

## 概述

CH579 是集成 BLE 无线通讯的 ARM 内核 32 位微控制器。片上集成低功耗蓝牙 BLE 通讯模块、以太网控制器及收发器、全速 USB 主机和设备控制器及收发器、段式 LCD 驱动模块、ADC、触摸按键检测模块、RTC 等丰富的外设资源。

## 功能

- **内核 Core:**
  - 32 位 ARM Cortex-M0 内核
  - 最高 40MHz 系统主频, 最低 32KHz 主频
- **257K 字节非易失存储 FlashROM:**
  - 250KB 用户应用程序存储区 CodeFlash
  - 2KB 用户非易失数据存储区 DataFlash
  - 4KB 系统引导程序存储区 BootLoader
  - 1KB 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
- **32K 字节易失数据存储 SRAM:**
  - 16KB 仅主电源供电的存储区 RAM16K
  - 14KB 双电源供电的睡眠保持存储区 RAM14K
  - 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, 范围 2.1V~3.6V
  - 内置 DC-DC 转换, 0dBm 发送功率时电流 6mA
  - 空闲模式 Idle: 1.2mA
  - 暂停模式 Halt: 420uA
  - 睡眠模式 Sleep: 0.6uA~2.0uA 多档
  - 下电模式 Shutdown: 0.2uA~1.3uA 多档
  - 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
  - 单端 RF 接口, 无需外部电感, 简化板级设计
  - 接收灵敏度-93dBm, 可编程+3dBm 发送功率
  - BLE 符合 Bluetooth Low Energy 4.2 规范
  - 在 0dBm 发送功率时无线通讯距离约 100 米
  - 可选多种形状和大小的 PCB 板载天线
  - 提供优化的协议栈和应用层 API, 支持组网
- **以太网 Ethernet:**
  - 内置 MAC 控制器, 支持帧过滤, 支持 DMA
  - 集成 10Mbps 收发器 PHY, 内置 50Ω 匹配电阻
  - 通讯距离 200 米, 支持自动协商, 支持节能
- **通用串行总线 USB:**
  - 内置 USB 控制器和 DMA, 支持 64 字节数据包
  - 集成 USB 2.0 全速收发器 PHY, 无需外围
  - 支持全/低速的 Host 主机和 Device 设备模式
  - 支持 USB type-C 主从/电流检测
- **实时时钟 RTC:** 支持定时和触发两种模式
- **段式 LCD:** 支持 96 点 (24×4) LCD 面板
- **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, 最高通讯波特率可达 5Mbps
  - UART0 支持 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
- **LED 点阵屏接口:** 支持 1/2/4 路数据线
- **时钟:** 内置 32MHz 和 32KHz 时钟, 内置 PLL
- **8 位被动并口**
- **温度传感器 TS**
- **通用输入输出端口 GPIO:**
  - 40 个 GPIO, 其中 4 个支持 5V 信号输入
  - 32 个中断输入, 32 个唤醒输入
- **封装形式: QFN48\_5X5、QFN28\_4X4**

# 第 1 章 引脚信息

## 1.1 引脚排列

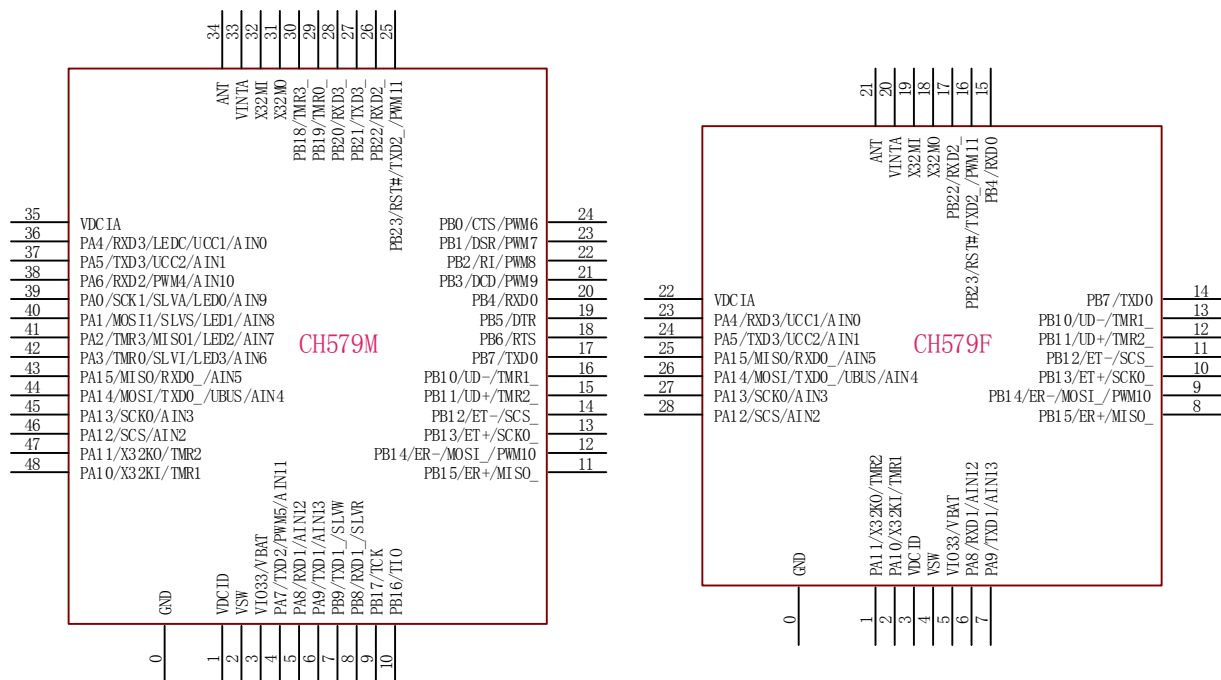


图 1-1 CH579M(QFN48\_5X5)和CH579F(QFN28\_4X4)封装引脚排列

## 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	-	公共接地端，电压 0V 参考点。
1	3	VDCID	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，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 22uH 或 33uH 电感连接到 VDCID，不启用 DC-DC 时可以直连 VDCID。
3	5	VI033	电源	VBAT	I/O 和 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，不启用 DC-DC 时建议不小于 1uF。
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。 AIN11：ADC 模拟信号输入通道 11。
5	6	PA8	I/O/A	RXD1 /AINT12	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	TXD1 /AINT13	PA9：通用双向数字 I/O 引脚。 TXD1：UART1 串行数据输出。 AIN13：ADC 模拟信号输入通道 13。
7	无	PB9	I/O	TXD1_ /SLVW	PB9：通用双向数字 I/O 引脚。 TXD1_：UART1 的 TXD1 引脚映射。

					SLVW: 被动并口的写控制信号输入, 低电平有效。
8	无	PB8	I/O	RXD1_ /SLVR	PB8: 通用双向数字 I/O 引脚。 RXD1_: UART1 的 RXD1 引脚映射。 SLVR: 被动并口的读控制信号输入, 低电平有效。
9	无	PB17	I/O	TCK	PB17: 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。
10	无	PB16	I/O	TIO	PB16: 通用双向数字 I/O 引脚。 TIO: 两线仿真调试接口的串行数据输入输出, 内置上拉。
11	8	PB15	I/O/A /5VT	ER+ /MISO_	PB15: 通用双向数字 I/O 引脚。 ER+: 以太网接收 RX+信号。 MISO_: SPI0 的 MISO 引脚映射。
12	9	PB14	I/O/A /5VT	ER- /MOSI_ /PWM10	PB14: 通用双向数字 I/O 引脚。 ER-: 以太网接收 RX-信号。 MOSI_: SPI0 的 MOSI 引脚映射。 PWM10: 脉宽调制输出通道 10。
13	10	PB13	I/O/A /5VT	ET+ /SCK0_	PB13: 通用双向数字 I/O 引脚。 ET+: 以太网发送 TX+信号。 SCK0_: SPI0 的 SCK 引脚映射。
14	11	PB12	I/O/A /5VT	ET- /SCS_	PB12: 通用双向数字 I/O 引脚。 ET-: 以太网发送 TX-信号。 SCS_: SPI0 的 SCS 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	PB7: 通用双向数字 I/O 引脚。 TXD0: UART0 串行数据输出。
18	无	PB6	I/O	RTS	PB6: 通用双向数字 I/O 引脚。 RTS: UART0 的 MODEM 输出信号, 请求发送。
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	PB4: 通用双向数字 I/O 引脚。 RXD0: UART0 串行数据输入。
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。
22	无	PB2	I/O	RI /PWM8	PB2: 通用双向数字 I/O 引脚。 RI: UART0 的 MODEM 输入信号, 振铃指示。 PWM8: 脉宽调制输出通道 8。
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。
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# /TXD2_	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。

				/PWM11	TXD2_: UART2 的 TXD2 引脚映射。 PWM11: 脉宽调制输出通道 11。
26	17	PB22	I/O	RXD2_	PB22: 通用双向数字 I/O 引脚。 RXD2_: UART2 的 RXD2 引脚映射。
27	无	PB21	I/O	TXD3_	PB21: 通用双向数字 I/O 引脚。 TXD3_: UART3 的 TXD3 引脚映射。
28	无	PB20	I/O	RXD3_	PB20: 通用双向数字 I/O 引脚。 RXD3_: UART3 的 RXD3 引脚映射。
29	无	PB19	I/O	TMRO_	PB19: 通用双向数字 I/O 引脚。 TMRO_: 定时器 0 的 TMRO 引脚映射。
30	无	PB18	I/O	TMR3_	PB18: 通用双向数字 I/O 引脚。 TMR3_: 定时器 3 的 TMR3 引脚映射。
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, 建议 2.2uF (未用 DC-DC 时可选 1uF, 略省电但降低 BLE 灵敏度)。
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 0.1uF, 建议直连 VDC1D。
36	23	PA4	I/O/A	RXD3 /LEDC /UCC1 /AIN0	PA4: 通用双向数字 I/O 引脚。 RXD3: UART3 串行数据输入。 LEDC: LED 屏接口串行时钟输出。 UCC1: USB type-C 双向配置通道 1。 AIN0: ADC 模拟信号输入通道 0。
37	24	PA5	I/O/A	TXD3 /UCC2 /AIN1	PA5: 通用双向数字 I/O 引脚。 TXD3: UART3 串行数据输出。 UCC2: USB type-C 双向配置通道 2。 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。 AIN10: ADC 模拟信号输入通道 10。
39	无	PA0	I/O/A	SCK1 /SLVA /LED0 /AIN9	PA0: 通用双向数字 I/O 引脚。 SCK1: SPI1 串行时钟输出。 SLVA: 被动并口的地址输入, 高电平选择命令和状态端口, 低电平选择数据端口。 LED0: LED 屏接口串行数据输出 0。 AIN9: ADC 模拟信号输入通道 9。
40	无	PA1	I/O/A	MOSI1 /SLVS /LED1 /AIN8	PA1: 通用双向数字 I/O 引脚。 MOSI1: SPI1 串行数据输出。 SLVS: 被动并口的片选控制输入, 低电平有效。 LED1: LED 屏接口串行数据输出 1。 AIN8: ADC 模拟信号输入通道 8。
41	无	PA2	I/O/A	TMR3 /MISO1 /LED2 /AIN7	PA2: 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。 MISO: SPI1 串行数据输入, 2 线模式下复用于输出。 LED2: LED 屏接口串行数据输出 2。 AIN7: ADC 模拟信号输入通道 7。
42	无	PA3	I/O/A	TMRO	PA3: 通用双向数字 I/O 引脚。

				/SLVI /LED3 /AIN6	TMRO: 定时器 0 的捕捉输入 0 和 PWM 输出通道 0。 SLVI: 被动并口的中断请求输出, 低电平有效。 LED3: LED 屏接口串行数据输出 3。 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_ /UBUS /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。 UBUS: USB type-C 总线电压检测输入。 AIN4: ADC 模拟信号输入通道 4。
45	27	PA13	I/O/A	SCK0 /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。 AIN3: ADC 模拟信号输入通道 3。
46	28	PA12	I/O/A	SCS /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。 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。

#### 复用于段式 LCD 驱动的引脚说明

引脚名称	类型	复用功能	功能描述
PA0~PA3	A	COM0~COM3	驱动段式 LCD 的各个公共端, 根据需要选用部分或全部。
PB0~PB23	A	SEGO~SEG23	驱动段式 LCD 的各个段, 根据需要选用部分或全部。 R16_PIN_ANALOG_IE 寄存器中有 6 位 RB_PIN_SEG*_IE, 每个位同时控制 4 个段引脚, 该位为 0 时 4 个引脚用于数字输入或其它非 LCD 功能, 该位为 1 时 4 个引脚用于 LCD 段驱动。

