

用户手册

User Manual

GW3323

基于 **RISC-V** 内核的 **32** 位带蓝牙的微控制器

版本: **V1.1**

| | |
|----------------------------------|-----------|
| 1 系统框图 | 6 |
| 2 系统管理 | 7 |
| 2.1 系统时钟 | 7 |
| 2.2 时钟寄存器 | 8 |
| 2.3 低压断电寄存器 | 16 |
| 3 内存访问 | 18 |
| 4 低功耗模式 | 19 |
| 4.1 睡眠模式(SLEEP MODE,500UA) | 19 |
| 4.1.1 睡眠模式的配置 | 19 |
| 4.2 呼吸模式(POWER OFF,4UA) | 19 |
| 4.2.1 呼吸模式的配置 | 19 |
| 5 中断 | 21 |
| 5.1 特征 | 21 |
| 5.2 中断向量表 | 21 |
| 5.3 特殊中断寄存器 | 23 |
| 5.4 SDK 中的开关所有中断的使用和开发 | 25 |
| 6 WATCHDOG | 26 |
| 6.1 功能配置 | 26 |
| 6.2 WDT 特殊功能寄存器 | 27 |
| 6.3 SDK 中的看门狗的使用和开发 | 28 |
| 7 GPIO 管理 | 29 |
| 7.1 特征 | 29 |
| 7.2 GPIO 内部结构框图 | 29 |
| 7.3 GPIO 通用控制寄存器 | 29 |
| 7.4 GPIO 功能映射 | 31 |
| 7.5 外部端口中断唤醒 | 34 |
| 8 DMA | 37 |
| 8.1 特征 | 37 |
| 8.2 功能配置 | 37 |
| 9 TIMER | 38 |
| 9.1 特征 | 38 |
| 9.2 定时器时钟选择 | 38 |
| 9.3 TIMER0/1/2 特殊功能寄存器 | 39 |

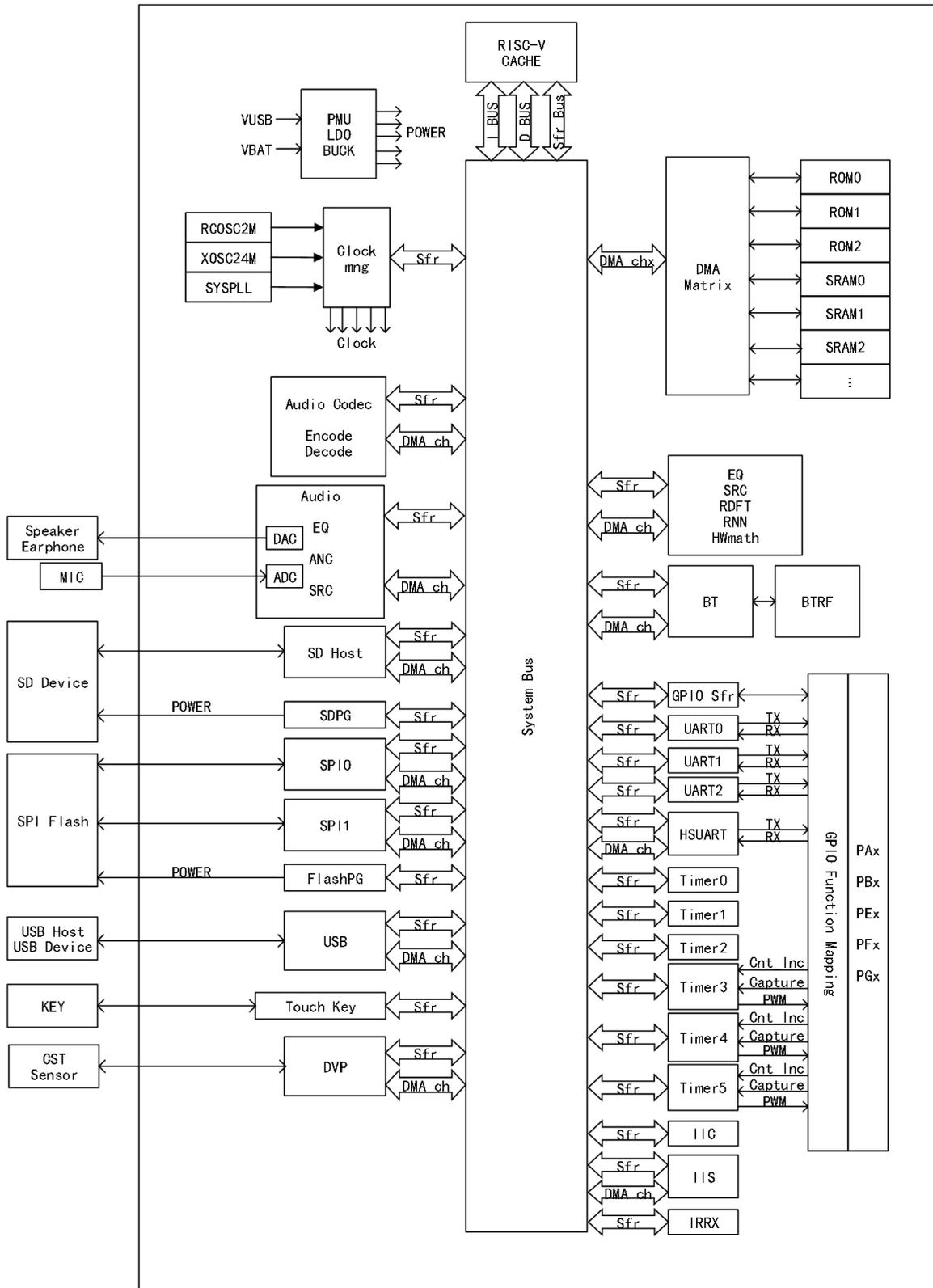
| | |
|--|-----------|
| 9.4 TIMER3/4/5 特殊功能寄存器 | 40 |
| 9.5 系统时钟函数 | 42 |
| 10 RTC | 43 |
| 10.1 特征 | 43 |
| 10.2 特殊功能寄存器 | 43 |
| 10.3 独立电源 RTC 寄存器 | 44 |
| 11 UART | 50 |
| 11.1 特征 | 50 |
| 11.2 UART 特殊功能寄存器 | 50 |
| 11.3 功能配置 | 52 |
| 11.4 SDK 中的串口使用和开发 | 52 |
| 12 HSUART | 53 |
| 12.1 特征 | 53 |
| 12.2 功能配置 | 53 |
| 12.2.1 SYNC 配置 | 53 |
| 12.2.2 TX 1 BYTE 带 BUFFER | 53 |
| 12.2.3 TX N BYTE 带 DMA | 54 |
| 12.2.4 RX 1 BYTE 带 BUFFER | 54 |
| 12.2.5 RX N BYTE 带有 DMA &禁用循环缓冲区 | 54 |
| 12.2.6 RX N BYTE 带有 DMA &循环缓冲器使能 | 55 |
| 12.2.7 DMA RX TIMER MODE: | 56 |
| 12.2.8 应用说明: | 56 |
| 12.3 HSUART 0 特殊功能寄存器 | 57 |
| 13 SPI | 61 |
| 13.1 特征 | 61 |
| 13.2 功能配置 | 61 |
| 13.3 SPI 特殊功能寄存器 | 63 |
| 14 IIC | 65 |
| 14.1 特征 | 65 |
| 14.2 功能配置 | 65 |
| 14.3 IIC 特殊功能寄存器 | 66 |
| 15 ADC | 68 |
| 15.1 特征 | 68 |
| 15.2 通道选择 | 68 |

| | |
|--------------------------------------|-----------|
| 15.3 ADC 时钟配置 | 68 |
| 15.4 功能配置 | 69 |
| 15.5 SARADC_CTL 特殊功能寄存器 | 70 |
| 16 DAC | 75 |
| 16.1 特征 | 75 |
| 16.2 控制使用 | 75 |
| 17 USB | 76 |
| 17.1 特征 | 76 |
| 17.2 控制使用 | 76 |
| 18 电源管理 | 77 |
| 18.1 充电流程 | 77 |
| 18.2 充电设置 | 77 |
| 18.3 充电控制函数 | 77 |
| 19 蓝牙 | 78 |
| 19.1 特征 | 78 |
| 19.2 时序逻辑描述 | 78 |
| 19.3 SPP 协议 | 79 |
| 19.3.1 SPP 协议基于 CREDIT 的流量控制机制 | 79 |
| 19.3.2 SDK 中的 SPP 使用和开发 | 80 |
| 19.4 BLE 协议 | 81 |
| 19.4.1 BLE 协议的流量控制机制 | 81 |
| 19.4.2 SDK 中 BLE 的使用和开发 | 82 |
| 19.5 FOTA 升级 | 82 |
| 19.5.1 特征 SDK 中 FOTA 的使用和开发 | 82 |
| 19.5.2 客户自定义升级的使用和开发 | 83 |
| 19.6 蓝牙地址 | 84 |
| 19.6.1 经典蓝牙 (BR/EDR) | 84 |
| 19.6.2 低功耗蓝牙 (BLE) | 84 |
| 19.7 蓝牙广播 | 85 |
| 20 烧录问题 | 86 |
| 20.1 特征 | 86 |
| 20.2 开发烧录 | 86 |
| 20.3 批量生产烧录问题 | 86 |
| 20.4 芯片 UID | 86 |

| | |
|------------------------|-----------|
| 20.5 安全 | 87 |
| 20.6 产线天线质量测试 | 87 |
| 21 认证 | 88 |
| 21.1 BQB 认证 | 88 |
| 21.2 FCC/SRRC 认证 | 88 |
| 22 版本历史 | 89 |

1 系统框图

图 1 GW3323 系统框图



说明: chx 表示通道编号 x

2.2 时钟寄存器

图 3 时钟门寄存器位定义

| CLOCK GATING BIT ENABLE | | | | | | | | | |
|-------------------------|--------|--|---------|--------|--|---------|------------|--|--|
| CLKGAT0 | | | CLKGAT1 | | | CLKGAT2 | | | |
| 0 | ROM0 | | 0 | - | | 0 | IIC | | |
| 1 | ROM1 | | 1 | - | | 1 | - | | |
| 2 | RAM0 | | 2 | SRC | | 2 | DVP | | |
| 3 | RAM1 | | 3 | IRRX | | 3 | - | | |
| 4 | RAM2 | | 4 | IIS | | 4 | - | | |
| 5 | RAM3 | | 5 | SBCEC | | 5 | - | | |
| 6 | RAM4 | | 6 | - | | 6 | - | | |
| 7 | - | | 7 | PLC | | 7 | - | | |
| 8 | TMR0 | | 8 | TMR3 | | 8 | - | | |
| 9 | SD0 | | 9 | TMR4 | | 9 | - | | |
| 10 | UART0 | | 10 | TMR5 | | 10 | BDP | | |
| 11 | HSUT0 | | 11 | UART2 | | 11 | PLLCLKDIV2 | | |
| 12 | DAC | | 12 | SPI1 | | 12 | PLLCLKDIV4 | | |
| 13 | SARADC | | 13 | - | | 13 | SYSPLLDIV | | |
| 14 | USB | | 14 | - | | 14 | ADDIV | | |
| 15 | SDADC | | 15 | PIANO | | 15 | DAC | | |
| 16 | AUDEC | | 16 | TICK0 | | 16 | DAC_25 | | |
| 17 | PORT | | 17 | - | | 17 | DACDIV2SEL | | |
| 18 | MBIST | | 18 | - | | 18 | DRC_PCLK | | |
| 19 | SPIO | | 19 | - | | 19 | RNN | | |
| 20 | BT | | 20 | - | | 20 | RDFT | | |
| 21 | UART1 | | 21 | - | | 21 | HWMATH | | |
| 22 | SDADCL | | 22 | - | | 22 | - | | |
| 23 | SDADCR | | 23 | AECRAM | | 23 | - | | |
| 24 | TMR1 | | 24 | ROM2 | | 24 | - | | |
| 25 | TMR2 | | 25 | ROM3 | | 25 | - | | |
| 26 | RTCC | | 26 | FRQEDT | | 26 | - | | |
| 27 | RECSRC | | 27 | PBF | | 27 | - | | |
| 28 | - | | 28 | DBG | | 28 | - | | |
| 29 | SPF | | 29 | X26M | | 29 | - | | |
| 30 | AEC | | 30 | M2MDMA | | 30 | - | | |
| 31 | CVSD | | 31 | EFUSE | | 31 | - | | |

寄存器 2-1 CLKCON0: 时钟控制寄存器

| 位数 | 名称 | 模式 | 默认值 | 描述 |
|-------|-------------------|----|-----|--|
| 31 | Sdadc_bqclkx2_sel | WR | 0x0 | sdadc clk 选择 0: addc_clk 1: addc_clkdiv2 |
| 30 | XSOC_x2en_a | WR | 0x0 | XSOC double 使能 0: disable 1: enable |
| 29:28 | Sarclk_sel | WR | 0x0 | Saradc clk 选择 0x0: rc2m 0x1: XSOC_div4 |
| 27:26 | — | — | — | — |
| 25 | Tmrck_async_sel | WR | 0x0 | Timer increase clk 异步选择 0: 选择系统时钟同步的 timer increase clk 1: 选择 timer increase clk |
| 24:23 | Tmrck_sel | WR | 0x0 | Timer increase clk 选择 0x0: osc32k 0x1: clkout 0x2: XSOC_div 0x3: rc2m_d0 |

| 位数 | 名称 | 模式 | 默认值 | 描述 |
|-------|------------|-----|-----|---|
| 22 | PlIts_oe | WRW | 0x0 | PlIts clk 输出使能 0: 输出 disable 1: 输出 enable |
| 21:19 | PlIts_sel | WR | 0x0 | PlIts clk 选择 0x0: pll0_tsck 0x1: pll1_tsck 0x2: bt_sx_tsck 0x3: 0x0 |
| 18:17 | Hutclk_sel | WR | 0x0 | Hsuart clk 选择 Hutclk_sel[0]:0 选择 pll0_out/xosc ,1 选择 pll1_out/xosc52m Hutclk_sel[1]: 0 选择系统时钟(xosc/xosc_double), 1 选择 pll_div clk |

| | | | | |
|-------|--------------|----|-----|---|
| 16:13 | Clkout_sel | WR | 0x0 | 时钟输出选择 0x1: xosc 0x2: XSOC_32k 0x3: osc32k 0x4: pll0div2_clk 0x5: xosc52m 0x6: pll1out 0x7: rc2m 0x8: rtc_rc2m 0x9: sys_clk 0xa: bt26m 0xb: bt_sx_tsck 0xc: dac_clk |
| 12 | Bt13m_sel | WR | 0x0 | Bt13m clk 选择 0: bt13m clk 1: PF0 输入 |
| 11:10 | Bt52m_sel | WR | 0x0 | Bt52m clk 选择: 0: xosc52m 1:无效 |
| 9:8 | - | - | - | - |
| 7 | Bt26m_sel | WR | 0x0 | Bt26m 时钟选择: 0:选择 bt26m clk 1:无效 |
| 6:4 | Syspll_sel_a | WR | 0x0 | Sys pll 选择 Syspll_sel[0]:0 选择 pll_clk,1 选择 xosc52m Syspll_sel[1]:0 选择 pll_clk/xosc52m Syspll_sel[2]:0 选择 pll_clk/xosc52m,1 选择 xosc26m |
| 3:2 | Sysck_sel | WR | 0x0 | Sys clk 时钟选择 00: rc2m clk 01: osc32k clk 10: pll_div clk 11: XSOC_div clk |
| 0 | Rcosc_en_sw | WR | 0x1 | RCOSC 时钟软件使能 0: RCOSC 使能失效 1: RCOSC 使能有效 |

寄存器 2-2 CLKGAT0: 时钟门寄存器

| 位数 | 名称 | 模式 | 默认值 | 描述 |
|----|--------|----|-----|--|
| 31 | CVSD | WR | 0x1 | cvsd clk enable bit 0: disable 1: enable |
| 30 | AEC | WR | 0x1 | aec clk enable bit 0: disable 1: enable |
| 29 | SPF | WR | 0x1 | spf clk enable bit 0: disable 1: enable |
| 28 | SDADCM | WR | 0x1 | sdadcm clk enable bit 0: disable 1: enable |
| 27 | RECSRC | WR | 0x1 | recsrc clk enable bit 0: disable 1: enable |
| 26 | RTCC | WR | 0x1 | rtc clk enable bit 0: disable 1: enable |
| 25 | TMR2 | WR | 0x1 | Timer2 clk enable bit 0: disable 1: enable |
| 24 | TMR1 | WR | 0x1 | Timer1 clk enable bit 0: disable 1: enable |
| 23 | SDADCR | WR | 0x1 | saadcr clk enable bit 0: disable 1: enable |
| 22 | SDADCL | WR | 0x1 | Sdadcl clk enable bit 0: disable 1: enable |
| 21 | UART1 | WR | 0x1 | Uart1 clk enable bit 0: disable 1: enable |
| 20 | BT | WR | 0x1 | bt clk enable bit 0: disable 1: enable |
| 19 | SPI0 | WR | 0x1 | Spi0 clk enable bit 0: disable 1: enable |
| 18 | MBIST | WR | 0x1 | mbist clk enable bit 0: disable 1: enable |
| 17 | PORT | WR | 0x1 | port clk enable bit 0: disable 1: enable |
| 16 | AUDEC | WR | 0x1 | audec clk enable bit 0: disable 1: enable |
| 15 | SDADC | WR | 0x1 | sdadc clk enable bit 0: disable 1: enable |
| 14 | USB | WR | 0x1 | usb clk enable bit 0: disable |

| 位数 | 名称 | 模式 | 默认值 | 描述 |
|----|--------|----|-----|---|
| | | | | 1: enable |
| 13 | SARADC | WR | 0x1 | saradc clk enable bit 0: disable 1: enable |
| 12 | DAC | WR | 0x1 | dac clk enable bit 0: disable 1: enable |
| 11 | HSUT0 | WR | 0x1 | hsuart0 clk enable bit 0: disable 1: enable |
| 10 | UART0 | WR | 0x1 | uart0 clk enable bit 0: disable 1: enable |
| 9 | SD0 | WR | 0x1 | sd0 clk enable bit 0: disable 1: enable |
| 8 | TMR0 | WR | 0x1 | timer0 clk enable bit 0: disable 1: enable |
| 7 | — | WR | 0x1 | |
| 6 | RAM4 | WR | 0x1 | Ram4 clk enable bit 0: disable 1: enable |
| 5 | RAM3 | WR | 0x1 | Ram3 clk enable bit 0: disable 1: enable |
| 4 | RAM2 | WR | 0x1 | Ram2 clk enable bit 0: disable 1: enable |
| 3 | RAM1 | WR | 0x1 | Ram1 clk enable bit 0: disable 1: enable |
| 2 | RAM0 | WR | 0x1 | Ram0 clk enable bit 0: disable 1: enable |
| 1 | ROM1 | WR | 0x1 | Rom1 clk enable bit 0: disable 1: enable |
| 0 | ROM0 | WR | 0x1 | Rom0 clk enable bit 0: disable 1: enable |

寄存器 2-2 CLKCON1: 时钟控制寄存器

| 位数 | 名称 | 模式 | 默认值 | 描述 |
|-------|-----------------|----|-----|---|
| 31 | Dvp_pclk_inv_en | WR | 0x0 | Dvp clk 翻转使能: 0x0:disable 0x1: enable |
| 30:29 | Dvp_delay_sel | WR | 0x0 | Dvp clk delay: 0:disable 1:enable |

| 位数 | 名称 | 模式 | 默认值 | 描述 |
|-------|-----------------|----|-----|---|
| 28:27 | Dvp_outclk_sel | WR | 0x0 | Dvp clk 选择 0x0:syspll_div 0x1:XSOC 0x2:sys_clk |
| 26:25 | Splldiv_sel | WR | 0x0 | Sys pll div 选择: Sysplldiv_sel[0]: 0 选择 syspll_div/bt26m, 1 选择 XSOC_clk/bt26m 取反 Sysplldiv_sel[1]: 0 选择 syspll_div/XSOC, 1 选择 bt26m/bt26m 取反 |
| 24 | Ft_clkpin_sel | WR | 0x0 | FT clk 选择 0x0:PF5 0x1:PA7 |
| 23 | licclk_sel | WR | 0x0 | lic clk 选择 0x0: rc2m 0x1: XSOC_div8 |
| 22 | Pll0sdmsel_a | WR | 0x0 | - |
| 21 | XSOC_lpm_gen | WR | 0x0 | XSOC_lpm enable bit 0x0: disable 0x1: enable |
| 20 | — | — | — | — |
| 19 | Aecram_div1_sel | WR | 0x0 | Aecram div1 选择 0x0:disable 0x1:enable |
| 18 | Usb6p5_sel | WR | 0x0 | - |
| 17 | K32_tscsel | WR | 0x0 | Osc32k 选择 0x0:osc32k 0x1:cp_pin |
| 16 | XSOC_dlysel_a | WR | 0x0 | XSOC clk delay 选择 0: disable 1: enable |
| 15 | Xo52m_insel_a | WR | 0x0 | Xo52m pre 选择 0: xo52m_pre0 1: xo52m_pre1 |
| 14 | Uartck_sel | WR | 0x0 | Uart inc clk 选择 0: XSOC_div 1: XSOC_div4 |
| 13:12 | Ttck_sel | WR | 0x0 | Tick inc clk 选择 0x0: XSOC_div 0x1: pll0div16 0x2: pll0div32 0x3: pll0div64 |
| 11:10 | Usbpll_sel | WR | 0x0 | - |
| 9:8 | lisclk_sel | WR | 0x0 | lis clk 选择 0x0: dac_clk 0x1:xosc52m 0x2:dac_clk/xosc52m(取决于 iisclk_sel[0]) 0x3:iis_div_clk |
| 7:6 | — | — | — | — |

| 位数 | 名称 | 模式 | 默认值 | 描述 |
|-----|--------------|----|-----|---|
| 5:4 | Irrxclkssel | WR | 0x0 | Ir rx clk 选择 0x0:XSOC_32k 0x1:XSOC_div 0x2:osc32k 0x3:rc32k |
| 3:2 | — | — | — | — |
| 1:0 | Adda_clkssel | WR | 0x0 | DAC clk 选择 0x3:adda_clk 0x2:XSOC 0x1:adpill_div_clk 0x0:0 |

