

## ***BL1820***

*低功耗蓝牙兼容 & 2.4-GHz 专用 SOC 芯片 V1.3*

数据手册

---

## 目录

版本历史.....	4
1. BL1820 概览 .....	5
1.1 简述 .....	5
1.2 芯片特性 .....	5
1.3 系统功能框图 .....	7
1.4 应用 .....	8
2. 脚位图 .....	9
2.1 BL1820 各型号引脚图 .....	9
2.2 BL1820 引脚描述 .....	10
3. MCU 系统 .....	12
3.1 MCU 调试 .....	12
3.2 中断向量表 .....	13
3.4 功能模块寻找表 .....	15
4. 内存 .....	16
4.1 内存介绍 .....	16
4.2 内存映射 .....	17
4.3 APB 寻址空间 .....	17
4.4 AHB 寻址空间 .....	18
5. PMU .....	19
5.1 电源管理介绍 .....	19
5.2 数字 LDO .....	19
5.3 POR/BOD .....	20
6. 外设 .....	20
6.1 管脚复用 .....	21
6.2 SDMA .....	22
6.3 SPI 控制接口 .....	26
6.4 GPIO .....	44
6.5 UART .....	50
6.6 TIMER .....	65
6.7 GPADC .....	70
6.8 IIC .....	77
6.8 WDT (看门狗定时器) .....	88
6.9 Flash .....	92
7. 电气特性 (续) .....	93
7.1 绝对最大额定值 .....	93
7.2 推荐工作条件 .....	94
7.3 复位特性 .....	94
7.4 射频特性 .....	95
7.2 GPIO 特性 (TA=25°C, VDDIO=3.0 V) .....	97
7.3 通用 ADC 特性 (TA=25°C, VIO=3.3 V) .....	98
7.4 时钟特性 .....	99
8. 参考设计 .....	101
8.1 最小系统设计 .....	101
8.2 典型应用电路示例 (无线传感器节点) .....	102
9. 封装信息 .....	104
9.1 SOP16 封装 (BL1820PPG/FPG) .....	104
9.2 SOP8 封装 (BL1820P8) .....	104

---

10. 缩略语 .....	105
---------------	-----

## 版本历史

Version	Revision	Date	Author	Reviewer
V1.08	Official version	2024/09		

## 1. BL1820 概览

### 1.1 简述

BL1820 是一款 **16 引脚 RISC-V 核 BLE/2.4G SoC** 芯片，具备高灵敏度（-96dBm@1Mbps）、宽发射功率范围（-10dBm ~ 10dBm）及高集成度（仅需 2~3 颗外部元器件）特性，支持 2.4G 私有协议，可编程通信速率覆盖 125 Kbps/250 Kbps/500 Kbps/1 Mbps/2 Mbps。

芯片支持 2400 MHz ~ 2500 MHz 宽频率输入，低至 60μS 的锁频时间结合专有自动增益控制算法和跳频算法，显著提升 2.4G 频段抗干扰性能，适用于对数据传输稳定性、延迟有高要求的订制类应用；同时针对物联网功耗敏感场景优化，接收/发射峰值电流低，内部集成 T/R 开关和 LDO，无需片外 LDO 退耦电容及射频匹配元件，简化设计并缩小产品尺寸。

### 1.2 芯片特性

#### 1.2.1 射频特性

- 通信速率：支持 125 Kbps/250 Kbps/500 Kbps/1 Mbps/2 Mbps
- 接收灵敏度：-96 dBm @ 1 Mbps
- 发射功率范围：-10 ~ 10 dBm
- RSSI：支持 8-bit RSSI 上报
- 射频匹配：无需外部匹配元件，支持天线直接连接
- 频率范围：2400 MHz ~ 2500 MHz

#### 1.2.2 MCU 子系统

- 内核：
    - 32 位 RISC-V 处理器，支持 SWD 调试，
    - 最高时钟 48 MHz
  - 外设：
    - 32 通道 System DMA
    - 硬件 AES-128 加密
    - 11 个 GPIO（全部支持低功耗唤醒）
    - UART × 2（波特率 1200 ~ 2 Mbps，可灵活配置）
- 
- SPI Master × 2（时钟速率最高 24 MHz）、SPI Slave × 1
- [www.bolingic.com](http://www.bolingic.com)

- 
- IIC × 1（通信速率最高 1 Mbps）
  - 8 路 16bit Timer/PWM
  - QSPI × 1
  - Watchdog Timer（WDT）× 1
  - 32-bit Real-Time Clock（RTC）
  - 8 通道 10-bit 3 Mbps ADC
  - 温度传感器

### 1.2.3 存储

- SRAM: 16 KB
- OTP: 16 KB（末尾 128 字节为 bootloader 配置区，用户不可用）
- Flash: 仅 Flash 版本（BL1820FPG）含 256 KB 片上 Flash，最大支持 64 MB 片外 Flash（可选，支持 XiP）

### 1.2.4 电源管理

- 供电电压: 1.7 V ~ 3.6 V（集成 LDO）
- 功耗:
  - RTC 模式: 3  $\mu$ A（保留 16K RAM）
  - 接收峰值功耗: 10 mA
  - 发射峰值功耗: 13 mA @ 0 dBm
- 保护机制: 集成 POR/BOR、低电压检测（LVD）中断，支持内部主电源/IO 电压测量

### 1.2.5 时钟

- 内部时钟: 高速 24 MHz RC 振荡器、低速 32 KHz RC 振荡器（支持所有时钟相互校准）
- 外部时钟: 24 MHz 晶体振荡器（支持 Crystal trimming）
- 倍频功能: 晶体 24 MHz 时钟可倍频至 48 MHz，供 CPU 及外设（QSPI、UART 等）使用

### 1.2.6 环境与封装

- 工作温度：-40℃ ~ +85℃
- 封装：SOP16、SOP8

## 1.3 系统功能框图

BL1820 系统架构核心分为 **MCU 子系统**、**存储模块**、**外设模块**、**电源管理单元**、**2.4G 无线模块**等几大部分，具体框图逻辑如下：

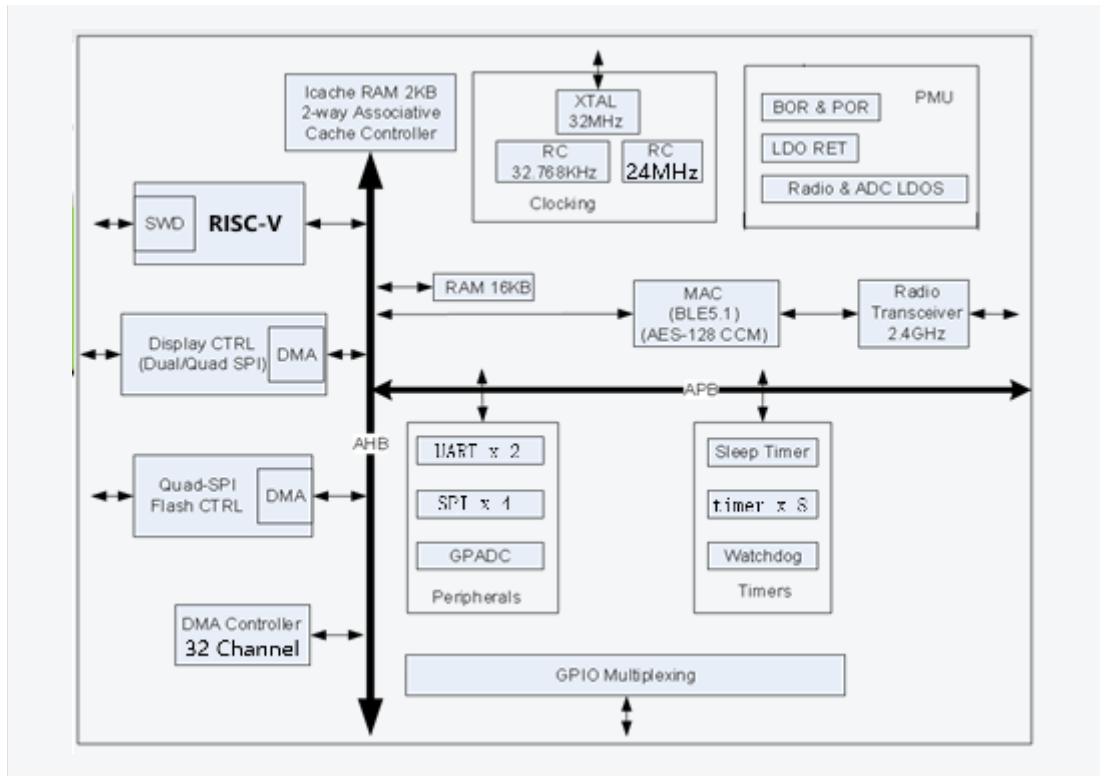


图 1 BL1820 系统框图

## 1.4 应用

BL1820 适用于以下场景：



图 2 BL1824 典型应用

1. **工业控制**：对数据传输延迟、稳定性要求高的定制化工业传感器（如无线温湿度传感器、设备状态监测模块）；
2. **物联网（IoT）**：功耗敏感型设备（如智能穿戴配件、低功耗无线节点）；
3. **消费电子**：2.4G 私有协议无线连接设备（如无线遥控器、游戏手柄、无线键鼠）；
4. **智能家居**：短距离低功耗无线通信模块（如智能灯具控制、门窗传感器）。
5. **运动健康**
6. **有源标签**



