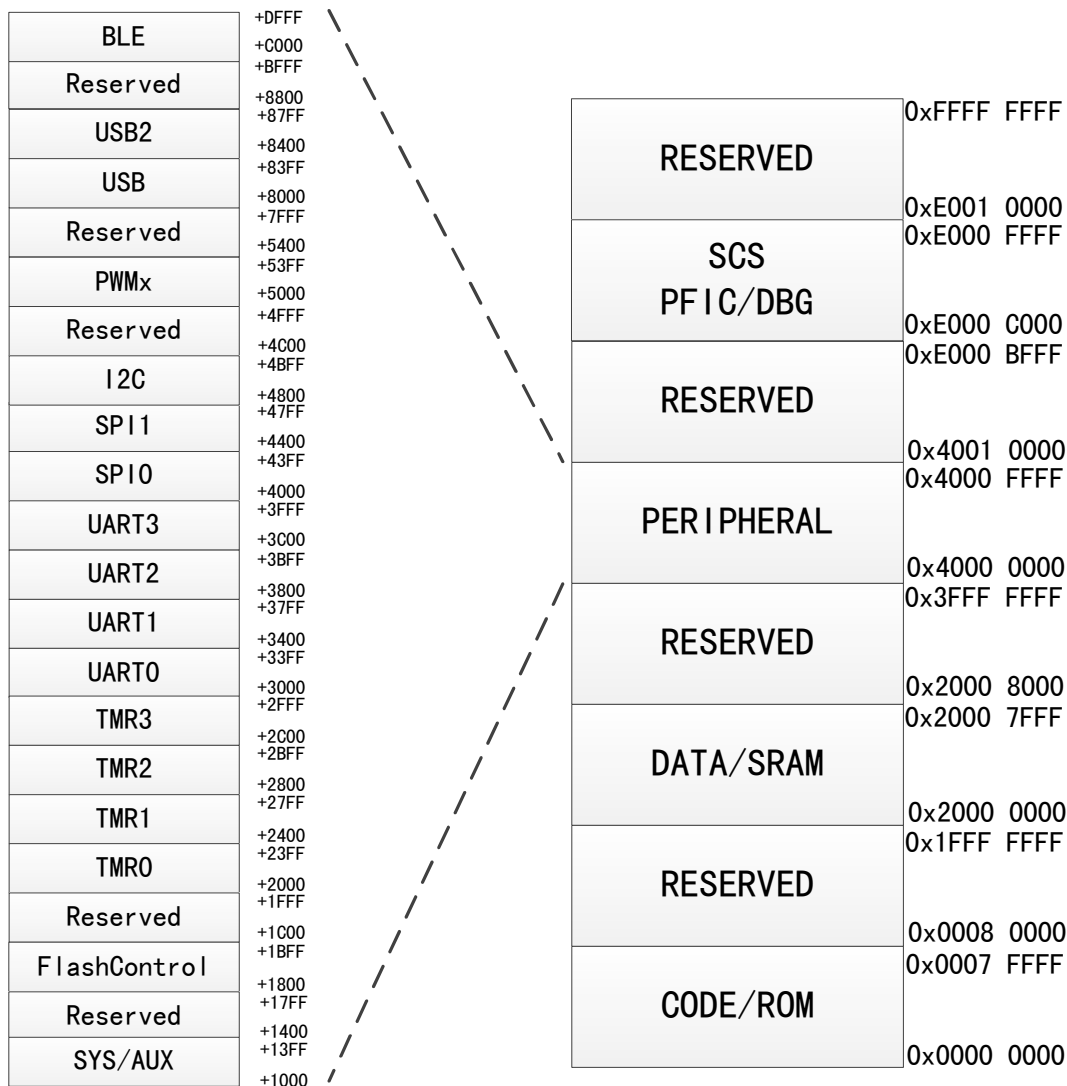


图 2-2 存储器映射图

2.3 存储器映射表

各存储器映射区域地址范围如下表所示：

表 2-1 存储器映射区域地址

| 地址范围 | 用途 | 描述 |
|-----------------------|-------------------|-----------------|
| 0x00000000-0x0007FFFF | 片上 CODE 区域，非易失存储器 | 512KB, FlashROM |
| 0x00080000-0x1FFFFFFF | 保留 | - |
| 0x20000000-0x20007FFF | 片上 DATA 区域，易失存储器 | 32KB, SRAM |
| 0x20008000-0x3FFFFFFF | 保留 | - |
| 0x40000000-0x4000FFFF | 各种外设 | 多个外设模块 |
| 0x40010000-0xE000BFFF | 保留 | - |
| 0xE000C000-0xE000FFFF | 系统内部各种外设 | 系统控制空间 SCS |
| 0xE0010000-0xFFFFFFFF | 保留 | - |

2.3.1 片上 CODE 区域映射表

表 2-2 CODE 区域地址

| 地址范围 | 用途 | 描述 |
|------|----|----|
|------|----|----|

| | | |
|-----------------------|----------------------------------|-------|
| 0x00000000-0x0006FFFF | CH583/CH582: 用户应用程序存储区 CodeFlash | 448KB |
| 0x00000000-0x0002FFFF | CH581: 用户应用程序存储区 CodeFlash | 192KB |
| 0x00070000-0x00077FFF | 用户非易失数据存储区 DataFlash | 32KB |
| 0x00078000-0x0007DFFF | 系统引导程序存储区 BootLoader | 24KB |
| 0x0007E000-0x0007FFFF | 系统非易失配置信息存储区 InfoFlash | 8KB |

地址 0x0007E000-0x0007EEFF 的配置信息可以由用户通过工具设置。

表 2-3 用户级非易失配置信息说明

| 位地址 | 名称 | 用途 | 默认值 |
|-----------|--------------|--|--------|
| 位 2~位 0 | RESERVED | 保留 | 101b |
| 位 3 | CFG_RESET_EN | RST#外部手工复位输入引脚使能 | 0 |
| 位 4 | CFG_DEBUG_EN | 两线仿真调试接口 SWD 使能 | 1 |
| 位 5 | RESERVED | 保留 | 0 |
| 位 6 | CFG_BOOT_EN | 系统引导程序 BootLoader 使能 | 1 |
| 位 7 | CFG_ROM_READ | FlashROM 中的代码和数据保护模式: 0-禁止编程器读出, 程序保密; 1-允许读出 | 1 |
| 位 27~位 8 | RESERVED | 保留 | FFF0Fh |
| 位 31~位 28 | VALID_SIG | 配置信息有效标志, 固定值 | 0100b |

注: 当 CFG_DEBUG_EN=1、CFG_RESET_EN=0、CFG_ROM_READ=1, 两线仿真调试接口启用。

2.3.2 片上 DATA 区域映射表

表 2-4 DATA 区域地址

| 地址范围 | 用途 | 描述 |
|-----------------------|--------------------------|------|
| 0x20000000-0x200077FF | 主+辅双电源供电的可独立保持存储区 RAM30K | 30KB |
| 0x20007800-0x20007FFF | 主+辅双电源供电的可独立保持存储区 RAM2K | 2KB |

2.3.3 外设地址分配

CH583 主要包含以下外设, 每个外设占用一定的地址空间, 外设寄存器的实际访问地址为: 基地址+偏移地址。在后续章节中, 寄存器的地址有详细说明。下表为各个外设基地址的分配表。

表 2-5 外设基地址分配表

| 外设编号 | 外设名称 | 外设基址 |
|------|--|-------------|
| 1 | SYS (PMU/RTC/GPIO 等) AUX (ADC/TKEY/PLL 等) | 0x4000 1000 |
| 2 | FlashROM-Control | 0x4000 1800 |
| 3 | TMR0 | 0x4000 2000 |
| 4 | TMR1 | 0x4000 2400 |
| 5 | TMR2 | 0x4000 2800 |
| 6 | TMR3 | 0x4000 2C00 |
| 7 | UART0 | 0x4000 3000 |
| 8 | UART1 | 0x4000 3400 |
| 9 | UART2 | 0x4000 3800 |
| 10 | UART3 | 0x4000 3C00 |
| 11 | SPI0 | 0x4000 4000 |
| 12 | SPI1 | 0x4000 4400 |
| 13 | I2C | 0x4000 4800 |

| | | |
|----|-------------------|----------------------------|
| 14 | PWMx (PWM4~PWM11) | 0x4000 5000 |
| 15 | USB | 0x4000 8000 |
| 16 | USB2 | 0x4000 8400 |
| 17 | Radio:BLE | 0x4000 C000 0x4000 D000 |

下表为后续章节寄存器描述中“访问”的解释说明：

表 2-6 访问属性说明

| 缩写词 | 描述 |
|-----|-----------------------|
| RF | 只读且读取值为固定值，不受复位影响。 |
| R0 | 只读。 |
| W0 | 只写，读取值为 0 或无效。 |
| RZ | 只读，读取完之后自动清 0。 |
| WZ | 写则清 0。 |
| RW | 可读可写。 |
| RW1 | 可读，写 1 则清 0。 |
| WA | 只写且仅安全模式下，读取值为 0 或无效。 |
| RWA | 可读，仅安全模式下可写。 |

下表为后续章节中使用的缩写解释说明：

表 2-7 名词缩写说明

| 缩写词 | 描述 |
|--------|----------------------------------|
| HSE | 外部高频晶体振荡时钟源（建议 32MHz） |
| LSE | 外部低频晶体振荡时钟源（建议 32KHz） |
| LSI | 内部低频 RC 时钟振荡源（应用软件运行时校准后为 32KHz） |
| CK32M | 高频时钟源（默认 32MHz） |
| CK32K | 低频时钟源（默认 32KHz） |
| Fp11 | PLL 输出时钟（默认频率为 480MHz） |
| HCLK | 系统主频时钟 |
| Fsys | 系统主频时钟频率 |
| Tsys | 系统主频时钟周期（1/Fsys） |
| RAM2K | 高地址的 2KB SRAM |
| RAM30K | 低地址的 30KB SRAM |
| 0x | 以其开头的数据表示 16 进制数 |
| H | 以其结束的数据表示 16 进制数 |
| B | 以其结束的数据表示 2 进制数 |

第 3 章 中断

系统内置快速可编程中断控制器 (PFIC: Programmable Fast Interrupt Controller), 最多支持 255 个中断向量。当前系统管理了 20 个外设中断通道和 8 个内核中断通道, 其他中断源保留。

3.1 中断控制器

20 个外设中断, 每个中断请求都有独立的触发和屏蔽控制位, 有专用的状态位。

1 个不可屏蔽中断 NMI。

特有快速中断进出机制, 硬件自动压栈和恢复, 无需指令开销。

特有快速中断响应机制, 4 路可编程直达中断向量地址。

3.2 系统 SysTick 定时器

内核自带了一个 64 位计数器 (SysTick), 支持 HCLK 或者 HCLK/8 作为时基, 具有较高优先级。

3.3 中断和异常向量

下表列出了芯片系统的向量表

表 3-1 中断向量表

| 编号 | 优先级 | 优先级类型 | 名称 | 说明 | 地址 |
|-------|-----|-------|------------|----------------------|-------------|
| 0 | - | - | - | 保留 | 0x0000_0000 |
| 1 | -3 | 固定 | Reset | 复位 | 0x0000_0004 |
| 2 | -2 | 固定 | NMI | 不可屏蔽中断 | 0x0000_0008 |
| 3 | -1 | 固定 | EXC | 所有类型的失效、异常中断 | 0x0000_000C |
| 4 | - | - | - | 保留 | - |
| 5 | -1 | 固定 | ECALL-M | 机器模式回调中断 | 0x0000_0014 |
| 6-7 | - | - | - | 保留 | - |
| 8 | -1 | 固定 | ECALL-U | 用户模式回调中断 | 0x0000_0020 |
| 9 | -1 | 固定 | BREAKPOINT | 断点回调中断 | 0x0000_0024 |
| 10-11 | - | - | - | 保留 | - |
| 12 | 0 | 可设置 | SysTick | 系统嘀嗒定时器 | 0x0000_0030 |
| 13 | - | - | - | 保留 | - |
| 14 | 1 | 可设置 | SWI | 软件中断 | 0x0000_0038 |
| 15 | - | - | - | 保留 | 0x0000_003C |
| 16 | 2 | 可设置 | TMRO | TMRO 定时器 0 中断 | 0x0000_0040 |
| 17 | 3 | 可设置 | GPIO_A | GPIO 端口 PA 通用 I/O 中断 | 0x0000_0044 |
| 18 | 4 | 可设置 | GPIO_B | GPIO 端口 PB 通用 I/O 中断 | 0x0000_0048 |
| 19 | 5 | 可设置 | SPI0 | SPI0 中断 | 0x0000_004C |
| 20 | 6 | 可设置 | BLEL | 无线模块的 LLE 中断 | 0x0000_0050 |
| 21 | 7 | 可设置 | BLEB | 无线模块的 BB 中断 | 0x0000_0054 |
| 22 | 8 | 可设置 | USB | USB 中断 | 0x0000_0058 |
| 23 | 9 | 可设置 | USB2 | USB2 中断 | 0x0000_005C |
| 24 | 10 | 可设置 | TMR1 | TMR1 定时器 1 中断 | 0x0000_0060 |
| 25 | 11 | 可设置 | TMR2 | TMR2 定时器 2 中断 | 0x0000_0064 |
| 26 | 12 | 可设置 | UART0 | UART0 异步串口 0 中断 | 0x0000_0068 |
| 27 | 13 | 可设置 | UART1 | UART1 异步串口 1 中断 | 0x0000_006C |
| 28 | 14 | 可设置 | RTC | RTC 实时时钟中断 | 0x0000_0070 |

| | | | | | |
|----|----|-----|-----------|----------------------------|-------------|
| 29 | 15 | 可设置 | ADC | ADC 和 TouchKey 中断 | 0x0000_0074 |
| 30 | 16 | 可设置 | I2C | I2C 中断 | 0x0000_0078 |
| 31 | 17 | 可设置 | PWMX_SPI1 | PWMX (PWM4~11) 中断和 SPI1 中断 | 0x0000_007C |
| 32 | 18 | 可设置 | TMR3 | TMR3 定时器 3 中断 | 0x0000_0080 |
| 33 | 19 | 可设置 | UART2 | UART2 异步串口 2 中断 | 0x0000_0084 |
| 34 | 20 | 可设置 | UART3 | UART3 异步串口 3 中断 | 0x0000_0088 |
| 35 | 21 | 可设置 | WDOG_BAT | 看门狗定时器中断/电池低电压中断 | 0x0000_008C |

3.4 寄存器

3.4.1 PFIC 寄存器描述

PFIC 相关寄存器基地址：0xE000E000

表 3-2 PFIC 相关寄存器列表

| 名称 | 偏移地址 | 描述 | 复位值 |
|-------------------|-------|--------------------|------------|
| R32_PFIC_ISR1 | 0x00 | PFIC 中断使能状态寄存器 1 | 0x00000000 |
| R32_PFIC_ISR2 | 0x04 | PFIC 中断使能状态寄存器 2 | 0x00000000 |
| R32_PFIC_IPR1 | 0x20 | PFIC 中断挂起状态寄存器 1 | 0x00000000 |
| R32_PFIC_IPR2 | 0x24 | PFIC 中断挂起状态寄存器 2 | 0x00000000 |
| R32_PFIC_ITHRESDR | 0x40 | PFIC 中断优先级阈值配置寄存器 | 0x00000000 |
| R32_PFIC_CFGR | 0x48 | PFIC 中断配置寄存器 | 0x00000000 |
| R32_PFIC_GISR | 0x4C | PFIC 中断全局状态寄存器 | 0x00000000 |
| R32_PFIC_IDCFGR | 0x50 | PFIC 快速中断 ID 配置寄存器 | 0x00000000 |
| R32_PFIC_FIADDRRO | 0x60 | PFIC 快速中断 0 地址寄存器 | 0x00000000 |
| R32_PFIC_FIADDRR1 | 0x64 | PFIC 快速中断 1 地址寄存器 | 0x00000000 |
| R32_PFIC_FIADDRR2 | 0x68 | PFIC 快速中断 2 地址寄存器 | 0x00000000 |
| R32_PFIC_FIADDRR3 | 0x6C | PFIC 快速中断 3 地址寄存器 | 0x00000000 |
| R32_PFIC_IENR1 | 0x100 | PFIC 中断使能设置寄存器 1 | 0x00000000 |
| R32_PFIC_IENR2 | 0x104 | PFIC 中断使能设置寄存器 2 | 0x00000000 |
| R32_PFIC_IRER1 | 0x180 | PFIC 中断使能清除寄存器 1 | 0x00000000 |
| R32_PFIC_IRER2 | 0x184 | PFIC 中断使能清除寄存器 2 | 0x00000000 |
| R32_PFIC_IPSR1 | 0x200 | PFIC 中断挂起设置寄存器 1 | 0x00000000 |
| R32_PFIC_IPSR2 | 0x204 | PFIC 中断挂起设置寄存器 2 | 0x00000000 |
| R32_PFIC_IPRR1 | 0x280 | PFIC 中断挂起清除寄存器 1 | 0x00000000 |
| R32_PFIC_IPRR2 | 0x284 | PFIC 中断挂起清除寄存器 2 | 0x00000000 |
| R32_PFIC_IACTR1 | 0x300 | PFIC 中断激活状态寄存器 1 | 0x00000000 |
| R32_PFIC_IACTR2 | 0x304 | PFIC 中断激活状态寄存器 2 | 0x00000000 |
| R32_PFIC_IPRIORx | 0x400 | PFIC 中断优先级配置寄存器 | 0x00000000 |
| R32_PFIC_SCTLR | 0xD10 | PFIC 系统控制寄存器 | 0x00000000 |

用户模式下，可以支持全局中断控制，请参考 EVT 评估板资料中提供的示例。

内核中断控制位说明：

- 1、Reset、NMI、EXC、ECALL-M、ECALL-U、BREAKPOINT 中断默认总是开启。
- 2、NMI、EXC 支持中断挂起清除和置位控制（PFIC_IPSR1 和 PFIC_IPRR1 寄存器控制），不支持中断使能设置和清除控制。
- 3、Reset、ECALL-M、ECALL-U、BREAKPOINT 不支持中断挂起清除和置位控制、中断使能设置和清除控制。

PFIC 中断使能状态寄存器 1 (PFIC_ISR1)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|----------|----|---|------|
| [31:12] | INTENSTA | RO | 31#及以下中断当前使能状态。 1: 当前编号中断已使能; 0: 当前编号中断未启用。 | 0 |
| [11:0] | Reserved | RO | 保留。 Reset、NMI、EXC、ECALL 等中断位, 下同。 | 0x0C |

PFIC 中断使能状态寄存器 2 (PFIC_ISR2)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|----------|----|---|-----|
| [31:4] | Reserved | RO | 保留。 | 0 |
| [3:0] | INTENSTA | RO | 32#及以上中断当前使能状态。 1: 当前编号中断已使能; 0: 当前编号中断未启用。 | 0 |

PFIC 中断挂起状态寄存器 1 (PFIC_IPR1)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|----------|----|---|-----|
| [31:12] | PENDSTA | RO | 31#及以下中断当前挂起状态。 1: 当前编号中断已挂起; 0: 当前编号中断未挂起。 | 0 |
| [11:0] | Reserved | RO | 保留。 | 0 |

PFIC 中断挂起状态寄存器 2 (PFIC_IPR2)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|----------|----|---|-----|
| [31:4] | Reserved | RO | 保留。 | 0 |
| [3:0] | PENDSTA | RO | 32#及以上中断当前挂起状态。 1: 当前编号中断已挂起; 0: 当前编号中断未挂起。 | 0 |

PFIC 中断优先级阈值配置寄存器 (PFIC_ITHRESDR)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|-----------|----|--|-----|
| [31:8] | Reserved | RO | 保留。 | 0 |
| [7:0] | THRESHOLD | RW | 中断优先级阈值设置值。 低于当前设置值的中断优先级值, 当挂起时不执行中断服务; 此寄存器为 0 时表示阈值寄存器功能无效。 [7:4]: 优先级阈值。 [3:0]: 保留, 固定为 0, 写无效。 | 0 |

PFIC 中断配置寄存器 (PFIC_CFGR)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|----------|----|---|-----|
| [31:16] | KEYCODE | WO | 对于不同的目标控制位, 需要同步写入相应的安全访问标识数据才能修改, 读出数据固定为 0: KEY1 = 0xFA05; KEY2 = 0xBCAF; KEY3 = 0xBEEF。 | 0 |
| [15:8] | Reserved | RO | 保留。 | 0 |

| | | | | |
|-------|----------|----|--|---|
| 7 | RESETSYS | WO | 系统复位（同步写入 KEY3）。自动清 0。 写 1 有效，写 0 无效。 与 PFIC_SCTLR 寄存器 SYSRESET 位作用相同。 | 0 |
| [6:0] | Reserved | RO | 保留。 | 0 |

PFIC 中断全局状态寄存器（PFIC_GISR）

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|----------|----|---|-----|
| [31:10] | Reserved | RO | 保留。 | 0 |
| 9 | GPENDSTA | RO | 当前是否有中断处于挂起： 1：有； 0：没有。 | 0 |
| 8 | GACTSTA | RO | 当前是否有中断被执行： 1：有； 0：没有。 | 0 |
| [7:0] | NESTSTA | RO | 当前中断嵌套状态，目前支持 2 级嵌套，[1:0] 有效。 3：第 2 级中断中； 1：第 1 级中断中； 0：没有中断发生； 其他：不可能情况。 | 0 |

PFIC 快速中断 ID 配置寄存器（PFIC_IDCFGR）

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|-------|----|-----------------|-----|
| [31:24] | FIID3 | RW | 配置快速中断 3 的中断编号。 | 0 |
| [23:16] | FIID2 | RW | 配置快速中断 2 的中断编号。 | 0 |
| [15:8] | FIID1 | RW | 配置快速中断 1 的中断编号。 | 0 |
| [7:0] | FIID0 | RW | 配置快速中断 0 的中断编号。 | 0 |

PFIC 快速中断 0 地址寄存器（PFIC_FIADDR0）

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|-------|----|---------------------------------------|-----|
| [31:1] | ADDR0 | RW | 快速中断 0 服务程序地址 bit[31:1], bit0 为 0。 | 0 |
| 0 | FI0EN | RW | 快速中断 0 通道使能位： 1：启用快速中断 0 通道； 0：关闭。 | 0 |

PFIC 快速中断 1 地址寄存器（PFIC_FIADDR1）

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|-------|----|---------------------------------------|-----|
| [31:1] | ADDR1 | RW | 快速中断 1 服务程序地址 bit[31:1], bit0 为 0。 | 0 |
| 0 | FI1EN | RW | 快速中断 1 通道使能位： 1：启用快速中断 1 通道； 0：关闭。 | 0 |

PFIC 快速中断 2 地址寄存器（PFIC_FIADDR2）

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|-------|----|---------------------------------------|-----|
| [31:1] | ADDR2 | RW | 快速中断 2 服务程序地址 bit[31:1], bit0 为 0。 | 0 |
| 0 | FI2EN | RW | 快速中断 2 通道使能位： 1：启用快速中断 2 通道； 0：关闭。 | 0 |

PFIC 快速中断 3 地址寄存器（PFIC_FIADDR3）

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---|----|----|----|-----|
|---|----|----|----|-----|

| | | | | |
|--------|-------|----|---|---|
| [31:1] | ADDR3 | RW | 快速中断 3 服务程序地址 bit[31:1], bit0 为 0。 | 0 |
| 0 | FI3EN | RW | 快速中断 3 通道使能位: 1: 启用快速中断 3 通道; 0: 关闭。 | 0 |

PFIC 中断使能设置寄存器 1 (PFIC_IENR1)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|----------|----|--|-----|
| [31:12] | INTEN | WO | 31#及以下中断使能控制。 1: 当前编号中断使能; 0: 无影响。 | 0 |
| [11:0] | Reserved | RO | 保留。 | 0 |

PFIC 中断使能设置寄存器 2 (PFIC_IENR2)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|----------|----|--|-----|
| [31:4] | Reserved | RO | 保留。 | 0 |
| [3:0] | INTEN | WO | 32#及以上中断使能控制。 1: 当前编号中断使能; 0: 无影响。 | 0 |

PFIC 中断使能清除寄存器 1 (PFIC_IRER1)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|----------|----|--|-----|
| [31:12] | INTRESET | WO | 31#及以下中断关闭控制。 1: 当前编号中断关闭; 0: 无影响。 | 0 |
| [11:0] | Reserved | RO | 保留。 | 0 |

PFIC 中断使能清除寄存器 2 (PFIC_IRER2)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|----------|----|--|-----|
| [31:4] | Reserved | RO | 保留。 | 0 |
| [3:0] | INTRESET | WO | 32#及以上中断关闭控制。 1: 当前编号中断关闭; 0: 无影响。 | 0 |

PFIC 中断挂起设置寄存器 1 (PFIC_IPSR1)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|----------|----|--|-----|
| [31:12] | PENDSET | WO | 31#及以下中断挂起设置。 1: 当前编号中断挂起; 0: 无影响。 | 0 |
| [11:0] | Reserved | RO | 保留。 | 0 |

PFIC 中断挂起设置寄存器 2 (PFIC_IPSR2)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|----------|----|--|-----|
| [31:4] | Reserved | RO | 保留。 | 0 |
| [3:0] | PENDSET | WO | 32#及以上中断挂起设置。 1: 当前编号中断挂起; 0: 无影响。 | 0 |

PFIC 中断挂起清除寄存器 1 (PFIC_IPRR1)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|-----------|----|--|-----|
| [31:12] | PENDRESET | WO | 31#及以下中断挂起清除。 1: 当前编号中断清除挂起状态; 0: 无影响。 | 0 |
| [11:0] | Reserved | RO | 保留。 | 0 |

PFIC 中断挂起清除寄存器 2 (PFIC_IPRR2)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|-----------|----|--|-----|
| [31:4] | Reserved | RO | 保留。 | 0 |
| [3:0] | PENDRESET | WO | 32#及以上中断挂起清除。 1: 当前编号中断清除挂起状态; 0: 无影响。 | 0 |

PFIC 中断激活状态寄存器 1 (PFIC_IACTR1)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|----------|-----|---|-----|
| [31:12] | IACTS | RW1 | 31#及以下中断执行状态。 1: 当前编号中断执行中; 0: 当前编号中断未执行。 | 0 |
| [11:0] | Reserved | RO | 保留。 | 0 |

PFIC 中断激活状态寄存器 2 (PFIC_IACTR2)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|----------|-----|---|-----|
| [31:4] | Reserved | RO | 保留。 | 0 |
| [3:0] | IACTS | RW1 | 32#及以上中断执行状态。 1: 当前编号中断执行中; 0: 当前编号中断未执行。 | 0 |

PFIC 中断优先级配置寄存器 (PFIC_IPRIOR_x) (x=0-63)

控制器支持 256 个中断 (0-255)，每个中断使用 8bit 来设置控制优先级。

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

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------------|--------|-----|---------------------------------|-----|
| [2047:2040] | IP_255 | RW | 同 IP_0 描述。 | 0 |
| ... | ... | ... | ... | ... |
| [31:24] | IP_3 | RW | 同 IP_0 描述。 | 0 |
| [23:16] | IP_2 | RW | 同 IP_0 描述。 | 0 |
| [15:8] | IP_1 | RW | 同 IP_0 描述。 | 0 |
| [7:0] | IP_0 | RW | 编号 0 中断优先级配置: [7:4]: 优先级控制位。 | 0 |

| | | | | |
|--|--|--|--|--|
| | | | [3:0]: 保留, 固定为 0, 写无效。 优先级数值越小则优先级越高。只有 2 级中断嵌套, 即只能抢占 1 次。 | |
|--|--|--|--|--|

PFIC 系统控制寄存器 (PFIC_SCTLR)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|-------------|----|--|-----|
| 31 | SYSRESET | WO | 系统复位。自动清 0。写 1 有效, 写 0 无效。 与 PFIC_CFGR 寄存器相同效果。 | 0 |
| [30:6] | Reserved | RO | 保留。 | 0 |
| 5 | SETEVENT | WO | 设置事件, 可以唤醒 WFE 的情况。 | 0 |
| 4 | SEVONPEND | RW | 当发生事件或者中断挂起状态时, 可以从 WFE 指令后唤醒系统, 如果未执行 WFE 指令, 将在下次执行该指令后立即唤醒系统。 1: 启用的事件和所有中断 (包括未开启中断) 都能唤醒系统; 0: 只有启用的事件和启用的中断可以唤醒系统。 | 0 |
| 3 | WFIOWFE | RW | 将 WFI 指令当成是 WFE 执行。 1: 将之后的 WFI 指令当做 WFE 指令; 0: 无作用。 | 0 |
| 2 | SLEEPDEEP | RW | 控制系统的低功耗模式: 1: deepsleep 0: sleep | 0 |
| 1 | SLEEPONEXIT | RW | 控制离开中断服务程序后, 系统状态: 1: 系统进入低功耗模式; 0: 系统进入主程序。 | 0 |
| 0 | Reserved | RO | 保留。 | 0 |

3.4.2 WCH 定义 CSR 寄存器

RISC-V 架构中定义了一些控制和状态寄存器 (Control and Status Register, CSR), 用于配置或标识或记录运行状态。CSR 寄存器属于内核内部的寄存器, 使用专用的 12 位地址空间。WCH 芯片除了 RISC-V 特权架构文档中定义的标准寄存器外, 还增加了一些厂商定义寄存器, 需要使用 `csr` 指令进行访问。说明: 此类寄存器标注为“MRW”属性的需要系统在机器模式下才能访问。

中断系统控制寄存器 (INTSYSCR)

CSR 地址: 0x804

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|----------|-----|-------------------------------|-----|
| [31:2] | Reserved | RO | 保留。 | 0 |
| 1 | INESTEN | MRW | 中断嵌套功能使能。 1: 使能; 0: 关闭。 | 0 |
| 0 | HWSTKEN | MRW | 硬件压栈功能使能。 1: 使能; 0: 关闭。 | 0 |

异常入口基地址寄存器 (MTVEC)

CSR 地址: 0x305

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|---------|----|---|-----|
| [31:2] | BASEADD | RO | 中断向量表基地址。 | 0 |
| 1 | MODE1 | RW | 中断向量表识别模式。 1: 按绝对地址识别, 支持全范围, 但必须跳转; | 0 |

| | | | | |
|---|-------|----|--|---|
| | | | 0: 按跳转指令识别, 有限范围, 支持非跳指令。 | |
| 0 | MODEO | RW | 中断或异常入口地址模式选择: 1: 根据其编号*4 进行地址偏移; 0: 使用统一入口地址。 | 0 |

3.4.3 物理内存保护单元 PMP

为了提高系统安全, RISC-V 架构中定义了一套物理地址访问限制, 可以为区域内物理内存设置其读、写、执行属性, 区域长度最小 4 字节保护。PMP 单元在用户模式下一直生效, 在机器模式下可选生效, 如果违背了当前内存限制, 将会产生系统异常中断 (EXC)。

PMP 单元包含 4 组 8bit 的配置寄存器 (合计 32bit) 和 4 组地址寄存器, 需要使用 csr 指令进行访问, 并且在机器模式下进行。

PMP 配置寄存器 (PMPCFG0)

CSR 地址: 0x3A0

| 位 | 名称 | 访问 | 描述 | 复位值 | | | | | | | | | | | | | | | | | | | | | |
|---------|---------|--------|---|-----|--|----|---|---|--|-------|---|----|-------|---|----------------|---|---|--------|---|---|--------|---|---|-------|---|
| [31:24] | pmp3cfg | MRW | 见 pmp0cfg。 | 0 | | | | | | | | | | | | | | | | | | | | | |
| [23:16] | pmp2cfg | MRW | 见 pmp0cfg。 | 0 | | | | | | | | | | | | | | | | | | | | | |
| [15:8] | pmp1cfg | MRW | 见 pmp0cfg。 | 0 | | | | | | | | | | | | | | | | | | | | | |
| [7:0] | pmp0cfg | MRW | <table border="1"> <thead> <tr> <th>位</th> <th>名称</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>7</td> <td>L</td> <td>锁定使能, 机器模式下可解锁: 1: 锁定相关寄存器; 0: 不锁定。</td> </tr> <tr> <td>[6:5]</td> <td>-</td> <td>保留</td> </tr> <tr> <td>[4:3]</td> <td>A</td> <td>地址对齐及保护区域范围选择。</td> </tr> <tr> <td>2</td> <td>X</td> <td>可执行属性。</td> </tr> <tr> <td>1</td> <td>W</td> <td>可写入属性。</td> </tr> <tr> <td>0</td> <td>R</td> <td>可读出属性</td> </tr> </tbody> </table> | 位 | 名称 | 描述 | 7 | L | 锁定使能, 机器模式下可解锁: 1: 锁定相关寄存器; 0: 不锁定。 | [6:5] | - | 保留 | [4:3] | A | 地址对齐及保护区域范围选择。 | 2 | X | 可执行属性。 | 1 | W | 可写入属性。 | 0 | R | 可读出属性 | 0 |
| | | | 位 | 名称 | 描述 | | | | | | | | | | | | | | | | | | | | |
| | | | 7 | L | 锁定使能, 机器模式下可解锁: 1: 锁定相关寄存器; 0: 不锁定。 | | | | | | | | | | | | | | | | | | | | |
| | | | [6:5] | - | 保留 | | | | | | | | | | | | | | | | | | | | |
| | | | [4:3] | A | 地址对齐及保护区域范围选择。 | | | | | | | | | | | | | | | | | | | | |
| | | | 2 | X | 可执行属性。 | | | | | | | | | | | | | | | | | | | | |
| 1 | W | 可写入属性。 | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | R | 可读出属性 | | | | | | | | | | | | | | | | | | | | | | | |

地址对齐及保护区域范围选择, 对于 $A_ADDR \leq region < B_ADDR$ 区域进行内存保护 (要求 A_ADDR 和 B_ADDR 均 4 字节对齐):

- 1、如果 $B_ADDR - A_ADDR == 2^2$, 则采用 NA4 方式;
- 2、如果 $B_ADDR - A_ADDR == 2^{(G+2)}$, $G \geq 1$, 且 A_ADDR 为 $2^{(G+2)}$ 对齐则采用 NAPOT 方式;
- 3、否则使用 TOR 方式。

| A 值 | 方式 | 描述 |
|-----|-------|--|
| 0 | OFF | 没有区域要保护 |
| 1 | TOR | 顶端对齐区域保护: pmp0cfg 下, $0 \leq region < pmpaddr0$; pmp1cfg 下, $pmpaddr0 \leq region < pmpaddr1$; pmp2cfg 下, $pmpaddr1 \leq region < pmpaddr2$; pmp3cfg 下, $pmpaddr2 \leq region < pmpaddr3$ 。 $pmpaddr_{i-1} = A_ADDR \gg 2$; $pmpaddr_i = B_ADDR \gg 2$ 。 |
| 2 | NA4 | 固定 4 字节区域保护。 pmp0cfg~pmp3cfg 对应 pmpaddr0~pmpaddr3 作为起始地址。 $pmpaddr_i = A_ADDR \gg 2$ 。 |
| 3 | NAPOT | 保护 $2^{(G+2)}$ 区域, $G \geq 1$, 其中 A_ADDR 为 $2^{(G+2)}$ 对齐。 $pmpaddr_i = (A_ADDR \gg 2) (2^{(G-1)} - 1)$ 。 |

PMP 地址 0 寄存器 (PMPADDR0)

CSR 地址: 0x3B0

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|-------|-----|----------------------------------|-----|
| [31:0] | ADDR0 | MRW | PMP 设置地址 0 的 bit[33:2]。实际高 2 位未用 | 0 |

PMP 地址 1 寄存器 (PMPADDR1)

CSR 地址: 0x3B1

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|-------|-----|----------------------------------|-----|
| [31:0] | ADDR1 | MRW | PMP 设置地址 1 的 bit[33:2]。实际高 2 位未用 | 0 |

PMP 地址 2 寄存器 (PMPADDR2)

CSR 地址: 0x3B2

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|-------|-----|----------------------------------|-----|
| [31:0] | ADDR2 | MRW | PMP 设置地址 2 的 bit[33:2]。实际高 2 位未用 | 0 |

PMP 地址 3 寄存器 (PMPADDR3)

CSR 地址: 0x3B3

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|-------|-----|----------------------------------|-----|
| [31:0] | ADDR3 | MRW | PMP 设置地址 3 的 bit[33:2]。实际高 2 位未用 | 0 |

3.4.4 SysTick 寄存器描述

STK 相关寄存器基地址: 0xE000F000

表 3-3 STK 相关寄存器列表

| 名称 | 偏移地址 | 描述 | 复位值 |
|---------------|------|------------|------------|
| R32_STK_CTRL | 0x00 | 系统计数控制寄存器 | 0x00000000 |
| R32_STK_SR | 0x04 | 系统计数状态寄存器 | 0x00000000 |
| R32_STK_CNTL | 0x08 | 系统计数器低位寄存器 | 0x00000000 |
| R32_STK_CNTH | 0x0C | 系统计数器高位寄存器 | 0x00000000 |
| R32_STK_CMPLR | 0x10 | 计数重加载低位寄存器 | 0x00000000 |
| R32_STK_CMPHR | 0x14 | 计数重加载高位寄存器 | 0x00000000 |

系统计数控制寄存器 (STK_CTRL)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|----------|----|---|-----|
| 31 | SWIE | RW | 软件中断触发使能 (SWI): 1: 触发软件中断; 0: 关闭触发。 进入软件中断后须将此位清 0, 否则一直触发。 | 0 |
| [30:6] | Reserved | RO | 保留。 | 0 |
| 5 | INIT | W1 | 计数器初始值更新: 1: 向上计数从 0 开始, 向下计数从比较值开始; 0: 无效。 | 0 |
| 4 | MODE | RW | 计数模式: 1: 向下计数; 0: 向上计数。 | 0 |
| 3 | STRE | RW | 自动重装载计数使能: 1: 向上计数到比较值之后从 0 开始计数, 向下计数到 0 之后从比较值开始向下计数; 0: 继续向上/向下计数。 | 0 |

| | | | | |
|---|-------|----|---|---|
| 2 | STCLK | RW | 计数器时钟源选择： 1: HCLK 做时基； 0: HCLK/8 做计数时基。 | 0 |
| 1 | STIE | RW | 计数器中断使能控制位： 1: 使能计数器中断； 0: 无计数器中断。 | 0 |
| 0 | STE | RW | 系统计数器使能控制位： 1: 启动系统计数器 STK； 0: 关闭系统计数器 STK，计数器停止计数。 | 0 |

计数状态寄存器 (STK_SR)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|----------|-----|--|-----|
| [31:1] | Reserved | RO | 保留。 | 0 |
| 0 | CNTIF | RWO | 计数值比较标志，写 0 清除，写 1 无效： 1: 向上计数达到比较值，向下计数达到 0； 0: 未达到比较值。 | 0 |

系统计数器低位寄存器 (STK_CNTL)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|------|----|-------------------|-----|
| [31:0] | CNTL | RW | STK 计数器计数值低 32 位。 | 0 |

系统计数器高位寄存器 (STK_CNTH)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|------|----|-------------------|-----|
| [31:0] | CNTH | RW | STK 计数器计数值高 32 位。 | 0 |

计数重加载低位寄存器 (STK_CMPLR)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|------|----|------------------|-----|
| [31:0] | CMPL | RW | 设置重加载计数器值低 32 位。 | 0 |

计数重加载高位寄存器 (STK_CMPHR)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|------|----|------------------|-----|
| [31:0] | CMPH | RW | 设置重加载计数器值高 32 位。 | 0 |

第 4 章 系统控制

4.1 复位控制

系统支持 6 种复位形式，分别为电源上电复位 RPOR(real power on reset)、外部手工复位 MR(manual reset)、内部软件复位 SR(software reset)、看门狗超时复位 WTR(watch-dog time-out reset)、下电模式下唤醒导致的全局复位 GRWSM(global reset by waking under shutdown mode)、常规唤醒导致的局部寄存器复位 LRW(local reset by waking)。

寄存器 R8_GLOB_RESET_KEEP 和 RB_ROM_CODE_OFS，只在 RPOR 或者 GRWSM 复位时被复位，而不受其它复位形式影响。

下图中时序参数和复位特性参数请参考第 20.5 节的时序参数表。

4.1.1 电源上电复位 RPOR

当电源上电时，芯片内部 POR 模块会产生上电复位，并延时以等待电源稳定。另外，在运行过程中，当电源电压低于 V_{lvr} 时，芯片内部 LVR 模块会产生低压复位直到电压回升，并延时以等待电源稳定。下图为上电复位过程和低压复位过程。

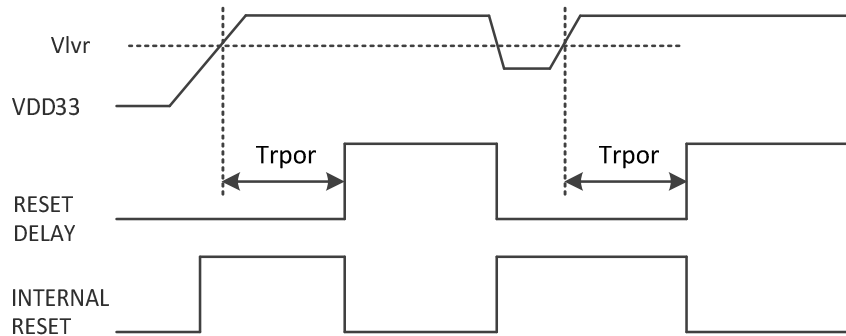


图 4-1 上电复位

4.1.2 外部手工复位 MR

外部手工复位由外部加到 RST#引脚的低电平触发，当复位低电平持续时间大于最小复位脉冲宽度 ($Trst$) 时即触发系统进行复位。

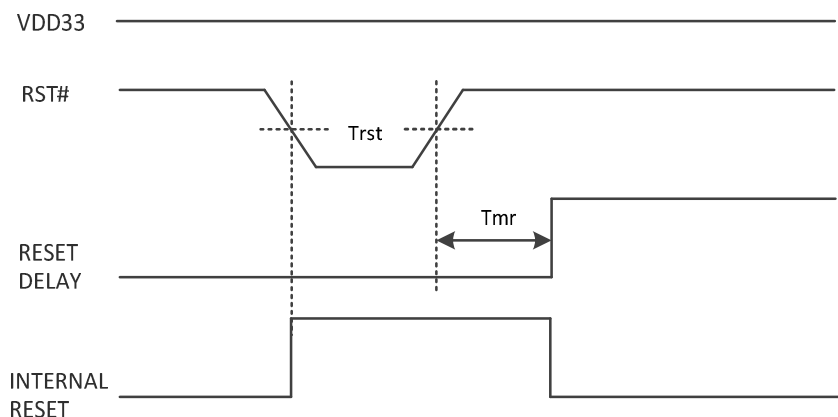


图 4-2 外部复位

4.1.3 内部软件复位 SR

内部软件复位，用于不需要外部干预而自行复位。设置全局复位配置寄存器 (R8_RST_WDOG_CTRL)