#### 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。

## 第 2 章 系统结构及存储器

### 2.1 系统结构

下图为 CH579 芯片系统结构框图。

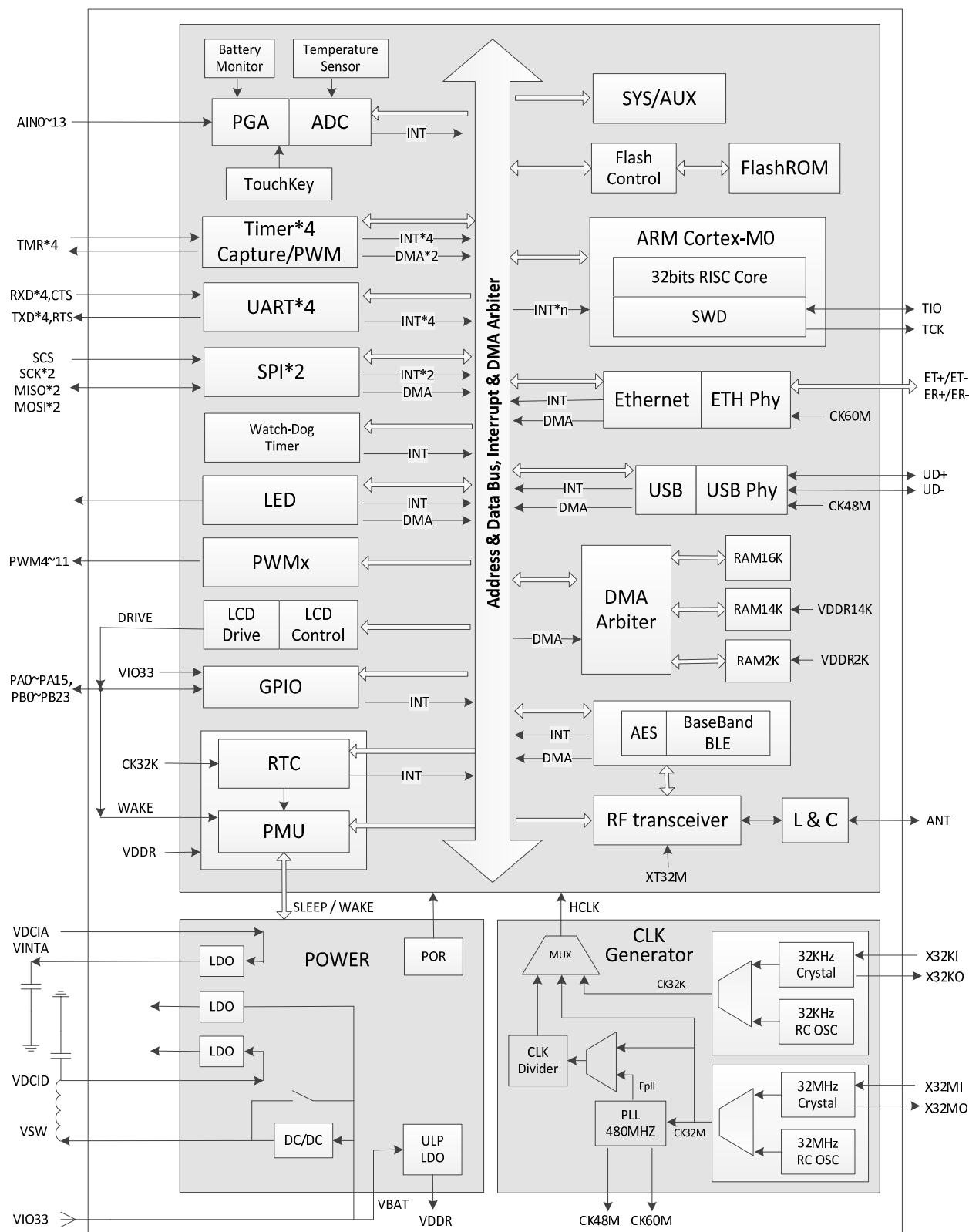


图 2-1 CH579 内部结构框图

## 2.2 存储器映射

CH579 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。

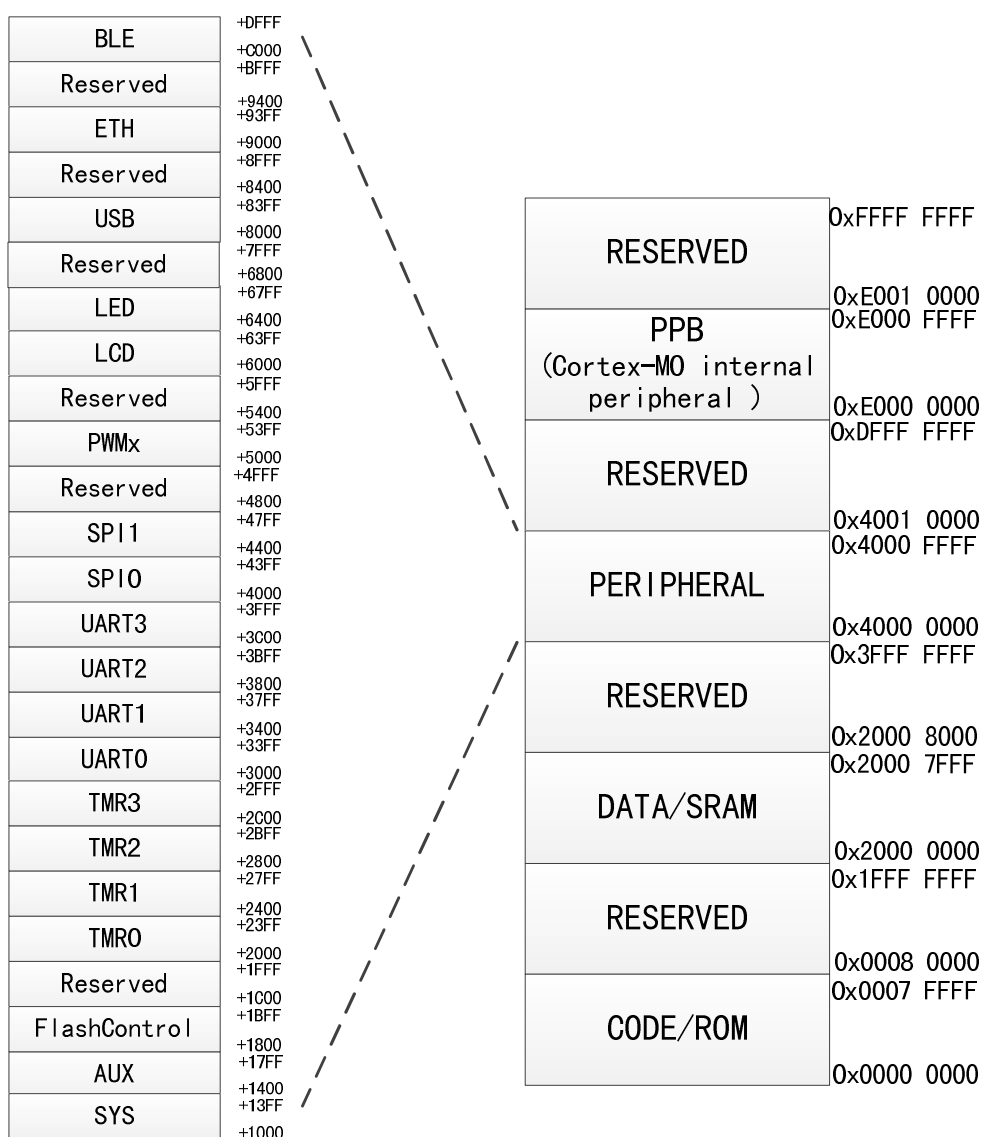


图 2-2 存储器映射图

## 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	257KB, 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
0x40010000-0xDFFFFFFF	保留	-
0xE0000000-0xE000FFFF	PPB	内部外设总线区域
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
0x00000000-0x0003E7FF	用户应用程序存储区 CodeFlash	250KB
0x0003E800-0x0003EFFF	用户非易失数据存储区 DataFlash	2KB
0x0003F000-0x0003FFFF	系统引导程序存储区 BootLoader	4KB
0x00040000-0x000403FF	系统非易失配置信息存储区 InfoFlash	1KB
0x00040400-0x0007FFFF	保留	-

地址 0x00040010 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	000b
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	0FFFFh
位 31~位 28	VALID_SIG	配置信息有效标志，固定值	0101b

### 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
0x20000000-0x20003FFF	仅主电源供电的存储区 RAM16K	16KB
0x20004000-0x200077FF	主+辅双电源供电的可独立保持存储区 RAM14K	14KB
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

CH579 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)	0x4000 1000
2	AUX (ADC/TKEY/PLL 等)	0x4000 1400
3	FlashROM-Control	0x4000 1800
4	TMR0	0x4000 2000
5	TMR1	0x4000 2400
6	TMR2	0x4000 2800
7	TMR3	0x4000 2C00
8	UART0	0x4000 3000
9	UART1	0x4000 3400



10	UART2	0x4000 3800
11	UART3	0x4000 3C00
12	SPI0	0x4000 4000
13	SPI1	0x4000 4400
14	PWMx (PWM4~PWM11)	0x4000 5000
15	LCD	0x4000 6000
16	LED	0x4000 6400
17	USB	0x4000 8000
18	ETH	0x4000 9000
19	Radio:BLE	0x4000 C000 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。
RO	只读。
WO	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）
HSI	内部高频 RC 时钟振荡源（出厂校准后为 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	最高地址的 2KB SRAM
RAM14K	次高地址的 14KB SRAM
0x	以其开头的数字表示 16 进制数
H	以其结束的数字表示 16 进制数
B	以其结束的数字表示 2 进制数

## 第 3 章 中断

### 3.1 中断控制器

系统支持多达 20 组中断信号源，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。

### 3.2 系统嘀嗒 SysTick 校准值

当系统嘀嗒时钟设定为 32MHz，系统嘀嗒校准值固定为 32000，将产生 1ms 时间基准。

### 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

位置	优先级	优先级类型	名称	说明	地址
	-	-	-	保留	0x0000_0000
-15	-3	固定	Reset	复位	0x0000_0004
-14	-2	固定	NMI	不可屏蔽中断	0x0000_0008
-13	-1	固定	Hard fault	所有类型的失效	0x0000_000C
	-	-	-	保留	-
-5	0	可设置	SVCall	通过 SWI 指令的系统服务调用	0x0000_002C
	-	-	-	保留	-
-2	1	可设置	PendSV	可挂起系统服务	0x0000_0038
-1	2	可设置	SysTick	系统嘀嗒定时器	0x0000_003C
0	3	可设置	TMR0	TMR0 定时器 0 中断	0x0000_0040
1	4	可设置	GPIO	GPIO 通用 I/O 中断	0x0000_0044
2	5	可设置	SLAVE	被动并口中断	0x0000_0048
3	6	可设置	SPI0	SPI0 中断	0x0000_004C
4	7	可设置	BLEL	无线模块的 LLE 中断	0x0000_0050
5	8	可设置	BLEB	无线模块的 BB 中断	0x0000_0054
6	9	可设置	USB	USB 中断	0x0000_0058
7	10	可设置	ETH	以太网中断	0x0000_005C
8	11	可设置	TMR1	TMR1 定时器 1 中断	0x0000_0060
9	12	可设置	TMR2	TMR2 定时器 2 中断	0x0000_0064
10	13	可设置	UART0	UART0 异步串口 0 中断	0x0000_0068
11	14	可设置	UART1	UART1 异步串口 1 中断	0x0000_006C
12	15	可设置	RTC	RTC 实时时钟中断	0x0000_0070
13	16	可设置	ADC	ADC 中断	0x0000_0074
14	17	可设置	SPI1	SPI1 中断	0x0000_0078
15	18	可设置	LED	LED 屏控制中断	0x0000_007C
16	19	可设置	TMR3	TMR3 定时器 3 中断	0x0000_0080
17	20	可设置	UART2	UART2 异步串口 2 中断	0x0000_0084
18	21	可设置	UART3	UART3 异步串口 3 中断	0x0000_0088
19	22	可设置	WDOG_BAT	看门狗定时器中断/电池低电压中断	0x0000_008C

## 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 和 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，芯片内部会产生上电复位并延时以等待电源稳定。下图为上电复位过程。

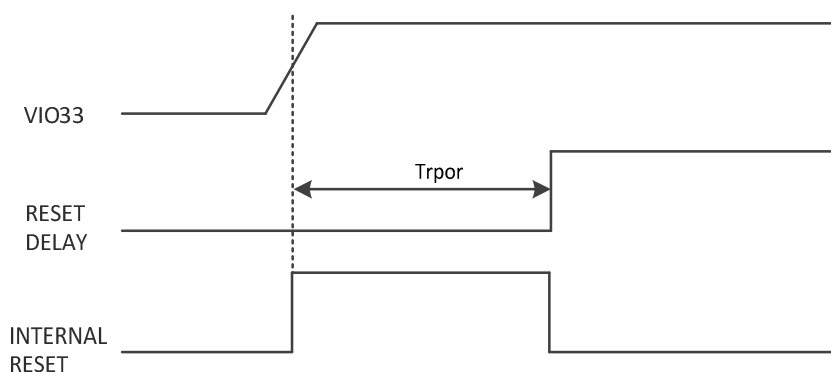


图 4-1 上电复位

#### 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( $T_{rst}$ )时即触发系统进行复位。

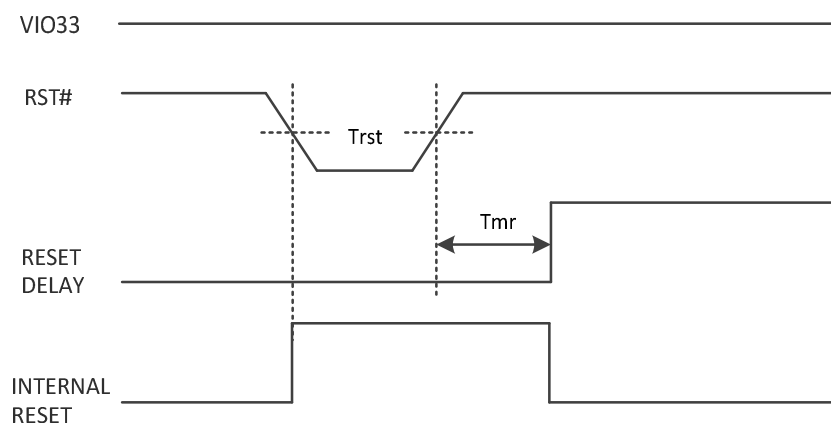


图 4-2 外部复位

#### 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	0x79
R8_SAFE_ACCESS_ID	0x40001042	安全访问 ID 寄存器	0x04
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_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
R8_CFG_FLASH	0x4000104A	FlashROM 配置寄存器	0x0X
R32_FLASH_DATA	0x40001800	FlashROM 数据寄存器	0xFFFFFFFF
R32_FLASH_ADDR	0x40001804	FlashROM 地址寄存器	0xFFFFFFFF
R8_FLASH_COMMAND	0x40001808	FlashROM 命令寄存器	0x00
R8_FLASH_PROTECT	0x40001809	FlashROM 保护控制寄存器	0x00
R16_FLASH_STATUS	0x4000180A	FlashROM 状态寄存器	0x0X00

## 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	固定值 79h, 用于识别芯片。	79h

## 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 04h。	04h

## 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)

位	名称	访问	描述	复位值
[7:5]	Reserved	RO	保留。	000b
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 或 WTR 影响, 仅在 RPOR 和 GRWSM 有效时才能清零: 0: 0x000000; 1: 0x008000 (跳过 ROM 中前 32KB)。	0
3	Reserved	RO	保留。	0
[2:0]	RB_RESET_FLAG	RO	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);	001b

			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。	
--	--	--	--	--

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

## FlashROM 配置寄存器 (R8\_CFG\_FLASH)

位	名称	访问	描述	复位值
7	RB_FLASH_BUSY_EN	RWA	FlashROM 突发访问时忙使能位： 1: 有等待（建议）；0: 无等待。	0
[6:4]	Reserved	RO	保留。	000b
[3:0]	RB_CFG_FLASH_X	RWA	FlashROM 配置数据，写入时必须保持原值不变。	XXXXb

## FlashROM 数据寄存器 (R32\_FLASH\_DATA)

位	名称	访问	描述	复位值
[31:0]	R32_FLASH_DATA	RW	FlashROM 数据寄存器。	XXXXXXXXh

## FlashROM 地址寄存器 (R32\_FLASH\_ADDR)

位	名称	访问	描述	复位值
[31:0]	R32_FLASH_ADDR	RW	FlashROM 地址寄存器。	XXXXXXXXh

## FlashROM 命令寄存器 (R8\_FLASH\_COMMAND)

位	名称	访问	描述	复位值
[7:0]	R8_FLASH_COMMAND	WO	0x9A: FlashROM 编程操作命令； 0xA6: FlashROM 扇区擦除操作命令。	00h

## FlashROM 保护控制寄存器 (R8\_FLASH\_PROTECT)

位	名称	访问	描述	复位值
[7:6]	RB_ROM_WE_MUST_10	WO	必须写 10b，否则无法写入。	10b
[5:4]	Reserved	RO	保留。	00b
3	RB_ROM_CODE_WE	RW	FlashROM 程序存储区 CodeFlash 的擦除/编程使能位： 1: 允许擦除/编程； 0: 擦写保护。	0
2	RB_ROM_DATA_WE	RW	FlashROM 数据存储区 DataFlash 的擦除/编程使能位： 1: 允许擦除/编程； 0: 擦写保护。	0
[1:0]	Reserved	RO	保留。	00b

## FlashROM 状态寄存器 (R16\_FLASH\_STATUS)

位	名称	访问	描述	复位值
[15:11]	Reserved	RO	保留。	00000b
[10:9]	Reserved	RO	保留。	01b
8	RB_ROM_READ_FREE	RO	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
7	Reserved	RO	保留。	0
6	RB_ROM_ADDR_OK	RO	FlashROM 擦除/编程操作地址有效标志，	0

			操作前后都可以进行确认： 1: 地址有效;           0: 地址无效。	
[5:2]	Reserved	R0	保留。	0000b
1	RB_ROM_CMD_ERR	R0	FlashROM 命令响应错误： 1: 未知命令;           0: 命令被接受。	0
0	RB_ROM_CMD_TOUT	R0	FlashROM 操作结果： 1: 超时;                0: 成功。	0

#### 4.4 Flash-ROM 操作步骤

##### 1. 擦除 Flash-ROM, 将目标扇区中全部数据位变为 1:

- (1)、设置 R8\_FLASH\_PROTECT 寄存器, 开启擦除/编程使能 RB\_ROM\_DATA\_WE 或 RB\_ROM\_CODE\_WE 位, 对应 DataFlash 或 CodeFlash, InfoFlash 则需要 RB\_ROM\_DATA\_WE 和 RB\_ROM\_CODE\_WE 都开启;
- (2)、设置地址寄存器 R32\_FLASH\_ADDR, 写入 32 位地址, 低 8 位无效, 一次擦除整块 512 字节;
- (3)、设置命令寄存器 R8\_FLASH\_COMMAND, 写入 0A6H, 执行扇区擦除操作, 操作期间 MCU 自动暂停运行;
- (4)、操作完成后程序恢复运行, 此时查询状态寄存器 R16\_FLASH\_STATUS 查看此次操作状态; 如果要擦除多个扇区, 则循环 (2) ~ (4) 步骤;
- (5)、再次设置 R8\_FLASH\_PROTECT 寄存器, 关闭擦除/编程使能控制位 (RB\_ROM\_DATA\_WE=0 或 RB\_ROM\_CODE\_WE=0)。

##### 2. 写 Flash-ROM, 将目标字中的部分数据位从 1 变为 0 (无法将位数据从 0 变为 1):

- (1)、设置 R8\_FLASH\_PROTECT 寄存器, 开启擦除/编程使能 RB\_ROM\_DATA\_WE 或 RB\_ROM\_CODE\_WE 位, 对应 DataFlash 或 CodeFlash, InfoFlash 则需要 RB\_ROM\_DATA\_WE 和 RB\_ROM\_CODE\_WE 都开启;
- (2)、设置地址寄存器 R32\_FLASH\_ADDR, 写入 32 位地址, 需要 4 字节对齐;
- (3)、设置数据寄存器 R32\_FLASH\_DATA 为要写入的 4 字节数据;
- (4)、设置命令寄存器 R8\_FLASH\_COMMAND, 写入 09AH, 执行编程/写操作, 操作期间 MCU 自动暂停运行;
- (5)、操作完成后程序恢复运行, 此时查询状态寄存器 R16\_FLASH\_STATUS 查看此次操作状态; 如果要写多个数据, 则循环 (2) ~ (4) 步骤;
- (5)、再次设置 R8\_FLASH\_PROTECT 寄存器, 关闭擦除/编程使能控制位 (RB\_ROM\_DATA\_WE=0 或 RB\_ROM\_CODE\_WE=0)。