## 2. 脚位图

### 2.1 BL1820 各型号引脚图

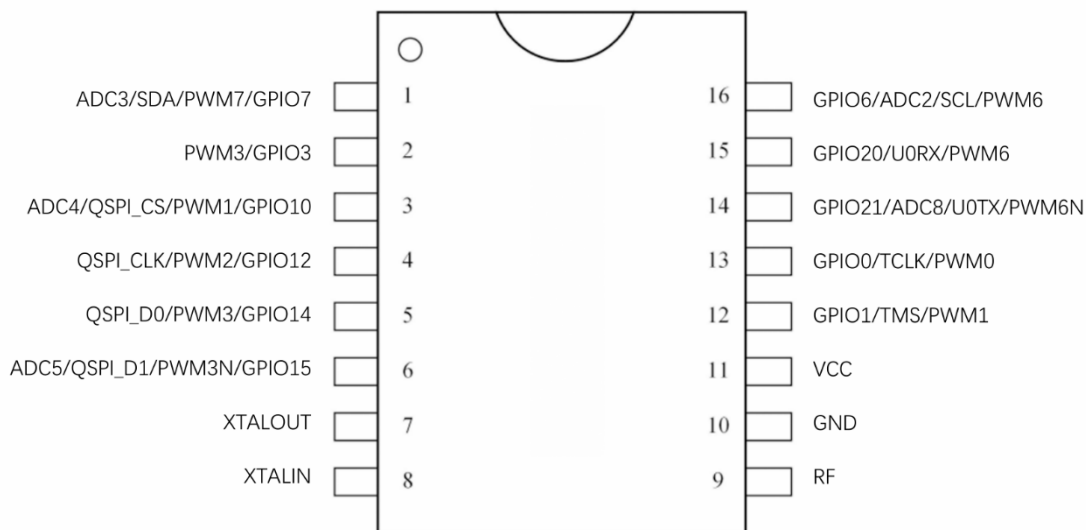


图 2-1 BL1820PPG

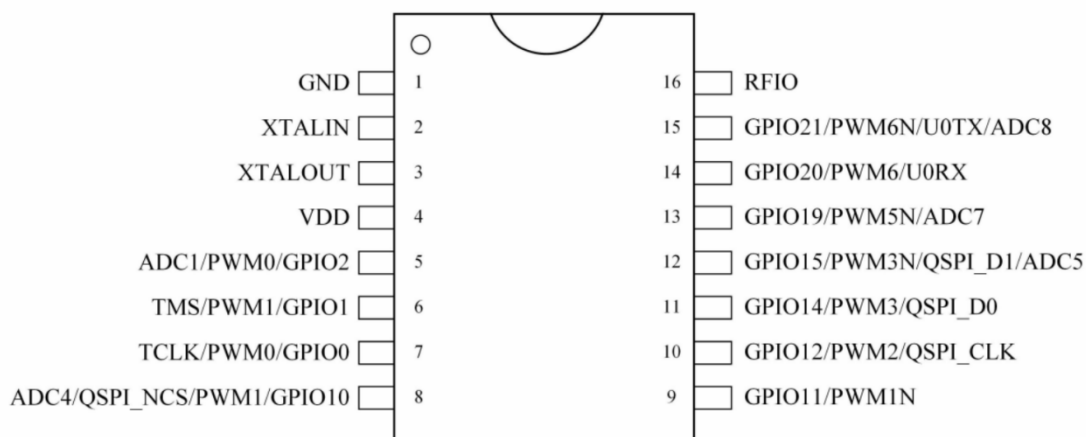


图 2-2 BL1820FPG

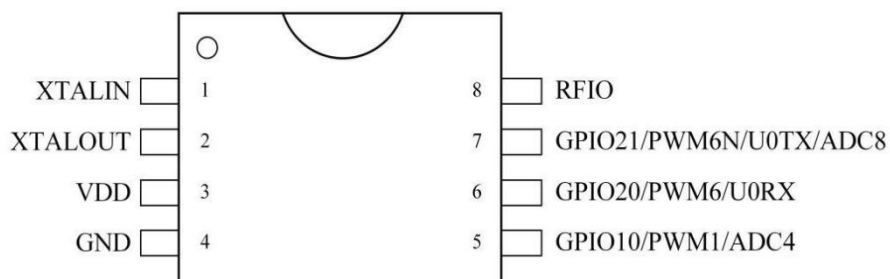


图 2-3 BL1820P8

## 2.2 BL1820 引脚描述

名称	1820PP G	1820FPG	1820P8	类型	描述
ADC3/SDA/PWM7/ GPIO7	1			AN/ I/O/ O/I/ O	ADC3 模拟输入口 / IIC 数据口 / PWM7 输出口 / 通用输入/输出口
PWM3 GPIO3	2			O/I/ O	PWM3 输出口 / 通用输入/输出口
ADC4/QSPI_CS/PW M1/GPIO10	3	8	5	AN/ O/O /I/O	ADC4 模拟输入口 / QSPI 片选信号口 / PWM1 输出口 / 通用输入/输出口
QSPI_CLK/PWM2/G PIO12	4	10		O/O /I/O	QSPI 时钟口 / PWM2 输出口 / 通用输入/输出口
QSPI_D0/PWM3/GP IO14	5	11		I/O/ O/I/ O	QSPI 数据输入/输出 0 / PWM3 输出口 / 通用输入/输出口
ADC5/QSPI_D1/PW M3N/GPIO15	6	12		AN/ I/O/ O/I/ O	ADC5 模拟输入口 / QSPI 数据输入/输出 1 / PWM3N 输出口 / 通用输入/输出口
XTALOUT	7	3	2	AN	晶体时钟输出引脚（外部 24 MHz 晶体输出）
XTALIN	8	2	1	AN	晶体时钟输入引脚（外部 24 MHz 晶体输入）
RF_ANT	9	16	8	AN	2.4G 射频收发引脚（直接连接天线，无需外部匹配）
GND	10	1	4		
VDD	11	4	3	P	RF 电源和数字电源
TMS/PWM1/GPIO1	12	6		I/O/ O/I/	JTAG 数据接收口 / PWM1

				O	输出口 / 通用输入/输出口
<b>TCLK /PWM0/GPIO0</b>	13	7		I O I/O	JTAG 时钟输入口/ PWM0 输出口 /输入/输出口
<b>ADC8/U0TX/PWM6 N/GPIO21</b>	14	15	7	AN/ O/O I/O	ADC8 模拟输入口 / UART0 输出口 / PWM6N 互补输出 口 / 通用输入/输出口
<b>U0RX/PWM6/GPIO2 0</b>	15	14	6	I/O/ O/I/ O	UART0 接收口 / PWM6 输 出口 / 通用输入/输出口
<b>SCL/PWM6/GPIO6</b>	16			I/O/ O/I/ O	IIC 时钟口 / PWM6 输出口 / 通用输入/输出口
<b>ADC7 /PWM5N / GPIO19</b>		13		AN O I/O	ADC7 输入口 /PWM5N 输 出口/输入/输出口
<b>PWM1N/GPIO11</b>		9		O/I/ O	PWM1N 互补输出口 / 通用 输入/输出口
<b>ADC1/PWM0/GPIO2</b>		5		AN/ O/I/ O	ADC1 模拟输入口 / PWM0 输出口 / 通用输入/输出口

注：类型缩写说明——I=输入，O=输出，I/O=输入/输出，P=电源，AN=模拟输入/输出。

## 2.3 端口复用功能说明

IO	Func1	Func2	Func3	Func4	Func5	Func7
GPIO0	JTAG TCLK	IIC0 SCL	SPIM0 NCS	UART0 CTS	SPIS0 NCS	PWM 0 P
GPIO1	JTAG_TMS	IIC0 SDA	SPIM0_SCK	UART0_RTS	SPIS0_SCK	PWM 1 P
GPIO2	JTAG_TRSTB	UART0_RX	SPIM0_MOSI	UART1_CTS	SPIS0_MISO	PWM_2 P

IO	Func1	Func2	Func3	Func4	Func5	Func7
GPIO3	Reserved	UART0_TX	SPIM0_MISO	UART1_RTS	SPIS0_MOSI	PWM_3_P
GPIO4	QSPI_NCS	UART1_RX	SPIM1_SCK	SPIS0_NCS	UART0_RX	PWM_4_P
GPIO5	QSPI_D3	UART1_TX	SPIM1_NCS	SPIS0_SCK	UART0_TX	PWM_5_P
GPIO6	QSPI_CLK	IIC0_SCL	SPIM1_MOSI	SPIS0_MOSI	UART1_RX	PWM_6_P
GPIO7	QSPI_D2	IIC0_SDA	SPIM1_MISO	SPIS0_MISO	UART1_TX	PWM_7_P
GPIO8	QSPI_D0	UART0_RX	SPIM0_NCS	UART1_CTS	SPIS0_NCS	PWM_0_P
GPIO9	QSPI_D1	UART0_TX	SPIM0_SCK	UART1_RTS	SPIS0_SCK	PWM_0_N
GPIO10	QSPI_NCS	UART1_RX	SPIM0_MOSI	UART0_CTS	SPIS0_MOSI	PWM_1_P
GPIO11	QSPI_D3	UART1_TX	SPIM0_MISO	UART0_RTS	SPIS0_MISO	PWM_1_N
GPIO12	QSPI_CLK	IIC0_SCL	SPIM1_SCK	SPIS0_NCS	UART0_RX	PWM_2_P
GPIO13	QSPI_D2	IIC0_SDA	SPIM1_NCS	SPIS0_SCK	UART0_TX	PWM_2_N
GPIO14	QSPI_D0	UART0_RX	SPIM1_MOSI	SPIS0_MOSI	UART1_RX	PWM_3_P
GPIO15	QSPI_D1	UART0_TX	SPIM1_MISO	SPIS0_MISO	UART1_TX	PWM_3_N
GPIO16	QSPI_NCS	UART1_RX	SPIM0_NCS	IIC0_SCL	SPIS0_NCS	PWM_4_P
GPIO17	QSPI_D3	UART1_TX	SPIM0_SCK	IIC0_SDA	SPIS0_SCK	PWM_4_N
GPIO18	QSPI_CLK	IIC0_SCL	SPIM0_MOSI	UART1_CTS	SPIS0_MOSI	PWM_5_P
GPIO19	QSPI_D2	IIC0_SDA	SPIM0_MISO	UART1_RTS	SPIS0_MISO	PWM_5_N
GPIO20	QSPI_D0	UART0_RX	SPIM1_SCK	SPIS0_NCS	IIC0_SCL	PWM_6_P
GPIO21	QSPI_D1	UART0_TX	SPIM1_NCS	SPIS0_SCK	IIC0_SDA	PWM_6_N
GPIO22	IIC0_SCL	UART1_RX	SPIM1_MOSI	SPIS0_MOSI	QSPI_CLK	PWM_7_P
GPIO23	IIC0_SDA	UART1_TX	SPIM1_MISO	SPIS0_MISO	QSPI_NCS	PWM_7_N

## 3. MCU 系统

### 3.1 MCU 调试

BL1820 的 MCU (32 位 RISC-V 核) 支持 **SWD (Serial Wire Debug)** 两线调试接口，调

试引脚复用 GPIO0 (TCLK) 和 GPIO1 (TMS)，具体功能如下：

- TCLK (GPIO0)：调试时钟信号，由调试器输出；
- TMS (GPIO1)：调试模式选择信号，双向传输，用于调试命令与数据交互；
- 调试功能：支持程序下载、断点调试、寄存器读写、内存访问，兼容 RISC-V debug 架构。

## 3.2 中断向量表

BL1820 中断控制器兼容 RISC-V CLIC 标准，支持中断嵌套，中断向量表如下：

Number	IRQ	Bit	Description
0~2	N/A	-	保留
3	SOFT_IRQ	1	软件中断
4~6	N/A	-	保留
7	TIMER_IRQ	1	定时器全局中断
8~15	N/A	-	保留
16	WDT	1	看门狗定时器中断
17	N/A	-	保留
18	RADIO_CTRL	1	2.4G 控制器中断
19	RADIO_TIMER	1	2.4G 定时器中断
20	RADIO_PTMR	1	2.4G 周期定时器中断
21~22	N/A	-	保留
23	GPADC	1	通用 ADC 中断
24	UART_0	1	UART0 中断
25	UART_1	1	UART1 中断
26	SPIM_0	1	SPI Master 0 中断

Number	IRQ	Bit	Description
27	SPIM_1	1	SPI Master 1 中断
28	IIC	1	IIC 中断
29	N/A	-	保留
30	SPIS	1	SPI Slave 中断
31	N/A	-	保留
32	GPIO	1	GPIO 全局中断
33	QDEC	1	正交解码器中断
34	AES	1	AES 加密模块中断
35	CAP	1	捕获模块中断
36	CLKCAL	1	时钟校准中断
37	RTC	1	实时时钟中断
38	SDMA	1	SDMA 控制器中断
39	LVD	1	低电压检测中断
40	TIMER_0	1	定时器 0 中断
41	TIMER_1	1	定时器 1 中断
42	TIMER_2	1	定时器 2 中断
43	TIMER_3	1	定时器 3 中断
44	TIMER_4	1	定时器 4 中断
45	TIMER_5	1	定时器 5 中断
46	TIMER_6	1	定时器 6 中断

Number	IRQ	Bit	Description
47	TIMER_7	1	定时器 7 中断

### 3.4 功能模块寻找表

BL1820 各功能模块的基地址如下，便于软件访问与配置：

BaseAddress	Module	Description
0x40000000	SYS_REG	系统控制寄存器
0x41008000	TIMER	定时器模块（Timer0~Timer7）
0x42000000	2.4G_RF	2.4G 射频模块
0x40010000	GPIO	GPIO 控制器
0x41001000	UART0	通用异步收发器 0
0x41002000	UART1	通用异步收发器 1
0x40040000	GPADC	通用 ADC 模块
0x41003000	SPIM0	SPI Master 0
0x41004000	SPIM1	SPI Master 1
0x41007000	SPIS	SPI Slave
0x41005000	IIC	IIC 控制器
0x1F800000	OTP	一次性可编程存储器（16KB）
0x20000000	SRAM	静态随机存储器（16KB）
0x10000000	Flash	片外 Flash（最大 64MB，支持 XiP）

## 4. 内存

### 4.1 内存介绍

BL1820 的存储系统包含片内存储（**SRAM、OTP**）和片外扩展存储（**Flash**），具体配置如下：

1. **SRAM**：总容量 16 KB，分为 3 个独立区块，每块可单独设置 Retention（低功耗模式下数据保持）：

- RAM Block 0：4 KB
- RAM Block 1：4 KB
- RAM Block 2：8 KB

2. **OTP**：总容量 16 KB，地址范围 0x1F80\_0000 ~ 0x1F803FFF，其中末尾 128 字节（0x1F803F80 ~ 0x1F803FFF）为 bootloader 配置区，用户不可访问，剩余 16 KB - 128 B 为用户可用区；