寄存器 2-2 CLKGAT1: 时钟门寄存器

| 位数 | 名称 | 模式 | 默认值 | 描述 |
|-------|---------|----|-----|---|
| 31 | EFUSE | WR | 0x1 | efuse clk enable bit 0: disable 1: enable |
| 30 | M2MDMA | WR | 0x1 | M2m clk enable bit 0: disable 1: enable |
| 29 | XSOC | WR | 0x0 | XSOC clk enable bit 0: disable 1: enable |
| 28 | DBG | WR | 0x1 | dbg clk enable bit 0: disable 1: enable |
| 27 | PBF | WR | 0x1 | pbk clk enable bit 0: disable 1: enable |
| 26 | FRQEDET | WR | 0x1 | freqdet clk enable bit 0: disable 1: enable |
| 25 | ROM3 | WR | 0x1 | Rom3 clk enable bit 0: disable 1: enable |
| 24 | ROM2 | WR | 0x1 | Rom2 clk enable bit 0: disable 1: enable |
| 23 | AECRAM | WR | 0x1 | aecram clk enable bit 0: disable 1: enable |
| 22:17 | — | — | 0x1 | — |
| 16 | TICK0 | WR | 0x1 | Tick0 clk enable bit 0: disable 1: enable |
| 15 | PIANO | WR | 0x1 | piano clk enable bit 0: disable 1: enable |
| 14:13 | — | — | 0x1 | — |
| 12 | SPI1 | WR | 0x1 | Spi1 clk enable bit 0: disable |

| 位数 | 名称 | 模式 | 默认值 | 描述 |
|-----|-------|----|-----|--|
| | | | | 1: enable |
| 11 | UART2 | WR | 0x1 | Uart2 clk enable bit 0: disable 1: enable |
| 10 | TMR5 | WR | 0x1 | Timer5 clk enable bit 0: disable 1: enable |
| 9 | TMR4 | WR | 0x1 | Timer4 clk enable bit 0: disable 1: enable |
| 8 | TMR3 | WR | 0x1 | Timer3 clk enable bit 0: disable 1: enable |
| 7 | PLC | WR | 0x1 | plc clk enable bit 0: disable 1: enable |
| 6 | — | — | 0x1 | — |
| 5 | SBCEC | WR | 0x1 | sbcec clk enable bit 0: disable 1: enable |
| 4 | IIS | WR | 0x1 | iis clk enable bit 0: disable 1: enable |
| 3 | IRRX | WR | 0x1 | lrrx clk enable bit 0: disable 1: enable |
| 2 | SRC | WR | 0x1 | src clk enable bit 0: disable 1: enable |
| 1:0 | — | — | 0x1 | — |

寄存器 2-3 CLKCON2: 时钟控制寄存器

| 位数 | 名称 | 模式 | 默认值 | 描述 |
|-------|------------|----|-----|--------------|
| 31:24 | XSOC_div | WR | 0x0 | 外部晶振时钟进行 分频 |
| 23:22 | Hut_div | WR | 0x0 | HSUART 分频 |
| 21:17 | Clkout_div | WR | 0x0 | CLKOUT 分频 |
| 16:13 | Audec_div | WR | 0x0 | AUDECPPLL 分频 |
| 12:8 | Syspll_div | WR | 0x0 | SYSPLL 分频 |
| 7:4 | Adpll_div | WR | 0x0 | ADPLL 分频 |
| 3:0 | Btpll_div | WR | 0x0 | BTPLL 分频 |

寄存器 2-2 CLKGAT2: 时钟门寄存器

| 位数 | 名称 | 模式 | 默认值 | 描述 |
|----|----|----|-----|----|
|----|----|----|-----|----|

| 位数 | 名称 | 模式 | 默认值 | 描述 |
|-------|------------|----|-----|---|
| 31:22 | — | — | 0x1 | — |
| 21 | HWMATH | WR | 0x1 | hwmath clk enable bit 0: disable 1: enable |
| 20 | RDFT | WR | 0x1 | rdft clk enable bit 0: disable 1: enable |
| 19 | RNN | WR | 0x1 | rnn clk enable bit 0: disable 1: enable |
| 18 | DRC_PCLK | WR | 0x1 | drc pclk enable bit 0: disable 1: enable |
| 17 | DACDIV2SEL | WR | 0x0 | Dacdiv2 clk enable bit 0: disable 1: enable |
| 16 | DAC_25 | WR | 0x0 | Dac25 clk enable bit 0: disable 1: enable |
| 15 | DAC | WR | 0x0 | dac clk enable bit 0: disable 1: enable |
| 14 | ADDIV | WR | 0x1 | adddiv clk enable bit 0: disable 1: enable |
| 13 | SYSPLLDIV | WR | 0x1 | sysplldiv clk enable bit 0: disable 1: enable |
| 12 | PLLCLKDIV4 | WR | 0x1 | Plldiv4 clk enable bit 0: disable 1: enable |
| 11 | PLLCLKDIV2 | WR | 0x1 | Plldiv2 clk enable bit 0: disable 1: enable |
| 10 | BDP | WR | 0x1 | bsp clk enable bit 0: disable 1: enable |
| 9:3 | — | WR | 0x1 | 闲置 |
| 2 | DVP | WR | 0x1 | dvp clk enable bit 0: disable 1: enable |
| 1 | — | WR | 0x1 | 闲置 |
| 0 | IIC | WR | 0x1 | iic clk enable bit 0: disable 1: enable |

寄存器 2-4 CLKCON3: 时钟控制寄存器

| 位数 | 名称 | 模式 | 默认值 | 描述 |
|-------|-----------|----|-----|----------|
| 26:23 | piano_div | WR | 0x0 | PIANO 分频 |

| 位数 | 名称 | 模式 | 默认值 | 描述 |
|-------|-----------|----|-----|----------|
| 26:23 | piano_div | WR | 0x0 | PIANO 分频 |
| 26:23 | cvsd_div | WR | 0x0 | CVSD 分频 |
| 22:19 | plc_div | WR | 0x0 | PLC 分频 |
| 18:16 | usb_div | WR | 0x0 | USB 分频 |
| 15:12 | sbcec_div | WR | 0x0 | SBCEC 分频 |
| 11:8 | iis_div | WR | 0x0 | IIS 分频 |
| 7:4 | src_div | WR | 0x0 | SRC 分频 |
| 3:0 | aec_div | WR | 0x0 | AEC 分频 |

寄存器 2-5 CLKCON4: 时钟控制寄存器

| 位数 | 名称 | 模式 | 默认值 | 描述 |
|-------|-------------|----|-----|------------|
| 26:23 | Dvp_out_div | WR | 0x0 | DVP_OUT 分频 |
| 22:20 | hwmath_div | WR | 0x0 | HWMATH 分频 |
| 19:16 | dvp_div | WR | 0x0 | DVP 分频 |
| 11:8 | rnn_div | WR | 0x0 | RNN 分频 |
| 6:0 | Btlp_div | WR | 0x0 | BTLP 分频 |

2.3 低压断电寄存器

LVD 是一种低压断电的电子电路,用于监测电源电压是否低于预设阈值,并在电压低于该阈值时触发断电保护机制。

寄存器 2-6 LVDCON: LVD 控制寄存器

| 位数 | 名称 | 模式 | 默认值 | 描述 |
|-------|---------|----|-----|--|
| 26:23 | - | - | - | - |
| 4 | LVDOE | WR | 0x1 | LVD output enable bit 0:disable 1:enable |
| 3 | LVDEN | WR | 0x1 | LVD enable bit 0:disable 1:enable |
| 2:0 | LVDSSEL | WR | 0x2 | LVD select bit 000:1.8V 001:2.0V 010:2.2V 011:2.4V |

| 位数 | 名称 | 模式 | 默认值 | 描述 |
|----|----|----|-----|--|
| | | | | 100:2.6V 101:2.8V 110:3.0V 111:3.2V |

注：在改变 LVDSEL 前，先关 LVDOE，以避免 LVD 复位。在改变 LVDSEL 后，延时 5us,再使能 LVDOE。

3 内存访问

表格 1 内存访问

| 编号 | 通道 | SRAM0~2 | SRAM3~4 | Cache RAM | AEC RAM | AUDEC RAM | SBCENC RAM | RDFT RAM | RNN RAM |
|----|----------------------------|---------|---------|-----------|---------|-----------|------------|----------|---------|
| 0 | USB BT_RFTS | RW | - | - | RW | - | - | - | - |
| 1 | BT | RW | RW | - | - | - | - | - | - |
| 2 | AUBUF AUBUF1 DACDMAO | RW | RW | RW | RW | RW | RW | - | - |
| 3 | SDADC | W | W | W | W | W | W | - | - |
| 4 | SD SPI0 SPI1 IIS | RW | RW | RW | RW | RW | RW | RW | - |
| 5 | CVSD EQ PSRC PLC | RW | RW | - | RW | RW | RW | - | - |
| 6 | RDFT | RW | RW | - | - | - | - | - | - |
| 7 | AUDEC SBCENC | RW | RW | - | - | RW | - | - | - |
| 8 | GPDMA BSP | RW | RW | - | RW | RW | RW | - | - |
| 9 | HSUT DVP SPF | RW | RW | - | - | - | - | - | - |
| 10 | - | - | - | - | - | - | - | - | - |
| 11 | - | - | - | - | - | - | - | - | - |
| 12 | - | - | - | - | - | - | - | - | - |
| 13 | - | - | - | - | - | - | - | - | - |
| 14 | - | - | - | - | - | - | - | - | - |
| 15 | - | - | - | - | - | - | - | - | - |

DMA 支持外设：HSUART，SPI，SD，USB and IIS，SDADC, SDDAC。

注：RDFT 指傅里叶变换。

GPDMA 指通用 DMA，用于通过链表在外设和/或内存之间传输数据；

BSP (Bitstream pickup) 获取比特流；

DVP, DVP 接口，是一种摄像头的标准协议之一；

SPF (SPDIF 缩写) 光纤音频输出；

4 低功耗模式

GW3323 支持两种低功耗模式：

4.1 睡眠模式(sleep mode,500uA)

睡眠模式将自动关断系统时钟，关闭内存访问，关闭 RC2M，但某些异步时钟应该被软件禁用。

睡眠模式唤醒源，如下所示。唤醒后，如果启用软件运行继续或进入中断。

- 蓝牙唤醒
- 端口外部中断边沿唤醒（PA7、PB1、PB2、PB3、PB4、PB5、INT_FALL、INT_RISE）
- RTC 1s 或报警唤醒

4.1.1 睡眠模式的配置

- (1) 设置并开启唤醒模式;
- (2) 记住所有 IO 口状态后把所有 IO 口设置为输入模式，系统时钟调到 24MHz,关闭 PLL;
- (3) 配置唤醒条件 `wake_config()`;
- (4) 进入睡眠 `bt_sleep_proc()`;
- (5) 被唤醒;
- (6) 恢复 IO 口状态，系统时钟还原，
- (7) `bt_exit_sleep()`;

4.2 呼吸模式(power off,4uA)

呼吸模式将自动关断系统时钟，关闭内存访问，关闭 RC2M，但某些异步时钟应该被软件禁用。

呼吸模式还会通过远程控制来更改 VDDIO/VDDCORE 电压。

呼吸模式模式唤醒源，如下所示。唤醒后，**芯片将重启**。

- 端口外部中断电平唤醒（VUSB、PB0、PB1、PB2、PB5）
- RTC 1s 或报警唤醒

备注：SARADC 等模式打开的情况下，如果是进入低功耗模式，需要手动关闭。

Powerdown 模式下，芯片内部不供电，故 Reset 引脚接的电路此时按复位无效。

当芯片在 Powerdown 模式下时死机了，要断点重启，或者在硬件电路上设计成“Reset 可控制 Vbat 也变低”，造成和断电重启相同的效果。

4.2.1 呼吸模式的配置

- (1) `set_buck_mode(0)`;
- (2) `gpio_deinit()`;
- (3) `poweroff_clk_disable()`;
- (4) 配置唤醒条件 `poweroff_wake_config()`;
- (5) 配置电源、RTC 时钟 `poweroff_config()`;

- (6) cpu 休眠 `WDT_DIS(),LPMCON |= (1 << 0);`
- (7) 被唤醒.

5 中断

5.1 特征

- (1) 库中已设置了蓝牙优先级，由高到底：蓝牙>定时器等其他硬件中断>线程处理。
- (2) 中断中的打印函数要用 `printk` 函数，里面的全局变量要用 `volatile` 申明，中断函数要放到 `com_text` 区域；即添加 `AT(.com_text.isr)`。

以下为示例：

```

AT(.com_rodata.isr)
const char str_1[] = "Tim3 done\r\n ";
AT(.com_rodata.isr)
const char str_2[] = "Tim3 %d\r\n ";

volatile u32 duty_n;
AT(.com_text.isr)
void timer3_isr(void)
{
    if (tmr_get_flag(TMR3, TMR_FLAG_UPDATE2) != RESET) {
        tmr_clear_flag(TMR3, TMR_FLAG_UPDATE2);
        func_A();//例如： gpio_toggle_bits(GPIOB_REG, GPIO_PIN_2);
        printk(str_1);
        printk(str_2,TMR3CON);
    }
}

```

例如 `func_A()`中调用了 `func_B()`,那么 `func_B()`也要放入 `AT(.com_text.isr)`区。

5.2 中断向量表

支持矢量化中断、非法指令上的异常、加载和存储指令到无效地址的异常。

表格 2 异常向量

| 中断编号 | 地址 | 描述 |
|------|------|---|
| 0 | 0x00 | 复位 |
| 1 | 0x04 | 1. 指令读取错误(无效内存访问) 2. 非法指令 3. LSU 错误(无效内存访问) 4. Ebreak 指令 |
| 2 | 0x08 | 硬件断点 |
| 3 | 0x0c | <ul style="list-style-type: none"> • 扩展中断指令 |
| 4 | 0x10 | 低优先级中断 |

| | | |
|---|-----------|----------------|
| 5 | 0x14 | Watch point 中断 |
| 6 | 0x18 | 保留 |
| 7 | 0x1c | 保留 |
| 8 | 0x20~0x9c | 高优先级中断（见下表） |

表格 3 高优先级中断向量

| 中断编号 | 地址 | 描述 |
|------|------|--|
| 0 | 0x20 | Icache miss interrupt Dcache Miss interrupt |
| 1 | 0x24 | BT 中断 BLE 中断 BTDM 中断 |
| 2 | 0x28 | 软件中断（系统调用，底层已固化） |
| 3 | 0x2c | Timer0 中断 |
| 4 | 0x30 | Timer1 中断 |
| 5 | 0x34 | Timer2 中断 |
| 6 | 0x38 | IR 接收器中断 |
| 7 | 0x3c | USB 控制中断 |
| 8 | 0x40 | SD 中断 |
| 9 | 0x44 | 音频 buffer 0 中断 音频 buffer 1 中断 |
| 10 | 0x48 | SDADC DMA 中断 PBF DMA(EQ)中断 |
| 11 | 0x4c | 音频编解码器中断 SBC 编码中断 AEC FFT 中断 |
| 12 | 0x50 | PLC 中断 CVSD 中断 |
| 13 | 0x54 | Piano 中断 SDADC 采样中断 |
| 14 | 0x58 | UART0 中断 UART1 中断 UART2 中断 |
| 15 | 0x5c | HSUART 中断 DVP 中断 |
| 16 | 0x60 | Timer3 中断 |
| 17 | 0x64 | Timer4 中断 |
| 18 | 0x68 | Timer5 中断 |
| 19 | 0x6c | GPDMA 中断 |
| 20 | 0x70 | SPI0 中断 SPI1 中断 |
| 21 | 0x74 | UART0 key 匹配中断 UART1 key 匹配中断 UART2 key 匹配中断 |

| 中断编号 | 地址 | 描述 |
|------|------|--|
| 22 | 0x78 | BT 调制解调器 TX 中断 BT 调制解调器 RX 中断 |
| 23 | 0x7c | 频率检测中断 触控键中断 |
| 24 | 0x80 | DMA 输出中断 RNN 中断 RDFT 中断 HWMATH 中断 |
| 25 | 0x84 | SRC 中断 |
| 26 | 0x88 | 端口中断 (wake 中断) |
| 27 | 0x8c | IIS 中断 |
| 28 | 0x90 | SARADC 中断 |
| 29 | 0x94 | RTC 秒钟和闹钟中断 LVD 中断 WDT 中断 |
| 30 | 0x98 | IIC 中断 BSP 中断 |
| 31 | 0x9c | Tick0 中断 Tick1 中断 |

注：此模块可选择高优先级 1，和低优先级 0。通过调用 `sys_irq_init(int vector, int pr, isr_t isr)`即可配置相关的中断，此部分不开放给客户。vector 是 0~31；pr 是 0 或 1；isr 是中断入口函数。

系统库中配置蓝牙的中断优先级是 1，USB 的中断优先级是 0。

同一个中断号只能有 1 个中断函数。例如，启动 `uart0~2`，都能响应串口中断时，要把这 3 个串口的中断处理函数写在 1 个函数内。

5.3 特殊中断寄存器

寄存器 5-1 PICCON: 外设中断控制寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|-------|---------|----|-----|--|
| 31:17 | - | - | - | 闲置 |
| 16 | GIEM | WR | 1 | 全局中断使能屏蔽位,需要将该寄存器和 GIE 寄存器同时置 1，整个中断程序才能运行。 0: 屏蔽 GIE 寄存器 1: 非屏蔽 GIE 寄存器 |
| 15:7 | - | - | - | 闲置 |
| 6:5 | HPSDEN | WR | 0x0 | 高优先级影子寄存器选择位 00: 高优先级 01: 高优先级 2 10/11:高优先级 3 |
| 4:2 | - | - | - | 闲置 |
| 1 | LPINTEN | WR | 0 | 低优先级中断使能位 0: 禁止 1: 使能 |

| 位数 | 名称 | 模式 | 定义 | 描述 |
|----|-----|----|----|-------------------------------|
| 0 | GIE | WR | 0 | 全局中断使能位 0: 禁止中断 1: 使能中断 |

寄存器 5-2 PICCONSET: 外设中断控制设置寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|-------|---------|----|----|-----------------|
| 31:17 | - | - | - | 闲置 |
| 16 | GIEM | W | 0 | 写入 1 启用全局中断使能掩码 |
| 15:8 | - | - | - | 闲置 |
| 7:3 | - | - | - | 闲置 |
| 2 | HPINTEN | W | 0 | 写入 1 启用高优先级中断 |
| 1 | LPINTEN | W | 0 | 写入 1 启用低优先级中断 |
| 0 | GIE | W | 0 | 写入 1 启用全局中断 |

寄存器 5-3 PICCONCLR: 外设中断控制清除寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|-------|----------|----|----|-----------------|
| 31:17 | - | - | - | 闲置 |
| 16 | GIEMDIS | W | 0 | 写入 1 禁用全局中断启用掩码 |
| 15:8 | - | - | - | 闲置 |
| 7:3 | - | - | - | 闲置 |
| 2 | HPINTDIS | W | 0 | 写入 1 禁用高优先级中断 |
| 1 | LPINTDIS | W | 0 | 写入 1 禁用低优先级中断 |
| 0 | GIEDIS | W | 0 | 写入 1 禁用全局中断 |

寄存器 5-4 PICEN: 外设中断使能寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|------|-------|----|-----|---------------------------------|
| 31:0 | IntEN | WR | 0x0 | 中断 31 至 0 使能位 0: 禁用 1: 启用 |

寄存器 5-5 PICENSET: 外设中断使能设置寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|------|-------|----|-----|------------------|
| 31:0 | IntEN | W | 0x0 | 写入 1 使能中断 31 至 0 |

寄存器 5-6 PICENCLR: 外设中断使能清除寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|------|--------|----|-----|------------------|
| 31:0 | IntDIS | W | 0x0 | 写入 1 禁用中断 31 至 0 |

寄存器 5-7 PICPR: 外设高优先级中断选择寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|----|----|----|----|----|
|----|----|----|----|----|

| | | | | |
|------|-------|----|-----|--|
| 31:0 | IntPR | WR | 0x0 | 中断 31 至 0 优先级选择位 0: 低优先级中断 1: 高优先级中断 |
|------|-------|----|-----|--|

寄存器 5-8 PICPR1: 外设高优先级中断选择寄存器 1

| 位数 | 名称 | 模式 | 定义 | 描述 |
|------|--------|----|-----|---|
| 31:0 | IntPR1 | WR | 0x0 | 中断 31~0 优先级选择 1 位;{PICPR1, PICPR} 00:低优先级中断 01:高优先级中断 10: 高优先级 2 中断 11: 高优先级 3 中断 |

寄存器 5-9 PICADR: 外设中断地址寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|------|------|----|-------|--------|
| 31:8 | BADR | WR | 0x800 | 中断输入地址 |
| 7:0 | - | - | 0x0 | |

寄存器 5-10 PICPND: 外设中断挂起寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|------|--------------|----|-----|---------------------------------------|
| 31:3 | IntPND[31:3] | R | 0x0 | 中断 31 至 3 个挂起位 0: 无中断挂起 1: 中断挂起 |
| 2 | SWIPND | WR | 0 | 软件中断挂起。写入 1 将清除软件中断挂起 |
| 1:0 | IntPND[1:0] | R | 0x0 | 中断 1 到 0 挂起位 0: 无中断挂起 1: 中断挂起 |

5.4 SDK 中的开关所有中断的使用和开发

例程中开关所有中断就是对 PICEN 寄存器进行控制。可参考以下操作：

```
u32 isq_ALL;
```

```
void ARMDisableInt(void)
```

```
{
```

```
    isq_ALL = PICEN;
```

```
    PICEN = 0;
```

```
}
```

```
void ARMEnableInt(void)
```

```
{
```

```
    PICEN = isq_ALL;
```

```
}
```