##### 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

CH579 内置有电源管理单元 PMU。系统电源从 VI033 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、USB、以太网和 LED 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。

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，CH579 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 14KB 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。

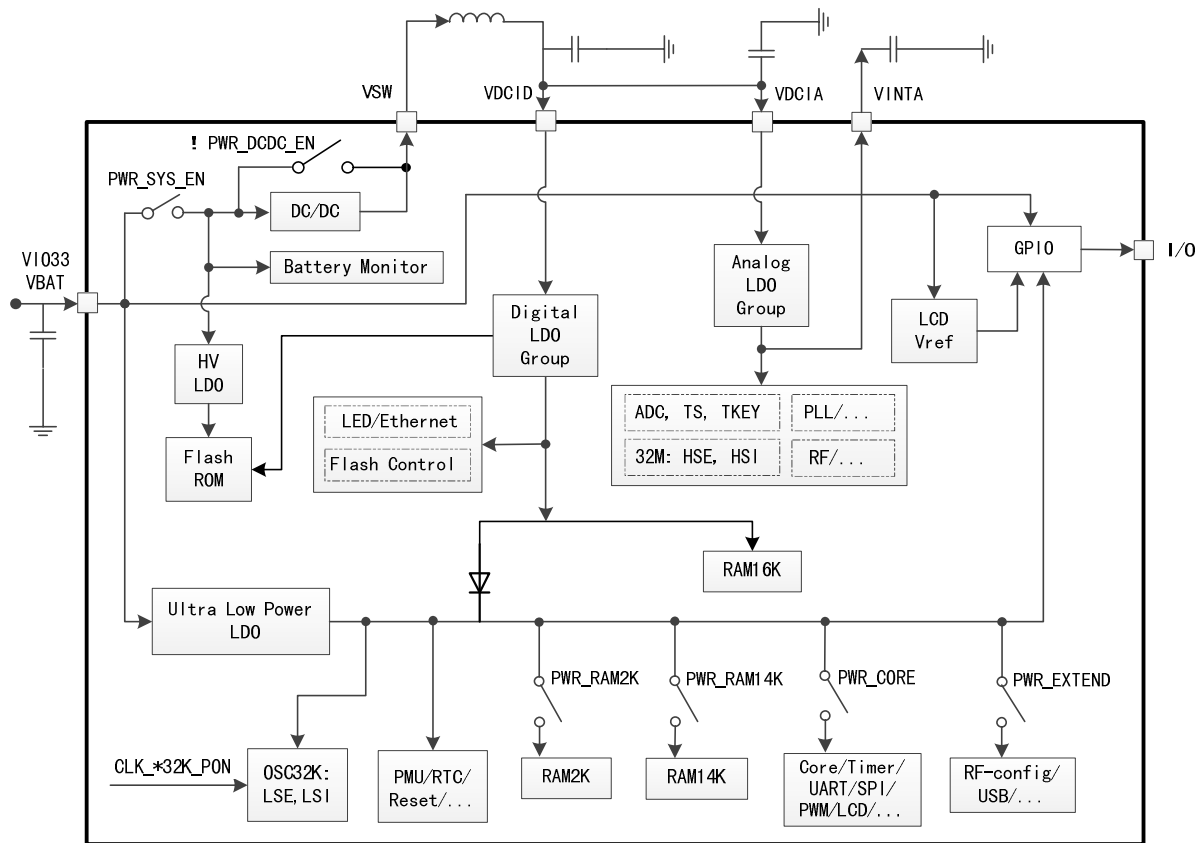


图 5-1 电源系统

### 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	0x08

R16_POWER_PLAN	0x40001020	睡眠电源管理寄存器	0x01DF
R8_AUX_POWER_ADJ	0x40001022	辅助电源调整控制寄存器	0xXX
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	0x01
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_TMR0	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	RB_SLP_CLK_LED	RWA	LED 控制器时钟源： 1: 关闭; 0: 开启。	0
5	RB_SLP_CLK_ETH	RWA	ETH 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_LCD	RWA	LCD 控制器时钟源： 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
[7:6]	Reserved	RO	保留。	00b
5	RB_SLP_BAT_WAKE	RWA	使能电池低压事件唤醒系统:	1

			1: 使能; 0: 关闭。	
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_ETH_WAKE	RWA	使能以太网事件唤醒系统: 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
[7:6]	Reserved	RO	保留。	00b
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 (RAM16K+RAM14K) 的时钟控制: 1: 关闭; 0: 开启。	0
3	RB_SLP_ROM_PWR_DN	RWA	在暂停模式下, FlashROM 掉电使能: 1: 暂停模式下使 ROM 进入停用状态; 0: 暂停模式下使 ROM 保持待机状态。	1
2	Reserved	RO	保留。	0
1	RB_SLP_ETH_PWR_DN	RWA	以太网收发器 ETH PHY 掉电使能: 1: 掉电/禁用; 0: 保持供电。	0
0	Reserved	RO	保留。	0

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。	0000b
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_RAM14K	RWA	RAM14K 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	Reserved	RO	保留。	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 (R8\_AUX\_POWER\_ADJ)

位	名称	访问	描述	复位值
[7:6]	Reserved	RO	保留。	00b
[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: 0.908V; 001: 0.931V; 010: 0.954V; 011: 0.977V; 100: 1.000V; 101: 1.023V; 110: 1.046V; 111: 1.069V。	XXXb (100b)

#### 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	Reserved	RO	保留。	0
0	RB_BAT_DET_EN	RWA	电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。	0

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	设置低电压及超低电压的检测阈值： （超低电压参考阈值，低电压参考阈值） 00: 1.97V , 2.25V; 01: 2.05V , 2.33V; 10: 2.13V , 2.41V; 11: 2.21V , 2.49V。	01b

#### 电池状态寄存器 (R8\_BAT\_STATUS)

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

[7:2]	Reserved	R0	保留。	000000b
1	RB_BAT_STAT_LOW	R0	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	R0	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，可将 FlashROM 进入停机模式，进一步降低功耗。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、RAM14K、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、RAM14K 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。	设置内核控制位 SCR=0，设置唤醒条件后执行 __WFI() 或 __WFE()。	I/O 或 RTC 或 BAT 或 USB 或 ETH	1.15mA~ 1.5mA
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/HSI 停止)。可以选择 FlashROM 待机或停用。	设置内核控制位 SCR=1，设置唤醒条件后执行 __WFI() 或 __WFE()。	I/O 或 RTC 或 BAT 或 USB 或 ETH	420uA~ 470uA
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、RAM14K、USB 和 RF 配置可以选择是否维持供电。	设置内核控制位 SCR=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	0.6uA~ 2.0uA
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、RAM14K 可以选择是否维持供电，用于保持数据。	设置内核控制位 SCR=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~ 1.3uA

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	RAM14K	CK32K	CORE	EXTEND	功耗（仅供参考）
维持供电	系统电源	数据区	数据区	LSE/LSI 二选一	MO 内核和	USB 和	PMU 和 RTC 寄存器

的功能	VSW	2KB	14KB	RTC 唤醒	基本外设	RF 配置	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~0.6uA
	0	1	0	1	0	0	0.8uA~0.9uA
睡眠模式 常用配置	0	1	0	0	1	0	0.6uA
	0	1	0	1	1	0	0.9uA~1.0uA
	0	0	1	0	1	0	1.1uA
	0	0	1	1	1	0	1.4uA~1.5uA
	0	1	1	1	1	1	1.8uA~2.0uA

## 5.4 DC-DC 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)、延时至少 10uS；
- (4)、再次进入安全访问模式：寄存器 R8\_SAFE\_ACCESS\_SIG 先写入 0x57，再写入 0xA8；
- (5)、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<sub>sys</sub>)

- HSE 或 HSI 原始时钟 CK32M。
- HSE 或 HSI 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

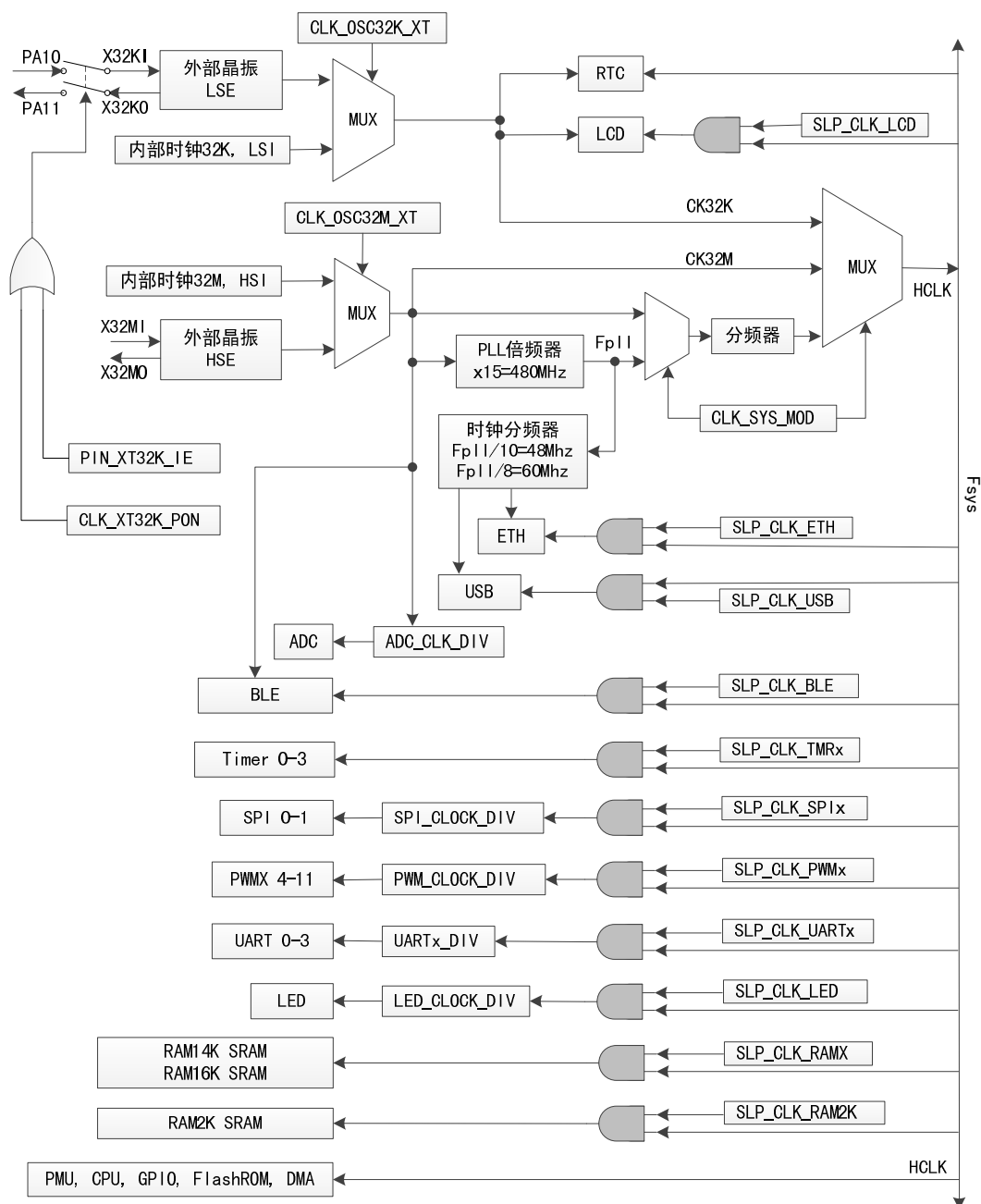


图 6-1 时钟树框图

上图是系统内部的时钟树结构，其中的 RTC 和 LCD 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 和以太网的数据传输依靠 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	0x05
R8_HFCK_PWR_CTRL	0x4000100A	高频时钟模块电源控制寄存器	0x0C
R16_INT32K_TUNE	0x4000102C	内部 32KHz 时钟校准寄存器	0x0200
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_INT32M_CALIB	0x4000104C	内部 32MHz 时钟校准寄存器	0xXX
R8_XT32M_TUNE	0x4000104E	外部 32MHz 时钟谐振控制寄存器	0x31
R16_OSC_CAL_CNT	0x40001050	振荡器频率校准计数值寄存器	0xFFFF
R8_OSC_CAL_CTRL	0x40001052	振荡器频率校准控制寄存器	0x02
R8_PLL_CONFIG	0x4000104B	PLL 配置寄存器	0xX0
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	RB_X0_DI	RO	X32M0 引脚的输入状态采样值。	0
[14:10]	Reserved	RO	保留。	00000b
9	RB_CLK_OSC32M_XT	RWA	CK32M（32MHz）时钟源选择位： 1：外部 32MHz 振荡器； 0：内部 32MHz 振荡器。	0
8	Reserved	RO	保留。	0
[7:6]	RB_CLK_SYS_MOD	RWA	HCLK 系统时钟源模式选择： 00：CK32M（默认 32MHz）进行分频；	00b



			01: PLL (默认 480MHz) 进行分频; 10: CK32M (默认 32MHz) 作为 HCLK; 11: CK32K (默认 32KHz) 作为 HCLK。	
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:

$F_{ck32m} = RB\_CLK\_OSC32M\_XT ? XT\_32MHz : RC\_32MHz;$

$F_{ck32k} = RB\_CLK\_OSC32K\_XT ? XT\_32KHz : RC\_32KHz;$

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

$F_{sys} = RB\_CLK\_SYS\_MOD[1] ? ( RB\_CLK\_SYS\_MOD[0] ? F_{ck32k} : F_{ck32m} ) :$   
 $( ( RB\_CLK\_SYS\_MOD[0] ? F_{pll} : F_{ck32m} ) / RB\_CLK\_PLL\_DIV );$

上电默认值  $F_{sys} = F_{ck32m} / RB\_CLK\_PLL\_DIV = 32MHz / 5 = 6.4MHz;$

$F_{sys}$  范围: 32KHz, 1MHz~16MHz, 32MHz, 15MHz~44MHz

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
[7:6]	Reserved	RWA	保留, 必须写 00。	00b
5	Reserved	RO	保留。	0
4	RB_CLK_PLL_PON	RWA	PLL 电源控制位: 1: 上电; 0: 掉电。	0
3	RB_CLK_INT32M_PON	RWA	内部 32MHz 振荡器电源控制位: 1: 上电; 0: 掉电。	1
2	RB_CLK_XT32M_PON	RWA	外部 32MHz 振荡器电源控制位: 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
[15:10]	Reserved	RO	保留。	000000b
[9:0]	RB_INT32K_TUNE	RWA	内部 RC 32KHz 时钟频率校准值。	1000000000b

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): 0000: 内置负载电容量为 2pF; 其它: 电容量 = RB_XT32K_C_LOAD + 12pF, 0001b~1111b 分别对应约 13pF~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。	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
[6:3]	Reserved	RO	保留。	0000b
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

## 内部 32MHz 时钟校准寄存器 (R8\_INT32M\_CALIB)

位	名称	访问	描述	复位值
[7:0]	R8_INT32M_CALIB	RWA	内部 RC 32MHz 时钟频率校准值。	XXXXXXXXb

## 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	RO	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	RO	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。	01b

## 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
[15:12]	Reserved	RO	保留。	0000b
[11:0]	RB_OSC_CAL_CNT	RO	对 5 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。	XXXh

## 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
[7:2]	Reserved	RO	保留。	000000b
1	RB_OSC_CNT_HALT	RO	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
0	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

## PLL 配置寄存器 (R8\_PLL\_CONFIG)

位	名称	访问	描述	复位值
7	RB_PLL_LOCKED	RO	PLL 锁定状态： 1：正在锁定； 0：当前没有锁定。	X
[6:2]	Reserved	RO	保留。	00000b
[1:0]	RB_PLL_CFG_DAT	RWA	PLL 配置参数。	00b

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_BO	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	R0	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 $\mu$ 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， $T32 = R32\_RTC\_CNT\_32K + DelayTime * 32768$ ，  
如果  $(T32 \& 0xFFFF) \neq 0$ ，那么  $T32 = T32 + 0x10000$ ，  
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，部分引脚具有中断、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；其中 32 个 I/O 管脚 PA[0]~PA[15]、PB[0]~PB[15]具有中断功能，并可实现睡眠唤醒功能。