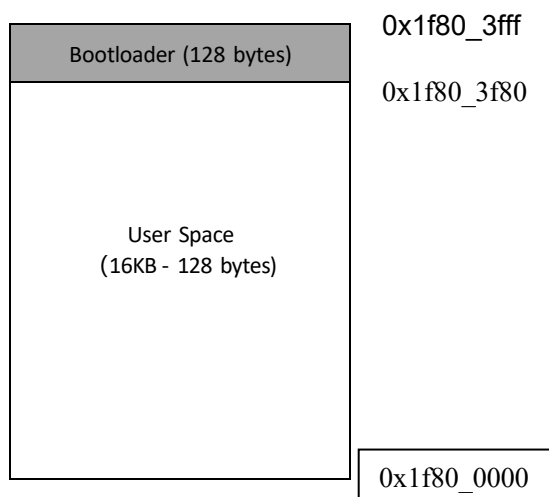


图 4-1 OTP 存储器映射

3. **Flash**：

- 片上 Flash：仅 BL1820FPG 型号集成 256 KB，地址范围 0x10000000 ~ 0x10040000；
- 片外 Flash：最大支持 64 MB，支持 XiP（Execute-in-Place，直接从 Flash 执行程序），通过 QSPI 接口访问。



## 4.2 内存映射

BL1820 的内存地址空间分布如下（32 位地址）：

AddressRange	Size	Module	Description
0x00000000 ~ 0x00007FFF	32 KB	ROM	只读存储器（存放 bootloader）
0x1F800000 ~ 0x1F803FFF	16 KB	OTP	一次性可编程存储器（用户区+boot 区）
0x20000000 ~ 0x20003FFF	16 KB	SRAM	静态随机存储器（Block0+Block1+Block2）
0x40000000 ~ 0x5FFFFFFF	0.5GB	Peripherals	外设寄存器区（GPIO、UART、SPI 等）
0x40000000 ~ 0x40FFFFFF	64 MB	APB1 Peripherals	APB1 总线外设（Timer、SPIS 等）
0x42000000 ~ 0x42FFFFFF	12 KB	2.4G Radio/Modem	2.4G 射频/调制解调器寄存器
0x1000_0000 ~ 0x13FFFFFF	64 MB	External Flash	片外 Flash（支持 XiP）
0x41000000 ~ 0x41FFFFFF	64 MB	AHB1 Peripherals	AHB1 总线外设（CPU、RAM 等）
0x40000000 ~ 0x4000FFFF	64 KB	System Control	系统控制寄存器（复位、时钟等）

## 4.3 APB 寻址空间

APB 总线（Advanced Peripheral Bus）用于连接低速外设，BL1820 的 APB 寻址空间细分如下：

Address Range	Size	Module	Description
0x41000000 ~ 0x41001FFF	4 KB	UART0	UART0 相关寄存器
0x41002000 ~ 0x41002FFF	4 KB	UART1	UART1 相关寄存器
0x41003000 ~ 0x41003FFF	4 KB	SPIM0	SPI Master 0 相关寄存器
0x41004000 ~ 0x41004FFF	4 KB	SPIM1	SPI Master 1 相关寄存器
0x41005000 ~ 0x41006FFF	4 KB	IIC	IIC 相关寄存器
0x41007000 ~ 0x41007FFF	4 KB	SPIS	SPI Slave 相关寄存器
0x41008000 ~ 0x41008FFF	4 KB	TIMER	定时器 (Timer0~Timer7) 相关寄存器

## 4.4 AHB 寻址空间

AHB 总线 (Advanced Peripheral Bus) 用于连接高速外设, BL1820 的 AHB 寻址空间细分如下:

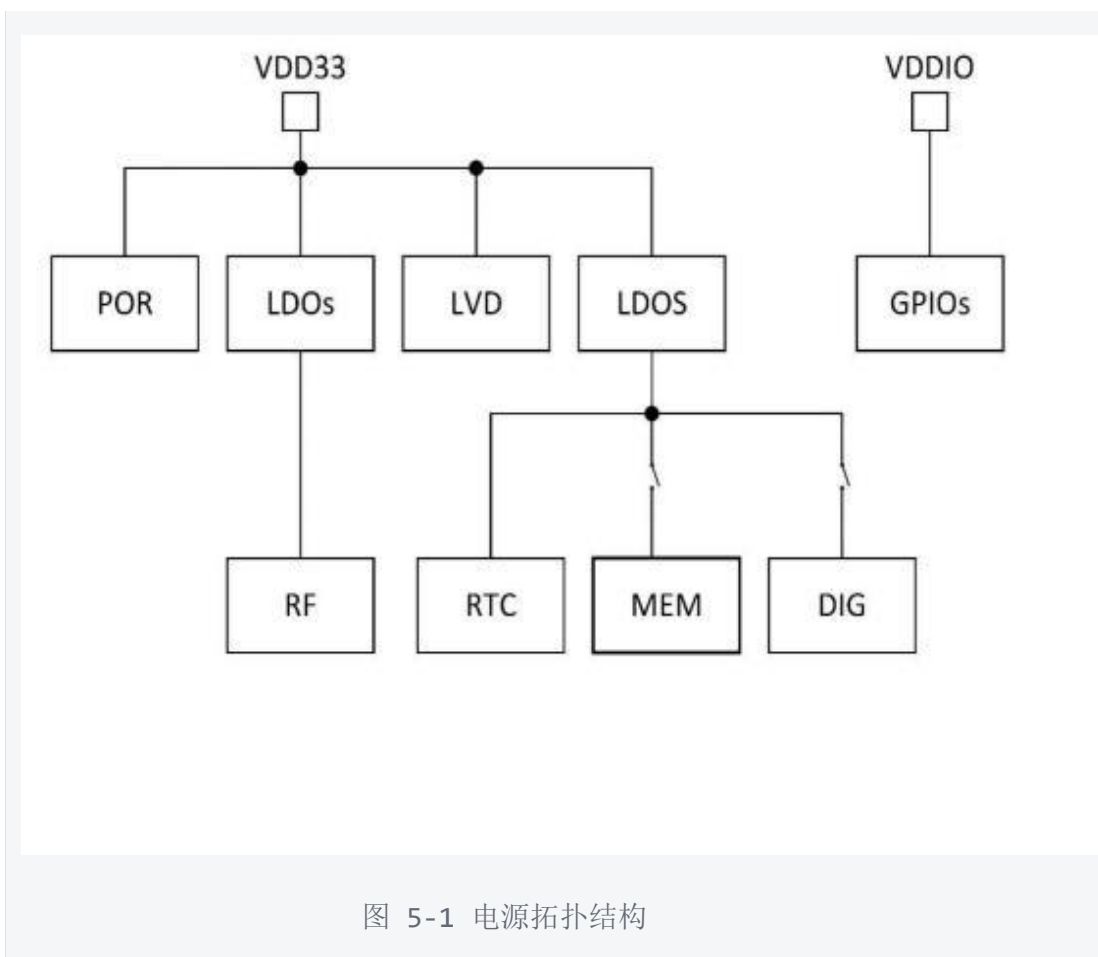
Address Range	Size	Module	Description
0x40000000 ~ 0x4000FFFF	64 KB	SYS_REG	SYS 相关寄存器
0x40010000 ~ 0x4001FFFF	64 KB	GPIO	GPIO 相关寄存器
0x40020000 ~ 0x4002FFFF	64 KB	QDEC	QDEC 相关寄存器
0x40030000 ~ 0x4003FFFF	64 KB	AES	AES 相关寄存器
0x40040000 ~ 0x4007FFFF	64 KB	GPADC	通用 ADC 寄存器

Address Range	Size	Module	Description
0x40080000 ~ 0x4008FFFF	64 KB	LPM	SPI Slave 寄存器

## 5. PMU

### 5.1 电源管理介绍

BL1820 的电源管理单元（PMU）负责为芯片各模块提供稳定供电，并实现低功耗控制，电源拓扑如下：



- **LDO 分类：**包含 RF 专用 LDO、数字电路 LDO、IO LDO，分别为 2.4G 射频模块、MCU/外设、GPIO 提供稳定电压；
- **功耗控制：**支持多档功耗模式（RTC 模式、休眠模式、正常工作模式），通过关闭闲置模块时钟/电源降低功耗。

### 5.2 数字 LDO

数字 LDO 为 MCU 内核、SRAM、数字外设（UART、SPI 等）提供供电，技术参数如下：

Parameter	Symbol	Condition	Min	Typ	Max	Unit
输入电压范围	VIN	来自 VDD 电源	1.7		3.9	V
输出电压范围	VOUT	数字内核供电	1.0	1.1	1.2	V
输出电流能力	IOUT_MAX	满载时	-	50	100	mA

## 5.3 POR/BOD

### 5.3.1 上电复位 (POR)

- **功能：**电源电压从 0 上升时，当 VDD 达到 VPOR（1.7 V 典型值）后，POR 模块延迟 TPOR（20 ms 典型值）后释放复位，确保芯片稳定启动；若 VDD 下降至 VPDR（1.2 V 典型值）以下，立即触发复位；
- **应用：**芯片上电初始化、电源电压骤降保护。

### 5.3.2 低电压检测 (LVD)

- **功能：**实时监测 VDD/VDDIO 电压，当电压低于设定阈值（可配置，默认 1.6 V）时，可触发中断（通知 CPU 处理）或直接复位（防止芯片异常工作）；
- **特性：**支持内部主电源（VDD）和 IO 电源（VDDIO）分别检测，中断可独立使能/禁用，复位仅影响非 AON（Always-On）区域。

## 6. 外设

## 6.1 管脚复用

BL1820 的内部有 24 个 GPIO（GPIO0~GPIO23）均支持多外设复用，通过寄存器 PIN\_MUX\_CTRL\_1~PIN\_MUX\_CTRL\_5 配置，部分常用管脚复用表如下：

IO	Func1	Func2	Func3	Func4	Func5	Func7
GPIO0	JTAG_TCLK	IIC0_SCL	SPIM0_NCS	UART0_CTS	SPIS0_NCS	PWM_0_P
GPIO1	JTAG_TMS	IIC0_SDA	SPIM0_SCK	UART0_RTS	SPIS0_SCK	PWM_1_P
GPIO2	JTAG_TRSTB	UART0_RX	SPIM0_MOSI	UART1_CTS	SPIS0_MISO	PWM_2_P
GPIO3	Reserved	UART0_TX	SPIM0_MISO	UART1_RTS	SPIS0_MOSI	PWM_3_P
GPIO4	QSPI_NCS	UART1_RX	SPIM1_SCK	SPIS0_NCS	UART0_RX	PWM_4_P
GPIO5	QSPI_D3	UART1_TX	SPIM1_NCS	SPIS0_SCK	UART0_TX	PWM_5_P
GPIO6	QSPI_CLK	IIC0_SCL	SPIM1_MOSI	SPIS0_MOSI	UART1_RX	PWM_6_P
GPIO7	QSPI_D2	IIC0_SDA	SPIM1_MISO	SPIS0_MISO	UART1_TX	PWM_7_P
GPIO8	QSPI_D0	UART0_RX	SPIM0_NCS	UART1_CTS	SPIS0_NCS	PWM_0_P
GPIO9	QSPI_D1	UART0_TX	SPIM0_SCK	UART1_RTS	SPIS0_SCK	PWM_0_N
GPIO10	QSPI_NCS	UART1_RX	SPIM0_MOSI	UART0_CTS	SPIS0_MOSI	PWM_1_P
GPIO11	QSPI_D3	UART1_TX	SPIM0_MISO	UART0_RTS	SPIS0_MISO	PWM_1_N
GPIO12	QSPI_CLK	IIC0_SCL	SPIM1_SCK	SPIS0_NCS	UART0_RX	PWM_2_P

IO	Func1	Func2	Func3	Func4	Func5	Func7
			K	S		P
GPIO13	QSPI_D2	IIC0_SDA	SPIM1_NC S	SPIS0_SC K	UART0_TX	PWM_2_ N
GPIO14	QSPI_D0	UART0_R X	SPIM1_MO SI	SPIS0_MO SI	UART1_RX	PWM_3_ P
GPIO15	QSPI_D1	UART0_T X	SPIM1_MIS O	SPIS0_MIS O	UART1_TX	PWM_3_ N

### 6.1.1 管脚复用寄存器映射

Address	Name	Description
0x40010204	GPIO_CFG_1	GPIO0~GPIO3 复用配置（每 GPIO 占 8 bit）
0x40010208	GPIO_CFG_2	GPIO4~GPIO7 复用配置（每 GPIO 占 8 bit）
0x4001020C	GPIO_CFG_3	GPIO8~GPIO11 复用配置（每 GPIO 占 8 bit）
0x40010210	GPIO_CFG_4	GPIO12~GPIO15 复用配置（每 GPIO 占 8 bit）
0x40010214	GPIO_CFG_5	GPIO16~GPIO18 复用配置（每 GPIO 占 8 bit）