6 WatchDog

6.1 功能配置

- (1) 配置 WDT 复位或中断
- (2) 选择 WDT 超时
- (3) 清除 WDT

6.2 WDT 特殊功能寄存器

寄存器 6-1 WDTCON: WDT 控制寄存器

| 位 | 名称 | 模式 | 定义 | 描述 |
|-------|-------------|----|-----|---|
| 31 | WDTPND | R | 0 | WDT 超时挂起 0: 无挂起 1: 挂起 |
| 30:28 | - | - | - | 闲置 |
| 27:24 | TMRSEL_WR | W | 0 | WDT 时间选择位写入使能 写入 0xa 时, bit20~bit22 可以写入 TMRSEL, 其他值不会影响 |
| 23 | - | - | - | 闲置 |
| 22:20 | TMRSEL | R | 0x4 | WDT 时间选择位 000: 1ms 001: 256ms 010: 512ms 011: 1024ms 100: 2048ms 101: 4096ms 110: 8192ms 111: 16384ms |
| 19:16 | WDTCSSEL_WR | W | 0 | WDT 时钟选择 当写入 0xa 时, WDTCSSEL = 0, 当写入 0x5 时, WDTCSSEL = 1。 其他值将不受影响 |
| 16 | WDTCSSEL | R | 0 | WDT 时钟选择位 0: RC32K 1: X32K 从 XSOC 分频器 |
| 15:12 | WDTIE_WR | W | 0 | 禁用 WDT 中断 当写入 0xa 时, WDTIE 将禁用, 当写入 0x5 时, WDTIE 将启用。其他值将不受影响 |
| 12 | WDTIE | R | 0 | WDT 中断使能位 0: 禁止 1: 使能 |
| 11:8 | WDTRSTEN_WR | W | 0 | 禁用 WDT 重置 当写入 0xa 时, WDTRSTEN 将禁用, 其他值将不受影响 |
| 8 | WDTRSTEN | WR | 1 | WDT 复位使能位 0: 禁用 1: 启用 |
| 7:4 | WDTEN_WR | W | 0 | 禁用 WDT 当写入 0xa 时, WDTEN 将禁用, 其他值将不受影响 |
| 4 | WDTEN | WR | 1 | WDT 使能位 0: 禁止 1: 使能 |
| 3:0 | WDTCLR | W | 0 | WDT 清除位 当写入 0xa 时, WDT 计数器和 WDTPND 将清除 |

6.3 SDK 中的看门狗的使用和开发

例程中看门狗默认 2048ms 不进行喂狗就复位。调用 WDT_DIS () 可关掉看门狗。

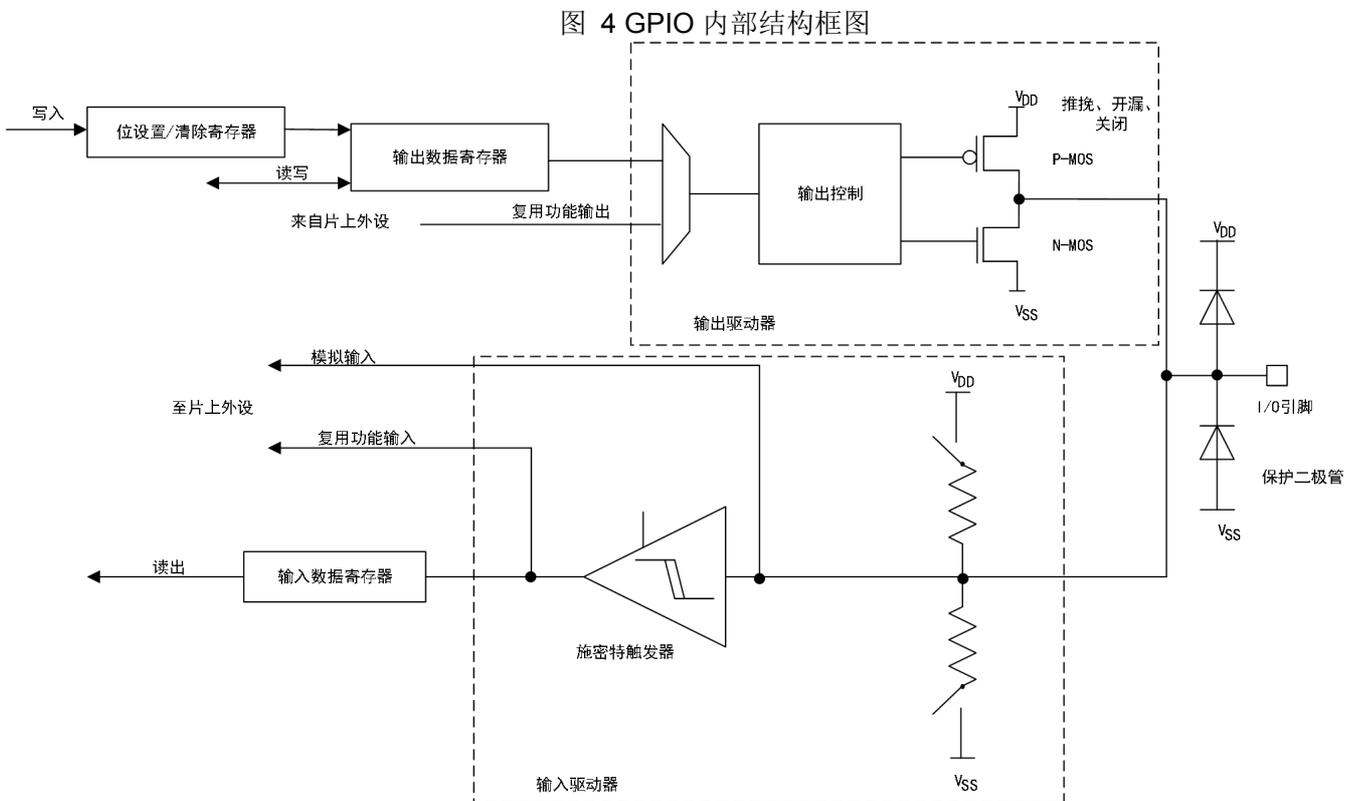
7 GPIO 管理

7.1 特征

- (1) 使用方向寄存器控制 GPIO 输入/输出方向
- (2) 内部上拉/下拉电阻采用上拉/下拉电阻控制寄存器
- (3) 选择合适的输出驱动电流能力(8mA,32mA)

注：上拉的 GPIOxPU、GPIOxPU200K、GPIOxPU300 和下拉的三个，这几个都可以同配置的，使用前最好清楚其他寄存器。即选择上拉 200K 模式时，把 GPIOxPU、GPIOxPU300、GPIOxPD、GPIOxPD200K、GPIOxPD300 对应的位清除，或者直接调用官方例程中的 gpio_init()函数。

7.2 GPIO 内部结构框图



7.3 GPIO 通用控制寄存器

寄存器 7-1 GPIOA: Port A 数据寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|------|-------|----|------|---|
| 31:8 | - | - | - | 闲置 |
| 7:0 | GPIOA | WR | 0x00 | PAx 数据 当 PAx 用作 GPIO 时有效 0: PAx 在读取时输入低电平，在 PAx 写入时输出低电平； 1: PAx 在读取时为输入高电平状态，在写入时在 PAx 处输出为高电平。 |

寄存器 7-2 GPIOASET: Port A 设置输出数据寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|------|----------|----|----|---|
| 31:8 | - | - | - | 闲置 |
| 7:0 | GPIOASET | WO | X | 设置 PAX 输出数据. 写入 1 组输出数据. 写入 0 不会影响任何内容. |

寄存器 7-3 GPIOACLR: Port A 清除寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|------|----------|----|----|---|
| 31:8 | - | - | - | 闲置 |
| 7:0 | GPIOACLR | WO | X | 清除 PAX 输出数据. 写入 1 个清晰的输出数据. 写 0 不会影响任何内容. . |

寄存器 7-4 GPIOADIR: Port A 方向寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|------|----------|----|------|----------------------------|
| 31:8 | - | - | - | 闲置 |
| 7:0 | GPIOADIR | WR | 0xFF | PAX 方向控制 0: 输出 1: 输入 |

寄存器 7-5 GPIOAPU: Port A 上拉电阻寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|------|---------|----|-----|---|
| 31:8 | - | - | - | 闲置 |
| 7:0 | GPIOAPU | WR | 0x0 | PAX 10KΩ上拉电阻控制, 当 PAX 用作输入时有效 0: 禁止 1: 使能 |

寄存器 7-6 GPIOAPD: Port A 下拉电阻寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|------|---------|----|-----|---|
| 31:8 | - | - | - | 闲置 |
| 7:0 | GPIOAPD | WR | 0x0 | PAX 10KΩ下拉电阻控制, 当 PAX 用作输入时有效 0: 禁止 1: 使能 |

寄存器 7-7 GPIOAPU200K: Port A 上拉电阻寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|------|---------|----|-----|--|
| 31:8 | - | - | - | 闲置 |
| 7:0 | GPIOAPU | WR | 0x0 | PAX 200KΩ上拉电阻控制. 当 PAX 用作输入时有效 0: 禁止 1: 使能 |

寄存器 7-8 GPIOAPD200K: Port A 下拉电阻寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|------|---------|----|-----|--|
| 31:8 | - | - | - | 闲置 |
| 7:0 | GPIOAPD | WR | 0x0 | PAX 200KΩ下拉电阻控制, 当 PAX 用作输入时有效 0: 禁止 1: 使能 |

寄存器 7-9 GPIOAPU300: Port A 上拉电阻寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|------|---------|----|-----|---|
| 31:8 | - | - | - | 闲置 |
| 7:0 | GPIOAPU | WR | 0x0 | PAx 300Ω上拉电阻控制. 当 PAx 用作输入时有效 0: 禁止 1: 使能 |

寄存器 7-10 GPIOAPD300: Port A 下拉电阻寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|------|---------|----|-----|---|
| 31:8 | - | - | - | 闲置 |
| 7:0 | GPIOAPD | WR | 0x0 | PAx 300Ω下拉电阻控制, 当 PAx 用作输入时有效 0: 禁止 1: 使能 |

寄存器 7-11 GPIOADE: Port A 数字函数使能寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|------|---------|----|------|--|
| 31:8 | - | - | - | 闲置 |
| 7:0 | GPIOADE | WR | 0xFF | PAx 数字功能使能 0: 用作模拟 IO 的端口 1: 用作数字 IO 的端口 |

寄存器 7-12 GPIOAFEN: Port A 函数映射使能寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|------|----------|----|------|--|
| 31:8 | - | - | - | 闲置 |
| 7:0 | GPIOAFEN | WR | 0xFF | PAx 复用映射使能 0: 用作普通 GPIO 的端口 1: 用作复用 IO 的端口 |

寄存器 7-13 GPIOADRV: Port A 输出驱动选择寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|------|----------|----|-----|---------------------------------|
| 31:8 | - | - | - | 闲置 |
| 7:0 | GPIOADRV | WR | 0x0 | PAx 输出驱动选择 0: 8mA 1: 32mA |

7.4 GPIO 功能映射

寄存器 7-14 FUNCMCN0: 端口功能映射控制寄存器 0

| 位数 | 名称 | 模式 | 定义 | 描述 |
|-------|----------|----|-----|---|
| 31:28 | UT1RXMAP | WR | 0x0 | UART1 RX 映射 0000: 没有影响 0001: 映射到 G1 0010: 映射到 G2 0011: 通过 UT1TXMAP 选择映射到 TX 引脚 1111: 清除这些位 其他保留 |
| 27:24 | UT1TXMAP | WR | 0x0 | UART1 TX 映射 0000: 没有影响 0001: 映射到 G1 0010: 映射到 G2 |

| 位数 | 名称 | 模式 | 定义 | 描述 |
|-------|----------|----|-----|---|
| | | | | 0011: 映射到 G3 1111: 清除这些位 其他保留 |
| 23:20 | - | - | - | - |
| 19:16 | - | - | - | - |
| 15:12 | UT0RXMAP | WR | 0x0 | UART0 RX 映射 0000: 没有影响 0001: 映射到 G1 0010: 映射到 G2 0011: 映射到 G3 0100: 映射到 G4 0101: 映射到 G5 0110: 映射到 G6 0111: 通过 UT0TXMAP 选择映射到 TX 引脚 1111: 清除这些位 其他保留 |
| 11:8 | UT0TXMAP | WR | 0x0 | UART0 TX 映射 0000: 没有影响 0001: 映射到 G1 0010: 映射到 G2 0011: 映射到 G3 0100: 映射到 G4 0101: 映射到 G5 0110: 映射到 G6 0111: 映射到 G7 1000: 映射到 G8 1111: 清除这些位 其他保留 |
| 7:4 | SPI0MAP | WR | 0x0 | SPI0 映射 0000: 没有影响 0001: 映射到 G1 0010: 映射到 G2 0011: 映射到 G3 1111: 清除这些位 其他保留 |
| 3:0 | SD0MAP | WR | 0x0 | SD0 映射 0000: no affect 0001: map to G1 0010: map to G2 0011: map to G3 0100: map to G4 0101: map to G5 0110: map to G6 1111: Clear these bits Others is reserved |

寄存器 7-15 FUNCMCON1: 端口功能映射控制寄存器 1

| 位数 | 名称 | 模式 | 定义 | 描述 |
|-------|----------|----|-----|--|
| 31:12 | - | - | - | - |
| 11:8 | UT2RXMAP | WR | 0x0 | UART2 RX 映射 0000: 没有影响 0001: 映射到 G1 0010: 映射到 G2 0011: 通过 UT2TXMAP 选择映射到 TX 引脚 |

| 位数 | 名称 | 模式 | 定义 | 描述 |
|-----|----------|----|-----|---|
| | | | | 1111: 清除这些位 其他保留 |
| 7:4 | UT2TXMAP | WR | 0x0 | UART2 TX 映射 0000: 没有影响 0001: 映射到 G1 0010: 映射到 G2 1111: 清除这些位 其他保留 |
| 3:1 | clkout | WR | 0x0 | Clkout 映射 0000: 没有影响 0001: 映射到 PA5 0010: 映射到 PA6 0011: 映射到 PB0 0100: 映射到 PB1 0101: 映射到 PE5 0110: 映射到 PE6 其他保留 |
| 0 | - | - | - | - |

寄存器 7-16 FUNCMCON2: 端口功能映射控制寄存器 2

| 位数 | 名称 | 模式 | 定义 | 描述 |
|-------|---------|----|-----|--|
| 31:28 | - | - | - | 闲置 |
| 27:24 | IICMAP | WR | 0x0 | IIC 映射 0000: 没有影响 0001: 映射到 G1 0010: 映射到 G2 0011: 映射到 G3 0100: 映射到 G4 0101: 映射到 G5 0110: 映射到 G6 0111: 映射到 G7 1000: 映射到 G8 1111: 清除这些位 其他保留 |
| 23:20 | IRMAP | WR | 0x0 | IR 映射 0000: 没有影响 0001: 映射到 G1 0010: 映射到 G2 0011: 映射到 G3 0100: 映射到 G4 0101: 映射到 G5 0110: 映射到 G6 0111: 映射到 G7 1000: 映射到 G8 1001: 映射到 G9 1111: 清除这些位 其他保留 |
| 19:16 | TMR5MAP | WR | 0x0 | Timer5 PWM 映射 0000: 没有影响 0001: 映射到 G1 1111: 清除这些位 其他保留 |
| 15:12 | TMR4MAP | WR | 0x0 | Timer4 PWM 映射 |

| 位数 | 名称 | 模式 | 定义 | 描述 |
|------|-----------|----|-----|--|
| | | | | 0000: 没有影响 0001: 映射到 G1 1111: 清除这些位 其他保留 |
| 11:8 | TMR3MAP | WR | 0x0 | Timer3 PWM 映射 0000: 没有影响 0001: 映射到 G1 1111: 清除这些位 其他保留 |
| 7:4 | TMR3PTMAP | WR | 0x0 | Timer3 捕获引脚映射 0000: 没有影响 0001: 映射到 G1 0010: 映射到 G2 0011: 映射到 G3 0100: 映射到 G4 0101: 映射到 G5 0110: 映射到 G6 0111: 映射到 G7 1111: 清除这些位 其他保留 |
| 3:0 | | | | |

寄存器 7-17 FUNCMCON3: 端口功能映射控制寄存器 3

| 位数 | 名称 | 模式 | 定义 | 描述 |
|------|---------|----|-----|--|
| 31:8 | - | - | - | 闲置 |
| 7:4 | MPDMMAP | WR | 0x0 | MPDM 接口映射 0000: 没有影响 0001: 映射到 G1 0010: 映射到 G2 0011: 映射到 G3 0100: 映射到 G4 1111: 清除这些位 其他保留 |
| 3:0 | PDMMAP | WR | 0x0 | PDM 接口（数字麦克风）映射 0000: 没有影响 0001: 映射到 G1 0010: 映射到 G2 0011: 映射到 G3 0100: 映射到 G4 1111: 清除这些位 其他保留 |

7.5 外部端口中断唤醒

支持八路唤醒源输入，如下表所示。唤醒电路 6 和唤醒电路 7 是专门用于 32 端口中断唤醒的。

端口中断源为：Port_intsrc = {PG[4:0], PF[5:0], PE[7:0], PB[4:0], PA[7:0]}。

//PG4 为端口中断 31，PA0 为端口中断 0。

注：当选择 PA0~PA6, PB0, PE0~PE7, PF0~PF5, PG0~PG4 的上升沿/下降沿唤醒时，需要额外设置 PORTINTEN、PORTINTEDG 这 2 个寄存器。

表格 4 外部端口中断唤醒

| 唤醒源 | 唤醒电路 |
|---------------|------------------|
| PA7 | Wakeup circuit 0 |
| PB1 | Wakeup circuit 1 |
| PB2 | Wakeup circuit 2 |
| PB3 | Wakeup circuit 3 |
| PB4 | Wakeup circuit 4 |
| WKO(PB5) | Wakeup circuit 5 |
| PORT_INT_FALL | Wakeup circuit 6 |
| PORT_INT_RISE | Wakeup circuit 7 |

寄存器 7-18 WKUPCON: 唤醒控制寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|-------|------|----|-----|-------------------------------|
| 31:17 | - | - | - | 闲置 |
| 16 | WKIE | WR | 0 | 唤醒中断使能 0: 禁止 1: 使能 |
| 15:8 | - | - | - | 闲置 |
| 7:0 | WKEN | WR | 0x0 | 唤醒输入 7~0 使能 0: 禁用 1: 使能 |

寄存器 7-19 WKUPEDG: 唤醒信号边缘选择寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|-------|-------|----|-----|-------------------------------------|
| 31:24 | - | - | - | 闲置 |
| 23:16 | WKPND | R | 0x0 | 唤醒输入 7~0 待处理 0: 没有待处理 1: 唤醒挂起 |
| 15:8 | - | - | - | 闲置 |
| 7:0 | WKEDG | WR | 0x0 | 唤醒输入 7~0 唤醒边沿选择 0: 上升沿 1: 下降沿 |

寄存器 7-20 WKUPCPND: 唤醒清除挂起寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|-------|--------|----|-----|---------------------------------------|
| 31:8 | - | - | - | 闲置 |
| 23:16 | WKCPND | W | 0x0 | 唤醒输入 7~0 清除挂起 0: 没有影响 1: 清除唤醒挂起 |
| 15:0 | - | - | - | 闲置 |

寄存器 7-21 PORTINTEN: 端口中断使能寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|----|----|----|----|----|
|----|----|----|----|----|

| | | | | |
|------|-----------|----|-----|---------------------------------|
| 31:0 | PORTINTEN | WR | 0x0 | 端口中断 0~31 使能位 0: 禁止 1: 使能 |
|------|-----------|----|-----|---------------------------------|

寄存器 7-22 PORTINTEDG: 端口中断边缘选择寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|------|------------|----|-----|-------------------------------------|
| 31:0 | PORTINTEDG | WR | 0x0 | 端口中断 0~31 边沿选择位 0: 上升沿 1: 下降沿 |

8 DMA

8.1 特征

- (1) DMA 支持外设：HSUART, SPI, SD, USB ， SDADC,SDDAC。
- (2) DMA 不可以进行寄存器与寄存器之间的传输，不可以实现两个内存地址之间的数据搬移。
- (3) DMA-SPI 只支持发送，不支持接收。

8.2 功能配置

表格 5 DMA 相关模块的函数列表

| DMA 模块 | DMA 初始化函数 | DMA 接收函数 | DMA 发送函数 |
|--------|----------------|--------------------|--|
| SD | ‘-’ | sdio_read_data() | sdio_write_data() |
| HSUART | hsuart_init(); | hsuart_dma_start() | hsuart_dma_start() |
| SPI | ‘-’ | 无 | spi_set_dma_addr(); spi_set_dma_cnt() |
| USB | 被封装，用户不可见 | 被封装，用户不可见 | usb_bulk_send() |
| ADC | 暂不开放 | 暂不开放 | 暂不开放 |
| DAC | 暂不开放 | 暂不开放 | 暂不开放 |