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：

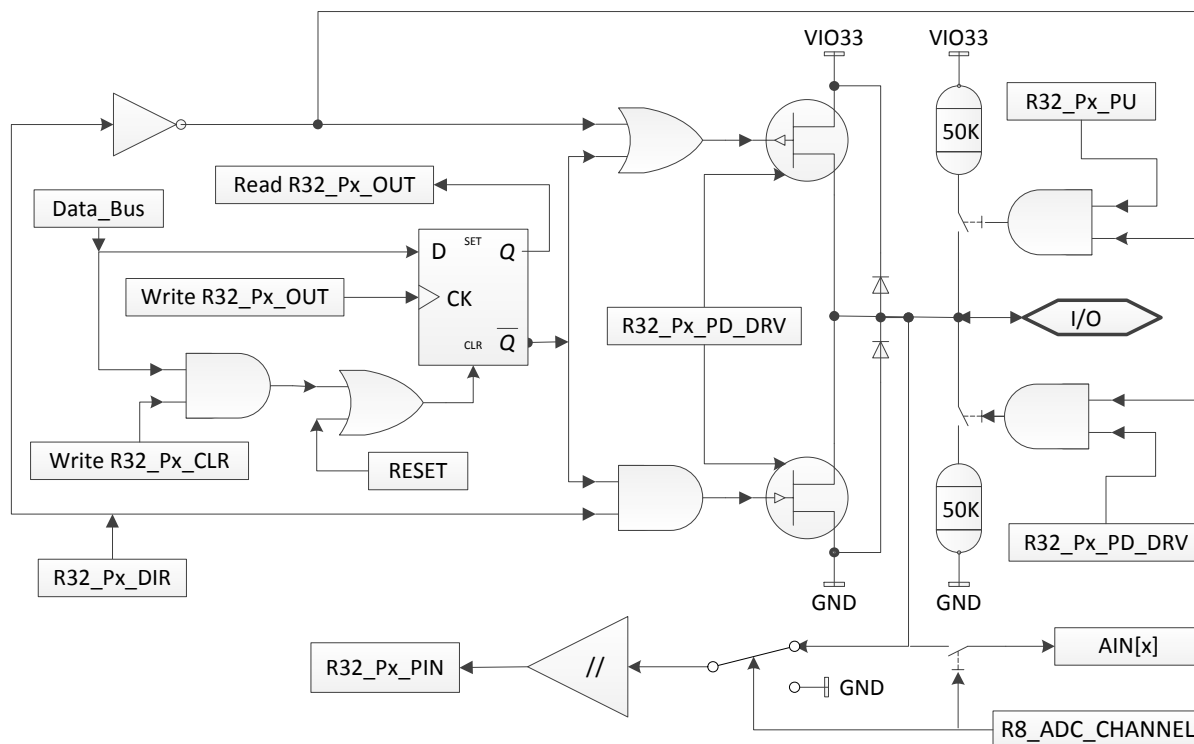


图 7-1 I/O 内部结构框图

### 7.2 外部中断/唤醒

芯片的 32 个 I/O 管脚 (PA[0]~PA[15]、PB[0]~PB[15]) 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：PA0 脚复用为 SCK1/SLVA/LED0，则 SPI1 的时钟功能优先，LED 屏接口串行数据输出 0 功能最低。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 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 7-5 USB

USB 信号引脚	功能配置	GPIO 配置
USB_DM	连接到内部 USB 收发器	浮空输入
USB_DP	连接到内部 USB 收发器	浮空输入

表 7-6 以太网

以太网信号引脚	功能配置	GPIO 配置
ET+/ET-	连接到内部以太网收发器	浮空输入
ER+/ER-	连接到内部以太网收发器	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
SPI0	PA[12]/PA[13]/PA[14]/PA[15]	PB[12]/PB[13]/PB[14]/PB[15]
RXD3/TXD3	PA[4]/PA[5]	PB[20]/PB[21]
RXD2/TXD2	PA[6]/PA[7]	PB[22]/PB[23]
RXD1/TXD1	PA[8]/PA[9]	PB[8]/PB[9]
RXD0/TXD0	PB[4]/PB[7]	PA[15]/PA[14]
TMR3/PWM3/CAP3	PA[2]	PB[18]
TMR2/PWM2/CAP2	PA[11]	PB[11]
TMR1/PWM1/CAP1	PA[10]	PB[10]
TMRO/PWMO/CAPO	PA[3]	PB[19]

## 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
---	----	----	----	-----



[15:12]	Reserved	R0	保留。	0000b
[11:9]	Reserved	R0	保留。	000b
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[8]/PB[9]； 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 PB[18]； 0: TMR3/PWM3/CAP3 映射到 PA[2]。	0
2	RB_PIN_TMR2	RW	TMR2 功能引脚映射选择位： 1: TMR2_/PWM2_/CAP2_映射到 PB[11]； 0: TMR2/PWM2/CAP2 映射到 PA[11]。	0
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[19]； 0: TMRO/PWMO/CAPO 映射到 PA[3]。	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_ADC12_13_IE	RW	ADC/TKEY 12/13 通道引脚数字输入禁用： 1: 关闭 PA8-9 数字输入，可节约功耗； 0: 打开数字输入。	0
11	RB_PIN_ADC10_11_IE	RW	ADC/TKEY 10/11 通道引脚数字输入禁用： 1: 关闭 PA6-7 数字输入，可节约功耗； 0: 打开数字输入。	0

10	RB_PIN_ADC0_1_IE	RW	ADC/TKEY 0/1 通道引脚数字输入禁用： 1: 关闭 PA4-5 数字输入，可节约功耗； 0: 打开数字输入。	0
9	RB_PIN_ADC6_7_IE	RW	ADC/TKEY 7/6 通道引脚数字输入禁用： 1: 关闭 PA2-3 数字输入，可节约功耗； 0: 打开数字输入。	0
8	RB_PIN_ADC8_9_IE	RW	ADC/TKEY 9/8 通道引脚数字输入禁用： 1: 关闭 PA0-1 数字输入，可节约功耗； 0: 打开数字输入。	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_ETH_IE	RW	ETH 引脚使能： 1: PB12-15 为 ETH 通讯引脚，如果 LCD 完全未启用，可将 RB_PIN_SEG12_15_IE 置为 1 以关闭数字输入，可节约功耗； 0: PB12-15 不用于 ETH 通讯。	0
5	RB_PIN_SEG20_23_IE	RW	LCD 20-23 段引脚使能/数字输入禁用： 1: PB20-23 为段驱动，关闭数字输入； 0: 打开数字输入，关闭 LCD 段驱动。	0
4	RB_PIN_SEG16_19_IE	RW	LCD 16-19 段引脚使能/数字输入禁用： 1: PB16-19 为段驱动，关闭数字输入； 0: 打开数字输入，关闭 LCD 段驱动。	0
3	RB_PIN_SEG12_15_IE	RW	LCD 12-15 段引脚使能/数字输入禁用： 1: PB12-15 为段驱动，关闭数字输入； 0: 打开数字输入，关闭 LCD 段驱动。	0
2	RB_PIN_SEG8_11_IE	RW	LCD 8-11 段引脚使能/数字输入禁用： 1: PB8-11 为段驱动，关闭数字输入； 0: 打开数字输入，关闭 LCD 段驱动。	0
1	RB_PIN_SEG4_7_IE	RW	LCD 4-7 段引脚使能/数字输入禁用： 1: PB4-7 为段驱动，关闭数字输入； 0: 打开数字输入，关闭 LCD 段驱动。	0
0	RB_PIN_SEGO_3_IE	RW	LCD 0-3 段引脚使能/数字输入禁用： 1: PB0-3 为段驱动，关闭数字输入； 0: 打开数字输入，关闭 LCD 段驱动。	0

注：如果引脚用于模拟功能（ADC/TouchKey/ETH），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

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

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

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	RO	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	RO	保留。	0000h
[15:8]	R8_PA_PIN_1	RO	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	RO		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	RO	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		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 引脚输出电平状态： 1: 输出高电平；0: 输出低电平。	00h

			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。	
--	--	--	---	--

## 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： 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
[15:8]	R8_PB_PD_DRV_1	RW		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：定时模式或 PWM 模式。	0

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	R0	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	R0	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	R0	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	R0	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： $R32\_TMRx\_FIFO / R32\_TMRx\_CNT\_END = 1/3$

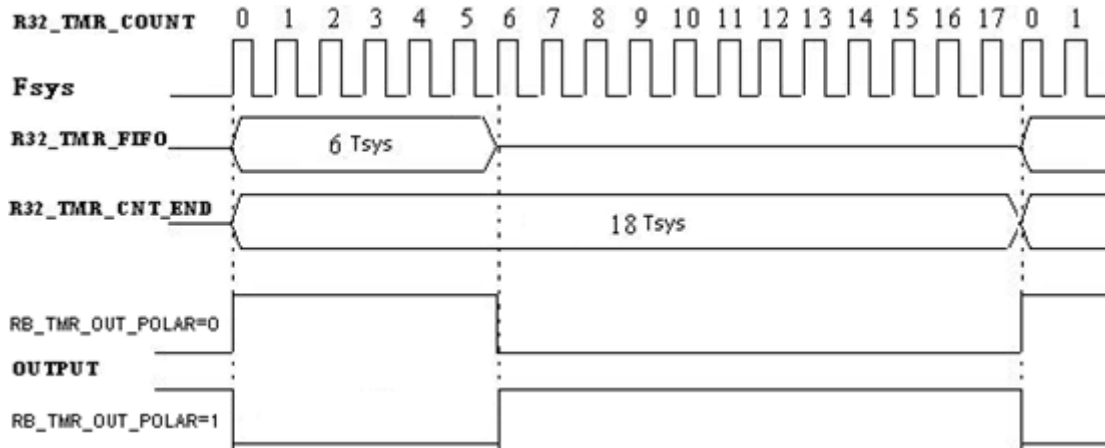


图 8-1 PWM 输出时序图

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：

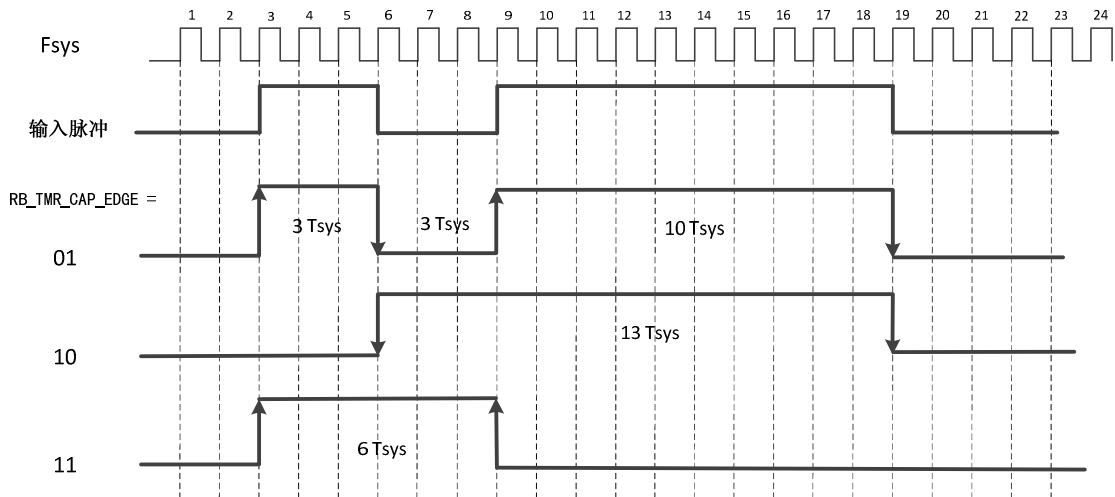


图 8-2 以系统时钟计数捕捉周期

如上图所示，每个时钟周期内采样一次：

当  $RB\_TMR\_CATCH\_EDGE=01b$  时，设置为边沿触发采样，采样到的时间宽度为 3、3、10；

当  $RB\_TMR\_CATCH\_EDGE=10b$  时，设置为下降沿至下降沿采样，采样到的时间宽度为 13；

当  $RB\_TMR\_CATCH\_EDGE=11b$  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  $RB\_TMR\_CAP\_EDGE$  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

芯片提供了 4 组全双工的异步串口，UART0/1/2/3。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 5Mbps 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\_UART<sub>x</sub>\_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):	0

			<p>1: 启用 CTS 和 RTS 硬件自动流控; 0: 禁用 CTS 和 RTS 硬件自动流控。</p> <p>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。</p>	
4	RB_MCR_LOOP	RW	<p>内部回路的测试模式使能(仅 UART0 支持):</p> <p>1: 使能内部回路的测试模式; 0: 禁止内部回路的测试模式。</p> <p>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。</p>	0
3	RB_MCR_OUT2 RB_MCR_INT_OE	RW	<p>串口的中断请求输出控制位:</p> <p>1: 允许发出请求; 0: 禁止。</p>	0
2	RB_MCR_OUT1	RW	<p>用户自定义 MODEM 控制位(仅 UART0 支持), 没有连接实际输出引脚:</p> <p>1: 置高; 0: 置低。</p>	0
1	RB_MCR_RTS	RW	<p>RTS 信号输出电平控制(仅 UART0 支持):</p> <p>1: RTS 信号输出有效(低电平); 0: RTS 信号输出高电平(默认)。</p>	0
0	RB_MCR_DTR	RW	<p>DTR 信号输出电平控制(仅 UART0 支持):</p> <p>1: DTR 信号输出有效(低电平); 0: DTR 信号输出高电平(默认)。</p>	0

#### 中断使能寄存器(R8\_UARTx\_IER) (x=0/1/2/3)

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

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： 0：一个停止位； 1：两个停止位。	0



[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	R0	串口 FIFO 启用状态位： 11: FIFO 已启用； 00: FIFO 未启用。	00b
[5:4]	Reserved	R0	保留。	00b
[3:0]	RB_IIR_INT_MASK	R0	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	R0	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	R0	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	R0	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	R0	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：	0

			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。	
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 引脚有效 (低电平); 0: DCD 引脚无效 (高电平)。	X
6	RB_MSR_RI	RO	RI 引脚状态位: 1: RI 引脚有效 (低电平); 0: RI 引脚无效 (高电平)。	X
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, 然后通过发送移位寄存器 TSR 逐个输出。	XXh

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	R0	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	R0	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 Fuart，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\_RECV\_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\_RECV\_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

芯片提供 2 个 SPI 接口，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<sub>sys</sub> 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	RO	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 时自动清除标志位 RB_SPI_IF_BYTE_END 的功能：	0

			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 <sub>sck</sub> = 系统主频 F <sub>sys</sub> / 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	R0	从机模式被片选状态位 (仅 SPI0 支持):	0



			1: 正被选中; 0: 没有被片选。	
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。 同 R8_SPIx_FIFO_COUNT。	00h

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

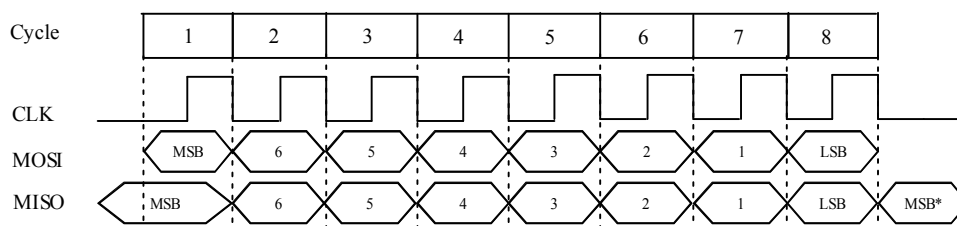


图 10-1 SPI 模式 0 传输格式

模式 3: RB\_SPI\_MST\_SCK\_MOD = 1

模式3时序图

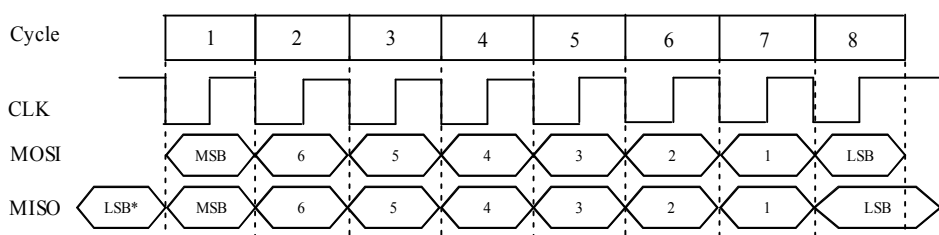


图 10-2 SPI 模式 3 传输格式

## 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\_SPIx\_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。

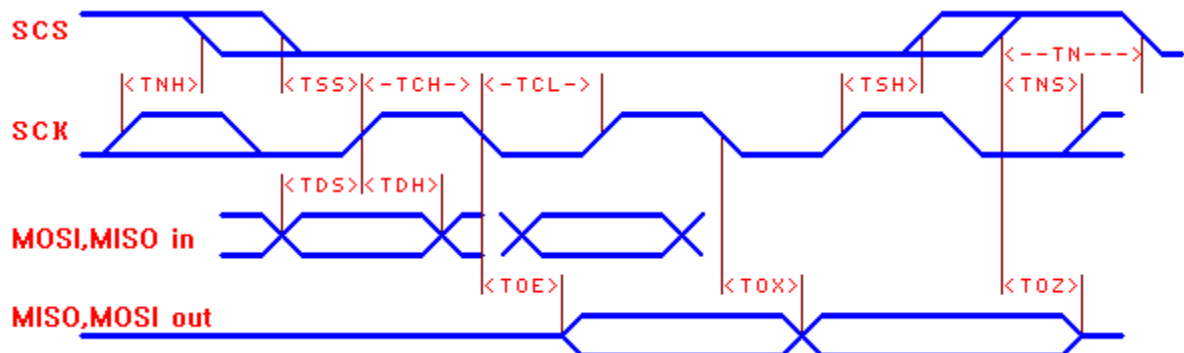
数据发送过程：

- (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，如果未满载则继续向 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\_SPIx\_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