的位 RB_SOFTWARE_RESET 为 1，即可实现软件复位。该位会自动清 0。

4.1.4 看门狗超时复位 WTR

看门狗功能是基于一个 8 位的递增计数器，计数时钟周期为 $131072/F_{sys}$ 。当开启了看门狗超时复位功能后，一旦此计数器溢出会复位整个系统。

4.1.5 下电模式唤醒后的复位 GRWSM

一旦系统进入下电模式（详见电源管理章节）后，在唤醒信号的作用下，系统将有序执行唤醒操作，唤醒之后系统将执行全局复位，此复位效果类似于上电复位。

4.1.6 常规唤醒操作引起的复位 LRW

如果系统是从睡眠模式中被唤醒，则在相关电源准备就绪之后会产生复位，此复位为局部复位，根据需要对睡眠模式下掉电的寄存器进行有选择的复位。

在睡眠模式下，各功能模块的寄存器分为三类：

第一种是属于需数据保持的功能模块的关键寄存器（例如配置/模式等），睡眠时由辅助电源继续供电，数据不丢失，睡眠和唤醒对其数据无影响；

第二种是属于需数据保持的功能模块的可再生寄存器（例如计数器、FIFO 等），睡眠时断电，唤醒后数据是随机数（例如 FIFO 存储单元）或者被复位（例如 FIFO 计数器）；

第三种是属于无需数据保持的功能模块的寄存器，睡眠时断电，唤醒后数据是随机数（例如 FIFO 存储单元）或者被复位（例如 FIFO 计数器、配置/模式寄存器）。

LRW 就是用于上述后两种被复位的寄存器。

4.2 安全访问

系统某些寄存器的属性是“RWA”或者“WA”，表示当前寄存器为安全访问寄存器，可以直接读取但是写入需要进入安全访问模式。

先写入 R8_SAFE_ACCESS_SIG 寄存器 0x57；

再写入 R8_SAFE_ACCESS_SIG 寄存器 0xA8；

即可进入安全访问模式，此时可以操作具有“RWA/WA”属性的寄存器，此后约 16 个系统主频周期(T_{sys})都处于安全模式下，该有效期内可以改写一个或多个安全类寄存器，超出上述有效期后将自动终止安全模式。或者可提前向 R8_SAFE_ACCESS_SIG 寄存器写入 0x00 提前终止安全模式。

4.3 寄存器描述

表 4-1 系统控制相关寄存器列表

| 名称 | 访问地址 | 描述 | 复位值 |
|--------------------|------------|------------------|------------|
| R8_SAFE_ACCESS_SIG | 0x40001040 | 安全访问标记寄存器 | 0x00 |
| R8_CHIP_ID | 0x40001041 | 芯片 ID 寄存器 | 0x83 |
| R8_SAFE_ACCESS_ID | 0x40001042 | 安全访问 ID 寄存器 | 0x0C |
| R8_WDOG_COUNT | 0x40001043 | 看门狗计数器寄存器 | 0x00 |
| R8_RESET_STATUS | 0x40001044 | 复位状态寄存器 | 0x01 |
| R8_GLOB_ROM_CFG | 0x40001044 | FlashROM 应用配置寄存器 | 0x01 |
| R8_GLOB_CFG_INFO | 0x40001045 | 全局配置信息状态寄存器 | 0xEX |
| R8_RST_WDOG_CTRL | 0x40001046 | 看门狗及复位配置寄存器 | 0x00 |
| R8_GLOB_RESET_KEEP | 0x40001047 | 复位保持寄存器 | 0x00 |
| R32_FLASH_DATA | 0x40001800 | FlashROM 字数据寄存器 | 0xFFFFFFFF |
| R32_FLASH_CONTROL | 0x40001804 | FlashROM 控制寄存器 | 0x074000XX |
| R8_FLASH_DATA | 0x40001804 | FlashROM 字节数据寄存器 | 0xFF |
| R8_FLASH_CTRL | 0x40001806 | FlashROM 存取控制寄存器 | 0x40 |

| | | | |
|--------------|------------|------------------|------|
| R8_FLASH_CFG | 0x40001807 | FlashROM 存取配置寄存器 | 0x07 |
|--------------|------------|------------------|------|

安全访问标记寄存器 (R8_SAFE_ACCESS_SIG)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|--------------------|----|--|------|
| [7:0] | R8_SAFE_ACCESS_SIG | WO | 安全访问标记寄存器。 部分寄存器(访问属性为 RWA)为保护寄存器, 必须进入安全访问模式才能进行写操作。对该寄存器先写入 0x57, 再写入 0xA8, 即可进入安全访问模式, 并且限时约 16 个主时钟周期(Tsys), 超过则自动保护。可以写入其它任意值强制直接退出安全访问模式, 回到保护状态。 | 00h |
| 7 | Reserved | RO | 保留。 | 0 |
| [6:4] | RB_SAFE_ACC_TIMER | RO | 当前安全访问时间计数。 | 000b |
| 3 | RB_SAFE_ACC_ACT | RO | 当前安全访问模式状态: 1: 未锁定/安全访问模式下, 可写; 0: 锁定, RWA 属性寄存器不可改写。 | 0 |
| 2 | Reserved | RO | 保留。 | 0 |
| [1:0] | RB_SAFE_ACC_MODE | RO | 当前安全访问模式状态: 11: 安全模式, 可写入属性 RWA 寄存器; 其他: 非安全模式。 | 00b |

芯片 ID 寄存器 (R8_CHIP_ID)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|------------|----|---|-------------------|
| [7: 0] | R8_CHIP_ID | RF | CH583: 固定值 83h, 用于识别芯片。 CH582: 固定值 82h, 用于识别芯片。 CH581: 固定值 81h, 用于识别芯片。 | 83h 82h 81h |

安全访问 ID 寄存器 (R8_SAFE_ACCESS_ID)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|-------------------|----|----------|-----|
| [7: 0] | R8_SAFE_ACCESS_ID | RF | 固定值 0Ch。 | 0Ch |

看门狗计数器寄存器 (R8_WDOG_COUNT)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|---------------|----|--|-----|
| [7: 0] | R8_WDOG_COUNT | RW | 可预置初值的看门狗计数器, 一直自动递增, 可从 0xFF 循环到 0x00 再继续。 计数周期 = 131072/Fsys。 | 00h |

复位状态寄存器 (R8_RESET_STATUS)、FlashROM 应用配置寄存器 (R8_GLOB_ROM_CFG)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---|----------------|-----|---|-----|
| 7 | RB_ROM_CODE_WE | RWA | FlashROM 程序存储区 CodeFlash 的擦除/编程使能位: 1: 允许擦除/编程; 0: 该区擦写保护。 | 0 |
| 6 | RB_ROM_DATA_WE | RWA | FlashROM 数据存储区 DataFlash 的擦除/编程使能位: 1: 允许擦除/编程; 0: 全部擦写保护。 | 0 |
| 5 | RB_ROM_CTRL_EN | RWA | FlashROM 存取控制接口使能: 1: 允许控制; 0: 禁止存取。 | 0 |

| | | | | |
|-------|-----------------|-----|---|------|
| 4 | RB_ROM_CODE_OFS | RWA | 选择用户程序代码在 FlashROM 的起始偏移地址,该值不受 MR、SR、WTR 或 GRWSM 影响,仅在 RPOR 有效时才能清零: 0: 0x000000; 1: 0x040000 (跳过 ROM 中前 256KB)。 | 0 |
| 3 | Reserved | R0 | 保留。 | 0 |
| [2:0] | RB_RESET_FLAG | R0 | 最近一次复位状态: 000: 软件复位 SR (RB_WDOG_RST_EN=0 时软件复位可产生此状态, 否则可复位但不产生此状态); 001: 上电复位 RPOR; 010: 看门狗超时复位 WTR; 011: 外部手动复位 MR; 101: 从下电模式唤醒时的复位 GRWSM; 100/110/111: 唤醒复位 LRW, 且此前的上一次复位分别是 SR/WTR/MR。 | 001b |

全局配置信息状态寄存器 (R8_GLOB_CFG_INFO)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|-----------------|----|--|-----|
| [7: 6] | Reserved | R0 | 保留。 | 11b |
| 5 | RB_BOOT_LOADER | R0 | Boot loader 状态: 1: 当前处于 Boot loader 状态; 0: 当前处于用户程序状态。 | 1/0 |
| 4 | RB_CFG_DEBUG_EN | R0 | 两线仿真调试接口使能状态: 1: 可仿真调试, 可读取 FlashROM; 0: 禁止仿真调试。 | 0 |
| 3 | RB_CFG_BOOT_EN | R0 | 系统引导程序 BootLoader 使能状态: 1: 已启用; 0: 未启用。 | 1 |
| 2 | RB_CFG_RESET_EN | R0 | RST#外部手动复位输入使能状态: 1: 已启用; 0: 未启用。 | 0 |
| 1 | Reserved | R0 | 保留。 | 0 |
| 0 | RB_CFG_ROM_READ | R0 | FlashROM 代码和数据区保护状态: 1: 外部编程器可读; 0: 保护, 外部不可访问, 程序保密。 | 0 |

看门狗及复位配置寄存器 (R8_RST_WDOG_CTRL)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|------------------|-----|---|------|
| [7:5] | Reserved | R0 | 保留。 | 000b |
| 4 | RB_WDOG_INT_FLAG | RW1 | 看门狗定时器中断标志: 1: 发生了看门狗计数溢出, 即检测到 R8_WDOG_COUNT 递增 0xFF 到 0x00; 0: 看门狗计数未溢出。 标志写 1 清 0, 或者重新加载看门狗计数器值 (R8_WDOG_COUNT) 清 0, 或者执行 __SEV() 清 0。 | 0 |
| 3 | Reserved | R0 | 保留。 | 0 |
| 2 | RB_WDOG_INT_EN | RWA | 看门狗定时器中断使能位: 1: 使能, 看门狗计数溢出后产生中断; 0: 关闭看门狗定时器中断。 | 0 |

| | | | | |
|---|-------------------|-----------|--|---|
| 1 | RB_WDOG_RST_EN | RWA | 看门狗超时复位使能位： 1：使能，看门狗计数溢出后系统复位； 0：仅作为看门狗定时器。 注：此位置 1 后软件复位操作将不影响 RB_RESET_FLAG 状态。 | 0 |
| 0 | RB_SOFTWARE_RESET | WA/ WZ | 系统软件复位控制，复位后将自动清零： 1：执行系统软件复位； 0：空闲，无动作。 | 0 |

复位保持寄存器 (R8_GLOB_RESET_KEEP)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|--------------------|----|--|-----|
| [7:0] | R8_GLOB_RESET_KEEP | RW | 复位保持寄存器，该寄存器值不受手动复位、软件复位、看门狗复位或者普通唤醒复位的影响。 | 00h |

有关 FlashROM 的操作或者设置可以参考相关子程序，本手册不提供有关 FlashROM 字数据寄存器和 FlashROM 控制寄存器的说明。

4.4 Flash-ROM 操作步骤

1. 擦除 Flash-ROM，请参考和调用相关子程序。
2. 写 Flash-ROM，请参考和调用相关子程序。
3. 读 Flash-ROM，通过指向程序存储空间的指针，读取目标地址的 code 或 data。

4.5 芯片唯一 ID 号

每个芯片出厂时都具有唯一的 ID 号，即芯片身份识别号。该 ID 数据及其校验和共 8 字节，存储于芯片内部只读区域中，具体操作请参考例子程序。

第 5 章 电源控制

5.1 电源管理

CH583 内置有电源管理单元 PMU。系统电源从 VDD33 输入，通过内置的多个 LDO 电压调整器为系统的 FlashROM、系统的数字电路（包括内核和 USB 等）和系统的模拟电路（包括高频振荡器、PLL、ADC 和 RF 收发器等）提供所需的电源。GPIO 和 FlashROM 的电源从 VI033 输入。

正常工作时的电源供电分为两种方式：直接电源和 DC-DC 转换。在正常工作之外，CH583 提供了 4 种低功耗模式：空闲模式、暂停模式、睡眠模式、下电模式。

上电后默认不启用 DC-DC，而是提供直通电源，其电压纹波较小。为了降低正常运行时的系统功耗，可以选择启用 DC-DC 提升电源能耗利用率，工作电流通常将下降到直通方式的 60%左右。

为了降低睡眠时的系统功耗，可以选择关闭系统主 LDO，切换成由系统内置的超低功耗 ULP-LDO 提供辅助电源。当系统进入睡眠或下电模式时，除了电源管理和 RTC 寄存器等常供电单元外，系统的高 2KB 和低 30KB 的 SRAM、内核及所有的外设皆可选择是否维持供电，LSE/LSI 可选择是否开启。

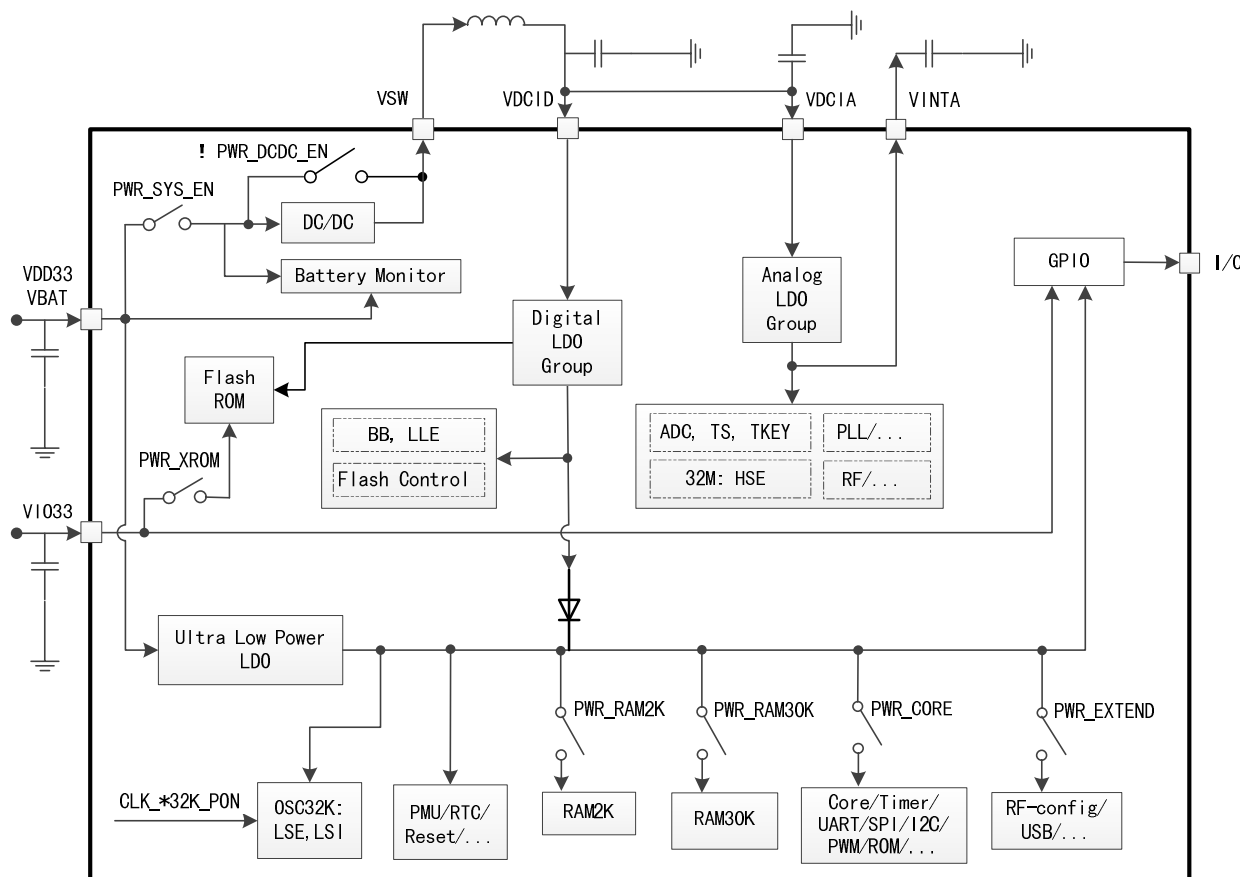


图 5-1 电源系统

5.2 寄存器描述

表 5-1 功耗管理相关寄存器列表

| 名称 | 访问地址 | 描述 | 复位值 |
|------------------|------------|-------------|--------|
| R16_SLP_CLK_OFF | 0x4000100C | 睡眠时钟控制寄存器 | 0x0000 |
| R8_SLP_CLK_OFF0 | 0x4000100C | 睡眠时钟控制寄存器 0 | 0x00 |
| R8_SLP_CLK_OFF1 | 0x4000100D | 睡眠时钟控制寄存器 1 | 0x00 |
| R8_SLP_WAKE_CTRL | 0x4000100E | 唤醒事件配置寄存器 | 0x20 |

| | | | |
|-------------------|------------|-------------|--------|
| R8_SLP_POWER_CTRL | 0x4000100F | 外设睡眠电源控制寄存器 | 0x00 |
| R16_POWER_PLAN | 0x40001020 | 睡眠电源管理寄存器 | 0x11DF |
| R16_AUX_POWER_ADJ | 0x40001022 | 辅助电源调整控制寄存器 | 0x0XXX |
| R8_BAT_DET_CTRL | 0x40001024 | 电池电压检测控制寄存器 | 0x00 |
| R8_BAT_DET_CFG | 0x40001025 | 电池电压检测配置寄存器 | 0x02 |
| R8_BAT_STATUS | 0x40001026 | 电池状态寄存器 | 0x00 |

睡眠时钟控制寄存器 0 (R8_SLP_CLK_OFF0)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---|------------------|-----|-----------------------------|-----|
| 7 | RB_SLP_CLK_UART3 | RWA | 串口 3 时钟源： 1: 关闭; 0: 开启。 | 0 |
| 6 | RB_SLP_CLK_UART2 | RWA | 串口 2 时钟源： 1: 关闭; 0: 开启。 | 0 |
| 5 | RB_SLP_CLK_UART1 | RWA | 串口 1 时钟源： 1: 关闭; 0: 开启。 | 0 |
| 4 | RB_SLP_CLK_UART0 | RWA | 串口 0 时钟源： 1: 关闭; 0: 开启。 | 0 |
| 3 | RB_SLP_CLK_TMR3 | RWA | 定时器 3 时钟源： 1: 关闭; 0: 开启。 | 0 |
| 2 | RB_SLP_CLK_TMR2 | RWA | 定时器 2 时钟源： 1: 关闭; 0: 开启。 | 0 |
| 1 | RB_SLP_CLK_TMR1 | RWA | 定时器 1 时钟源： 1: 关闭; 0: 开启。 | 0 |
| 0 | RB_SLP_CLK_TMRO | RWA | 定时器 0 时钟源： 1: 关闭; 0: 开启。 | 0 |

睡眠时钟控制寄存器 1 (R8_SLP_CLK_OFF1)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---|-----------------|-----|-------------------------------|-----|
| 7 | RB_SLP_CLK_BLE | RWA | BLE 控制器时钟源： 1: 关闭; 0: 开启。 | 0 |
| 6 | Reserved | RO | 保留。 | 0 |
| 5 | RB_SLP_CLK_USB2 | RWA | USB2 控制器时钟源： 1: 关闭; 0: 开启。 | 0 |
| 4 | RB_SLP_CLK_USB | RWA | USB 控制器时钟源： 1: 关闭; 0: 开启。 | 0 |
| 3 | RB_SLP_CLK_I2C | RWA | I2C 时钟源： 1: 关闭; 0: 开启。 | 0 |
| 2 | RB_SLP_CLK_PWMX | RWA | PWMx 时钟源： 1: 关闭; 0: 开启。 | 0 |
| 1 | RB_SLP_CLK_SPI1 | RWA | SPI1 时钟源： 1: 关闭; 0: 开启。 | 0 |
| 0 | RB_SLP_CLK_SPI0 | RWA | SPI0 时钟源： 1: 关闭; 0: 开启。 | 0 |

唤醒事件配置寄存器 (R8_SLP_WAKE_CTRL)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---|-----------------|-----|---------------|-----|
| 7 | Reserved | RO | 保留。 | 0 |
| 6 | RB_WAKE_EV_MODE | RWA | 唤醒事件内部记忆模式使能： | 0 |

| | | | | |
|---|------------------|-----|---|---|
| | | | 1: 使能记忆, 支持短脉冲事件唤醒; 0: 不记忆, 事件需保持有效直到唤醒。 | |
| 5 | RB_SLP_BAT_WAKE | RWA | 使能电池低压事件唤醒系统: 1: 使能; 0: 关闭。 | 1 |
| 4 | RB_SLP_GPIO_WAKE | RWA | 使能 GPIO 事件唤醒系统: 1: 使能; 0: 关闭。 | 0 |
| 3 | RB_SLP_RTC_WAKE | RWA | 使能 RTC 事件唤醒系统: 1: 使能; 0: 关闭。 | 0 |
| 2 | Reserved | RO | 保留。 | 0 |
| 1 | RB_SLP_USB2_WAKE | RWA | 使能 USB2 事件唤醒系统: 1: 使能; 0: 关闭。 | 0 |
| 0 | RB_SLP_USB_WAKE | RWA | 使能 USB 事件唤醒系统: 1: 使能; 0: 关闭。 | 0 |

外设睡眠电源控制寄存器(R8_SLP_POWER_CTRL)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|------------------|-----|--|-----|
| 7 | Reserved | RO | 保留。 | 0 |
| 6 | RB_RAM_RET_LV | RWA | SRAM 睡眠时辅助电源低压使能: 0: 正常电源电压, 睡眠时功耗略大; 1: 较低电源电压, 睡眠时功耗略小。 | 0 |
| 5 | RB_SLP_CLK_RAM2K | RWA | RAM2K 的 SRAM 时钟控制: 1: 关闭; 0: 开启。 | 0 |
| 4 | RB_SLP_CLK_RAMX | RWA | 主 SRAM (RAM30K) 的时钟控制: 1: 关闭; 0: 开启。 | 0 |
| [3:2] | Reserved | RO | 保留。 | 00b |
| [1:0] | RB_WAKE_DLY_MOD | RWA | 选择唤醒后的延时周期数: 11: 无延时, 8 个周期+ T_{SUCLK} , 禁用; 10: 超短延时, 70 个周期+ T_{SUCLK} ; 01: 短延时, 520 个周期+ T_{SUCLK} , 建议; 00: 长延时, 3590 个周期+ T_{SUCLK} 。 其中, T_{SUCLK} 取决于睡眠模式和时钟配置, 可能包含 T_{SUHSE} 或 PLL 或两者合计的启动 时间, 具体组合参考评估板中示例。 | 00b |

睡眠电源管理寄存器 (R16_POWER_PLAN)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|------------------|------------|--|-------|
| 15 | RB_PWR_PLAN_EN | RWA/ WZ | 睡眠电源规划控制使能: 1: 开启规划; 0: 关闭或结束规划。 开启电源规划, 用于稍后进入睡眠或下 电模式时执行, 执行后该位自动清 0。 | 0 |
| [14:11] | RB_PWR_MUST_0010 | RWA | 保留, 必须写 0010b。 | 0010b |
| 10 | RB_PWR_DCDC_PRE | RWA | DC-DC 偏置电路使能 (即时生效): 1: 使能; 0: 禁止。 | 0 |
| 9 | RB_PWR_DCDC_EN | RWA | DC-DC 使能位 (即时生效): 1: DC-DC 使能, 直通电源关闭; 0: DC-DC 禁止, 直通电源开启。 | 0 |
| 8 | RB_PWR_LDO_EN | RWA | 内部 LDO 控制 (睡眠规划): 1: 开启 LDO; 0: 规划将关闭 LDO, 更省。 | 1 |

| | | | | |
|---|---------------|-----|---|---|
| 7 | RB_PWR_SYS_EN | RWA | 系统电源控制（睡眠规划）： 1：提供系统电源（在 VSW 引脚）； 0：关闭系统电源，规划将进入睡眠模式或者下电模式。 | 1 |
| 6 | Reserved | RWA | 保留，必须写 0。 | 1 |
| 5 | Reserved | RO | 保留。 | 0 |
| 4 | RB_PWR_RAM30K | RWA | RAM30K 的 SRAM 供电（睡眠规划）： 1：双供电； 0：不用辅助电源。 | 1 |
| 3 | RB_PWR_EXTEND | RWA | USB 和 RF 配置供电（睡眠规划）： 1：双供电； 0：不用辅助电源。 | 1 |
| 2 | RB_PWR_CORE | RWA | 内核和基本外设供电（睡眠规划）： 1：双供电； 0：不用辅助电源。 | 1 |
| 1 | RB_PWR_RAM2K | RWA | RAM2K 的 SRAM 供电（睡眠规划）： 1：双供电； 0：不用辅助电源。 | 1 |
| 0 | RB_PWR_XROM | RWA | FlashROM 供电（睡眠规划）： 1：持续供电； 0：睡眠时关闭电源。 | 1 |

此寄存器除 RB_PWR_DCDC_PRE 和 RB_PWR_DCDC_EN 之外均为睡眠规划预置，其电源配置在进入低功耗的睡眠模式和下电模式后生效。

辅助电源调整控制寄存器(R16_AUX_POWER_ADJ)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|----------------|-----|--|----------------|
| [15:12] | Reserved | RO | 保留。 | 0000b |
| [11:8] | Reserved | RO | 保留，只读位，写操作不影响。 | XXXXb |
| 7 | RB_DCDC_CHARGE | RWA | 低功耗辅助 DC-DC 使能位： 1：辅助 DC-DC 使能； 0：辅助 DC-DC 禁止。 | 0 |
| 6 | Reserved | RO | 保留。 | 0 |
| [5:3] | Reserved | RWA | 保留，写入时必须保持原值不变。 | 1XXb |
| [2:0] | RB_ULPLDO_ADJ | RWA | 超低功耗 LDO 的辅助电源输出电压调节值（数值仅供参考，不建议修改）： 000: 0.77V; 001: 0.80V; 010: 0.84V; 011: 0.88V; 100: 0.91V; 101: 0.95V; 110: 0.99V; 111: 1.03V。 | XXXb (010b) |

电池电压检测控制寄存器(R8_BAT_DET_CTRL)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|-----------------|-----|---|-------|
| [7:4] | Reserved | RO | 保留。 | 0000b |
| 3 | RB_BAT_LOW_IE | RWA | 电池低电压中断使能： 1：使能； 0：关闭。 | 0 |
| 2 | RB_BAT_LOWER_IE | RWA | 电池超低电压中断使能： 1：使能； 0：关闭。 | 0 |
| 1 | RB_BAT_MON_EN | RWA | 低功耗的电池电压监控功能使能： 1：使能低压监控，增加电流约 1uA； 0：关闭低功耗的电池低压监控。 | 0 |
| 0 | RB_BAT_LOW_VTHX | RWA | 当 RB_BAT_MON_EN=1 时，设置低功耗的低压监控阈值。参考 RB_BAT_LOW_VTH。 | 0 |
| 0 | RB_BAT_DET_EN | RWA | 当 RB_BAT_MON_EN=0 时，是高精度的电 | 0 |

| | | | | |
|--|--|--|--|--|
| | | | 池电压检测功能使能： 1: 使能低压检测，同时开启基准电压等模块，睡眠模式下电流 210uA； 0: 关闭高精度的电池低压检测。 | |
|--|--|--|--|--|

注：如果电池电压达到超低电压检测阈值，且 RB_BAT_LOWER_IE 和 RB_BAT_LOW_IE 都使能（正常情况下只使能其中之一），那么将产生 NMI 不可屏蔽中断，等效于提高中断优先级。

电池电压检测配置寄存器 (R8_BAT_DET_CFG)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|----------------|-----|--|---------|
| [7:2] | Reserved | RO | 保留。 | 000000b |
| [1:0] | RB_BAT_LOW_VTH | RWA | 当 RB_BAT_MON_EN=0 时，设置高精度的超低电压及低电压的检测阈值： (超低电压参考阈值，低电压参考阈值) 00: 1.7V ， 1.95V； 01: 1.9V ， 2.15V； 10: 2.1V ， 2.35V； 11: 2.3V ， 2.55V。 当 RB_BAT_MON_EN=1 时，以 RB_BAT_LOW_VTHX 为最高位加上此 2 位合并 3 位设置低功耗的低压监控阈值： 000: 1.7V; 001: 1.8V; 010: 1.9V; 011: 2.0V; 100: 2.1V; 101: 2.2V; 110: 2.3V; 111: 2.4V。 | 10b |

电池状态寄存器 (R8_BAT_STATUS)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|-------------------|----|--|---------|
| [7:2] | Reserved | RO | 保留。 | 000000b |
| 1 | RB_BAT_STAT_LOW | RO | 电池低压检测或低压监控的结果，指示电池电压处于低电压状态： 1: 低于低电压阈值；0: 否。 | 0 |
| 0 | RB_BAT_STAT_LOWER | RO | 当 RB_BAT_MON_EN=0 时，指示电池电压处于超低电压状态： 1: 低于超低电压阈值；0: 否。 | 0 |

5.3 低功耗模式

在系统复位后，微控制器处于正常运行状态。当 MCU 不需要运行时，可以选择适当的低功耗模式来节省功耗。用户需要根据最低电源消耗、最快启动时间和可用唤醒事件等条件，选定一个合适的低功耗模式。

芯片提供以下 4 种主要的低功耗模式：

- 空闲模式 (Idle)
所有外设保持正常供电，内核停止运行，时钟系统运转。检测到唤醒事件后，可以立即唤醒。
- 暂停模式 (Halt)
在空闲模式的基础上，时钟系统停止。检测到唤醒事件后，首先时钟运转，然后唤醒内核运行。
- 睡眠模式 (Sleep):
主 LDO 关闭，由超低功耗 ULP-LDO 维持 PMU、内核和基本外设供电，LSE 或 LSI 可以选择是否开启，RAM2K、RAM30K、USB 和 RF 配置可以选择是否维持供电。检测到唤醒事件后，首先主 LDO 开启，然后时钟运转，最后唤醒内核，程序继续运行，需要时可以重新设置到更高主频。

- 下电模式 (Shutdown):

在睡眠模式的基础上，关闭了内核和基本外设以及 USB 和 RF 配置，LSE 或 LSI 可以选择是否开启，RAM2K、RAM30K 可以选择是否维持供电。检测到唤醒事件后，PMU 将执行 GRWSM 复位，软件可根据复位标志 RB_RESET_FLAG 和可选的 RAM 中的保持数据区分于 RPOR。

下表详细描述了几种低功耗模式的特征及唤醒途径：

表 5-2 低功耗模式

| 模式 | 特征 | 进入条件 | 唤醒事件 | 功耗 |
|------------------|---|---|--------------------------------|-----------------|
| 空闲模式 Idle | 外设均正常供电，内核停止运行，时钟系统运转，但可以通过外设时钟控制位选择关闭各外设的时钟。 | 设置 SLEEPDEEP=0，设置唤醒条件后执行 __WFI() 或 __WFE()。 | I/O 或 RTC 或 BAT 或 USB | 1.6mA |
| 暂停模式 Halt | 外设均正常供电，内核停止运行，时钟系统停止 (PLL/HSE 停止)。 | 设置 SLEEPDEEP=1，设置唤醒条件后执行 __WFI() 或 __WFE()。 | I/O 或 RTC 或 BAT 或 USB | 320uA |
| 睡眠模式 Sleep | 主 LDO 关闭，超低功耗 ULP-LDO 维持 PMU、内核和基本外设供电，LSE 或 LSI 可以选择是否开启，RAM2K、RAM30K、USB 和 RF 配置可以选择是否维持供电。 | 设置 SLEEPDEEP=1，设置 POWER_PLAN，设置唤醒条件后执行 __WFI() 或 __WFE()。 | I/O 或 RTC 或 BAT。 芯片唤醒后会继续运行 | 0.7uA~ 2.8uA |
| 下电模式 Shutdown | 超低功耗 LDO 维持 PMU 供电，LSE 或 LSI 可以选择是否开启，RAM2K、RAM30K 可以选择是否维持供电，用于保持数据。 | 设置 SLEEPDEEP=1，设置 POWER_PLAN，设置唤醒条件后执行 __WFI() 或 __WFE()。 | I/O 或 RTC 或 BAT。 芯片唤醒后会自动复位 | 0.2uA~ 2.3uA |

下表描述了几种低功耗模式的详细配置：

表 5-3 低功耗模式详细配置示例

| 规划配置 | SYS_EN | RAM2K | RAM30K | CK32K | CORE | EXTEND | 功耗 (仅供参考) |
|--------------|-------------|------------|-------------|-----------------------|-----------------|----------------|------------------------------|
| 维持供电的功能 | 系统电源 VSW | 数据区 2KB | 数据区 30KB | LSE/LSI 二选一 RTC 唤醒 | CPU 内核和 基本外设 | USB 和 RF 配置 | PMU 和 RTC 寄存器 常供电，约 0.2uA |
| 下电模式 常用配置 | 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 | 1 | 0 | 1 | 0 | 0 | 0.8uA |
| 睡眠模式 常用配置 | 0 | 1 | 0 | 0 | 1 | 0 | 0.7uA |
| | 0 | 1 | 0 | 1 | 1 | 0 | 1.0uA |
| | 0 | 0 | 1 | 0 | 1 | 0 | 1.8uA |
| | 0 | 0 | 1 | 1 | 1 | 0 | 2.1uA |
| | 0 | 1 | 1 | 1 | 1 | 1 | 2.4uA |

5.4 DC-DC 操作步骤

启用 DC-DC 电源模式 (启用前需确认外部硬件电路上存在 DC-DC 所需的电感和电容)

- (1)、进入安全访问模式：寄存器 R8_SAFE_ACCESS_SIG 先写入 0x57，再写入 0xA8；
- (2)、打开 DC-DC 偏置电路：将寄存器 R16_POWER_PLAN 的 RB_PWR_DCDC_PRE 置 1；
- (3)、打开 DC-DC 电源：将寄存器 R16_POWER_PLAN 的 RB_PWR_DCDC_EN 置 1，启用 DC-DC。

关闭 DC-DC，切换到直通电源模式

- (1)、进入安全访问模式：寄存器 R8_SAFE_ACCESS_SIG 先写入 0x57，再写入 0xA8；

(2)、清零寄存器 R16_POWER_PLAN 的 RB_PWR_DCDC_EN 和 RB_PWR_DCDC_PRE 控制位。

第 6 章 系统时钟及 RTC

6.1 系统时钟简介

以下几种不同的时钟源可被选择来驱动系统时钟 HCLK (F_{sys})

- HSE 之分频。
- 内部 PLL (默认 480MHz) 分频。
- LSE 或 LSI 原始时钟 CK32K。

任意一个时钟源都可以被独立地启动或关闭，由此可以优化系统功耗。

6.1.1 时钟结构

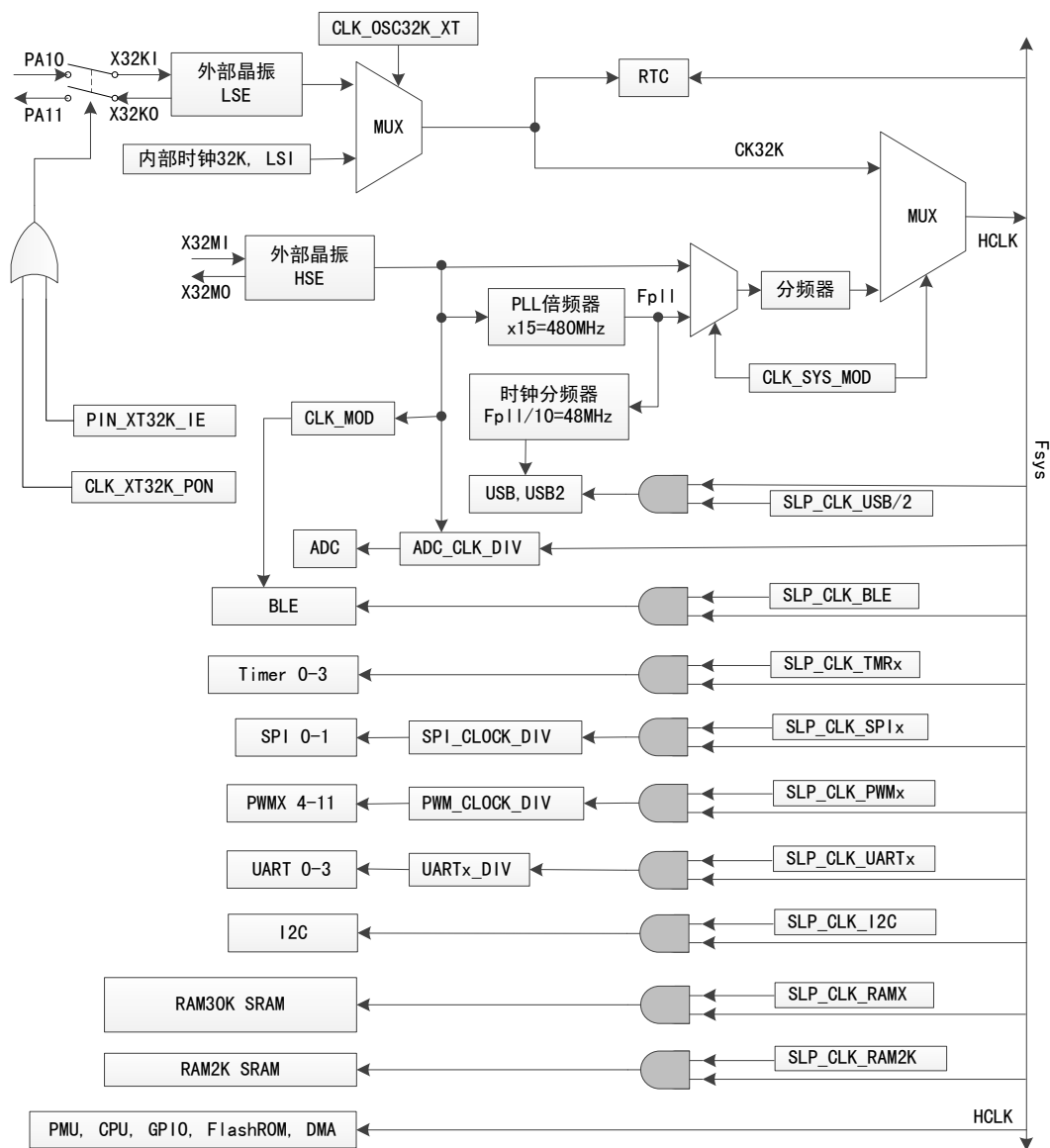


图 6-1 时钟树框图

上图是系统内部的时钟树结构，其中的 RTC 功能使用的是 32KHz 时钟源 CK32K，所以使用这些功能必须打开低频时钟；USB 的数据传输依靠 PLL 分频后产生的时钟源；其他的外设驱动时钟以及数字控制逻辑由系统时钟或者再分频驱动。

6.2 RTC 简介

实时时钟（RTC）是一个独立的定时器，包含一组连续计数的计数器。在相应软件配置下，可提供简单日历功能。修改计数器的值可以重新设置当前的时间和日期。

RTC 寄存器与 PMU 一样常供电，在系统复位或从低功耗模式唤醒后，RTC 的设置和时间维持不变。

6.2.1 主要特性

- 可配置 2 种模式：
 - 定时模式：软件可选择固定周期时间（定时）产生中断通知。
 - 触发模式：匹配一个软件预设的目标闹钟时间，产生中断通知。
- 3 组 16 位计数器，提供了 CK32K 原始周期、2 秒周期、1 天周期的计数。

6.3 寄存器描述

表 6-1 时钟及振荡器控制相关寄存器列表

| 名称 | 访问地址 | 描述 | 复位值 |
|-------------------|------------|-----------------------|------------|
| R16_CLK_SYS_CFG | 0x40001008 | 系统时钟配置寄存器 | 0x0005 |
| R8_HFCK_PWR_CTRL | 0x4000100A | 高频时钟模块电源控制寄存器 | 0x14 |
| R16_INT32K_TUNE | 0x4000102C | 内部 32KHz 时钟校准寄存器 | 0x1011 |
| R8_XT32K_TUNE | 0x4000102E | 外部 32KHz 时钟谐振控制寄存器 | 0xC3 |
| R8_CK32K_CONFIG | 0x4000102F | 32KHz 振荡器配置寄存器 | 0xX2 |
| R8_XT32M_TUNE | 0x4000104E | 外部 32MHz 时钟谐振控制寄存器 | 0x32 |
| R16_OSC_CAL_CNT | 0x40001050 | 振荡器频率校准计数值寄存器 | 0xFFFF |
| R8_OSC_CAL_OV_CNT | 0x40001052 | 振荡器频率校准溢出次数寄存器 | 0x00 |
| R8_OSC_CAL_CTRL | 0x40001053 | 振荡器频率校准控制寄存器 | 0x09 |
| R8_PLL_CONFIG | 0x4000104B | PLL 配置寄存器 | 0x4A |
| R8_RTC_FLAG_CTRL | 0x40001030 | RTC 标志和控制寄存器 | 0x30 |
| R8_RTC_MODE_CTRL | 0x40001031 | RTC 模式配置寄存器 | 0x02 |
| R32_RTC_TRIG | 0x40001034 | RTC 触发数值寄存器 | 0x00000000 |
| R16_RTC_CNT_32K | 0x40001038 | RTC 基于 32768Hz 计数值寄存器 | 0xFFFFFFFF |
| R16_RTC_CNT_2S | 0x4000103A | RTC 以 2S 为单位的计数值寄存器 | 0xFFFFFFFF |
| R32_RTC_CNT_DAY | 0x4000103C | RTC 以天为单位的计数值寄存器 | 0x0000XXXX |

系统时钟配置寄存器 (R16_CLK_SYS_CFG)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|----------------|-----|--|--------|
| [15:8] | Reserved | RO | 保留。 | 00h |
| [7:6] | RB_CLK_SYS_MOD | RWA | HCLK 系统时钟源模式选择： 00/10: CK32M（默认 32MHz）进行分频； 01: PLL（默认 480MHz）进行分频； 11: CK32K（默认 32KHz）作为 HCLK。 | 00b |
| 5 | Reserved | RO | 保留。 | 0 |
| [4:0] | RB_CLK_PLL_DIV | RWA | HCLK 输出时钟分频系数，最小值为 2，0 代表最大值 32，写 1 将关闭 HCLK。 | 00101b |

计算：

$$F_{ck32m} = XT_32MHz;$$

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

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

$$F_{sys} = RB_CLK_SYS_MOD==3?F_{ck32k}:(RB_CLK_SYS_MOD[0]? F_{pll} :F_{ck32m})/RB_CLK_PLL_DIV;$$

上电默认值 $F_{sys} = F_{ck32m} / RB_CLK_PLL_DIV = 32MHz / 5 = 6.4MHz$;