寄存器配置规则：每个 GPIO 的复用控制位为 6 bit（如 GPIO0 对应 PIN\_MUX\_CTRL\_1[5:0]），不同值对应不同功能（参考上表 Func1~Func7），默认值为 0（GPIO 模式）。

## 6.2 SDMA

## 6.2.1 简介

SDMA (Simple Direct Memory Access) 将数据从一个地址空间搬移到另一个地址空间而不需要 CPU 参与, 提供 在外设和存储器之间或者存储器 和存储器之间的高速数据传输。当 CPU 初始化这 个传输动作, 传输动作本身是由 DMA 控制器来实现和完成。DMA 传输方式无需 CPU 直接控制传输, 也没有中断处理方式那样保留现场和恢复现场过程, 通过硬件 为内存与外设开辟一条直接传输数据的通道, 使得 CPU 的效率大大提高。

## 6.2.2 主要特性

- 通道数量: 32 路独立通道, 支持优先级配置 (高/低两档);
- 传输宽度: 8-bit、32-bit;
- 传输模式: 标准模式、FIFO 模式、CMD 模式;
- 地址模式: 两种地址类型, 即外设、存储器; 两种地址变化模式: 固定、自增。
- 寻址范围: 32 位全空间 (最大传输 64 KB);
- 中断: 每路通道传输完成可独立触发中断。

## 6.2.3 功能描述

### [6.2.3.1]传输模式

**标准模式:** SDMA 标准模式是从源地址区域向目的地址区域搬移数据内容, 搬移过程中源区域和 目的区域内存地址线性增加, 或者不增加, 源和目的侧可以单独设置。标准模式搬移指定 大小数据内容后, 完成通道任务。数据搬移单位由配置域 DMA\_WORD\_EN 进行设置, 进 行 8 位字节搬移或 32 位字进行搬移。数据搬移长度由配置域 NUM\_BYTES 设置, 最大搬 移大小为 64KB, 当 DMA\_WORD\_EN 为“1”时, NUM\_BYTES 最低两位必须是 0, 以保证 32 位字对齐。

**CMD 模式:** CMD 模式传输除了在原有标准模式传输的基础上, 会在标准传输前或传输后额外 执行 一条 CMD 指令。DMA 配置域 CMD\_PRE\_EN 用于决定 CMD 执行先后顺序, 该位为 1 时, 先 执行 CMD 指令, 再执行 DMA 传输; 该位为 0 时, 限制性 DMA 输, 再执行 CMD 操作。CMD 写 入内容由配置域 CMD\_WAR 确定, 该位为 0 时, 为 Mask 写模式, 直接向 CMD\_ADDR 写入 (CMD\_WDATA&(~CMD\_MASK))。该位为 1 时, 为“读改写”模式, 首先读取 CMD\_ADDR, 并存储至 CMD\_RDATA, 再向 CMD\_ADDR 写入(CMD\_RDATA& CMD\_MASK) | (CMD\_WDATA &(~CMD\_MASK))。

**FIFO 模式:** FIFO 模式和标准模式传输类似, 不同的是源起始地址与目标地址在 FIFO 模式下 递增 到 FIFO\_SIZE 边界时, 源起始地址或目的起始地址。FIFO 模式由配置域 FIFO\_MODE<1:0> 进行配置:

- Bit[1]: 源起始地址 FIFO 模式使能开关
- Bit[0]: 目标地址 FIFO 模式使能开关。

### [6.2.3.2] SDMA 中断

SDMA 32 个任务均可独立产生中断，通过任务描述符 DMA\_TASK\_CFG0<31> INT\_EN 控制中断，当对应任务的 INT\_EN 配置为“1”时，当任务传输完成后，将触发产生 SDMA 中断。SDMA 每个任务的中断状态存储在寄存器 SDMA\_INT\_STATUS<31:0>中，SDMA 中断服务程序应先读取 SDMA\_INT\_STATUS<31:0>，确定中断来自哪一个 SDMA 任务，从而选择对应的中断处理程序，中断处理完成后，通过寄存器 SDMA\_INT\_CLEAR<31:0>清除对应的中断标志位。

- **触发条件：**通道传输完成（需通过 DMA\_TASK\_CFG0[31]使能中断）；
- **状态查询：**中断状态存储于 SDMA\_INT\_STATUS[31:0]，每 bit 对应一路通道；
- **中断清除：**通过 SDMA\_INT\_CLEAR[31:0]写入 1 清除对应通道中断标志。

## 6.2.4 寄存器映射

SDMA 基地址：0x40000000，其他有关 SDMA 的寄存器配置，在此基地址加上相应的偏移地址。

### 1. SDMA 使能寄存器（sdma\_en）

偏移地址：0x110

sdma_en	[0]	RW
---------	-----	----

SDMA 失能/使能：写 1 使能，写 0 失能。

### 2. SDMA 任务表地址寄存器（sdma\_task\_table\_addr）

偏移地址：0x114

sdma_task_table_addr	[31:0]	RW
----------------------	--------	----

32dma 逻辑信道表的起始地址，每个字指向每个信道的任务描述符。

### 3. SDMA 操作寄存器（sdma\_action）

偏移地址：0x118

RW	RW	RW
[2]	[1]	[0]
sdma_task_remove	sdma_task_add	sdma_task_load



bit0	sdma_task_load: 任务装载动作位, 写 1 对 sdma_task_val 中被置 1 的通道执行“装载任务”操作
bit1	sdma_task_add: 任务添加动作位, 写 1 对 sdma_task_val 中被置 1 的通道执行“添加任务”操作
bit2	sdma_task_remove: 任务移除动作位, 写 1 对 sdma_task_val 中被置 1 的通道执行“移除任务”操作

#### 4. SDMA 任务值寄存器 (sdma\_task\_val)

偏移地址: 0x11C

sdma_task_status	[31:0]	RO
------------------	--------	----

每个逻辑 dma 信道的状态, 1: 通道正在运行, 0: 通道空闲

#### 5. SDMA 任务值寄存器 (sdma\_task\_val)

偏移地址: 0x120

sdma_task_val	[31:0]	RW
---------------	--------	----

dma 任务操作通道, 每个比特对应每个 dma 信道。

#### 6. SDMA 中断状态寄存器 (sdma\_int\_status)

偏移地址: 0x124

sdma_int_status	[31:0]	RO
-----------------	--------	----

DMA 通道中断状态, 1 表示该通道有待处理中断; 0 表示无中断

#### 7. SDMA 中断清除寄存器 (sdma\_int\_clear)

偏移地址: 0x128

sdma_int_clear	[31:0]	RW
----------------	--------	----

中断清除寄存器, 对任意位写 1 将立即清除 sdma\_int\_status 中的对应中断标志; 写 0 无影响。

## 6.3 SPI 控制接口

### 6.3.1 简介

串行外设接口（SPI）是同步串行通信协议，BL1820 集成 **2 个 SPI Master（SPIM0/SPIM1）** 和 **1 个 SPI Slave（SPIS）**，支持半/全双工通信，适用于与外部 SPI 设备（如 Flash、传感器、显示屏）的数据交互。

- **SPI Master**: 主动发起通信，提供时钟信号（SCK），控制片选（NCS），适用于主机控制从机的场景；
- **SPI Slave**: 被动响应通信，时钟由 Master 提供，仅在选中（NCS 有效）时传输数据，适用于从机接收主机指令的场景。

### 6.3.2 SPI Master

#### [6.3.2.1] SPI Master 简介

SPI Master 通过 MOSI（主发从收）、MISO（主收从发）、SCK（时钟）、NCS（片选）4 个引脚与外部从设备通信，支持四线、三线传输模式，最高时钟速率 24 MHz，满足中高速数据传输需求。

#### [6.3.2.2] SPI Master 主要特性

特性分类	具体参数/功能
通信模式	全双工四线、半双工四线、半双工三线
帧格式	8-bit/16-bit 可选

特性分类	具体参数/功能
时钟性能	最高 24 MHz，时钟极性（SCPOL）、相位（SCPH）可配置，支持 4 种 SPI 传输模式
FIFO 缓冲	发送/接收 FIFO 深度均为 16 字节，减少 CPU 中断频率
传输控制	支持 EEPROM READ 模式（专为 Flash 读命令设计，含 3 字节命令帧）
中断支持	发送空、发送溢出、接收满、接收溢出、接收欠流中断，可独立使能/屏蔽

### [6.3.2.3] SPI Master 功能描述

#### 1. 传输模式

通过 CTRLR0 寄存器配置，支持 4 种传输模式：

- **TX AND RX（全双工）**：发送数据的同时接收从机反馈数据（如 SPI Flash 读写）；
- **TX ONLY（半双工发送）**：仅通过 MOSI 发送数据，MISO 引脚无效（如仅写传感器配置）；
- **RX ONLY（半双工接收）**：仅通过 MISO 接收数据，MOSI 引脚无效（如仅读传感器数据）；
- **EEPROM READ（Flash 专用）**：发送 3 字节命令帧（8 位命令+16 位地址），用于读取片外 Flash 数据。

#### 2. 时钟配置

- **时钟公式**：  $f_{clk\_out} = \frac{f_{spi\_clk}}{SCKDV}$ ，其中， $f_{spi\_clk}$  为 SPI 参考时钟（最大 48 MHz），SCKDV 为分频系数（由 BAUDR 寄存器配置，取值为 0~65534 的偶数，最小 2，对应最高时钟

24 MHz)。

- 时钟极性 (SCPOL) :
  - 0: 空闲时 SCK 为低电平;
  - 1: 空闲时 SCK 为高电平。
- 时钟相位 (SCPH) :
  - 0: 数据在 SCK 第一个边沿 (SCPOL=0 为上升沿, SCPOL=1 为下降沿) 采样;
  - 1: 数据在 SCK 第二个边沿 (SCPOL=0 为下降沿, SCPOL=1 为上升沿) 采样。

### 3. 线路模式

- 四线模式: 使用 MOSI、MISO、SCK、NCS, 支持全双工/半双工;
- 三线模式: MOSI 与 MISO 复用为 DIO 引脚, 仅支持半双工 (发送时用 TX ONLY, 接收时用 RX ONLY)。

### 4. 中断

中断类型	触发条件	清除方式
发送空中断	发送 FIFO 数据量 $\leq$ 阈值 (TXFTLR 配置, 0~15)	写入数据至 FIFO, 使其数据量 $>$ 阈值
发送溢出中断	发送 FIFO 满时继续写入数据	读取 ICR (中断清除寄存器) 写 1 清除
接收满中断	接收 FIFO 数据量 $\geq$ 阈值+1 (RXFTLR 配置)	读取 FIFO 数据, 使其数据量 $<$ 阈值
接收溢出中断	接收 FIFO 满时继续接收数据	读取 ICR 写 1 清除
接收欠流中断	接收 FIFO 空时尝试读取数据	读取 ICR 写 1 清除

### [6.3.2.4] 寄存器映射（以 SPIM0 为例）

SPIM0 基地址：0x41003000，其他有关 SPIM0 寄存器配置，在此基地址加上相应的偏移地址。同样地，SPIM1 基地址：0x41004000，其他有关 SPIM1 寄存器配置，在此基地址加上相应的偏移地址。