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。	00h

			$F_{pwm} = F_{sys} / R8\_PWM\_CLOCK\_DIV。$	
--	--	--	--	--

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

### 11.3 PWM 配置

- (1)、设置寄存器 R8\_PWM\_CLOCK\_DIV，配置 PWM 的基准时钟频率；
- (2)、设置 PWM 输出极性配置寄存器 R8\_PWM\_POLAR，配置对应 PWMx 的输出极性；
- (3)、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；
- (4)、设置 PWM 输出使能寄存器 R8\_PWM\_OUT\_EN，开启对应的 PWMx 输出使能；
- (5)、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\_PWMx\_DATA；
- (6)、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；
- (7)、根据需要更新 R8\_PWMx\_DATA 中的数据，更新输出占空比。

调整 R8\_PWMx\_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)，Ncyc 结果在 63~256 之间。

$$PWMx \text{ 输出有效电平的占空比} = R8\_PWMx\_DATA / N_{cyc} * 100\%$$

$$PWMx \text{ 输出频率 } F_{pwmout} = F_{pwm} / N_{cyc} = F_{sys} / R8\_PWM\_CLOCK\_DIV / N_{cyc}$$

如果需要通过 PWM 产生直流信号，那么可以 PWMx 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 LED 屏控制器

### 12.1 LED 控制器简介

芯片提供了 LED 屏控制卡接口，内置 2 字节 FIFO，支持 DMA 和中断，节约 CPU 处理时间，支持 1/2/4 路数据线接口。

### 12.2 寄存器描述

表 12-1 LED 相关寄存器列表

名称	访问地址	描述	复位值
R8_LED_CTRL_MOD	0x40006400	LED 模式配置寄存器	0x02
R8_LED_CLOCK_DIV	0x40006401	LED 串行时钟分频寄存器	0x10
R8_LED_STATUS	0x40006404	LED 状态寄存器	0xX0
R16_LED_FIFO	0x40006408	LED 数据 FIFO 寄存器	0xXXXX
R16_LED_DMA_CNT	0x40006410	LED DMA 剩余计数寄存器	0x0000
R16_LED_DMA_MAIN	0x40006414	LED 主缓冲区 DMA 地址	0xXXXX
R16_LED_DMA_AUX	0x40006418	LED 辅助缓冲区 DMA 地址	0xXXXX

LED 模式配置寄存器 (R8\_LED\_CTRL\_MOD)

位	名称	访问	描述	复位值
[7:6]	RB_LED_CHAN_MOD	RW	LED 通道模式选择： 00: LED0，单通道输出； 01: LED0/1，双通道输出； 10: LED0~3，4 通道输出； 11: LED0~3，4 通道输出，其中 LED2/3 通道数据来源辅助缓冲区。	00b
5	RB_LED_IE_FIFO	RW	FIFO 计数过半中断使能： 1: FIFO 计数 $\leq$ 2 中断触发； 0: 禁止中断。	0
4	RB_LED_DMA_EN	RW	LED DMA 功能和 DMA 中断使能： 1: 使能；                  0: 禁止。	0
3	RB_LED_OUT_EN	RW	LED 信号输出使能： 1: 使能；                  0: 禁止。	0
2	RB_LED_OUT_POLAR	RW	LED 数据输出极性控制位： 1: 反相，数据 0 输出 1，数据 1 输出 0； 0: 直通，数据 0 输出 0，数据 1 输出 1。	0
1	RB_LED_ALL_CLEAR	RW	LED 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_LED_BIT_ORDER	RW	LED 串行数据位序选择： 1: 低位在前；          0: 高位在前。	0

LED 串行时钟分频寄存器 (R8\_LED\_CLOCK\_DIV)

位	名称	访问	描述	复位值
[7:0]	R8_LED_CLOCK_DIV	RW	LED 输出时钟分频系数。 LEDC 频率 = $F_{sys} / R8\_LED\_CLOCK\_DIV$ 。	10h

## LED 状态寄存器 (R8\_LED\_STATUS)

位	名称	访问	描述	复位值
7	RB_LED_IF_DMA_END	RW1	DMA 完成标志位，写 1 清零或者写 R16_LED_DMA_CNT 清零： 1: 已完成； 0: 未完成。	0
6	RB_LED_FIFO_EMPTY	RO	FIFO 空状态位： 1: FIFO 空； 0: FIFO 中有数据。	1
5	RB_LED_IF_FIFO	RW1	FIFO 计数过半中断标志位，写 1 清零或者写 R16_LED_FIFO 清零： 1: FIFO 计数 $\leq$ 2; 0: FIFO 计数 $>$ 2。	0
4	RB_LED_CLOCK	0	当前 LED 时钟信号电平状态： 1: 高电平； 0: 低电平。	X
3	Reserved	RO	保留。	0
[2:0]	RB_LED_FIFO_COUNT	RO	当前 FIFO 中字节计数值，一定偶数。	000b

## LED 数据 FIFO 寄存器 (R16\_LED\_FIFO)

位	名称	访问	描述	复位值
[15:0]	R16_LED_FIFO	WO	LED 数据 FIFO 入口，16 位写入。	XXXXb

## LED DMA 剩余计数寄存器 (R16\_LED\_DMA\_CNT)

位	名称	访问	描述	复位值
[15:0]	R16_LED_DMA_CNT	RW	LED_DMA_MAIN 主缓冲区当前 DMA 剩余字（16 位）计数，DMA 启动后自动递减，仅低 12 位有效。不包括辅助缓冲区。	0000b

## LED 主缓冲区 DMA 地址 (R16\_LED\_DMA\_MAIN)

位	名称	访问	描述	复位值
[15:0]	R16_LED_DMA_MAIN	RW	主缓冲区 DMA 起始地址/当前地址，预设初值后，自动递增，必须 2 字节对齐。	XXXXb

## LED 辅助缓冲区 DMA 地址 (R16\_LED\_DMA\_AUX)

位	名称	访问	描述	复位值
[15:0]	R16_LED_DMA_AUX	RW	辅助缓冲区 DMA 起始地址/当前地址，预设初值后，自动递增，必须 2 字节对齐。	XXXXb

## 12.3 LED 配置

- (1)、设置 R8\_LED\_CLOCK\_DIV 选择 LED 输出时钟频率；
- (2)、设置 R16\_LED\_DMA\_MAIN 指向准备输出数据的缓冲区，即主缓冲区；
- (3)、如果选择 LED 通道模式 3，那么还要设置 R16\_LED\_DMA\_AUX 指向辅助缓冲区；
- (4)、设置 R8\_LED\_CTRL\_MOD，选择通道模式、输出极性、位顺序、启用中断和 DMA 功能等；
- (5)、设置 LEDC 和必要的 LED0~LED3 引脚方向为输出，可选地，设置相应 I/O 的驱动能力；
- (6)、设置 DMA 计数寄存器 R16\_LED\_DMA\_CNT，启动 DMA 发送，或者用写 FIFO 的方式发送数据。

## 第 13 章 段式 LCD

### 13.1 段式 LCD 控制简介

芯片提供了段式 LCD 控制器接口，可支持 96 个点（24×4）LCD 面板。支持驱动电压可调，扫描频率可调，支持 1/2 duty、1/3 duty、1/4 duty 和 1/3 bias、1/2 bias 规格 LCD。

### 13.2 寄存器描述

表 13-1 LCD 相关寄存器列表

名称	访问地址	描述	复位值
R8_LCD_CTRL_MOD	0x40006000	LCD 模式配置寄存器	0x00
R32_LCD_RAM0	0x40006004	LCD 数据寄存器组 0	0x00000000
R32_LCD_RAM1	0x40006008	LCD 数据寄存器组 1	0x00000000
R32_LCD_RAM2	0x4000600C	LCD 数据寄存器组 2	0x00000000

LCD 模式配置寄存器 (R8\_LCD\_CTRL\_MOD)

位	名称	访问	描述	复位值
7	RB_LCD_V_SEL	WO	LCD 驱动电压选择： 1: V1033 * 76% (2.5V)； 0: V1033 * 100% (3.3V)。	0
[6:5]	RB_LCD_SCAN_CLK	RW	LCD 扫描时钟频率选择： 00: 256Hz； 01: 512Hz； 10: 1KHz； 11: 128Hz。	00b
[4:3]	RB_LCD_DUTY	RW	LCD 驱动 duty 选择： 00: 1/2 duty (COM0-COM1)； 01: 1/3 duty (COM0-COM2)； 10: 1/4 duty (COM0-COM3)。	00b
2	RB_LCD_BIAS	RW	LCD 驱动 bias 选择位： 1: 1/3 bias； 0: 1/2 bias。	0
1	RB_LCD_POWER_ON	RW	LCD 模拟电路使能位： 1: 使能； 0: 关闭。	0
0	RB_SYS_POWER_ON	RW	LCD 逻辑电路使能位： 1: 使能； 0: 关闭。	0

LCD 数据寄存器组 0 (R32\_LCD\_RAM0)

位	名称	访问	描述	复位值
[31:0]	R32_LCD_RAM0	RW	SEG0-SEG7 段的数据，每段 4 位。	00000000h

LCD 数据寄存器组 1 (R32\_LCD\_RAM1)

位	名称	访问	描述	复位值
[31:0]	R32_LCD_RAM1	RW	SEG8-SEG15 段的数据，每段 4 位。	00000000h

LCD 数据寄存器组 2 (R32\_LCD\_RAM2)

位	名称	访问	描述	复位值
[31:0]	R32_LCD_RAM2	RW	SEG16-SEG23 段的数据，每段 4 位。	00000000h

---

### 13.3 段式 LCD 配置

- (1)、选择并打开 32KHz 时钟源；
- (2)、设置将要用到的 LCD 驱动管脚为浮空输入，可选，设置 R16\_PIN\_ANALOG\_IE 以节约系统功耗，注意，外部复位 RST#复用了 LCD 驱动管脚 SEG23，如果要使用 SEG23，则要取消外部复位功能；
- (3)、向 LCD 数据寄存器 R32\_LCD\_RAM0/1/2 加载要显示的段码数据；
- (4)、配置 R8\_LCD\_CTRL\_MOD，设置驱动电压、扫描频率、bias、duty 等参数，并将 RB\_LCD\_POWER\_ON 和 RB\_SYS\_POWER\_ON 置位，开启段式 LCD 驱动；
- (5)、可以随时更新 LCD 数据寄存器中的数据，改变显示内容。

## 第 14 章 被动并口

### 14.1 被动并口简介

芯片提供了被动并口，包含 8 根双向数据线 PB7~PB0、4 根输入控制线 SLVS/SLVW/SLVR/SLVA 及 1 根输出中断请求线 SLVI。其中，SLVS 为片选，低电平有效，SLVW/SLVR 分别控制写和读，SLVA=0 表示操作的是数据；SLVA=1 表示操作的是命令或者状态。

表 14-1 被动并口 I/O 操作表

SLVS	SLVW	SLVR	SLVA	PB7~PB0	操作
1	X	X	X	三态禁止	未选中，无操作
0	1	1	X	三态禁止	虽然选中，但无操作
0	0	1/X	1	输入态	向 CH579 写入命令，命令存在 R8_SLV_WR_DATA
0	0	1/X	0	输入态	向 CH579 写入数据，数据存在 R8_SLV_WR_DATA
0	1	0	1	输出态	从 CH579 读出状态，状态来自 R8_SLV_RD_STAT
0	1	0	0	输出态	从 CH579 读出数据，数据来自 R8_SLV_RD_DATA

寄存器 R8\_SLV\_WR\_DATA 是对 R16\_PB\_INT\_MODE 高字节的复用，寄存器 R8\_SLV\_RD\_STAT 和 R8\_SLV\_RD\_DATA 是对 R16\_PB\_INT\_MODE 低字节和 R8\_PB\_OUT\_0 的复用，所以启用被动并口后，寄存器被复用了的模块的功能将不能使用。

中断请求输出 SLVI 是 R8\_SLV\_RD\_STAT[7]和 R8\_PA\_OUT\_0[3]的“逻辑或”，两者任意之一为 1 则输出默认的高电平，两者都为 0 才能输出低电平有效的中断请求。

### 14.2 寄存器描述

表 14-2 被动并口相关寄存器列表

名称	访问地址	描述	复位值
R8_SLV_CONFIG	0x4000101C	被动并口配置寄存器	0x00
R8_SLV_RD_DATA	0x400010C8	被动并口读出数据寄存器	0x00
R8_SLV_RD_STAT	0x40001096	被动并口读出状态寄存器	0x00
R8_SLV_WR_DATA	0x40001097	被动并口写入数据/命令寄存器	0x00

被动并口配置寄存器 (R8\_SLV\_CONFIG)

位	名称	访问	描述	复位值
7	RB_IF_SLV_RD	RW1	数据被读取中断标志位，写 1 清 0： 1：检测到被读事件；0：无事件。	0
6	RB_IF_SLV_WR	RW1	数据被写入中断标志位，写 1 清 0： 1：检测到被写事件；0：无事件。	0
5	RB_IF_SLV_CMD	RO	命令被写入标志； 1：当前被写入的是命令； 0：当前无写入或被写入的是数据。	0
4	Reserved	RO	保留。	0
3	RB_SLV_IE_RD	RW	数据读取中断使能位： 1：使能；0：关闭。	0
2	RB_SLV_IE_WR	RW	数据写入中断使能位： 1：使能；0：关闭。	0
1	RB_SLV_IE_CMD	RW	命令写入中断使能位： 1：使能；0：关闭。	0
0	RB_SLV_ENABLE	RW	被动并口使能位：	0

			1: 使能;	0: 关闭。	
--	--	--	--------	--------	--

### 14.3 被动并口配置

- (1)、可选的，如果要启用 SLVI 中断输出，那么首先需在 R8\_SLV\_RD\_STAT 中预置状态值，尤其是与 SLVI 中断引脚相关的 R8\_SLV\_RD\_STAT[7]，应该设置为 1，以及清零 R8\_PA\_OUT\_0[3]；
- (2)、设置 SLVA(PA0)、SLVS(PA1)、SLVR(PB8)、SLVW(PB9)对应的 GPIO 引脚为上拉输入（建议）或浮空输入模式，设置 PB7~PB0 引脚为上拉输入（建议）或浮空输入模式；
- (3)、设置 R8\_SLV\_CONFIG 的 RB\_SLV\_ENABLE 使能被动并口；
- (4)、可选的，如果要启用 SLVI 中断输出，那么要设置 SLVA(PA3)对应的 GPIO 引脚为输出；
- (5)、可选的，根据需要开启被动并口中断响应，组合 RB\_SLV\_IE\_RD/RB\_SLV\_IE\_WR/RB\_SLV\_IE\_CMD，并配置内核 NVIC 相应通道使能，建议只启用 RB\_SLV\_IE\_CMD；
- (6)、可选的，在 R8\_SLV\_RD\_DATA 中预置待发送数据。
- (7)、可选的，如果已经启用 SLVI 中断输出，并且需要通过 SLVI 输出低电平的中断请求，那么可以更新 R8\_SLV\_RD\_STAT 中的预置状态值，并使 R8\_SLV\_RD\_STAT[7]为 0，如果需要撤消中断请求恢复 SLVI 为默认的高电平，那么可以设置 R8\_SLV\_RD\_STAT[7]为 1，或设置 R8\_PA\_OUT\_0[3]为 1；
- (8)、被写入命令中断或查询到 RB\_IF\_SLV\_WR 有效后，当 RB\_IF\_SLV\_CMD 为 1 时，从 R8\_SLV\_WR\_DATA 得到写入的命令码，当 RB\_IF\_SLV\_CMD 为 0 时，从 R8\_SLV\_WR\_DATA 得到写入的数据；
- (9)、查询到 RB\_IF\_SLV\_RD 有效后，可以预置下一个数据或者状态。