F_{sys} 范围: 32KHz, 2MHz~10MHz, 15MHz~80MHz

高频时钟模块电源控制寄存器 (R8_HFCK_PWR_CTRL)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|-------------------|-----|--|------|
| [7:5] | Reserved | RO | 保留。 | 000b |
| 4 | RB_CLK_PLL_PON | RWA | PLL 电源控制位: 1: 上电; 0: 掉电。 | 1 |
| 3 | RB_CLK_XT32M_KEEP | RWA | 用于暂停模式下控制时钟系统的停止: 1: Halt 模式下不自动停止 HSE 和 PLL; 0: Halt 模式下自动停止 HSE 和 PLL。 | 0 |
| 2 | RB_CLK_XT32M_PON | RWA | 外部 32MHz 振荡器 HSE 电源控制位: 1: 上电; 0: 掉电。 | 1 |
| [1:0] | Reserved | RO | 保留。 | 00b |

内部 32KHz 时钟校准寄存器 (R16_INT32K_TUNE)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|----------------|-----|----------------------|-------|
| [15:13] | Reserved | RO | 保留。 | 000b |
| [12:0] | RB_INT32K_TUNE | RWA | 内部 RC 32KHz 时钟频率校准值。 | 1011h |

外部 32KHz 时钟谐振控制寄存器 (R8_XT32K_TUNE)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|-----------------|-----|--|-------|
| [7:4] | RB_XT32K_C_LOAD | RWA | 选择与外部 32KHz 晶体匹配的内置负载电容 (可能影响 RTC 时钟精度): 电容量 = $RB_XT32K_C_LOAD + 12pF$, 0000b~1111b 分别对应约 12pF~27pF。 根据所用晶体参数选择。 | 1100b |
| [3:2] | Reserved | RO | 保留。 | 00b |
| [1:0] | RB_XT32K_I_TUNE | RWA | 外部 32KHz 振荡器偏置电流选择: 00: 70%额定电流; 01: 额定电流; 10: 140%额定电流; 11: 200%额定电流。 在晶体振荡器稳定后可改为额定电流。 | 11b |

32KHz 振荡器配置寄存器 (R8_CK32K_CONFIG)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|--------------------|-----|---|------|
| 7 | RB_32K_CLK_PIN | RO | 32KHz 时钟引脚状态 (异步信号)。 | X |
| [6:4] | Reserved | RO | 保留。 | 000b |
| 3 | RB_CLK_OSC32K_FILT | RWA | 内部 32KHz 振荡器噪声过滤模式: 1: 禁用; 0: 启用。 | 0 |
| 2 | RB_CLK_OSC32K_XT | RWA | CK32K (32KHz) 时钟源选择位: 1: 外部 32KHz 振荡器; 0: 内部 32KHz 振荡器。 | 0 |
| 1 | RB_CLK_INT32K_PON | RWA | 内部 32KHz 振荡器电源控制位: 1: 上电; 0: 掉电。 | 1 |
| 0 | RB_CLK_XT32K_PON | RWA | 外部 32KHz 振荡器电源控制位: 1: 上电; 0: 掉电。 | 0 |

外部 32MHz 时钟谐振控制寄存器 (R8_XT32M_TUNE)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|-----------------|-----|---|------|
| 7 | Reserved | R0 | 保留。 | 0 |
| [6:4] | RB_XT32M_C_LOAD | RWA | 选择与外部 32MHz 晶体匹配的内置负载电容（可能影响无线通讯）： 电容量 = RB_XT32M_C_LOAD * 2 + 10pF， 000b~111b 分别对应约 10pF~24pF。 根据所用晶体参数选择，常用为 111b。 | 011b |
| [3:2] | Reserved | R0 | 保留。 | 00b |
| [1:0] | RB_XT32M_I_BIAS | RWA | 外部 32MHz 振荡器偏置电流选择： 00: 75%额定电流；01: 额定电流； 10: 125%额定电流；11: 150%额定电流。 | 10b |

振荡器频率校准计数值寄存器 (R16_OSC_CAL_CNT)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|-------------------|-----|---|-------|
| 15 | RB_OSC_CAL_IF | RW1 | 振荡器捕捉完成中断标志位，写 1 清零： 1: 有中断；0: 无中断。 | 0 |
| 14 | RB_OSC_CAL_OV_CLR | RW1 | R8_OSC_CAL_OV_CNT 寄存器数值非零指示，写 1 将 R8_OSC_CAL_OV_CNT 清零。 | 0 |
| [13:0] | RB_OSC_CAL_CNT | R0 | 对多个 CK32K 周期基于系统主频的计数值，用于校准内部 32KHz 振荡器频率。 | XXXXh |

振荡器频率校准溢出次数寄存器 (R8_OSC_CAL_OV_CNT)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|-------------------|----|---|-----|
| [7:0] | RB_OSC_CAL_OV_CNT | R0 | 振荡器频率校准计数溢出次数，向 RB_OSC_CAL_OV_CLR 写 1 清零此寄存器。 | XXh |

振荡器频率校准控制寄存器 (R8_OSC_CAL_CTRL)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|------------------|-----|--|------|
| 7 | Reserved | R0 | 保留。 | 0 |
| 6 | RB_OSC_CNT_END | RWA | 振荡器捕捉终点选择： 1: 追加 2 个周期；0: 不追加。 | 0 |
| 5 | RB_OSC_CNT_EN | RWA | 振荡器频率校准计数器使能位： 1: 使能计数；0: 禁止计数。 | 0 |
| 4 | RB_OSC_CAL_IE | RWA | 振荡器捕捉完成中断使能位： 1: 使能中断；0: 禁止中断。 | 0 |
| 3 | RB_OSC_CNT_HALT | R0 | 振荡器频率校准计数器计数状态位： 1: 正在暂停计数；0: 正在计数中。 | 1 |
| [2:0] | RB_OSC_CNT_TOTAL | RWA | 振荡器捕捉总周期数选择： 000: 1 个；001: 2 个；010: 4 个； 011: 32 个；100: 64 个；101: 128 个； 110: 1024 个；111: 2047 个。 | 001b |

PLL 配置寄存器 (R8_PLL_CONFIG)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|-----------------|-----|------------------|----------|
| 7 | RB_FLASH_IO_MOD | RWA | FlashROM 控制操作模式。 | 0 |
| [6:0] | RB_PLL_CFG_DAT | RWA | PLL 配置参数。 | 1001010b |

RTC 标志和控制寄存器 (R8_RTC_FLAG_CTRL)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|------------------|----|---|-------|
| 7 | RB_RTC_TRIG_FLAG | RO | RTC 触发模式激活标志。 | 0 |
| 6 | RB_RTC_TMR_FLAG | RO | RTC 定时模式激活标志。 | 0 |
| 5 | RB_RTC_TRIG_CLR | RW | 禁用触发模式时，此位固定为 1。 使能触发模式时，写 1 清零触发模式激活标志 RB_RTC_TRIG_FLAG 并自动清 0。 | 1 |
| 4 | RB_RTC_TMR_CLR | RW | 禁用定时模式时，此位固定为 1。 使能定时模式时，写 1 清零定时模式激活标志 RB_RTC_TMR_FLAG 并自动清 0。 | 1 |
| [3:0] | Reserved | RO | 保留。 | 0000b |

RTC 模式配置寄存器 (R8_RTC_MODE_CTRL)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|------------------|-----|--|------|
| 7 | RB_RTC_LOAD_HI | RWA | 写 1 将加载 RTC 计数器高字，加载后自动清 0。将 R32_RTC_TRIG（实际仅低 14 位）加载到 R32_RTC_CNT_DAY。 | 0 |
| 6 | RB_RTC_LOAD_LO | RWA | 写 1 将加载 RTC 计数器低字，加载后自动清 0。将 R32_RTC_TRIG 高 16 位加载到 R16_RTC_CNT_2S；将 R32_RTC_TRIG 低 16 位加载到 R16_RTC_CNT_32K。 | 0 |
| 5 | RB_RTC_TRIG_EN | RWA | RTC 触发模式使能位： 1：使能；0：禁用。 | 0 |
| 4 | RB_RTC_TMR_EN | RWA | RTC 定时模式使能位： 1：使能；0：禁用。 | 0 |
| 3 | RB_RTC_IGNORE_B0 | RWA | 触发模式下忽略比较匹配值的最低位： 1：忽略最低位；0：比较最低位。 | 0 |
| [2:0] | RB_RTC_TMR_MODE | RWA | RTC 定时模式固定周期（定时）选择： 000：0.125S； 001：0.25S； 010：0.5S； 011：1S； 100：2S； 101：4S； 110：8S； 111：16S。 | 010b |

RTC 触发数值寄存器 (R32_RTC_TRIG)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|--------------|-----|--|-------|
| [31:0] | R32_RTC_TRIG | RWA | RTC 触发模式下的预设匹配数值，其高 16 位和低 16 位分别与 R16_RTC_CNT_2S 和 R16_RTC_CNT_32K 进行匹配。 与 RB_RTC_LOAD_LO 及 RB_RTC_LOAD_HI 配合，用于更新 RTC 计数器当前值。 | 0000h |

注：预设匹配数值不是直接写入目标时间，涉及简单计算，请参考后面的说明。

RTC 基于 32768Hz 计数值寄存器 (R16_RTC_CNT_32K)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|-----------------|----|------------------------|-------|
| [15:0] | R16_RTC_CNT_32K | RO | RTC 基于 32768Hz 计数值寄存器。 | XXXXh |

RTC 以 2S 为单位的计数值寄存器 (R16_RTC_CNT_2S)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|----------------|----|---------------------|-------|
| [15:0] | R16_RTC_CNT_2S | RO | RTC 以 2S 为单位的当前计数值。 | XXXXh |

RTC 以天为单位的计数值寄存器 (R32_RTC_CNT_DAY)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|-----------------|----|------------------|----------------|
| [31:16] | Reserved | R0 | 保留。 | 0000h |
| [15:14] | Reserved | R0 | 保留。 | 00b |
| [13:0] | R32_RTC_CNT_DAY | R0 | RTC 以天为单位的当前计数值。 | XXXXXXXXXXXXXb |

6.4 功能描述及配置

6.4.1 RTC 计数器初始化

- (1)、设置 R32_RTC_TRIG 寄存器数值，将 RB_RTC_LOAD_HI 置位，即可将 R32_RTC_TRIG 寄存器的数值加载到 R32_RTC_CNT_DAY 寄存器中；
- (2)、设置 R32_RTC_TRIG 寄存器数值，将 RB_RTC_LOAD_LO 置位，即可将 R32_RTC_TRIG 寄存器的高低各 16 位数值分别加载到 R16_RTC_CNT_2S 寄存器和 R16_RTC_CNT_32K 寄存器。

6.4.2 RTC 时钟源切换为 LSE 晶振

- (1)、确认 X32K1 和 X32K0 所在的 GPIO 引脚没有设置为输出，没有设置上拉和下拉电阻，只有晶体；
- (2)、配置 R8_CK32K_CONFIG 寄存器，置 RB_CLK_XT32K_PON 为 1，开启外部 32KHz 晶体振荡器；
- (3)、建议先置 RB_XT32K_I_TUNE 为最大，等待晶体振荡器稳定（约数百 mS）后再改为额定电流；
- (4)、配置 R8_CK32K_CONFIG 寄存器，置 RB_CLK_OSC32K_XT 为 1，要求时钟源切换到晶体振荡器；
- (5)、等待至少半个 32KHz 时钟周期，通常 16 μ S，真正完成时钟源的切换。

6.4.3 RTC 定时功能

- (1)、配置 R8_RTC_MODE_CTRL 寄存器，设置 RB_RTC_TMR_MODE 选择合适的定时周期，置 RB_RTC_TMR_EN 为 1，打开 RTC 定时功能；
- (2)、达到定时周期后，会产生 RTC 定时激活标志 RB_RTC_TMR_FLAG 及中断，查询 R8_RTC_FLAG_CTRL 寄存器，置位 RB_RTC_TMR_CLR 可清零标志。

6.4.4 RTC 触发功能

- (1)、在 R32_RTC_TRIG 寄存器中设置好目标匹配数值，计算和操作步骤：
以当前时间 R32_RTC_CNT_32K（高 16 位 R16_RTC_CNT_2S 和低 16 位 R16_RTC_CNT_32K）加上间隔时间 DelayTime（单位 S），计算出目标时间数值， $T32 = R32_RTC_CNT_32K + DelayTime * 32768$ ，将 T32 写入 R32_RTC_TRIG 寄存器中，完成匹配数值设定；
- (2)、配置 R8_RTC_MODE_CTRL 寄存器，置 RB_RTC_TRIG_EN 为 1，打开 RTC 触发功能；
- (3)、当 RTC 当前计数值 R16_RTC_CNT_2S 和 R16_RTC_CNT_32K 分别与 R32_RTC_TRIG 预设的高和低 16 位匹配时，产生 RTC 触发激活标志 RB_RTC_TRIG_FLAG 及中断，置位 RB_RTC_TRIG_CLR 可清零标志。
- (4)、如果 RTC 已经过校时，那么可以支持目标绝对时间触发，由目标年/月/日/时/分/秒/毫秒计算出目标时间数值 T32，其它步骤同上。具体可参考评估板例子程序。

6.4.5 用 HSE 校准内部 32K 时钟 LSI

参考评估板例子程序

第 7 章 通用 I/O 和复用功能

7.1 GPIO 简介

芯片提供了 2 组 GPIO 端口 PA 和 PB，共 40 个通用输入输出引脚，其中 32 个引脚具有中断和唤醒功能，部分引脚具有复用及映射功能。

每个 GPIO 端口有一个 32 位方向配置寄存器 R32_Px_DIR，一个 32 位引脚输入寄存器 R32_Px_PIN，一个 32 位数据输出寄存器 R32_Px_OUT，一个 32 位数据复位寄存器 R32_Px_CLR，一个 32 位上拉电阻配置寄存器 R32_Px_PU，一个 32 位下拉电阻/驱动能力配置寄存器 R32_Px_PD_DRV。

PA 端口中 PA[0]~PA[15]位有效，对应芯片上 16 个 GPIO 引脚。

PB 端口中 PB[0]~PB[23]位有效，对应芯片上 24 个 GPIO 引脚。

每个 I/O 端口位可以自由编程，但是 I/O 端口寄存器必须按 8 位、16 位或者 32 位字访问。如果引脚的复用功能没有开启，则默认作为通用 I/O 口使用。

下图是 GPIO 内部结构框图：

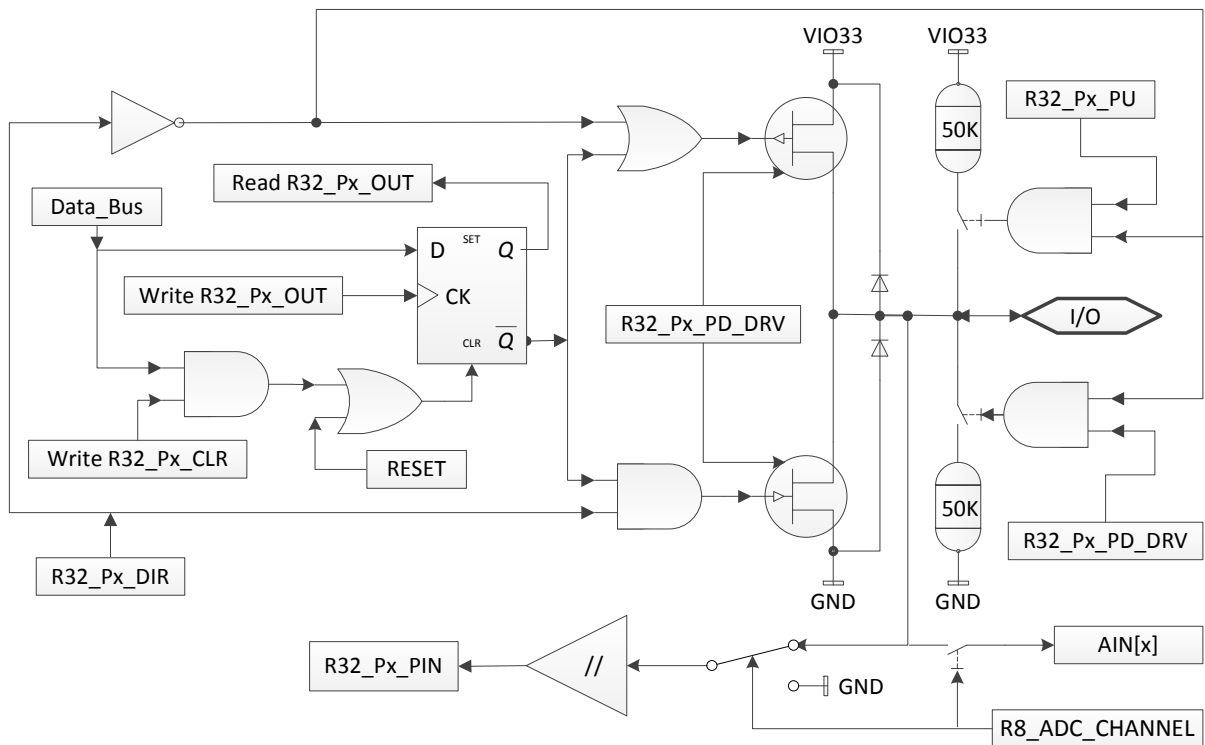


图 7-1 I/O 内部结构框图

7.2 外部中断/唤醒

芯片的部分 I/O 管脚具有中断功能，并可实现睡眠唤醒。

为了使用外部中断，端口位必须配置成输入模式。并提供 4 种触发模式：高电平、低电平、上升沿、下降沿。

唤醒功能需要打开端口位的中断使能 R16_Px_INT_EN，并开启寄存器 R8_SLP_WAKE_CTRL 中的 GPIO 唤醒控制位 RB_SLP_GPIO_WAKE。

7.3 GPIO 的复用与重映射

7.3.1 复用功能

部分 I/O 引脚具有复用功能，上电后默认所有 I/O 引脚均为通用 I/O 功能，启用各功能模块后，相应的原 GPIO 引脚被配置成各自功能模块对应的功能引脚。

如果一个管脚复用多个功能，并且多个功能都已开启，那么复用功能的优先级顺序请参考 1.2 节引脚描述的“复用功能”列表中功能顺序。

例如：PB23 脚复用为 RST#/TMRO_/TXD2_/PWM11，则 RST#复位输入功能优先，PWM11 输出功能的优先级最低。这样可以在多个复用功能中，将功能优先级最低的部分不需使用的引脚启用相对更高优先级的复用功能。

下表列出了部分用于外设模块的功能引脚的 GPIO 配置。

表 7-1 定时器 x

| TMRO/1/2/3 引脚 | 功能配置 | GPIO 配置 |
|---------------|-------------|--------------------|
| TMRx | 输入捕捉通道 x | 输入（浮空输入/上拉输入/下拉输入） |
| | 输出 PWM 通道 x | 推挽输出 |

表 7-2 UARTx

| UART0/1/2/3 引脚 | 功能配置 | GPIO 配置 |
|-------------------|----------------------|---------------|
| TXDx | 串口发送 x | 推挽输出 |
| RXDx | 串口接收 x | 上拉输入（推荐）或浮空输入 |
| RTS, DTR | MODEM 信号输出或 RS485 控制 | 推挽输出 |
| CTS, DSR, RI, DCD | MODEM 信号输入 | 上拉输入（推荐）或浮空输入 |

表 7-3 SPIx

| SPIO/1 引脚 | 功能配置 | GPIO 配置 |
|-----------|-----------|---------------------------------------|
| SCKx | 主模式时钟输出 | 推挽输出 |
| | 从模式时钟输入 | 输入（浮空输入/上拉输入/下拉输入） |
| MOSIx | 全双工模式-主模式 | 推挽输出 |
| | 全双工模式-从模式 | 输入（浮空输入/上拉输入/下拉输入） |
| | 半双工模式-主模式 | 未用到，可做通用 I/O |
| | 半双工模式-从模式 | 未用到，可做通用 I/O |
| MISOx | 全双工模式-主模式 | 输入（浮空输入/上拉输入/下拉输入） |
| | 全双工模式-从模式 | 输入（推荐上拉，片选后自动切为推挽输出）或推挽输出（禁止用于总线连接方式） |
| | 半双工模式-主模式 | 输入或推挽输出，手工切换 |
| | 半双工模式-从模式 | 输入（推荐上拉，片选后自动切为推挽输出） |
| SCS | 主模式片选输出 | 推挽输出（可换用其它引脚） |
| | 从模式片选输入 | 上拉输入（推荐）或浮空输入 |

表 7-4 I2C

| I2C 引脚 | 功能配置 | GPIO 配置 |
|--------|----------------|--------------------|
| SCL | 串行时钟输出-主模式 | 推挽输出（此模式不支持多主机） |
| | 串行时钟输出/输入-多主模式 | 输入（推荐上拉，需要时自动开漏输出） |
| | 串行时钟输入-从模式 | 上拉输入（推荐）或浮空输入 |
| SDA | 串行数据输入输出 | 输入（推荐上拉，需要时自动开漏输出） |

表 7-5 ADC

| ADC 采样通道引脚 | 功能配置 | GPIO 配置 |
|------------|------------|---------|
| AINx | 模拟数字转换输入通道 | 浮空输入 |

表 7-6 USBx

| USB 信号引脚 | 功能配置 | GPIO 配置 |
|-----------|---------------|---------|
| UD-, U2D- | 连接到内部 USB 收发器 | 浮空输入 |
| UD+, U2D+ | 连接到内部 USB 收发器 | 浮空输入 |

7.3.2 功能引脚重映射

为了使外设功能的同时利用率达到最优，可以通过设置 R16_PIN_ALTERNATE 功能引脚重映射寄存器把一些功能引脚重新映射到其他引脚上。

表 7-7 复用功能重映射引脚

| 外设功能引脚 | 默认所在的 GPIO 引脚 | 重映射到的 GPIO 引脚 |
|--------------------|-----------------------------|-----------------------------|
| SCS/SCK0/MOS1/MISO | 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[12]/PB[13] |
| RXD0/TXD0/DSR/DTR | PB[4]/PB[7]/PB[1]/PB[5] | PA[15]/PA[14]/PB[14]/PB[15] |
| TMR3/PWM3/CAP3 | PB[22] | PA[2] |
| TMR2/PWM2/CAP2 | PA[11] | PB[11] |
| TMR1/PWM1/CAP1 | PA[10] | PB[10] |
| TMR0/PWM0/CAPO | PA[9] | PB[23] |
| PWM4/PWM5 | PA[12]/PA[13] | PA[6]/PA[7] |
| PWM7/PWM8/PWM9 | PB[4]/PB[6]/PB[7] | PB[1]/PB[2]/PB[3] |
| SCL/SDA | PB[13]/PB[12] | PB[21]/PB[20] |

7.4 寄存器描述

表 7-8 GPIO 相关寄存器列表

| 名称 | 访问地址 | 描述 | 复位值 |
|-------------------|------------|-----------------|------------|
| R16_PIN_ALTERNATE | 0x40001018 | 功能引脚重映射寄存器 | 0x0000 |
| R16_PIN_ANALOG_IE | 0x4000101A | 外设模拟管脚配置寄存器 | 0x0000 |
| R16_PA_INT_EN | 0x40001090 | PA 端口中断使能寄存器 | 0x0000 |
| R16_PB_INT_EN | 0x40001092 | PB 端口中断使能寄存器 | 0x0000 |
| R16_PA_INT_MODE | 0x40001094 | PA 端口中断模式配置寄存器 | 0x0000 |
| R16_PB_INT_MODE | 0x40001096 | PB 端口中断模式配置寄存器 | 0x0000 |
| R16_PA_INT_IF | 0x4000109C | PA 端口中断标志寄存器 | 0x0000 |
| R16_PB_INT_IF | 0x4000109E | PB 端口中断标志寄存器 | 0x0000 |
| R32_PA_DIR | 0x400010A0 | PA 端口方向配置寄存器 | 0x00000000 |
| R32_PA_PIN | 0x400010A4 | PA 端口引脚输入寄存器 | 0x0000XXXX |
| R32_PA_OUT | 0x400010A8 | PA 端口数据输出寄存器 | 0x00000000 |
| R32_PA_CLR | 0x400010AC | PA 端口数据复位寄存器 | 0x00000000 |
| R32_PA_PU | 0x400010B0 | PA 端口上拉电阻配置寄存器 | 0x00000000 |
| R32_PA_PD_DRV | 0x400010B4 | PA 端口下拉/驱动配置寄存器 | 0x00000000 |
| R32_PB_DIR | 0x400010C0 | PB 端口方向配置寄存器 | 0x00000000 |
| R32_PB_PIN | 0x400010C4 | PB 端口引脚输入寄存器 | 0x00XXXXXX |
| R32_PB_OUT | 0x400010C8 | PB 端口数据输出寄存器 | 0x00000000 |
| R32_PB_CLR | 0x400010CC | PB 端口数据复位寄存器 | 0x00000000 |
| R32_PB_PU | 0x400010D0 | PB 端口上拉电阻配置寄存器 | 0x00000000 |
| R32_PB_PD_DRV | 0x400010D4 | PB 端口下拉/驱动配置寄存器 | 0x00000000 |

功能引脚重映射寄存器 (R16_PIN_ALTERNATE)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|----|-----------------|----|--|-----|
| 15 | RB_RF_ANT_SW_EN | RW | RF 天线开关控制输出使能: 1: 开关控制输出到 PB[16]~PB[21]; 0: 禁止输出。 | 0 |
| 14 | RB_PIN_U0_INV | RW | UART0 输入输出反相使能: 1: RXD0/RXD0_反相输入, TXD0/TXD0_反相输出; 0: 正常同相输入、输出。 | 0 |
| 13 | RB_PIN_INTX | RW | INT24/INT25 功能引脚映射选择位: 1: INT24_/25_映射到 PB[22]/PB[23]; 0: INT24/25 映射到 PB[8]/PB[9]。 注: INT24/INT25 是 R16_PB_INT_EN、R16_PB_INT_MODE、R16_PB_INT_IF 中 [9:8] 对应的中断输入。 | 0 |
| 12 | RB_PIN_MODEM | RW | UART0 MODEM 功能引脚映射选择位: 1: DSR_/DTR_映射到 PB[14]/PB[15]; 0: DSR/DTR 映射到 PB[1]/PB[5]。 | 0 |
| 11 | RB_PIN_I2C | RW | I2C 功能引脚映射选择位: 1: SCL_/SDA_映射到 PB[21]/PB[20]; 0: SCL/SDA 映射到 PB[13]/PB[12]。 | 0 |
| 10 | RB_PIN_PWMX | RW | PWMx 功能引脚映射选择位: 1: PWM4/5/7/8/9 映射到 PA[6]/PA[7]/PB[1]/PB[2]/PB[3]; 0: PWM4/5/7/8/9 映射到 PA[12]/PA[13]/PB[4]/PB[6]/PB[7]。 | 0 |
| 9 | Reserved | RO | 保留。 | 0 |
| 8 | RB_PIN_SPI0 | RW | SPI0 功能引脚映射选择位: 1: SCK0_/SCS_/MOSI_/MISO_映射到 PB[12]/PB[13]/PB[14]/PB[15]; 0: SCK0/SCS/MOSI/MISO 映射到 PA[12]/PA[13]/PA[14]/PA[15]。 | 0 |
| 7 | RB_PIN_UART3 | RW | UART3 功能引脚映射选择位: 1: RXD3_/TXD3_映射到 PB[20]/PB[21]; 0: RXD3/TXD3 映射到 PA[4]/PA[5]。 | 0 |
| 6 | RB_PIN_UART2 | RW | UART2 功能引脚映射选择位: 1: RXD2_/TXD2_映射到 PB[22]/PB[23]; 0: RXD2/TXD2 映射到 PA[6]/PA[7]。 | 0 |
| 5 | RB_PIN_UART1 | RW | UART1 功能引脚映射选择位: 1: RXD1_/TXD1_映射到 PB[12]/PB[13]; 0: RXD1/TXD1 映射到 PA[8]/PA[9]。 | 0 |
| 4 | RB_PIN_UART0 | RW | UART0 功能引脚映射选择位: 1: RXD0_/TXD0_映射到 PA[15]/PA[14]; 0: RXD0/TXD0 映射到 PB[4]/PB[7]。 | 0 |
| 3 | RB_PIN_TMR3 | RW | TMR3 功能引脚映射选择位: 1: TMR3_/PWM3_/CAP3_映射到 PA[2]; 0: TMR3/PWM3/CAP3 映射到 PB[22]。 | 0 |
| 2 | RB_PIN_TMR2 | RW | TMR2 功能引脚映射选择位: | 0 |

| | | | | |
|---|-------------|----|---|---|
| | | | 1: TMR2_/PWM2_/CAP2_映射到 PB[11]; 0: TMR2/PWM2/CAP2 映射到 PA[11]。 | |
| 1 | RB_PIN_TMR1 | RW | TMR1 功能引脚映射选择位: 1: TMR1_/PWM1_/CAP1_映射到 PB[10]; 0: TMR1/PWM1/CAP1 映射到 PA[10]。 | 0 |
| 0 | RB_PIN_TMRO | RW | TMRO 功能引脚映射选择位: 1: TMRO_/PWMO_/CAPO_映射到 PB[23]; 0: TMRO/PWMO/CAPO 映射到 PA[9]。 | 0 |

外设模拟管脚配置寄存器(R16_PIN_ANALOG_IE)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|----|-------------------|----|--|-----|
| 15 | RB_PIN_ADC4_5_IE | RW | ADC/TKEY 4/5 通道引脚数字输入禁用: 1: 关闭 PA14-15 数字输入, 可节约功耗; 0: 打开数字输入。 | 0 |
| 14 | RB_PIN_ADC2_3_IE | RW | ADC/TKEY 2/3 通道引脚数字输入禁用: 1: 关闭 PA12-13 数字输入, 可节约功耗; 0: 打开数字输入。 | 0 |
| 13 | RB_PIN_XT32K_IE | RW | 32KHz 晶振 LSE 引脚数字输入禁用: 1: 关闭 PA10-11 数字输入, 可节约功耗; 0: 打开数字输入。 | 0 |
| 12 | RB_PIN_ADC13_IE | RW | ADC/TKEY 13 通道引脚数字输入禁用: 1: 关闭 PA9 数字输入, 可节约功耗; 0: 打开数字输入。 | 0 |
| 11 | RB_PIN_ADC12_IE | RW | ADC/TKEY 12 通道引脚数字输入禁用: 1: 关闭 PA8 数字输入, 可节约功耗; 0: 打开数字输入。 | 0 |
| 10 | RB_PIN_ADC1_IE | RW | ADC/TKEY 1 通道引脚数字输入禁用: 1: 关闭 PA5 数字输入, 可节约功耗; 0: 打开数字输入。 | 0 |
| 9 | RB_PIN_ADC0_IE | RW | ADC/TKEY 0 通道引脚数字输入禁用: 1: 关闭 PA4 数字输入, 可节约功耗; 0: 打开数字输入。 | 0 |
| 8 | Reserved | RW | 保留。 | 0 |
| 7 | RB_PIN_USB_IE | RW | USB 引脚使能: 1: PB10-11 为 USB 通讯引脚; 0: PB10-11 不用于 USB 通讯。 | 0 |
| 6 | RB_PIN_USB_DP_PU | RW | USB UD+引脚内部上拉电阻使能: 1: 强制使能上拉 (RB_UC_DEV_PU_EN 在睡眠或下电模式下不起作用, 由此代替); 0: 由 RB_UC_DEV_PU_EN 控制是否上拉。 | 0 |
| 5 | RB_PIN_USB2_IE | RW | USB2 引脚使能: 1: PB12-13 为 USB2 通讯引脚; 0: PB12-13 不用于 USB2 通讯。 | 0 |
| 4 | RB_PIN_USB2_DP_PU | RW | USB2 U2D+引脚内部上拉电阻使能: 1: 强制使能上拉 (RB_UC_DEV_PU_EN 在睡眠或下电模式下不起作用, 由此代替); 0: 由 RB_UC_DEV_PU_EN 控制是否上拉。 | 0 |
| 3 | RB_PIN_ADC11_IE | RW | ADC/TKEY 11 通道引脚数字输入禁用: 1: 关闭 PA7 数字输入, 可节约功耗; | 0 |

| | | | | |
|---|------------------|----|--|---|
| | | | 0: 打开数字输入。 | |
| 2 | RB_PIN_ADC10_IE | RW | ADC/TKEY 10 通道引脚数字输入禁用: 1: 关闭 PA6 数字输入, 可节约功耗; 0: 打开数字输入。 | 0 |
| 1 | RB_PIN_ADC6_7_IE | RW | ADC/TKEY 7/6 通道引脚数字输入禁用: 1: 关闭 PA2/PA3 数字输入, 可节约功耗; 0: 打开数字输入。 | 0 |
| 0 | RB_PIN_ADC8_9_IE | RW | ADC/TKEY 9/8 通道引脚数字输入禁用: 1: 关闭 PA0/PA1 数字输入, 可节约功耗; 0: 打开数字输入。 | 0 |

注: 如果引脚用于模拟功能 (ADC/TouchKey), 建议将该引脚的数字输入功能关闭, 即设置数字输入禁用, 从而可以降低功耗, 并有利于减少干扰。

PA 端口中断使能寄存器 (R16_PA_INT_EN)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|---------------|----|---|-------|
| [15:0] | R16_PA_INT_EN | RW | PA 引脚中断使能位: 1: 使能相应中断; 0: 禁止相应中断。 | 0000h |

PB 端口中断使能寄存器 (R16_PB_INT_EN)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|---------------|----|---|-------|
| [15:0] | R16_PB_INT_EN | RW | PB 引脚中断使能位: 1: 使能相应中断; 0: 禁止相应中断。 | 0000h |

注: R16_PB_INT_EN[9:8] 由 RB_PIN_INTX 选择对应 PB[23:22] 或 PB[9:8]。

PA 端口中断模式配置寄存器 (R16_PA_INT_MODE)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|-----------------|----|------------------------------------|-------|
| [15:0] | R16_PA_INT_MODE | RW | PA 引脚中断模式选择位: 1: 边沿触发; 0: 电平触发。 | 0000h |

PB 端口中断模式配置寄存器 (R16_PB_INT_MODE)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|-----------------|----|------------------------------------|-------|
| [15:0] | R16_PB_INT_MODE | RW | PB 引脚中断模式选择位: 1: 边沿触发; 0: 电平触发。 | 0000h |

注: R16_PB_INT_MODE[9:8] 由 RB_PIN_INTX 选择对应 PB[23:22] 或 PB[9:8]。

PA 端口中断标志寄存器 (R16_PA_INT_IF)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|---------------|-----|--|-------|
| [15:0] | R16_PA_INT_IF | RW1 | PA 引脚中断标志位, 写 1 清零: 1: 有中断; 0: 无中断。 | 0000h |

PB 端口中断标志寄存器 (R16_PB_INT_IF)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|---------------|-----|--|-------|
| [15:0] | R16_PB_INT_IF | RW1 | PB 引脚中断标志位, 写 1 清零: 1: 有中断; 0: 无中断。 | 0000h |

注: R16_PB_INT_IF[9:8] 由 RB_PIN_INTX 选择对应 PB[23:22] 或 PB[9:8]。

PA 端口方向配置寄存器 (R32_PA_DIR)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|-------------|----|--|-------|
| [31:16] | Reserved | R0 | 保留。 | 0000h |
| [15:8] | R8_PA_DIR_1 | RW | PA 引脚当前输入输出方向配置： 1: 引脚为输出模式； 0: 引脚为输入模式。 | 00h |
| [7:0] | R8_PA_DIR_0 | RW | | 00h |

PA 端口引脚输入寄存器 (R32_PA_PIN)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|-------------|----|--|-------|
| [31:16] | Reserved | R0 | 保留。 | 0000h |
| [15:8] | R8_PA_PIN_1 | R0 | PA 引脚当前电平状态（仅在 R32_PA_DIR 对应位为 0 时，该位值有效）： 1: 引脚输入高电平； 0: 引脚输入低电平。 | XXh |
| [7:0] | R8_PA_PIN_0 | R0 | | XXh |

PA 端口数据输出寄存器 (R32_PA_OUT)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|-------------|----|--|-------|
| [31:16] | Reserved | R0 | 保留。 | 0000h |
| [15:8] | R8_PA_OUT_1 | RW | 方向寄存器 R32_PA_DIR 对应位为 1 时： 控制 PA 引脚输出电平状态： 1: 输出高电平；0: 输出低电平。 方向寄存器 R32_PA_DIR 对应位为 0 时： 控制 PA 引脚中断极性选择： 1: 高电平/上升沿；0: 低电平/下降沿。 | 00h |
| [7:0] | R8_PA_OUT_0 | RW | | 00h |

PA 端口数据复位寄存器 (R32_PA_CLR)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|-------------|----|---|-------|
| [31:16] | Reserved | R0 | 保留。 | 0000h |
| [15:8] | R8_PA_CLR_1 | WZ | PA 数据寄存器复位控制： 1: R32_PA_OUT 对应位数据清 0； 0: 无影响。 | 00h |
| [7:0] | R8_PA_CLR_0 | WZ | | 00h |

PA 端口上拉电阻配置寄存器 (R32_PA_PU)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|------------|----|--|-------|
| [31:16] | Reserved | R0 | 保留。 | 0000h |
| [15:8] | R8_PA_PU_1 | RW | PA 引脚上拉电阻使能控制： 1: 启用上拉电阻；0: 关闭上拉电阻。 | 00h |
| [7:0] | R8_PA_PU_0 | RW | | 00h |

PA 端口下拉/驱动配置寄存器 (R32_PA_PD_DRV)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|----------------|----|---|-------|
| [31:16] | Reserved | R0 | 保留。 | 0000h |
| [15:8] | R8_PA_PD_DRV_1 | RW | 方向寄存器 R32_PA_DIR 对应位为 0 时： PA 引脚下拉电阻使能控制： 1: 启用下拉电阻；0: 关闭下拉电阻。 方向寄存器 R32_PA_DIR 对应位为 1 时： PA 引脚电流驱动能力选择： 1: 20mA 级别；0: 5mA 级别。 | 00h |
| [7:0] | R8_PA_PD_DRV_0 | RW | | 00h |

PB 端口方向配置寄存器 (R32_PB_DIR)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|-------------|----|--|-----|
| [31:24] | Reserved | R0 | 保留。 | 00h |
| [23:16] | R8_PB_DIR_2 | RW | PB 引脚当前输入输出方向配置： 1: 引脚为输出模式； 0: 引脚为输入模式。 | 00h |
| [15:8] | R8_PB_DIR_1 | RW | | 00h |
| [7:0] | R8_PB_DIR_0 | RW | | 00h |

PB 端口引脚输入寄存器 (R32_PB_PIN)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|-------------|----|--|-----|
| [31:24] | Reserved | R0 | 保留。 | 00h |
| [23:16] | R8_PB_PIN_2 | R0 | PB 引脚当前电平状态（仅在 R32_PB_DIR 对应位为 0 时，该位值有效）： 1: 引脚输入高电平； 0: 引脚输入低电平。 | XXh |
| [15:8] | R8_PB_PIN_1 | R0 | | XXh |
| [7:0] | R8_PB_PIN_0 | R0 | | XXh |

PB 端口数据输出寄存器 (R32_PB_OUT)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|-------------|----|---|-----|
| [31:24] | Reserved | R0 | 保留。 | 00h |
| [23:16] | R8_PB_OUT_2 | RW | 方向寄存器 R32_PB_DIR 对应位为 1 时： 控制 PB 引脚输出电平状态： 1: 输出高电平；0: 输出低电平。 | 00h |
| [15:8] | R8_PB_OUT_1 | RW | 方向寄存器 R32_PB_DIR 对应位为 1 时： 控制 PB 引脚输出电平状态： 1: 输出高电平；0: 输出低电平。 | 00h |
| [7:0] | R8_PB_OUT_0 | RW | 方向寄存器 R32_PB_DIR 对应位为 0 时： 控制 PB 引脚中断极性选择： 1: 高电平/上升沿；0: 低电平/下降沿。 | 00h |

PB 端口数据复位寄存器 (R32_PB_CLR)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|-------------|----|---|-----|
| [31:24] | Reserved | R0 | 保留。 | 00h |
| [23:16] | R8_PB_CLR_2 | WZ | PB 数据寄存器复位控制： 1: R32_PB_OUT 对应位数据清 0； 0: 无影响。 | 00h |
| [15:8] | R8_PB_CLR_1 | WZ | | 00h |
| [7:0] | R8_PB_CLR_0 | WZ | | 00h |

PB 端口上拉电阻配置寄存器 (R32_PB_PU)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|------------|----|--|-----|
| [31:24] | Reserved | R0 | 保留。 | 00h |
| [23:16] | R8_PB_PU_2 | RW | PB 引脚上拉电阻使能控制： 1: 启用上拉电阻；0: 关闭上拉电阻。 | 00h |
| [15:8] | R8_PB_PU_1 | RW | | 00h |
| [7:0] | R8_PB_PU_0 | RW | | 00h |

PB 端口下拉/驱动配置寄存器 (R32_PB_PD_DRV)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|----------------|----|----------------------------|-----|
| [31:24] | Reserved | R0 | 保留。 | 00h |
| [23:16] | R8_PB_PD_DRV_2 | RW | 方向寄存器 R32_PB_DIR 对应位为 0 时： | 00h |

| | | | | |
|--------|----------------|----|---|-----|
| [15:8] | R8_PB_PD_DRV_1 | RW | PB 引脚下拉电阻使能控制： 1：启用下拉电阻；0：关闭下拉电阻。 方向寄存器 R32_PB_DIR 对应位为 1 时： PB 引脚电流驱动能力选择： 1：20mA 级别；0：5mA 级别。 | 00h |
| [7:0] | R8_PB_PD_DRV_0 | RW | | 00h |

7.5 GPIO 引脚模式配置

每个 GPIO 都可以配置成 5 种模式，具体见下表：

表 7-9 端口配置表

| 模式 | R32_Px_DIR | R32_Px_PU | R32_Px_PD_DRV |
|-------------------|------------|-----------|---------------|
| 浮空输入/高阻输入/模拟输入 | 0 | 0 | 0 |
| 带上拉电阻的输入 | 0 | 1 | 0 |
| 带下拉电阻的输入 | 0 | 0 | 1 |
| 推挽输出，驱动能力 5mA 级别 | 1 | X | 0 |
| 推挽输出，驱动能力 20mA 级别 | 1 | X | 1 |

第 8 章 通用定时器 TMRx

8.1 TMRx 简介

芯片提供了 4 个 26 位定时器，TMR0、TMR1、TMR2 和 TMR3，最长定时时间为 2^{26} 个时钟周期。它适用于多种场合，包括测量输入信号脉冲长度（输入捕捉）或者产生输出波形（PWM），另外 TMR1 和 TMR2 支持 DMA 功能。每个定时器都是完全独立的，可以一起同步操作。

8.1.1 主要特性

- 4 个 26 位定时器，每个定时器定时时间最大为 2^{26} 个时钟周期。
- 支持定时器中断，其中 TMR1 和 TMR2 支持 DMA 及中断。
- 支持捕捉功能，测量输入脉冲长度或周期。
- 捕捉功能可设置为电平变化捕捉和高或低电平保持时间捕捉功能。
- 支持 26 位 PWM 功能，可动态调整 PWM 占空比设置。