#### 1. 控制寄存器 0 (CTRLR0)

偏移地址：0x0

RO	RO	RO	RW	RW	RW	RW	RW	RW	RW	RW
[24]	[22:21]	[20:16]	[15:12]	[11]	[10]	[9:8]	[7]	[6]	[5:4]	[3:0]
SSTE	SPI_FRF	DFS $\frac{3}{2}$	CFS	SRL	SLV_OE	TMOD	SCPOL	SCPH	FRF	DFS

bit[3:0]	DFS: 数据帧大小 此寄存器字段仅当 SSI_MAX_XFER_SIZE=16 时有效
----------	---

bit[5:4]	FRF: 帧格式选择。 0x0: Motorola SPI 协议 0x1: TI SSP 协议 0x2: National Microwire 协议 0x3: 保留值
bit[6]	SCPH: 串行时钟相位 Motorola SPI 模式下定义时钟与从机选择信号的时序关系 0x0: SCPH_MIDDLE 数据在首个时钟边沿采样（时钟在首数据位中间翻转） 0x1: SCPH_START 数据在第二个时钟边沿采样（时钟在从机选择激活后延迟一个周期翻转）
bit[7]	SCPOL: 串行时钟极性 Motorola SPI 模式下定义空闲时钟电平。 0x0: SCLK_LOW 空闲状态为低电平 0x1: SCLK_HIGH 空闲状态为高电平

bit[9:8]	TMOD: 传输模式。定义数据有效方向 0x0: TX_AND_RX 全双工 (TX/RX 均有效) 0x1: TX_ONLY 仅发送 (忽略 RX 数据) 0x2: RX_ONLY 仅接收 (忽略 TX 数据) 0x3: EEPROM_READ EEPROM 读模式 (主机发送控制字后切换为接收, 仅主模式有效)
bit[10]	SLV_OE: 从机输出使能仅对从机模式有效 0x0: 允许从机正常输出数据 (默认状态) 0x1: 强制 ssi_oe_n 为高阻态, 禁用 TXD 输出 (广播模式下防止总线冲突)
bit[11]	SRL: 移位寄存器环回测试 测试专用。将 TX 移位寄存器输出直连至 RX 移位寄存器输入 0x0: NORMAL_MODE 正常操作模式 0x1: TEST_MODE 环回测试模式 (需外部提供 ss_in_n 和 ssi_clk 信号)
bit[15:12]	CFS: 控制帧长度 (Microwire 模式) 选择 Microwire 帧格式下控制字的位宽
bit[20:16]	DFS_32:32 位传输模式下数据帧长度 仅当 SSI_MAX_XFER_SIZE=32 时有效。接收数据自动右对齐, 高位补零至 32 位
bit[22:21]	SPI_FRF: SPI 帧格式选择 仅当 SSI_SPI_MODE 配置为 Dual、Quad 或 Octal 模式时有效 0x0: STD_SPI_FRF 标准 SPI 帧格式 0x1: Dual_SPI_FRF 双线 SPI 帧格式 0x2: Quad_SPI_FRF 四线 SPI 帧格式 0x3: Octal_SPI_FRF 八线 SPI 帧格式

## 2. 控制寄存器 1 (CTRLR1)

偏移地址: 0x4

NDF	[15:0]	RW
-----	--------	----

数据帧数量, 当 TMOD=2 或 TMOD=3 时, 此寄存器字段设置 spi 连续接收的数据帧数。spi 继续接收串行数据, 直到接收到的数据帧数等于此寄存器值加 1, 这使您能够在连续传输中接收高达 64KB 的数据。当选用 RX ONLY 和 EEPROM READ 模式时 SPI Master 接收数据的个数由寄存器 CTRLR1 控制。

## 3. SSI 使能寄存器 (SSIENR)

偏移地址: 0x8

RW		RW	
MODE_3WIRE		SSI_EN	
[1]		[0]	

bit0	SPI使能: 启用和禁用所有spi操作。禁用后, 所有串行传输将立即停止。当设备被禁用时, 发送和接收 FIFO缓冲区被清除。启用时, 无法对某些spi控制寄存器进行编程。禁用时, 设置ssi_sleep输出(延迟后), 通知系统可以安全地删除ssi_clk, 从而节省系统的功耗。 0x0: 禁用串行传输 0x1: 启用串行传输		
bit1	MODE_3WIRE: SPIM3线选择。 0x0: 四线模式 0x1: 三线模式		

#### 4. 从机选择使能寄存器 (SER)

偏移地址: 0x10

SER	[0]	RW
-----	-----	----

从机选择使能标志。该寄存器中的每个位对应于来自 spi 主机的从属选择线 (ss\_x\_n)。当该寄存器设置为 1 时, 在串行传输开始时, 来自主设备的相应从属选择线被激活。在开始传输之前, 此寄存器中的设置或清除位对相应的从属选择输出没有影响。在开始传输之前, 应该启用此寄存器中与从属设备对应的位。主机想要与之通信的设备。当不在广播模式下操作时, 此字段中只应设置一个位。

0x0 (NOT\_SELECTED): 未选择从属设备; 0x1 (已选择): 已选择从属设备。

#### 5. 波特率 (时钟分频) 寄存器 (BAUDR)

偏移地址: 0x14

SCKDV	[15:0]	RW
-------	--------	----

SSI 时钟分频器。

此字段的 LSB 始终设置为 0, 不受写入操作的影响, 这确保了此寄存器中保持偶数值。如果该值为 0, 则禁用串行输出时钟 (sclk\_out)。sclk\_out 的频率由以下方程式得出:

$$F_{sclkout} = F_{ssiclk} / SCKDV$$

其中 SCKDV 是 2 到 65534 之间的任何偶数值。

#### 6. 发送 FIFO 阈值寄存器 (TXFTLR)

偏移地址：0x18

TXFTLR	[7:0]	RW
--------	-------	----

发送 FIFO 阈值。

控制传输 FIFO 控制器触发中断的条目级别（或以下）。FIFO 深度可在 2-256 范围内配置；该寄存器的大小被设定为访问 FIFO 所需的地址位数。如果您试图将此值设置为大于或等于 FIFO 的深度，则不会写入此字段并保留其当前值。当传输 FIFO 条目数小于或等于此值时，会触发传输 FIFO 空中断。

## 7. 接收 FIFO 阈值寄存器 (RXFTLR)

偏移地址：0x1c

RXFTLR	[7:0]	RW
--------	-------	----

接收 FIFO 阈值。

控制接收 FIFO 控制器触发中断的条目级别（或以上）。FIFO 深度可在 2-256 的范围内配置。该寄存器的大小与访问 FIFO 所需的地址位数相匹配。如果试图将此值设置为大于 FIFO 的深度，则不会写入此字段并保留其当前值。当接收 FIFO 条目数大于或等于 该值+1 时，触发接收 FIFO 满中断。

## 8. 发送 FIFO 当前条目数量寄存器 (TXFLR)

偏移地址：0x20

TXFLR	[15:0]	RO
-------	--------	----

发送 FIFO 深度，实时反映发送 FIFO 中有效数据条目数量，范围：0 至 FIFO 深度。

## 9.接收 FIFO 当前条目数量寄存器 (RXFLR)

偏移地址：0x24

RXFLR	[15:0]	RO
-------	--------	----

接收 FIFO 深度,实时反映接收 FIFO 中有效数据条目数量, 范围：0 至 FIFO 深度。

## 10.状态寄存器 (SR)

偏移地址：0x28



RO	RO	RO	RO	RO	RO	RO
[6]	[5]	[4]	[3]	[2]	[1]	[0]
DCOL	TXE	RFF	RFNE	TFE	TFNF	BUSY

bit0	BUSY: SPI 忙标志 置 1 表示当前正在进行串行传输; 清 0 表示 SPI 处于空闲或被禁用。
bit1	TFNF:发送 FIFO 未满足标志 只要发送 FIFO 还有至少一个空位即置 1; FIFO 满时清 0。
bit2	TFE: 发送 FIFO 空标志 发送 FIFO 完全空时置 1; 只要存在有效条目即清 0。该位不会产生中断请求。
bit3	RFNE: 接收 FIFO 非空标志 只要接收 FIFO 中至少有一个有效条目即置 1; FIFO 为空时清 0。软件可轮询此位以确认是否已读空 FIFO。
bit4	RFF:接收 FIFO 满标志 接收 FIFO 完全满时置 1; 一旦出现空位即清 0。
bit5	TXE:发送错误 仅从机模式有效, 当传输开始时发送 FIFO 为空, 则置 1; 此时 txd 线会重复上一次的数据。读寄存器后自动清 0。
bit6	DCOL: 数据冲突错误 仅主机模式有效。当 DW_apb_ssi 作为主机正在传输时, 若 ss_in_n 被其它主机拉低, 则置1, 表示上一次传输被中断未完成。读寄存器后自动清 0。

## 11.中断屏蔽寄存器 (IMR)

偏移地址: 0x2c

RW	RW	RW	RW	RW	RW
[5]	[4]	[3]	[2]	[1]	[0]
MSTIM	RXFIM	RXOIM	RXUIM	TXOIM	TXEIM

bit0	TXEIM: 发送FIFO空中断屏蔽 0x0: 屏蔽, ssi_txe_intr中断被屏蔽 0x1: 开启此中断
------	--

bit1	TXOIM:发送FIFO溢出中断屏蔽 0x0: 屏蔽, ssi_txo_intr中断被屏蔽 0x1: 开启此中断
bit2	RXUIM: 接收FIFO 下溢中断屏蔽 0x0: 屏蔽, ssi_rxu_intr中断被屏蔽 0x1: 开启此中断
bit3	RXOIM: 接收 FIFO 溢出中断屏蔽 0x0: 屏蔽, ssi_rxo_intr中断被屏蔽 0x1: 开启此中断
bit4	RXFIM:接收 FIFO 满中断屏蔽 0x0: 屏蔽, ssi_rxf_intr中断被屏蔽 0x1: 开启此中断
bit5	MSTIM:多主机竞争中断屏蔽. 如果SPI配置为从机模式, 则该位不存在。 0x0: 屏蔽, ssi_mst_intr中断被屏蔽 0x1: 开启此中断

## 12.中断屏蔽寄存器 (IMR)

偏移地址: 0x30

RO	RO	RO	RO	RO	RO
[5]	[4]	[3]	[2]	[1]	[0]
MSTIS	RXFIS	RXOIS	RXUIS	TXOIS	TXEIS

bit0	TXEIM: 发送FIFO空中断屏蔽 0x0: 屏蔽, ssi_txe_intr中断被屏蔽 0x1: 开启此中断
bit1	TXOIM:发送FIFO溢出中断屏蔽 0x0: 屏蔽, ssi_txo_intr中断被屏蔽 0x1: 开启此中断
bit2	RXUIM: 接收FIFO 下溢中断屏蔽 0x0: 屏蔽, ssi_rxu_intr中断被屏蔽 0x1: 开启此中断
bit3	RXOIM: 接收 FIFO 溢出中断屏蔽 0x0: 屏蔽, ssi_rxo_intr中断被屏蔽 0x1: 开启此中断

bit4	RXFIM:接收 FIFO 满中断屏蔽 0x0: 屏蔽, ssi_rxf_intr中断被屏蔽 0x1: 开启此中断
bit5	MSTIM:多主机竞争中断屏蔽. 如果SPI配置为从机模式, 则该位不存在。 0x0: 屏蔽, ssi_mst_intr中断被屏蔽 0x1: 开启此中断

### 13. 中断状态寄存器 (ISR)

偏移地址: 0x30

RO	RO	RO	RO	RO	RO
[5]	[4]	[3]	[2]	[1]	[0]
MSTIS	RXFIS	RXOIS	RXUIS	TXOIS	TXEIS

bit0	TXEIS:发送FIFO空中断状态位 0x0: 无效, ssi_txe_intr中断无效 (未触发) 0x1: 有效, ssi_txe_intr中断有效 (触发)
bit1	TXOIS:发送FIFO溢出中断状态位 0x0: 无效, ssi_txo_intr中断无效 (未触发) 0x1: 有效, ssi_txo_intr中断有效 (触发)
bit2	RXUIS: 接收FIFO 下溢中断状态位 0x0: 无效, ssi_rxu_intr中断无效 (未触发) 0x1: 有效, ssi_rxu_intr中断有效 (触发)
bit3	RXOIS: 接收 FIFO 溢出中断状态位 0x0: 无效, ssi_rxo_intr中断无效 (未触发) 0x1: 有效, ssi_rxo_intr中断有效 (触发)
bit4	RXFIS:接收 FIFO 满中断状态位 0x0: 无效, ssi_rxf_intr中断无效 (未触发) 0x1: 有效, ssi_rxf_intr中断有效 (触发)
bit5	MSTIS:多主机竞争中断状态位如果SPI配置为从机模式, 则该位不存在。 0x0: 无效, ssi_mst_intr中断无效 (未触发) 0x1: 有效, ssi_mst_intr中断有效 (触发)

### 14. 中断清除寄存器 (ICR)

偏移地址: 0x48

ICR	[0]	RO
-----	-----	----

清除中断。

如果下面的任何中断有效，则该寄存器置位。读操作会清除 ssi\_txo\_intr、ssi\_rxu\_intr、ssi\_rxo\_intr 和 ssi\_mst\_intr 中断。