9 Timer

9.1 特征

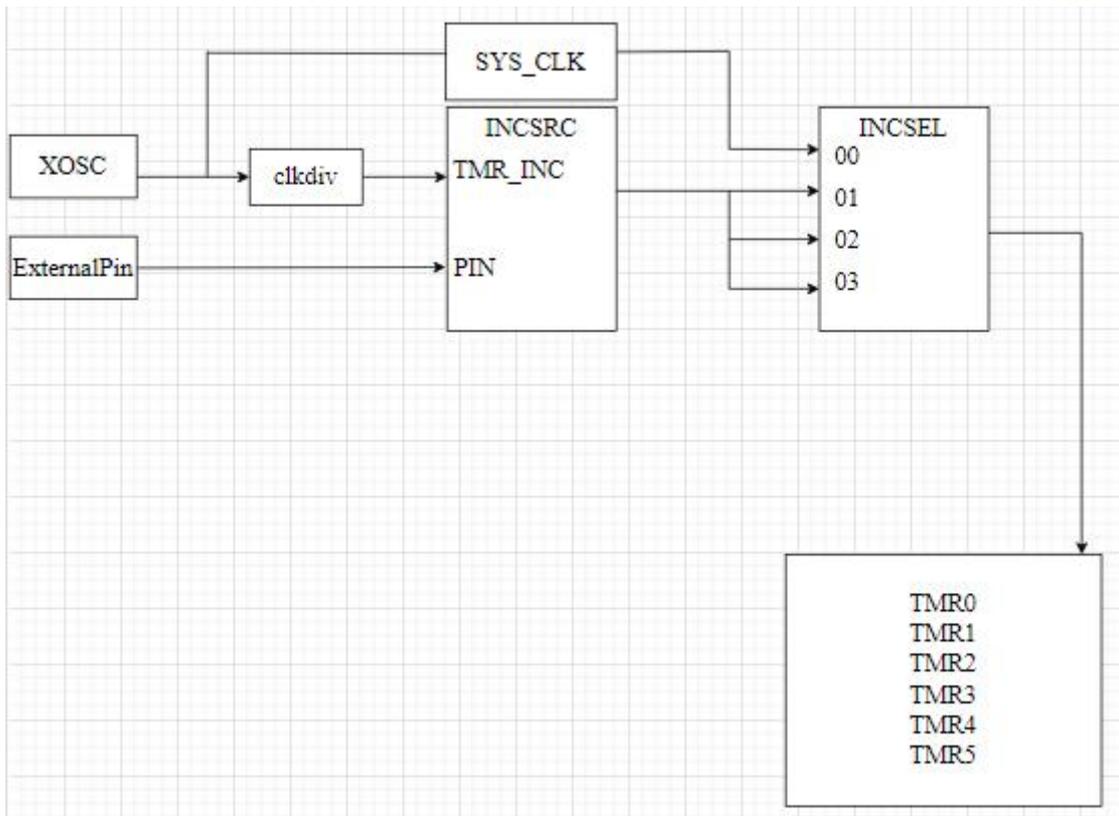
- (1) Timer0/1/2 仅支持 32 位 timer 功能。
- (2) Timer3/4/5 可配置为 timer 模式、计数器模式、捕获模式和 PWM 模式。
- (3) Timer 在主频 160MHz 情况下，选择系统时钟，就可以把定时器频率设置为 160MHz。
- (4) 官方带系统的 SDK,是把 timer0 用作 tick,频率是 1MHz。
- (5) GW3323 定时器的 PWM 为边沿对齐模式（脉冲计数器为循环递增计数，计数初值为 0 ），无中心对齐方式（脉冲计数器为双向计数，计数初值为 0 ），也无互补输出模式。

表格 6 定时器的通道对应的 IO 口表格

| 定时器 | 通道 G1 | 通道 G2 | 通道 G3 | 通道 G4 | 通道 G5 | 通道 G6 | 通道 G7 |
|------------------|-------|-------|-------|-------|-------|-------|-------|
| Timer3-pwm0 | PB0 | PB3 | PF0 | PE0 | - | - | - |
| Timer3-pwm1 | PB1 | PB4 | PA3 | PE4 | - | - | - |
| Timer3-pwm2 | PB2 | PB5 | PA4 | - | - | - | - |
| Timer4-pwm0 | PE5 | PF1 | - | - | - | - | - |
| Timer4-pwm1 | PE6 | PF2 | - | - | - | - | - |
| Timer4-pwm2 | PE7 | PF3 | - | - | - | - | - |
| Timer5-pwm0 | PA5 | PF4 | - | - | - | - | - |
| Timer5-pwm1 | PA6 | PF5 | - | - | - | - | - |
| Timer5-pwm2 | PA7 | - | - | - | - | - | - |
| Timer3-CPT(捕获通道) | PA5 | PA6 | PB0 | PB1 | PE0 | PE5 | PE6 |
| Timer4-CPT(捕获通道) | PE7 | - | - | - | - | - | - |
| Timer5-CPT(捕获通道) | PF1 | - | - | - | - | - | - |

9.2 定时器时钟选择

图 5 定时器时钟结构框图



9.3 Timer0/1/2 特殊功能寄存器

寄存器 9-1 TMR0CON/TMR1CON/TMR2CON: Timer0/1/2 控制寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|-------|--------|----|-----|---|
| 31:10 | - | - | - | 闲置 |
| 9 | TPND | WR | 0 | 计时器溢出挂起 0: 不溢出 1: 溢出 |
| 8 | - | - | - | 闲置 |
| 7 | TIE | WR | 0 | Timer 溢出中断使能 0: 禁止 1: 使能 |
| 6 | INCSRC | WR | 0 | 计数源选择 0: 选择 TMR_INC 1: 选择外部 PIN, 外部时钟源 |
| 5:4 | - | - | - | 闲置 |
| 3:2 | INCSEL | WR | 0x0 | 增加时钟选择 00: 系统时钟 01: 计数器输入上升 10: 计数器输入下降 11: 计数器输入边沿 |
| 1 | - | - | - | 闲置 |
| 0 | TMREN | WR | 0 | Timer 使能位 0: 禁止 1: 使能 |

寄存器 9-2 TMR0CPND/TMR1CPND/TMR2CPND: Timer0/1/2 清除挂起寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|-------|-------|----|----|---------------------------------|
| 31:16 | - | - | - | 闲置 |
| 9 | TPCLR | W | 0 | 计时器溢出挂起的清除位 0: 无效 1: 清除挂起 |
| 8:0 | - | - | - | 闲置 |

寄存器 9-3 TMR0CNT/TMR1CNT/TMR2CNT: Timer0/1/2 计数寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|------|--------|----|-----|---|
| 31:0 | TMRCNT | WR | 0x0 | Timer 计数器 当使能 Timer 的时候, TMRCNT 将增加。当 TMRCNT=TMRPR 时, 它会溢出, 当溢出时, TMRCNT 将清除到 0x0000, 并且中断标志将设置为“1”。 |

寄存器 9-4 TMR0PR/TMR1PR/TMR2PR: Timer0/1/2 周期寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|------|-------|----|------------|---------------------|
| 31:0 | TMRPR | WR | 0xffffffff | Timer 周期= TMRPR + 1 |

9.4 Timer3/4/5 特殊功能寄存器

寄存器 9-5 TMR3CON/TMR4CON/TMR5CON: Timer3/4/5 控制寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|-------|--------|----|----|----------------------------------|
| 31:18 | - | - | - | 闲置 |
| 17 | CPND | WR | 0 | Timer 捕获挂起 0: 不捕获 1: 捕获 |
| 16 | TPND | WR | 0 | 计时器溢出挂起 0: 不溢出 1: 溢出 |
| 15:12 | - | - | - | 闲置 |
| 11 | PWM2EN | WR | 0 | Timer pwm2 使能位 0: 禁止 1: 使能 |
| 10 | PWM1EN | WR | 0 | Timer pwm1 使能位 0: 禁止 1: 使能 |
| 9 | PWM0EN | WR | 0 | Timer pwm0 使能位 0: 禁止 1: 使能 |
| 8 | CIE | WR | 0 | Timer 捕获中断使能 0: 禁止 1: 使能 |
| 7 | TIE | WR | 0 | Timer 溢出中断使能 0: 禁止 1: 使能 |

| 位数 | 名称 | 模式 | 定义 | 描述 |
|-----|----------|----|-----|--|
| 6 | INCSRC | WR | 0 | 增加源选择 0: 选择 TMR_INC 1: 选择外部 PIN |
| 5:4 | CPTEDSEL | WR | 0x0 | Timer 捕获边缘选择 00: 不捕获 01: 捕获 PIN 上升沿 10: 捕获 PIN 下降沿 11: 捕获 PIN 边缘 |
| 3:2 | INCSEL | WR | 0x0 | 增加时钟选择 00: 系统时钟 01: 计数器输入上升 10: 计数器输入下降 11: 计数器输入边缘 |
| 1 | CPTEN | WR | 0 | Timer 捕获使能位 0: 禁止 1: 使能 |
| 0 | TMREN | WR | 0 | Timer 使能位 0: 禁止 1: 使能 |

寄存器 9- 6 TMR3CPND/TMR4CPND/TMR5CPND: Timer3/4/5 清除挂起寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|-------|-------|----|----|------------------------------------|
| 31:18 | - | - | - | 闲置 |
| 17 | CPCLR | W | 0 | 捕获挂起的清除位 0: 无效 1: 清除挂起 |
| 16 | TPCLR | W | 0 | Timer 溢出挂起的清除位 0: 无效 1: 清除挂起 |
| 15:0 | - | - | - | 闲置 |

寄存器 9- 7 TMR3CNT/TMR4CNT/TMR5CNT: Timer3/4/5 计数寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|------|--------|----|-----|---|
| 31:0 | TMRCNT | WR | 0x0 | Timer 计数器 使能 Timer 时 TMRCNT 将增加。当 TMRCNT=TMRPR 时，它会溢出；当溢出时，TMRCNT 将清除到 0x0000，并且中断标志将设置为“1”。 |

寄存器 9- 8 TMR3PR/TMR4PR/TMR5PR: Timer3/4/5 周期寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|------|-------|----|------------|------------------|
| 31:0 | TMRPR | WR | 0xffffffff | Timer 周期=TMRPR+1 |

寄存器 9- 9 TMR3CPT/TMR4CPT/TMR5CPT: Timer3/4/5 捕获数值寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|------|--------|----|----|------------|
| 31:0 | TMRCPT | R | x | Timer 捕获数值 |

寄存器 9- 10 TMR3DUTY0/TMR4DUTY0/TMR5DUTY0: Timer3/4/5 pwm0 占空寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|-------|----------|----|----|--|
| 31:16 | - | - | - | 闲置 |
| 15:0 | TMRDUTY0 | W | x | Timer pwm0 占空比 PWM0 低电平长度为 TMRDUTY0+1 PWM0 高电平长度为 TMRPR-TMRDUTY0 |

寄存器 9- 11 TMR3DUTY1/TMR4DUTY1/TMR5DUTY1: Timer3/4/5 pwm1 占空寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|-------|----------|----|----|--|
| 31:16 | - | - | - | 闲置 |
| 15:0 | TMRDUTY1 | W | x | Timer pwm1 占空比 PWM1 低电平长度为 TMRDUTY1+1 PWM1 高电平长度为 TMRPR-TMRDUTY1 |

寄存器 9- 12 TMR3DUTY2/TMR4DUTY2/TMR5DUTY2: Timer3/4/5 pwm2 占空寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|-------|----------|----|----|--|
| 31:16 | - | - | - | 闲置 |
| 15:0 | TMRDUTY2 | W | x | Timer pwm2 占空比 PWM2 低电平长度为 TMRDUTY2+1 PWM2 高电平长度为 TMRPR-TMRDUTY2 |

9.5 系统时钟函数

```
tick_check_expire(u32 tick, u32 expire_val)//当时间超过 expire_val*1ms 后, 返回 ture。
tick_get()//
```

10 RTC

10.1 特征

- (1) 支持 32 位独立电源实时计数器。
- (2) 支持报警中断和二次中断。

当启用 RTC 唤醒时，有 LDO 和 Buck 这 2 种模式进行选择。

10.2 特殊功能寄存器

寄存器 10-1 RTCCON: RTC 控制寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|-------|-------------|----|----|--|
| 31:23 | - | - | - | 闲置 |
| 22 | INBOX | R | 0 | INBOX 状态 0: box 之外 1: box 里面，检测 VUSB 的管脚电压高于 2V 左右则置 1 |
| 21 | VUSBOFF | R | 0 | VUSB 关闭状态 0: 在线 1: 关闭状态，检测 VUSB 的管脚电压低于(Vbat-0.8) |
| 20 | VUSBONLINE | R | 0 | VUSB 在线状态 0: 不在线 1: 在线，检测 VUSB 的管脚电压是否高于 5V |
| 19 | RTCWKP | R | 0 | RTC WK 引脚状态 0: WK 引脚状态为 0 1: WK 引脚状态为 1 |
| 18 | RTCWKSLPPND | R | 0 | RTC 唤醒睡眠挂起 0: 无待处理 1: 待定，从唤醒模式唤醒后自动变 1 |
| 17 | ALMPND | R | 0 | RTC 闹钟挂起 0: 无待处理 1: 报警待定 |
| 16:9 | - | - | - | 闲置 |
| 8 | ALM_WKEN | WR | 0 | RTC 闹钟唤醒使能 0: 禁用 1: 启用 |
| 7 | RTC_WKSLPEN | WR | 0 | RTC 闹钟睡眠使能 0: 禁用 1: 启用 |
| 6 | VUSBIRSTEN | WR | 0 | VUSB 插入复位系统使能 0: 禁用 1: 启用，充电时想让芯片复位就打开此位 |
| 5 | WKUPRSTEN | WR | 0 | RTC 唤醒关断模式复位系统使能 0: 禁用 1: 启用。用 4uA 低功耗模式前，就要使能，否则唤醒后无法复位，无法正常工作。 |
| 4 | ALMIE | WR | 0 | RTC 闹钟中断使能 |

| 位数 | 名称 | 模式 | 定义 | 描述 |
|-----|---------|----|-----|---|
| | | | | 0: 禁用 1: 启用 |
| 3 | RTC1SIE | WR | 0 | RTC 1S 中断使能, 当 rtc1s_wkslp_pnd 到来时, 若 rtc1s_ie 使能有效, rtc_int 中断有效 0: 禁用 1: 启用 |
| 2:1 | BAUDSEL | WR | 0x1 | 增加时钟选择 00: 系统时钟分频 4 01: 系统时钟分频 8 10: 系统时钟分频 16 11: 系统时钟分频 32 |
| 0 | - | - | - | 闲置 |

注: INBOX 指检测 VUSB 的管脚电压高于 2V 左右则置 1,一般用于检测耳机充电仓是否插入。

寄存器 10-2 RTCCPND: RTC 清除挂起寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|-------|-----------|----|----|--------------------------------------|
| 31:19 | - | - | - | 闲置 |
| 18 | CWKSLPPND | W | 0 | 写入 1 将清除 RTC 唤醒睡眠挂起, RTCWKSLPPND 变 0 |
| 17 | CALMPND | W | 0 | 写入 1 将清除 RTC 闹钟挂起, ALMPND 变 0 |
| 16:0 | - | - | - | 闲置 |

10.3 独立电源 RTC 寄存器

寄存器 10-3 RTCCNT: RTC 计数寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|------|--------|----|-----|-----------------------------|
| 31:0 | RTCCNT | WR | 0x0 | 32bit RTC 计数器, 此单位是 1s,无法修改 |

寄存器 10-4 RTCALM: RTC 闹钟寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|------|--------|----|------------|--------------|
| 31:0 | RTCALM | WR | 0xffffffff | 32bit RTC 警报 |

寄存器 10-5 RTCCON0: RTC 控制寄存器 0

| 位数 | 名称 | 模式 | 定义 | 描述 |
|-------|----------|----|-----|---------------------------------|
| 31:18 | - | - | - | - |
| 17 | TKSWRSTN | WR | 0 | 触摸键复位 0: 触摸键复位 1: 释放触摸键复位 |
| 16:15 | VRTCSEL | WR | 0x0 | — |
| 14 | RCSEL | WR | 0 | RC 选择 0: RCOSC 1:RING RC |

| 位数 | 名称 | 模式 | 定义 | 描述 |
|-------|--------------|----|-----|---|
| 13 | RCOSCEN | WR | 0 | — |
| 12 | VIOAONS | WR | 0 | — |
| 11:10 | CLK2MBTSSSEL | WR | 0x0 | CLK2M 至 BT 低功耗时钟选择位 00: 0 01: RTC 时钟 32K 10: RTC 2M 11: XOSC 分频 8 (3.25M) |
| 9 | CLK2MTKSSEL | WR | 0 | 触摸键电源域源选择位中的 CLK2M 0: RTC 2M 1: XOSC 分频 8 |
| 8 | CLK2MRTCSSEL | WR | 0 | RTC 电源域选择位中的 CLK2M 0: RTC 2M 1: XOSC 分频 8 |
| 7 | PWRUP1ST | WR | 1 | RTC 第一个通电标志 0: 不是第一次通电 1: 首次通电。连续拔插电源, 此位会从 1 变成 0。拔掉 (长时间, 芯片不工作) 就变 1。 |
| 6 | EXT32KS | WR | 0 | 外部 32K 选择 0: 使用 RTC 内部 32K 振荡器 1: 使用外部 32K 振荡器 |
| 5 | - | - | - | - |
| 4 | TKITF_EN | WR | 1 | 内核接口之间触摸键使能位 0: 禁用 1: 启用 |
| 3 | SNIFF_EN | WR | 0 | sniff 模式禁用 VDDCORE_EN 启用 0: 禁用 1: 启用 |
| 2 | CLK2M_EN | WR | 0 | CLK2M 分频至 RTC 32K 时钟源使能 0: 禁用 1: 启用 |
| 1 | X32KEN | WR | 0 | XOSC32K 使能位 0: 禁用 1: 启用 |
| 0 | RCEN | WR | 0 | RCOSC 使能位 0: 禁用 1: 启用 |

寄存器 10-6 RTCCON1: RTC 控制寄存器 1

| 位数 | 名称 | 模式 | 定义 | 描述 |
|----|---------|----|----|---|
| 7 | VRTCEN | WR | 0 | 用于 ADC 的 VRTC 使能位、VRTC 电压 0: 禁用 1: 启用 |
| 6 | WKPLVLS | WR | 0 | WK 引脚唤醒电平选择位 0: 低电平唤醒 1: 高级唤醒 |
| 5 | WKPAEN | WR | 0 | WK 引脚模拟使能位, 用于 ADC 的输出 WKO 电压 0: 禁用 |

| 位数 | 名称 | 模式 | 定义 | 描述 |
|-----|---------|----|-----|--|
| | | | | 1: 启用 |
| 4 | WKPPUEN | WR | 1 | WK 引脚上拉使能位 0: 禁用 1: 启用 |
| 3:2 | WKPPUS | WR | 0x1 | WK 引脚上拉选择位 00: 80K 01: 90K 10: 100K 11: 400K |
| 1 | WKPPD | WR | 0 | WK 引脚下拉 10K 使能位 0: 禁用 1: 启用 |
| 0 | WKPIE | WR | 1 | WK 引脚输入使能位 0: 禁用 1: 启用 |

寄存器 10-7 RTCCON2: RTC 控制寄存器 2

| 位数 | 名称 | 模式 | 定义 | 描述 |
|-----|----------|----|-----|-------------------------------------|
| 7 | SELVDDPU | WR | 1 | 使能 SEL VDD 上拉 0: 禁用 1: 启用 |
| 6 | 32KSEL | WR | 0 | 32K osc 选择位 0: 32.768K 1: 32K |
| 5:4 | RSV | WR | 0x0 | 保留, 无法更改默认值 |
| 3:2 | RSV | WR | 0x0 | 保留, 无法更改默认值 |
| 1:0 | RSV | WR | 0x0 | 保留, 无法更改默认值 |

寄存器 10-8 RTCCON3: RTC 控制寄存器 3

| 位数 | 名称 | 模式 | 定义 | 描述 |
|----|------------|----|----|-------------------------------|
| 15 | - | - | - | 闲置 |
| 14 | TK_WKEN | WR | 0 | 触摸键长按唤醒启用位 0: 禁用 1: 启用 |
| 13 | - | - | - | Unused |
| 12 | INBOX_WKEN | WR | 0 | INBOX 唤醒启用位 0: 禁用 1: 启用 |
| 11 | VSUB_WKEN | WR | 0 | VUSB 唤醒使能位 0: 禁用 1: 启用 |
| 10 | WKP_WKEN | WR | 0 | WK 管脚唤醒使能位 0: 禁用 1: 启用 |
| 9 | RTC1S_WKEN | WR | 0 | RTC1 秒唤醒使能位 0: 禁用 1: 启用 |

| 位数 | 名称 | 模式 | 定义 | 描述 |
|----|------------|----|----|-----------------------------------|
| 8 | ALM_WKEN | WR | 0 | RTC 闹钟唤醒使能位 0: 禁用 1: 启用 |
| 7 | - | - | - | 闲置 |
| 6 | PDCOREEN | WR | 1 | 内核断电使能位 0: 禁用 1: 启用 |
| 5 | VCORESHTEN | WR | 1 | VDDCORE 短使能位 0: 禁用 1: 启用 |
| 4 | VDDXOEN | WR | 1 | VDDXO 使能位 0: 禁用 1: 启用 |
| 3 | VCOREAONEN | WR | 1 | VDDCORE AON 使能位 0: 禁用 1: 启用 |
| 2 | VCOREEN | WR | 1 | VDDCORE 使能位 0: 禁用 1: 启用 |
| 1 | VIOEN | WR | 1 | VDDIO 使能位 0: 禁用 1: 启用 |
| 0 | BUCKEN | WR | 0 | BUCK 使能位 0: 禁用 1: 启用 |

寄存器 10-9 RTCCON5: RTC 控制寄存器 5

| 位数 | 名称 | 模式 | 定义 | 描述 |
|-----|---------|----|-----|--|
| 7 | RSV | WR | 0 | 保留, 无法更改默认值。 |
| 6 | RSV | WR | 0 | 保留, 无法更改默认值。 |
| 5:4 | RSV | WR | 0x0 | 保留, 无法更改默认值。 |
| 3:2 | RSV | WR | 0x0 | 保留, 无法更改默认值。 |
| 1 | BUCKLPM | WR | 0 | BUCK 低功耗模式使能 0: 禁用 1: 启用 |
| 0 | LDO | WR | 1 | BUCK LDO 模式选择位 0: buck 模式。DC-DC 要外挂电感, 功耗低, 转换效率高, 500uA 低功耗在此模式下测的。 1: LDO 模式 |