8.2 寄存器描述

表 8-1 TMR0 相关寄存器列表

| 名称 | 访问地址 | 描述 | 复位值 |
|--------------------|------------|------------|------------|
| R8_TMR0_CTRL_MOD | 0x40002000 | 模式设置寄存器 | 0x02 |
| R8_TMR0_INTER_EN | 0x40002002 | 中断使能寄存器 | 0x00 |
| R8_TMR0_INT_FLAG | 0x40002006 | 中断标志寄存器 | 0x00 |
| R8_TMR0_FIFO_COUNT | 0x40002007 | FIFO 计数寄存器 | 0x0X |
| R32_TMR0_COUNT | 0x40002008 | 当前计数值寄存器 | 0x0XXXXXXX |
| R32_TMR0_CNT_END | 0x4000200C | 计数终值设置寄存器 | 0x0XXXXXXX |
| R32_TMR0_FIFO | 0x40002010 | FIFO 寄存器 | 0x0XXXXXXX |

表 8-2 TMR1 相关寄存器列表

| 名称 | 访问地址 | 描述 | 复位值 |
|--------------------|------------|-------------|------------|
| R8_TMR1_CTRL_MOD | 0x40002400 | 模式设置寄存器 | 0x02 |
| R8_TMR1_CTRL_DMA | 0x40002401 | DMA 控制寄存器 | 0x00 |
| R8_TMR1_INTER_EN | 0x40002402 | 中断使能寄存器 | 0x00 |
| R8_TMR1_INT_FLAG | 0x40002406 | 中断标志寄存器 | 0x00 |
| R8_TMR1_FIFO_COUNT | 0x40002407 | FIFO 计数寄存器 | 0x0X |
| R32_TMR1_COUNT | 0x40002408 | 当前计数值寄存器 | 0x0XXXXXXX |
| R32_TMR1_CNT_END | 0x4000240C | 计数终值寄存器 | 0x0XXXXXXX |
| R32_TMR1_FIFO | 0x40002410 | FIFO 寄存器 | 0x0XXXXXXX |
| R16_TMR1_DMA_NOW | 0x40002414 | DMA 当前缓冲区地址 | 0x0000XXXX |
| R16_TMR1_DMA_BEG | 0x40002418 | DMA 起始缓冲区地址 | 0x0000XXXX |
| R16_TMR1_DMA_END | 0x4000241C | DMA 结束缓冲区地址 | 0x0000XXXX |

表 8-3 TMR2 相关寄存器列表

| 名称 | 访问地址 | 描述 | 复位值 |
|------------------|------------|-----------|------|
| R8_TMR2_CTRL_MOD | 0x40002800 | 模式设置寄存器 | 0x02 |
| R8_TMR2_CTRL_DMA | 0x40002801 | DMA 控制寄存器 | 0x00 |
| R8_TMR2_INTER_EN | 0x40002802 | 中断使能寄存器 | 0x00 |

| | | | |
|--------------------|------------|-------------|------------|
| R8_TMR2_INT_FLAG | 0x40002806 | 中断标志寄存器 | 0x00 |
| R8_TMR2_FIFO_COUNT | 0x40002807 | FIFO 计数寄存器 | 0x0X |
| R32_TMR2_COUNT | 0x40002808 | 当前计数值寄存器 | 0x0XXXXXXX |
| R32_TMR2_CNT_END | 0x4000280C | 计数终值寄存器 | 0x0XXXXXXX |
| R32_TMR2_FIFO | 0x40002810 | FIFO 寄存器 | 0x0XXXXXXX |
| R16_TMR2_DMA_NOW | 0x40002814 | DMA 当前缓冲区地址 | 0x0000XXXX |
| R16_TMR2_DMA_BEG | 0x40002818 | DMA 起始缓冲区地址 | 0x0000XXXX |
| R16_TMR2_DMA_END | 0x4000281C | DMA 结束缓冲区地址 | 0x0000XXXX |

表 8-4 TMR3 相关寄存器列表

| 名称 | 访问地址 | 描述 | 复位值 |
|--------------------|------------|------------|------------|
| R8_TMR3_CTRL_MOD | 0x40002C00 | 模式设置寄存器 | 0x02 |
| R8_TMR3_INTER_EN | 0x40002C02 | 中断使能寄存器 | 0x00 |
| R8_TMR3_INT_FLAG | 0x40002C06 | 中断标志寄存器 | 0x00 |
| R8_TMR3_FIFO_COUNT | 0x40002C07 | FIFO 计数寄存器 | 0x0X |
| R32_TMR3_COUNT | 0x40002C08 | 当前计数值寄存器 | 0x0XXXXXXX |
| R32_TMR3_CNT_END | 0x40002C0C | 计数终值设置寄存器 | 0x0XXXXXXX |
| R32_TMR3_FIFO | 0x40002C10 | FIFO 寄存器 | 0x0XXXXXXX |

模式设置寄存器 (R8_TMRx_CTRL_MOD) (x=0/1/2/3)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|-------------------|----|---|-----|
| [7:6] | RB_TMR_CAP_EDGE | RW | 捕捉模式下, 选择捕捉触发方式: 00: 不触发; 01: 捕捉任何边沿变化之间的时间; 10: 捕捉下降沿到下降沿之间时间; 11: 捕捉上升沿到上升沿之间时间。 计数模式下, 选择计数的边沿: 00: 不采样计数; 01: 采样到任意边沿计数; 10: 采样到下降沿计数; 11: 采样到上升沿计数。 | 00b |
| [7:6] | RB_TMR_PWM_REPEAT | RW | PWM 模式下, 选择数据重复方式: 00: 重复 1 次; 01: 重复 4 次; 10: 重复 8 次; 11: 重复 16 次。 | 00b |
| 5 | Reserved | RO | 保留。 | 0 |
| 4 | RB_TMR_CAP_COUNT | RW | RB_TMR_MODE_IN=1 输入模式的子模式: 1: 计数模式; 0: 捕捉模式。 | 0 |
| 4 | RB_TMR_OUT_POLAR | RW | PWM 模式下, 输出极性设置位: 1: 默认高电平, 低电平有效; 0: 默认低电平, 高电平有效。 | 0 |
| 3 | RB_TMR_OUT_EN | RW | 定时器输出使能位: 1: 输出使能; 0: 输出禁止。 | 0 |
| 2 | RB_TMR_COUNT_EN | RW | 定时器计数使能位: 1: 使能计数; 0: 停止计数。 | 0 |
| 1 | RB_TMR_ALL_CLEAR | RW | 定时器的 FIFO/计数器/中断标志清零: 1: 强制清空和清零; 0: 不清。 | 1 |
| 0 | RB_TMR_MODE_IN | RW | 定时器模式设置位: 1: 输入模式 (捕捉模式或计数模式); | 0 |

| | | | | |
|--|--|--|------------------|--|
| | | | 0: 定时模式或 PWM 模式。 | |
|--|--|--|------------------|--|

中断使能寄存器 (R8_TMRx_INTER_EN) (x=0/1/2/3)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|--------------------|----|---|------|
| [7:5] | Reserved | RO | 保留。 | 000b |
| 4 | RB_TMR_IE_FIFO_OV | RW | FIFO 溢出 (捕捉模式 FIFO 满或 PWM 模式 FIFO 空) 中断使能位: 1: 使能中断; 0: 禁止中断。 | 0 |
| 3 | RB_TMR_IE_DMA_END | RW | DMA 结束中断使能位 (仅 TMR1/2 支持): 1: 使能中断; 0: 禁止中断。 | 0 |
| 2 | RB_TMR_IE_FIFO_HF | RW | FIFO 使用过半 (捕捉模式 FIFO ≥ 4 或 PWM 模式 FIFO < 4) 中断使能位: 1: 使能中断; 0: 禁止中断。 | 0 |
| 1 | RB_TMR_IE_DATA_ACT | RW | 数据激活 (捕捉模式指每次捕捉到新数据, PWM 模式指数值触发导致有效电平结束) 中断使能位: 1: 使能中断; 0: 禁止中断。 | 0 |
| 0 | RB_TMR_IE_CYC_END | RW | 周期结束 (捕捉模式指超时, PWM 模式和定时模式指周期结束) 中断使能位: 1: 使能中断; 0: 禁止中断。 | 0 |

中断标志寄存器 (R8_TMRx_INT_FLAG) (x=0/1/2/3)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|--------------------|-----|--|------|
| [7:5] | Reserved | RO | 保留。 | 000b |
| 4 | RB_TMR_IF_FIFO_OV | RW1 | FIFO 溢出 (捕捉模式 FIFO 满或 PWM 模式 FIFO 空) 标志位, 写 1 清零: 1: 已溢出; 0: 未溢出。 | 0 |
| 3 | RB_TMR_IF_DMA_END | RW1 | DMA 完成标志位, 写 1 清零: 1: 已完成; 0: 未完成。 | 0 |
| 2 | RB_TMR_IF_FIFO_HF | RW1 | FIFO 使用过半 (捕捉模式 FIFO ≥ 4 或 PWM 模式 FIFO < 4) 标志位, 写 1 清零: 1: FIFO 使用已过半; 0: FIFO 使用未过半。 | 0 |
| 1 | RB_TMR_IF_DATA_ACT | RW1 | 数据激活 (捕捉模式指每次捕捉到新数据, PWM 模式指数值触发导致有效电平结束) 标志位, 写 1 清零: 1: 产生/用了数据; 0: 未产生/未用。 | 0 |
| 0 | RB_TMR_IF_CYC_END | RW1 | 周期结束 (捕捉模式指超时, PWM 模式和定时模式指周期结束, 计数模式指计数溢出) 标志位, 写 1 清零: 1: 超时/周期结束; 0: 未超时/未结束。 | 0 |

FIFO 计数寄存器 (R8_TMRx_FIFO_COUNT) (x=0/1/2/3)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|--------------------|----|--------------------|------|
| [7:0] | R8_TMRx_FIFO_COUNT | RO | FIFO 内数据计数, 最大值 8。 | 0x0X |

当前计数值寄存器 (R32_TMRx_COUNT) (x=0/1/2/3)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---|----|----|----|-----|
|---|----|----|----|-----|

| | | | | |
|--------|----------------|----|-----------|----------|
| [31:0] | R32_TMRx_COUNT | RO | 计数器当前计数值。 | 0XXXXXXh |
|--------|----------------|----|-----------|----------|

计数终值设置寄存器 (R32_TMRx_CNT_END) (x=0/1/2/3)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|------------------|----|--|----------|
| [31:0] | R32_TMRx_CNT_END | RW | 定时器模式下，一个定时周期时钟数； PWM 模式下，PWM 单周期总时钟数； 捕捉模式下，捕捉超时时钟数。 仅低 26 位有效，最大值 67108863。 计数模式下，计数值终值-2（溢出）。 注：对此寄存器的写操作，将自动清零 R32_TMRx_COUNT 寄存器中的值。 | 0XXXXXXh |

FIFO 寄存器 (R32_TMRx_FIFO) (x=0/1/2/3)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|---------------|-----------|-----------------------|----------|
| [31:0] | R32_TMRx_FIFO | RO/ WO | FIFO 数据寄存器，仅低 26 位有效。 | 0XXXXXXh |

DMA 控制寄存器 (R8_TMRx_CTRL_DMA) (x=1/2) (仅 TMR1/2 支持)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|-------------------|----|---|--------|
| [7:3] | Reserved | RO | 保留。 | 00000b |
| 2 | RB_TMR_DMA_LOOP | RW | DMA 地址循环功能使能位： 1：使能地址循环；0：禁止地址循环。 如果使能 DMA 地址循环，当 DMA 地址增加到设置的末尾地址时，自动循环指向设置的首地址。 | 0 |
| 1 | Reserved | RO | 保留。 | 0 |
| 0 | RB_TMR_DMA_ENABLE | RW | DMA 功能使能位： 1：使能 DMA；0：禁止 DMA。 | 0 |

DMA 当前缓冲区地址 (R16_TMRx_DMA_NOW) (x=1/2)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|------------------|----|--|-------|
| [15:0] | R16_TMRx_DMA_NOW | RO | DMA 数据缓冲区当前地址。 可用于计算已转换次数，计算方法： COUNT=(TMR_DMA_NOW-TMR_DMA_BEG)/4。 | XXXXh |

DMA 起始缓冲区地址 (R16_TMRx_DMA_BEG) (x=1/2)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|------------------|----|--------------------------------------|-------|
| [15:0] | R16_TMRx_DMA_BEG | RW | DMA 数据缓冲区起始地址，仅低 15 位有效，地址必须 4 字节对齐。 | XXXXh |

DMA 结束缓冲区地址 (R16_TMRx_DMA_END) (x=1/2)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|------------------|----|--|-------|
| [15:0] | R16_TMRx_DMA_END | RW | DMA 数据缓冲区结束地址（不含），仅低 15 位有效，地址必须 4 字节对齐。 | XXXXh |

8.3 功能描述及配置

8.3.1 定时、计数功能

芯片的每个定时器都支持最长定时时间 2^{26} 个时钟周期，执行增量计数模式。如果系统时钟周期为 32MHz，则最长定时时间为： $31.25\text{nS} * 2^{26} \approx 2\text{S}$ 。每个定时器都有独立的中断。

定时功能操作步骤如下：

- (1)、将 RB_TMR_ALL_CLEAR 置位，清零 R32_TMRx_COUNT 和中断标志等；
- (2)、设置寄存器 R32_TMRx_CNT_END 为需要定时的时间值；
具体计算方法为： $\text{Time} = T_{\text{sys}} * R32_TMRx_CNT_END$ ；
- (3)、清零 RB_TMR_ALL_CLEAR，清零 RB_TMR_MODE_IN 对应定时模式；
- (4)、可选步骤，设置 R8_TMRx_INTER_EN 寄存器，置位 RB_TMR_IE_CYC_END 可打开定时周期中断；
- (5)、将 R8_TMRx_CTRL_MOD 寄存器的 RB_TMR_COUNT_EN 置位，启动定时器计数；
- (6)、当计数到 R32_TMRx_COUNT 等于 R32_TMRx_CNT_END 时，定时完成。此时 R8_TMRx_INT_FLAG 的 RB_TMR_IF_CYC_END 将置 1，可写 1 清零。

计数功能操作步骤如下：

- (1)、将计数对应的 I/O 引脚方向设置为输入；
- (2)、在 R32_TMRx_CNT_END 中设定计数溢出终值；
- (3)、配置 R8_TMRx_CTRL_MOD，置位 RB_TMR_MODE_IN 和 RB_TMR_CAP_COUNT 对应计数模式，清零 RB_TMR_ALL_CLEAR 位，通过 RB_TMR_CAP_EDGE 选择采样边沿方式，将 R8_TMRx_CTRL_MOD 的 RB_TMR_COUNT_EN 置 1，使能计数功能；
- (4)、可选步骤，如果需要启用中断则设置相应的中断使能寄存器位；
- (5)、R32_TMRx_COUNT 存放当前计数值，每次计数达到计数终值，RB_TMR_IE_CYC_END 都会置 1 并且 R32_TMRx_COUNT 清 0，如果开启中断则会触发硬件中断。

8.3.2 PWM 功能

芯片的每个定时器均具有 PWM 功能。其中 TMR1 和 TMR2 的 PWM 功能支持 DMA 数据加载。PWM 可设置默认输出极性为高电平或低电平，同一数据重复输出次数可选为 1, 4, 8 或 16 次，该重复功能结合 DMA 可以用于模仿 DAC 的效果。PWM 输出有效电平的最短时间单位为 1 个系统时钟，可动态修改 PWM 的占空比，模仿出特殊波形。

PWM 操作步骤如下：

- (1)、将 RB_TMR_ALL_CLEAR 置位，清空和清零 R32_TMRx_FIFO 和中断标志等；
- (2)、设置 PWM 总周期寄存器 R32_TMRx_CNT_END，该值应该不小于 R32_TMRx_FIFO 寄存器中的值；
- (3)、配置 R8_TMRx_CTRL_MOD，清零 RB_TMR_ALL_CLEAR，清零 RB_TMR_MODE_IN 对应 PWM 模式，通过 RB_TMR_OUT_POLAR 选择输出极性，根据需要通过 RB_TMR_PWM_REPEAT 选择同一数据重复次数；
- (4)、设置数据寄存器 R32_TMRx_FIFO，最小值为 0，对应占空比 0%，最大值同 R32_TMR_CNT_END，对应占空比 100%，占空比计算： $R32_TMRx_FIFO / R32_TMRx_CNT_END$ 。TMR1 和 TMR2 可通过 DMA 加载连续动态数据，结合同一数据重复输出次数，可以模仿出特殊波形；
- (5)、配置 R8_TMRx_CTRL_MOD，置位 RB_TMR_COUNT_EN 启动计数和 RB_TMR_OUT_EN 允许 PWM 输出；
- (6)、将 PWM 对应的 I/O 引脚设置为输出；
- (7)、可选步骤，如果需要启用中断则设置相应的中断使能寄存器位；
- (8)、PWM 一个周期完成后，如果开启中断则当 RB_TMR_IF_DATA_ACT 或 RB_TMR_IF_CYC_END 置位后会触发硬件中断；
- (9)、更新 R32_TMRx_FIFO 中数据可以动态改变 PWM 的占空比，建议通过 DMA 加载。

例如：设置 RB_TMR_OUT_POLAR 位为 0，R32_TMRx_FIFO 为 6，R32_TMRx_CNT_END 为 18，则产生 PWM 的基本时序图如下所示，其占空比为： $R32_TMRx_FIFO / R32_TMRx_CNT_END = 1/3$

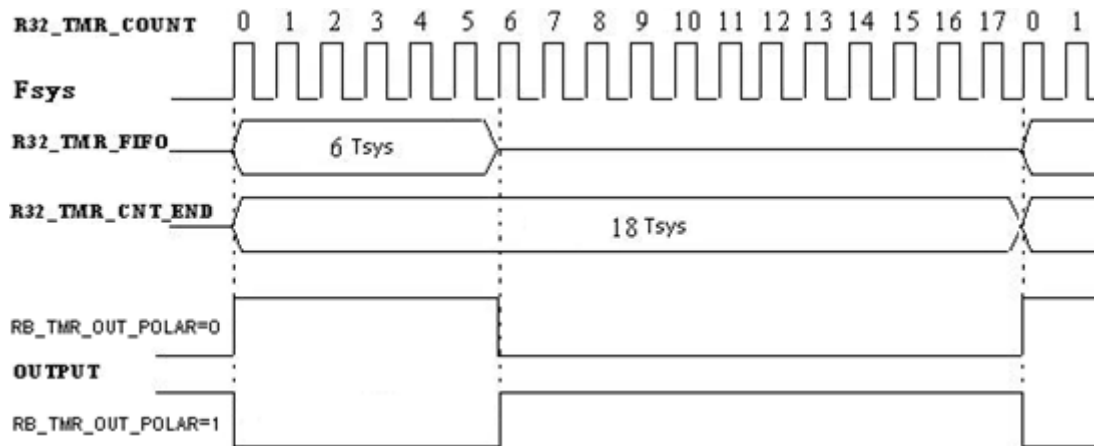


图 8-1 PWM 输出时序图

如果 RB_TMR_PWM_REPEAT 设置为 00 则表示上述过程重复 1 次，01 表示重复 4 次，10 表示重复 8 次，11 表示重复 16 次。重复之后再加载 FIFO 中的下一个数据继续。

8.3.3 捕捉功能

芯片的每个定时器都具备捕捉功能，其中 TMR1 和 TMR2 的捕捉功能支持 DMA 数据存储。捕捉模式可以选择任何边沿触发开始至任何边沿触发结束、上升沿触发开始至上升沿触发结束、下降沿触发开始至下降沿触发结束三种模式。以下为捕捉触发模式说明表：

表 8-5 捕捉触发模式说明表

| 捕捉模式选择位 RB_TMR_CATCH_EDGE | 触发方式 | 图示 |
|------------------------------|---------------|----|
| 00 | 不捕捉 | 无 |
| 01 | 边沿触发 边沿至边沿 | |
| 10 | 下降沿至下降沿 | |
| 11 | 上升沿至上升沿 | |

边沿触发模式下有 2 种触发状态，可以捕捉高电平宽度或低电平宽度。数据寄存器 R32_TMRx_FIFO 的有效数据的最高位（位 25）为 1 表示捕捉到的是高电平，为 0 表示捕捉到的是低电平。如果连续多组数据的位 25 都是 1（或 0），说明该高（或低）电平的宽度超过超时值，需多组合并累计。

下降沿至下降沿、上升沿至上升沿触发模式下，可以捕捉一个输入变化周期。数据寄存器 R32_TMRx_FIFO 的有效数据的最高位（位 25）为 0 表示正常采样到一个周期，为 1 则表示输入变化周期超过超时值 R32_TMRx_CNT_END，需加上后一组数据累计为单个输入变化周期。

具体说明如下图所示：

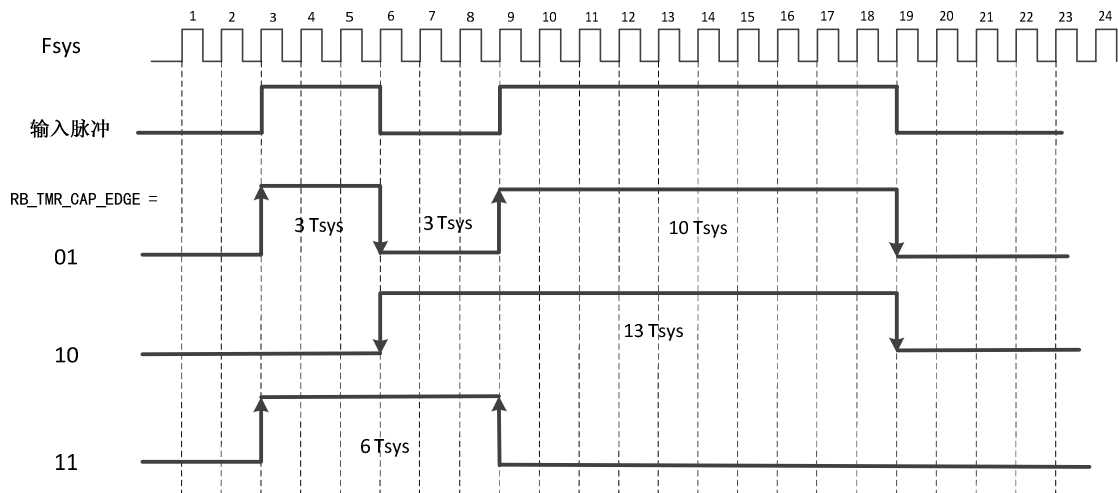


图 8-2 以系统时钟计数捕捉周期

如上图所示，每个时钟周期内采样一次：

当 RB_TMR_CAPTURE_EDGE=01b 时，设置为边沿触发采样，采样到的时间宽度为 3、3、10；

当 RB_TMR_CAPTURE_EDGE=10b 时，设置为下降沿至下降沿采样，采样到的时间宽度为 13；

当 RB_TMR_CAPTURE_EDGE=11b 时，设置为上升沿至上升沿采样，采样到的时间宽度为 6。

捕捉模式操作步骤：

- (1)、将 RB_TMR_ALL_CLEAR 置位，清空和清零 R32_TMRx_FIFO 和中断标志等；
- (2)、将捕捉对应的 I/O 引脚方向设置为输入；
- (3)、在 R32_TMRx_CNT_END 中设定合理的捕捉超时时间，可用于在输入信号长时间无变化时产生超时中断，也可以在输入信号无变化超时后，产生超时数据（数据的位 25 为 1，低 25 位可向后累计）；
- (4)、配置 R8_TMRx_CTRL_MOD，置位 RB_TMR_MODE_IN 对应捕捉模式，通过 RB_TMR_CAPTURE_EDGE 选择捕捉的边沿方式，将 R8_TMRx_CTRL_MOD 的 RB_TMR_COUNT_EN 置 1，使能计数；
- (5)、可选步骤，如果需要启用中断则设置相应的中断使能寄存器位；
- (6)、如果需要采用 DMA 方式保存捕捉的数据，需要设置寄存器 R16_TMRx_DMA_BEG 为存储捕捉数据缓冲区的首地址，设置寄存器 R16_TMRx_DMA_END 为存储捕捉数据缓冲区的结束地址（不含），并设置 R8_TMRx_CTRL_DMA 的 RB_TMR_DMA_ENABLE 为 1，使能 DMA 功能；
- (7)、将 R8_TMRx_CTRL_MOD 的 RB_TMR_ALL_CLEAR 清零，启动捕捉功能；
- (8)、每次捕捉到数据，RB_TMR_IF_DATA_ACT 都会置 1，如果开启中断则会触发硬件中断，捕捉到的数据默认存放在 R32_TMRx_FIFO 中，如果开启 DMA，则捕捉到的数据会自动存放在 DMA 设置的数据缓冲区中。

第 9 章 通用异步收发器 UART

9.1 UART 简介

CH583 和 CH582 芯片提供了 4 组全双工的异步串口 UART0/1/2/3；CH581 芯片提供了 2 组全双工的异步串口 UART0 和 UART1。支持全双工和半双工串口通讯，其中 UART0 提供发送状态引脚用于切换 RS485，并且支持 MODEM 调制解调器信号 CTS、DSR、RI、DCD、DTR、RTS。

9.1.1 主要特性

- 兼容 16C550 异步串口并且有所增强。
- 支持 5、6、7 或者 8 个数据位以及 1 或者 2 个停止位。
- 支持奇、偶、无校验、空白 0、标志 1 等校验方式。
- 可编程通讯波特率，最高达 6Mbps 波特率。
- 内置 8 个字节的 FIFO 先进先出缓冲器，支持 4 个 FIFO 触发级。
- UART0 支持 MODEM 调制解调器信号 CTS、DSR、RI、DCD、DTR、RTS。
- UART0 支持硬件流控制信号 CTS 和 RTS 自动握手和自动传输速率控制，兼容 TL16C550C。
- 支持串口帧错误检测、支持 Break 线路间隔检测。
- 支持全双工和半双工串口通讯，UART0 提供发送状态引脚用于切换 RS485。

9.2 寄存器描述

表 9-1 UART0 相关寄存器列表

| 名称 | 访问地址 | 描述 | 复位值 |
|--------------|------------|-------------------|------|
| R8_UART0_MCR | 0x40003000 | 调制解调器 MODEM 控制寄存器 | 0x00 |
| R8_UART0_IER | 0x40003001 | 中断使能寄存器 | 0x00 |
| R8_UART0_FCR | 0x40003002 | FIFO 控制寄存器 | 0x00 |
| R8_UART0_LCR | 0x40003003 | 线路控制寄存器 | 0x00 |
| R8_UART0_IIR | 0x40003004 | 中断识别寄存器 | 0x01 |
| R8_UART0_LSR | 0x40003005 | 线路状态寄存器 | 0x60 |
| R8_UART0_MSR | 0x40003006 | 调制解调器 MODEM 状态寄存器 | 0xX0 |
| R8_UART0_RBR | 0x40003008 | 接收缓冲寄存器 | 0xXX |
| R8_UART0_THR | 0x40003008 | 发送保持寄存器 | 0xXX |
| R8_UART0_RFC | 0x4000300A | 接收 FIFO 计数寄存器 | 0x00 |
| R8_UART0_TFC | 0x4000300B | 发送 FIFO 计数寄存器 | 0x00 |
| R16_UART0_DL | 0x4000300C | 波特率除数锁存器 | 0xXX |
| R8_UART0_DIV | 0x4000300E | 预分频除数寄存器 | 0xXX |
| R8_UART0_ADR | 0x4000300F | 从机地址寄存器 | 0xFF |

表 9-2 UART1 相关寄存器列表

| 名称 | 访问地址 | 描述 | 复位值 |
|--------------|------------|-------------------|------|
| R8_UART1_MCR | 0x40003400 | 调制解调器 MODEM 控制寄存器 | 0x00 |
| R8_UART1_IER | 0x40003401 | 中断使能寄存器 | 0x00 |
| R8_UART1_FCR | 0x40003402 | FIFO 控制寄存器 | 0x00 |
| R8_UART1_LCR | 0x40003403 | 线路控制寄存器 | 0x00 |
| R8_UART1_IIR | 0x40003404 | 中断识别寄存器 | 0x01 |
| R8_UART1_LSR | 0x40003405 | 线路状态寄存器 | 0x60 |

| | | | |
|--------------|------------|---------------|------|
| R8_UART1_RBR | 0x40003408 | 接收缓冲寄存器 | 0xXX |
| R8_UART1_THR | 0x40003408 | 发送保持寄存器 | 0xXX |
| R8_UART1_RFC | 0x4000340A | 接收 FIFO 计数寄存器 | 0x00 |
| R8_UART1_TFC | 0x4000340B | 发送 FIFO 计数寄存器 | 0x00 |
| R16_UART1_DL | 0x4000340C | 波特率除数锁存器 | 0xXX |
| R8_UART1_DIV | 0x4000340E | 预分频除数寄存器 | 0xXX |

表 9-3 UART2 相关寄存器列表

| 名称 | 访问地址 | 描述 | 复位值 |
|--------------|------------|-------------------|------|
| R8_UART2_MCR | 0x40003800 | 调制解调器 MODEM 控制寄存器 | 0x00 |
| R8_UART2_IER | 0x40003801 | 中断使能寄存器 | 0x00 |
| R8_UART2_FCR | 0x40003802 | FIFO 控制寄存器 | 0x00 |
| R8_UART2_LCR | 0x40003803 | 线路控制寄存器 | 0x00 |
| R8_UART2_IIR | 0x40003804 | 中断识别寄存器 | 0x01 |
| R8_UART2_LSR | 0x40003805 | 线路状态寄存器 | 0x60 |
| R8_UART2_RBR | 0x40003808 | 接收缓冲寄存器 | 0xXX |
| R8_UART2_THR | 0x40003808 | 发送保持寄存器 | 0xXX |
| R8_UART2_RFC | 0x4000380A | 接收 FIFO 计数寄存器 | 0x00 |
| R8_UART2_TFC | 0x4000380B | 发送 FIFO 计数寄存器 | 0x00 |
| R16_UART2_DL | 0x4000380C | 波特率除数锁存器 | 0xXX |
| R8_UART2_DIV | 0x4000380E | 预分频除数寄存器 | 0xXX |

表 9-4 UART3 相关寄存器列表

| 名称 | 访问地址 | 描述 | 复位值 |
|--------------|------------|-------------------|------|
| R8_UART3_MCR | 0x40003C00 | 调制解调器 MODEM 控制寄存器 | 0x00 |
| R8_UART3_IER | 0x40003C01 | 中断使能寄存器 | 0x00 |
| R8_UART3_FCR | 0x40003C02 | FIFO 控制寄存器 | 0x00 |
| R8_UART3_LCR | 0x40003C03 | 线路控制寄存器 | 0x00 |
| R8_UART3_IIR | 0x40003C04 | 中断识别寄存器 | 0x01 |
| R8_UART3_LSR | 0x40003C05 | 线路状态寄存器 | 0x60 |
| R8_UART3_RBR | 0x40003C08 | 接收缓冲寄存器 | 0xXX |
| R8_UART3_THR | 0x40003C08 | 发送保持寄存器 | 0xXX |
| R8_UART3_RFC | 0x40003C0A | 接收 FIFO 计数寄存器 | 0x00 |
| R8_UART3_TFC | 0x40003C0B | 发送 FIFO 计数寄存器 | 0x00 |
| R16_UART3_DL | 0x40003C0C | 波特率除数锁存器 | 0xXX |
| R8_UART3_DIV | 0x40003C0E | 预分频除数寄存器 | 0xXX |

调制解调器 MODEM 控制寄存器 (R8_UARTx_MCR) (x=0/1/2/3)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---|-------------|----|---|-----|
| 7 | RB_MCR_HALF | RW | 半双工收发模式控制 (仅 UART0 支持): 1: 进入半双工收发模式, 发送优先, 不发送时为接收; 0: 禁止半双工模式。 | 0 |
| 6 | RB_MCR_TNOW | RW | DTR 引脚输出正在发送状态使能 (仅 UART0 支持): 1: 将正在发送指示状态输出到 DTR 引脚, 可以用于控制 RS485 收发切换; 0: DTR 引脚为正常功能。 | 0 |

| | | | | |
|---|------------------------------|----|---|---|
| 5 | RB_MCR_AU_FLOW_EN | RW | CTS 和 RTS 硬件自动流控使能 (仅 UART0 支持): 1: 启用 CTS 和 RTS 硬件自动流控; 0: 禁用 CTS 和 RTS 硬件自动流控。 流控模式下, 如果此位为 1, 那么仅在检测到 CTS 引脚输入有效 (低电平有效) 时串口才继续发送下一个数据, 否则暂停串口发送, 当此位为 1 时的 CTS 输入状态变化不会产生 MODEM 状态中断。如果此位为 1 并且 RTS 为 1, 那么当接收 FIFO 空时, 串口会自动有效 RTS 引脚 (低电平有效), 直到接收的字节数达到 FIFO 的触发点时, 串口才自动无效 RTS 引脚, 并能够在接收 FIFO 空时再次有效 RTS 引脚。使用硬件自动流控, 可将己方的 CTS 引脚接到对方的 RTS 引脚, 并将己方的 RTS 引脚送到对方的 CTS 引脚。 | 0 |
| 4 | RB_MCR_LOOP | RW | 内部回路的测试模式使能 (仅 UART0 支持): 1: 使能内部回路的测试模式; 0: 禁止内部回路的测试模式。 在内部回路的测试模式下, 串口所有对外输出引脚均为无效状态, TXD 内部返回到 RXD, RTS 内部返回到 CTS, DTR 内部返回到 DSR, OUT1 内部返回到 RI, OUT2 内部返回到 DCD。 | 0 |
| 3 | RB_MCR_OUT2 RB_MCR_INT_OE | RW | 串口的中断请求输出控制位: 1: 允许发出请求; 0: 禁止。 | 0 |
| 2 | RB_MCR_OUT1 | RW | 用户自定义 MODEM 控制位 (仅 UART0 支持), 没有连接实际输出引脚: 1: 置高; 0: 置低。 | 0 |
| 1 | RB_MCR_RTS | RW | RTS 信号输出电平控制 (仅 UART0 支持): 1: RTS 信号输出有效 (低电平); 0: RTS 信号输出高电平 (默认)。 | 0 |
| 0 | RB_MCR_DTR | RW | DTR 信号输出电平控制 (仅 UART0 支持): 1: DTR 信号输出有效 (低电平); 0: DTR 信号输出高电平 (默认)。 | 0 |

中断使能寄存器 (RB_UARTx_IER) (x=0/1/2/3)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---|------------------|----|--|-----|
| 7 | RB_IER_RESET | WZ | 串口软件复位控制位, 自动清零: 1: 软件复位该串口; 0: 正常工作。 | 0 |
| 6 | RB_IER_TXD_EN | RW | 串口 TXD 引脚输出使能位: 1: 使能引脚输出; 0: 禁止引脚输出。 | 0 |
| 5 | RB_IER_RTS_EN | RW | RTS 引脚输出使能位 (仅 UART0 支持): 1: 使能引脚输出; 0: 禁止引脚输出。 | 0 |
| 4 | RB_IER_DTR_EN | RW | DTR 引脚输出使能位 (仅 UART0 支持): 1: 使能引脚输出; 0: 禁止引脚输出。 | 0 |
| 3 | RB_IER_MODEM_CHG | RW | 调制解调器输入状态变化中断使能位 | 0 |

| | | | | |
|---|------------------|----|-------------------------------------|---|
| | | | (仅 UART0 支持): 1: 使能中断; 0: 禁止中断。 | |
| 2 | RB_IER_LINE_STAT | RW | 接收线路状态中断使能位: 1: 使能中断; 0: 禁止中断。 | 0 |
| 1 | RB_IER_THR_EMPTY | RW | 发送保持寄存器空中断使能位: 1: 使能中断; 0: 禁止中断。 | 0 |
| 0 | RB_IER_RECV_RDY | RW | 接收数据中断使能位: 1: 使能中断; 0: 禁止中断。 | 0 |

FIFO 控制寄存器 (R8_UARTx_FCR) (x=0/1/2/3)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|--------------------|----|---|------|
| [7:6] | RB_FCR_FIFO_TRIG | RW | 接收 FIFO 的中断和硬件流控制的触发点选择: 00: 1 字节; 01: 2 字节; 10: 4 字节; 11: 7 字节。 用来设置接收 FIFO 的中断和硬件流控制的触发点, 例如: 10 对应 4 个字节, 即接收满 4 个字节产生接收数据可用的中断, 并在使能硬件流控制时自动无效 RTS 引脚。 | 00b |
| [5:3] | Reserved | RO | 保留。 | 000b |
| 2 | RB_FCR_TX_FIFO_CLR | WZ | 发送 FIFO 数据清空使能位, 自动清零: 1: 清空发送 FIFO 的数据 (不含 TSR); 0: 不清空发送 FIFO 的数据。 | 0 |
| 1 | RB_FCR_RX_FIFO_CLR | WZ | 接收 FIFO 数据清空使能位, 自动清零: 1: 清空接收 FIFO 的数据 (不含 RSR); 0: 不清空接收 FIFO 的数据。 | 0 |
| 0 | RB_FCR_FIFO_EN | RW | FIFO 使能位: 1: 启用 8 字节 FIFO; 0: 禁用 FIFO。 禁用 FIFO 后为 16C450 兼容模式, 相当于 FIFO 只有一个字节 (RECV_TG1=0、RECV_TG0=0、FIFO_EN=1), 建议启用。 | 0 |

线路控制寄存器 (R8_UARTx_LCR) (x=0/1/2/3)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|------------------------------|----|--|-----|
| 7 | RB_LCR_DLAB RB_LCR_GP_BIT | RW | 串口通用位, 用户自定义。 | 0 |
| 6 | RB_LCR_BREAK_EN | RW | 强制产生 BREAK 线路间隔使能位: 1: 强制产生; 0: 不产生。 | 0 |
| [5:4] | RB_LCR_PAR_MOD | RW | 奇偶校验位格式选择: 00: 奇校验; 01: 偶校验; 10: 标志位 (MARK, 置 1); 11: 空白位 (SPACE, 清 0)。 仅当 RB_LCR_PAR_EN 位为 1 时有效。 | 00b |
| 3 | RB_LCR_PAR_EN | RW | 奇偶校验位使能位: 1: 允许发送时产生和接收时校验奇偶校验位; 0: 无奇偶校验位。 | 0 |
| 2 | RB_LCR_STOP_BIT | RW | 停止位格式设置位: | 0 |

| | | | | |
|-------|----------------|----|---|-----|
| | | | 0: 一个停止位; 1: 两个停止位。 | |
| [1:0] | RB_LCR_WORD_SZ | RW | 串口数据长度选择: 00: 5 个数据位; 01: 6 个数据位; 10: 7 个数据位; 11: 8 个数据位。 | 00b |

中断识别寄存器 (R8_UARTx_IIR) (x=0/1/2/3)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|-----------------|----|---|-------|
| [7:6] | RB_IIR_FIFO_ID | RO | 串口 FIFO 启用状态位: 11: FIFO 已启用; 00: FIFO 未启用。 | 00b |
| [5:4] | Reserved | RO | 保留。 | 00b |
| [3:0] | RB_IIR_INT_MASK | RO | 中断标志: 如果 RB_IIR_NO_INT 位为 0, 则表示有中断产生, 需要读取后判断中断源。具体请参看表 9-5。 | 0000b |
| 0 | RB_IIR_NO_INT | RO | 串口无中断标志位: 1: 无中断; 0: 有中断。 | 1 |

中断识别寄存器 R8_UARTx_IIR 的 RB_IIR_NO_INT 位以及 RB_IIR_INT_MASK 的每一个位所表示的含义如下表所示:

表 9-5 IIR 寄存器中 RB_IIR_INT_MASK 含义

| IIR 寄存器位 | | | | 优先级 | 中断类型 | 中断源 | 清中断方法 |
|----------|------|------|-------|-----|------------|--|---------------|
| IID3 | IID2 | IID1 | NOINT | | | | |
| 0 | 0 | 0 | 1 | 无 | 没有中断产生 | 没有中断 | |
| 1 | 1 | 1 | 0 | 0 | 总线地址匹配 | 接收到 1 个数据是串口总线地址, 且该地址与预置从机值相匹配或是广播地址。(仅 UART0 支持) | 读 IIR 或禁用多机模式 |
| 0 | 1 | 1 | 0 | 1 | 接收线路状态 | OVER_ERR、PAR_ERR、FRAM_ERR、BREAK_ERR | 读 LSR |
| 0 | 1 | 0 | 0 | 2 | 接收数据可用 | 接收到的字节数达到 FIFO 的触发点。 | 读 RBR |
| 1 | 1 | 0 | 0 | 2 | 接收数据超时 | 超过 4 个数据时间未收到下一数据。 | 读 RBR |
| 0 | 0 | 1 | 0 | 3 | THR 寄存器空 | 发送保持寄存器空, 或者, RB_IER_THR_EMPTY 位从 0 变 1 触发。 | 读 IIR 或写 THR |
| 0 | 0 | 0 | 0 | 4 | MODEM 输入变化 | △CTS、△DSR、△RI、△DCD 置 1 触发。 | 读 MSR |

线路状态寄存器 (R8_UARTx_LSR) (x=0/1/2/3)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---|--------------------|----|---|-----|
| 7 | RB_LSR_ERR_RX_FIFO | RO | 接收 FIFO 错误标志位: 1: 接收 FIFO 中存在至少一个 PAR_ERR、FRAM_ERR 或 BREAK_ERR 错误; 0: 接收 FIFO 中不存在错误。 | 0 |
| 6 | RB_LSR_TX_ALL_EMP | RO | 发送保持寄存器 THR 和发送移位寄存器 TSR 全空标志位: 1: 两者全空; 0: 两者非全空。 | 1 |
| 5 | RB_LSR_TX_FIFO_EMP | RO | 发送 FIFO 空标志位: 1: 发送 FIFO 空; 0: 发送 FIFO 非空。 | 1 |
| 4 | RB_LSR_BREAK_ERR | RZ | BREAK 线路间隔检测标志位: 1: 检测到 BREAK; 0: 未检测到 BREAK。 | 0 |
| 3 | RB_LSR_FRAME_ERR | RZ | 数据帧错误标志位: 1: 表示正在从接收 FIFO 中读取的数据存在帧错误, 缺少有效的停止位; 0: 当前读取的数据帧没有错误。 | 0 |

| | | | | |
|---|-----------------|----|---|---|
| 2 | RB_LSR_PAR_ERR | RZ | 接收数据奇偶校验错误标志位： 1：表示正在从接收 FIFO 中读取的数据存在奇偶校验错； 0：当前读取的数据奇偶校验正确。 | 0 |
| 1 | RB_LSR_OVER_ERR | RZ | 接收 FIFO 缓冲区溢出标志位： 1：已溢出；0：未溢出。 | 0 |
| 0 | RB_LSR_DATA_RDY | RO | 接收 FIFO 中有接收到的数据标志位： 1：FIFO 中有数据；0：无数据。 读取 FIFO 中所有数据后，该位自动清 0。 | 0 |