## 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，支持 14 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。

#### 15.1.2 功能描述

下图为一个 ADC 模块的框图。

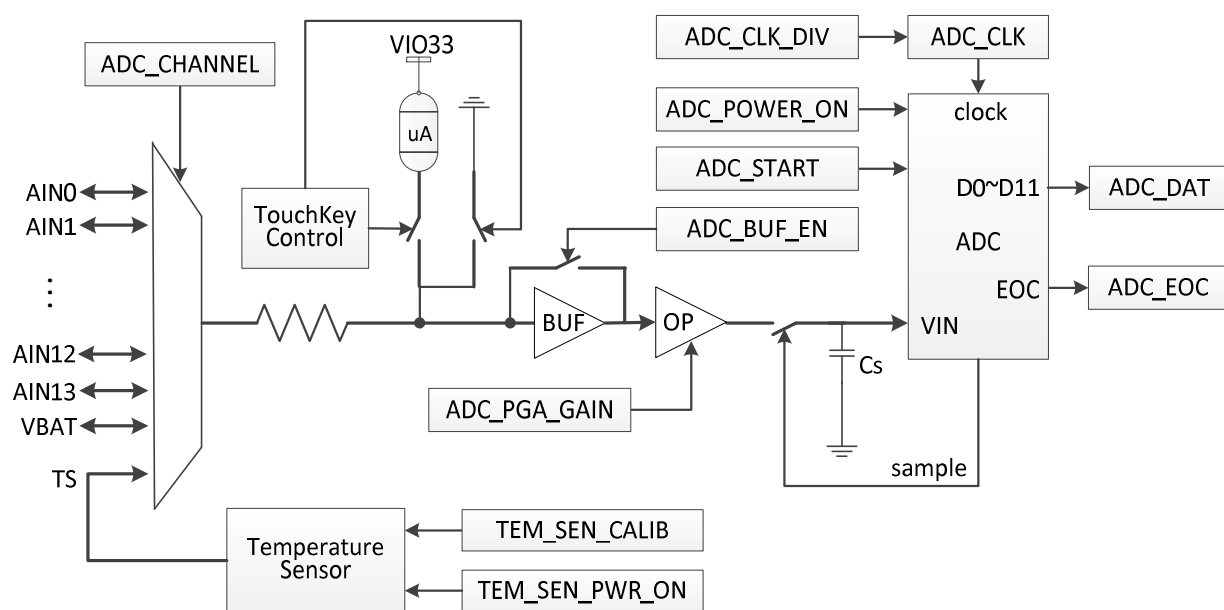


图 15-1 ADC 结构图

### 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	0x04
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

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 AINO~AIN13； 0Eh：电池电压 VBAT； 0Fh：内置温度传感器 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 12 分频，2.67MHz； 10：基于 CK32M 的 6 分频，5.33MHz； 11：基于 CK32M 的 8 分频，4MHz。 实际采样速率约为时钟频率的 1/16。	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：测试模式（仅通道 1 支持），测试模式下数据寄存器 R16_ADC_DATA 低 12 位数据将按位取反（0x0579 反为 0x0A86）； 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)	$(ADC/512-3)*V_{ref}$	$5*V_{ref}$	$-0.2V \sim V_{I033}+0.2V$	$2.9V \sim V_{I033}$
-6dB (1/2 倍)	$(ADC/1024-1)*V_{ref}$	$3*V_{ref}$	$-0.2V \sim 3.15V$	$1.9V \sim 3V$
0dB (1 倍)	$(ADC/2048)*V_{ref}$	$2*V_{ref}$	$0V \sim 2.1V$	$0V \sim 2V$
6dB (2 倍)	$(ADC/4096+0.5)*V_{ref}$	$1.5*V_{ref}$	$0.525V \sim 1.575V$	$0.6V \sim 1.5V$

ADC：ADC 转换后的数字量，即 R16\_ADC\_DATA。

$V_{ref}$ ：内部模拟电路的电源节点 VINTA 的实际电压值，通常为  $1.05V \pm 0.015V$ 。

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  $1K\Omega$  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	Reserved	RO	保留。	0
5	RB_TKEY_CHG_ACT	RO	TouchKey 电容充电状态： 1：电容充电中； 0：没有在充电。	0
4	RB_TKEY_ACTION	RO	TouchKey 操作状态： 1：充电或放电或 ADC 转换中； 0：空闲状态。	0
3	RB_TKEY_PWR_ON	RW	TouchKey 模块电源使能控制： 1：使能； 0：关闭。	0
[2:1]	Reserved	RO	保留。	00b
0	RB_ADC_START	RW	ADC 转换启动控制及状态，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
[6:3]	Reserved	RW	保留。	0000b
[2:0]	RB_TEM_SEN_CALIB	RW	TS 温度传感器校准值，需保持不变。	100b

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 转换完成中断标志。 对寄存器 R8_ADC_CONVERT 进行写操作 可清零此标志。	0
[6:0]	Reserved	RW	保留。	0000000b

## 15.3 ADC 配置

### 15.3.1 外部通道采样

- 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；
- 设置 R8\_ADC\_CHANNEL 寄存器，选择外部或内部信号通道；
- 设置 R8\_ADC\_CONVERT 寄存器，置位 RB\_ADC\_START，启动 ADC 转换；
- 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；
- 重复 2、3、4 步骤，可以继续采样另一通道或者下一组数据。
- 单次 ADC 转换周期：ADC 采样（4 个时钟）+ 转换时间（12 个时钟） $\approx$  16 个 T<sub>adc</sub>，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  $V_{ref}$ 。

### 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，清零 RB\_ADC\_DIFF\_EN，设置 RB\_ADC\_CLK\_DIV，设置 RB\_ADC\_BUF\_EN 为 1，设置 RB\_ADC\_PGA\_GAIN 为 0dB 增益；

(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。

## 第 16 章 触摸按键 Touch-Key

### 16.1 Touch-Key 简介

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。

### 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_CTRL	0x4000105A	TouchKey 控制寄存器	0x00
R8_TKEY_CNT	0x4000105F	TouchKey 充放电时间配置寄存器	0x00

寄存器 R8\_TKEY\_CTRL 是 R8\_ADC\_CONVERT 的别名。

TouchKey 控制寄存器 (R8\_TKEY\_CTRL)

位	名称	访问	描述	复位值
7	RB_ADC_EOC_X	RO	ADC 转换结束标志。 1: 完成; 0: 进行中。	0
6	Reserved	RO	保留。	0
5	RB_TKEY_CHG_ACT	RO	TouchKey 电容充电状态: 1: 电容充电中; 0: 没有在充电。	0
4	RB_TKEY_ACTION	RO	TouchKey 操作状态: 1: 充电或放电或 ADC 转换中; 0: 空闲状态。	0
3	RB_TKEY_PWR_ON	RW	TouchKey 模块电源使能控制: 1: 使能; 0: 关闭。	0
[2:1]	Reserved	RO	保留。	00b
0	RB_ADC_START	RW	ADC 转换启动控制及状态, 自动清零: 1: 开始转换/正在转换; 0: 停止转换。	0

TouchKey 充放电时间配置寄存器 (R8\_TKEY\_CNT)

位	名称	访问	描述	复位值
[7:0]	R8_TKEY_CNT	WO	Touch-Key 充放电周期数 (以 T <sub>sys</sub> 计)。 完整 8 位是充电周期数。 建议不小于 0x10。 放电周期数根据充电周期数自动产生。 当充电周期数小于 128 时, 放电周期数为 8; 当充电周期数大于或等于 128 时, 放电周期数为 14。	00h

### 16.3 Touch-Key 配置

- (1)、设置 RB\_ADC\_POWER\_ON 为 1 开启 ADC, 清零 RB\_ADC\_DIFF\_EN, 设置 RB\_ADC\_CLK\_DIV, 设置 RB\_ADC\_BUF\_EN 为 1, 设置 RB\_ADC\_PGA\_GAIN 为 0dB 或 -6dB 增益, 后者用于更小电容;
- (2)、设置 R8\_TKEY\_CTRL 的 RB\_TKEY\_PWR\_ON 为 1, 启用 TouchKey 模块;
- (3)、设置 R8\_ADC\_CHANNEL 选择目标按键所在通道;

- 
- (4)、可选的，如果使用中断方式，那么必须写 R8\_TKEY\_CTRL 以清零 RB\_ADC\_IF\_EOC，同步骤 2；
- (5)、估算目标按键电容，计算其充电时间，将充电周期数写入 R8\_TKEY\_CNT 寄存器，计算：  
充电周期数  $R8\_TKEY\_CNT = (C_{key} + 5pF \text{ 寄生}) * V_{key} / I_{key} / T_{sys}$ ，  
(其中，0dB 增益时  $V_{key}$  建议 1.6V，-6dB 增益时  $V_{key}$  建议 2.4V， $I_{key}$  约为 35uA)  
假定触摸按键外部电容  $C_{key}=35pF$ ， $F_{sys}=16MHz$ ，那么  $R8\_TKEY\_CNT=29$ ，  
当计算结果小于 16 时用 16，在 110 与 128 之间时用 128；
- (6)、查询等待 RB\_TKEY\_ACTION 为 0（或者，如果之前清零过则可等待 RB\_ADC\_IF\_EOC 置 1），读取 R16\_ADC\_DATA 获取 12 位 ADC 转换数据，与同一通道的历次数值比较，分析判断出按键是否按下，建议用软件分析去除干扰；
- (7)、重复 3、4、5、6 步骤，可以继续检测下一通道的按键。
- (8)、如果使用中断方式，那么将在中断服务程序中重复以下 4 个步骤：6 读取数据和判断、4 或 2 清中断标志、3 选择通道、5 启动检测。

## 第 17 章 USB 控制器

### 17.1 USB 控制器简介

芯片内嵌 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

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_TYPE_C_CTRL	0x40008038	USB type-C 配置寄存器	0x00
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 type-C 配置寄存器 (R8\_USB\_TYPE\_C\_CTRL)

位	名称	访问	描述	复位值
7	RB_UTCC_GP_BIT	RW	USB 通用标志位，用户自定义。	0
6	RB_UCC2_PD_EN	RW	UCC2 引脚的内部 5.1K 下拉电阻： 1：使能； 0：禁止。	0
[5:4]	RB_UCC2_PU_EN	RW	UCC2 引脚内部上拉电阻控制位： 00：禁止内部上拉电阻； 01：开启内部 36K 上拉，提供默认 USB 电流； 10：开启内部 12K 上拉，可提供 1.5A 电流； 11：开启内部 4.7K 上拉，可提供 3A 电流。	00b
3	RB_VBUS_PD_EN	RW	VBUS 引脚内部 10K 下拉电阻： 1：使能； 0：禁止。	0
2	RB_UCC1_PD_EN	RW	UCC1 引脚的内部 5.1K 下拉电阻： 1：使能； 0：禁止。	0
[1:0]	RB_UCC1_PU_EN	RW	UCC1 引脚内部上拉电阻控制位： 00：禁止内部上拉电阻；	00b

			01: 开启内部 36K 上拉, 提供默认 USB 电流; 10: 开启内部 12K 上拉, 可提供 1.5A 电流; 11: 开启内部 4.7K 上拉, 可提供 3A 电流。	
--	--	--	--	--

上述 USB type-C 上拉电阻和下拉电阻独立于 GPIO 端口方向控制和上拉电阻使能寄存器所控制的端口上拉电阻, 当一引脚用于 USB type-C 时, 应该禁止该引脚对应的端口上拉电阻, 建议对该引脚启用高阻输入模式 (避免该引脚输出低电平或者高电平)。

#### 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。	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, 主机模式下自动暂停后续传输; 0: 不暂停。	0
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。
0	01	使能 USB 设备功能, 关闭内部上拉电阻, 需加外部上拉。
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_UDA_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	0

			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	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_IS_NAK	RO	USB 设备模式下，NAK 响应状态位，同 RB_U_IS_NAK： 1: 当前 USB 传输过程中回应 NAK； 0: 无 NAK 响应。	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	XXb



			PID 标识。	
[3:0]	MASK_UIS_ENDP	R0	设备模式下, 当前 USB 传输事务的端点号。	XXXXb
	MASK_UIS_H_RES	R0	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 SETUP 包。

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 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	R0	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 共 5 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 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), 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。

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
R16_UEP0_DMA	0x40008010	端点 0 缓冲区起始地址	0xFFFF
R16_UEP1_DMA	0x40008014	端点 1 缓冲区起始地址	0xFFFF
R16_UEP2_DMA	0x40008018	端点 2 缓冲区起始地址	0xFFFF
R16_UEP3_DMA	0x4000801c	端点 3 缓冲区起始地址	0xFFFF
R8_UEP0_T_LEN	0x40008020	端点 0 发送长度寄存器	0xFF
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	0xFF
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	0xFF
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	0xFF
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	0xFF
R8_UEP4_CTRL	0x40008032	端点 4 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。	0

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 UEPO_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： UEPO_DMA+0 地址：端点 0 收发共用缓冲区 64 字节起始地址； UEPO_DMA+64 地址：端点 4 接收缓冲区 64 字节起始地址； UEPO_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 (N)。
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 n 缓冲区起始地址 (R16\_UEPn\_DMA) (n=0/1/2/3)

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)

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)

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。	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； 10: 应答 NAK 或忙； 11: 应答 STALL 或错误。	00b
[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 17-5 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： 1：高电平； 0：低电平。	X
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

			0: 禁止端点发送。	
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 17-6 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 17-7 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	RO	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	RO	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	RO	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 以太网控制器 ETH

### 18.1 以太网控制器简介

芯片集成了以太网控制器 MAC 和 PHY 以及 DMA，兼容 IEEE802.3 协议。内部 DMA 传输数据和接收数据到系统 RAM 中。PHY 物理层是一个 10Mbit/S 的以太网络收发器，提供部分网络 PHY 寄存器，可以设置发送和接收性能。

网络底层操作主要以子程序库提供应用支持，对寄存器不作详细介绍。主要特性：

- 支持全双工和半双工。
- 支持短包填充设置。
- 支持 CRC 设置和填充。
- 支持巨型帧接收。
- 支持不同的过滤模式组合。
- 支持 pause 帧发送和设置。
- 支持自动协商机制。
- 支持 DMA，用于发送和接收数据。
- PHY 收发器兼容 10BASE-T，发送模块支持节能模式。
- 内置 50 欧姆传输阻抗匹配电阻，也可选择外接。
- 提供由 IEEE 分配的全球唯一 MAC 地址。

### 18.2 寄存器描述

表 18-1 以太网控制器相关寄存器列表

名称	偏移地址	描述	复位值
R8_ETH_EIE	0x40009003	中断使能寄存器	0x00
R8_ETH_EIR	0x40009004	中断标志寄存器	0x00
R8_ETH_ESTAT	0x40009005	状态寄存器	0x00
R8_ETH_ECON2	0x40009006	PHY 模拟参数设置寄存器	0x06
R8_ETH_ECON1	0x40009007	收发控制寄存器	0x00
R32_ETH_TX	0x40009008	发送 DMA 控制寄存器	0xFFFFFFFF
R16_ETH_ETXST	0x40009008	发送 DMA 缓存起始地址寄存器	0xFFFF
R16_ETH_ETXLN	0x4000900A	发送长度寄存器	0xFFFF
R32_ETH_RX	0x4000900C	接收 DMA 控制寄存器	0x00000000
R16_ETH_ERXST	0x4000900C	接收 DMA 缓冲起始地址寄存器	0x0000
R16_ETH_ERXLN	0x4000900E	接收长度寄存器	0x0000
R32_ETH_HTL	0x40009010	Hash 表低字节寄存器	0x484EA033
R32_ETH_HTH	0x40009014	Hash 表高字节寄存器	0x5000EF97
R32_ETH_MACON	0x40009018	接收过滤设置寄存器	0x10000000
R8_ETH_ERXFCN	0x40009018	接收包过滤控制寄存器	0x00
R8_ETH_MACON1	0x40009019	Mac 层流控制寄存器	0x00
R8_ETH_MACON2	0x4000901A	Mac 层封包控制寄存器	0x00
R8_ETH_MABBIPG	0x4000901B	最小包间间隔寄存器	0x10
R32_ETH_TIM	0x4000901C	流控制暂停帧时间寄存器	0xFFFFFFFF
R16_ETH_EPAUS	0x4000901C	流控制暂停帧时间寄存器	0xFFFF
R16_ETH_MAMXFL	0x4000901E	最大接收包长度寄存器	0x0000
R16_ETH_MIRD	0x40009020	MII 读数据寄存器	0x1100
R32_ETH_MIWR	0x40009024	MII 写寄存器	0x00000000
R8_ETH_MIREGADR	0x40009024	MII 地址寄存器	0x00



R8_ETH_MISTAT	0x40009025	MII 状态寄存器	0x00
R16_ETH_MIWR	0x40009026	MII 写数据寄存器	0x0000
R32_ETH_MAADRL	0x40009028	MAC 地址低字节寄存器	0xFFFFFFFF
R16_ETH_MAADRH	0x4000902C	MAC 地址高字节寄存器	0XXXXX