## 15.数据寄存器（DR16）

偏移地址：0x60

DR16	[15:0]	RW
------	--------	----

数据寄存器。

写入该寄存器时，数据自动右对齐。如果 SSI\_MAX\_XFER\_SIZE 配置参数设置为 32，则所有 32 位都有效。否则，寄存器中只有 16 位([15:0])有效。

## 6.3.3 SPI Slave

### [6.3.3.1] 简介

SPI Slave 被动响应 SPI Master 的通信请求，时钟（SCK）由 Master 提供，仅在 NCS（片选）有效时与 Master 交互数据。BL1820 的 SPI Slave 支持四线、三线、两线模式，适用于作为从设备接收主机控制指令（如传感器数据上传、配置接收）。

### [6.3.3.2]主要特性

特性分类	具体参数/功能
通信模式	全双工四线、半双工四线、半双工三线、半双工两线（无 NCS，片选由寄存器控制）
帧格式	8-bit/16-bit 可选
时钟性能	最高支持 ( $f_{PCLK}/4$ ) (PCLK 为外设时钟，最大 48 MHz，对应最

特性分类	具体参数/功能
	高 12 MHz)
FIFO 缓冲	发送/接收 FIFO 深度均为 16 字节，支持 FIFO 阈值中断
线路灵活性	两线模式下 MOSI/MISO 复用为 DIO，NCS 由内部寄存器控制（无需外部引脚）
中断支持	与 SPI Master 一致（发送空、发送溢出、接收满、接收溢出、接收欠流）

### [6.3.3.3] 功能描述

#### 1. 传输模式

通过 CTRLR0 寄存器配置，支持 3 种传输模式：

- **TX AND RX（全双工）**：接收 Master 数据的同时，向 Master 发送反馈数据；
- **TX ONLY（半双工发送）**：仅向 Master 发送数据，不接收（如传感器主动上传数据）；
- **RX ONLY（半双工接收）**：仅接收 Master 数据，不发送（如接收主机配置指令）。

#### 2. 时钟特性

- **时钟来源**：完全由 SPI Master 提供，Slave 无需配置时钟分频，仅需确保 SCPOL/SCPH 与 Master 一致；
- **采样时序**：与 SPI Master 相同（SCPH=0 在第一个边沿采样，SCPH=1 在第二个边沿采样），需提前与 Master 协商配置。

#### 3. 线路模式

- **四线模式**：MOSI（收）、MISO（发）、SCK（时钟）、NCS（片选），全双工/半双工；
- **三线模式**：MOSI 与 MISO 复用为 DIO 引脚，半双工（发送时用 TX ONLY，接收时用 RX ONLY）；
- **两线模式**：仅使用 SCK 和 DIO，NCS 由 SPIS\_NCS\_CTRL 寄存器控制（1=选中，0=未选中），适用于引脚资源紧张的场景。

## 4. 中断

与 SPI Master 中断类型完全一致，仅触发条件依赖 Master 的通信节奏（如 Master 发送数据时 Slave 接收 FIFO 满触发中断），清除方式相同（读取 ICR 寄存器写 1）。

### [6.3.3.4] 寄存器映射（SPIS）

SPI Slave 基地址：0x41007000，其他有关 SPIS 寄存器配置，在此基地址加上相应的偏移地址。

#### 1. 控制寄存器 0（CTRLR0）

偏移地址：0x0

RO	RO	RO	RW	RW	RW	RW	RW	RW	RW	RW
[24]	[22:21]	[20:16]	[15:12]	[11]	[10]	[9:8]	[7]	[6]	[5:4]	[3:0]
SSTE	SPI_FRF	DFS_32	CFS	SRL	SLV_OE	TMOD	SCPOL	SCPH	FRF	DFS

bit[3:0]	DFS: 数据帧大小 此寄存器字段仅当 SSI_MAX_XFER_SIZE=16 时有效
bit[5:4]	FRF: 帧格式选择。0x0: Motorola SPI协议 0x1: TI SSP协议 0x2: National Microwire协议 0x3: 保留值

bit[6]	SCPH: 串行时钟相位Motorola SPI模式下定义时钟与从机选择信号的时序关系 0x0: SCPH_MIDDLE 数据在首个时钟边沿采样（时钟在首数据位中间翻转） 0x1: SCPH_START 数据在第二个时钟边沿采样（时钟在从机选择激活后延迟一个周期翻转）
bit[7]	SCPOL: 串行时钟极性 Motorola SPI模式下定义空闲时钟电平。 0x0: SCLK_LOW 空闲状态为低电平 0x1: SCLK_HIGH 空闲状态为高电平
bit[9:8]	TMOD: 传输模式。定义数据有效方向 0x0: TX_AND_RX 全双工（TX/RX均有效） 0x1: TX_ONLY 仅发送（忽略RX数据） 0x2: RX_ONLY 仅接收（忽略TX数据） 0x3: EEPROM_READ EEPROM读模式（主机发送控制字后切换为接收，仅主模式有效）
bit[10]	SLV_OE: 从机输出使能仅对从机模式有效 0x0: 允许从机正常输出数据（默认状态） 0x1: 强制ssi_oe_n为高阻态，禁用TXD输出（广播模式下防止总线冲突）
bit[11]	SRL: 移位寄存器环回测试 测试专用。将TX移位寄存器输出直连至RX移位寄存器输入 0x0: NORMAL_MODE 正常操作模式 0x1: TEST_MODE 环回测试模式（需外部提供ss_in_n和ssi_clk信号）

bit[15:12]	CFS: 控制帧长度 (Microwire模式) 选择Microwire帧格式下控制字的位宽
bit[20:16]	DFS_32:32位传输模式下数据帧长度 仅当SSI_MAX_XFER_SIZE=32时有效。接收数据自动右对齐，高位补零至32位
bit[22:21]	SPI_FRF:SPI帧格式选择 仅当SSI_SPI_MODE配置为Dual、Quad或Octal模式时有效 0x0: STD_SPI_FRF 标准SPI帧格式 0x1: Dual_SPI_FRF 双线SPI帧格式 0x2: Quad_SPI_FRF 四线SPI帧格式 0x3: Octal_SPI_FRF 八线SPI帧格式
bit[24]	SSTE:从机选择信号反转使能 在SPI模式下，当时钟相位 (SCPH) 设为0时，该寄存器控制数据帧间从机选择信号线 (ss_*n) 的行为

## 2. SSI 使能寄存器 (SSIENR)

偏移地址: 0x8

RW	RW	RW	RW	RW
NCS_MODE	FRX_MODE	FIFO_EN	MODE_3WIRE	SSI_EN
[5:4]	[3]	[2]	[1]	[0]

bit0	SPI使能: 启用和禁用所有spi操作。禁用后，所有串行传输将立即停止。当设备被禁用时，发送和接收FIFO缓冲区被清除。启用时，无法对某些spi控制寄存器进行编程。禁用时，设置 ssi_sleep输出（延迟后），通知系统可以安全地删除ssi_clk，从而节省系统的功耗。 0x0: 禁用串行传输 0x1: 启用串行传输
bit1	MODE_3WIRE: SPIM3线选择。 0x0: 四线模式 0x1: 三线模式
bit2	FIFO强制使能: 0x0: 当SSI_EN为0，FIFO失能 0x1: 当SSI_EN为1，FIFO使能
bit3	FIFO接收模式控制: 0x0:正常模式 0x1:CPHA=1，快速写入数据以满足3线定时



bit[5:4]	NCS处理模式: 0x0: 正常进行 0x1: 过滤一个高脉冲 0x2: 过滤两个高脉冲 0x3: 过滤一个低脉冲
----------	---

### 3. 发送 FIFO 阈值寄存器 (TXFTLR)

偏移地址: 0x18

TXFTLR	[7:0]	RW
--------	-------	----

发送 FIFO 阈值。

控制传输 FIFO 控制器触发中断的条目级别（或以下）。FIFO 深度可在 2-256 范围内 配置；该寄存器的大小被设定为访问 FIFO 所需的地址位数。如果您试图将此值设置为 大于或等于 FIFO 的深度，则不会写入此字段并保留其当前值。 当传输 FIFO 条目数小于 或等于此值时，会触发传输 FIFO 空中断。

### 4. 接收 FIFO 阈值寄存器 (RXFTLR)

偏移地址: 0x1c

RXFTLR	[7:0]	RW
--------	-------	----

接收 FIFO 阈值。

控制接收 FIFO 控制器触发中断的条目级别（或以上）。FIFO 深度可在 2-256 的范围内配置。该寄存器的大小与访问 FIFO 所需的地址位数相匹配。如果试图将此值设置为 大于 FIFO 的深度，则不会写入此字段并保留其当前值。当接收 FIFO 条目数大于或等于 该值+1 时，触发接收 FIFO 满中断。

### 5. 发送 FIFO 当前条目数量寄存器 (TXFLR)

偏移地址: 0x20

TXFLR	[15:0]	RO
-------	--------	----

发送 FIFO 深度，实时反映发送 FIFO 中有效数据条目数量，范围：0 至 FIFO 深度。

### 6. 接收 FIFO 当前条目数量寄存器 (RXFLR)

偏移地址：0x24

RXFLR	[15:0]	RO
-------	--------	----

接收 FIFO 深度,实时反映接收 FIFO 中有效数据条目数量, 范围：0 至 FIFO 深度。

## 7. 状态寄存器 (SR)

偏移地址：0x28

RO	RO	RO	RO	RO	RO	RO
[6]	[5]	[4]	[3]	[2]	[1]	[0]
DCOL	TXE	RFF	RFNE	TFE	TFNF	BUSY

bit0	BUSY: SPI 忙标志 置 1 表示当前正在进行串行传输；清 0 表示 SPI 处于空闲或被禁用。
bit1	TFNF:发送 FIFO 未满足标志 只要发送 FIFO 还有至少一个空位即置 1；FIFO 满时清 0。
bit2	TFE: 发送 FIFO 空标志 发送 FIFO 完全空时置 1；只要存在有效条目即清 0。该位不会产生中断请求。
bit3	RFNE: 接收 FIFO 非空标志 只要接收 FIFO 中至少有一个有效条目即置 1；FIFO 为空时清 0。软件可轮询此位以确认是否已读空 FIFO。
bit4	RFF:接收 FIFO 满标志 接收 FIFO 完全满时置 1；一旦出现空位即清 0。
bit5	TXE:发送错误 仅从机模式有效，当传输开始时发送 FIFO 为空，则置 1；此时 txd 线会重复上一次的数据。读寄存器后自动清 0。
bit6	DCOL: 数据冲突错误 仅主机模式有效。当 DW_apb_ssi 作为主机正在传输时，若 ss_in_n 被其它主机拉低，则置1，表示上一次传输被中断未完成。读寄存器后自动清 0。

## 8. 中断屏蔽寄存器 (IMR)

偏移地址：0x2c

RW	RW	RW	RW	RW
----	----	----	----	----

[4]	[3]	[2]	[1]	[0]
RXFIM	RXOIM	RXUIM	TXOIM	TXEIM

bit0	TXEIM: 发送FIFO空中断屏蔽 0x0: 屏蔽, ssi_txe_intr中断被屏蔽 0x1: 开启此中断
bit1	TXOIM:发送FIFO溢出中断屏蔽 0x0: 屏蔽, ssi_txo_intr中断被屏蔽 0x1: 开启此中断
bit2	RXUIM: 接收FIFO 下溢中断屏蔽 0x0: 屏蔽, ssi_rxu_intr中断被屏蔽 0x1: 开启此中断
bit3	RXOIM: 接收 FIFO 溢出中断屏蔽 0x0: 屏蔽, ssi_rxo_intr中断被屏蔽 0x1: 开启此中断
bit4	RXFIM:接收 FIFO 满中断屏蔽 0x0: 屏蔽, ssi_rxf_intr中断被屏蔽 0x1: 开启此中断

## 9. 中断状态寄存器 (ISR)

偏移地址: 0x30

RO	RO	RO	RO	RO
[4]	[3]	[2]	[1]	[0]
RXFIS	RXOIS	RXUIS	TXOIS	TXEIS

bit0	TXEIS: 发送FIFO空中断状态位 0x0: 无效, ssi_txe_intr中断无效 (未触发) 0x1: 有效, ssi_txe_intr中断有效 (触发)
bit1	TXOIS:发送FIFO溢出中断状态位 0x0: 无效, ssi_txo_intr中断无效 (未触发) 0x1: 有效, ssi_txo_intr中断有效 (触发)
bit2	RXUIS: 接收FIFO 下溢中断状态位 0x0: 无效, ssi_rxu_intr中断无效 (未触发) 0x1: 有效, ssi_rxu_intr中断有效 (触发)

bit3	RXOIS: 接收 FIFO 溢出中断状态位 0x0: 无效, ssi_rxo_intr中断无效 (未触发) 0x1: 有效, ssi_rxo_intr中断有效 (触发)
bit4	RXFIS:接收 FIFO 满中断状态位 0x0: 无效, ssi_rxf_intr中断无效 (未触发) 0x1: 有效, ssi_rxf_intr中断有效 (触发)