调制解调器 MODEM 状态寄存器 (R8_UART0_MSR) (仅 UART0 支持)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---|----------------|----|--|-----|
| 7 | RB_MSR_DCD | RO | DCD 引脚状态位： 1：DCD 有效（低电平）； 0：DCD 无效（高电平）。 | 0 |
| 6 | RB_MSR_RI | RO | RI 引脚状态位： 1：RI 有效（低电平）； 0：RI 无效（高电平）。 | 0 |
| 5 | RB_MSR_DSR | RO | DSR 引脚状态位： 1：DSR 引脚有效（低电平）； 0：DSR 引脚无效（高电平）。 | X |
| 4 | RB_MSR_CTS | RO | CTS 引脚状态位： 1：CTS 引脚有效（低电平）； 0：CTS 引脚无效（高电平）。 | X |
| 3 | RB_MSR_DCD_CHG | RZ | DCD 引脚输入状态变化标志位： 1：发生过变化；0：无变化。 | 0 |
| 2 | RB_MSR_RI_CHG | RZ | RI 引脚输入状态变化标志位： 1：发生过变化；0：无变化。 | 0 |
| 1 | RB_MSR_DSR_CHG | RZ | DSR 引脚输入状态变化标志位： 1：发生过变化；0：无变化。 | 0 |
| 0 | RB_MSR_CTS_CHG | RZ | CTS 引脚输入状态变化标志位： 1：发生过变化；0：无变化。 | 0 |

接收缓冲寄存器 (R8_UARTx_RBR) (x=0/1/2/3)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|--------------|----|--|-----|
| [7:0] | R8_UARTx_RBR | RO | 数据接收缓冲寄存器。 如果 LSR 的 DATA_RDY 位为 1，则可以从该寄存器读取接收到的数据； 如果 FIFO_EN 为 1，则从串口移位寄存器 RSR 接收到的数据首先被存放于接收 FIFO 中，然后通过该寄存器读出。 | XXh |

发送保持寄存器 (R8_UARTx_THR) (x=0/1/2/3)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|--------------|----|--|-----|
| [7:0] | R8_UARTx_THR | WO | 发送保持寄存器。 包括发送 FIFO，用于写入准备发送的数据；如果 FIFO_EN 为 1，则写入的数据首先被存放于发送 FIFO 中，然后通过发 | XXh |

| | | | | |
|--|--|--|------------------|--|
| | | | 送移位寄存器 TSR 逐个输出。 | |
|--|--|--|------------------|--|

接收 FIFO 计数寄存器 (R8_UARTx_RFC) (x=0/1/2/3)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|--------------|----|------------------|-----|
| [7:0] | R8_UARTx_RFC | RO | 当前接收 FIFO 中数据计数。 | 00h |

发送 FIFO 计数寄存器 (R8_UARTx_TFC) (x=0/1/2/3)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|--------------|----|------------------|-----|
| [7:0] | R8_UARTx_TFC | RO | 当前发送 FIFO 中数据计数。 | 00h |

波特率除数锁存器 (R16_UARTx_DL) (x=0/1/2/3)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|--------------|----|---|-------|
| [15:0] | R16_UARTx_DL | RW | 16 位除数用于计算波特率。 公式：除数 = 串口内部基准时钟 Fuart / 16 / 所需通讯波特率。 例：如果串口内部基准时钟 Fuart 为 1.8432MHz，所需波特率为 9600bps，则除数=1843200/16/9600=12。 | XXXXh |

预分频除数寄存器 (R8_UARTx_DIV) (x=0/1/2/3)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|--------------|----|--|-----|
| [7:0] | R8_UARTx_DIV | RW | 用于计算串口的内部基准时钟，低 7 位有效。 公式：除数 = Fsys * 2 / 串口内部基准时钟，最大值 127。 | XXh |

从机地址寄存器 (R8_UART0_ADR) (仅 UART0 支持)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|--------------|----|--------------------------------------|------|
| [7:0] | R8_UART0_ADR | RW | 串口 0 多机通讯时的从机地址： FFh：不使用；其他：从机地址。 | 0FFh |

R8_UART0_ADR 预置本机作为从机时的地址，用于在多机通讯时自动比较接收到的地址，并在地址匹配或者在接收到广播地址 0FFh 时产生中断，同时允许接收后续数据包。在地址没有匹配之前不接收任何数据，开始发送数据后或者重写 R8_UART0_ADR 寄存器后停止接收任何数据，直到下次地址再次匹配或者接收到广播地址时再允许接收。

R8_UART0_ADR 为 0FFh 时或者 RB_LCR_PAR_EN=0 时，禁用总线地址自动比较功能。

R8_UART0_ADR 不为 0FFh 并且 RB_LCR_PAR_EN=1 时，启用总线地址自动比较功能，同时应该配置下述参数：RB_LCR_WORD_SZ 为 11b 以选择 8 个数据位方式，对于地址字节为 MARK 的情况（即数据字节的位 9 为 0），应设置 RB_LCR_PAR_MOD 为 10b，对于地址字节为 SPACE 的情况（即数据字节的位 9 为 1），应设置 RB_LCR_PAR_MOD 为 11b。

9.3 功能描述及配置

UART0/1/2/3 输出引脚都是 3.3V LVC MOS 电平。异步串口方式下引脚包括：数据传输引脚和 MODEM 联络信号引脚（仅 UART0 支持）。数据传输引脚包括：TXD 引脚和 RXD 引脚，默认都是高电平；MODEM 联络信号引脚包括：CTS 引脚、DSR 引脚、RI 引脚、DCD 引脚、DTR 引脚、RTS 引脚，默认都是高电平。所有这些 MODEM 联络信号都可以作为通用 I/O 引脚，由应用程序控制并定义其用途。

4 组 UART 各自拥有独立的收发缓冲区及 8 字节 FIFO，支持单工、半双工或者全双工异步串行通

讯。串行数据包括 1 个低电平起始位，5、6、7 或 8 个数据位，0 个或者 1 个附加校验位或者标志位，1 个或者 2 个高电平停止位，支持奇校验/偶校验/标志校验/空白校验。串口发送信号的波特率误差小于 0.5%，串口接收信号的允许波特率误差不大于 2%。

9.3.1 波特率计算

1) 计算串口内部基准时钟 F_{uart} ，设置 $R8_UARTx_DIV$ 寄存器，最大值 127，通常写入 1。

2) 计算波特率，设置 $R16_UARTx_DL$ 寄存器。

波特率公式 $=F_{sys} * 2 / R8_UARTx_DIV / 16 / R16_UARTx_DL$ 。

9.3.2 串口发送

串口发送的“THR 寄存器空”中断 $UART_II_THR_EMPTY$ 是指当前发送 FIFO 空。当读取 IIR 寄存器后，该中断被清除，或者当向 THR 写入下一个数据后，该中断也能被清除。如果仅仅是向 THR 写入一个字节，那么由于该字节很快被转移到发送移位寄存器 TSR 中开始发送，所以很快会再次产生发送 THR 寄存器空中断的请求，此时可以写入下一个准备发送的数据。当 TSR 寄存器中的数据被全部移出后，串口发送才真正完成，此时 LSR 寄存器的 $RB_LSR_TX_ALL_EMP$ 位变为 1 有效。

在中断触发方式下，当收到串口发送保持寄存器 THR 空的中断后，如果已使能 FIFO，那么可以向 THR 寄存器及 FIFO 一次写入最多 8 字节，然后控制器会按顺序自动发送；如果禁止 FIFO，那么一次只能写入一个字节；如果没有数据需要发送，那么可以直接退出（之前读取 IIR 时已经自动清除中断）。

在查询方式下，可以根据 LSR 寄存器的 $RB_LSR_TX_FIFO_EMP$ 位判断发送 FIFO 是否为空，当此位为 1 则可以向 THR 寄存器及 FIFO 写入数据，如果使能 FIFO，那么一次可以写入最多 8 个字节。

也可读取 $R8_UARTx_TFC$ 寄存器判断当前 FIFO 中待发送的剩余数据个数，如果不等于 8，则可继续向 FIFO 中写入待发送数据，这种方式可以节约填充时间。

9.3.3 串口接收

串口接收数据可用中断 $UART_II_RCV_RDY$ 是指接收 FIFO 中的已有数据字节数已经到或超过由 FCR 寄存器的 $RB_FCR_FIFO_TRIG$ 设置选择的 FIFO 触发点。当从 RBR 读取数据使 FIFO 字数低于 FIFO 触发点时，该中断被清除。

串口接收数据超时中断 $UART_II_RCV_TOUT$ 是指接收 FIFO 中至少有一个字节的数据，并且从上一次串口接收到数据和从上一次被系统取走数据开始，已经等待了相当于接收 4 个数据的时间。当再次接收到一个新的数据后，该中断被清除，或者当单片读取一次 RBR 寄存器后，该中断也能被清除。当接收 FIFO 全空时，LSR 寄存器的 $RB_LSR_DATA_RDY$ 位为 0，当接收 FIFO 中有数据时， $RB_LSR_DATA_RDY$ 位为 1 有效。

在中断触发方式下，当收到串口接收数据超时的中断后，可以读取 $R8_UARTx_RFC$ 寄存器查询当前 FIFO 中剩余数据计数，直接读取全部数据，或者不断查询 LSR 寄存器的 $RB_LSR_DATA_RDY$ ，如果此位有效则读数据，直到此位无效。当收到串口接收数据可用的中断后，可以先从 RBR 寄存器一次性读取 $RB_FCR_FIFO_TRIG$ 设定字节个数的数据，或者也可以根据 $RB_LSR_DATA_RDY$ 位和 $R8_UARTx_RFC$ 寄存器读取当前 FIFO 中所有数据。

在查询方式下，可以根据 LSR 寄存器的 $RB_LSR_DATA_RDY$ 位判断接收 FIFO 是否为空，或读取 $R8_UARTx_RFC$ 寄存器获取当前 FIFO 中数据计数，来获取串口接收的所有数据。

9.3.4 硬件流控制

硬件流控制包括自动 CTS（ $RB_MCR_AU_FLOW_EN$ 置 1）和自动 RTS（ $RB_MCR_AU_FLOW_EN$ 和 RB_MCR_RTS 都置 1）。

如果使能自动 CTS，那么 CTS 引脚在串口发送数据之前必须有效。串口发送器在发送下一个数据之前会检测 CTS 引脚，当 CTS 引脚状态有效时，发送器发送下一个数据。为了使发送器停止发送后面

的数，CTS 引脚必须在当前发送的最后一个停止位的中间时刻之前被无效。自动 CTS 功能减少了向系统申请的中断。当使能硬件流控制后，由于控制器会根据 CTS 引脚状态自动控制发送器，所以 CTS 引脚电平的改变不会触发 MODEM 中断。如果使能自动 RTS，那么仅当 FIFO 中有足够空间接收数据时才使 RTS 引脚输出有效，而在接收 FIFO 满时使 RTS 引脚输出无效。如果接收 FIFO 中的数据被全部取走或清空，那么 RTS 引脚输出有效。当到达接收 FIFO 的触发点时（接收 FIFO 中已有字节数不少于 FCR 寄存器的 RB_FCR_FIFO_TRIG 设定的字节数），RTS 引脚输出无效，并且允许对方发送器在 RTS 引脚无效后再发送一个另外的数据。一旦接收 FIFO 被取空数据，RTS 引脚就会自动重新有效，从而使对方的发送器恢复发送。如果自动 CTS 和自动 RTS 都被使能（MCR 寄存器的 RB_MCR_AU_FLOW_EN 和 RB_MCR_RTS 都为 1），那么当己方的 RTS 引脚连接对方的 CTS 引脚时，除非己方的接收 FIFO 中有足够的空间，否则对方不会发送数据。因此，通过这种硬件流控制，可以避免串口接收时的 FIFO 溢出和超时错误。

第 10 章 串行外设接口 SPI

10.1 SPI 简介

SPI 是一种全双工串行接口，总线上连接有一个主机和若干从机，同一时刻，仅有一对主从在通讯。通常 SPI 接口由 4 个引脚组成：SPI 片选引脚 SCS、SPI 时钟引脚 SCK、SPI 串行数据引脚 MISO（主机输入/从机输出引脚）和 SPI 串行数据引脚 MOSI（主机输出/从机输入引脚）。

10.1.1 主要特性

CH583 芯片提供 2 个 SPI 接口，CH582 和 CH581 芯片仅提供了 SPI0，特性如下：

- SPI0 支持主机模式（Master）和从机模式（Slave），SPI1 只支持主机模式（Master）。
- 兼容串行外设接口（SPI）规范。
- 支持模式 0 和模式 3 数据传输方式。
- 8 位数据传输方式，数据位序可选：字节低位在前或者高位在前。
- 时钟频率最高可达系统主频 F_{sys} 的一半。
- 8 字节 FIFO。
- SPI0 从机模式支持首字节为命令模式或数据流模式。
- SPI0 支持 DMA，数据传输效率更高。

10.2 寄存器描述

表 10-1 SPI0 相关寄存器列表

| 名称 | 访问地址 | 描述 | 复位值 |
|--|------------|--------------------------------------|--------|
| R8_SPI0_CTRL_MOD | 0x40004000 | SPI0 模式控制寄存器 | 0x02 |
| R8_SPI0_CTRL_CFG | 0x40004001 | SPI0 配置寄存器 | 0x00 |
| R8_SPI0_INTER_EN | 0x40004002 | SPI0 中断使能寄存器 | 0x00 |
| R8_SPI0_CLOCK_DIV R8_SPI0_SLAVE_PRE | 0x40004003 | SPI0 主机模式时钟分频寄存器 SPI0 从机模式预置数据寄存器 | 0x10 |
| R8_SPI0_BUFFER | 0x40004004 | SPI0 数据缓冲区 | 0xFF |
| R8_SPI0_RUN_FLAG | 0x40004005 | SPI0 工作状态寄存器 | 0x00 |
| R8_SPI0_INT_FLAG | 0x40004006 | SPI0 中断标志寄存器 | 0x40 |
| R8_SPI0_FIFO_COUNT | 0x40004007 | SPI0 收发 FIFO 计数寄存器 | 0x00 |
| R16_SPI0_TOTAL_CNT | 0x4000400C | SPI0 收发数据总长度寄存器 | 0x0000 |
| R8_SPI0_FIFO | 0x40004010 | SPI0 数据 FIFO 寄存器 | 0xFF |
| R8_SPI0_FIFO_COUNT1 | 0x40004013 | SPI0 收发 FIFO 计数寄存器 | 0x00 |
| R16_SPI0_DMA_NOW | 0x40004014 | SPI0 DMA 缓冲区当前地址 | 0xFFFF |
| R16_SPI0_DMA_BEG | 0x40004018 | SPI0 DMA 缓冲区起始地址 | 0xFFFF |
| R16_SPI0_DMA_END | 0x4000401C | SPI0 DMA 缓冲区结束地址 | 0xFFFF |

表 10-2 SPI1 相关寄存器列表

| 名称 | 访问地址 | 描述 | 复位值 |
|-------------------|------------|------------------|------|
| R8_SPI1_CTRL_MOD | 0x40004400 | SPI1 模式控制寄存器 | 0x02 |
| R8_SPI1_CTRL_CFG | 0x40004401 | SPI1 配置寄存器 | 0x00 |
| R8_SPI1_INTER_EN | 0x40004402 | SPI1 中断使能寄存器 | 0x00 |
| R8_SPI1_CLOCK_DIV | 0x40004403 | SPI1 主机模式时钟分频寄存器 | 0x10 |
| R8_SPI1_BUFFER | 0x40004404 | SPI1 数据缓冲区 | 0xFF |

| | | | |
|---------------------|------------|--------------------|------|
| R8_SPI1_RUN_FLAG | 0x40004405 | SPI1 工作状态寄存器 | 0x00 |
| R8_SPI1_INT_FLAG | 0x40004406 | SPI1 中断标志寄存器 | 0x40 |
| R8_SPI1_FIFO_COUNT | 0x40004407 | SPI1 收发 FIFO 计数寄存器 | 0x00 |
| R16_SPI1_TOTAL_CNT | 0x4000440C | SPI1 收发数据总长度寄存器 | 0x00 |
| R8_SPI1_FIFO | 0x40004410 | SPI1 数据 FIFO 寄存器 | 0xXX |
| R8_SPI1_FIFO_COUNT1 | 0x40004413 | SPI1 收发 FIFO 计数寄存器 | 0x00 |

SPI 模式控制寄存器 (R8_SPIx_CTRL_MOD) (x=0/1)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---|--------------------|----|--|-----|
| 7 | RB_SPI_MISO_OE | RW | MISO 引脚输出使能位 (可在 2 线模式数据线切换方向使用): 1: MISO 输出使能; 0: MISO 输出禁止。 | 0 |
| 6 | RB_SPI_MOSI_OE | RW | MOSI 引脚输出使能位: 1: MOSI 输出使能; 0: MOSI 输出禁止。 | 0 |
| 5 | RB_SPI_SCK_OE | RW | SCK 引脚输出使能位: 1: SCK 输出使能; 0: SCK 输出禁止。 | 0 |
| 4 | RB_SPI_FIFO_DIR | RW | FIFO 方向设置位: 1: 输入模式 (指示接收数据); 0: 输出模式 (指示发送数据)。 | 0 |
| 3 | RB_SPI_SLV_CMD_MOD | RW | SPI0 从机模式下首字节方式选择 (仅 SPI0 支持): 1: 首字节命令模式; 0: 数据流模式。 在首字节命令模式下, 当接收到 SPI 片选有效后的首字节数据后, 将视为命令码, 且 RB_SPI_IF_FST_BYTE 将置 1。 | 0 |
| 3 | RB_SPI_MST_SCK_MOD | RW | 主机模式时钟空闲方式选择: 1: 模式 3 (空闲时 SCK 为高电平); 0: 模式 0 (空闲时 SCK 为低电平)。 | 0 |
| 2 | RB_SPI_2WIRE_MOD | RW | 从机模式 2 线或 3 线 SPI 模式选择 (仅 SPI0 支持, SPI1 无需此控制位): 1: 2 线模式/半双工 (SCK/MISO); 0: 3 线模式/全双工 (SCK/MOSI/MISO)。 | 0 |
| 1 | RB_SPI_ALL_CLEAR | RW | SPI 的 FIFO/计数器/中断标志清零: 1: 强制清空和清零; 0: 不清。 | 1 |
| 0 | RB_SPI_MODE_SLAVE | RW | SPI0 主从模式选择 (仅 SPI0 支持): 1: 从机模式; 0: 主机模式。 | 0 |

SPI 配置寄存器 (R8_SPIx_CTRL_CFG) (x=0/1)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---|-------------------|----|--|-----|
| 7 | Reserved | R0 | 保留。 | 0 |
| 6 | RB_SPI_MST_DLY_EN | RW | 主机模式下输入延迟使能: 1: 使能, 用于 SPI 时钟接近 Fsys 一半等高速应用; 0: 禁止, 常规应用。 | 0 |
| 5 | RB_SPI_BIT_ORDER | RW | SPI 数据位序选择: 1: 低位在前; 0: 高位在前。 | 0 |
| 4 | RB_SPI_AUTO_IF | RW | 使能访问 BUFFER/FIFO 时自动清除标志 | 0 |

| | | | | |
|---|-------------------|----|--|---|
| | | | 位 RB_SPI_IF_BYTE_END 的功能： 1: 使能; 0: 禁止。 | |
| 3 | Reserved | R0 | 保留。 | 0 |
| 2 | RB_SPI_DMA_LOOP | RW | DMA 地址循环功能使能位 (仅 SPI0 支持): 1: 使能地址循环; 0: 禁止地址循环。 如果使能 DMA 地址循环, 当 DMA 地址增加到设置的末尾地址时, 自动循环指向设置的首地址。 | 0 |
| 1 | Reserved | R0 | 保留。 | 0 |
| 0 | RB_SPI_DMA_ENABLE | RW | DMA 功能使能位 (仅 SPI0 支持): 1: 使能 DMA; 0: 禁止 DMA。 | 0 |

SPI 中断使能寄存器 (R8_SPIx_INTER_EN) (x=0/1)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|--------------------|----|--|-----|
| 7 | RB_SPI_IE_FST_BYTE | RW | 从机模式的首字节命令模式下, 首字节接收中断使能位 (仅 SPI0 支持): 1: 使能接收到第一个字节中断; 0: 禁止接收到第一个字节中断。 | 0 |
| [6: 5] | Reserved | R0 | 保留。 | 00b |
| 4 | RB_SPI_IE_FIFO_OV | RW | FIFO 溢出 (接收时 FIFO 满或发送时 FIFO 空) 中断使能位 (仅 SPI0 支持): 1: 使能中断; 0: 禁止中断。 | 0 |
| 3 | RB_SPI_IE_DMA_END | RW | DMA 结束中断使能位 (仅 SPI0 支持): 1: 使能中断; 0: 禁止中断。 | 0 |
| 2 | RB_SPI_IE_FIFO_HF | RW | FIFO 使用过半中断使能位: 1: 使能中断; 0: 禁止中断。 | 0 |
| 1 | RB_SPI_IE_BYTE_END | RW | SPI 单字节传输完成中断使能位: 1: 使能中断; 0: 禁止中断。 | 0 |
| 0 | RB_SPI_IE_CNT_END | RW | SPI 全部字节传输完成中断使能位: 1: 使能中断; 0: 禁止中断。 | 0 |

SPI 主机模式时钟分频寄存器 (R8_SPIx_CLOCK_DIV) (x=0/1)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|------------------|----|---|-----|
| [7: 0] | R8_SPI_CLOCK_DIV | RW | 主机模式分频系数, 最小值为 2。 F _{sck} = 系统主频 F _{sys} / 分频系数。 | 10h |

SPI 从机模式预置数据寄存器 (R8_SPI0_SLAVE_PRE) (仅 SPI0 支持)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|-------------------|----|---------------------------------------|-----|
| [7: 0] | R8_SPI0_SLAVE_PRE | RW | 从机模式下, 预置的首个返回数据。 用于接收首字节数据后的返回数据。 | 10h |

SPI 数据缓冲区 (R8_SPIx_BUFFER) (x=0/1)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|----------------|----|-----------------|-----|
| [7: 0] | R8_SPIx_BUFFER | RW | SPI 数据发送和接收缓冲区。 | XXh |

SPI 工作状态寄存器 (R8_SPIx_RUN_FLAG) (x=0/1)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---|----|----|----|-----|
|---|----|----|----|-----|

| | | | | |
|-------|--------------------|----|---|-------|
| 7 | RB_SPI_SLV_SELECT | RO | 从机模式被片选状态位（仅 SPI0 支持）： 1：正被选中；0：没有被片选。 | 0 |
| 6 | RB_SPI_SLV_CS_LOAD | RO | 从机模式被片选后首次加载状态位（仅 SPI0 支持）： 1：正在加载 R8_SPI0_SLAVE_PRE； 0：尚未加载或者已完成。 | 0 |
| 5 | RB_SPI_FIFO_READY | RO | FIFO 准备就绪状态位： 1：FIFO 就绪（R16_SPIx_TOTAL_CNT 非 0，且接收时 FIFO 未满载或发送时 FIFO 不空）； 0：FIFO 未准备好。 | 0 |
| 4 | RB_SPI_SLV_CMD_ACT | RO | 从机模式下命令接收完成状态位，即交换完首字节数据（仅 SPI0 支持）： 1：指示刚刚交换完成的是首字节； 0：首字节尚未交换或不是首字节。 | 0 |
| [3:0] | Reserved | RO | 保留。 | 0000b |

SPI 中断标志寄存器 (R8_SPIx_INT_FLAG) (x=0/1)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---|--------------------|-----|---|-----|
| 7 | RB_SPI_IF_FST_BYTE | RW1 | 从机模式下，接收到首字节标志位（仅 SPI0 支持）： 1：已接收到首字节；0：未接收到。 | 0 |
| 6 | RB_SPI_FREE | RO | 当前 SPI 空闲状态位： 1：当前 SPI 空闲；0：当前 SPI 非空闲。 | 1 |
| 5 | Reserved | RO | 保留。 | 0 |
| 4 | RB_SPI_IF_FIFO_OV | RW1 | FIFO 溢出（接收时 FIFO 满或发送时 FIFO 空）标志位，写 1 清零： 1：FIFO 溢出；0：FIFO 未溢出。 | 0 |
| 3 | RB_SPI_IF_DMA_END | RW1 | DMA 完成标志位（仅 SPI0 支持），写 1 清零： 1：已完成；0：未完成。 | X |
| 2 | RB_SPI_IF_FIFO_HF | RW1 | FIFO 使用过半（接收时 FIFO>=4 或发送时 FIFO<4）标志位，写 1 清零： 1：FIFO 使用已过半； 0：FIFO 使用未过半。 | 0 |
| 1 | RB_SPI_IF_BYTE_END | RW1 | SPI 单字节传输完成标志位，写 1 清零： 1：单字节传输完成；0：传输未完成。 | 0 |
| 0 | RB_SPI_IF_CNT_END | RW1 | SPI 全部字节传输完成标志位，写 1 清零： 1：全部字节传输完成；0：传输未完成。 | 1 |

SPI 收发 FIFO 计数寄存器 (R8_SPIx_FIFO_COUNT) (x=0/1)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|--------------------|----|----------------|-----|
| [7:0] | R8_SPIx_FIFO_COUNT | RW | 当前 FIFO 中字节计数。 | 00h |

SPI 收发 FIFO 计数寄存器 (R8_SPIx_FIFO_COUNT1) (x=0/1)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|---------------------|----|----------------|-----|
| [7:0] | R8_SPIx_FIFO_COUNT1 | RW | 当前 FIFO 中字节计数。 | 00h |

| | | | | |
|--|--|--|-----------------------|--|
| | | | 同 R8_SPIx_FIFO_COUNT。 | |
|--|--|--|-----------------------|--|

SPI 收发数据总长度寄存器 (R16_SPIx_TOTAL_CNT) (x=0/1)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|--------------------|----|---|-------|
| [15: 0] | R16_SPIx_TOTAL_CNT | RW | 主机模式下 SPI 数据收发总字节数，低 12 位有效。在使用 DMA 时一次最多可以收发 4095 个字节。不支持从机模式。 | 0000h |

SPI 数据 FIFO 寄存器 (R8_SPIx_FIFO) (x=0/1)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|--------------|-----------|--------------|-----|
| [7: 0] | R8_SPIx_FIFO | RO/ WO | 数据 FIFO 寄存器。 | XXh |

寄存器 R8_SPIx_BUFFER 和 R8_SPIx_FIFO 均为 SPI 数据相关寄存器，主要区别在于：

读 R8_SPIx_BUFFER 是取自 SPI 最近一次交换到的数据，不影响 FIFO 和 R8_SPIx_FIFO_COUNT，主机模式下写 R8_SPIx_BUFFER 是直接发送该字节，从机模式下写操作未定义；

读 R8_SPIx_FIFO 是取自 FIFO 中最早交换到的数据，将减少 FIFO 和 R8_SPIx_FIFO_COUNT，

写 R8_SPIx_FIFO 是将数据暂存到 FIFO 中，在从机模式下由外部 SPI 主机决定何时取走，在主机模式下当 R16_SPIx_TOTAL_CNT 非 0 时自动启动发送。

SPI0 DMA 缓冲区当前地址 (R16_SPI0_DMA_NOW)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|------------------|----|--|-------|
| [15: 0] | R16_SPI0_DMA_NOW | RW | DMA 数据缓冲区当前地址。 可用于计算已转换次数，计算方法： COUNT=SPI0_DMA_NOW-SPI0_DMA_BEG。 | XXXXh |

SPI0 DMA 缓冲区起始地址 (R16_SPI0_DMA_BEG)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|------------------|----|--------------------------|-------|
| [15: 0] | R16_SPI0_DMA_BEG | RW | DMA 数据缓冲区起始地址，仅低 15 位有效。 | XXXXh |

SPI0 DMA 缓冲区结束地址 (R16_SPI0_DMA_END)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|------------------|----|------------------------------|-------|
| [15: 0] | R16_SPI0_DMA_END | RW | DMA 数据缓冲区结束地址（不含），仅低 15 位有效。 | XXXXh |

10.3 SPI 传输格式

SPI 支持模式 0 和模式 3 两种传输格式，通过设置 R8_SPIx_CTRL_MOD 的 RB_SPI_MST_SCK_MOD 进行选择。总是在 SCK 上升沿采样串行数据输入，在下降沿输出串行数据。

数据传输格式如下图所示：

模式 0: RB_SPI_MST_SCK_MOD = 0

模式0时序图

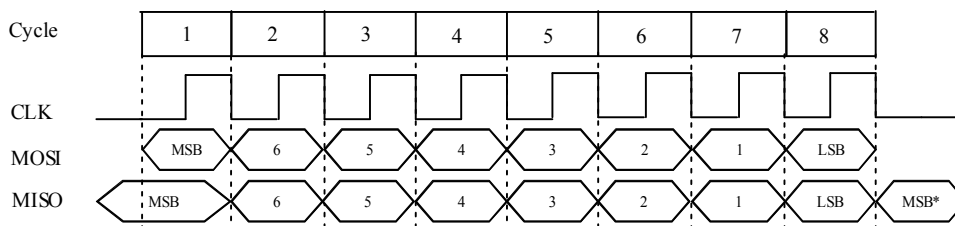


图 10-1 SPI 模式 0 传输格式

模式 3: RB_SPI_MST_SCK_MOD = 1

模式3时序图

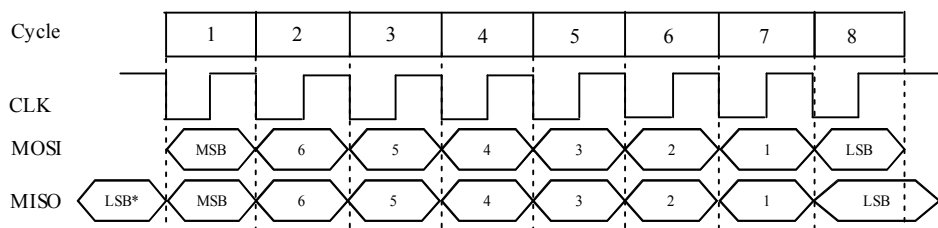


图 10-2 SPI 模式 3 传输格式

10.4 SPI 配置

10.4.1 SPI 主机模式

SPI 主机模式下，在 SCK 引脚产生串行时钟，片选引脚可以指定为任意 I/O 引脚。

配置步骤：

- (1)、设置 R8_SPIx_CLOCK_DIV，配置 SPI 时钟频率；
- (2)、设置 R8_SPIx_CTRL_MOD 的 RB_SPI_MODE_SLAVE 为 0，配置 SPI 为主机模式；
- (3)、设置 R8_SPIx_CTRL_MOD 的 RB_SPI_MST_SCK_MOD，选择时钟空闲模式 0 或模式 3；
- (4)、设置 R8_SPIx_CTRL_MOD 的 RB_SPI_FIFO_DIR 配置 FIFO 方向，为 1 则 FIFO 用于接收，为 0 则 FIFO 用于发送。
- (5)、设置 R8_SPIx_CTRL_MOD 的 RB_SPI_MOSI_OE 和 RB_SPI_SCK_OE 为 1，RB_SPI_MISO_OE 为 0，并设置 GPIO 方向配置寄存器 (R32_PA/PB_DIR) 使 MOSI 引脚和 SCK 引脚为输出，MISO 引脚为输入；
- (6)、2 线模式下 SCK 不变，RB_SPI_MOSI_OE=0，不用 MOSI，由 MISO 半双工实现输入（同 3 线模式，RB_SPI_MISO_OE=0 且引脚置为输入）和输出（RB_SPI_MISO_OE=1 且引脚置为输出），手工切换方向；
- (7)、可选步骤，如果启用 DMA，那么需将收发缓冲区起始地址写入 R16_SPI_DMA_BEG，结束地址（不含）写入 R16_SPI_DMA_END，建议在设置完 RB_SPI_FIFO_DIR 后再置位 RB_SPI_DMA_ENABLE，如果确认 R16_SPIx_TOTAL_CNT 为 0，那么也可在此先设置 RB_SPI_DMA_ENABLE 为 1，使能 DMA 功能。

数据发送过程：

- (1)、设置 RB_SPI_FIFO_DIR 为 0，当前 FIFO 方向为输出；
- (2)、写 R16_SPIx_TOTAL_CNT 寄存器，设置要发送的数据长度；
- (3)、写 R8_SPIx_FIFO 寄存器，向 FIFO 中写入要发送的数据，如果 R8_SPIx_FIFO_COUNT 小于 FIFO 容量则可以继续写 FIFO，如果是启用 DMA，则由 DMA 自动加载 FIFO 完成此步骤；
- (4)、只要 R16_SPIx_TOTAL_CNT 非 0 并且 FIFO 中有数据，SPI 主机就会自动发送数据，否则暂停；
- (5)、等待 R16_SPIx_TOTAL_CNT 寄存器为 0，说明数据发送完成，如果只发送一字节，也可以查询等待 RB_SPI_FREE 为空闲、或等待 R8_SPIx_FIFO_COUNT 为 0。

数据接收过程：

- (1)、设置 RB_SPI_FIFO_DIR 为 1，当前 FIFO 方向为输入；
- (2)、写 R16_SPIx_TOTAL_CNT 寄存器，设置要接收的数据长度；
- (3)、只要 R16_SPIx_TOTAL_CNT 非 0 并且 FIFO 未滿，SPI 主机就会自动接收数据，否则暂停；
- (4)、等待 R8_SPIx_FIFO_COUNT 寄存器不为 0，则说明接收到返回数据，读取 R8_SPI_FIFO 中的值即为接收到的数据，如果是启用 DMA，则由 DMA 自动读取 FIFO 完成此步骤。

10.4.2 SPI 从机模式

SPI0 支持从机模式，在从机模式下，SCK 引脚用于接收外部连接的 SPI 主机的串行时钟。

配置步骤：

- (1)、设置 R8_SPI0_CTRL_MOD 的 RB_SPI_MODE_SLAVE 为 1，配置 SPI0 为从机模式；
- (2)、根据需要设置 R8_SPI0_CTRL_MOD 的 RB_SPI_SLV_CMD_MOD，选择从机首字节模式或数据流模式；
- (3)、设置 R8_SPI0_CTRL_MOD 的 RB_SPI_FIFO_DIR，配置 FIFO 方向，为 1 则 FIFO 用于接收，为 0 则 FIFO 用于发送；
- (4)、设置 R8_SPI0_CTRL_MOD 的 RB_SPI_MOSI_OE 和 RB_SPI_SCK_OE 为 0，RB_SPI_MISO_OE 为 1，并设置 GPIO 方向配置寄存器 (R32_PA/PB_DIR) 使 MOSI 引脚、SCK 引脚和 SCS 引脚为输入，MISO 引脚为输入（支持总线下多个从机连接，被片选后 MISO 会自动切换为输出，也支持一主一从）或输出（仅用于一主一从对连）。在 SPI 从机模式下 MISO 的 I/O 引脚方向，除了能够由 GPIO 方向配置寄存器设为输出之外，还支持 SPI 片选有效期间自动切换为输出，但其输出数据由 RB_SPI_MISO_OE 选择，为 1 时输出 SPI 数据，为 0 时输出 GPIO 数据输出寄存器的数据。建议，设置 MISO 引脚为输入，使 MISO 在片选无效时不输出，便于多机操作时共享 SPI 总线；
- (5)、可选的，设置 SPI0 从机模式预置数据寄存器 R8_SPI0_SLAVE_PRE，用于被片选后首次自动加载到缓冲区中用于对外输出。当 8 个时钟过去之后（即首个数据字节在主从双方之间交换完毕），控制器得到外部 SPI 主机发来的首字节数据（命令码），外部 SPI 主机交换得到 R8_SPI0_SLAVE_PRE 中的预置数据（状态值）。R8_SPI0_SLAVE_PRE 的位 7 将在 SPI 片选有效后的 SCK 低电平期间自动加载到 MISO 引脚上，对于 SPI 模式 0（CLK 默认为低电平），如果预置了 R8_SPI0_SLAVE_PRE 的位 7，那么外部 SPI 主机将在 SPI 片选有效但尚未传输数据时，就能够通过查询 MISO 引脚得到 R8_SPI0_SLAVE_PRE 的位 7 的预置值，从而通过仅仅有效一下 SPI 片选就能快速获得 R8_SPI0_SLAVE_PRE 的位 7 的值（通常是向主机提供一个忙状态，便于主机快速查询）；
- (6)、可选步骤，如果启用 DMA，那么需将收发缓冲区起始地址写入 R16_SPI_DMA_BEG，结束地址（不含）写入 R16_SPI_DMA_END，必须在设置完 RB_SPI_FIFO_DIR 后才能置位 RB_SPI_DMA_ENABLE。

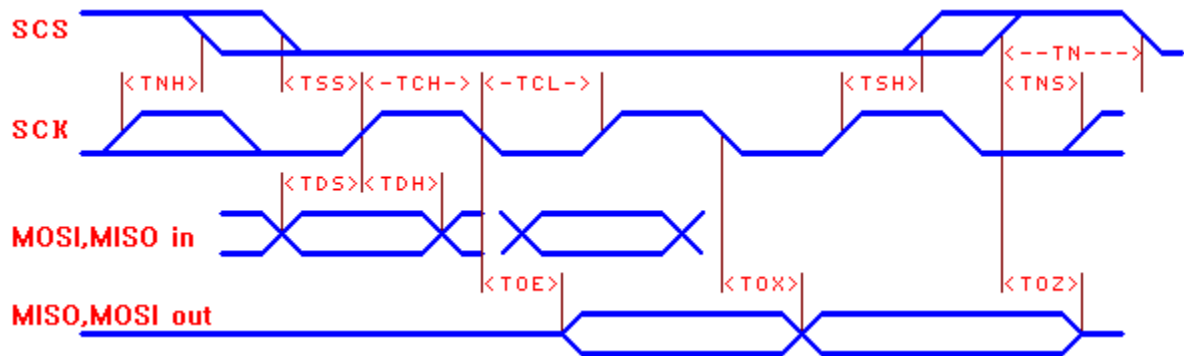
数据发送过程：

- (1)、设置 R8_SPI0_CTRL_MOD 的 RB_SPI_FIFO_DIR 为 0，当前 FIFO 方向为输出；
- (2)、可选步骤，如果启用 DMA，那么需设置 RB_SPI_DMA_ENABLE 为 1，使能 DMA 功能；
- (3)、将多个发送数据写入到 FIFO 寄存器 R8_SPI0_FIFO 中，由外部 SPI 主机决定何时取走，如果是启用 DMA，则由 DMA 自动加载 FIFO 完成此步骤；
- (4)、查询 R8_SPI0_FIFO_COUNT，如果未滿则继续向 FIFO 写入待发送的数据。

数据接收过程：

- (1)、设置 R8_SPI0_CTRL_MOD 的 RB_SPI_FIFO_DIR 为 1，当前 FIFO 方向为输入；
- (2)、可选步骤，如果启用 DMA，那么需设置 RB_SPI_DMA_ENABLE 为 1，使能 DMA 功能；
- (3)、查询 R8_SPI0_FIFO_COUNT，如果不空则说明已接收到数据，通过读取 R8_SPI0_FIFO 取走数据，如果是启用 DMA，则由 DMA 自动读取 FIFO 完成此步骤；
- (4)、单个字节的数据接收，也可以不使用 FIFO，可以直接读取 R8_SPI0_BUFFER。

10.5 SPI 时序



| 名称 | 参数说明 (TA=25°C, VIO33=3.3V) | 最小值 | 典型值 | 最大值 | 单位 |
|-----|-----------------------------|------------------|-----|-----|----|
| TSS | SCK 上升沿之前 SCS 有效的建立时间 | $T_{sys} * 1.05$ | | | nS |
| TSH | SCK 上升沿之后 SCS 有效的保持时间 | $T_{sys} * 1.05$ | | | nS |
| TNS | SCK 上升沿之前 SCS 无效的建立时间 | 15 | | | nS |
| TNH | SCK 上升沿之后 SCS 无效的保持时间 | 15 | | | nS |
| TN | SCS 无效时间 (SPI 操作间隔时间) | $T_{sys} * 2$ | | | nS |
| TCH | SCK 时钟的高电平时间 | $T_{sys} * 0.55$ | | | nS |
| TCL | SCK 时钟的低电平时间 | $T_{sys} * 0.55$ | | | nS |
| TDS | SCK 上升沿之前 MOSI/MISO 输入的建立时间 | 8 | | | nS |
| TDH | SCK 上升沿之后 MOSI/MISO 输入的保持时间 | 5 | | | nS |
| TOE | SCK 下降沿到 MISO/MOSI 输出有效 | 0 | | 18 | nS |
| TOX | SCK 下降沿到 MISO/MOSI 输出改变 | 0 | 5 | 16 | nS |
| TOZ | SCS 无效到 MISO/MOSI 输出无效 | 2 | | 24 | nS |

注: T_{sys} 是系统主频时钟周期 ($1/F_{sys}$)。

第 11 章 PWM

11.1 PWM 控制器简介

除了定时器提供的 4 路 26 位 PWM 输出之外，系统还提供了 8 路 8 位 PWM 输出（PWM4~PWM11），占空比可调，PWM 周期固定可选 8 种周期，操作简单。

11.2 寄存器描述

表 11-1 PWMx 相关寄存器列表

| 名称 | 访问地址 | 描述 | 复位值 |
|------------------|------------|----------------------|------------|
| R8_PWM_OUT_EN | 0x40005000 | PWMx 输出使能寄存器 | 0x00 |
| R8_PWM_POLAR | 0x40005001 | PWMx 输出极性配置寄存器 | 0x00 |
| R8_PWM_CONFIG | 0x40005002 | PWMx 配置控制寄存器 | 0x0X |
| R8_PWM_CLOCK_DIV | 0x40005003 | PWMx 时钟分频寄存器 | 0x00 |
| R32_PWM4_7_DATA | 0x40005004 | PWM4/5/6/7 数据保持寄存器 | 0xFFFFFFFF |
| R8_PWM4_DATA | 0x40005004 | PWM4 数据保持寄存器 | 0xXX |
| R8_PWM5_DATA | 0x40005005 | PWM5 数据保持寄存器 | 0xXX |
| R8_PWM6_DATA | 0x40005006 | PWM6 数据保持寄存器 | 0xXX |
| R8_PWM7_DATA | 0x40005007 | PWM7 数据保持寄存器 | 0xXX |
| R32_PWM8_11_DATA | 0x40005008 | PWM8/9/10/11 数据保持寄存器 | 0xFFFFFFFF |
| R8_PWM8_DATA | 0x40005008 | PWM8 数据保持寄存器 | 0xXX |
| R8_PWM9_DATA | 0x40005009 | PWM9 数据保持寄存器 | 0xXX |
| R8_PWM10_DATA | 0x4000500A | PWM10 数据保持寄存器 | 0xXX |
| R8_PWM11_DATA | 0x4000500B | PWM11 数据保持寄存器 | 0xXX |
| R8_PWM_INT_CTRL | 0x4000500C | PWMx 中断控制和状态寄存器 | 0x00 |