## 中断使能寄存器 (R8\_ETH\_EIE)

位	名称	访问	描述	复位值
7	RB_ETH_EIE_INTIE	RW	以太网中断使能： 0: 关闭中断； 1: 开启中断。	0
6	RB_ETH_EIE_RXIE	RW	接收完成中断使能： 0: 关闭中断； 1: 开启中断。	0
5	Reserved	RO	保留。	0
4	RB_ETH_EIE_LINKIE	RW	Link 变化中断使能： 0: 关闭中断； 1: 开启中断。	0
3	RB_ETH_EIE_TXIE	RW	发送完成中断使能： 0: 关闭中断； 1: 开启中断。	0
2	RB_ETH_EIE_R_EN50	RW	内置的 50 欧姆阻抗匹配电阻使能： 0: 片内电阻断开； 1: 片内电阻连接。	0
1	RB_ETH_EIE_TXERIE	RW	发送错误中断使能： 0: 关闭中断； 1: 开启中断。	0
0	RB_ETH_EIE_RXERIE	RW	接收错误中断使能： 0: 关闭中断； 1: 开启中断。	0

## 中断标志寄存器 (R8\_ETH\_EIR)

位	名称	访问	描述	复位值
7	Reserved	RO	保留。	0
6	RB_ETH_EIR_RXIF	RW1	接收完成标志。	0
5	Reserved	RO	保留。	0
4	RB_ETH_EIR_LINKIF	RW1	Link 变化标志。	0
3	RB_ETH_EIR_TXIF	RW1	发送完成标志。	0
2	Reserved	RO	保留。	0
1	RB_ETH_EIR_TXERIF	RW1	发送错误标志。	0
0	RB_ETH_EIR_RXERIF	RW1	接收错误标志。	0

## 状态寄存器 (R8\_ETH\_ESTAT)

位	名称	访问	描述	复位值
7	RB_ETH_ESTAT_INT	RW1	中断。	0
6	RB_ETH_ESTAT_BUFER	RW1	Buffer 错误。	0
5	RB_ETH_ESTAT_RXCRCER	RO	接收 CRC 出错。	0
4	RB_ETH_ESTAT_RXNIBBLE	RO	接收 nibble 错误。	0
3	RB_ETH_ESTAT_RXMORE	RO	接收超过设定最大数据包。	0
2	RB_ETH_ESTAT_RXBUSY	RO	接收进行中。	0
1	RB_ETH_ESTAT_TXABRT	RO	发送被 MCU 打断。	0
0	Reserved	RO	保留。	0

## PHY 模拟参数设置寄存器 (R8\_ETH\_ECON2)

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

[7:4]	Reserved	RO	保留。	0000b
[3:1]	RB_ETH_ECON2_RX RB_ETH_ECON2_MUST	RW	保留，必须写入 011。	011b
0	RB_ETH_ECON2_TX	RW	发送端节能驱动控制： 0：额定驱动； 1：节能驱动。	0

## 收发控制寄存器 (R8\_ETH\_ECON1)

位	名称	访问	描述	复位值
7	RB_ETH_ECON1_TXRST	RW	发送模块复位： 0：不复位； 1：复位发送模块。	0
6	RB_ETH_ECON1_RXRST	RW	接收模块复位： 0：不复位； 1：复位接收模块。	0
[5:4]	Reserved	RO	保留。	00b
3	RB_ETH_ECON1_TXRTS	RW	发送开始，发送完成后自动清零： 1：启动发送； 0：无动作。	0
2	RB_ETH_ECON1_RXEN	RW	接收使能控制： 0：关闭接收； 1：开启接收。	0
[1:0]	Reserved	RO	保留。	00b

## 发送 DMA 缓存地址寄存器 (R16\_ETH\_ETXST)

位	名称	访问	描述	复位值
[15:0]	R16_ETH_ETXST	RW	发送 DMA 缓冲区起始地址。 低 15 位有效，地址必须 4 字节对齐。	XXXXh

## 发送长度 (R16\_ETH\_ETXLN)

位	名称	访问	描述	复位值
[15:0]	R16_ETH_ETXLN	RW	发送长度。	XXXXh

## 接收 DMA 缓冲地址寄存器 (R16\_ETH\_ERXST)

位	名称	访问	描述	复位值
[15:0]	R16_ETH_ERXST	RW	接收 DMA 缓冲区起始地址。 低 15 位有效，地址必须 4 字节对齐。	XXXXh

## 接收长度寄存器 (R16\_ETH\_ERXLN)

位	名称	访问	描述	复位值
[15:0]	R16_ETH_ERXLN	RO	接收长度。	0000h

## Hash 表低字节寄存器 (R32\_ETH\_HTL)

位	名称	访问	描述	复位值
[31:24]	R8_ETH_EHT3	RW	Hash Table 字节 3。	48h
[23:16]	R8_ETH_EHT2	RW	Hash Table 字节 2。	4Eh
[15:8]	R8_ETH_EHT1	RW	Hash Table 字节 1。	A0h
[7:0]	R8_ETH_EHT0	RW	Hash Table 字节 0。	33h

## Hash 表高字节寄存器 (R32\_ETH\_HTH)

位	名称	访问	描述	复位值
[31:24]	R8_ETH_EHT7	RW	Hash Table 字节 7。	50h

[23:16]	R8_ETH_EHT6	RW	Hash Table 字节 6。	00h
[15:8]	R8_ETH_EHT5	RW	Hash Table 字节 5。	EFh
[7:0]	R8_ETH_EHT4	RW	Hash Table 字节 4。	97h

## 接收过滤设置寄存器 (R8\_ETH\_ERXFCON)

位	名称	访问	描述	复位值
7	RB_ETH_ERXFCON_UCEN	RW	目标地址匹配过滤设置： 0: 不使用该过滤条件； 1: RB_ETH_ERXFCON_ANDOR=0 目标地址匹配将被接收，RB_ETH_ERXFCON_ANDOR=1 目标地址不匹配将被过滤。	0
6	RB_ETH_ERXFCON_ANDOR	RW	过滤条件 AND 和 OR 设置： 0: 任一过滤条件满足包均被接收； 1: 所有过滤条件都满足包才被接收。	0
5	RB_ETH_ERXFCON_CRCEN	RW	CRC 校验过滤设置： 0: 不使用该过滤条件； 1: RB_ETH_ERXFCON_ANDOR=0 CRC 校验正确将被接收，RB_ETH_ERXFCON_ANDOR=1 CRC 校验错将被过滤。	0
4	Reserved	RO	保留。	0
3	RB_ETH_ERXFCON_MPEN	RW	魔法包过滤设置： 0: 不使用该过滤条件； 1: RB_ETH_ERXFCON_ANDOR=0 魔法包将被接收，RB_ETH_ERXFCON_ANDOR=1 非魔法包将被过滤。	0
2	RB_ETH_ERXFCON_HTEN	RW	hash 表匹配过滤设置： 0: 不使用该过滤条件； 1: RB_ETH_ERXFCON_ANDOR=0 hash 表匹配将被接收，RB_ETH_ERXFCON_ANDOR=1 hash 表不匹配将被过滤。	0
1	RB_ETH_ERXFCON_MCEN	RW	组播包匹配过滤设置： 0: 不使用该过滤条件； 1: RB_ETH_ERXFCON_ANDOR=0 组播包匹配将被接收，RB_ETH_ERXFCON_ANDOR=1 组播包不匹配将被过滤。	0
0	RB_ETH_ERXFCON_BCEN	RW	广播包匹配过滤设置： 0: 不使用该过滤条件； 1: RB_ETH_ERXFCON_ANDOR=0 广播包将被接收，RB_ETH_ERXFCON_ANDOR=1 非广播包将被过滤。	0

## Mac 层流控制寄存器 (R8\_ETH\_MACON1)

位	名称	访问	描述	复位值
[7:6]	Reserved	RO	保留。	00b
[5:4]	RB_ETH_MACON1_FCEN	RW	pause 帧设置，全双工下有效： 00: 停止发送暂停帧； 01: 发送一次暂停帧，然后停止发送； 10: 周期性发送暂停帧； 11: 发送 0 timer 暂停帧，然后停止发	00b

			送。	
3	RB_ETH_MACON1_TXPAUS	RW	发送 pause 帧使能控制： 0: 不发送 pause 帧；1: 使能发送。	0
2	RB_ETH_MACON1_RXPAUS	RW	接收 pause 帧使能： 0: 不接收 pause 帧；1: 使能接收。	0
1	RB_ETH_MACON1_PASSALL	RW	控制帧设置： 0: 控制帧将被过滤； 1: 没被过滤的控制帧将写入缓存。	0
0	RB_ETH_MACON1_MARXEN	RW	MAC 层接收使能： 0: MAC 不接收数据；1: MAC 接收使能。	0

Mac 层封包控制寄存器 (R8\_ETH\_MACON2)

位	名称	访问	描述	复位值
[7:5]	RB_ETH_MACON2_PADCFG	RW	短包填充设置： 7: 所有短包填充 0 至 64 字节，再 4 字节 CRC； 6: 不填充短包； 5: 检测到字段为 8100h 的 VLAN 网络包自动填充 0 至 64 字节，否则短包填充 60 字节 0，填充后再 4 字节 CRC； 4: 不填充短包； 3: 所有短包填充 0 至 64 字节，再 4 字节 CRC； 2: 不填充短包； 1: 所有短包填充 0 至 60 字节，再 4 字节 CRC； 0: 不填充短包。	000b
4	RB_ETH_MACON2_TXCRGEN	RW	发送添加 CRC 控制： 0: 硬件不填充 CRC；1: 硬件填充 CRC。	0
3	RB_ETH_MACON2_PHDREN	RW	特殊 4 字节不参与 CRC 校验。	0
2	RB_ETH_MACON2_HFRMEN	RW	允许接收巨型帧： 0: 不允许接收巨型帧；1: 允许接收。	0
1	Reserved	RO	保留。	0
0	RB_ETH_MACON2_FULDPX	RW	以太网通讯模式： 0: 半双工； 1: 全双工。	0

最小包间间隔寄存器 (R8\_ETH\_MABBIPG)

位	名称	访问	描述	复位值
7	Reserved	RO	保留。	0
[6:0]	R8_ETH_MABBIPG	RW	最小包间间隔字节数。	0010000b

流控制暂停帧时间寄存器 (R16\_ETH\_EPAUS)

位	名称	访问	描述	复位值
[15:0]	R16_ETH_EPAUS	RW	流控制暂停帧时间。	XXXXh

最大接收包长度寄存器 (R16\_ETH\_MAMXFL)

位	名称	访问	描述	复位值
[15:0]	R16_ETH_MAMXFL	RW	最大接收包长度。	0000h

## MII 读数据寄存器 (R16\_ETH\_MIRD)

位	名称	访问	描述	复位值
[15:0]	R16_ETH_MIRD	RW	MII 读数据寄存器。	1100h

## MII 地址寄存器 (R8\_ETH\_MIREGADR)

位	名称	访问	描述	复位值
[7:5]	Reserved	RO	保留。	000b
[4:0]	RB_ETH_MIREGADR_MIRDL	RW	PHY 寄存器地址。	00000b

## MII 状态寄存器 (R8\_ETH\_MISTAT)

位	名称	访问	描述	复位值
[7:1]	Reserved	RO	保留。	0000000b
0	R8_ETH_MII_STA	RO	MII 寄存器操作状态： 1: 写 MII 寄存器；0: 读 MII 寄存器	0

## MII 写数据寄存器 (R16\_ETH\_MIWR)

位	名称	访问	描述	复位值
[15:0]	R16_ETH_MIWR	WO	MII 写数据寄存器。	0000h

## MAC 地址寄存器 (R32\_ETH\_MAADRL、R16\_ETH\_MAADRH)

位	名称	访问	描述	复位值
[31:0]	R32_ETH_MAADRL	RW	MAC Address 字节 1~4。	XXXXXXXXh
[15:0]	R16_ETH_MAADRH	RW	MAC Address 字节 5~6。	XXXXh

## 18.3 操作指南

### 1. 初始化

- (1)、配置安全寄存器进入安全模式，打开以太网络的时钟和电源；
- (2)、开启相应的中断，可选的，启用阻抗匹配电阻；
- (3)、配置接收过滤模式、CRC 功能、MAC 地址；
- (4)、设置缓存；
- (5)、启动接收，开启中断。

### 2. 发送数据

- (1)、写入 R16\_ETH\_ETXLN 数据长度；
- (2)、写入 R16\_ETH\_ETXST 数据地址；
- (3)、使能 RB\_ETH\_ECON1\_TXRTS 标志，启动发送。

### 3. 接收数据

- (1)、预先设置好接收地址，使能接收；
- (2)、使用中断或查询到接收完成状态；
- (3)、读取 R16\_ETH\_ERXLN 接收长度；
- (4)、更新 R16\_ETH\_ERXST 接收地址

具体的应用请基于以太网协议栈库使用，并参考提供的网络应用示例。

## 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。内部提供一百多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，只列出部分应用层常用的寄存器，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 规范。
- 单端 RF 接口，无需外部电感和电容匹配滤波网络，简化板级设计。
- 接收灵敏度-93dBm。
- 可编程-20dBm 到+3dBm 发送功率，支持动态调整。
- 在 0dBm 发送功率时使用 PCB 板载天线的无线通讯距离约 100 米。
- 使用内置 DC-DC 转换后，0dBm 发送功率时电流 6mA 多。
- 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。

#### 19.2.1 寄存器描述

表 19-1 LLE 模块相关寄存器列表

名称	偏移地址	描述	复位值
R32_LLE_CTRL_CMD	0x4000C200	LLE 命令	0x00000000
R32_LLE_CTRL_CFG	0x4000C204	LLE 配置	0x00000F01
R32_LLE_STATUS	0x4000C208	LLE 状态	0x00000000
R32_LLE_INT_EN	0x4000C20C	LLE 中断使能	0x00003F1F
R32_LLE_CTRL_MOD	0x4000C250	LLE 模式控制	0x00000011

#### 19.2.2 功能及配置

初始化流程：

- (1)、设置 R32\_LLE\_CTRL\_MOD 寄存器，写入 58H，配置为 LLE 模式，并且开启电源，使能 DMA 和 BB 模块功能。
- (2)、设置 R32\_LLE\_CTRL\_CFG 寄存器，配置 LLE 为自动收发数据模式，手动收发数据模式。
- (3)、设置 R32\_LLE\_CTRL\_CMD 寄存器，选择 4 种工作状态，发送、接收、停止、关闭。具体如下：
  - CMD\_RX 使用：设置当前模式为接收，手动模式下接收完成之后自动清 0，自动模式下会一直循环收发的流程，可通过 CMD\_STOP 或 CMD\_SHUT 停止。
  - CMD\_TX 使用：设置当前模式为发送，手动模式下发送完成之后自动清 0，自动模式下会一直循环发收的流程，可通过 CMD\_STOP 或 CMD\_SHUT 停止。
  - CMD\_STOP 使用：自动发送模式下，在帧间隔期间 CMD\_STOP 置 1，将会清零 CMD\_TX 状态跳转至 LLE\_IDLE，软件无法直接清零 CMD\_TX 状态。自动接收模式下，在帧间隔期间 CMD\_STOP 置 1，将会清零 CMD\_TX 状态跳转至 LLE\_IDLE，软件无法直接清零 CMD\_RX 状态。
  - CMD\_SHUT 使用：CMD\_SHUT 置位后，CMD\_TX/CMD\_RX/CMD\_STOP/CMD\_SHUT 将会全部清零，同时

LLE 的状态机复位。

## 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。

### 19.3.1 寄存器描述

表 19-2 DMA 模块相关寄存器列表

名称	偏移地址	描述	复位值
R32_DMA0_CTRL_CFG	0x4000C008	DMA0 配置寄存器	0x00002000
R32_DMA2_CTRL_CFG	0x4000C00C	DMA2 配置寄存器	0x00002000
R32_DMA0_TX_SRC	0x4000C010	DMA0 源地址， 用于配置发送数据起始地址	0x00000000
R32_DMA2_TX_SRC	0x4000C014	DMA2 源地址，自动接收模式， 用于配置发送数据起始地址	0x00000000
R32_DMA0_RX_DST	0x4000C018	DMA0 目的地址， 用于配置接收数据起始地址	0x00000000
R32_DMA2_RX_DST	0x4000C01C	DMA2 目的地址，自动发送模式， 用于配置接收数据起始地址	0x00000000

### 19.3.2 功能及配置

发送操作流程：

- (1)、设置 R32\_DMA0\_TX\_SRC 寄存器，写入发送数据地址。
- (2)、设置 R32\_DMA0\_CTRL\_CFG 寄存器，写入发送数据长度。
- (3)、设置 R32\_DMA0\_CTRL\_CFG 寄存器的 IRQMASK 为 1。
- (4)、自动模式下，设置 R32\_DMA2\_RX\_DST 寄存器，写入接收数据地址。
- (5)、自动模式下，设置 R32\_DMA2\_CTRL\_CFG 寄存器的 IRQMASK 为 1。