寄存器 10-10 RTCCON10: RTC 控制寄存器 10

| 位数 | 名称 | 模式 | 定义 | 描述 |
|----|---------|----|----|--|
| 10 | WKP10SC | WR | 0 | 写入时: WK 引脚 10 秒待定清除 0: 无影响 1: 清除 10 秒待定 阅读时: WK 引脚 10s 待定 |

| 位数 | 名称 | 模式 | 定义 | 描述 |
|----|---------|----|----|--|
| | | | | 0: 没有 10 秒待定 1: 10 秒待定 |
| 9 | WK3P | R | 0 | WK pin3 唤醒挂起 0: 不挂起 1: 挂起 |
| 8 | WK2P | R | 0 | WK pin2 唤醒挂起 0: 不挂起 1: 挂起 |
| 7 | WK1P | R | 0 | WK pin1 唤醒挂起 0: 不挂起 1: 挂起 |
| 6 | TKP | R | 0 | TK 唤醒挂起 0: 不挂起 1: 挂起 |
| 5 | - | - | - | 闲置 |
| 4 | INBOXP | R | 0 | INBOX 唤醒挂起 0: 不挂起 1: 挂起 |
| 3 | VUSBP | R | 0 | VUSB 唤醒挂起 0: 不挂起 1: 挂起 |
| 2 | WKP | R | 0 | WK 管脚唤醒挂起 0: 不挂起 1: 挂起 |
| 1 | RTC1SPC | WR | 0 | 写入时: RTC 1s 清除挂起 0: 没有影响 1: 清除 1s 挂起 读取时: RTC 1s 挂起 0: 没有秒挂起 1: 秒挂起 |
| 0 | ALMPC | WR | 0 | 写入时: RTC 警报挂起清除 0: 没有影响 1: 清除警报挂起 读取时: 警报挂起 0: 没有警报挂起 1: 警报挂起 |

寄存器 10-11 RTCCON11: RTC 控制寄存器 11

| 位数 | 名称 | 模式 | 定义 | 描述 |
|-----|------------|----|-----|---|
| 10 | RTCWKSLEN | WR | 0 | RTC 定时器唤醒睡眠使能 0: 禁用 1: 启用 |
| 9:8 | RTCWKSLEPS | WR | 0x0 | RTC 定时器唤醒睡眠时间选择 00: 110ms 01: 220ms 10: 440ms 11: 880ms |

| 位数 | 名称 | 模式 | 定义 | 描述 |
|-----|-------------|----|-----|---|
| 7 | VIOCHG_SWEN | WR | 0 | VUSB 到 VDDIO LDO 选择控制位 0: VUSB 到 VDDIO LDO 通过 pmu_normal & VIOCHG_EN 启用 1: VUSB 到 VDDIO LDO 通过 VIOCHG_SWEN & VIOCHG_EN 启用 |
| 6 | VUSBWKSEL | WR | 0 | VUSB 唤醒选择 0: VUSB 插入滤波器唤醒 1: VUSB 拔出滤波器唤醒 |
| 5 | VUSBFILSEL | WR | 0 | VUSB 关闭滤波器选择 0: 840us 1: 12ms |
| 4 | WKOPRT | WR | 0 | WKO 保护位 |
| 3 | - | - | - | - |
| 2 | WKPFEN | WR | 1 | WK 管脚滤波器使能位 0:禁止 1:使能 |
| 1:0 | WKPFSEL | WR | 0x0 | WK 管脚滤波器选择位，有效电平超过此时间后才唤醒 00:8ms 01:32ms 10:128ms 11:512ms |

寄存器 10-12 RTCCON12: RTC 控制寄存器 12

| 位数 | 名称 | 模式 | 定义 | 描述 |
|-----|----------|----|-----|---|
| 7:4 | - | - | - | 闲置 |
| 3:0 | WKP10SEN | WR | 0xa | WK 管脚 10s 复位使能 0xa: 禁止 Others: 使能 |

11 UART

11.1 特征

- (1) UART 是一个能够异步传输的串口。
- (2) UART 可以在全双工模式下工作。

表格 7 串口的通道对应的 IO 口表格

| 串口 | 通道 G1 | 通道 G2 | 通道 G3 | 通道 G4 | 通道 G5 | 通道 G6 | 通道 G7 | 通道 G8 | 通道 G9 | 通道 G11 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|
| TX0 | PA7* | PB2 | PB3 | PE7 | PE0 | PF1 | PF5 | VUSB | - | - |
| RX0 | PA6 | PB1 | PB4 | PE6 | - | - | - | - | - | - |
| TX1 | PA7* | PA4 | VUSB | - | - | - | - | - | - | - |
| RX1 | PA6 | PA3 | - | - | - | - | - | - | - | - |
| TX2 | - | PB2 | VUSB | - | - | - | - | - | - | - |
| RX2 | - | PB1 | - | - | - | - | - | - | - | - |
| HSTTX | - | - | - | - | - | - | - | - | - | - |
| HSTRX | PA7 | PB2 | PB3 | PE7 | - | PA6 | PB1 | PB4 | PE6 | VUSB |

11.2 UART 特殊功能寄存器

寄存器 11- 1 UARTCON: UART 控制寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|-------|---------|----|----|---------------------------------------|
| 31:10 | - | - | - | 闲置 |
| 9 | RXPND | R | 0 | RX 挂起 0: RX 单字节未完成 1: RX 单字节完成 |
| 8 | TXPND | R | 0 | TX 挂起 0: TX 单字节未完成 1: TX 单字节完成 |
| 7 | RXEN | WR | 0 | RX 使能 0: RX 禁止 1: RX 使能 |
| 6 | ONELINE | WR | 0 | 单行模式 0: TX/RX 分离 1: TX/RX 单线 |
| 5 | CLKSRC | WR | 0 | 时钟源选择 0: 系统时钟 1: uart_inc |
| 4 | SB2EN | WR | 0 | 两个停止位使能 0: 1bit 停止位 1: 2bit 停止位 |
| 3 | TXIE | WR | 0 | 传输中断使能 0 = 传输中断禁用 |

| 位数 | 名称 | 模式 | 定义 | 描述 |
|----|--------|----|----|--|
| | | | | 1 =传输中断使能 |
| 2 | RXIE | WR | 0 | 接收器中断使能 0: 接收器中断禁用 1:接收器中断使能 |
| 1 | BIT9EN | WR | 0 | BIT9 使能位 0: 8bit 模式 1: 9bit 模式 |
| 0 | UTEN | WR | 0 | UART 使能位 0: 禁用 UART 模块 1: 使能 UART 模块 |

寄存器 11- 2 UARTCPND: UART 清除挂起寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|-------|------------|----|----|---|
| 31:18 | - | - | - | 闲置 |
| 17 | CRSTKEYPND | W | 0 | 重置密钥匹配挂起清除 0: N/A 1: 清除重置密钥匹配挂起 |
| 16 | CKEYPND | W | 0 | 密钥匹配待清除 0: N/A 1:清除密钥匹配挂起 |
| 15:10 | - | - | - | 闲置 |
| 9 | CRXPND | W | 0 | RX 挂起清除 0: N/A 1: 清除 RX 挂起 |
| 8 | CTXPND | W | 0 | TX 挂起清除 0: N/A 1: 清除 TX 挂起, 写入数据到 UTBUF 将清除 TXPND |
| 7:0 | - | - | - | 闲置 |

寄存器 11- 3 UARTBAUD: UART 波特率寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|-------|------------|----|----|--|
| 31:16 | UARTRXBAUD | W | 0 | UART RX 波特率 波特率= $F_{\text{sys clock}} / (\text{UART0RXBAUD} + 1)$ |
| 15:0 | UARTTXBAUD | W | 0 | UART TX 波特率 波特率 = $F_{\text{sys clock}} / (\text{UART0TXBAUD} + 1)$ |

寄存器 11- 4 UARTDATA: UART 数据寄存器

| 位数 | 名称 | 模式 | 定义 | 描述 |
|------|----------|----|----|--|
| 31:9 | - | - | - | 闲置 |
| 8 | UARTBIT8 | WR | x | UART 数据 bit 8 |
| 7:0 | UARTDAT | WR | x | UART 数据 写入此寄存器会将数据加载到发送器缓冲器。 读取此寄存器将从接收器缓冲器读取数据。 |

11.3 功能配置

- (1) 将 IO 设置为正确的方向。
- (2) 配置 UART0BAUD 以选择采样率。
- (3) 通过设置启用 UART0。
- (4) 如果需要，将 TXIE 或 RXIE 设置为“1”。
- (5) 将数据写入 UART0DATA。
- (6) 等待 PND 更改为“1”，或等待中断。
- (7) 如果需要，从 UART0DATA 读取接收到的数据。

11.4 SDK 中的串口使用和开发

Printf 是固化在库里，对用户不可见，默认 PB3 作为 uart0 的输出，波特率 1.5MHz。

12 HSUART

12.1 特征

- (1) 支持全双工模式
- (2) 支持 UART 接口部分与控制部分之间的异步时钟
- (3) 支持中断
- (4) 支持 8/9 位数据模组，但不支持数据奇偶校验。
- (5) 支持 16 位波特率设置。

● HUART TX:

- 支持单字节 TX
- 支持 DMA TX，最小 1 字节，最大 1024 字节
- 支持按字节配置 DMA 长度
- 支持按字节配置 DMA 地址
- 支持 1/2 停止位
- 支持 TXIE 控制
- 支持 TX 标志位

● HUART RX:

- 支持单字节 RX
- 支持 DMA RX，最小 1 字节，最大 1024 字节
- 支持循环缓冲 DMA 模式
- 支持按字节配置 DMA 长度
- 支持按字节配置 DMA 地址
- 始终 1 个停止位，不检查停止位
- RX DMA 仅 8 位数据模式
- 支持 RXIE 控制
- 支持 RX 标志位

12.2 功能配置

12.2.1 SYNC 配置

- (1) 通过 RXBITSEL_NS, HSUTTMREN_NS, HSUT0TMR 配置 RX 8/9 位数据模式。
- (2) 配置 TX TXBITSEL_NS, SPBITSEL_NS
- (3) 配置 HSUT0BAUD
- (4) 配置 RXEN_NS, TXEN_NS
- (5) 设置 UPRXCFG (HSUT0CON |= 1 << 16;)
- (6) 设置 UPTXCFG (HSUT0CON |= 1 << 17;)

12.2.2 TX 1 byte 带 buffer

- (1) 按正确的方向配置 GPIO。
- (2) 配置 HSUT0CON CLKSRC 位以实现高速接口工作源时钟
- (3) 配置 HSUT0CON 缓冲模式，8/9 位模式，1/2 停止位模式
- (4) 配置 HSUT0BAUD 波特率

- (5) 置位使能 UTXEN
- (6) 设置 UPTXCFG 同步设置到接口时钟域
- (7) 根据需要配置 TXIEto1'
- (8) 将数据写入 HSUT0DATA 以启动 TX
- (9) 等待 TXPND 变为“1”，或等待中断
- (10) 清除 PNDing

12.2.3 TX n byte 带 DMA

- (1) 按正确的方向配置 GPIO 。
- (2) 配置 HSUT0CON CLKSRC 位以实现高速接口工作源时钟
- (3) 配置 HSUT0CON DMA 模式，8 位模式，1/2 停止位模式
- (4) 配置 HSUT0BAUD 波特率
- (5) 配置 TX DMA 起始地址 HSUT0TXADR
- (6) 置位使能 UTXEN
- (7) 设置 UPTXCFG 来同步设置到接口时钟域
- (8) 根据需要配置 TXIEto1'
- (9) 将数据计数写入 HSUT0TXCNT 以启动 TX
- (10) 等待 TXPND 变为“1”，或等待中断
- (11) 清除 PNDing

12.2.4 RX 1 byte 带 buffer

- (1) 按正确的方向配置 GPIO 。
- (2) 配置 HSUT0CON CLKSRC 位以实现高速接口工作源时钟
- (3) 配置 HSUT0CON buffer 模式,8/9 位模式
- (4) 配置 HSUT0BAUD 波特率
- (5) 设置使能 URXEN
- (6) 根据需要配置 RXIEto1'
- (7) 设置 UPRXCFG 来同步设置到接口时钟域
- (8) 等待 RXPND 变为“1”，或等待中断
- (9) 从 HSUT0FIFO 读取数据
- (10) 清除 PNDing 为下一个 RX 数据做准备

注意：如果未及时读取 RX 数据，则数据将被下一个 RX 数据覆盖

12.2.5 RX n byte 带有 DMA &禁用循环缓冲区

- (1) 按正确的方向配置 GPIO
- (2) 配置 HSUT0CON CLKSRC 位以实现高速接口工作源时钟
- (3) 配置 HSUT0CON DMA 模式,8bit 模式，非循环缓冲模式
- (4) 配置 HSUT0BAUD 波特率
- (5) 配置 RX DMA 起始地址 HSUT0RXDADR

- (6) 设置使能 URXEN
- (7) 设置 UPRXCFG 来同步设置到接口时钟域
- (8) 根据需要配置 RXIET01'
- (9) 写入计数数据到 HSUT0RXCNT 来启动 RX
- (10) 等待 RXPND 变为“1”，或等待中断
- (11) 从 SRAM 或者 HSUT0RXFIFO 读取 RX 数据
- (12) 清除 PNDing, 同时清除 HSUT0FIFOCNT

12.2.6 RX n byte 带有 DMA & 循环缓冲器使能

- (1) 按正确的方向配置 GPIO
- (2) 配置 HSUT0CON CLKSRC 位以实现高速接口工作源时钟
- (3) 配置 HSUT0CON DMA 模式, 8bit 模式, 循环缓冲模式
- (4) 配置 HSUT0BAUD 波特率
- (5) 配置 RX DMA 上限地址 HSUT0RXUADR
- (6) 配置 RX DMA 下限地址 HSUT0RXDADR, 更新 DMA RX 起始地址
- (7) 将数据 RX 计数和 FIFO 溢出计数写入 HSUT0RXCNT
- (8) 设置使能 URXEN
- (9) 设置 UPRXCFG 来同步设置到接口时钟域
- (10) 写入计数数据到 HSUT0RXCNT 来启动 RX
- (11) 根据需要配置 RXIET01'
- (12) 等待 RXPND 更改为“1”，或等待中断，或时间检查 HSUT0FIFOCNT
- (13) 从 SRAM 或 HSUT0RXFIFO 读取 RX 数据
- (14) 清除 PNDing, 不会同时清除 HSUT0FIFOCNT
- (15) 写入数据计数到 HSUT0RXCNT, 按时等待下一个 RXDONE

从 SRAM 读取:

- (1) 从 HSUT0FIFOADR 或程序保存变量中获取起始地址
- (2) 从 HSUT0FIFOCNT 或程序保存变量中获取数据计数
- (3) 从 SRAM 读取 N byte 数据
- (4) 如果不是 DMA 循环缓冲模式, 清除 RXPND 将会清除 HSUT0FIFOCNT
- (5) 如果是 DMA 循环缓冲模式, 写入 N 到 SUBRXCNT (HSUT0CPND[16:0]) 给十进制的 HSUT0FIFOCNT.

从 HSUT0FIFO 读取

- (1) 从 HSUT0FIFOCNT 获取数据计数
- (2) 将 RXFIFO[8] 设置为 1, 并将自动清除 RXFIFO[9]

- (3) 等待 RXFIFO[9] 变为 1
- (4) 从 RXFIFO[7:0] 获取数据
- (5) 从步骤 2 到步骤 4 循环 n 次

12.2.7 DMA RX TIMER MODE:

DMA RX HSUT0TMR 支持所有 DMA RX 模式。

HSUT0TMR 计数器启用:

- 设置功能使能位
- 写入数据计数到 HSUT0XCNT 将预启用
- 等待 HSUT0FIFOCNT != 0 并且计数器将启用

HSUT0TMR 计数器增加:

- 当 RX BUS IDLE 为 1 bit 波特率时间时, TMRCNT 增加 1

HSUT0TMR 清除:

- 通过 RX BUS 下降沿清除 TMRCNT
- 通过 DMA RX 启动清除 TMRCNT
- 通过清除 HSTMROV_PND 清除 TMRCNT

如果 HSTMROV_PND 置位 1, HSUT0TMR 将禁用直到下一次 DMA RX 启动。

12.2.8 应用说明:

12.2.8.1 系统域 到 uart 时钟域的同步配置:

以下的模式必须同步配置:

更改 SPBITSEL_NS (TX 停止位), TXBITSEL_NS (TX 数据位) 后, 需同步设置 UPTXCFG (置 1)

更改 RXBITSEL_NS (RX 数据位) 后, 需同步设置 UPRXCFG (置 1)

更改 HSUT0BAUD (波特率寄存器) 后, 需同步设置 UPTXCFG (置 1) 或 UPRXCFG (置 1)

12.2.8.2 DMA 失败标志:

如果 RX DMA 到 SRAM 长时间被其他 DMA 通道阻塞, 就会发生 DMA 错误。下一个 Rx 数据将覆盖当前数据。

12.2.8.3 FLAG 置位和清除

表格 8 TXPND (TX 挂起)

| 标志位 | 读到的值 | Buffer 模式 | DMA 模式 | 所有模式 |
|---------------|------|---------------|---------------|------------------------|
| TXPND (TX 挂起) | 0 | 正在写 HSUT0DATA | 正在写 HSUT0XCNT | 写完后 HSUT0CPND[13]自动变 1 |
| | 1 | 发送 1 字节完成 | 发送 n 字节完成 | ----- |

图 6 DMA 高速串口发送函数

```

} else if (rec tra sel == HSUT TRANSMIT) {
    while(hsuart_get_flag(HSUART_FLAG_TX) == RESET);
    hsuart_clear_flag(HSUART_FLAG_TX);
    memcpy(SEND_BUF, (u8*)addr, len);
    HSUART->tx_cnt = 0;
    HSUART->tx_adr = SEND_BUF;
    HSUART->tx_cnt = (uint32_t)len;
}
    
```

表格 9 RXPND (RX 挂起)

| 标志位 | 读到的值 | Buffer 模式 | DMA 模式 | 备注 |
|-----|------|-----------|--------|----|
|-----|------|-----------|--------|----|

| | | | | |
|---------------|-------|-----------|---------------------------|---------------------------|
| RXPND (RX 挂起) | ----- | ----- | ----- | ----- |
| | 1 | 接收 1 字节完成 | 接收 n 字节完成并写 n 字节到 SARM 完成 | 使能 HSUT0TMRcnt 和设置 TMROV。 |

表格 10 RXOV_set

| 标志位 | 写值 | 备注 |
|----------|-------|-----------------------------------|
| RXOV_set | ----- | ----- |
| | 0 | 写 USHT0CPND[12] 或者 HSUT0RXCNT 寄存器 |

表格 11 RXOVPND (RX 溢出标志位)

| 标志位 | 读到的值 | 备注 |
|--------------------|------|---|
| RXOVPND (RX 溢出标志位) | 1 | 设置 DMA loopbuffer 模式、RXFIFOCNT 最大限度、写数据到 SRAM |
| | 0 | 失能 URXEN、失能 DMA 模式、失能 LOOPBUFFER 模式、写 1 给 HSUT0CPND[11] |

表格 12 RXFAIL

| 标志位 | 读到的值 | 备注 |
|--------|------|--|
| RXFAIL | 1 | DMA 正在向 SRAM 写入数据，但下一个数据已输入以更新 DMA 数据 |
| | 0 | 失能 URXEN、失能 DMA 模式、写 1 给 HSUT0CPND[14] |

表格 13 TMR_OV

| 标志位 | 读到的值 | 备注 |
|--------|------|--|
| TMR_OV | 1 | TMRcnt 检查 UTRX BUS IDLE 的时间大于设定值 |
| | 0 | 写 1 给 HSUT0CPND[15] RXDMAEN 或者写 HSUT0RXCNT |

12.2.8.4 HSUT0DADR 的写入

写入 HSUT0DASR，也会更新这些内部地址：

FIFO 读取的起始地址: fiford_adr

RXDMA 起始地址: dmarx_wadr

12.2.8.5 RXOVPND 进程

这个进程设置 DMA loopbuffer 模式

在 RXOVPND 为 1 时，如下就是相关的程序步骤：

- (1) 如果发生数据溢出，RXPND 和 RXOVPND 将同时置位。
- (2) 首先进入 RXPND 进程，获取 HSUT0FIFOCNT，并从 RXFIFO 读取数据。
- (3) 进入 RXOVPND，获取 HSUT0FIFOCNT，并从 RXFIFO 读取数据。
 - HSUT0FIFOCNT != 0 接收到 RXCNT 数据后 RXPND 会置一
 - HSUT0FIFOCNT = 0 如果 RXOVPND 溢出寄存器变为 1，RXPND 挂起也会被置一
- (4) 清除 RXOVPND
 - 通过 HSUT0CPND[11]来清除 RXFIFOCNT、RXFIFO_ADR 初始值、RXWR_ADR 现在值。