## 10. 中断清除寄存器 (ICR)

偏移地址: 0x48

ICR	[0]	RO
-----	-----	----

清除中断。

如果下面的任何中断有效, 则该寄存器置位。读操作会清除 ssi\_txo\_intr、ssi\_rxu\_intr、ssi\_rxo\_intr 和 ssi\_mst\_intr 中断。

## 11. 数据寄存器 (DR16)

偏移地址: 0x60

DR16	[15:0]	RW
------	--------	----

数据寄存器。

写入该寄存器时, 数据自动右对齐。如果 SSI\_MAX\_XFER\_SIZE 配置参数设置为 32, 则所有 32 位都有效。否则, 寄存器中只有 16 位 ([15:0]) 有效。

## 6.4 GPIO

### 6.4.1 简介

BL1820 内部包含 24 个通用输入/输出口 (GPIO0~GPIO23), 支持数字输入/输出、模拟输入、外设复用, 可配置上拉/下拉、开漏/推挽输出、中断触发方式。

### 6.4.2 主要特性

- 输入模式: 浮空、上拉、下拉、模拟输入;
- 输出模式: 推挽 (4 档驱动能力)、开漏;
- 中断: 支持高电平、低电平、上升沿、下降沿、任意边沿触发;
- 唤醒: 所有 GPIO 可配置为深度休眠模式唤醒源;
- 复用: 支持 UART、SPI、IIC、Timer 等外设复用。

### 6.4.3 功能描述

### [6.4.3.1] 输入配置

- 浮空输入：无上下拉电阻，适用于外部有电平驱动的场景；
- 上拉输入：内部上拉电阻（25 k $\Omega$  典型值），默认电平为高；
- 下拉输入：内部下拉电阻（25 k $\Omega$  典型值），默认电平为低；
- 模拟输入：禁用数字电路，用于 ADC 采样。

### [6.4.3.2] 输出配置

- 推挽输出：可输出高/低电平，驱动能力分 4 档（通过 GPIO\_CFG\_N[4:3] 配置，11>10>01>00）；
- 开漏输出：仅能拉低电平，高电平需外部上拉电阻，适用于 IIC 总线等多设备共享总线场景。

### [6.4.3.3] 中断配置

每个 GPIO 的中断通过以下寄存器配置：

- GPIO\_IT\_HI：高电平中断使能；
- GPIO\_IT\_LO：低电平中断使能；
- GPIO\_IT\_PEDGE：上升沿中断使能；
- GPIO\_IT\_NEDGE：下降沿中断使能；
- GPIO\_INT\_STATUS：中断状态寄存器（读 1 表示对应 GPIO 触发中断）；

## 6.4.4 寄存器映射

GPIO 基地址：0x40010000，其他 GPIO 功能配置寄存器地址加上相应的偏移地址。

### 1. 输入电平状态 (GPIO\_I)

偏移地址：0x20

使用 GPIO 用作数字端口时，通过 GPIO\_I[23:0] 寄存器查看端口输入情况。

对应 bit 为 1 时表示端口输入电平为高电平，为 0 时表示端口输入电平为低电平。

RO	RO	RO	RO	RO	RO	RO	RO
GPIO_I[23]	GPIO_I[22]	GPIO_I[21]	GPIO_I[20]	GPIO_I[19]	GPIO_I[18]	GPIO_I[17]	GPIO_I[16]
RO	RO	RO	RO	RO	RO	RO	RO

GPIO_I[15]	GPIO_I[14]	GPIO_I[13]	GPIO_I[12]	GPIO_I[11]	GPIO_I[10]	GPIO_I[9]	GPIO_I[8]
RO	RO	RO	RO	RO	RO	RO	RO
GPIO_I[7]	GPIO_I[6]	GPIO_I[5]	GPIO_I[4]	GPIO_I[3]	GPIO_I[2]	GPIO_I[1]	GPIO_I[0]

对于不同的 GPIO\_I[x]取值，0: 低电平；1: 高电平，所有引脚均为只读。

## 2. IO 输入/输出方向选择 (GPIO\_OE)

偏移地址：0x30

使用 GPIO 用作数字端口时，通过 GPIO\_OE[23:0]寄存器控制方向。对应 bit 为 1 时端口 方向为输出，为 0 时端口方向为输入。

RW	RW	RW	RW	RW	RW	RW	RW
GPIO_OE[23]	GPIO_OE[22]	GPIO_OE[21]	GPIO_OE[20]	GPIO_OE[19]	GPIO_OE[18]	GPIO_OE[17]	GPIO_OE[16]
RW	RW	RW	RW	RW	RW	RW	RW
GPIO_OE[15]	GPIO_OE[14]	GPIO_OE[13]	GPIO_OE[12]	GPIO_OE[11]	GPIO_OE[10]	GPIO_OE[9]	GPIO_OE[8]
RW	RW	RW	RW	RW	RW	RW	RW
GPIO_OE[7]	GPIO_OE[6]	GPIO_OE[5]	GPIO_OE[4]	GPIO_OE[3]	GPIO_OE[2]	GPIO_OE[1]	GPIO_OE[0]

对于不同的 GPIO\_OE[x]取值为： 0: 使能输出；1: 失能输出

## 3. 输出高低电平选择 (GPIO\_O)

偏移地址：0x40

使用 GPIO 用作数字端口时，通过 GPIO\_O[23:0]寄存器控制端口输出电平。对应 bit 为 1 时端口输出电平为高电平，为 0 时端口输出电平为低电平。

RW	RW	RW	RW	RW	RW	RW	RW
GPIO_O[23]	GPIO_O[22]	GPIO_O[21]	GPIO_O[20]	GPIO_O[19]	GPIO_O[18]	GPIO_O[17]	GPIO_O[16]
RW	RW	RW	RW	RW	RW	RW	RW
GPIO_O[15]	GPIO_O[14]	GPIO_O[13]	GPIO_O[12]	GPIO_O[11]	GPIO_O[10]	GPIO_O[9]	GPIO_O[8]
RW	RW	RW	RW	RW	RW	RW	RW
GPIO_O[7]	GPIO_O[6]	GPIO_O[5]	GPIO_O[4]	GPIO_O[3]	GPIO_O[2]	GPIO_O[1]	GPIO_O[0]

当 GPIO 功能为正常 GPIO，GPIO\_oe 为“1”时，生效。对于不同的 GPIO\_O[x]取值为： 0:低电平；1: 高电平

## 4. 端口配置寄存器 (GPIO\_CFG\_N)

偏移地址：0x200

GPIO 寄存器划分为 6 组，即 GPIO\_CFG\_[0..5]。每个 GPIO\_CFG\_N[32:0]寄存器包含有 4 个 GPIO 端口的配置信息，每个端口信息占 8 个 bit。

[31:24]	[23:16]	[15:8]	[7:0]
gpio3_cfg	gpio2_cfg	gpio1_cfg	gpio0_cfg

对于每个 gpio\_cfg\_0 组中的单个 GPIO 端口来说，8 个 bit 的配置如下：

bit[2:0]	GPIO功能选择：具体使用查看端口复用说明（表1）
bit[4:3]	驱动强度控制： 0x0: 3mA 0x1: 9mA 0x2: 15mA 0x3: 21mA
bit[7:5]	GPIO模式控制： 0x0: 输入浮空 0x1: 上拉输入 0x2: 下拉输入 0x3: 模拟输入 0x4: 开漏模式 其他：高阻抗

注：推挽输出模式：需要端口配置为[7:5]000 后，再配置 GPIO 输出使能寄存器（GPIO\_OE）

## 5. GPIO 使能寄存器(GPIO\_IE)

偏移地址：0x140

gpio_ie	[23:0]	RW
---------	--------	----

描述：通过设置对应位，使能/失能对应引脚[23:0]中断，0:Disable/1:Enable

## 6. GPIO 清除中断寄存器(GPIO\_IS)

偏移地址：0x150

gpio_is	[23:0]	RW
---------	--------	----

描述：清除 GPIO 引脚[23:0]的中断状态，写入“1”清除对应位

## 7. GPIO 屏蔽中断寄存器(GPIO\_MIS)

偏移地址：0x160

gpio_mis	[23:0]	RW1C
----------	--------	------

描述：每个 GPIO 屏蔽中断状态，写入“1”清除相应位

## 8. GPIO 高电平中断寄存器(GPIO\_IT\_HI)

偏移地址：0x170

gpio_it_hi	[23:0]	RW1C
------------	--------	------

描述：启用 GPIO[23:0] 引脚高电平中断，可以与其他中断条件一起设置，以启用多 条件中断

## 9. GPIO 低电平中断寄存器(GPIO\_IT\_LO)

偏移地址：0x180

gpio_it_lo	[23:0]	RW
------------	--------	----

描述：启用 GPIO[23:0] 引脚低电平中断，可以与其他中断条件一起设置，以启用多 条件中断

## 10. GPIO 上升沿中断寄存器(GPIO\_IT\_PEDGE)

偏移地址：0x190

gpio_it_pedge	[23:0]	R/W
---------------	--------	-----

描述：启用 GPIO[23:0] 引脚上升沿中断，可以与其他中断条件一起设置，以启用多 条件中断

## 12. GPIO 下降沿中断寄存器(GPIO\_IT\_NEDGE)

偏移地址：0x1a0

gpio_it_nedge	[23:0]	R/W
---------------	--------	-----

描述：启用 GPIO[23:0] 引脚下降沿中断，可以与其他中断条件一起设置，以启用多 条件中断。

## 12. GPIO 重映射控制寄存器 0(GPIO\_REMAP\_CTRL0)

偏移地址：0x300

RW	RW	RW	RW	RW	RW
[22:20]	[18:16]	[14:12]	[10:8]	[6:4]	[2:0]
qspi0_pin5_sel	qspi0_pin4_sel	qspi0_pin3_sel	qspi0_pin2_sel	qspi0_pin1_sel	qspi0_pin0_sel



bit[2:0]	qspi0_pin0_sel:QSPI0_NCS 引脚功能重映射选择, 取值与 qspi0_pin5_sel 完全一致, 复位默认 0x0
bit[6:4]	qspi0_pin1_sel:QSPI0_SCK 引脚功能重映射选择, 取值与 qspi0_pin5_sel 完全一致, 复位默认 0x1
bit[10:8]	qspi0_pin2_sel:QSPI0_D0 引脚功能重映射选择, 取值与 qspi0_pin5_sel 完全一致, 复位默认 0x2
bit[14:12]	qspi0_pin3_sel:QSPI0_D1 引脚功能重映射选择, 取值与 qspi0_pin5_sel 完全一致, 复位默认 0x3
bit[18:16]	qspi0_pin4_sel:QSPI0_D2 引脚功能选择, 取值与 qspi0_pin5_sel 完全一致, 复位默认 0x4
bit[22:20]	qspi0_pin5_sel:QSPI0_D3 引脚功能选择 0x0:映射为 QSPI_NCS 0x1: 映射为 QSPI_SCK 0x2: 映射为 QSPI_D0 0x3: 映射为 QSPI_D1 0x4: 映射为 QSPI_D2 0x5: 映射为 QSPI_D3 (复位默认值, 正常使用)