PWMx 输出使能寄存器 (R8_PWM_OUT_EN)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---|-----------------|----|-------------------------------|-----|
| 7 | RB_PWM11_OUT_EN | RW | PWM11 输出使能位： 1: 使能； 0: 禁止。 | 0 |
| 6 | RB_PWM10_OUT_EN | RW | PWM10 输出使能位： 1: 使能； 0: 禁止。 | 0 |
| 5 | RB_PWM9_OUT_EN | RW | PWM9 输出使能位： 1: 使能； 0: 禁止。 | 0 |
| 4 | RB_PWM8_OUT_EN | RW | PWM8 输出使能位： 1: 使能； 0: 禁止。 | 0 |
| 3 | RB_PWM7_OUT_EN | RW | PWM7 输出使能位： 1: 使能； 0: 禁止。 | 0 |
| 2 | RB_PWM6_OUT_EN | RW | PWM6 输出使能位： 1: 使能； 0: 禁止。 | 0 |
| 1 | RB_PWM5_OUT_EN | RW | PWM5 输出使能位： 1: 使能； 0: 禁止。 | 0 |
| 0 | RB_PWM4_OUT_EN | RW | PWM4 输出使能位： 1: 使能； 0: 禁止。 | 0 |

PWMx 输出极性配置寄存器 (R8_PWM_POLAR)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---|----------------|----|--|-----|
| 7 | RB_PWM11_POLAR | RW | PWM11 输出极性控制位： 1: 默认高电平，低有效； 0: 默认低电平，高有效。 | 0 |
| 6 | RB_PWM10_POLAR | RW | PWM10 输出极性控制位： 1: 默认高电平，低有效； 0: 默认低电平，高有效。 | 0 |
| 5 | RB_PWM9_POLAR | RW | PWM9 输出极性控制位： 1: 默认高电平，低有效； 0: 默认低电平，高有效。 | 0 |
| 4 | RB_PWM8_POLAR | RW | PWM8 输出极性控制位： 1: 默认高电平，低有效； 0: 默认低电平，高有效。 | 0 |
| 3 | RB_PWM7_POLAR | RW | PWM7 输出极性控制位： 1: 默认高电平，低有效； 0: 默认低电平，高有效。 | 0 |
| 2 | RB_PWM6_POLAR | RW | PWM6 输出极性控制位： 1: 默认高电平，低有效； 0: 默认低电平，高有效。 | 0 |
| 1 | RB_PWM5_POLAR | RW | PWM5 输出极性控制位： 1: 默认高电平，低有效； 0: 默认低电平，高有效。 | 0 |
| 0 | RB_PWM4_POLAR | RW | PWM4 输出极性控制位： 1: 默认高电平，低有效； 0: 默认低电平，高有效。 | 0 |

PWMx 配置控制寄存器 (R8_PWM_CONFIG)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|---------------------|----|--|-----|
| 7 | RB_PWM10_11_STAG_EN | RW | PWM10/11 交错输出使能位： 1: 交错输出； 0: 独立输出。 | 0 |
| 6 | RB_PWM8_9_STAG_EN | RW | PWM8/9 交错输出使能位： 1: 交错输出； 0: 独立输出。 | 0 |
| 5 | RB_PWM6_7_STAG_EN | RW | PWM6/7 交错输出使能位： 1: 交错输出； 0: 独立输出。 | 0 |
| 4 | RB_PWM4_5_STAG_EN | RW | PWM4/5 交错输出使能位： 1: 交错输出； 0: 独立输出。 | 0 |
| [3:2] | RB_PWM_CYC_MOD | RW | PWM 数据宽度选择： 00: 8 位数据宽度； 01: 7 位数据宽度； 10: 6 位数据宽度； 11: 5 位数据宽度。 | 00b |
| 1 | RB_PWM_STAG_ST | RO | PWM 交错标志位： 1: 指示 PWM5/7/9/11 被允许输出； 0: 指示 PWM4/6/8/10 被允许输出； | X |
| 0 | RB_PWM_CYCLE_SEL | RW | PWM 周期选择，和 PWM 数据宽度匹配： 1: 255/127/63/31 个时钟周期； 0: 256/128/64/32 个时钟周期。 | 0 |

PWMx 时钟分频寄存器 (R8_PWM_CLOCK_DIV)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---|----|----|----|-----|
|---|----|----|----|-----|

| | | | | |
|-------|------------------|----|--|-----|
| [7:0] | R8_PWM_CLOCK_DIV | RW | PWM 基准时钟分频系数。 $F_{pwm}=F_{sys}/R8_PWM_CLOCK_DIV$ 。 | 00h |
|-------|------------------|----|--|-----|

PWM 数据保持寄存器组 1 (R32_PWM4_7_DATA)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|--------------|----|---------------|-----|
| [31:24] | R8_PWM7_DATA | RW | PWM7 数据保持寄存器。 | XXb |
| [23:16] | R8_PWM6_DATA | RW | PWM6 数据保持寄存器。 | XXb |
| [15:8] | R8_PWM5_DATA | RW | PWM5 数据保持寄存器。 | XXb |
| [7:0] | R8_PWM4_DATA | RW | PWM4 数据保持寄存器。 | XXb |

PWM 数据保持寄存器组 2 (R32_PWM8_11_DATA)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|---------------|----|----------------|-----|
| [31:24] | R8_PWM11_DATA | RW | PWM11 数据保持寄存器。 | XXb |
| [23:16] | R8_PWM10_DATA | RW | PWM10 数据保持寄存器。 | XXb |
| [15:8] | R8_PWM9_DATA | RW | PWM9 数据保持寄存器。 | XXb |
| [7:0] | R8_PWM8_DATA | RW | PWM8 数据保持寄存器。 | XXb |

PWM_x 中断控制和状态寄存器 (R8_PWM_INT_CTRL)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|----------------|-----|--|--------|
| 7 | RB_PWM_IF_CYC | RW1 | PWM 周期结束标志位，写 1 清零： 1：周期结束；0：未结束。 | 0 |
| [6:2] | Reserved | R0 | 保留。 | 00000b |
| 1 | RB_PWM_CYC_PRE | RW | 选择 PWM 周期结束中断时间点： 1：提前 16 个计数产生中断（以 8 位数据宽度为例，计数到 240 时产生中断）； 0：提前 2 个计数产生中断（以 8 位数据宽度为例，计数到 254 时产生中断）。 | 0 |
| 0 | RB_PWM_IE_CYC | RW | PWM 周期结束中断使能位： 1：使能中断； 0：禁止中断。 | 0 |

11.3 PWM 配置

- 设置寄存器 R8_PWM_CLOCK_DIV，配置 PWM 的基准时钟频率；
- 设置 PWM 输出极性配置寄存器 R8_PWM_POLAR，配置对应 PWM_x 的输出极性；
- 设置 PWM 配置控制寄存器 R8_PWM_CONFIG，设置 PWM 的模式、数据位宽、周期；
- 设置 PWM 输出使能寄存器 R8_PWM_OUT_EN，开启对应的 PWM_x 输出使能；
- 根据需要的占空比计算出数据，写入对应的数据保持寄存器 R8_PWM_x_DATA；
- 设置 PWM4-PWM11 中所需的 PWM 引脚方向为输出，可选地，设置相应 I/O 的驱动能力；
- 根据需要更新 R8_PWM_x_DATA 中的数据，更新输出占空比。

调整 R8_PWM_x_DATA 寄存器可修改输出 PWM 的占空比，占空比计算公式：

$$N_{cyc} = RB_PWM_CYCLE_SEL ? (2^{n-1}) : (2^n)$$

（其中 n=数据位宽），N_{cyc} 结果在 63~256 之间。

$$PWM_x \text{ 输出有效电平的占空比} = R8_PWM_x_DATA / N_{cyc} * 100\%$$

$$PWM_x \text{ 输出频率 } F_{pwmout} = F_{pwm} / N_{cyc} = F_{sys}/R8_PWM_CLOCK_DIV / N_{cyc}$$

如果需要通过 PWM 产生直流信号，那么可以 PWM_x 输出端使用 R/C 等电路滤波，建议用时间常数远大于 4 / F_{pwmout} 的两级 RC，或时间常数远大于 100 / F_{pwmout} 的一级 RC。

第 12 章 两线串行接口 I2C

12.1 I2C 简介

I2C 是一种中低速串行总线，总线上可以连接有多个主机和若干从机。通常 I2C 接口由 2 个引脚组成：串行时钟引脚 SCL、串行数据引脚 SDA。

12.1.1 主要特性

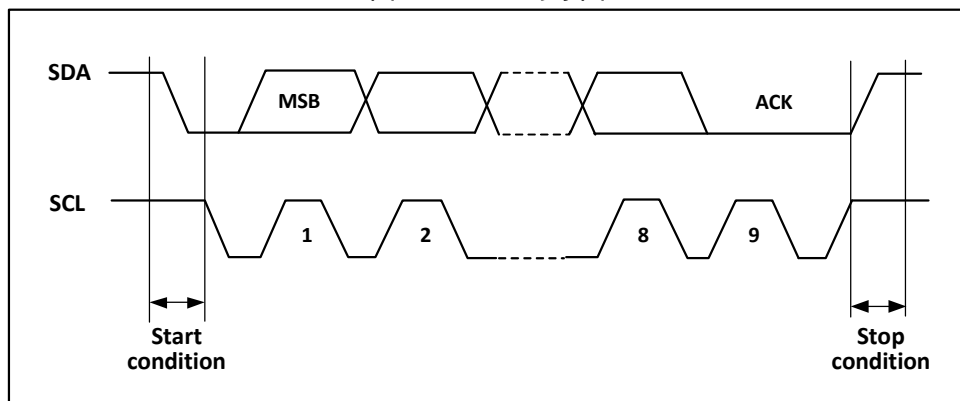
CH583 和 CH582 芯片提供 I2C 接口，特性如下：

- 支持主机模式（Master）和从机模式（Slave），支持多主多从。
- 支持两种速度模式：100KHz 和 400KHz，兼容 I²C 两线串行总线规范。
- 支持 7 位或 10 位地址。
- 从设备支持双 7 位地址。
- 支持总线广播。
- 支持总线仲裁、错误检测、PEC 校验、延长时钟。
- 兼容 SMBus。

12.2 I2C 概述

I2C 是个半双工的总线，它同时只能运行在下列四种模式之一：主设备发送模式、主设备接收模式、从设备发送模式和从设备接收模式。I2C 模块默认工作在从模式，在产生起始条件后，会自动地切换到主模式，当仲裁丢失或者产生停止信号后，会切换到从模式。I2C 模块支持多主机功能。工作在主模式时，I2C 模块会主动发出数据和地址。数据和地址都以 8 位为单位进行传输，高位在前，低位在后，在起始事件后的是一个字节（7 位地址模式下）或两个字节（10 位地址模式下）地址，主机每发送 8 位数据或地址，从机需要回复一个应答 ACK，即把 SDA 总线拉低，如图 12-1 所示。

图 12-1 I2C 时序图



12.3 主模式

主模式时，I2C 模块主导数据传输并输出时钟信号，数据传输以开始事件开始，以结束事件结束。使用主模式通讯的步骤为：

- 1) 在控制寄存器 2 (R16_I2C_CTRL2) 和时钟控制寄存器 (R16_I2C_CKCFGR) 中设置正确的时钟；
- 2) 在上升沿寄存器 (R16_I2C_RTR) 设置合适的上升沿；
- 3) 在控制寄存器 (R16_I2C_CTRL1) 中置 PE 位启动外设；
- 4) 在控制寄存器 (R16_I2C_CTRL1) 中置 START 位，产生起始事件。

在置 START 位后，I2C 模块会自动切换到主模式，MSL 位会置位，产生起始事件，在产生起始事件后，SB 位会置位，如果 ITEVTEN 位（在 R16_I2C_CTRL2）被置位，则会产生中断。此时应该读取状

态寄存器 1 (R16_I2C_STAR1)，写从地址到数据寄存器后，SB 位会自动清除；

5) 如果是使用 10 位地址模式，那么写数据寄存器发送头序列（头序列为 11110xx0b，其中的 xx 位是 10 位地址的最高两位）。

在发送完头序列之后，状态寄存器的 ADD10 位会被置位，如果 ITEVTEN 位已经置位，则会产生中断，此时应读取 R16_I2C_STAR1 寄存器后，写第二个地址字节到数据寄存器后，清除 ADD10 位。

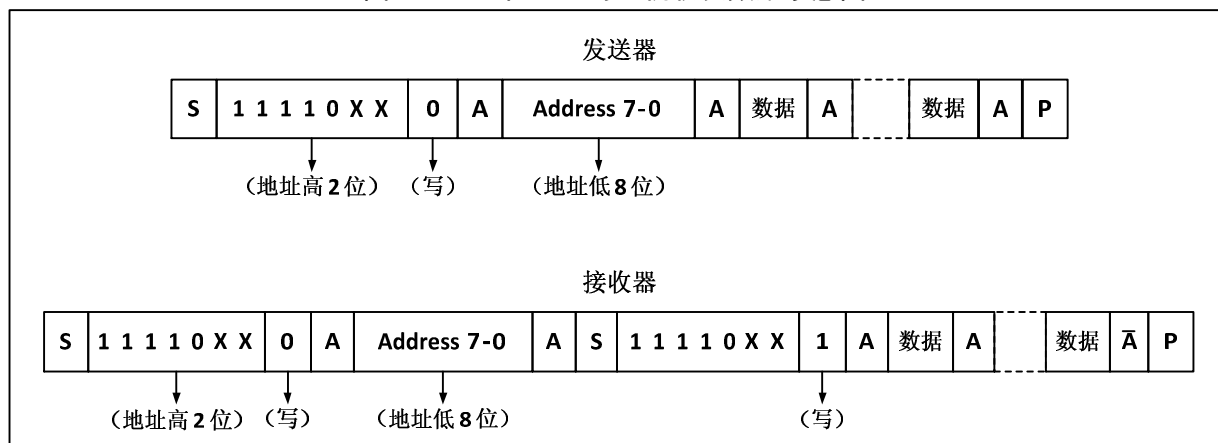
然后写数据寄存器发送第二个地址字节，在发送完第二个地址字节后，状态寄存器的 ADDR 位会被置位，如果 ITEVTEN 位已经置位，则会产生中断，此时应读取 R16_I2C_STAR1 寄存器后再读一次 R16_I2C_STAR2 寄存器以清除 ADDR 位；

如果使用的是 7 位地址模式，那么写数据寄存器发送地址字节，在发送完地址字节后，状态寄存器的 ADDR 位会被置位，如果 ITEVTEN 位已经置位，则会产生中断，此时应读取 R16_I2C_STAR1 寄存器后再读一次 R16_I2C_STAR2 寄存器以清除 ADDR 位；

在 7 位地址模式下，发送的第一个字节为地址字节，头 7 位代表的是目标从设备地址，第 8 位决定了后续报文的方向，0 代表是主设备写入数据到从设备，1 代表是主设备向从设备读取信息。

在 10 位地址模式下，如图 12-3 所示，在发送地址阶段，第一个字节为 11110xx0，xx 为 10 位地址的最高 2 位，第二个字节为 10 位地址的低 8 位。若后续进入主设备发送模式，则继续发送数据；若后续准备进入主设备接收模式，则需要重新发送一个起始条件，跟随发送一个字节为 11110xx1，然后进入主设备接收模式。

图 12-3 10 位地址时主机收发数据示意图



6) 发送模式时，主设备内部的移位寄存器将数据从数据寄存器发送到 SDA 线上，当主设备接收到 ACK 时，状态寄存器 1 (R16_I2C_STAR1) 的 TxE 被置位，如果 ITEVTEN 和 ITBUFEN 被置位，还会产生中断。向数据寄存器写入数据将会清除 TxE 位。

如果 TxE 位被置位且上次发送数据之前没有新的数据被写入数据寄存器，那么 BTF 位会被置位，在其被清除之前，SCL 将保持低电平，读 R16_I2C_STAR1 后，向数据寄存器写入数据将会清除 BTF 位。

而在接收模式时，I2C 模块会从 SDA 线接收数据，通过移位寄存器写进数据寄存器。在每个字节之后，如果 ACK 位被置位，那么 I2C 模块将会发出一个应答低电平，同时 RxNE 位会被置位，如果 ITEVTEN 和 ITBUFEN 被置位，还会产生中断。如果 RxNE 被置位且在新的数据被接收前，原有的数据没有被读出，则 BTF 位将被置位，在清除 BTF 之前，SCL 将保持低电平，读取 R16_I2C_STAR1 后，再读取数据寄存器将会清除 BTF 位。

7) 主设备在结束发送数据时，会主动发一个结束事件，即置 STOP 位。在接收模式时，主设备需要在最后一个数据位的应答位置 NAK。注意，产生 NAK 后，I2C 模块将会切换至从模式。

12.4 从模式

从模式时，I2C 模块能识别它自己的地址和广播呼叫地址。软件能控制开启或禁止广播呼叫地址的识别。一旦检测到起始事件，I2C 模块将 SDA 的数据通过移位寄存器与自己的地址（位数取决于 ENDUAL 和 ADDMODE）或广播地址（ENGC 置位时）相比较，如果不匹配将会忽略，直到产生新的起始事件。如果与头序列相匹配，则会产生一个 ACK 信号并等待第二个字节的地址；如果第二个字节的地址

也匹配或者 7 位地址情况下全段地址匹配,那么:首先产生一个 ACK 应答;ADDR 位被置位,如果 ITEVTEN 位已经置位,那么还会产生相应的中断;如果使用的是双地址模式 (ENDUAL 位被置位),还需要读取 DUALF 位来判断主机唤起的是哪一个地址。

从模式默认是接收模式,在接收的头序列的最后一位为 1,或者 7 位地址最后一位为 1 时(取决于第一次接收到头序列还是普通的 7 位地址),I2C 模块将进入到发送器模式,TRA 位将指示当前是接收器还是发送器模式。

发送模式时,在清除 ADDR 位后,I2C 模块将字节从数据寄存器通过移位寄存器发送到 SDA 线上。在收到一个应答 ACK 后,TxE 位将被置位,如果设置了 ITEVTEN 和 ITBUFEN,还会产生一个中断。如果 TxE 被置位但在下一个数据发送结束前没有新的数据被写入数据寄存器时,BTF 位将被置位。在清除 BTF 前,SCL 将保持低电平,读取状态寄存器 1 (R16_I2C_STAR1)后,再向数据寄存器写入数据将会清除 BTF 位。

接收模式时,在 ADDR 被清除后,I2C 模块将 SDA 上的数据通过移位寄存器存进数据寄存器,在每接收到一个字节后,I2C 模块都会置一个 ACK 位,并置 RxNE 位,如果设置了 ITEVTEN 和 ITBUFEN,还会产生一个中断。如果 RxNE 被置位,且在接收到新的数据前旧的数据没有被读出,那么 BTF 会被置位。在清除 BTF 位之前 SCL 会保持低电平。读取状态寄存器 1 (R16_I2C_STAR1)并读取数据寄存器里的数据会清除 BTF 位。

当 I2C 模块检测到停止事件时,将置 STOPF 位,如果设置了 ITEVFEN 位,还会产生一个中断。用户需要读取状态寄存器 (R16_I2C_STAR1)再写控制寄存器(比如复位控制字 SWRST)来清除。

12.5 错误

12.5.1 总线错误 BERR

在传输地址或者数据期间,I2C 模块检测到外部的起始或者停止事件时,将产生一个总线错误。产生总线错误时,BERR 位被置位,如果设置了 ITERREN 还会产生一个中断。在从模式下,数据被丢弃,硬件释放总线。如果是起始信号,硬件会认为是重启信号,开始等待地址或停止信号;如果是停止信号,则提前按正常的停止条件操作。在主模式下,硬件不会释放总线,同时不影响当前传输,由用户代码决定是否中止传输。

12.5.2 应答错误 AF

当 I2C 模块检测到一个字节后没有应答时,会产生应答错误。产生应答错误时:AF 会被置位,如果设置了 ITERREN 还会产生一个中断;遇到 AF 错误,如果 I2C 模块工作在从模式,硬件必须释放总线,如果处于主模式,软件必须生成一个停止事件。

12.5.3 仲裁丢失 ARLO

当 I2C 模块检测到仲裁丢失时,产生仲裁丢失错误。产生仲裁丢失错误时:ARLO 位被置位,如果设置了 ITERREN 还会产生一个中断;I2C 模块切换到从模式,并不再响应针对它的从地址发起的传输,除非有主机发起新的起始事件;硬件会释放总线。

12.5.4 过载/欠载错误 OVR

1) 过载错误:

在从机模式下,如果禁止时钟延长,I2C 模块正在接收数据,如果已经接受到一个字节的数据,但是上一次接收到数据还没有被读出,则会产生过载错误。发生过载错误时,最后收到的字节将被丢弃,发送方应当重发最后一次发送的字节。

2) 欠载错误:

在从模式下,如果禁止时钟延长,I2C 模块正在发送数据,如果在下一个字节的时钟到来之前新

的数据还没有被写入到数据寄存器，那么将产生欠载错误。在发生欠载错误时，前一次数据寄存器里的数据将被发送两次，如果发生欠载错误，那么接收方应该丢弃重复收到的数据。为了不产生欠载错误，I2C 模块应当在下一个字节的第一个上升沿之前将数据写入数据寄存器。

12.6 时钟延长

如果禁止时钟延长，那么就存在发生过载/欠载错误的可能。但如果使能了时钟延长：

- 1) 在发送模式下，如果 TxE 置位且 BTF 置位，SCL 将一直为低，一直等待用户读取状态寄存器，并向数据寄存器写入待发送的数据；
 - 2) 在接收模式下，如果 RxNE 置位且 BTF 置位，那么 SCL 在接收到数据后将保持低，直到用户读取状态寄存器，并读取数据寄存器；
- 由此可见，使能时钟延长可以避免出现过载/欠载错误。

12.7 SMBus

SMBus 也是一种两线接口，一般应用于在系统和电源管理之间。SMBus 和 I2C 有很多相似的地方，例如 SMBus 使用和 I2C 一样的 7 位地址模式，以下是他们的共同点：

- 1) 主从通信模式，主机提供时钟，支持多主多从；
- 2) 两线通讯结构；
- 3) 支持 7 位地址格式。

区别：

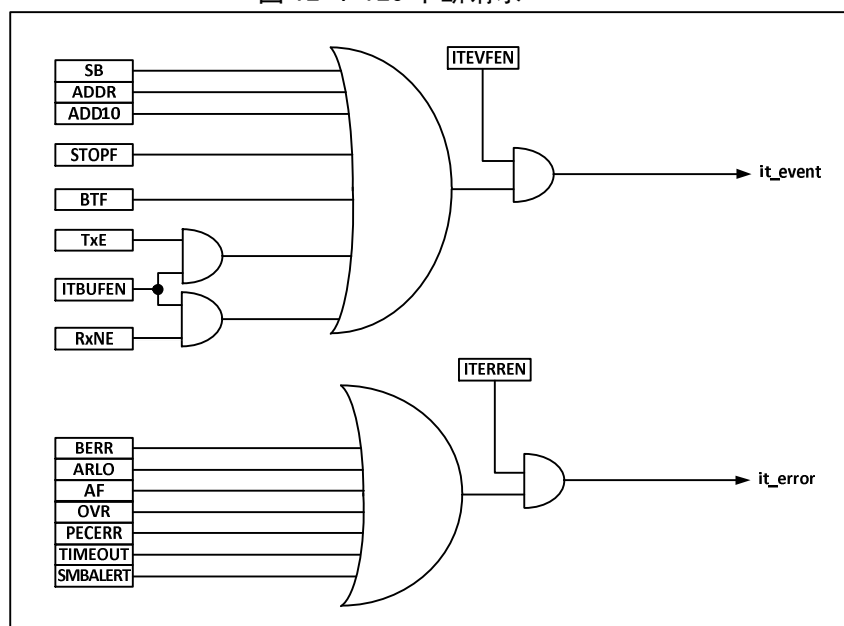
- 1) I2C 支持的速度可达 400KHz，SMBus 速度仅达 100KHz，且 SMBus 有最小 10KHz 的速度限制；
- 2) SMBus 的时钟为低超过 35mS 时，会报超时，但 I2C 无此限制；
- 3) SMBus 有固定的逻辑电平，而 I2C 没有，取决于芯片电源电压 VDD；
- 4) SMBus 有总线协议，而 I2C 没有。

SMBus 还包括设备识别、地址解析协议、唯一的设备标识符、SMBus 提醒和各种总线协议，具体请参考 SMBus 规范 2.0 版本。当使用 SMBus 时，只需要置控制寄存器的 SMBus 位，按需配置 SMBTYPE 位和 ENAARP 位。

12.8 中断

I2C 模块提供事件中断和错误中断，进入同一个中断服务程序，经查询后区分处理。

图 12-4 I2C 中断请求



12.9 包校验错误 PEC

包错误校验(PEC)是为了提供传输的可靠性而增加一项 CRC8 校验的步骤,使用以下多项式对每一位串行数据进行计算: $C=X^8+X^2+X+1$ 。

PEC 计算是由控制寄存器的 ENPEC 位激活,对所有信息字节进行计算,包括地址和读写位在内。在发送时,启用 PEC 会在最后一字节数据之后加上一个字节的 CRC8 计算结果;而在接收模式,在最后一字节被认为是 CRC8 校验结果,如果和内部的计算结果不符合,就会回复一个 NAK,如果是主接收器,无论校验结果正确与否,都会回复一个 NAK。

12.10 寄存器描述

表 12-1 I2C 相关寄存器列表

| 名称 | 访问地址 | 描述 | 复位值 |
|----------------|------------|-------------|--------|
| R16_I2C_CTRL1 | 0x40004800 | I2C 控制寄存器 1 | 0x0000 |
| R16_I2C_CTRL2 | 0x40004804 | I2C 控制寄存器 2 | 0x0000 |
| R16_I2C_OADDR1 | 0x40004808 | I2C 地址寄存器 1 | 0x0000 |
| R16_I2C_OADDR2 | 0x4000480C | I2C 地址寄存器 2 | 0x0000 |
| R16_I2C_DATAR | 0x40004810 | I2C 数据寄存器 | 0x0000 |
| R16_I2C_STAR1 | 0x40004814 | I2C 状态寄存器 1 | 0x0000 |
| R16_I2C_STAR2 | 0x40004818 | I2C 状态寄存器 2 | 0x0000 |
| R16_I2C_CKCFGR | 0x4000481C | I2C 时钟寄存器 | 0x0000 |
| R16_I2C_RTR | 0x40004820 | I2C 上升时间寄存器 | 0x0002 |

I2C 控制寄存器 (I2C_CTRL1)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|----|----------|----|---|-----|
| 15 | SWRST | RW | 软件重置,用户代码置此位会使 IIC 外设重置。在复位前确定 IIC 总线的引脚被释放,总线处于空闲状态。 注:该位可以在总线上没有检测到停止条件但是 busy 位为 1 时,重置 IIC 模块。 | 0 |
| 14 | Reserved | RO | 保留。 | 0 |
| 13 | ALERT | RW | SMBus 提醒位,用户代码可以设置此位或者清除此位;当 PE 置位后,此位可以被硬件清除。 1: 驱动 SMBusALERT 引脚使其变低,响应地址头应紧跟在 ACK 信号后面; 0: 释放 SMBusALERT 引脚使其变高,响应地址头应紧跟在 NACK 信号后面。 | 0 |
| 12 | PEC | RW | 数据包出错检测使能位,置此位启用数据包出错检测。用户代码可以对此位置位或清零;当 PEC 被传输后,或产生开始或结束信号,或者 PE 位清 0 时,硬件清零该位; 1: 带 PEC; 0: 不带 PEC。 注:仲裁丢失时,PEC 失效。 | 0 |
| 11 | POS | RW | ACK 和 PEC 位置设置位,该位可以被用户代码置位或者清零,在 PE 被清零后,可以被硬件清除; 1: ACK 位控制在移位寄存器里接收的下一个字节的 ACK 或者 NAK。PEC 移位寄存器里接收的下一字节是 PEC; | 0 |

| | | | | |
|----|-----------|----|---|---|
| | | | <p>0: ACK 位控制当前移位寄存器内正在接受的字节的 ACK 或者 NAK。PEC 位表明当前移位寄存器的字节是 PEC。</p> <p>注: POS 位在 2 字节数据接收中的用法如下: 必须在接收之前配置好。为了 NACK 第 2 个字节, 必须在清除 ADDR 位后立刻清除 ACK 位; 为了检测第二个字节的 PEC, 必须在 ADDR 事件发生后, 配置 POS 位后设置 PEC 位。</p> | |
| 10 | ACK | RW | <p>应答使能位, 该位可以被用户代码置位或者清零, 当 PE 位被置位时, 该位可以被硬件清除;</p> <p>1: 在接收到一个字节后返回一个应答;</p> <p>0: 不设应答。</p> | 0 |
| 9 | STOP | RW | <p>停止事件产生位, 该位可以被用户代码置位或清零, 或当检测到停止事件时, 由硬件清除, 或检测到超时错误时, 由硬件将其置位。</p> <p>主模式下:</p> <p>1: 在当前字节传输或者当前起始条件发出后产生停止事件;</p> <p>0: 无停止事件产生。</p> <p>从模式下:</p> <p>1: 在当前字节传输后释放 SCL 和 SDA 线;</p> <p>0: 无停止事件产生。</p> | 0 |
| 8 | START | RW | <p>起始事件产生位, 该位可以被用户代码置位或者清零, 当起始条件发出后或者 PE 被清零时, 由硬件清零。</p> <p>主模式下:</p> <p>1: 重复产生起始事件;</p> <p>0: 无起始事件产生。</p> <p>从模式下:</p> <p>1: 当总线空闲时, 产生起始事件;</p> <p>0: 无起始事件产生。</p> | 0 |
| 7 | NOSTRETCH | RW | <p>禁止时钟延长位, 此位用于在 ADDB 或 BTF 标志被置位的情况下, 禁止从模式下的时钟延长, 直至被软件清零。</p> <p>1: 禁止时钟延长; 0: 允许时钟延长。</p> | 0 |
| 6 | ENGCG | RW | <p>广播呼叫使能位, 置此位使能广播呼叫, 应答广播地址 00h。</p> | 0 |
| 5 | ENPEC | RW | <p>PEC 使能位, 置此位开启 PEC 计算。</p> | 0 |
| 4 | ENARP | RW | <p>ARP 使能位, 置此位使能 ARP。</p> <p>如果 SMBTYPE=0, 则使用 SMBus 设备的默认地址; 如果 SMBTYPE=1, 则使用 SMBus 的主地址。</p> | 0 |
| 3 | SMBTYPE | RW | <p>SMBus 设备类型, 置 1 为 SMBus 主设备, 置 0 为 SMBus 从设备。</p> | 0 |
| 2 | Reserved | RO | 保留。 | 0 |
| 1 | SMBUS | RW | <p>SMBus 模式选择位, 置 1 为使用 SMBus 模式, 置 0 为使用 IIC 模式。</p> | 0 |
| 0 | PE | RW | <p>IIC 外设使能位。</p> <p>1: 启用 IIC 模块; 0: 禁用 IIC 模块。</p> | 0 |

I2C 控制寄存器 2 (I2C_CTRL2)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|----------|----|--|-----|
| [15:11] | Reserved | RO | 保留。 | 0 |
| 10 | ITBUFEN | RW | 缓冲器中断使能位。 1: 当 TxE 或者 RxEN 被置位时, 产生事件中断; 0: 当 TxE 或者 RxEN 被置位时, 不产生中断。 | 0 |
| 9 | ITEVTEN | RW | 事件中断使能位, 置此位使能事件中断。 在下列条件下, 将产生此中断: SB=1 (主模式); ADDR=1 (主从模式); ADDR10=1 (主模式); STOPF=1 (从模式); BTF=1, 但是没有 TxE 或者 RxEN 事件; 如果 ITBUFEN=1, TxE 事件为 1; 如果 ITBUFEN=1, RxNE 事件为 1。 | 0 |
| 8 | ITERREN | RW | 出错中断使能位, 置位表示允许出错中断。 在下列条件下, 将产生该中断: BERR=1; ARLO=1; AF=1; OVR=1; PECERR=1; TIMEOUT=1; SMBAlert=1。 | 0 |
| [7:6] | Reserved | RO | 保留。 | 0 |
| [5:0] | FREQ | RW | IIC 模块时钟频率域, 必须输入正确的时钟频率以产生正确的时序, 允许的范围在 2~36MHz 之间。必须设置在 000010b 到 100100b 之间, 单位为 MHz。 建议: 标准模式下, 输入时钟最低为 2MHz; 快速模式下, 输入时钟最低为 4MHz。 | 0 |

I2C 地址寄存器 1 (I2C_OADDR1)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|----------|-----|---|-----|
| 15 | ADDMODE | RW | 地址模式。 1: 10 位从机地址 (不响应 7 位地址); 0: 7 位从机地址 (不响应 10 位地址)。 | 0 |
| 14 | MUST1 | RW1 | 必须始终由软件写 1。 | 0 |
| [13:10] | Reserved | RO | 保留。 | 0 |
| [9:8] | ADD9_8 | RW | 总线地址, 在使用 10 位地址时为第 9-8 位, 在使用 7 位地址时忽略。 | 0 |
| [7:1] | ADD7_1 | RW | 总线地址, 第 7-1 位。 | 0 |
| 0 | ADD0 | RW | 总线地址, 使用 10 位地址时为第 0 位, 在使用 7 位地址时忽略。 | 0 |

I2C 地址寄存器 2 (I2C_OADDR2)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|----------|----|------------------------------|-----|
| [15:8] | Reserved | RO | 保留。 | 0 |
| [7:1] | ADD2 | RW | 总线地址, 双地址模式下地址的 7-1 位。 | 0 |
| 0 | ENDUAL | RW | 双地址模式使能位, 置此位可以让 ADD2 也能被识别。 | 0 |

I2C 数据寄存器 (I2C_DATAR)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|----------|----|----------------------------------|-----|
| [15:8] | Reserved | RO | 保留。 | 0 |
| [7:0] | DATAR | RW | 数据寄存器，用来存放接收到的数据或者存放用于发送到总线的的数据。 | 0 |

I2C 状态寄存器 1 (I2C_STAR1)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|----|----------|-----|---|-----|
| 15 | SMBALERT | RWO | SMBus 警示位，该位可以由用户写 0 复位，或在 PE 变低时由硬件复位。 在 SMBus 主机模式下： 1: 产生了 SMBus 警示； 0: 无 SMBus 警示。 在 SMBus 从机模式下： 1: 收到 SMBAlert 响应地址头序列直到 SMBAlert 变低； 0: 没有收到 SMBAlert 响应地址头序列。 | 0 |
| 14 | TIMEOUT | RWO | 超时或者 Tlow 错误标志位，该位可以由用户写 0 复位，或在 PE 变低时由硬件复位。 1: SCL 处于低已达到 25mS，或者主机低电平累计时钟扩招时间超过 10mS，或者从设备低电平累计时间超过 25mS； 0: 无超时错误。 注：在从模式下此位被置位，从设备会复位通讯，硬件会释放总线；在主模式下此位被置位，硬件会发出停止条件。 | 0 |
| 13 | Reserved | RO | 保留。 | 0 |
| 12 | PECERR | RWO | 在接收时发生 PEC 错误标志位，该位可以由用户写 0 复位，或在 PE 变低时由硬件复位。 1: 有 PEC 错误，接收到 PEC 后，返回 NAK； 0: 无 PEC 错误。 | 0 |
| 11 | OVR | RWO | 过载、欠载标志位。 1: 有过载、欠载事件发生：当 NOSTRETCH=1 时，在接收模式中收到一个新的字节时，数据寄存器里的内容还未被读出，则新接收的字节将丢失；在发送模式时，没有新的数据写入数据寄存器，同样的字节将被发送两次； 0: 无过载、欠载事件。 | 0 |
| 10 | AF | RWO | 应答失败标志位，该位可以由用户写 0 复位，或在 PE 变低时由硬件复位。 1: 应答错误； 0: 应答正常。 | 0 |
| 9 | ARLO | RWO | 仲裁丢失标志位，该位可以由用户写 0 复位，或在 PE 变低时由硬件复位。 1: 检测到仲裁丢失，模块失去对总线的控制； 0: 仲裁正常。 | 0 |
| 8 | BERR | RWO | 总线出错标志位，该位可以由用户写 0 复位，或在 PE 变低时由硬件复位。 1: 起始或者停止条件出错； 0: 正常。 | 0 |
| 7 | TxE | RO | 数据寄存器为空标志位，向数据寄存器写数据 | 0 |

| | | | | |
|---|----------|-----|---|---|
| | | | 可以清除，或者产生一个起始或者停止位后，或当 PE 为 0 后，由硬件自动清除。 1: 发送数据时，发送数据寄存器为空； 0: 数据寄存器非空。 | |
| 6 | RxNE | RO | 数据寄存器非空标志位，对数据寄存器的读写操作将清除此位，或者当 PE 为 0 后，由硬件清除此位。 1: 接收数据时，数据寄存器不为空； 0: 正常。 | 0 |
| 5 | Reserved | RO | 保留。 | 0 |
| 4 | STOPF | RO | 停止事件标志位，用户读取状态寄存器 1 之后，对控制寄存器 1 的写操作将会清除该位，或者当 PE 为 0 后，由硬件清除此位。 1: 在应答之后，从设备在总线上检测到停止事件； 0: 没有检测到停止事件。 | 0 |
| 3 | ADD10 | RO | 10 位地址头序列发送标志位，用户读取状态寄存器 1 之后，对控制寄存器 1 的写操作将会清除该位，或者当 PE 为 0 后，由硬件清除此位。 1: 在 10 位地址模式下，主设备已经将第一个地址字节发送出去； 0: 无。 | 0 |
| 2 | BTF | RO | 字节发送结束标志位，用户读取状态寄存器 1 后，对数据寄存器的读写将清除此位；在传输中，发起一个起始或者停止事件后，或当 PE 为 0 后，由硬件清除此位。 1: 字节发送结束。当 NOSTRETCH=0 时：发送时，当一个新数据被发送且数据寄存器还未被写入新数据；接收时，当接收一个新的字节但是数据寄存器还未被读取； 0: 无。 | 0 |
| 1 | ADDR | RWO | 地址被发送/地址匹配标志位，用户读取状态寄存器 1 后，对状态寄存器 2 的读操作将会清除此位，或当 PE 为 0 时，由硬件清除此位。 主模式： 1: 地址发送结束：在 10 位地址模式下，当收到地址的第二个字节的 ACK 后改为被置位；在 7 位地址模式下，当收到地址的 ACK 后被置位； 0: 地址发送没有结束。 从模式： 1: 收到的地址匹配； 0: 地址不匹配或者没有收到地址。 | 0 |
| 0 | SB | RO | 起始位发送标志位，读取状态寄存器 1 后写数据寄存器的操作将清除该位，或当 PE 为 0 时，硬件将会清除此位。 1: 已发送起始位； 0: 未发送起始位。 | 0 |

I2C 状态寄存器 2 (I2C_STAR2)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---|----|----|----|-----|
|---|----|----|----|-----|

| | | | | |
|--------|------------|----|--|---|
| [15:8] | PEC | RO | 包错误检查域，当 PEC 使能时（ENPEC 置位），此域存放 PEC 的值。 | 0 |
| 7 | DUALF | RO | 匹配检测标志位，在产生停止位或起始位时，或者在 PE=0 时，硬件会将该位清零。 1: 接收到的地址与 OADDR2 中的内容相符； 0: 接收到的地址与 OADDR1 中的内容相符。 | 0 |
| 6 | SMBHOST | RO | SMBus 主机头标志位，在产生停止位或起始位时，或者在 PE=0 时，硬件会将该位清零。 1: 当 SMBTYPE=1 且 ENARP=1 时，收到了 SMBus 主机地址； 0: 未接收到 SMBus 主机地址。 | 0 |
| 5 | SMBDEFAULT | RO | SMBus 设备默认地址标志位，在产生停止位或起始位时，或者在 PE=0 时，硬件会将该位清零。 1: 当 ENARP=1，收到 SMBus 设备的默认地址； 0: 未收到地址。 | 0 |
| 4 | GENCALL | RO | 广播呼叫地址标志位，在产生停止位或起始位时，或者在 PE=0 时，硬件会将该位清零。 1: 当 ENGC=1 时，收到广播呼叫的地址； 0: 未收到广播呼叫地址。 | 0 |
| 3 | Reserved | RO | 保留。 | 0 |
| 2 | TRA | RO | 发送/接收标志位，在检测到停止事件（STOPF=1），重复的起始条件或者总线仲裁丢失（ARLO=1）或者 PE=0 时，硬件会将其清零。 1: 数据已发送； 0: 接收了数据。 该位根据地址字节的 R/W 位来决定。 | 0 |
| 1 | BUSY | RO | 总线忙标志位，该位在检测到一个停止位时会被清零。在接口被禁用时（PE=0），该信息仍被更新。 1: 总线忙：SDA 或 SCL 存在低电平； 0: 总线空闲无通讯。 | 0 |
| 0 | MSL | RO | 主从模式指示位，当接口处于主模式时（SB=1），硬件将该位置位；当总线检测到一个停止位，仲裁丢失时，或者 PE=0 时，硬件会清除该位。 | 0 |

I2C 时钟寄存器（I2C_CKCFGR）

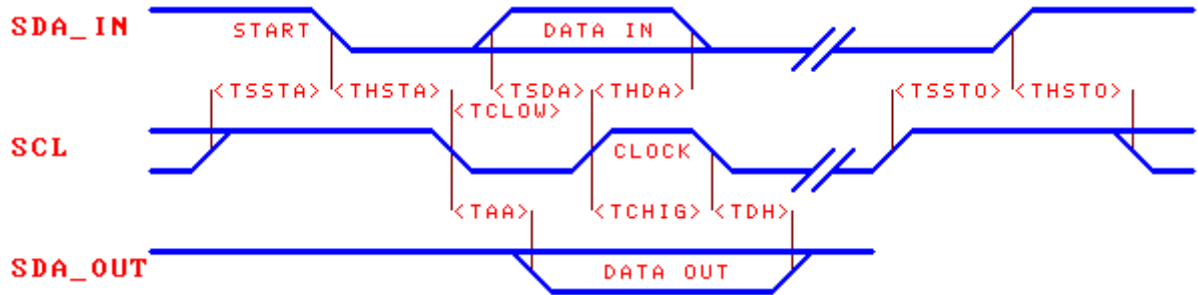
| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|----------|----|---|-----|
| 15 | F/S | RW | 主模式选择位。 1: 快速模式； 0: 标准模式。 | 0 |
| 14 | DUTY | RW | 快速模式时的时钟高电平时间的占空比。 1: 36%； 0: 33.3%。 | 0 |
| [13:12] | Reserved | RO | 保留。 | 0 |
| [11:0] | CCR | RW | 时钟分频系数域，决定 SCL 时钟的频率波形。 | 0 |