12.3 HSUART 0 特殊功能寄存器

寄存器 12-1 HSUT0CON: 高速 UART 控制寄存器

| 位 | 名称 | 模式 | 定义 | 描述 |
|-------|----|----|----|----|
| 31:18 | - | - | - | 闲置 |

| 位 | 名称 | 模式 | 定义 | 描述 |
|----|--------------|----|----|--|
| 17 | UPTXCFG | W | 0 | HS TX 接口设置更新 0: N/A 1: 将配置从 sysclk 更新到 hsuart clk 域 |
| 16 | UPRXCFG | W | 0 | HS RX 接口设置更新 0: N/A 1: 将配置从 Sysclk 更新到 HSUARTCLK 域 |
| 15 | TMROV | R | 0 | RX TIMER 溢出标志 0: RX timer 未溢出 1: RX timer 溢出 |
| 14 | RXFAIL | R | 0 | RX 失败标志 0: RX DMA 没有错误 1: RX DMA 错误 |
| 13 | TXPND | R | 1 | TX pending 标志位 0: TX one byte/DMA n byte 没有结束 1: TX one byte/DMA n byte 结束 |
| 12 | RXPND | R | 0 | RX pending 标志位 0: RX one byte/DMA n byte 没有结束 1: RX one byte/DMA n byte 结束 |
| 11 | RXOVPND | R | 0 | RX 溢出标志位 0: RX DMA buffer 没有溢出 1: RX DMA buffer 溢出 |
| 10 | HSUTTMREN_NS | WR | 0 | HSUART DMA RX TIMER CNT 使能 0: 禁止 1: 使能 |
| 9 | SPBITSEL_NS* | WR | 0 | TX 停止位选择 0: 1 bit 停止位 1: 2 bit 停止位 |
| 8 | TXBITSEL_NS* | WR | 0 | TX 数据位选择 0: 8-bit 模式 1: 9-bit 模式 |
| 7 | TXTRSMODE | WR | 0 | TX 传输模式选择 0: 缓冲模式 1: DMA 模式 |
| 6 | RXLPBUFEN | WR | 0 | RX DMA 循环缓冲模式使能位 0: 禁止 1: 使能 |
| 5 | RXBITSEL_NS* | WR | 0 | RX 数据位选择 0: 8-bit 模式 1: 9-bit 模式 |
| 4 | RXTRSMODE | WR | 0 | RX 传输模式选择 0: 缓冲模式 1: DMA 模式 |
| 3 | TXIE | WR | 0 | 发送中断使能 0 = 发送中断禁用 1 = 发送中断使能 |
| 2 | RXIE | WR | 0 | 接收中断使能 0: 接收中断禁用 1: 接收中断使能 |
| 1 | UTXEN_NS | WR | 0 | UART TX 使能位 0: 禁用 TX UART 模块 1: 启用 TX UART 模块 |
| 0 | URXEN_NS | WR | 0 | UART RX 使能位 0: 禁用 UART RX 模块 1: 启用 UART RX 模块 |

*_NS 需要通过 UPTXCFG 或 UPRXCFG 同步到 UART 接口时钟域

寄存器 12-2 HSUT0CPND: HSUART 清除标志位寄存器

| 位 | 名称 | 模式 | 定义 | 描述 |
|-------|-----------|----|----|---|
| 31:16 | SUBRXCNT* | W | 0 | 在 RX DMA 模式下降低 RXFIFOCNT |
| 15 | CTMROV | W | 0 | RX timer 溢出标志清除 0: N/A 1: 清除标志 |
| 14 | CRXFAIL | W | 0 | RX 失败清除 0: N/A 1: 清除 RX 失败标志 |
| 13 | CTXPND | W | 0 | TX 标志位清除 0: N/A 1: 清除 TX 标志位 |
| 12 | CRXPND | W | 0 | RX 标志位清除 0: N/A 1: 清除 RX 标志位. 将数据写入 UTBUF 将清除 TXPND |
| 11 | CRXOVND | W | 0 | RX 溢出标志位清除 0: N/A 1: 清除 RX 溢出标志位 |
| 10:2 | - | - | - | 闲置 |
| 1 | CUTTX | W | 0 | HS TX 接口清除 0: N/A 1: 在 HSUARTCLK 域中清除到空闲状态 |
| 0 | CUTRX | W | 0 | HS RX 接口清除 0: N/A 1: 在 HSUARTCLK 域中清除到空闲状态 |

*SUBRXCNT: 子计数器在 1 时钟后写入此寄存器

寄存器 12-3 HSUT0BAUD: HSUART 波特率寄存器

| 位 | 名称 | 模式 | 定义 | 描述 |
|-------|------------|----|----|--|
| 31:16 | HSUTRXBAUD | W | 0 | HSUART RX 波特率 波特率=source clock / (HSUTRXBAUD + 1) |
| 15:0 | HSUTTXBAUD | W | 0 | HSUART TX 波特率 波特率=source clock / (HSUTTXBAUD + 1) |

寄存器 12-4 HSUT0DATA: HSUART 数据寄存器

| 位 | 名称 | 模式 | 定义 | 描述 |
|------|----------|----|----|---------------------------------|
| 31:9 | - | - | - | 闲置 |
| 8 | HSUTBIT8 | W | X | UART 数据位 8 |
| 7:0 | HSUTDAT | W | X | UART 数据 写入此寄存器会将数据加载到发送器缓冲区。 |

寄存器 12-5 HSUT0TXCNT: HSUART TX 计数器寄存器

| 位 | 名称 | 模式 | 定义 | 描述 |
|-------|-------|----|----|---------------------|
| 31:16 | - | - | - | 闲置 |
| 15:0 | TXCNT | WR | 0 | HSUART TX DMA 字节计数器 |

寄存器 12-6 HSUT0TXADR: HSUART TX DMA 起始地址寄存器

| 位 | 名称 | 模式 | 定义 | 描述 |
|------|-----------|----|----|--------------------|
| 31:0 | HSUTTXADR | WR | 0 | HSUART TX DMA 起始地址 |

寄存器 12-7 HSUT0RXCNT: HSUART RX 计数器寄存器

| 位 | 名称 | 模式 | 定义 | 描述 |
|-------|-------|----|----|---------------------|
| 31:16 | - | - | - | 闲置 |
| 15:0 | RXCNT | WR | 0 | HSUART RX DMA 字节计数器 |

寄存器 12-8 HSUT0FIFOCNT: HSUART RX FIFO 计数器寄存器

| 位 | 名称 | 模式 | 定义 | 描述 |
|-------|-----------|----|----|--------------------|
| 31:16 | - | - | - | 闲置 |
| 15:0 | RXFIFOCNT | R | 0 | HSUART RX DMA byte |

寄存器 12-9 HSUT0RXUADR: HSUART RX 上行地址寄存器

| 位 | 名称 | 模式 | 定义 | 描述 |
|------|--------|----|----|-----------------------|
| 31:0 | RXUADR | WR | 0 | HSUART RX DMA 缓冲区上限地址 |

寄存器 12-10 HSUT0RXDADR: HSUART RX 下限地址寄存器

| 位 | 名称 | 模式 | 定义 | 描述 |
|------|--------|----|----|-----------------------|
| 31:0 | RXDADR | WR | 0 | HSUART RX DMA 缓冲区下限地址 |

寄存器 12-11 HSUT0RXMCNT: HSUART RX 最大计数寄存器

| 位 | 名称 | 模式 | 定义 | 描述 |
|-------|----------|----|----|-----------------------|
| 31:16 | - | - | - | 闲置 |
| 15:0 | RXMAXCNT | WR | 0 | HSUART RX DMA 字节最大计数器 |

寄存器 12-12 HSUT0FIFOADR: HSUART RX FIFO point 地址寄存器

| 位 | 名称 | 模式 | 定义 | 描述 |
|------|-----------|----|----|-----------------------|
| 31:0 | RXFIFOADR | R | 0 | HSUART RX DMA 缓冲区下限地址 |

寄存器 12-13 HSUT0RXFIFO: HSUART RX Data FIFO 寄存器

| 位 | 名称 | 模式 | 定义 | 描述 |
|------|-------------|----|----|---|
| 31:8 | - | - | - | 闲置 |
| 9 | RXFIFO_DONE | R | 0 | RXFIFO 从 SRAM 读取完成 0: 未完成读取 1: 读取完成 通过启动 RXFIFO_RD 位自动清除 |
| 8 | RXFIFO_RD | W | 0 | RX 读取设置 0: N/A 1: fifo 读取启动 |
| 7:0 | RXFIFO | R | x | UART RX 数据 读取此寄存器将从接收器 SRAM 缓冲器读取数据 |

寄存器 12-14: HSUT0FIFOADR HSUART RX Data FIFO 地址寄存器

| 位 | 名称 | 模式 | 定义 | 描述 |
|------|-------------|----|----|-----------|
| 31:0 | HSUTFIFOADR | R | - | FIFO 读取地址 |

寄存器 12-15: HSUT0TMR HSUART RX TIMER 寄存器

| 位 | 名称 | 模式 | 定义 | 描述 |
|-------|------------|----|----|-------------------------|
| 31:16 | - | - | - | 闲置 |
| 15:0 | HSUTTMR_NS | WR | 0 | HSUART DMA RX TIMER CNT |

13 SPI

13.1 特征

- (1) 此芯片有 2 路 SPI，其中 SPI0 直接连芯片内部 flash,用户无法外部使用;SP1 通过绑线接到外部 IO 口，用户可用。
- (2) 闪存 SPI 是系统时钟的 2 分频。SPI 闪存起始地址为 0x1000 0000。
- (3) SPI 可以支持不同的模式：
 - 通用 3 线模式，1 位时钟输入/输出，1 位数据输出，1 位数据输入
 - 2 线模式，1 位时钟输入/输出，1 位数据输出或输入（硬件上 Di 和 Do 间串 240Ω 电阻）
 - 2 种数据总线模式，1 位时钟输入/输出，2 位数据输出或输入
 - 4 种数据总线模式，1 位时钟输入/输出，4 位数据输出或输入

13.2 功能配置

SPI 正常 1 位模式操作流程

- (1) 设置 3 线模式或 2 线模式并选择引脚图
- (2) 选择 RXSEL 进行发送或接收
- (3) 配置时钟频率
- (4) 选择四种定时模式之一
- (5) 通过设置 SPIEN '1'启用 SPI 模块
- (6) 如果需要，设置 SPIIE '1'
- (7) 将数据写入 SPIBUF 以启动该过程
- (8) 等待 SPIPND 更改为“1”，或等待中断
- (9) 如果需要，从 SPIBUF 读取接收到的数据
- (10) 如果需要，请转到步骤 8 以启动另一个进程，或通过清除 SPIIE 和 SPIEN 关闭 SPI0

SPI 正常多位模式操作流程

- (1) 设置数据总线宽度（总线 4 或总线 2）并选择引脚映射
- (2) 选择 RXSEL 进行发送或接收
- (3) 配置时钟频率
- (4) 选择四种定时模式之一
- (5) 通过设置 SPIEN '1'启用 SPI 模块
- (6) 如果需要，设置 SPIIE '1'
- (7) 将数据写入 SPIBUF 以启动该过程
- (8) 如果数据总线宽度为 2 位，则写入 SPIBUF 两次启动传输
- (9) 如果数据总线宽度为 4 位，则写入 SPIBUF 四次启动传输

- (10) 但是，在接收数据时，只需要写入一次即可启动接收过程
- (11) 等待 SPIPND 更改为“1”，或等待中断
- (12) 如果需要，从 SPIBUF 读取接收到的数据
- (13) 如果需要，请转到步骤 8 以启动另一个进程，或通过清除 SPIIE 和 SPIEN 来关闭 SPI

SPI DMA 模式操作流程

- (1) 将 IO 设置为正确的方向和数据宽度模式。
- (2) 为 DMA 方向选择 TXSEL
- (3) 配置时钟频率
- (4) 选择四种定时模式之一
- (5) 通过将 SPIEN 设置为“1”来启用 SPI 模块
- (6) 如果需要，设置 SPIIE '1'
- (7) 配置 SPIDMAADR;
- (8) 将数据写入 SPI_DMACNT 以启动 DMA 进程
- (9) 等待 SPIPND 更改为“1”，或等待中断
- (10) 如果需要，请转到步骤 8 以启动另一个 DMA 进程，或通过清除 SPIEN 关闭 SPI

13.3 SPI 特殊功能寄存器

寄存器 13-1 SPIxCON: SPI 控制寄存器

| 位 | 名称 | 模式 | 定义 | 描述 |
|-------|----------|----|-----|---|
| 31:17 | - | - | - | 闲置 |
| 16 | SPIPND | R | 0 | SPI 挂起 0: 未完成 SPI 接收/发射 1: 完成 SPI 接收/发射 |
| 15:14 | - | - | - | Unused |
| 13 | HOLDENSW | WR | 0 | SPI 软件保持使能 0: 禁用 1: 启用 |
| 12 | HOLDENTX | WR | 0 | BT tx 时 SPI 保持使能 0: 禁用 1: 启用 |
| 11 | HOLDENRX | WR | 0 | BT rx 时 SPI 保持使能 0: 禁用 1: 启用 |
| 10 | SPIOSS | WR | 0 | SPI 采样数据与输出数据处于同一时钟边沿 0: SPI 采样数据与输出数据处于差分时钟边沿 1: SPI 采样数据与输出数据处于同一时钟边沿 |
| 9 | SPIBEN | WR | 0 | SPI 多位总线使能位 0: 禁用 1: 启用 |
| 8 | SPILF_EN | WR | 0 | SPI LFSR 使能位 0: 禁用 1: 启用 |
| 7 | SPIIE | WR | 0 | SPI 中断使能 0: 禁用 1: 启用 |
| 6 | SMPS | WR | 0 | SPI 输出边沿选择位, 当 SPIOSS = 0 时, 采样数据和输出数据处于不同的时钟边沿; 当 SPIOSS = 1 时, 采样数据和输出数据处于同一时钟边沿。 0: 输出数据处于下降沿; 1: 输出数据处于上升沿; |
| 5 | CLKIDS | WR | 0 | 空闲时的 SPI 时钟状态 0: 时钟保持在 0 1: 时钟保持在 1 |
| 4 | RXSEL | WR | 0 | 当处于 DMA 模式或 2 线模式时, 配置 SPI 接收或发送选择位 0: 发送 1: 接收 |
| 3:2 | BUSMODE | WR | 0x0 | 数据总线宽度选择位 00: 3 线模式; 1 位数据输入, 1 位数据输出 01: 2 线模式; 1 位数据输入/输出 10: 2 位双向数据总线 11: 4 位双向数据总线 |
| 1 | SPI SM | WR | 0 | 从模式选择位 0: 主模式 1: 从模式 |
| 0 | SPIEN | WR | 0 | SPI 使能位 |

| 位 | 名称 | 模式 | 定义 | 描述 |
|---|----|----|----|----------------|
| | | | | 0: 禁用 1: 启用 |

寄存器 13-2 SPIxBAUD: SPI 波特率寄存器

| 位 | 名称 | 模式 | 定义 | 描述 |
|-------|----------|----|----|--|
| 31:16 | - | - | - | 闲置 |
| 15:0 | SPIxBAUD | W | 0 | SPI 波特率 波特率 = Fsys clock / (SPI_BAUD+1) |

寄存器 13-3 SPIxCPND: SPI 清除挂起寄存器

| 位 | 名称 | 模式 | 定义 | 描述 |
|-------|---------|----|----|------------------|
| 31:17 | - | - | - | 闲置 |
| 16 | SPICPND | W | 0 | 写入 1 将会清除 SPI 挂起 |
| 15:0 | - | - | - | 闲置 |

寄存器 13-4 SPIxBUF: SPIx 接收/发送数据寄存器

| 位 | 名称 | 模式 | 定义 | 描述 |
|------|---------|----|----|--|
| 31:8 | - | - | - | 闲置 |
| 7:0 | SPIxBUF | WR | x | SPI 数据 写入此寄存器会将数据加载到发送缓冲器。 读取此寄存器将从接收器缓冲器读取数据。 |

寄存器 13-5 SPIxDMACNT: SPIx DMA 计数寄存器

| 位 | 名称 | 模式 | 定义 | 描述 |
|-------|------------|----|----|---|
| 31:11 | - | - | - | 闲置 |
| 10:0 | SPIxDMACNT | W | x | SPIDMA 字节计数器 写入此寄存器将启动 spi 发送/接收数据 接收/发送的总字节数为 SPIxDMACNT |

寄存器 13-6 SPIxDMAADR: SPIx DMA 地址寄存器

| 位 | 名称 | 模式 | 定义 | 描述 |
|-------|------------|----|----|--------------|
| 31:21 | - | - | - | 闲置 |
| 20:0 | SPIxDMAADR | W | x | SPI DMA 字节地址 |

14 IIC

14.1 特征

- (1) 支持 IIC 一个主机
- (2) 支持来自 RC2M 或 XOSC-的异步时钟源
- (3) 支持输出数据最大 4 字节,不支持 DMA, 单次最多支持 4 byte 数据收发
- (4) 支持输入数据最大 4 字节
- (5) 支持 IIC 完成中断

14.2 功能配置

IIC 主模式操作流程:

- (1) 配置 IO 映射, SDA 设置上拉使能
- (2) 配置 IIC 时钟, 从 RC 2M 或 XOSC 中选择, 设置 pre_div 时钟
- (3) 配置 IIC IICCON0
- (4) 为控制字节和地址字节配置 ICICCMDA
- (5) 为写入数据配置 IICDATA
- (6) 配置 IICCON1
- (7) 启动
- (8) 等待完成标志或中断 (如果需要)
- (9) 清除 DMA DONE 标志并更新 IICCMDA 或 IICDATA
- (10) 循环步骤 7

14.3 IIC 特殊功能寄存器

寄存器 14-1 IICON0: IIC 控制寄存器 0

| 位 | 名称 | 模式 | 定义 | 描述 |
|-------|-----------|----|----|---|
| 31 | DONE | R | 0 | IIC DONE 标志 |
| 30 | ACKSTATUS | R | 0 | RX IIC 从机 ACK 状态 0: RX ACK 1: RX NAK |
| 29 | CLR_DONE | W | 0 | DONE 标志清除 0: 1: 清除 |
| 28 | KS | W | 0 | 启动 0: 1:启动 |
| 27 | CLR_ALL | W | 0 | 清除所有状态 0: 1: 清除 |
| 26:10 | Rev. | WR | - | 闲置 |
| 9:4 | POSDIV | WR | 0 | IIC SCL 产生分频计数器 0: div 1 1: div2 ... N: div N+1 |
| 3:2 | HOLDCNT | WR | 0 | SDA 在 SCL 下降时保持 cnt 0: 1 cycle 1: 2 cycle ... |
| 1 | INTEN | WR | 0 | IIC 中断 0: 禁止 1: 使能 |
| 0 | IIC_EN | WR | 0 | IIC 使能位 0: 禁止 1: 使能 |

寄存器 14-2 IICON1: IIC 控制寄存器 1

| 位 | 名称 | 模式 | 定义 | 描述 |
|-------|----------|----|----|--------------------------|
| 31:13 | - | - | - | 闲置 |
| 12 | TXNAK_EN | WR | 0 | IIC TX NAK, 当读取最后一个数据启用时 |
| 11 | STOP_EN | WR | 0 | IIC TX 停止使能 |
| 10 | WDAT_EN | WR | 0 | IIC TX 数据使能 |
| 9 | RDAT_EN | WR | 0 | IIC RX 数据使能 |
| 8 | CTL1_EN | WR | 0 | IIC TX ctl 1 使能 |

| | | | | |
|-----|-----------|----|---|---|
| 7 | START1_EN | WR | 0 | IIC TX start 1 使能 |
| 6 | ADR1_EN | WR | 0 | IIC TX adr 1 使能 |
| 5 | ADR0_EN | WR | 0 | IIC TX adr 0 使能 |
| 4 | CTL0_EN | WR | 0 | IIC TX ctl 0 使能 |
| 3 | START0_EN | WR | 0 | IIC TX start 0 使能 |
| 2:0 | DATA_CNT | WR | 0 | RX/TX 数据计数器 0: 0 byte 1: 1 byte ... N: N byte |

IIC 时钟配置

IICCLK = source clk / (preclkdiv+1)

SCL = IICCLK / (posdiv+1)

寄存器 14-3 IICMDA: IIC 指令/地址寄存器

| 位 | 名称 | 模式 | 定义 | 描述 |
|-------|------|----|----|---------|
| 31:24 | CTL1 | WR | 0 | 控制 1 数据 |
| 23:16 | ADR1 | WR | 0 | 地址 1 数据 |
| 15:8 | ADR0 | WR | 0 | 地址 0 数据 |
| 7:0 | CTL0 | WR | 0 | 控制 0 数据 |

寄存器 14-4 IICDATA: IIC 数据寄存器

| 位 | 名称 | 模式 | 定义 | 描述 |
|-------|-------|----|----|------|
| 31:24 | DATA3 | WR | 0 | 数据 3 |
| 23:16 | DATA2 | WR | 0 | 数据 2 |
| 15:8 | DATA1 | WR | 0 | 数据 1 |
| 7:0 | DATA0 | WR | 0 | 数据 0 |

15 ADC

15.1 特征

该 ADC 具备以下特征：

- (1) 支持 16 通道；
- (2) 最快 11us 完成 1 次 ADC 采样;ADC 位时钟值分别为 2MHZ 和 6MHZ;
- (3) ADC 具有内部 100K 上拉电阻。
- (4) ADC 根据基准电压 bandgao 计算时 ， 误差在 3mV 以内（10bit 分辨率，3300mV/1024=3.2mV）。

15.2 通道选择

代码可借鉴：

```

/*****
* Module      :ADC 通路选择列表
*****/
#define ADCCH_PA5      0          //SARADC channel 0
#define ADCCH_PA6      1          //SARADC channel 1
#define ADCCH_PA7      2          //SARADC channel 2
#define ADCCH_PB1      3          //SARADC channel 3          WK2
#define ADCCH_PB2      4          //SARADC channel 4          WK3
#define ADCCH_PB3      5          //SARADC channel 5
#define ADCCH_PB4      6          //SARADC channel 6
#define ADCCH_PE5      7          //SARADC channel 7
#define ADCCH_PE6      8          //SARADC channel 8
#define ADCCH_PE7      9          //SARADC channel 9          ADS7844 ADC0
#define ADCCH_PF5      10         //SARADC channel 10
#define ADCCH_PB0      11         //SARADC channel 11          WK1
#define ADCCH_WKO      12         //SARADC channel 12          WKO/PB5
#define ADCCH_BGOP     13         //SARADC channel 13,内部校准电压 bandgap 电压为 0.75V
#define ADCCH_VBAT     14         //SARADC channel 14
#define ADCCH_VUSB     15         //SARADC channel 15

```

15.3 ADC 时钟配置

ADC 时钟只有 2MHz 和 6MHz，波特率 = Fadc clock/（BAUD +1），其中 BAUD 可取 0,1,2,3.