接收操作流程：

- (1)、设置 R32\_DMA0\_RX\_DST 寄存器，写入接收数据地址。
- (2)、设置 R32\_DMA0\_CTRL\_CFG 寄存器的 IRQMASK 为 1。
- (3)、自动模式下，设置 R32\_DMA2\_TX\_SRC，写入发送数据地址。
- (4)、自动模式下，设置 R32\_DMA2\_CTRL\_CFG 寄存器，写入发送数据长度。
- (5)、自动模式下，设置 R32\_DMA2\_CTRL\_CFG 寄存器的 IRQMASK 为 1。

## 19.4 BB 模块

### 19.4.1 寄存器描述

表 19-3 BB 模块相关寄存器列表

名称	偏移地址	描述	复位值
R32_BB_CTRL_CFG	0x4000C100	BB 配置寄存器	0x000822A7
R32_BB_TXCRC_INIT	0x4000C104	发送 CRC 初始值配置寄存器	0x0056BC9B
R32_BB_TXACCS_ADDR	0x4000C108	发送接入地址配置寄存器	0x8E89BED6
R32_BB_RXCRC_INIT	0x4000C10C	接收 CRC 初始值配置寄存器	0x0056BC9B
R32_BB_RXACCS_ADDR	0x4000C110	接收接入地址配置寄存器	0x8E89BED6

R32_BB_CTRL_TX	0x4000C12C	发送控制寄存器	0x00010008
R32_BB_RSSI_ST	0x4000C130	BB 接收信号质量状态	0x00000000
R32_BB_INT_EN	0x4000C134	BB 中断使能	0x00000003
R32_BB_INT_ST	0x4000C138	BB 中断状态	0x00000000

## 19.4.2 功能及配置

数据白化可配置，发送、接收 AA 可配置，发送、接收 CRC 初值可配置，可配置通信频道。

发送数据配置：

- (1)、配置发送接入地址。
- (2)、配置发送 CRC 初始值。
- (3)、配置发送数据的通信频道。

接收数据配置：

- (1)、配置接收接入地址。
- (2)、配置接收 CRC 初始值。
- (3)、配置接收数据的通信频道。

## 19.5 AES 模块

### 19.5.1 寄存器描述

表 19-4 AES 模块相关寄存器列表

名称	偏移地址	描述	复位值
R32_AES_CTRL_GCMOD	0x4000C300	AES 模式寄存器	0x00000060
R32_AES_CCMINT_EN	0x4000C304	AES 中断寄存器	0x00000000
R32_AES_CCMVT_INIT0	0x4000C308	AES CCM 模式初始矢量值	0x00000000
R32_AES_CCMVT_INIT1	0x4000C30C	AES CCM 模式初始矢量值	0x00000000
R32_AES_PKT_CNT0	0x4000C310	AES CCM 模式数据包计数值	0x00000000
R32_AES_PKT_CNT1	0x4000C314	AES CCM 模式数据包计数值	0x00000000
R32_AES_DATA0	0x4000C318	AES 数据寄存器	0x00000000
R32_AES_DATA1	0x4000C31C	AES 数据寄存器	0x00000000
R32_AES_DATA2	0x4000C320	AES 数据寄存器	0x00000000
R32_AES_DATA3	0x4000C324	AES 数据寄存器	0x00000000
R32_AES_KEY0	0x4000C328	AES 加密密钥寄存器 0	0x00000000
R32_AES_KEY1	0x4000C32C	AES 加密密钥寄存器 1	0x00000000
R32_AES_KEY2	0x4000C330	AES 加密密钥寄存器 2	0x00000000
R32_AES_KEY3	0x4000C334	AES 加密密钥寄存器 3	0x00000000
R32_AES_KEY4	0x4000C338	AES 加密密钥寄存器 4	0x00000000
R32_AES_KEY5	0x4000C33C	AES 加密密钥寄存器 5	0x00000000
R32_AES_KEY6	0x4000C340	AES 加密密钥寄存器 6	0x00000000
R32_AES_KEY7	0x4000C344	AES 加密密钥寄存器 7	0x00000000
R32_AES_RAND0	0x4000C348	AES 随机值	0x96906220
R32_AES_RAND1	0x4000C34c	AES 随机值	0x8A3BBF80
R32_AES_RAND2	0x4000C350	AES 随机值	0xB5625304
R32_AES_RAND3	0x4000C354	AES 随机值	0x2A43B7E8

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	105	°C
VI033	系统电源电压 (VI033 接电源, GND 接地)	-0.4	3.9	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	VDC1D/VDC1A 引脚上的电压 (如果用外置 DC-DC)	-0.4	VI033+0.4	V
VXCK	X32M1/X32M0/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, 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
VI033	系统电源电压	2.1	3.3	3.6	V
ICC <sub>8</sub>	直通方式 静态电源电流	Fsys=8M	1.3	1.5	mA
ICC <sub>16</sub>		Fsys=16M	1.6	1.9	mA
ICC <sub>32</sub>		Fsys=32M	2.3	2.8	mA
IDDC <sub>8</sub>	启用内置 DC-DC 后 静态电源电流	Fsys=8M	0.8	1.1	mA
IDDC <sub>16</sub>		Fsys=16M	1.0	1.3	mA
IDDC <sub>32</sub>		Fsys=32M	1.5	1.8	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	µA
IUP	GPIO 内置上拉电阻的输入端的输入电流	25	60	90	µA
IDN	GPIO 内置下拉电阻的输入端的输入电流	-90	-60	25	µA
Vref	VINTA 引脚的电压 (ADC 参考电压)	1.035	1.05	1.065	V
Vdci	VDC1D 引脚在启用 DC-DC 后的电压	1.33	1.37	1.43	V

### 20.3 低功耗模式功耗

测试条件: TA=25°C, 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	1.15	1.2	1.6	mA
空闲模式, 关闭所有睡眠时钟		1.15		mA

暂停模式, FlashROM 待机		470		uA
暂停模式, FlashROM 停用		420		uA
睡眠模式, 多种组合, 参考表 5-3	0.6		2.0	uA
睡眠模式, PMU+内核+RAM2K, GPIO 唤醒, 无 RTC		0.6		uA
下电模式, 多种组合, 参考表 5-3	0.2		1.3	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 <sub>DD</sub> (RAM2K)	RAM2K: 2KB SRAM			0.3		uA
I <sub>DD</sub> (RAM14K)	RAM14K: 14KB SRAM			0.8		uA
I <sub>DD</sub> (LSI)	内部 LSI 振荡器			0.3		uA
I <sub>DD</sub> (LSE)	外部 LSE 振荡器			0.4		uA
I <sub>DD</sub> (HSI)	内部 HSI 振荡器			160		uA
I <sub>DD</sub> (HSE)	外部 HSE 振荡器		100	200	300	uA
I <sub>DD</sub> (BM)	电池低压监控 BM 模块 (暂停模式下)			3		uA
I <sub>DD</sub> (PLL)	内部 PLL 振荡器			1.0		mA
I <sub>DD</sub> (ADC)	ADC 模块			0.27		mA
I <sub>DD</sub> (TKEY)	TouchKey 模块			0.08		mA
I <sub>DD</sub> (TS)	温度传感器 TS 模块			0.1		mA
I <sub>DD</sub> (USB)	USB 模块	非发送状态	1.2	1.6	2.0	mA
		发送状态		3		mA
I <sub>DD</sub> (ETH)	以太网 ETH 模块	非发送状态	2.2	2.6	3.0	mA
		连续发送: 额定驱动	75	100		mA
		连续发送: 节能驱动	56	75		mA
I <sub>DD</sub> (BLE)	BLE 蓝牙	接收	直通电源		11	mA
			启用 DC-DC		6	mA
		-20dBm 发送功率	直通电源		4	mA
			启用 DC-DC		2	mA
		0dBm 发送功率	直通电源		10	mA
			启用 DC-DC		5	mA
+3dBm 发送功率	直通电源		16	mA		
	启用 DC-DC		8	mA		

## 20.4 时钟源

表 20-5 高速振荡器 HSI 和 HSE

名称	参数说明		最小值	典型值	最大值	单位
F <sub>HSI</sub>	内部 HSI 振荡器频率			32		MHz
A <sub>HSI</sub>	HSI 振荡器精度	TA=-40°C~85°C		2.0	3.0	%
		TA=-5°C~70°C		0.7	1.0	%
		TA=10°C~45°C		0.3	0.5	%
T <sub>SUHSI</sub>	内部 HSI 振荡器启动到稳定时间			0.2	1	uS
F <sub>HSE</sub>	外部 HSE 振荡器频率 (关闭无线通讯时)		24	32	36	MHz
T <sub>SUHSE</sub>	外部 HSE 振荡器启动到稳定时间		200	500	1200	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 <sub>LSIR</sub>	内部 LSI 振荡器频率 (校准前)	20K	32K	48K	Hz
F <sub>LSI</sub>	内部 LSI 频率 (应用软件运行时校准后)	32700	32768	32836	Hz
A <sub>LSI</sub>	LSI 振荡器精度 (软件校准后)	TA=-40℃~85℃	0.2	0.9	%
		TA=0℃~60℃	0.1	0.3	%
T <sub>SULSI</sub>	内部 LSI 振荡器启动到稳定时间		40	100	μs
T <sub>SULSE</sub>	外部 LSE 振荡器启动到稳定时间	150	450	1000	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 <sub>PLL</sub>	PLL 倍频输出时钟 (CK32M * 15 倍)		480		MHz
T <sub>PLLLK</sub>	PLL 锁相时间		1	3	ms

## 20.5 时间参数

测试条件: TA=25℃, VI033=3.3V, F<sub>sys</sub>=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 <sub>VR</sub>	VI033 电压上升时间 (再慢可外加 CH410)	1		50000	μs	
T <sub>rpor</sub>	电源上电复位 RPOR 后的复位延时	11	15	20	ms	
T <sub>rst</sub>	RST#有效信号宽度		100		ns	
T <sub>mr</sub>	外部复位 MR 后的复位延时	2	8	18	μs	
T <sub>sr</sub>	软件复位 SR 后的复位延时	2	8	18	μs	
T <sub>wtr</sub>	看门狗复位 WTR 后的复位延时	10	12	18	μs	
T <sub>WAK</sub>	从低功耗状态 退出的唤醒时间	空闲模式	0.6	0.8	1.6	μs
		暂停模式, ROM 待机	0.7	1.0	2.0	μs
		暂停模式, ROM 停用	9	11	15	μs
		睡眠模式	300	330	400	μs
		下电模式	350	380	450	μs

注: 上表中延时参数均是基于 T<sub>sys</sub> 的倍数, 降低主频将增加延时。

上表中延时参数是基于使用内部 HSI 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<sub>WAK</sub> 均会额外再增加约 1.5ms (启动到稳定)。

## 20.6 其它参数

测试条件: TA=25℃, VI033=3.3V, F<sub>sys</sub>=16MHz。

表 20-9 其它参数

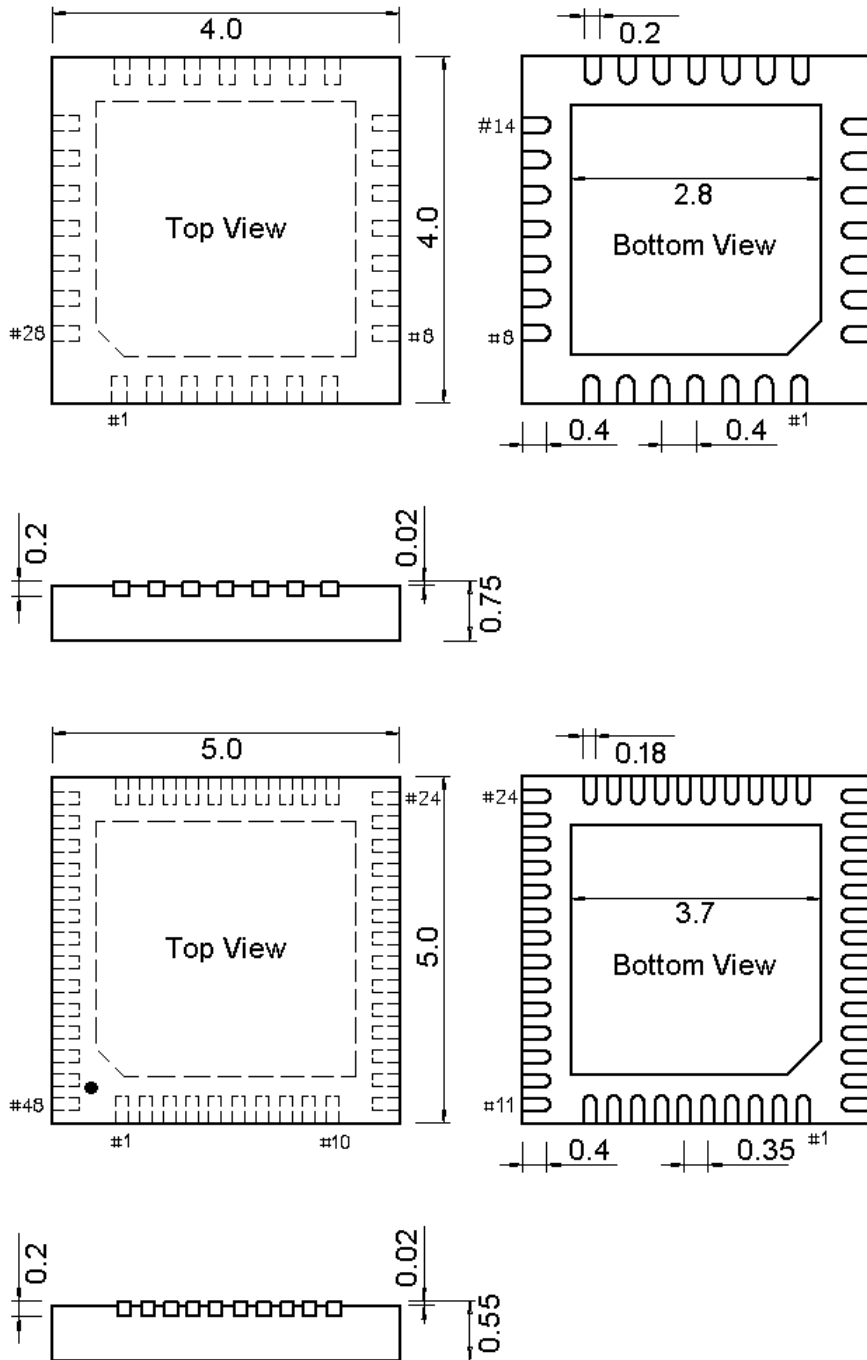
名称	参数说明	最小值	典型值	最大值	单位
R <sub>TS</sub>	TS 温度传感器的测量范围	-40		90	℃
A <sub>TSC</sub>	温度传感器经软件校准后的测量误差		±6		℃
T <sub>FRER</sub>	Flash-ROM 的单次扇区擦除操作时间	1.1	1.4	2.4	ms
T <sub>FRPG</sub>	Flash-ROM 的单次字编程操作时间	22	27	36	μs
N <sub>EPCE</sub>	Flash-ROM 的擦写次数 erase/program cycle endurance	5~45℃	100K	800K (非担保)	times
		-40~85℃	30K	100K (非担保)	
T <sub>DR</sub>	Flash-ROM 的数据保持能力	10			years
V <sub>ESD</sub>	I/O 输入或者输出 引脚上的 ESD 耐压	天线 ANT	2K	4K (非担保)	V
		I/O 引脚: PA 和 PB	4K	6K (非担保)	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
QFN28	4*4mm	0.4mm	15.7mil	方形无引线 28 脚	CH579F
QFN48	5*5mm	0.35mm	13.8mil	方形无引线 48 脚	CH579M

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  $\pm 0.2\text{mm}$ 。



## 第 22 章 修改记录

版本	日期	说明
V0.9	2018.05.09	初版发行
V1.0	2018.09.28	正式版发行
V1.1	2018.10.23	5.2 节加 RB_PWR_MUST_0010, 6.4.4 节改(1), 15.2 节改 R8_ADC_CFG, 20.6 节改 N <sub>EPCE</sub> , 6.3 节 R16_OSC_CAL_CNT 改笔误, 表 17-3 改笔误
V1.2	2019.03.04	1.2 节 VSW 改电感, 5.2 节改 R16_POWER_PLAN, 5.3 节睡眠模式删除低压检测, 6.3 节改 R16_CLK_SYS_CFG, 16.2 节和 16.3 节删除 RB_CHG_CNT
V1.3	2019.04.13	表 5-2 加 WFE, 表 20-9 调参数
V1.4	2019.07.26	7.4 节 R16_PIN_ANALOG_IE 部分控制位, 14.1 节 R8_SLV_RD_DATA 地址
V1.5	2020.04.15	RF 只保留 BLE, 6.4.4 节步骤调整, 17.2.2 节改低速笔误
V1.6	2020.12.02	改笔误, 6.3 节改 R16_CLK_SYS_CFG, 定时器增加计数模式说明, 表 2-3 改默认值, 表 20-8 加 T <sub>vr</sub> , 增加 10.5 节 SPI 时序

<http://wch.cn>