I2C 上升时间寄存器（I2C_RTR）

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|----------|----|-----|-----|
| [15:6] | Reserved | RO | 保留。 | 0 |

| | | | | |
|-------|-------|----|--|---------|
| [5:0] | TRISE | RW | 最大上升时间域。这个位设置主模式的 SCL 的上升时间。最大的上升沿时间等于 TRISE-1 个时钟周期。此位只能在 PE 清零下设置。 | 000010b |
|-------|-------|----|--|---------|

12.11 IIC 时序



| 名称 | 参数说明 (TA=25°C, VI033=3.3V) | 最小值 | 典型值 | 最大值 | 单位 |
|-------|----------------------------|--------|-----|-----|----|
| TSSTA | SDA 下降沿之前 SCL 高电平的建立时间 | 0.4 | | | uS |
| THSTA | SDA 下降沿之后 SCL 高电平的保持时间 | 0.4 | | | uS |
| TSDA | SCL 上升沿之前 SDA 数据的建立时间 | 0.05 | | | uS |
| THDA | SCL 上升沿之后 SDA 数据的保持时间 | >TCHIG | | | uS |
| TSSTO | SDA 上升沿之前 SCL 高电平的建立时间 | 0.4 | | | uS |
| THSTO | SDA 上升沿之后 SCL 高电平的保持时间 | 0.8 | | | uS |
| TCLOW | SCL 时钟的低电平时间 | 0.6 | | | uS |
| TCHIG | SCL 时钟的高电平时间 | 0.5 | | | uS |
| TAA | SCL 下降沿到 SDA 输出有效 | 0.006 | | 0.4 | uS |
| TDH | SCL 下降沿之后 SDA 输出保持时间 | 0.006 | | | uS |
| TR | SCL 或 SDA 输入上升沿时间 | | | 0.2 | uS |

第 13 章 预留

第 14 章 预留

第 15 章 模数转换器 ADC

15.1 ADC 简介

芯片提供一个 12 位逐次逼近型模拟数字转换器 ADC，提供多达 16 个通道，CH583 和 CH582 芯片支持 14 个外部信号源和 2 个内部信号源，CH581 芯片支持前 6 个外部信号源和 2 个内部信号源。

15.1.1 主要特性

- 12 位分辨率。
- 14 个外部电压采样通道、内部的温度检测通道、内部的电池电压检测通道。
- 单端输入模式和差分输入模式检测。
- 采样时钟频率可选。
- ADC 输入电压范围 $0V \sim V_{I033}$ 。
- 可选 PGA，提供增益调节选择。
- 可选输入缓冲器 BUF，支持高阻信号源。
- CH583 和 CH582 支持 DMA 和定时间隔自动连续 ADC 采样，间隔可调。CH581 不支持。

15.1.2 功能描述

下图为一个 ADC 模块的框图。

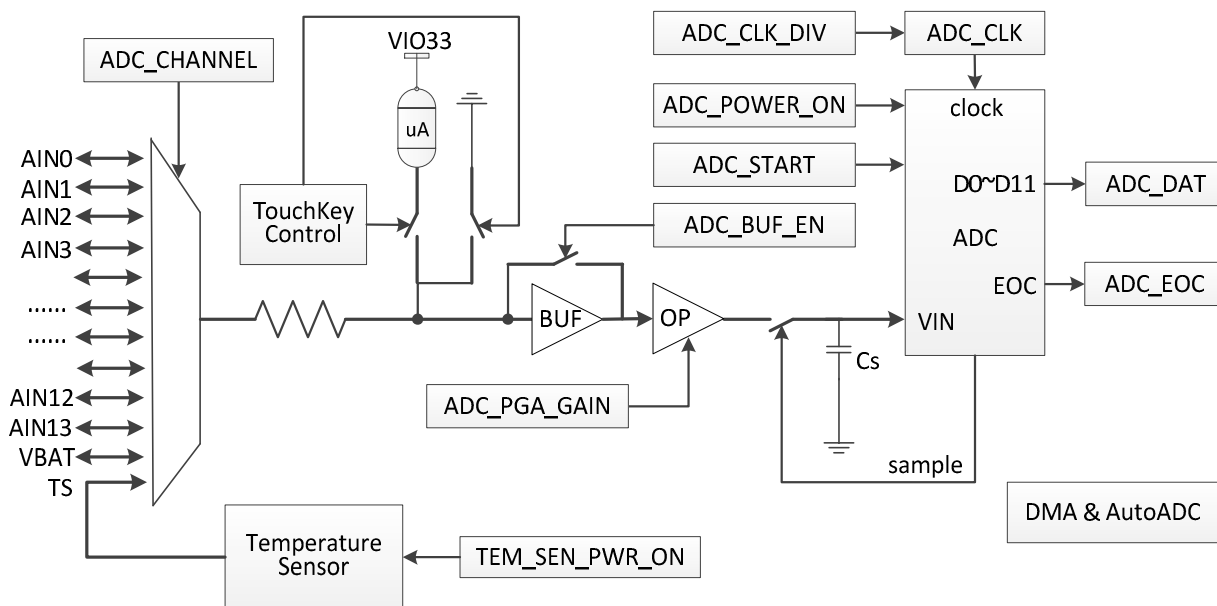


图 15-1 ADC 结构图

15.2 寄存器描述

表 15-1 ADC 相关寄存器列表

| 名称 | 访问地址 | 描述 | 复位值 |
|----------------|------------|---------------|------|
| R8_ADC_CHANNEL | 0x40001058 | ADC 输入通道选择寄存器 | 0x0F |
| R8_ADC_CFG | 0x40001059 | ADC 配置寄存器 | 0xA0 |
| R8_ADC_CONVERT | 0x4000105A | ADC 转换控制寄存器 | 0x00 |
| R8_TEM_SENSOR | 0x4000105B | 温度传感器控制寄存器 | 0x00 |

| | | | |
|-------------------|------------|-------------------|------------|
| R16_ADC_DATA | 0x4000105C | ADC 数据寄存器 | 0x0XXX |
| R8_ADC_INT_FLAG | 0x4000105E | ADC 中断标志寄存器 | 0x00 |
| R32_ADC_DMA_CTRL | 0x40001060 | DMA 控制和状态寄存器 | 0x00 |
| R8_ADC_CTRL_DMA | 0x40001061 | DMA 和中断控制寄存器 | 0x00 |
| R8_ADC_DMA_IF | 0x40001062 | ADC 和 DMA 中断标志寄存器 | 0x00 |
| R8_ADC_AUTO_CYCLE | 0x40001063 | 连续 ADC 定时周期寄存器 | 0xXX |
| R16_ADC_DMA_NOW | 0x40001064 | DMA 当前缓冲区地址 | 0x0000XXXX |
| R16_ADC_DMA_BEG | 0x40001068 | DMA 起始缓冲区地址 | 0x0000XXXX |
| R16_ADC_DMA_END | 0x4000106C | DMA 结束缓冲区地址 | 0x0000XXXX |

ADC 输入通道选择寄存器 (R8_ADC_CHANNEL)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|---------------|----|--|-------|
| [7:4] | Reserved | RO | 保留。 | 0000b |
| [3:0] | RB_ADC_CH_INX | RW | ADC 通道索引号，共 16 个通道： 00h~0Dh：外部信号通道 AIN0~AIN13； 0Eh：电池电压 VBAT； 0Fh：内置温度传感器 TS。 CH581 仅支持 AIN0~AIN5 和 VBAT 及 TS。 | 1111b |

ADC 配置寄存器 (R8_ADC_CFG)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|-----------------|----|---|-----|
| [7:6] | RB_ADC_CLK_DIV | RW | ADC 时钟频率选择： 00：基于 CK32M 的 10 分频，3.2MHz； 01：基于 CK32M 的 4 分频，8MHz； 10：基于 CK32M 的 6 分频，5.33MHz； 11：基于 CK32M 的 8 分频，4MHz。 实际采样速率约为该时钟频率的 1/18。 | 10b |
| [5:4] | RB_ADC_PGA_GAIN | RW | ADC 的输入 PGA 增益选择： 00：-12dB，1/4 倍； 01：-6dB，1/2 倍； 10：0dB，1 倍，无增益； 11：6dB，2 倍。 | 10b |
| 3 | RB_ADC_OFS_TEST | RW | ADC 偏移误差测试模式： 1：测试/校准模式，测试模式下数据寄存器 R16_ADC_DATA 低 12 位数据将按位取反（0x0573 反为 0x0A8C）； 0：正常模式。 | 0 |
| 2 | RB_ADC_DIFF_EN | RW | ADC 通道信号输入模式： 1：差分输入；0：单端输入。 | 0 |
| 1 | RB_ADC_BUF_EN | RW | ADC 输入缓冲器 BUF 使能： 1：开启；0：关闭。 | 0 |
| 0 | RB_ADC_POWER_ON | RW | ADC 模块电源使能控制： 1：使能；0：关闭。 | 0 |

表 15-2 PGA 增益选择与输入电压范围表

| PGA 增益选择 | 由 ADC 转换后的数据 计算被采样的电压 V_i | 理论可测 电压上限 | 理论可测电压范围 (假定 $V_{ref}=1.05V$) | 建议实际可用 测量电压范围 |
|---------------|--------------------------------|--------------|-----------------------------------|-------------------|
| -12dB (1/4 倍) | $(ADC/512-3)*V_{ref}$ | $5*V_{ref}$ | -0.2V ~ $V_{I033}+0.2V$ | 2.9V ~ V_{I033} |

| | | | | |
|--------------|---------------------|----------|-----------------|-------------|
| -6dB (1/2 倍) | (ADC/1024-1)*Vref | 3*Vref | -0.2V ~ 3.15V | 1.9V ~ 3V |
| 0dB (1 倍) | (ADC/2048)*Vref | 2*Vref | 0V ~ 2.1V | 0V ~ 2V |
| 6dB (2 倍) | (ADC/4096+0.5)*Vref | 1.5*Vref | 0.525V ~ 1.575V | 0.6V ~ 1.5V |

ADC: ADC 转换后的数字量, 即 R16_ADC_DATA。

Vref: 内部模拟电路的电源节点 VINTA 的实际电压值, 通常为 1.05V±0.015V。

说明: 较低的电压如果进行负增益 (信号减小) 后采样, 可能将会导致一段电压范围内误差大; 而较高的电压如果经过正增益 (信号放大) 后采样, 可能将会导致 ADC 转换值溢出, 因此建议根据被测信号电压范围选择合理的增益模式。

默认建议开启输入缓冲器, 只有当外部信号源内阻小于 1KΩ 时, 才可以关闭输入缓冲器进行 ADC。

当采用差分输入时, 建议关闭输入缓冲器。当用于 TouchKey 检测时, 必须开启输入缓冲器, 且建议选择 0dB (优先) 或 -6dB 两种增益之一。

ADC 转换控制寄存器 (R8_ADC_CONVERT)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|--------------|----|--|---------|
| 7 | RB_ADC_EOC_X | RO | ADC 转换结束标志 (异步信号): 1: 完成; 0: 进行中。 | 0 |
| [6:1] | Reserved | RO | 保留。 | 000000b |
| 0 | RB_ADC_START | RW | ADC 转换启动控制及状态, 在非连续 ADC 结束时或在 DMA 结束时自动清零: 1: 开始转换/正在转换; 0: 停止转换。 | 0 |

温度传感器控制寄存器 (R8_TEM_SENSOR)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|-------------------|----|----------------------------------|----------|
| 7 | RB_TEM_SEN_PWR_ON | RW | TS 温度传感器电源使能控制: 1: 使能; 0: 关闭。 | 0 |
| [6:0] | Reserved | RW | 保留。 | 0000000b |

ADC 数据寄存器 (R16_ADC_DATA)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---------|-------------|----|-------------|-------|
| [15:12] | Reserved | RW | 保留。 | 0000b |
| [11:0] | RB_ADC_DATA | RO | ADC 转换后的数据。 | XXXh |

ADC 中断标志寄存器 (R8_ADC_INT_FLAG)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|---------------|----|---|----------|
| 7 | RB_ADC_IF_EOC | RO | ADC 转换完成标志。 对寄存器 R8_ADC_CONVERT 或 R8_TKEY_CONVERT 进行写操作可清零此标志。 | 0 |
| [6:0] | Reserved | RW | 保留。 | 0000000b |

DMA 和中断控制寄存器 (R8_ADC_CTRL_DMA)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---|----------------|----|---|-----|
| 7 | RB_ADC_AUTO_EN | RW | 定时间隔自动连续 ADC 采样使能: 1: 使能自动 ADC; 0: 禁止自动 ADC。 | 0 |
| 6 | RB_ADC_CONT_EN | RW | ADC 连续转换模式使能: 1: 使能连续 ADC; 0: 禁止连续 ADC。 | 0 |
| 5 | Reserved | RO | 保留。 | 00b |
| 4 | RB_ADC_IE_EOC | RW | ADC 转换完成中断使能位: | 0 |

| | | | | |
|---|-------------------|----|--|---|
| | | | 1: 使能中断; 0: 禁止中断。 | |
| 3 | RB_ADC_IE_DMA_END | RW | DMA 结束中断使能位: 1: 使能中断; 0: 禁止中断。 | 0 |
| 2 | RB_ADC_DMA_LOOP | RW | DMA 地址循环功能使能位: 1: 使能地址循环; 0: 禁止地址循环。 如果使能 DMA 地址循环, 当 DMA 地址增加到设置的末尾地址时, 自动循环指向设置的首地址。 | 0 |
| 1 | Reserved | RO | 保留。 | 0 |
| 0 | RB_ADC_DMA_ENABLE | RW | DMA 功能使能位: 1: 使能 DMA; 0: 禁止 DMA。 | 0 |

ADC 和 DMA 中断标志寄存器 (R8_ADC_DMA_IF)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|-------------------|-----|--|------|
| [7:5] | Reserved | RO | 保留。 | 000b |
| 4 | RB_ADC_IF_END_ADC | RW1 | ADC 转换完成中断标志, 写 1 清零或 DMA 取走数据或对寄存器 R8_ADC_CONVERT 进行写操作可清零此标志: 1: 已完成一次 ADC; 0: 未完成。 | 0 |
| 3 | RB_ADC_IF_DMA_END | RW1 | DMA 完成标志位, 写 1 清零: 1: 已完成; 0: 未完成。 | 0 |
| [2:0] | Reserved | RO | 保留。 | 000b |

连续 ADC 定时周期寄存器 (R8_ADC_AUTO_CYCLE)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|-------------------|----|--|-----|
| [7:0] | R8_ADC_AUTO_CYCLE | RW | 连续 ADC 定时周期起始值, 以 16 个系统时钟为单位计数, 满 256 后重新加载。 计算方法: 定时 $= (256 - R8_ADC_AUTO_CYCLE) * 16 * T_{sys}$ | XXh |

DMA 当前缓冲区地址 (R16_ADC_DMA_NOW)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|-----------------|----|--|-------|
| [15:0] | R16_ADC_DMA_NOW | RO | DMA 数据缓冲区当前地址。 可用于计算已转换次数, 计算方法: $COUNT = (ADC_DMA_NOW - ADC_DMA_BEG) / 2$ 。 | XXXXh |

DMA 起始缓冲区地址 (R16_ADC_DMA_BEG)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|-----------------|----|--|-------|
| [15:0] | R16_ADC_DMA_BEG | RW | DMA 数据缓冲区起始地址, 仅低 15 位有效, 地址必须 2 字节对齐。 | XXXXh |

DMA 结束缓冲区地址 (R16_ADC_DMA_END)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|-----------------|----|---|-------|
| [15:0] | R16_ADC_DMA_END | RW | DMA 数据缓冲区结束地址 (不含), 仅低 15 位有效, 地址必须 2 字节对齐。 | XXXXh |

15.3 ADC 配置

15.3.1 外部通道采样

- (1)、设置 R8_ADC_CFG 的 RB_ADC_POWER_ON 为 1 开启 ADC，通过 RB_ADC_CLK_DIV 选择采样频率，通过 RB_ADC_BUF_EN 和 RB_ADC_PGA_GAIN 启用输入缓冲器及选择信号增益等；
- (2)、设置 R8_ADC_CHANNEL 寄存器，选择外部或内部信号通道；
- (3)、设置 R8_ADC_CONVERT 寄存器，置位 RB_ADC_START，启动 ADC 转换；
- (4)、查询等待 RB_ADC_START 自动清零或者等待 RB_ADC_IF_EOC 置 1，表示转换完成，可以读取 R16_ADC_DATA 获取 12 位 ADC 转换数据，时间足够时建议再次转换并丢弃首次 ADC 数据；
- (5)、重复 2、3、4 步骤，可以继续采样另一通道或者下一组数据。
- (6)、单次 ADC 转换周期：ADC 采样（4 个时钟）+ 切换 1 个时钟 + 转换时间（12 个时钟） \approx 17 个 Tadc，连续 ADC 时还要加上 1 个时间间隔，其中， $T_{adc} = T_{sys} @ RB_ADC_CLK_DIV$ 。
- (7)、如果使用差分输入：
使能差分，选择 0#通道：实际是对 AIN0（正端）和 AIN2（负端）的电压进行差分转换；
使能差分，选择 1#通道：实际是对 AIN1（正端）和 AIN3（负端）的电压进行差分转换；
ADC 转换后的结果，如果数据大于 0x800 表示差分正端的电压高于差分负端的电压；如果数据小于 0x800 表示差分正端的电压低于差分负端的电压。以 PGA 增益选择 0dB 为例，理论可测电压范围为 -1.05V~1.05V，0x400 表示差分正端的电压低于差分负端的电压约 0.5 个 Vref。

15.3.2 温度传感器采样

- (1)、设置 R8_TEM_SENSOR 寄存器的 RB_TEM_SEN_PWR_ON 为 1 开启温度传感器，设置 R8_ADC_CHANNEL 为 15，选择温度传感器信号连接到 ADC 输入；
- (2)、设置 RB_ADC_POWER_ON 为 1 开启 ADC，设置 RB_ADC_DIFF_EN 为 1，设置 RB_ADC_CLK_DIV，设置 RB_ADC_BUF_EN 为 0，设置 RB_ADC_PGA_GAIN 为 11；
- (3)、设置 R8_ADC_CONVERT 寄存器，置位 RB_ADC_START 为 1，启动 ADC 转换；
- (4)、查询等待 RB_ADC_START 自动清零或者等待 RB_ADC_IF_EOC 置 1，读取 R16_ADC_DATA 获取 12 位 ADC 转换数据，精度要求较高时，建议重复 3 和 4 步骤多次，计算 ADC 数据的平均值；
- (5)、根据电压与温度换算关系得到温度值，具体请参考评估板例子程序。

15.3.4 启用 DMA 自动连续 ADC

- (1)、参考非 DMA 方式设置 ADC 参数和选择通道等；
- (2)、设置 R8_ADC_AUTO_CYCLE 选择连续 ADC 的周期；
- (3)、设置寄存器 R16_ADC_DMA_BEG 为存储 ADC 数据缓冲区的首地址，设置寄存器 R16_ADC_DMA_END 为存储 ADC 数据缓冲区的结束地址（不含），并设置 R8_ADC_CTRL_DMA 的 RB_ADC_DMA_ENABLE 为 1，使能 DMA 功能；
- (4)、可选步骤，如果需要启用中断则设置相应的中断使能寄存器位，当 RB_ADC_IE_EOC=1 时将由 RB_ADC_IF_END_ADC 触发 ADC 完成中断，当 RB_ADC_IE_EOC=0 且 RB_ADC_IE_DMA_END=0 时将由 RB_ADC_IF_EOC 触发 ADC 完成中断；
- (5)、将 RB_ADC_AUTO_EN 置 1 以开启自动连续 ADC；
- (6)、每次 ADC 完成后，RB_ADC_IF_EOC 和 RB_ADC_IF_END_ADC 都会置 1，但 RB_ADC_IF_END_ADC 会在 DMA 取走数据时自动清零，所以，如果需要查询 ADC 完成状态，那么可以查询 RB_ADC_IF_EOC。

第 16 章 触摸按键 Touch-Key

16.1 Touch-Key 简介

CH583 和 CH582 芯片提供了电容检测模块，配合 ADC 模块使用，可以实现电容类触摸按键检测功能。共 14 个输入通道，支持触摸按键电容值范围 10pF~100pF，提供驱动屏蔽输出以提高灵敏度。

16.2 寄存器描述

表 16-1 TouchKey 相关寄存器列表

| 名称 | 访问地址 | 描述 | 复位值 |
|-----------------|------------|-------------------|------|
| R8_TKEY_COUNT | 0x40001054 | TouchKey 充放电时间寄存器 | 0xXX |
| R8_TKEY_CONVERT | 0x40001056 | TouchKey 检测控制寄存器 | 0x00 |
| R8_TKEY_CFG | 0x40001057 | TouchKey 配置寄存器 | 0x00 |

TouchKey 充放电时间寄存器 (R8_TKEY_COUNT)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|-------------------|----|---|--------|
| [7:5] | RB_TKEY_DISCH_CNT | RW | Touch-Key 放电周期数，以由 RB_ADC_CLK_DIV 选择的 ADC 时钟为计数单位。计算方法：放电时间 = (RB_TKEY_DISCH_CNT+1)*Tadc。 | XXXb |
| [4:0] | RB_TKEY_CHARG_CNT | RW | Touch-Key 充电周期数，以由 RB_ADC_CLK_DIV 选择的 ADC 时钟为计数单位。计算方法：充电时间 = (RB_TKEY_CHARG_CNT+4)*Tadc。 | XXXXXb |

TouchKey 检测控制寄存器 (R8_TKEY_CONVERT)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|---------------|----|---|----------|
| [7:1] | Reserved | RO | 保留。 | 0000000b |
| 0 | RB_TKEY_START | RW | TouchKey 检测控制及状态，自动清零： 1: 开始检测/正在转换；0: 停止转换。 | 0 |

TouchKey 配置寄存器 (R8_TKEY_CFG)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|-----------------|----|--|-------|
| [7:4] | Reserved | RO | 保留。 | 0000b |
| 3 | RB_TKEY_PGA_ADJ | RW | 选择 ADC 中 PGA 的运行速度： 1: 高速但功耗略大；0: 正常速度。 | 0 |
| 2 | RB_TKEY_DRV_EN | RW | TouchKey 驱动屏蔽使能： 1: 使能； 0: 禁用。 | 0 |
| 1 | RB_TKEY_CURRENT | RO | TouchKey 充电电流选择： 1: 60%电流；0: 额定电流。 | 0 |
| 0 | RB_TKEY_PWR_ON | RW | TouchKey 模块电源使能控制： 1: 使能； 0: 关闭。 | 0 |

16.3 Touch-Key 配置

请参考和调用相关子程序。

第 17 章 USB 控制器

17.1 USB 控制器简介

CH581 内嵌 1 个、CH583 和 CH582 内嵌 2 个完全独立的 USB 主从控制器及收发器，特性如下：

- 支持 USB Host 主机功能和 USB Device 设备功能。
- 支持 USB2.0 全速 12Mbps 或者低速 1.5Mbps。
- 支持 USB 控制传输、批量传输、中断传输、同步/实时传输。
- 支持最大 64 字节的数据包，内置 FIFO，支持中断和 DMA。

17.2 寄存器描述

两个 USB 控制器的基地址分别是 0x40008000 和 0x40008400，除此之外都相同，本手册以 USB 为例说明，USB2 参考使用。

USB 相关寄存器分为 3 个部分，部分寄存器是在主机和设备模式下进行复用的。

- (1)、USB 全局寄存器
- (2)、USB 设备控制寄存器
- (3)、USB 主机控制寄存器

17.2.1 全局寄存器描述

表 17-1 USB 相关寄存器列表（标灰受 RB_UC_RESET_SIE 复位控制）

| 名称 | 访问地址 | 描述 | 复位值 |
|----------------|------------|-------------|------------|
| R8_USB_CTRL | 0x40008000 | USB 控制寄存器 | 0x06 |
| R8_USB_INT_EN | 0x40008002 | USB 中断使能寄存器 | 0x00 |
| R8_USB_DEV_AD | 0x40008003 | USB 设备地址寄存器 | 0x00 |
| R32_USB_STATUS | 0x40008004 | USB 状态寄存器 | 0xXX20XXXX |
| R8_USB_MIS_ST | 0x40008005 | USB 杂项状态寄存器 | 0xXX |
| R8_USB_INT_FG | 0x40008006 | USB 中断标志寄存器 | 0x20 |
| R8_USB_INT_ST | 0x40008007 | USB 中断状态寄存器 | 0xXX |
| R8_USB_RX_LEN | 0x40008008 | USB 接收长度寄存器 | 0xXX |

USB 控制寄存器 (R8_USB_CTRL)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|------------------|----|---|-----|
| 7 | RB_UC_HOST_MODE | RW | USB 工作模式选择位： 1：主机模式 (HOST)； 0：设备模式 (DEVICE)。 | 0 |
| 6 | RB_UC_LOW_SPEED | RW | USB 总线信号传输速率选择位： 1：1.5Mbps；0：12Mbps。 | 0 |
| 5 | RB_UC_DEV_PU_EN | RW | USB 设备模式下，USB 设备使能和内部上拉电阻控制位，为 1 则使能 USB 设备传输并且启用内部上拉电阻。 RB_PIN_USB_DP_PU 可代替此位。 | 0 |
| [5:4] | MASK_UC_SYS_CTRL | RW | 见下表配置 USB 系统。 | 00b |
| 3 | RB_UC_INT_BUSY | RW | USB 传输完成中断标志未清零前自动暂停使能位： 1：在中断标志 UIF_TRANSFER 未清零前自动暂停，设备模式下自动应答忙 NAK， | 0 |

| | | | | |
|---|-----------------|----|--|---|
| | | | 主机模式下自动暂停后续传输； 0：不暂停。 | |
| 2 | RB_UC_RESET_SIE | RW | USB 协议处理器软件复位控制位： 1：强制复位 USB 协议处理器（SIE），需要软件清零； 0：不复位。 | 1 |
| 1 | RB_UC_CLR_ALL | RW | USB 的 FIFO 和中断标志清零： 1：强制清空和清零； 0：不清。 | 1 |
| 0 | RB_UC_DMA_EN | RW | USB 的 DMA 和 DMA 中断控制位： 1：使能 DMA 功能和 DMA 中断； 0：关闭 DMA。 | 0 |

由 RB_UC_HOST_MODE 和 MASK_UC_SYS_CTRL 组成 USB 系统控制组合：

| RB_UC_HOST_MODE | MASK_UC_SYS_CTRL | USB 系统控制描述 |
|-----------------|------------------|--|
| 0 | 00 | 禁止 USB 设备功能，关闭内部上拉电阻。 注：如果 RB_PIN_USB_DP_PU=1 则强制启用 DP 上拉电阻。 |
| 0 | 01 | 使能 USB 设备功能，关闭内部上拉电阻，需加外部上拉。 注：如果 RB_PIN_USB_DP_PU=1 则强制启用 DP 上拉电阻。 |
| 0 | 1x | 使能 USB 设备功能，启用内部 1.5K 上拉电阻。该上拉电阻优先于下拉电阻，也可用于 GPIO 模式。 |
| 1 | 00 | USB 主机模式，正常工作状态。 |
| 1 | 01 | USB 主机模式，强制 DP/DM 输出 SEO 状态。 |
| 1 | 10 | USB 主机模式，强制 DP/DM 输出 J 状态。 |
| 1 | 11 | USB 主机模式，强制 DP/DM 输出 K 状态/唤醒。 |

USB 中断使能寄存器 (R8_USB_INT_EN)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---|-----------------|----|---|-----|
| 7 | RB_UIE_DEV_SOF | RW | USB 设备模式，接收 SOF 包中断： 1：使能中断；0：禁止中断。 | 0 |
| 6 | RB_UIE_DEV_NAK | RW | USB 设备模式，接收到 NAK 中断： 1：使能中断；0：禁止中断。 | 0 |
| 5 | Reserved | RO | 保留。 | 0 |
| 4 | RB_UIE_FIFO_OV | RW | FIFO 溢出中断： 1：使能中断；0：禁止中断。 | 0 |
| 3 | RB_UIE_HST_SOF | RW | USB 主机模式，SOF 定时中断： 1：使能中断；0：禁止中断。 | 0 |
| 2 | RB_UIE_SUSPEND | RW | USB 总线挂起或唤醒事件中断： 1：使能中断；0：禁止中断。 | 0 |
| 1 | RB_UIE_TRANSFER | RW | USB 传输完成中断： 1：使能中断；0：禁止中断。 | 0 |
| 0 | RB_UIE_DETECT | RW | USB 主机模式，USB 设备连接或断开事件中断： 1：使能中断；0：禁止中断。 | 0 |
| | RB_UIE_BUS_RST | RW | USB 设备模式，USB 总线复位事件中断： 1：使能中断；0：禁止中断。 | 0 |

USB 设备地址寄存器 (R8_USB_DEV_AD)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---|----|----|----|-----|
|---|----|----|----|-----|

| | | | | |
|-------|---------------|----|--|----------|
| 7 | RB_UA_GP_BIT | RW | USB 通用标志位，用户自定义。 | 0 |
| [6:0] | MASK_USB_ADDR | RW | 主机模式：当前操作的 USB 设备地址； 设备模式：该 USB 自身地址。 | 0000000b |

USB 杂项状态寄存器(R8_USB_MIS_ST)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---|-------------------|----|---|-----|
| 7 | RB_UMS_SOF_PRES | RO | USB 主机模式下 SOF 包预示状态位： 1：将要发送 SOF 包，此时如有其它 USB 数据包将被自动延后； 0：无 SOF 包发送。 | X |
| 6 | RB_UMS_SOF_ACT | RO | USB 主机模式下 SOF 包传输状态位： 1：正在发出 SOF 包； 0：发送完成或者空闲。 | X |
| 5 | RB_UMS_SIE_FREE | RO | USB 协议处理器的空闲状态位： 1：协议器空闲； 0：忙，正在进行 USB 传输。 | 1 |
| 4 | RB_UMS_R_FIFO_RDY | RO | USB 接收 FIFO 数据就绪状态位： 1：接收 FIFO 非空； 0：接收 FIFO 为空。 | 0 |
| 3 | RB_UMS_BUS_RESET | RO | USB 总线复位状态位： 1：当前 USB 总线处于复位态； 0：当前 USB 总线处于非复位态。 | X |
| 2 | RB_UMS_SUSPEND | RO | USB 挂起状态位： 1：USB 总线处于挂起态，有一段时间没有 USB 活动； 0：USB 总线处于非挂起态。 | 0 |
| 1 | RB_UMS_DM_LEVEL | RO | USB 主机模式下，设备刚连入 USB 端口是 DM 引脚的电平状态，用于判断速度： 1：高电平/低速； 0：低电平/全速。 | 0 |
| 0 | RB_UMS_DEV_ATTACH | RO | USB 主机模式下端口的 USB 设备连接状态位： 1：端口已经连接 USB 设备； 0：端口没有 USB 设备连接。 | 0 |

USB 中断标志寄存器(R8_USB_INT_FG)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---|----------------|----|--|-----|
| 7 | RB_U_IS_NAK | RO | USB 设备模式下，NAK 响应状态位： 1：当前 USB 传输过程中回应 NAK； 0：无 NAK 响应。 | 0 |
| 6 | RB_U_TOG_OK | RO | 当前 USB 传输 DATA0/1 同步标志匹配状态位： 1：同步；0：不同步。 | 0 |
| 5 | RB_U_SIE_FREE | RO | USB 协议处理器空闲状态位： 1：USB 空闲； 0：忙，正在进行 USB 传输。 | 1 |
| 4 | RB_UIF_FIFO_OV | RW | USB FIFO 溢出中断标志位，写 1 清零： 1：FIFO 溢出触发；0：无事件。 | 0 |

| | | | | |
|---|-----------------|----|--|---|
| 3 | RB_UIF_HST_SOF | RW | USB 主机模式下 SOF 定时中断标志位，写 1 清零： 1: SOF 包传输完成触发；0: 无事件。 | 0 |
| 2 | RB_UIF_SUSPEND | RW | USB 总线挂起或唤醒事件中断标志位，写 1 清零： 1: USB 挂起事件或唤醒事件触发； 0: 无事件。 | 0 |
| 1 | RB_UIF_TRANSFER | RW | USB 传输完成中断标志位，写 1 清零： 1: 一个 USB 传输完成触发；0: 无事件。 | 0 |
| 0 | RB_UIF_DETECT | RW | USB 主机模式下 USB 设备连接或断开事件中断标志位，写 1 清零： 1: 检测到 USB 设备连接或断开触发； 0: 无事件。 | 0 |
| | RB_UIF_BUS_RST | RW | USB 设备模式下 USB 总线复位事件中断标志位，写 1 清零： 1: USB 总线复位事件触发； 0: 无事件。 | 0 |

USB 中断状态寄存器 (R8_USB_INT_ST)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|------------------|----|--|-------|
| 7 | RB_UIS_SETUP_ACT | RO | USB 设备模式下，该位为 1 表示当前已成功接收到 8 字节的 SETUP 请求包，SETUP 令牌不影响 RB_UIS_TOG_OK 和 MASK_UIS_TOKEN、MASK_UIS_ENDP 及 R8_USB_RX_LEN。 | 0 |
| 6 | RB_UIS_TOG_OK | RO | 当前 USB 传输 DATA0/1 同步标志匹配状态位，同 RB_U_TOG_OK： 1: 同步；0: 不同步。 | 0 |
| [5:4] | MASK_UIS_TOKEN | RO | 设备模式下，当前 USB 传输事务的令牌 PID 标识。 | XXb |
| [3:0] | MASK_UIS_ENDP | RO | 设备模式下，当前 USB 传输事务的端点号。 | XXXXb |
| | MASK_UIS_H_RES | RO | 主机模式下，当前 USB 传输事务的应答 PID 标识，0000 表示设备无应答或超时；其它值表示应答 PID。 | XXXXb |

MASK_UIS_TOKEN 用于 USB 设备模式下标识当前 USB 传输事务的令牌 PID: 00 表示 OUT 包；01 表示 SOF 包；10 表示 IN 包；11 表示空闲。

当 MASK_UIS_TOKEN 非空闲、并且 RB_UIS_SETUP_ACT 也为 1 时，必须先处理前者，处理完前者后清零一次 RB_UIF_TRANSFER 使前者进入空闲状态，再处理后，最后再清零一次 RB_UIF_TRANSFER。

MASK_UIS_H_RES 仅在主机模式下有效。在主机模式下，若主机发送 OUT/SETUP 令牌包时，则该 PID 是握手包 ACK/NAK/STALL，或者是设备无应答/超时。若主机发送 IN 令牌包，则该 PID 是数据包的 PID (DATA0/DATA1) 或者握手包 PID。

USB 接收长度寄存器 (R8_USB_RX_LEN)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|---------------|----|--------------------|-----|
| [7:0] | R8_USB_RX_LEN | RO | 当前 USB 端点接收的数据字节数。 | XXh |

17.2.2 设备寄存器描述

在 USB 设备模式下，芯片提供了端点 0、1、2、3、4、5、6、7 共 8 组双向端点，所有端点的最大数据包长度都是 64 字节。

端点 0 是默认端点，支持控制传输，发送和接收共用一个 64 字节数据缓冲区。

端点 1、端点 2、端点 3 各自包括一个发送端点 IN 和一个接收端点 OUT，发送和接收各有一个独立的 64 字节或者双 64 字节数据缓冲区，支持批量传输、中断传输和实时/同步传输。

端点 4、端点 5、端点 6、端点 7 包括一个发送端点 IN 和一个接收端点 OUT，发送和接收各有一个独立的 64 字节数据缓冲区，支持批量传输、中断传输和实时/同步传输。

每组端点都具有一个控制寄存器 R8_UEPn_CTRL 和发送长度寄存器 R8_UEPn_T_LEN

(n=0/1/2/3/4/5/6/7)，用于设定该端点的同步触发位、对 OUT 事务和 IN 事务的响应以及发送数据的长度等。

作为 USB 设备所必要的 USB 总线上拉电阻可以由软件随时设置是否启用，当 USB 控制寄存器 R8_USB_CTRL 中的 RB_UC_DEV_PU_EN 置 1 时，控制器根据 RB_UD_LOW_SPEED 的速度设置，在内部为 USB 总线的 DP/DM 引脚连接上拉电阻，并启用 USB 设备功能。上述控制不能用于睡眠或下电模式，但 R16_PIN_ANALOG_IE 中的 RB_PIN_USB_DP_PU 可以不受睡眠模式的影响而强制启用 DP 引脚的上拉电阻。

当检测到 USB 总线复位、USB 总线挂起或唤醒事件，或者当 USB 成功处理完数据发送或者数据接收后，USB 协议处理器都将设置相应的中断标志，如果中断使能打开，还会产生相应的中断请求。应用程序可以直接查询或在 USB 中断服务程序中查询并分析中断标志寄存器 R8_USB_INT_FG，根据 RB_UIF_BUS_RST 和 RB_UIF_SUSPEND 进行相应的处理；并且，如果 RB_UIF_TRANSFER 有效，那么还需要继续分析 USB 中断状态寄存器 R8_USB_INT_ST，根据当前端点号 MASK_UIS_ENDP 和当前事务令牌 PID 标识 MASK_UIS_TOKEN 进行相应的处理。如果事先设定了各个端点的 OUT 事务的同步触发位 RB_UEP_R_TOG，那么可以通过 RB_U_TOG_OK 或者 RB_UIS_TOG_OK 判断当前所接收到的数据包同步触发位是否与该端点的同步触发位匹配，如果数据同步，则数据有效；如果数据不同步，则数据应该被丢弃。每次处理完 USB 发送或者接收中断后，都应该正确修改相应端点的同步触发位，用于下次所发送的数据包或者下次所接收的数据包是否同步检测；另外，设置 RB_UEP_AUTO_TOG 可以在发送成功或者接收成功后自动翻转相应的同步触发位。

各个端点准备发送的数据在各自的缓冲区中，准备发送的数据长度是独立设定在 R8_UEPn_T_LEN 中；各个端点接收到的数据在各自的缓冲区中，但是接收到的数据长度都在 USB 接收长度寄存器 R8_USB_RX_LEN 中，可以在 USB 接收中断时根据当前端点号区分。

表 17-2 USB 设备相关寄存器列表（标灰受 RB_UC_RESET_SIE 复位控制）

| 名称 | 访问地址 | 描述 | 复位值 |
|---------------|------------|------------------|--------|
| R8_UDEV_CTRL | 0x40008001 | USB 设备物理端口控制寄存器 | 0xX0 |
| R8_UEP4_1_MOD | 0x4000800c | 端点 1/4 模式控制寄存器 | 0x00 |
| R8_UEP2_3_MOD | 0x4000800d | 端点 2/3 模式控制寄存器 | 0x00 |
| R8_UEP567_MOD | 0x4000800e | 端点 5/6/7 模式控制寄存器 | 0x00 |
| R16_UEP0_DMA | 0x40008010 | 端点 0 缓冲区起始地址 | 0xXXXX |
| R16_UEP1_DMA | 0x40008014 | 端点 1 缓冲区起始地址 | 0xXXXX |
| R16_UEP2_DMA | 0x40008018 | 端点 2 缓冲区起始地址 | 0xXXXX |
| R16_UEP3_DMA | 0x4000801c | 端点 3 缓冲区起始地址 | 0xXXXX |
| R8_UEP0_T_LEN | 0x40008020 | 端点 0 发送长度寄存器 | 0xXX |
| R8_UEP0_CTRL | 0x40008022 | 端点 0 控制寄存器 | 0x00 |
| R8_UEP1_T_LEN | 0x40008024 | 端点 1 发送长度寄存器 | 0xXX |
| R8_UEP1_CTRL | 0x40008026 | 端点 1 控制寄存器 | 0x00 |
| R8_UEP2_T_LEN | 0x40008028 | 端点 2 发送长度寄存器 | 0xXX |
| R8_UEP2_CTRL | 0x4000802a | 端点 2 控制寄存器 | 0x00 |
| R8_UEP3_T_LEN | 0x4000802c | 端点 3 发送长度寄存器 | 0xXX |
| R8_UEP3_CTRL | 0x4000802e | 端点 3 控制寄存器 | 0x00 |
| R8_UEP4_T_LEN | 0x40008030 | 端点 4 发送长度寄存器 | 0xXX |
| R8_UEP4_CTRL | 0x40008032 | 端点 4 控制寄存器 | 0x00 |

| | | | |
|---------------|------------|--------------|--------|
| R16_UEP5_DMA | 0x40008054 | 端点 5 缓冲区起始地址 | 0xXXXX |
| R16_UEP6_DMA | 0x40008058 | 端点 6 缓冲区起始地址 | 0xXXXX |
| R16_UEP7_DMA | 0x4000805c | 端点 7 缓冲区起始地址 | 0xXXXX |
| R8_UEP5_T_LEN | 0x40008064 | 端点 5 发送长度寄存器 | 0xXX |
| R8_UEP5_CTRL | 0x40008066 | 端点 5 控制寄存器 | 0x00 |
| R8_UEP6_T_LEN | 0x40008068 | 端点 6 发送长度寄存器 | 0xXX |
| R8_UEP6_CTRL | 0x4000806a | 端点 6 控制寄存器 | 0x00 |
| R8_UEP7_T_LEN | 0x4000806c | 端点 7 发送长度寄存器 | 0xXX |
| R8_UEP7_CTRL | 0x4000806e | 端点 7 控制寄存器 | 0x00 |

USB 设备物理端口控制寄存器 (R8_UDEV_CTRL)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---|-----------------|----|--|-----|
| 7 | RB_UD_PD_DIS | RW | USB 设备端口 UD+/UD-引脚内部下拉电阻控制位： 1: 禁用内部下拉；0: 使能内部下拉。 可用于 GPIO 模式提供下拉电阻。 | 1 |
| 6 | Reserved | RO | 保留。 | 0 |
| 5 | RB_UD_DP_PIN | RO | 当前 UD+引脚状态： 1: 高电平； 0: 低电平。 | X |
| 4 | RB_UD_DM_PIN | RO | 当前 UD-引脚状态： 1: 高电平； 0: 低电平。 | X |
| 3 | Reserved | RO | 保留。 | 0 |
| 2 | RB_UD_LOW_SPEED | RW | USB 设备物理端口低速模式使能位： 1: 选择 1.5Mbps 低速模式； 0: 选择 12Mbps 全速模式。 | 0 |
| 1 | RB_UD_GP_BIT | RW | USB 设备模式通用标志位，用户自定义。 | 0 |
| 0 | RB_UD_PORT_EN | RW | USB 设备物理端口使能位： 1: 使能物理端口；0: 禁用物理端口。 | 0 |