图 7 ADC 时钟配置

```
//SARADC CLK enumeration
typedef enum {
    SADC_CLK_2M      = 0x00,
    SADC_CLK_6M      = 0x01,
} SADC_CLK_TYPEDEF;

//SARADC setup timing enumeration
typedef enum {
    SADC_ST_0_CLK    = 0x00,
    SADC_ST_2_CLK    = 0x01,
    SADC_ST_4_CLK    = 0x10,
    SADC_ST_8_CLK    = 0x11,
} SADC_ST_TYPEDEF;
```

15.4 功能配置

- (1) 配置 SADCBAUD
- (2) 根据需要配置 SADCST
- (3) 启用 SARADC
- (4) 编写 SADCCH 以启用要转换的通道。可以启用多个通道。写入 SADCCH 将启动 ADC 转换。
- (5) 等待 ADC_PND

15.5 SARADC_CTL 特殊功能寄存器

寄存器 15-1 SADCCON: SARADC 控制寄存器

| 位 | 名称 | 模式 | 定义 | 描述 |
|-------|----------|----|----|--|
| 31:20 | - | - | - | 闲置 |
| 19 | ADCAEN | WR | 0 | Saradc 自动使能模拟使能位 0: 禁用 1: 启用 |
| 18 | ADCANGIO | WR | 0 | Saradc 自动使能模拟 IO 使能位 0: 禁用 1: 启用 |
| 17 | ADCIE | WR | 0 | Saradc 中断使能位 0: 禁用 1: 启用 |
| 16 | ADCEN | WR | 0 | Saradc 使能位 0: 禁用 1: 启用 |
| 15 | CH15PUEN | WR | 0 | 通道 15 内部上拉使能位 0: 禁用 1: 启用 |
| 14 | CH14PUEN | WR | 0 | 通道 14 内部上拉使能位 0: 禁用 1: 启用 |
| 13 | CH13PUEN | WR | 0 | 通道 13 内部上拉使能位 0: 禁用 1: 启用 |
| 12 | CH12PUEN | WR | 0 | 通道 12 内部上拉使能位 0: 禁用 1: 启用 |
| 11 | CH11PUEN | WR | 0 | 通道 11 内部上拉使能位 0: 禁用 1: 启用 |
| 10 | CH10PUEN | WR | 0 | 通道 10 内部上拉使能位 0: 禁用 1: 启用 |
| 9 | CH9PUEN | WR | 0 | 通道 9 内部上拉使能位 0: 禁用 1: 启用 |
| 8 | CH8PUEN | WR | 0 | 通道 8 内部上拉使能位 0: 禁用 1: 启用 |
| 7 | CH7PUEN | WR | 0 | 通道 7 内部上拉使能位 0: 禁用 1: 启用 |
| 6 | CH6PUEN | WR | 0 | 通道 6 内部上拉使能位 0: 禁用 1: 启用 |
| 5 | CH5PUEN | WR | 0 | 通道 5 内部上拉使能位 0: 禁用 1: 启用 |
| 4 | CH4PUEN | WR | 0 | 通道 4 内部上拉使能位 0: 禁用 |

| 位 | 名称 | 模式 | 定义 | 描述 |
|---|---------|----|----|--------------------------------|
| | | | | 1: 启用 |
| 3 | CH3PUEN | WR | 0 | 通道 3 内部上拉使能位 0: 禁用 1: 启用 |
| 2 | CH2PUEN | WR | 0 | 通道 2 内部上拉使能位 0: 禁用 1: 启用 |
| 1 | CH1PUEN | WR | 0 | 通道 1 内部上拉使能位 0: 禁用 1: 启用 |
| 0 | CH0PUEN | WR | 0 | 通道 0 内部上拉使能位 0: 禁用 1: 启用 |

寄存器 15-2 SADCCH: SARADC 通道使能寄存器

| 位 | 名称 | 模式 | 定义 | 描述 |
|-------|--------|----|----|--|
| 31:17 | - | - | - | 闲置 |
| 16 | ADCPND | WR | 0 | Saradc 完成挂起 0:未完成 1:完成 写 SARADCH 寄存器将清除此位 |
| 15 | CH15EN | WR | 0 | 通道 15 使能位 0: 禁止 1: 使能 |
| 14 | CH14EN | WR | 0 | 通道 14 使能位 0: 禁止 1: 使能 |
| 13 | CH13EN | WR | 0 | 通道 13 使能位 0: 禁止 1: 使能 |
| 12 | CH12EN | WR | 0 | 通道 12 使能位 0: 禁止 1: 使能 |
| 11 | CH11EN | WR | 0 | 通道 11 使能位 0: 禁止 1: 使能 |
| 10 | CH10EN | WR | 0 | 通道 10 使能位 0: 禁止 1: 使能 |
| 9 | CH9EN | WR | 0 | 通道 9 使能位 0: 禁止 1: 使能 |
| 8 | CH8EN | WR | 0 | 通道 8 使能位 0: 禁止 1: 使能 |
| 7 | CH7EN | WR | 0 | 通道 7 使能位 0: 禁止 1: 使能 |
| 6 | CH6EN | WR | 0 | 通道 6 使能位 0: 禁止 |

| 位 | 名称 | 模式 | 定义 | 描述 |
|---|-------|----|----|----------------------------|
| | | | | 1: 使能 |
| 5 | CH5EN | WR | 0 | 通道 5 使能位 0: 禁止 1: 使能 |
| 4 | CH4EN | WR | 0 | 通道 4 使能位 0: 禁止 1: 使能 |
| 3 | CH3EN | WR | 0 | 通道 3 使能位 0: 禁止 1: 使能 |
| 2 | CH2EN | WR | 0 | 通道 2 使能位 0: 禁止 1: 使能 |
| 1 | CH1EN | WR | 0 | 通道 1 使能位 0: 禁止 1: 使能 |
| 0 | CH0EN | WR | 0 | 通道 0 使能位 0: 禁止 1: 使能 |

寄存器 15-3 SADCST: SAR ADC 设置定时寄存器

| 位 | 名称 | 模式 | 定义 | 描述 |
|-------|--------|----|-----|--|
| 31:30 | CH15ST | WO | 0x0 | 通道 15 设置时间 00:0 SARADC_CLK 01:2 SARADC_CLK 10: 4 SARADC_CLK 11: 8 SARADC_CLK |
| 29:28 | CH14ST | WO | 0x0 | 通道 14 设置时间 00:0 SARADC_CLK 01:2 SARADC_CLK 10: 4 SARADC_CLK 11: 8 SARADC_CLK |
| 27:26 | CH13ST | WO | 0x0 | 通道 13 设置时间 00:0 SARADC_CLK 01:2 SARADC_CLK 10: 4 SARADC_CLK 11: 8 SARADC_CLK |
| 25:24 | CH12ST | WO | 0x0 | 通道 12 设置时间 00:0 SARADC_CLK 01:2 SARADC_CLK 10: 4 SARADC_CLK 11: 8 SARADC_CLK |
| 23:22 | CH11ST | WO | 0x0 | 通道 11 设置时间 00:0 SARADC_CLK 01:2 SARADC_CLK 10: 4 SARADC_CLK 11: 8 SARADC_CLK |
| 21:20 | CH10ST | WO | 0x0 | 通道 10 设置时间 00:0 SARADC_CLK 01:2 SARADC_CLK 10: 4 SARADC_CLK 11: 8 SARADC_CLK |

| 位 | 名称 | 模式 | 定义 | 描述 |
|-------|-------|----|-----|---|
| 19:18 | CH9ST | WO | 0x0 | 通道 9 设置时间 00:0 SARADC_CLK 01:2 SARADC_CLK 10: 4 SARADC_CLK 11: 8 SARADC_CLK |
| 17:16 | CH8ST | WO | 0x0 | 通道 8 设置时间 00:0 SARADC_CLK 01:2 SARADC_CLK 10: 4 SARADC_CLK 11: 8 SARADC_CLK |
| 15:14 | CH7ST | WO | 0x0 | 通道 7 设置时间 00:0 SARADC_CLK 01:2 SARADC_CLK 10: 4 SARADC_CLK 11: 8 SARADC_CLK |
| 13:12 | CH6ST | WO | 0x0 | 通道 6 设置时间 00:0 SARADC_CLK 01:2 SARADC_CLK 10: 4 SARADC_CLK 11: 8 SARADC_CLK |
| 11:10 | CH5ST | WO | 0x0 | 通道 5 设置时间 00:0 SARADC_CLK 01:2 SARADC_CLK 10: 4 SARADC_CLK 11: 8 SARADC_CLK |
| 9:8 | CH4ST | WO | 0x0 | 通道 4 设置时间 00:0 SARADC_CLK 01:2 SARADC_CLK 10: 4 SARADC_CLK 11: 8 SARADC_CLK |
| 7:6 | CH3ST | WO | 0x0 | 通道 3 设置时间 00:0 SARADC_CLK 01:2 SARADC_CLK 10: 4 SARADC_CLK 11: 8 SARADC_CLK |
| 5:4 | CH2ST | WO | 0x0 | 通道 2 设置时间 00:0 SARADC_CLK 01:2 SARADC_CLK 10: 4 SARADC_CLK 11: 8 SARADC_CLK |
| 3:2 | CH1ST | WO | 0x0 | 通道 1 设置时间 00:0 SARADC_CLK 01:2 SARADC_CLK 10: 4 SARADC_CLK 11: 8 SARADC_CLK |
| 1:0 | CH0ST | WO | 0x0 | 通道 0 设置时间 00:0 SARADC_CLK 01:2 SARADC_CLK 10: 4 SARADC_CLK 11: 8 SARADC_CLK |

寄存器 15-4 SADCBAUD: SARADC 波特率寄存器

| 位 | 名称 | 模式 | 定义 | 描述 |
|-------|----|----|----|----|
| 31:10 | - | - | - | 闲置 |

| 位 | 名称 | 模式 | 定义 | 描述 |
|-----|----------|----|-----|---|
| 9:0 | SADCBAUD | WO | 0x0 | SARADC 波特率 波特率 = Fadc clock / [2(SADCBAUD+1)]. |

寄存器 15-5 SADCDAT0~15: SARADC 通道 0~15 数据寄存器

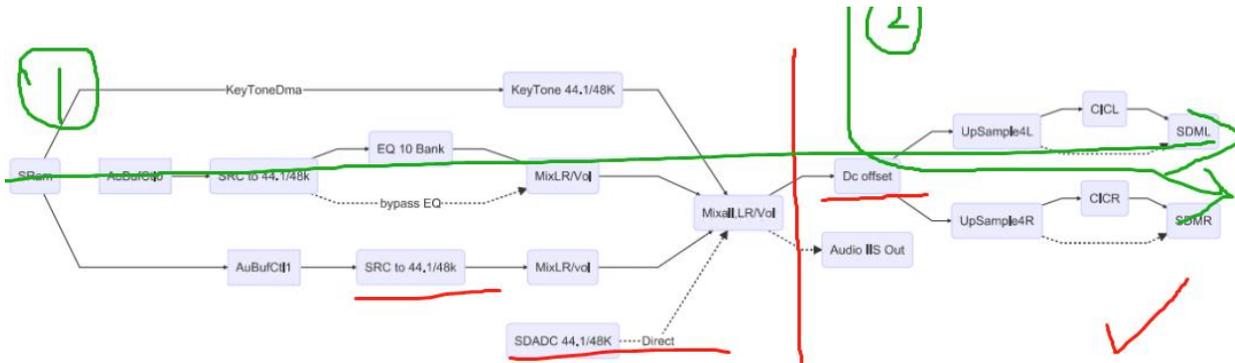
| 位 | 名称 | 模式 | 定义 | 描述 |
|-------|---------|----|-----|----------------------------|
| 31:10 | - | - | - | 闲置 |
| 9:0 | SADCDAT | R | 0x0 | SARADC 数据, 通道 0 至通道 15 寄存器 |

16 DAC

16.1 特征

- (1) 48KHz 采样率(DMA 推数据到 DAC 上), 16bit 采样精度, DR 98dB SNR;
- (2) 最快 780KHz (1.3us) 输出 DC 的偏移值, 以更改输出的 DAC 电平;

图 8 DAC 内部结构框图



16.2 控制使用

- (1) `my_dac_init()` 设置启动 DAC_R 或 DAC_L 引脚的输出 DAC 电平;
- (2) `set_bia_voltage_reg_R()` 设置 DAC_R 电平; `set_bia_voltage_reg_L()` 设置 DAC_L 电平;

17 USB

17.1 特征

USB OTG(USB 的 D+内部有上拉电阻,全速模式, 速度 12Mbps)

17.2 控制使用

usb_detect();//此函数会检测 USB 是否接上, 或者断开。每 5ms 被调用一次。

usb_insert_callback();//USB 插入后需调用。

usb_device_exit();//USB 被拔后需调用。

ude_rx_ack();//发送 ack 信号给主机, 要是不调用此函数, 默认回复 nack 给主机。

bulk_rx_handler();//USB 收到数据后, 进入此中断函数。

usb_bulk_send();//USB 发送函数。

18 电源管理

18.1 充电流程

当 vusb 引脚接入 5V 电压后，芯片先判断 vusb 接入的电压是否够高（vusb 电压要比电池电压高 0.2v 且 vusb 大于 4.5v，如果 vusb 电压满足充电条件，进入充电模式。

电池电压低于 2.2V 时的充电状态：

0V 充电功能(电池带保护板)，硬件固化，不能关闭。芯片判断锂电池是否需要激活（激活条件为芯片第一次上电，且电池电压低于 2.2v），如果满足激活条件，先以 120mA 电流激活电池，到电池电压高于 2.2v 后，用软件配置的充电电流进行充电。只有第一次上电时的激活电流不可配，上电后可以配置，如果芯片不再次完全掉电，以后充电时将以配置的激活电流进行电池激活。

电池电压高于 2.2V 时的充电状态：

在第一次上电电池电压高于 2.2v 后，芯片以涓流模式（第一次上电默认设置在 30mA，开机后可以修改）进行充电。如果不是第一次上电，且芯片没有完全掉电（vbat 没有低于 1.5v），那么芯片开始会以涓流模式进行充电。当电池电压高于 3v 后，芯片以恒流模式进行充电，涓流充电电流和恒流充电电流可以分别配置，通常涓流模式充电电流小于恒流模式充电电流。当电池节点接近 4.2v 时，进入恒压模式充电，充电电路最高输出电压等于拔掉电池后的输出电压，此电压被校准到 $4.2v \pm 42mV$ ，无论充多长时间，最终电池的充电电压也不会超过此电压。

恒压模式时，充电电流会逐渐减小，当充电电流小于设定的充电截止电流后，系统判断充电结束。另外，系统可以配置定时器，在电池电压达到 4.17v 后启动定时器，达到计时时间后，强制停止充电。

18.2 充电设置

恒流充电电流：16 个档位，10~200mA。

涓流充电电流：3 个档位，10~30mA。

关断充电电流：8 个档位，2.5~35mA。

关断充电电压：2 个档位，4.2V 或 4.3V。

涓流充电电压：2 个档位，2.9V 或 3.0V。

18.3 充电控制函数

`Bt_charge_off();`//停止充电

`Charge_start(u8 mode);`//启动充电，1 表示恒流，0 表示涓流

`sadc_get_data(ADCCH_VBAT);`//获取电池端电压采样值

`Pmu_init();`//1 表示进入 buck mode,低功耗模式。

19 蓝牙

19.1 特征

- (1) 核心规范: $\sqrt{\text{BR}}$ 、 $\sqrt{\text{VEDR}}$ 、 $\sqrt{\text{LE}}$
- (2) 功率等级: Class1(11dbm \geq Pmax \geq 4dbm)
- (3) BREDR 调制方式: $\sqrt{\text{GFSK}}$ 、 $\sqrt{\pi/4\text{-DQPSK}}$ 、 $\sqrt{8\text{DPSK}}$
- (4) BREDR 参数: Image frequency(+2Mhz)、Value n(3)
- (5) LE 调制方式: $\sqrt{\text{IM PHY}}$ 、 $\sqrt{2\text{M PHY}}$ 、 $\times\text{coded PHY}$ 、 $\times\text{Stable Modulation Index(TX \& RX)}$
- (6) LE 特性: $\times\text{AOA}$ 、 $\times\text{AOD}$ 、 $\times\text{CTE}$
- (7) LE 参数: Image frequency(+2Mhz)、Value n(3)

注: 蓝牙数据通讯可靠, 包含校验与加密。其中加密可选。

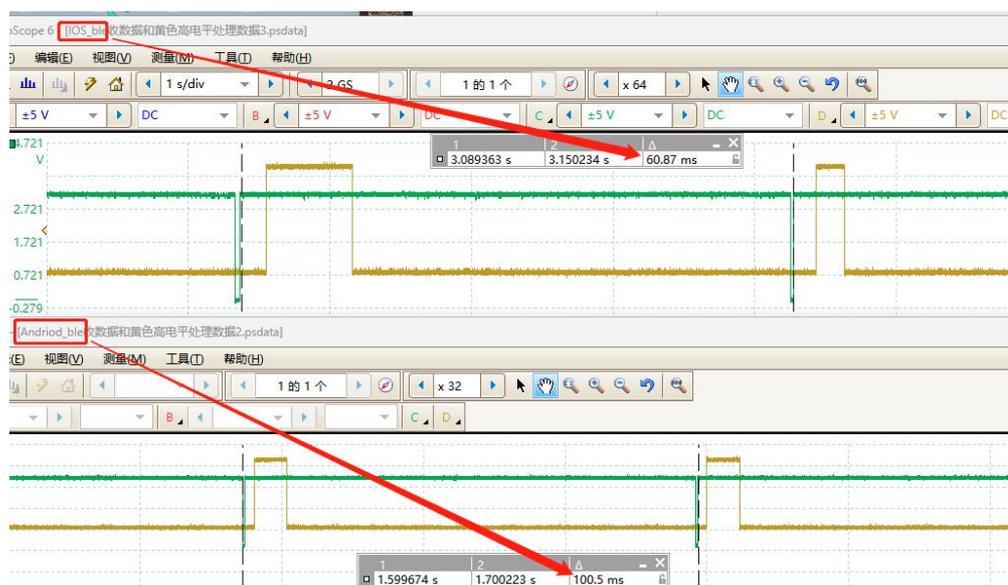
19.2 时序逻辑描述

蓝牙中断优先级最高, 详情见《表格 4 高优先级中断向量》。

当蓝牙中断函数完成后, 不是马上进入 rx_callback 线程, 而是收完或收到 MTU 个字节 (芯片和手机端两者中最小的 MTU 数值) 后才进入 rx_callback 线程。

下图是同一芯片在安卓手机上, 用 ble 收 509 数据后进入 rx_callback 回调函数, 每 100ms 进入一次; 在 ios 手机上, 用 ble 接收 509 字节, 每 60ms 进入一次。可见蓝牙的传输速度还与主机端的手机型号相关。

图 9



19.3 SPP 协议

19.3.1 SPP 协议基于 credit 的流量控制机制

SPP协议是基于蓝牙的RFCOMM协议的应用层协议，可以使用基于credit的流量控制机制。

基于credit的流量控制机制大致流程如下。

图 10



串口双方各自都会有两个credit的计数，分别是TX和RX的credit。在初始化的时候，串口双方会互相告知对方初始的RX的credit，获取到对方的RX credit就会将值记录到自己的TX credit中。

此后，每TX一个数据包，就会将TX credit减1；每RX一个数据包（即 `spp_rx_callback` 函数起来一次），就会将RX credit减1。当TX credit减到0时，就不能发送数据，需要等待对方补充credit或者断开连接；同样的，当RX credit减到0时，对方就会停止发送数据，需要等待自己补充credit或者断开连接。

在通信过程中，为了保证能够不断接收数据和发送数据，就需要不断的补充credit给对方，而补充多少credit就可以根据RX buffer的空间来计算和补充。这样就能实现一个根据RX buffer的空间来管理流量的一个控制机制。

19.3.2 SDK 中的 SPP 使用和开发

19.3.2.1 功能开启

在 config.h 文件中打开 BT_SPP_RX_FLOW_CTRL_EN 的宏。

19.3.2.2 API 的说明

开放了两个API用于通知SPP对方补充和更新credit，分别是 `void spp_set_rx_new_credit(uint8_t credit)` 和 `int spp_notify_rx_buffer_len(u16 len)`。

前者 `void spp_set_rx_new_credit(uint8_t credit)` 直接通知蓝牙底层去告知对方更新credit。

后者 `int spp_notify_rx_buffer_len(u16 len)`是将传入的buffer剩余可用的长度，除以SPP的MTU（最大传输单元，也就是一包数据最大的长度），得出剩余的buffer空间最少可以接收多少个数据包，然后再通知蓝牙底层去告知对方更新credit。

如果不知道如何更好的计算buffer剩余空间和credit的关系，建议直接调用 `int spp_notify_rx_buffer_len(u16 len)` 这个接口，使用MTU来计算可以保证buffer空间不会由于每次收到的数据包长度不同而导致溢出的问题。SPP协议的MTU大小由 `SPP_MTU_SIZE` 的宏控制，底层会通过 `u16 get_spp_mtu_size(void)` 函数获取MTU大小。

`spp_tx_pool`是给底层用的，用户不能操作此数据。

19.3.2.3 RX buffer 和 credit 的更新

请用户自定义SPP的RX buffer空间，并且做好buffer的管理，以便可以统计buffer的剩余可使用的空间，从而计算credit。

函数 `u16 spp_rxpkt_init(void)`，在这个函数里面可以做一些初始化RX buffer的相关内容，同时需要返回buffer的总长度，以便于建立SPP连接时可以设置初始的credit。

`void bsp_spp_process(void)` 函数，该函数处于主循环中不断地循环调用，可以在这里放rx的数据处理，处理完一次数据就调用 `spp_notify_rx_buffer_len`，更新buffer的剩余可用长度。不建议直接在回调函数 `void spp_rx_callback` 中直接进行数据处理，回调函数负担较大会影响蓝牙的运行。`spp_rx_callback()` 中一次最多接收511个字节，

启用spp流控时，要在 `spp_rx_callback()` 中补包给手机，即调用 `spp_set_rx_new_credit(1)`。不要手机继续发数据时，就不补包，记录所欠的包数，待需要手机继续发数据时，再统一把欠的包数补上。

`spp_notify_rx_buffer_len()` 返回值为0x75表示“没有开启流控配置”，需要打开 `BT_SPP_RX_FLOW_CTRL_EN`；返回值为0x76表示“RX credit error”。RX credit是蓝牙底层自动减的，返回0x76，就是传入的len参数，除以MTU得到的结果，小于当前的credit，此时不能增加credit。

19.4 BLE 协议

图 11

BLE 服务特性

| Description | UUID | Properties |
|-------------------------------|--------|-----------------------------|
| Service | 0xFF12 | |
| WriteWithNoRsp Characteristic | 0xFF15 | Read/Write Without Response |
| Notify Characteristic | 0xFF14 | Notify |

19.4.1 BLE 协议的流量控制机制

图 12

MTU 交换 是为了在主从双方设置一个PDU中最大能够交换的数据量，通过MTU的交换和双方确认（注意这个MTU是不可以协商的，只是通知对方，双方在知道对方的极限后会选择一个较小的值作为以后的MTU，比如说，主设备发出一个150个字节的MTU请求，但是从设备回应 MTU是23字节，那么今后双方要以较小的值23字节作为以后的MTU），主从双方约定每次在做数据传输时不超过这个最大数据单元。如下图的交换：

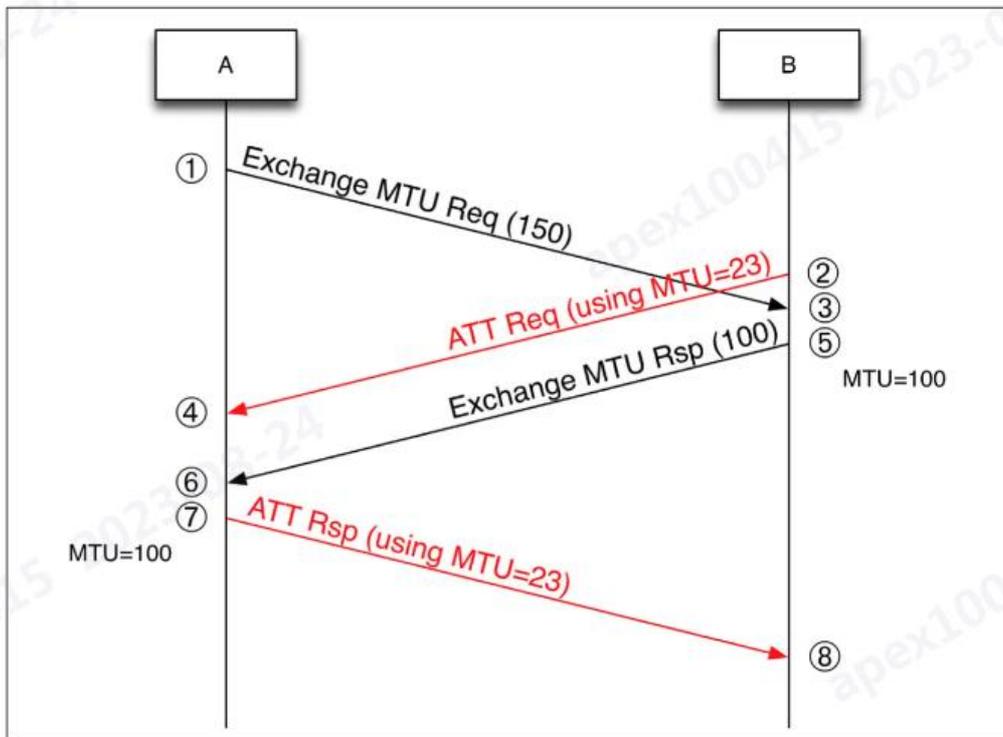


Figure 3.1: MTU Request and Response exchange

gatt 规定 ATT_MTU 默认支持不能小于 23。规范已经限制了 attribute value 只有 512 字节。

19.4.2 SDK 中 BLE 的使用和开发

19.4.2.1 ble 缩短每包间隔时间

`ble_update_conn_param(12, 0, 400);`//12*1.25ms 手机发包的间隔，第一个参数最小是 6,此值要 6 的倍数。

19.4.2.2 Ble 设置 read value 的值

```
void ble_init_att_for_handle(u16 handle, u8* buf, u16 len);
void ble_read_value(u8 *buf, u8 len)//ble 读厂商名，电量等蓝牙服务
{
    ble_init_att_for_handle(gatts_Datas_Characteristic_base.handle, buf, len);
}
```

19.4.2.3 蓝牙发送函数返回值

`ble_tx_notify` 返回 0 说明发送成功；发送不成功返回，返回“报错编码”，实际一个字节都没发出去。所以没发送成功，得重新发送整个数据。十进制 86 表示“packet 为空，或者发送长度大于 buf 大小”。

19.4.2.4 清除手机 app 端蓝牙缓存

`ble_service_change_indicate()`即：向"0x1801"服务的"0x2a05"发送 `ble_tx_indication(0x0100ffff)` 数据。

19.4.2.5 设置蓝牙地址

`ble_get_local_bd_addr ()`

19.5 FOTA 升级

19.5.1 特征 SDK 中 FOTA 的使用和开发

芯片支持使用 BLE 或 SPP 协议进行 FOTA 升级。

FOTA 采用的是双备份升级，因此需要保证 FLASH 的空间足够备份程序。

程序的大小必须小于 $(FLASH_SIZE/2 - 12)$ KB，FLASH_SIZE 请查看 `config.h` 文件中的宏 `FLASH_SIZE`。例如：FLASH 为 1M 的，那么程序的大小最多是 $(1024/2 - 12) = 500$ KB。客户需要根据实际情况设置 FLASH 的大小。

FOTA 有关的实现代码主要在 `bsp_fot.c`、`bsp_fot.h`、`spp.c` 和 `app.c` 文件中。注意 SPP、BLE 的回调函数以及 BLE 的注册服务中，有 FOTA 相关的代码，如需要使用 FOTA 功能，注意在开发过程不要修改此处 的内容。参考《FOT 协议文档》

19.5.2 客户自定义升级的使用和开发

(1) 识别到是升级指令时，要调用 `bsp_fot_init();fot_init();`并且把收到的数据存到 `fot_data` 中；

图 13 接收函数里识别升级数据后的代码样例 1

```

if( (buf[0]==0x46) && (buf[1]==0x4f) && (buf[2]==0x54) )
{
    //-----receive usb_update_file-----

    bsp_fot_init();fot_init();
    fot_file_sumlen = 8+((buf[7]<<24)|(buf[6]<<16)|(buf[5]<<8)|buf[4]);

    memcpy(fot_data+fot_perPage_len,buf, len);
    fot_file_reclen = len;
    fot_perPage_len = len;
    _usb_update_file();
}
//-----receive usb_update_file-----

```

(2) `fot_data`满512字节就可以通过将`fot_var.fot_rcv_ok`置1，使程序执行`bsp_fot.c`中`bsp_fot_process`的`fot_write`函数。（1字节也能调用，只是512字节调用效率更高）。

图 14 接收函数里识别升级数据后的代码样例 2

```

memcpy(fot_data+fot_perPage_len,buf, len);
fot_file_reclen += len;
fot_perPage_len += len;
if( (fot_perPage_len>=448) || (fot_file_reclen==fot_file_sumlen) )
{
    if(fot_file_reclen==fot_file_sumlen)
        fot_usb_flag = 0;
    fot_var.data_pos = 0;
    fot_var.total_len = fot_perPage_len;
    fot_var.fot_rcv_ok = 1;
}
else
{
    _usb_update_file();
}

```

图 15 升级函数里的代码样例

```
volatile uint8_t fot_perPage_len=0;
AT(.text.fot.cache)
void bsp_fot_process(void)
{
    if(fot_var.fot_recv_ok)
    {
        //      printf("--->%x %x %x %x\n",fot_data[0],fot_data[1],fot_get_curaddr()

        fot_write(fot_data, fot_get_curaddr(), fot_var.total_len);

        if(is_fot_update_success()){
            printf("--->fot update success\n");//FOT_DEBUG("--->fot update su
            fot_flag |= (FOT_FLAG_UPDATE_OK | FOT_FLAG_SYS_RESET);
            fot_dev_notify_sta(FOT_UPDATE_DONE);
            fot_var.tick = tick_get();
        }else{
            // fot_dev_notify_sta(fot_get_err());
            //      printf("--->err %d\n",fot_get_err());//
        }
        fot_perPage_len = 0;
        //      "    "    /Adjust_read_rec_buf_pt();
        fot_var.fot_recv_ok = 0;
    }

    if(fot_flag & FOT_FLAG_SYS_RESET)
    .
}
```

(3) 升级文件发送完后也将fot_var.fot_recv_ok置1，使程序执行bsp_fot.c中bsp_fot_process的fot_write函数。升级包每包fot_get_err()返回值为0（FOT_ERR_OK），且所有数据都写入，is_fot_update_success()就会返回1，从而执行WDT_RST();使芯片复位。

(4) 升级过程中的ble流控。

让手机端每509字节发一包回来，(MTU为512，但是ble收到的前3个字节是一个包头，所以gatt_callback_app实际收到的长度只有509)，然后GW3323执行完fot_write(fot_data, xxx, 509)后，就回复ack（可把TXBUF[x] = fot_get_err()）给手机。手机跟据ack判断是否升级成功，是否继续发送数据。

(5) 升级过程中的spp流控。

GW3323可以每进入一次spp_rx_callback就调用spp_set_rx_new_credit(1);只要fot_data数据不丢失即可。也可以等fot_write()后再调用spp_set_rx_new_credit(x);跟据用户自己习惯自己定义。

19.6 蓝牙地址

19.6.1 经典蓝牙（BR/EDR）

bt_get_local_bd_addr()可获取芯片的 mac 地址

正常情况下，该地址需要企业向 IEEE 申请，以保证地址的唯一性。

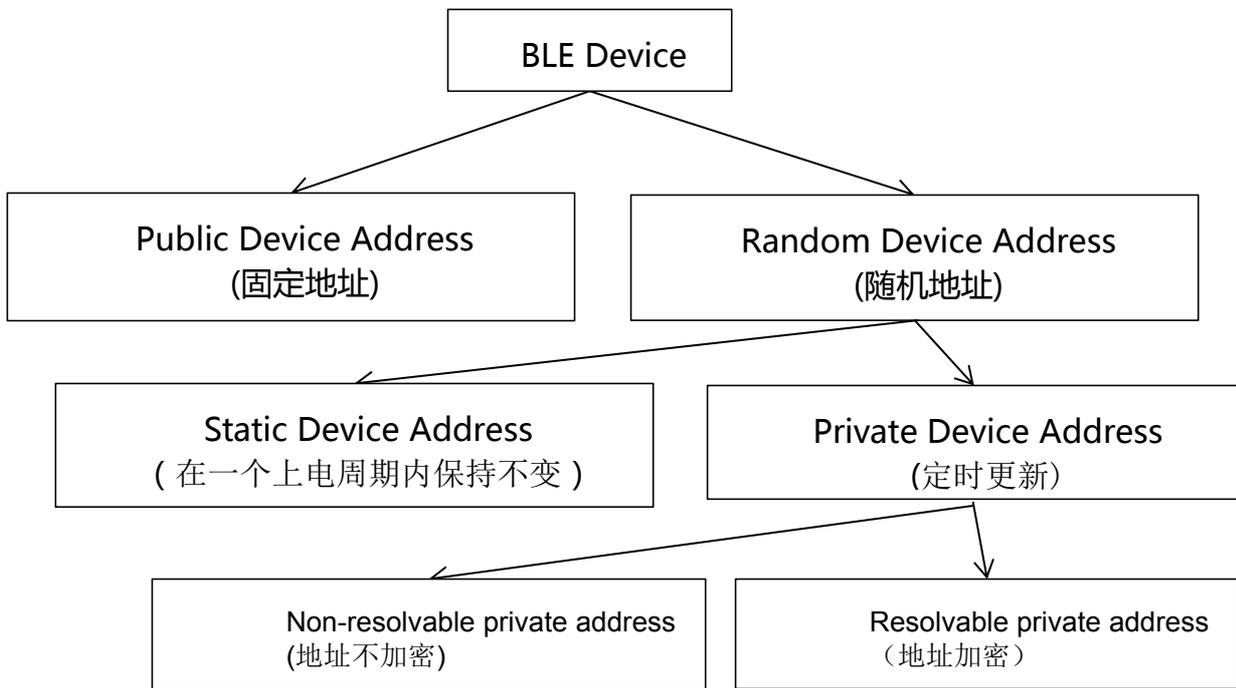
19.6.2 低功耗蓝牙（BLE）

BLE 有 4 个地址，其中 Public Device Address 需要向 IEEE 购买。

BLE 广播设备的地址（Public Device Address），若采用固定的设备地址，加大了信息泄漏的风

险；故可采用 **Random Device Address**，即设备地址不是固定分配的，而是在设备启动后随机生成的；下一次上电的时候可以改变；但不是强制的。

图 16 BLE 蓝牙设备地址分类



工程中调用以下函数设置BLE地址类型。

```
gap_random_address_set_mode();
```

19.7 蓝牙广播

蓝牙广播数据结构：字节长度+数据类型+数据内容

此款芯片广播数据见 `adv_data_const[]`数组。

20 烧录问题

20.1 特征

此芯片没有设断点的调试功能，只能通过“Downloader v2.7.2”的单线烧录，或者通过批量生产设备进行烧录。

此芯片通过 PB3 单线烧录，频率为 1.5MHz;

20.2 开发烧录

此芯片没有设断点的调试功能，只能通过“Downloader v2.7.2”的串口单线烧录。

Setting 中可以选择配置，这些配置会在 xcfg_cb 变量里。

详情见《GW3323 Printer 开发板 使用说明书》中 3.下载说明。

20.3 批量生产烧录问题

(1) GW3323 烧录器（一代）

支持 1 拖四，支持离线烧录，无显示屏；详情见《GW3323 烧录器说明 V0.2.pdf》。

(2) GW3323 烧录器（二代）

支持 1 拖四，支持离线烧录，有显示屏；详情见《GW3323 二代烧录器使用说明.pdf》

(3) GW3323 烧录盒

支持 1 拖 24，不支持离线烧录，必须接电脑主机，无显示屏；详情见《GW3323 烧录盒使用说明.pdf》。

20.4 芯片 UID

芯片唯一 uid 可以根据芯片内置 flash 的 uid 进行识别，相关函数是 spiflash_get_id(flash_id_t id)。

注意：要用 20230606 号及之后的库（api_sys.h,libplatform.a）

spiflash_get_id 得到 16 字节的数据，是 GW3323 内部 flash 芯片的 UID 号。例如：

GW3323HGU6:

22 32 66 23 a 0 13 26 3 43 52 31 3a 34 37 ..

GW3323HIU6:

42 50 30 56 39 38 31 17 0 92 47 26 4c 1 36 ..

42 50 30 56 39 38 31 17 0 22 49 26 4c 1 33 ..

20.5 安全

此款芯片任何方案商都不提供读 flash 接口，以确保芯片的保密性。此款芯片自己有双备份功能，所以出现烧录异常或者升级异常的可能性极小。

客户若出现极其特殊的异常芯片，可以发给我司，我们协助你们分析。

20.6 产线天线质量测试

未避免量产中元器件的电气偏差，PCBA 板的工艺偏差，造成蓝牙信号质量和发射功率的差异，可用测试设备进行抽检。

测试设备见《BLE 测试指导说明_V0.1.pdf》。

固件设计：收到某特定动作/指令,把 `cfg_bt_work_mode` 改为 `MODE_BQB_RF_BREDR`,然后重新调用 `bt_setup()`。测试完信号质量后断电即可。由于 `cfg_bt_work_mode` 存放在 ram 中，掉电消失，所以再次上电时，只要不发特定动作/指令,就是进入正常模式。

信令模式相当于是仪器要与 dut 端进行互动，非信令模式就是 dut 端单方面进行收或者发。

21 认证

21.1 BQB 认证

《BLE 测试指导说明_V0.1.pdf》为 BQB 蓝牙认证的 3 种模式的测试方法。

《GW3323_ble_bqb_2023_10_7.rar》为 BQB 蓝牙认证的低功耗蓝牙烧录文件。

《GW3323_bt_bqb_2023_10_7.rar》为 BQB 蓝牙认证的经典蓝牙烧录文件。

21.2 FCC/SRRC 认证

《GW3323_FCC 认证相关-v0.0.2.pdf》为 FCC 等认证相关说明。

《4.定频软件 BT_Tool 和定频方法.rar》为定频测试方法。

《5.增益天线报告.rar》为天线报告。

《bt_tool_v1.1.2.zip》为定频软件。

《GW3323_fcc_20231128.rar》进行认证时芯片的烧录文件,能开放到 2M PHY,频偏要是未达到要求的频率,可以通过打开“配置->管理配置->产测保留->不使用产测电容值”,更改自定义晶振电容值达到想要的频率。实际值偏大,就把电容改小;反之改大。

22 版本历史

表格 14 文件版本历史记录

| 日期 | 版本 | 变更历史 |
|------------|-----|--|
| 2023.05.10 | 0.1 | 新建 |
| 2023.08.24 | 0.2 | 添加充电模块，USB，ble，时钟寄存器等说明；更改其他模块的说明。 |
| 2023.12.21 | 0.3 | 添加 19.4.2 客户自定义升级的使用和开发,修改 5.1 |
| 2024.04.25 | 1.0 | 添加 21.认证说明，更改图表编号，添加开关中断、关看门狗函数，添加蓝牙时序逻辑描述， |
| 2024.08.22 | 1.1 | 添加 20.6 产线天线质量测试；修改 17.1 的 USB 全速 12Mbps；修改 4.2 呼吸模式重启；更改 14.2 功能配置的 IIC；添加 2.3 低压断电寄存器；添加 uid 说明，芯片 uid 示例；添加 19.6 和 19.7 蓝牙章节；更改 21.2 认证中固件版本。 |

声明

本手册由珠海极海半导体有限公司（以下简称“极海”）制订并发布，所列内容均受商标、著作权、软件著作权相关法律法规保护，极海保留随时更正、修改本手册的权利。使用极海产品前请仔细阅读本手册，一旦使用产品则表明您（以下称“用户”）已知悉并接受本手册的所有内容。用户必须按照相关法律法规和本手册的要求使用极海产品。

1、权利所有

本手册仅应当被用于与极海所提供的对应型号的芯片产品、软件产品搭配使用，未经极海许可，任何单位或个人均不得以任何理由或方式对本手册的全部或部分内容进行复制、抄录、修改、编辑或传播。

本手册中所列带有“®”或“™”的“极海”或“Geehy”字样或图形均为极海的商标，其他在极海产品上显示的产品或服务名称均为其各自所有者的财产。

2、无知识产权许可

极海拥有本手册所涉及的全部权利、所有权及知识产权。

极海不应因销售、分发极海产品及本手册而被视为将任何知识产权的许可或权利明示或默示地授予用户。

如果本手册中涉及任何第三方的产品、服务或知识产权，不应被视为极海授权用户使用前述第三方产品、服务或知识产权，也不应被视为极海对第三方产品、服务或知识产权提供任何形式的保证，包括但不限于任何第三方知识产权的非侵权保证，除非极海在销售订单或销售合同中另有约定。

3、版本更新

用户在下单购买极海产品时可获取相应产品的最新版的手册。

如果本手册中所述的内容与极海产品不一致的，应以极海销售订单或销售合同中的约定为准。

4、信息可靠性

本手册相关数据经极海实验室或合作的第三方测试机构批量测试获得，但本手册相关数据难

免会出现校正笔误或因测试环境差异所导致的误差，因此用户应当理解，极海对本手册中可能出现的该等错误无需承担任何责任。本手册相关数据仅用于指导用户作为性能参数参照，不构成极海对任何产品性能方面的保证。

用户应根据自身需求选择合适的极海产品，并对极海产品的应用适用性进行有效验证和测试，以确认极海产品满足用户自身的需求、相应标准、安全或其它可靠性要求；若因用户未充分对极海产品进行有效验证和测试而致使用户损失的，极海不承担任何责任。

5、合规要求

用户在使用本手册及所搭配的极海产品时，应遵守当地所适用的所有法律法规。用户应了解产品可能受到产品供应商、极海、极海经销商及用户所在地等各国有关出口、再出口或其它法律的限制，用户（代表其本身、子公司及关联企业）应同意并保证遵守所有关于取得极海产品及/或技术与直接产品的出口和再出口适用法律与法规。

6、免责声明

本手册由极海“按原样”（as is）提供，在适用法律所允许的范围内，极海不提供任何形式的明示或暗示担保，包括但不限于对产品适销性和特定用途适用性的担保。

极海产品并非设计、授权或担保适合用于军事、生命保障系统、污染控制或有害物质管理系统中的关键部件，亦非设计、授权或担保适合用于在产品失效或故障时可导致人员受伤、死亡、财产或环境损害的应用。

如果产品未标明“汽车级”，则表示不适用于汽车应用。如果用户对产品的应用超出极海提供的规格、应用领域、规范，极海不承担任何责任。

用户应该确保对产品的应用符合相应标准以及功能安全、信息安全、环境标准等要求。用户对极海产品的选择和使用负全部的责任。对于用户后续在针对极海产品进行设计、使用的过程中所引起的任何纠纷，极海概不承担责任。

7、责任限制

在任何情况下，除非适用法律要求或书面同意，否则极海和/或以“按原样”形式提供本手册及产品的任何第三方均不承担损害赔偿，包括任何一般、特殊因使用或无法使用本手册及产品而产生的直接、间接或附带损害（包括但不限于数据丢失或数据不准确，或用户或第三方遭受

的损失），这涵盖了可能导致的人身安全、财产或环境损害等情况，对于这些损害极海概不承担责任。

8、适用范围

本手册的信息用以取代本手册所有早期版本所提供的信息。

©2024 珠海极海半导体有限公司 – 保留所有权利