端点 1/4 模式控制寄存器 (R8_UEP4_1_MOD)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|-----------------|----|--------------------------------------|-----|
| 7 | RB_UEP1_RX_EN | RW | 1: 使能端点 1 接收 (OUT)； 0: 禁止端点 1 接收。 | 0 |
| 6 | RB_UEP1_TX_EN | RW | 1: 使能端点 1 发送 (IN)； 0: 禁止端点 1 发送。 | 0 |
| 5 | Reserved | RO | 保留。 | 0 |
| 4 | RB_UEP1_BUF_MOD | RW | 端点 1 数据缓冲区模式控制位。 | 0 |
| 3 | RB_UEP4_RX_EN | RW | 1: 使能端点 4 接收 (OUT)； 0: 禁止端点 4 接收。 | 0 |
| 2 | RB_UEP4_TX_EN | RW | 1: 使能端点 4 发送 (IN)； 0: 禁止端点 4 发送。 | 0 |
| [1:0] | Reserved | RO | 保留。 | 00b |

bUEP4_RX_EN 和 bUEP4_TX_EN 组合配置 USB 端点 0 和 4 的数据缓冲区模式，具体参考下表：

表 17-3 端点 0 和 4 缓冲区模式

| bUEP4_RX_EN | bUEP4_TX_EN | 描述：以 UEP0_DMA 为起始地址由低向高排列 |
|-------------|-------------|--|
| 0 | 0 | 端点 0 单 64 字节收发共用缓冲区 (IN 和 OUT)。 |
| 1 | 0 | 端点 0 单 64 字节收发共用缓冲区；端点 4 单 64 字节接收缓冲区 (OUT)。 |

| | | |
|---|---|---|
| 0 | 1 | 端点 0 单 64 字节收发共用缓冲区; 端点 4 单 64 字节发送缓冲区 (IN)。 |
| 1 | 1 | 端点 0 单 64 字节收发共用缓冲区; 端点 4 单 64 字节接收缓冲区(OUT); 端点 4 单 64 字节接收缓冲区 (IN)。总共 192 字节排列如下: UEP0_DMA+0 地址: 端点 0 收发共用缓冲区 64 字节起始地址; UEP0_DMA+64 地址: 端点 4 接收缓冲区 64 字节起始地址; UEP0_DMA+128 地址: 端点 4 发送缓冲区 64 字节起始地址。 |

端点 2/3 模式控制寄存器 (R8_UEP2_3_MOD)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---|-----------------|----|--------------------------------------|-----|
| 7 | RB_UEP3_RX_EN | RW | 1: 使能端点 3 接收 (OUT); 0: 禁止端点 3 接收。 | 0 |
| 6 | RB_UEP3_TX_EN | RW | 1: 使能端点 3 发送 (IN); 0: 禁止端点 3 发送。 | 0 |
| 5 | Reserved | RO | 保留。 | 0 |
| 4 | RB_UEP3_BUF_MOD | RW | 端点 3 数据缓冲区模式控制位。 | 0 |
| 3 | RB_UEP2_RX_EN | RW | 1: 使能端点 2 接收 (OUT); 0: 禁止端点 2 接收。 | 0 |
| 2 | RB_UEP2_TX_EN | RW | 1: 使能端点 2 发送 (IN); 0: 禁止端点 2 发送。 | 0 |
| 1 | Reserved | RO | 保留。 | 0 |
| 0 | RB_UEP2_BUF_MOD | RW | 端点 2 数据缓冲区模式控制位。 | 0 |

由 RB_UEPn_RX_EN 和 RB_UEPn_TX_EN 以及 RB_UEPn_BUF_MOD(n=1/2/3)组合分别配置 USB 端点 1、2、3 的数据缓冲区模式, 具体参考下表。其中, 在双 64 字节缓冲区模式下, USB 数据传输时将根据 RB_UEP*_TOG=0 选择前 64 字节缓冲区, 根据 RB_UEP*_TOG=1 选择后 64 字节缓冲区, 设置 RB_UEP_AUTO_TOG=1 可实现自动切换。

表 17-4 端点 n 缓冲区模式 (n=1/2/3)

| RB_UEPn_RX_EN | RB_UEPn_TX_EN | RB_UEPn_BUF_MOD | 描述: 以 R16_UEPn_DMA 为起始地址由低向高排列 |
|---------------|---------------|-----------------|--|
| 0 | 0 | X | 端点被禁用, 未用到 R16_UEPn_DMA 缓冲区。 |
| 1 | 0 | 0 | 单 64 字节接收缓冲区(OUT)。 |
| 1 | 0 | 1 | 双 64 字节接收缓冲区(OUT), 由 RB_UEP_R_TOG 选择。 |
| 0 | 1 | 0 | 单 64 字节发送缓冲区(IN)。 |
| 0 | 1 | 1 | 双 64 字节发送缓冲区(IN), 由 RB_UEP_T_TOG 选择。 |
| 1 | 1 | 0 | 单 64 字节接收缓冲区(OUT), 单 64 字节发送缓冲区(IN)。 |
| 1 | 1 | 1 | 双 64 字节接收缓冲区(OUT), 通过 RB_UEP_R_TOG 选择, 双 64 字节发送缓冲区(IN), 通过 RB_UEP_T_TOG 选择。 全部 256 字节排列如下: UEPn_DMA+0 地址: RB_UEP_R_TOG=0 时端点接收地址; UEPn_DMA+64 地址: RB_UEP_R_TOG=1 时端点接收地址; UEPn_DMA+128 地址: RB_UEP_T_TOG=0 时端点发送地址; UEPn_DMA+192 地址: RB_UEP_T_TOG=1 时端点发送地址。 |

端点 5/6/7 模式控制寄存器 (R8_UEP567_MOD)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|---------------|----|--------------------------------------|-----|
| [7:6] | Reserved | RO | 保留。 | 00b |
| 5 | RB_UEP7_RX_EN | RW | 1: 使能端点 7 接收 (OUT); 0: 禁止端点 7 接收。 | 0 |

| | | | | |
|---|---------------|----|--------------------------------------|---|
| 4 | RB_UEP7_TX_EN | RW | 1: 使能端点 7 发送 (IN); 0: 禁止端点 7 发送。 | 0 |
| 3 | RB_UEP6_RX_EN | RW | 1: 使能端点 6 接收 (OUT); 0: 禁止端点 6 接收。 | 0 |
| 2 | RB_UEP6_TX_EN | RW | 1: 使能端点 6 发送 (IN); 0: 禁止端点 6 发送。 | 0 |
| 1 | RB_UEP5_RX_EN | RW | 1: 使能端点 5 接收 (OUT); 0: 禁止端点 5 接收。 | 0 |
| 0 | RB_UEP5_TX_EN | RW | 1: 使能端点 5 发送 (IN); 0: 禁止端点 5 发送。 | 0 |

由 RB_UEPn_RX_EN 和 RB_UEPn_TX_EN (n=5/6/7) 组合分别配置 USB 端点 5、6、7 的数据缓冲区模式，具体参考下表。

表 17-5 端点 n 缓冲区模式 (n=5/6/7)

| RB_UEPn_RX_EN | RB_UEPn_TX_EN | 描述: 以 R16_UEPn_DMA 为起始地址由低向高排列 |
|---------------|---------------|--|
| 0 | 0 | 端点被禁用, 未用到 R16_UEPn_DMA 缓冲区。 |
| 1 | 0 | 单 64 字节接收缓冲区 (OUT)。 |
| 0 | 1 | 单 64 字节发送缓冲区 (IN)。 |
| 1 | 1 | 单 64 字节接收缓冲区 (OUT), 单 64 字节发送缓冲区 (IN)。 |

端点 n 缓冲区起始地址 (R16_UEPn_DMA) (n=0/1/2/3/5/6/7)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|--------------|----|---|-------|
| [15:0] | R16_UEPn_DMA | RW | 端点 n 缓冲区起始地址。 低 15 位有效, 地址必须 4 字节对齐。 | XXXXh |

注: 接收数据的缓冲区的长度 $\geq \min$ (可能收到的最大数据包长度 + 2 字节, 64 字节)。

端点 n 发送长度寄存器 (R8_UEPn_T_LEN) (n=0/1/2/3/4/5/6/7)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|---------------|----|-------------------------|-----|
| [7:0] | R8_UEPn_T_LEN | RW | 设置 USB 端点 n 准备发送的数据字节数。 | XXh |

端点 n 控制寄存器 (R8_UEPn_CTRL) (n=0/1/2/3/4/5/6/7)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|-----------------|----|--|-----|
| 7 | RB_UEP_R_TOG | RW | USB 端点 n 的接收器 (处理 OUT 事务) 期望的同步触发位: 1: 期望 DATA1; 0: 期望 DATA0。 | 0 |
| 6 | RB_UEP_T_TOG | RW | USB 端点 n 的发送器 (处理 IN 事务) 准备的同步触发位: 1: 发送 DATA1; 0: 发送 DATA0。 | 0 |
| 5 | Reserved | RO | 保留。 | 0 |
| 4 | RB_UEP_AUTO_TOG | RW | 同步触发位自动翻转使能控制位: 1: 数据发送或接收成功后自动翻转相应的同步触发位; 0: 不自动翻转, 可以手动切换。 只支持端点 1/2/3/5/6/7。 | 0 |
| [3:2] | MASK_UEP_R_RES | RW | 端点 n 的接收器对 OUT 事务的响应控制: 00: 应答 ACK; 01: 超时/无响应, 用于非端点 0 的实时/同步传输; | 00b |

| | | | | |
|-------|----------------|----|--|-----|
| | | | 10: 应答 NAK 或忙; 11: 应答 STALL 或错误。 | |
| [1:0] | MASK_UEP_T_RES | RW | 端点 n 的发送器对 IN 事务的响应控制: 00: DATA0/DATA1 数据就绪并期望 ACK; 01: 应答 DATA0/DATA1 并期望无响应, 用于非端点 0 的实时/同步传输; 10: 应答 NAK 或忙; 11: 应答 STALL 或错误。 | 00b |

17.2.3 主机寄存器描述

在 USB 主机模式下，芯片提供了一组双向主机端点，包括一个发送端点 OUT 和一个接收端点 IN，数据包的最大长度是 64 字节，支持控制传输、中断传输、批量传输和实时/同步传输。

主机端点发起的每一个 USB 事务，在处理结束后总是自动设置 RB_UIF_TRANSFER 中断标志。应用程序可以直接查询或在 USB 中断服务程序中查询并分析中断标志寄存器 R8_USB_INT_FG，根据各中断标志分别进行相应的处理；并且，如果 RB_UIF_TRANSFER 有效，那么还需要继续分析 USB 中断状态寄存器 R8_USB_INT_ST，根据当前 USB 传输事务的应答 PID 标识 MASK_UIS_H_RES 进行相应的处理。

如果事先设定了主机接收端点的 IN 事务的同步触发位 RB_UH_R_TOG，那么可以通过 RB_U_TOG_OK 或者 RB_UIS_TOG_OK 判断当前所接收到的数据包的同步触发位是否与主机接收端点的同步触发位匹配，如果数据同步，则数据有效；如果数据不同步，则数据应该被丢弃。每次处理完 USB 发送或者接收中断后，都应该正确修改相应主机端点的同步触发位，用于同步下次所发送的数据包和检测下次所接收的数据包是否同步；另外，通过设置 RB_UH_T_AUTO_TOG 和 RB_UH_R_AUTO_TOG 可以实现在发送成功或接收成功后自动翻转相应的同步触发位。

USB 主机令牌设置寄存器 R8_UH_EP_PID 用于设置被操作的目标设备的端点号和本次 USB 传输事务的令牌 PID 包标识。SETUP 令牌和 OUT 令牌所对应的数据由主机发送端点提供，准备发送的数据在 R16_UH_TX_DMA 缓冲区中，准备发送的数据长度设置在 R16_UH_TX_LEN 中；IN 令牌所对应数据由目标设备返回给主机接收端点，接收到数据存放 R16_UH_RX_DMA 缓冲区中，接收到的数据长度存放在 R8_USB_RX_LEN 中。

表 17-6 USB 主机相关寄存器列表（标灰受 RB_UC_RESET_SIE 复位控制）

| 名称 | 访问地址 | 描述 | 复位值 |
|---------------|------------|-----------------|--------|
| R8_UHOST_CTRL | 0x40008001 | USB 主机物理端口控制寄存器 | 0xX0 |
| R8_UH_EP_MOD | 0x4000800d | USB 主机端点模式控制寄存器 | 0x00 |
| R16_UH_RX_DMA | 0x40008018 | USB 主机接收缓冲区起始地址 | 0xXXXX |
| R16_UH_TX_DMA | 0x4000801c | USB 主机发送缓冲区起始地址 | 0xXXXX |
| R8_UH_SETUP | 0x40008026 | USB 主机辅助设置寄存器 | 0x00 |
| R8_UH_EP_PID | 0x40008028 | USB 主机令牌设置寄存器 | 0x00 |
| R8_UH_RX_CTRL | 0x4000802a | USB 主机接收端点控制寄存器 | 0x00 |
| R8_UH_TX_LEN | 0x4000802c | USB 主机发送长度寄存器 | 0xXX |
| R8_UH_TX_CTRL | 0x4000802e | USB 主机发送端点控制寄存器 | 0x00 |

USB 主机物理端口控制寄存器 (R8_UHOST_CTRL)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---|--------------|----|--|-----|
| 7 | RB_UH_PD_DIS | RW | USB 主机端口 UD+/UD-引脚内部下拉电阻控制位： 1: 禁用内部下拉；0: 使能内部下拉。 可用于 GPIO 模式提供下拉电阻。 | 1 |
| 6 | Reserved | RO | 保留。 | 0 |
| 5 | RB_UH_DP_PIN | RO | 当前 UD+引脚状态： | X |

| | | | | |
|---|-----------------|----|--|---|
| | | | 1: 高电平; 0: 低电平。 | |
| 4 | RB_UH_DM_PIN | RO | 当前 UD-引脚状态: 1: 高电平; 0: 低电平。 | X |
| 3 | Reserved | RO | 保留。 | 0 |
| 2 | RB_UH_LOW_SPEED | RW | USB 主机端口低速模式使能位: 1: 选择 1.5Mbps 低速模式; 0: 选择 12Mbps 全速模式。 | 0 |
| 1 | RB_UH_BUS_RESET | RW | USB 主机模式总线复位控制位: 1: 强制输出 USB 总线复位; 0: 结束输出。 | 0 |
| 0 | RB_UH_PORT_EN | RW | USB 主机端口使能位: 1: 使能主机端口; 0: 禁用主机端口。 当 USB 设备断开连接时, 该为自动清 0。 | 0 |

USB 主机端点模式控制寄存器 (R8_UH_EP_MOD)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|-------------------|----|---|-----|
| 7 | Reserved | RO | 保留。 | 0 |
| 6 | RB_UH_EP_TX_EN | RW | 主机发送端点发送 (SETUP/OUT) 使能位: 1: 使能端点发送; 0: 禁止端点发送。 | 0 |
| 5 | Reserved | RO | 保留。 | 0 |
| 4 | RB_UH_EP_TBUF_MOD | RW | 主机发送端点发送数据缓冲区模式控制位。 | 0 |
| 3 | RB_UH_EP_RX_EN | RW | 主机接收端点接收 (IN) 使能位: 1: 使能端点接收; 0: 禁止端点接收。 | 0 |
| [2:1] | Reserved | RO | 保留。 | 00b |
| 0 | RB_UH_EP_RBUF_MOD | RW | USB 主机接收端点接收数据缓冲区模式控制位。 | 0 |

由 RB_UH_EP_TX_EN 和 RB_UH_EP_TBUF_MOD 组合控制主机发送端点数据缓冲区模式, 参考下表。

表 17-7 主机发送缓冲区模式

| RB_UH_EP_TX_EN | RB_UH_EP_TBUF_MOD | 描述: 以 R16_UH_TX_DMA 为起始地址 |
|----------------|-------------------|--|
| 0 | X | 端点被禁用, 未用到 R16_UH_TX_DMA 缓冲区。 |
| 1 | 0 | 单 64 字节发送缓冲区 (SETUP/OUT)。 |
| 1 | 1 | 双 64 字节发送缓冲区, 通过 RB_UH_T_TOG 选择: 当 RB_UH_T_TOG=0 时选择前 64 字节缓冲区; 当 RB_UH_T_TOG=1 时选择后 64 字节缓冲区。 |

由 RB_UH_EP_RX_EN 和 RB_UH_EP_RBUF_MOD 组合控制主机接收端点数据缓冲区模式, 参考下表。

表 17-8 主机接收缓冲区模式

| RB_UH_EP_RX_EN | RB_UH_EP_RBUF_MOD | 结构描述: 以 R16_UH_TX_DMA 为起始地址 |
|----------------|-------------------|--|
| 0 | X | 端点被禁用, 未用到 R16_UH_RX_DMA 缓冲区。 |
| 1 | 0 | 单 64 字节接收缓冲区 (IN)。 |
| 1 | 1 | 双 64 字节接收缓冲区, 通过 RB_UH_R_TOG 选择: 当 RB_UH_R_TOG=0 时选择前 64 字节缓冲区; 当 RB_UH_R_TOG=1 时选择后 64 字节缓冲区。 |

USB 主机接收缓冲区起始地址 (R16_UH_RX_DMA)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|---|----|----|----|-----|
|---|----|----|----|-----|

| | | | | |
|--------|---------------|----|---|-------|
| [15:0] | R16_UH_RX_DMA | RW | 主机端点数据接收缓冲区起始地址。 低 15 位有效，地址必须 4 字节对齐。 | XXXXb |
|--------|---------------|----|---|-------|

USB 主机发送缓冲区起始地址 (R16_UH_TX_DMA)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|--------|---------------|----|---|-------|
| [15:0] | R16_UH_TX_DMA | RW | 主机端点数据发送缓冲区起始地址。 低 15 位有效，地址必须 4 字节对齐。 | XXXXb |

USB 主机辅助设置寄存器 (R8_UH_SETUP)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|------------------|----|---|---------|
| 7 | RB_UH_PRE_PID_EN | RW | 低速前导包 PRE PID 使能位： 1：使能，用于通过外部 HUB 与低速 USB 设备通讯。 0：禁用低速前导包。 | 0 |
| 6 | RB_UH_SOF_EN | RW | 自动产生 SOF 包使能位： 1：主机自动产生 SOF 包； 0：不自动产生，但可手工产生。 | 0 |
| [5:0] | Reserved | RO | 保留。 | 000000b |

USB 主机令牌设置寄存器 (R8_UH_EP_PID)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|---------------|----|---------------------------|-------|
| [7:4] | MASK_UH_TOKEN | RW | 设置本次 USB 传输事务的令牌 PID 包标识。 | 0000b |
| [3:0] | MASK_UH_ENDP | RW | 设置本次被操作的目标设备的端点号。 | 0000b |

USB 主机接收端点控制寄存器 (R8_UH_RX_CTRL)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|------------------|----|--|-----|
| 7 | RB_UH_R_TOG | RW | USB 主机接收器（处理 IN 事务）期望的同步触发位： 1：期望 DATA1；0：期望 DATA0。 | 0 |
| [6:5] | Reserved | RO | 保留。 | 00b |
| 4 | RB_UH_R_AUTO_TOG | RW | 同步触发位自动翻转使能控制位： 1：数据接收成功后自动翻转相应的期待同步触发位 (RB_UH_R_TOG)； 0：不自动翻转，可以手动切换。 | 0 |
| 3 | Reserved | RO | 保留。 | 0 |
| 2 | RB_UH_R_RES | RW | 主机接收器对 IN 事务的响应控制位： 1：无响应，用于非 0 端点的实时/同步传输； 0：应答 ACK。 | 0 |
| [1:0] | Reserved | RO | 保留。 | 00b |

USB 主机发送长度寄存器 (R8_UH_TX_LEN)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|--------------|----|--------------------------|-----|
| [7:0] | R8_UH_TX_LEN | RW | 设置 USB 主机发送端点准备发送的数据字节数。 | XXh |

USB 主机发送端点控制寄存器 (R8_UH_TX_CTRL)

| 位 | 名称 | 访问 | 描述 | 复位值 |
|-------|------------------|----|--|------|
| 7 | Reserved | R0 | 保留。 | 0 |
| 6 | RB_UH_T_TOG | RW | USB 主机发送器（处理 SETUP/OUT 事务）准备的同步触发位： 1：表示发送 DATA1；0：表示发送 DATA0。 | 0 |
| 5 | Reserved | R0 | 保留。 | 0 |
| 4 | RB_UH_T_AUTO_TOG | RW | 同步触发位自动翻转使能控制位： 1：数据发送成功后自动翻转相应的同步触发位 (RB_UH_T_TOG)； 0：不自动翻转，可以手动切换。 | 0 |
| [3:1] | Reserved | R0 | 保留。 | 000b |
| 0 | RB_UH_T_RES | RW | USB 主机发送器对 SETUP/OUT 事务的响应控制位： 1：期望无响应，用于非 0 端点的实时/同步传输； 0：期望应答 ACK。 | 0 |

第 18 章 预留

第 19 章 无线通讯

19.1 简介

芯片集成低功耗 2.4-GHz 无线通讯模块，包括 RF 收发器、基带和链路控制以及天线匹配网络，支持低功耗蓝牙 BLE。内部提供数百个寄存器用于调节参数和控制过程及状态，本手册不对寄存器作详细介绍，无线通讯底层操作主要以子程序库提供应用支持。

主要特性：

- 集成 2.4GHz 射频收发器、BaseBand 基带和 LLE 链路控制。
- 支持低功耗蓝牙 BLE，符合 Bluetooth Low Energy 4.2/5.0 及以上规范。
- 单端 RF 接口，简化板级设计。
- 支持 2Mbps、1Mbps、500Kbps 和 125Kbps。
- 接收灵敏度-98dBm。
- 可编程-16dBm 到+6dBm 发送功率，支持动态调整。
- 在 0dBm 发送功率、PCB 板载天线时开阔地通讯距离约 210 米，6dBm 发送功率时约 500 米。
- 在 6dBm 发送功率、125Kbps 模式下开阔地通讯距离约 1000 米。
- 支持 AES 加解密。
- 支持 DMA。
- 提供优化的协议栈和应用层 API，支持组网。

19.2 LLE 模块

LLE 模块支持自动收发模式和手动收发模式，5 组独立的硬件定时器，可控制收发数据任意一过程的时间点。

19.3 DMA 模块

控制器共有 2 组 DMA，每组 DMA 有两个通道。DMA0 的两个通道分别用于发送数据和接收数据，DMA2 的两个通道用于自动模式，在自动发送模式中，可以同时配置发送 DMA 的地址和接收 DMA 的地址，这样在帧间隔期间不需要再进行配置。

19.4 BB 模块

19.5 AES 模块

具体的应用请基于 BLE 协议栈库使用，并参考提供的 BLE 应用示例。

第 20 章 参数

20.1 绝对最大值

临界或者超过绝对最大值将可能导致芯片工作不正常甚至损坏。

表 20-1 绝对最大值参数表

| 名称 | 参数说明 | 最小值 | 最大值 | 单位 |
|-------|-------------------------------------|------|-----------|----|
| TA | 工作时的环境温度 | -40 | 85 | °C |
| TS | 储存时的环境温度 | -40 | 125 | °C |
| VDD33 | 系统电源电压 (VDD33 接电源, GND 接地) | -0.4 | 4.2 | V |
| VI033 | I/O 电源电压 (VI033 接电源, GND 接地) | -0.4 | 4.2 | V |
| VI0 | 输入或者输出引脚上的电压 | -0.4 | VI033+0.4 | V |
| VI05 | 支持 5V 耐压的输入或者输出引脚上的电压 | -0.4 | 5.5 | V |
| VDCI | VDCID/VDCIA 引脚上的电压 (如果用外置 DC-DC) | -0.4 | VDD33+0.4 | V |
| VXCK | X32MI/X32MO/启用 LSE 后的 PA10/PA11 的电压 | -0.3 | 1.4 | V |

20.2 电气参数

测试条件: TA=25°C, VDD33=VI033=3.3V, Fsys=16MHz。

表 20-2 电气参数表

| 名称 | 参数说明 | 最小值 | 典型值 | 最大值 | 单位 | |
|--------------------|--|-----------|-----------|-------|-----|----|
| VDD33 | 系统电源电压@VDD33 | CH583M | 1.7 | 3.3 | 3.6 | V |
| | | 非 CH583M | 2.3 | 3.3 | 3.6 | V |
| VI033 | I/O 电源电压@VI033 | CH583M | 1.7 | 3.3 | 3.6 | V |
| | | 非 CH583M | 2.3 | 3.3 | 3.6 | V |
| ICC ₈ | 直通方式静态电源电流 条件: 代码运行于 RAM 中。 如运行于 Flash 中另加 5mA | Fsys=8M | | 2.0 | | mA |
| ICC ₁₆ | | Fsys=16M | | 2.2 | | mA |
| ICC ₄₈ | | Fsys=48M | | 6.0 | | mA |
| IDDC ₈ | 启用 DC-DC 后静态电源电流 条件: 代码运行于 RAM 中。 如运行于 Flash 中另加 4mA | Fsys=8M | | 1.5 | | mA |
| IDDC ₁₆ | | Fsys=16M | | 1.6 | | mA |
| IDDC ₄₈ | | Fsys=48M | | 4.0 | | mA |
| VIL | GPIO 低电平输入电压 | 0 | | 0.9 | V | |
| VIH | GPIO 高电平输入电压 | 2.0 | | VI033 | V | |
| VIL5 | 支持 5V 耐压的 GPIO 低电平输入电压 | 0 | | 0.9 | V | |
| VIH5 | 支持 5V 耐压的 GPIO 高电平输入电压 | 2.0 | | 5.0 | V | |
| VOL | 低电平输出电压 (5mA/20mA 吸入电流) | 0 | 0.3 | 0.4 | V | |
| VOH | 高电平输出电压 (5mA/20mA 输出电流) | VI033-0.4 | VI033-0.3 | VI033 | V | |
| IIN | GPIO 浮空输入端的输入电流 | -3 | 0 | 3 | uA | |
| IUP | GPIO 内置上拉电阻的输入端的输入电流 | 25 | 60 | 90 | uA | |
| IDN | GPIO 内置下拉电阻的输入端的输入电流 | -90 | -60 | 25 | uA | |
| Vref | VINTA 引脚的电压 (ADC 参考电压) | 1.035 | 1.05 | 1.065 | V | |
| Vdci | VDCID 引脚在启用 DC-DC 后的电压 | 1.18 | 1.3 | 1.38 | V | |
| Vlvr | LVR 低压复位的门限电压 | CH583X | 1.3 | 1.5 | 1.7 | V |
| | | 非 CH583X | 1.8 | 2.05 | 2.3 | V |

20.3 低功耗模式功耗

测试条件：TA=25℃，VDD33=VI033=3.3V，Fsys=16MHz。

表 20-3 低功耗参数表（仅供参考，与温度相关）

| 低功耗模式 | 最小值 | 典型值 | 最大值 | 单位 |
|---------------------------------|-----|---------|-----|----|
| 空闲模式，开启各模块时钟组合 | 1.2 | 1.6 | 1.8 | mA |
| 暂停模式，关闭所有时钟 | | 320 | | uA |
| 睡眠模式，多种组合，参考表 5-3 | | 0.7~2.8 | | uA |
| 睡眠模式，PMU+内核+RAM2K，GPIO 唤醒，无 RTC | | 0.7 | | uA |
| 下电模式，多种组合，参考表 5-3 | | 0.2~2.3 | | uA |
| 下电模式，仅 PMU，GPIO 唤醒后复位，无 RTC | | 0.2 | | uA |

表 20-4 各模块电流（仅供参考，与温度相关）

| 名称 | 参数说明 | 最小值 | 典型值 | 最大值 | 单位 | |
|--------------------------|-------------------|----------------|----------|------|-----|----|
| I _{DD} (RAM2K) | RAM2K: 2KB SRAM | | 0.3 | | uA | |
| I _{DD} (RAM30K) | RAM30K: 30KB SRAM | | 1.4 | | uA | |
| I _{DD} (LSI) | 内部 LSI 振荡器 | | 0.3 | | uA | |
| I _{DD} (LSE) | 外部 LSE 振荡器 | | 0.4 | | uA | |
| I _{DD} (HSE) | 外部 HSE 振荡器 | 100 | 200 | 300 | uA | |
| I _{DD} (BM) | 低功耗的电池低压监控 BM 模块 | | 0.9 | | uA | |
| I _{DD} (BD) | 高精度的电池低压检测 BD 模块 | | 100 | | uA | |
| I _{DD} (PLL) | 内部 PLL 振荡器 | | 150 | | uA | |
| I _{DD} (ADC) | ADC 模块 | | 0.4 | | mA | |
| I _{DD} (TKEY) | TouchKey 模块 | | 0.1 | | mA | |
| I _{DD} (TS) | 温度传感器 TS 模块 | | 0.1 | | mA | |
| I _{DD} (USB) | USB 模块 | 非发送状态 | 1.2 | 1.6 | 2.0 | mA |
| | | 发送状态 | | 3 | | mA |
| I _{DD} (BLE) | BLE 蓝牙 | 接收 | 直通电源 | 13.5 | | mA |
| | | | 启用 DC-DC | 6.5 | | mA |
| | | -16dBm 发送功率 | 直通电源 | 5 | | mA |
| | | | 启用 DC-DC | 2.5 | | mA |
| | | 0dBm 发送功率 | 直通电源 | 10 | | mA |
| | | | 启用 DC-DC | 4.5 | | mA |
| +6dBm 发送功率 | 直通电源 | 15.5 | | mA | | |
| | 启用 DC-DC | 7 | | mA | | |

20.4 时钟源

表 20-5 高速振荡器 HSE

| 名称 | 参数说明 | 最小值 | 典型值 | 最大值 | 单位 |
|--------------------|-------------------|-----|-----|------|-----|
| F _{HSE} | 外部 HSE 振荡器频率 | | 32 | | MHz |
| T _{SUHSE} | 外部 HSE 振荡器启动到可用时间 | 80 | 200 | 500 | uS |
| T _{STHSE} | 外部 HSE 振荡器启动到稳定时间 | 200 | 500 | 3000 | uS |

表 20-6 低速振荡器 LSI 和 LSE

| 名称 | 参数说明 | 最小值 | 典型值 | 最大值 | 单位 |
|-------------------|-------------------|-----|-----|-----|----|
| F _{LSIR} | 内部 LSI 振荡器频率（校准前） | 20K | 32K | 48K | Hz |

| | | | | | | |
|--------------------|-----------------------|-------------|-------|-------|-------|----|
| F _{LSI} | 内部 LSI 频率（应用软件运行时校准后） | | 32726 | 32768 | 32810 | Hz |
| A _{LSI} | LSI 振荡器精度 （软件校准后） | TA=-40℃~85℃ | | 0.1 | 0.5 | % |
| | | TA=0℃~60℃ | | 0.04 | 0.2 | % |
| T _{SULSI} | 内部 LSI 振荡器启动到稳定时间 | | | 40 | 100 | μs |
| T _{SULSE} | 外部 LSE 振荡器启动到可用时间 | | 100 | 300 | 1500 | ms |
| T _{STLSE} | 外部 LSE 振荡器启动到稳定时间 | | 500 | 1500 | 5000 | ms |

表 20-7 PLL 特性

| 名称 | 参数说明 | 最小值 | 典型值 | 最大值 | 单位 |
|--------------------|--------------------------|-----|-----|-----|-----|
| F _{PLL} | PLL 倍频输出时钟（CK32M * 15 倍） | | 480 | | MHz |
| T _{PLLLK} | PLL 锁相时间 | | 15 | 30 | μs |

20.5 时间参数

测试条件：TA=25℃，VDD33=V1033=3.3V，Fsys=6.4MHz。

表 20-8 时间参数

| 名称 | 参数说明 | 最小值 | 典型值 | 最大值 | 单位 | |
|-------------------|--------------------|------|-------------------------|-------------------------|-------------------------|----|
| T _{rpor} | 电源上电复位 RPOR 后的复位延时 | 11 | 15 | 20 | ms | |
| T _{rst} | RST#有效信号宽度 | | 100 | | ns | |
| T _{mr} | 外部复位 MR 后的复位延时 | 2 | 8 | 18 | μs | |
| T _{sr} | 软件复位 SR 后的复位延时 | 2 | 8 | 18 | μs | |
| T _{wtr} | 看门狗复位 WTR 后的复位延时 | 10 | 12 | 18 | μs | |
| T _{wak} | 从低功耗状态 退出的唤醒时间 | 空闲模式 | 0.6 | 1 | 3 | μs |
| | | 暂停模式 | T _{SUHSE} +1 | T _{SUHSE} +80 | T _{SUHSE} +150 | μs |
| | | 睡眠模式 | T _{SUHSE} +1 | T _{SUHSE} +300 | T _{SUHSE} +400 | μs |
| | | 下电模式 | T _{SUHSE} +0.4 | T _{SUHSE} +1 | T _{SUHSE} +5 | ms |

注：上表中延时参数均是基于 T_{sys} 的倍数，降低主频将增加延时。

上表中延时参数是基于使用外部 HSE 时钟源，如果睡眠期间使用外部 HSE 时钟源，那么表中暂停模式/睡眠模式/下电模式的延时参数 T_{wak} 均额外增加约 0.2~1ms（启动到可用 T_{SUHSE}）。

20.6 其它参数

测试条件：TA=25℃，VDD33=V1033=3.3V，Fsys=16MHz。

表 20-9 其它参数

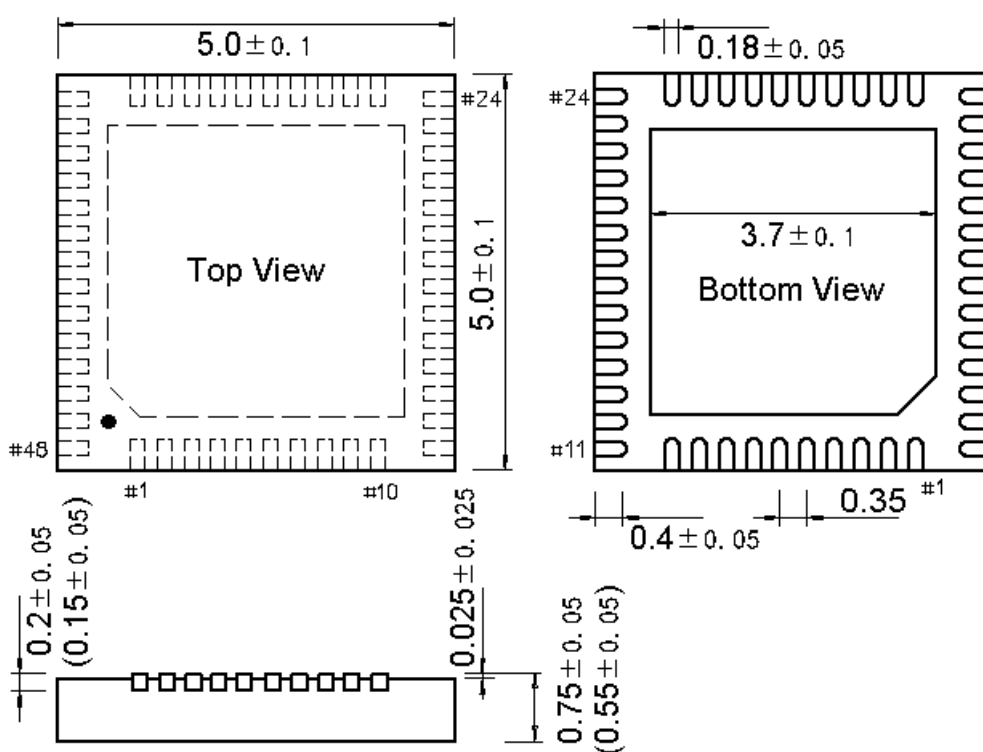
| 名称 | 参数说明 | 最小值 | 典型值 | 最大值 | 单位 |
|-------------------|--|----------------|------|-----------|-------|
| R _{ts} | TS 温度传感器的测量范围 | -40 | | 85 | ℃ |
| A _{TSC} | 温度传感器经软件校准后的测量误差 | | ±10 | | ℃ |
| T _{FRER} | Flash-ROM 的单个扇区擦除操作时间 | 6 | 16 | 30 | ms |
| T _{FRPG} | Flash-ROM 的单个字编程操作时间 | 1 | 2 | 4 | ms |
| N _{EPCE} | Flash-ROM 的擦写次数 erase/program cycle endurance | 5~45℃ | 100K | 1000K（抽测） | times |
| | | -40~85℃ | 50K | 200K（抽测） | |
| T _{DR} | Flash-ROM 的数据保持能力 | 20 | | | years |
| V _{ESD} | I/O 输入或者输出 引脚上的 ESD 耐压 | 天线 ANT | 2K | 4K（抽测） | V |
| | | I/O 引脚：PA 和 PB | 4K | 6K（抽测） | V |

第 21 章 封装

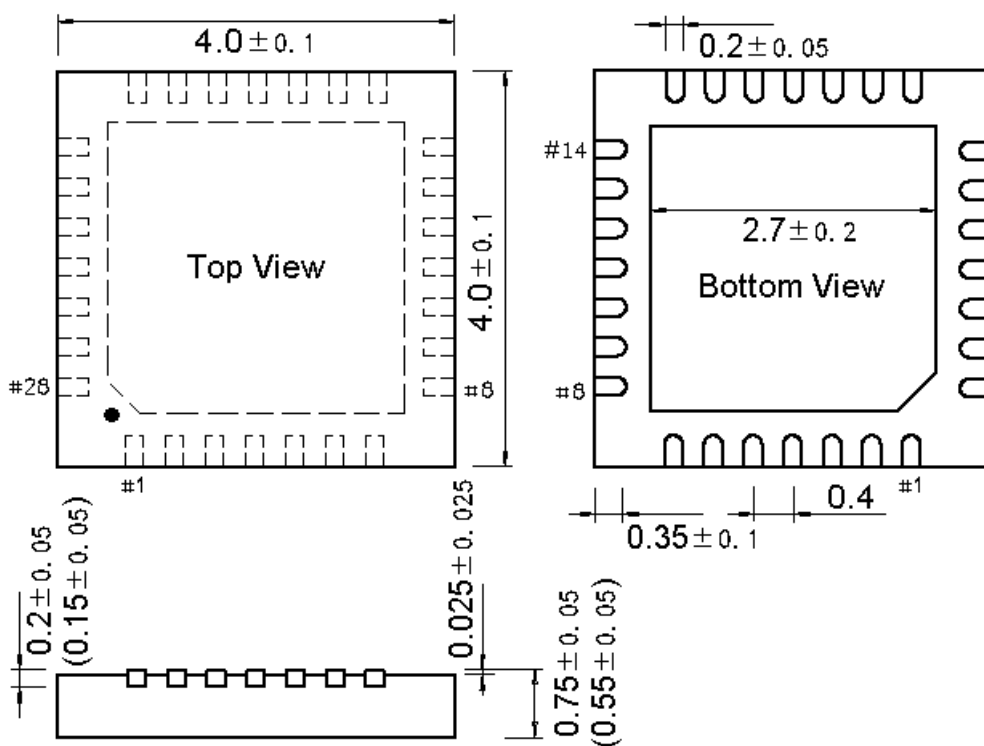
芯片封装

| 封装形式 | 塑体宽度 | 引脚间距 | | 封装说明 | 订货型号 |
|-------|-------|--------|---------|------------|--------|
| QFN48 | 5*5mm | 0.35mm | 13.8mil | 方形无引线 48 脚 | CH583M |
| QFN48 | 5*5mm | 0.35mm | 13.8mil | 方形无引线 48 脚 | CH582M |
| QFN28 | 4*4mm | 0.4mm | 15.7mil | 方形无引线 28 脚 | CH582F |
| QFN28 | 4*4mm | 0.4mm | 15.7mil | 方形无引线 28 脚 | CH581F |

说明：尺寸标注的单位是 mm（毫米），引脚中心间距是标称值，除此之外的尺寸误差不大于±0.2mm。



QFN48_5X5



QFN28_4X4

第 22 章 修改记录

| 版本 | 日期 | 说明 |
|------|------------|--|
| V0.9 | 2020.05.11 | 初版发行 |
| V1.0 | 2020.12.22 | 增加 3.4.2 和 3.4.3 节并修改 3.4.1 中断和 3.4.4 系统计数器说明，微调蓝牙接收灵敏度，提示触摸检测支持驱动屏蔽，微调睡眠电流 |
| V1.1 | 2021.06.22 | 4.1 节 R8_GLOB_RESET_KEEP 特性变化，在 shutdown/GRWSR 时将复位 |
| V1.2 | 2021.08.10 | 修正 20.2 节型号笔误，替换 QFN48 封装图（厚度默认为 0.75） |
| V1.3 | 2021.09.23 | 更新 RB_WAKE_DLY_MOD，更新封装形式加注尺寸 |
| V1.4 | 2022.01.21 | 修正 I2C_CTRL 笔误，限制最高+6dBm 发送功率，PB12 和 PB13 不支持 5V，辅助电源调整控制寄存器 R16_AUX_POWER_ADJ 如有写操作需 16 位 |
| V1.5 | 2022.05.24 | CH583M 存储容量更新为与 CH582M 相同 |
| V1.6 | 2022.06.27 | 修正笔误：温度传感器采样需设置 RB_ADC_BUF_EN 为 0 |
| V1.7 | 2022.10.18 | 修正 RB_XT32K_C_LOAD 描述，外部 HSE 频率删除非 32MHz |
| V1.8 | 2023.12.27 | 修改 VINTA 引脚说明中的电容值建议，删减 4.4 节 FLASH 擦写步骤 |
| | | |

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for [32-bit Microcontrollers - MCU category](#):

Click to view products by [Qin Heng manufacturer](#):

Other Similar products are found below :

[MCF51AC256AVFUE](#) [MCF51AC256BCFUE](#) [MCF51AC256BVFUE](#) [MB91F464AAPMC-GSE2](#) [R5S726B0D216FP#V0](#) [MB91F248PFV-GE1](#) [MB91243PFV-GS-136E1](#) [SAK-TC1782F-320F180HR BA](#) [TC364DP64F300WAAKXUMA1](#) [R5F566NNDDFP#30](#)
[R5F566NNDDFC#30](#) [R5F566NNDDBD#20](#) [MC96F8216ADBN](#) [A96G181HDN](#) [A96G140KNN](#) [A96G174FDN](#) [A31G213CL2N](#)
[A96G148KNN](#) [A96G174AEN](#) [AC33M3064TLBN-01](#) [V3s](#) [T3](#) [A40i-H](#) [V526](#) [A83T](#) [R11](#) [V851s](#) [A133](#) [V833](#) [F1C100S](#) [T3L](#) [T507](#) [A33](#)
[A63](#) [T113-i](#) [H616](#) [V853](#) [V533](#) [R16-J](#) [V536-H](#) [A64-H](#) [V831](#) [V3LP](#) [T113-S3](#) [F1C200S](#) [F133-A](#) [R128-S2](#) [D1-H](#) [ADUCM360BCPZ128-TR](#)
[APT32S003F8PT](#)