### 13. GPIO 重映射控制寄存器 1(GPIO\_REMAP\_CTRL1)

偏移地址: 0x304

RW	RW	RW	RW	RW	RW
[31:30]	[29:28]	[27:26]	[25:24]	[1]	[0]
spis0_pin3_sel	spis0_pin2_sel	spis0_pin1_sel	spis0_pin0_sel	i2c0_pin1_sel	i2c0_pin0_sel

bit[0]	i2c0_pin0_sel:I2C0_SCL 引脚功能重映射选择 0x0: 将该引脚配置为 I2C0_SCL 功能 (复位默认值) 0x1: 将该引脚配置为 I2C0_SDA 功能
bit[1]	i2c0_pin1_sel:I2C0_SDA 引脚功能重映射选择器 0x0: 将该引脚配置为 I2C0_SCL 功能 0x1: 将该引脚配置为 I2C0_SDA 功能 (复位默认值)
bit[25:24]	spis0_pin0_sel:SPIS0_NCS 引脚功能重映射选择, 取值与 spis0_pin3_sel 完全一致, 复位默认0x0
bit[27:26]	spis0_pin1_sel:SPIS0_SCK 引脚功能重映射选择, 取值与 spis0_pin3_sel 完全一致, 复位默认0x1
bit[29:28]	spis0_pin2_sel:SPIS0_MOSI 引脚功能重映射选择, 取值与 spis0_pin3_sel 完全一致, 复位默认0x2

bit[31:30]	spis0_pin3_sel:SPIS0_MISO 引脚功能重映射选择 0x0: 配置为 SPIS0_NCS 功能 0x1: 配置为 SPIS0_SCK 功能 0x2: 配置为 SPIS0_MOSI 功能 0x3: 配置为 SPIS0_MISO 功能（复位默认值）
------------	--

## 6.5 UART

### 6.5.1 简介

BL1820 包含的 UART 支持异步全双工串行通信，用于与 PC、模块等外设通信，最高波特率 2 Mbps。

### 6.5.2 主要特性

- 波特率：1200 ~ 2 Mbps（可配置，时钟源可选 RC\_24M、XTAL\_24M、DPLL\_48M）；
- 数据格式：8/9 位数据位、1/1.5/2 位停止位、奇偶校验（奇/偶/无）；
- FIFO：发送/接收 FIFO 深度均为 16 字节；
- 错误检测：溢出错误（OE）、帧错误（FE）、奇偶错误（PE）；
- 中断：发送空、接收满、线路错误、BUSY 中断。

### 6.5.3 功能描述

#### [6.5.3.1] 波特率计算

UART 波特率公式：

$$Baudrate = f_{uartclk} / (16 * DIVISOR)$$

- (f\_{uartclk}): UART 参考时钟（RC\_24M/XTAL\_24M 支持最高 1.5 Mbps，DPLL\_48M 支持最高 2 Mbps）；
- DIVISOR：分频系数（整数部分 DIVN+小数部分 DLF/16，DIVN 由 DLH（高 8 位）和 DLL（低 8 位）控制，DLF 由 DLF 寄存器控制，精度 1/16）。

示例：配置波特率 115200，时钟 48 MHz

1. (DIVISOR = 48000000 / (16 \* 115200) ≈ 26.0417)；
2. DIVN=26 (DLH=0x00, DLL=0x1A), DLF=0.0417\*16≈1 (DLF=0x01)；
3. 实际波特率=48000000/(16\*(26+1/16))≈114942.5 bps（误差<0.2%）。

### [6.5.3.2] 硬件自动流控

- **Auto RTS**: 接收 FIFO 数据量达到阈值（1/4/8/14 字节，FCR[7:6]配置）时，RTS 输出高电平，通知发送方暂停发送；FIFO 空时，RTS 输出低电平，允许继续发送；
- **Auto CTS**: CTS 输入高电平时，禁用 UART 发送器；CTS 输入低电平时，恢复发送。

### [6.5.3.3] 中断

UART 支持以下四种中断，可通过配置寄存器 IER 启用：

- **发送空中断**：当 UART 发送 FIFO 里没有数据时触发该中断。清除方法：读取寄存器 IIR 或者往 FIFO 里写入数据使其电平大于发送阈值。
- **接收满中断**：当 UART 接收 FIFO 里的数据个数大于阈值时触发该中断。清除方法：当没有启用 FIFO 时读取 RBR 寄存器；当启用 FIFO 时读取 FIFO 使其电平低于阈值。
- **线路错误检测中断**：当接收线路上的数据存在帧错误、校验错误或者有溢出时将会触发该中断。清除方法：读取寄存器 RBR 或者 LSR。
- **BUSY 中断**：在 UART 进行通信时设置寄存器 LCR 将会触发该中断。清除方法：读取寄存器 USR。

### 6.5.4 寄存器映射 (UART0)

UART0 基地址：0x41001000，其他 UART0 功能配置寄存器地址加上相应的偏移地址。同样地，UART1 基地址：0x41002000，其他 UART1 功能配置寄存器地址加上相应的偏移地址。

#### 1. 接收缓冲寄存器(RBR)

偏移地址：0x0

RBR	[8:0]	RO
-----	-------	----

该寄存器包含 UART 模式下串行输入端口 (sin) 或红外模式下串行红外输入 (sir\_i n) 接收到的数据字节。仅当线路状态寄存器 (LSR) 中的数据就绪 (DR) 位设置时，此寄存器中的数据才有效。只有当 DLAB 位 (LCR[7]) 被清除时，才能访问此寄存器。当 9BIT\_DATA\_EN = 0 时，字段宽度为 8。9BIT\_DATA\_EN = 1，字段宽度为 9。

#### 2. 发送保持寄存器(THR)

偏移地址：0x0

THR	[8:0]	WO
-----	-------	----

该寄存器包含在 UART 模式下通过串行输出端口 (sout) 传输的数据，或在红外模式下通过红外串行输出 (sir\_out\_n) 传输的信息。只有当设置了 THRE 位 (LSR[5]) 时，才应将数据写入 THR。只有当 DLAB 位 (LCR[7]) 被清除时，才能访问此寄存器。当 9BIT\_DATA\_EN = 0 时，字段宽度为 8。9BIT\_DATA\_EN = 1，字段宽度为 9。

#### 3. 除数锁存寄存器(整数部分低 8 位) (DLL)

偏移地址：0x0

DLL	[7:0]	RW
-----	-------	----

该寄存器构成了一个 16 位读/写除数锁存寄存器的低 8 位，包含 UART 的波特率除数。如果 UART\_16550\_COMPATIBLE = No，则仅当 DLAB 位 (LCR[7]) 被设置且 UART 不忙 (即 USR[0]为 0) 时，才能访问此寄存器；否则，仅仅需 DLAB 位 (LCR[7]) 被设置时，才能访问此寄存器。

#### 4. 除数锁存寄存器(整数部分高 8 位) (DLH)

偏移地址：0x0

DLH	[7:0]	RW
-----	-------	----

该寄存器构成了一个 16 位读/写除数锁存寄存器的高 8 位，包含 UART 的波特率除数。如果 UART\_16550\_COMPATIBLE = No，则仅当 DLAB 位（LCR[7]）被设置且 UART 不忙（即 USR[0]为 0）时，才能访问此寄存器；否则，仅仅需 DLAB 位（LCR[7]）被设置时，才能访问此寄存器。

#### 5. 中断使能寄存器(IER)

偏移地址：0x4

RW	#	#	RO	RW	RW	RW	RW
[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
PTME	保留	保留	ELCOLR	EDSSI	ELSI	ETBEI	ERBFI

bit0	<p>ERBFI: 启用接收数据可用中断 用于启用或禁用接收数据可用中断和字符超时中断的产生（如果处于FIFO模式且FIFO已启用）。属于优先级第二高的中断。 0x0: 禁用接收数据中断 0x1: 启用接收数据中断</p>
bit1	<p>ETBEI: 启用发送保持寄存器空中断 用于启用或禁用发送保持寄存器空中断的产生。属于优先级第三高的中断。 0x0: 禁用发送空中断 0x1: 启用发送空中断</p>
bit2	<p>ELSI: 启用接收数据可用中断 用于启用或禁用接收数据可用中断的产生，以及字符超时中断的产生（如果处于 FIFO 模式且 FIFO 已启用）。属于优先级第二高的中断。 0x0: 禁用接收数据中断 0x1: 启用接收数据中断</p>
bit3	<p>EDSSI: 启用调制解调器状态中断 用于启用或禁用调制解调器状态中断的产生。属于优先级第四高的中断。 0x0: 禁用调制解调器状态中断 0x1: 启用调制解调器状态中断</p>

bit4	<p>ELCOLR: 中断使能寄存器 用于控制清除LSR寄存器状态的方法。这仅适用于以下状态位：溢出错误（Overrun Error） 奇偶校验错误（Parity Error）帧错误（Framing Error） 中断状态位（Break Interrupt） 仅当LSR_STATUS_CLEAR == Enabled时可写，始终可读。 0x0：禁用ALC 0x1：启用ALC</p>
bit7	<p>PTME: 可编程THRE中断模式使能 用于启用或禁用THRE中断的产生。仅当THRE_MODE_USER == Enabled时可写，始终可读。 0x0：禁用可编程THRE中断模式 0x1：启用可编程THRE中断模式</p>

## 6. 中断识别寄存器(IIR)

偏移地址：0x8

RO	RO
[7:6]	[3:0]
FIFOSE	IID

bit[3:0]	<p>FIFOSE: 用于查看FIFO是否已启用。 0x0：FIFO禁用 0x3：FIFO启用</p>
bit[7:6]	<p>IID: 用于查看当前最高优先级的挂起中断类型，具体类型在取值中列出。第3位表示该中断仅在FIFO启用时才会发生，用于区分字符超时条件中断。 0x0：调制解调器状态中断 0x1：无挂起中断 0x2：发送保持寄存器（THR）为空 0x4：接收数据可用 0x6：接收器线路状态中断 0x7：繁忙检测中断 0xC：字符超时中断</p>

## 7. FIFO 控制寄存器(FCR)

偏移地址：0x8

WO	RW	WO	WO	WO	WO
[7:6]	[5:4]	[3]	[2]	[1]	[0]
RT	TET	DMAM	XFIFOR	RFIFOR	FIFOE

bit0	<p><b>FIFOE: FIFO使能</b> 用于启用或禁用发送（XMIT）和接收（RCVR）FIFO。每当该位的值发生变化时，发送和接收FIFO的控制器部分都会被复位。</p> <p>0x0：FIFO禁用 0x1：FIFO启用</p>
bit1	<p><b>RFIFOR: 接收FIFO复位</b>用于复位接收FIFO的控制部分，并将FIFO视为“空”。当选择额外的DMA握手信号（DMA_EXTRA == YES）时，它还会解除DMA RX请求信号和单一信号的断言。注意，该位是“自清除”的，无需手动清除。</p> <p>0x1：接收FIFO复位</p>
bit2	<p><b>XFIFOR: 发送FIFO复位</b>用于复位发送FIFO的控制部分，并将FIFO视为“空”。当选择额外的DMA握手信号（DMA_EXTRA == YES）时，它还会解除DMA TX请求信号和单一信号的断言。注意，该位是“自清除”的，无需手动清除。</p> <p>0x1：发送FIFO复位</p>
bit3	<p><b>DMAM: DMA模式</b> 该字段决定了在未选择额外DMA握手信号（DMA_EXTRA == NO）时，dma_tx_req_n和 dma_rx_req_n输出信号所使用的DMA信号模式。</p> <p>0x0：模式0 0x1：模式1</p>
bit[5:4]	<p><b>TET: 发送FIFO空触发器</b> 当THRE_MODE_USER禁用时，对该字段的写操作无效。该字段用于选择发送FIFO的空阈值电平，当模式激活时，达到该电平时将产生THRE中断。此外，在某些工作模式下，它还决定何时断言dma_tx_req_n信号。</p> <p>0x0：FIFO为空 0x1：FIFO中有2个字符 0x2：FIFO 1/4满 0x3：FIFO 1/2满</p>

bit[7:6]	<b>RT: 接收器FIFO触发器</b> 用于选择接收FIFO中的触发电平，当达到该电平时将产生接收数据可用中断。在自动流量控制模式下，仅当RTC_FCT禁用时，该字段用于决定何时解除rts_n信号的断言。此外，在某些工作模式下，它还决定何时断言dma_rx_req_n信号。 0x0: FIFO中有1个字符 0x1: FIFO 1/4满 0x2: FIFO 1/2满 0x3: FIFO接近满（差2个字符）
----------	--

## 8. 线路控制寄存器(LCR)

偏移地址：0xC

RW	RW	RW	RW	RW	RW	RW
[7]	[6]	[5]	[4]	[3]	[2]	[1:0]
DLAB	BC	SP	EPS	PEN	STOP	DLS

bit[1:0]	<b>DLS: 数据长度选择</b> 如果UART_16550_COMPATIBLE == NO，则仅在UART空闲（USR[0]为0）时可写；否则，始终可写且始终可读。当LCR_EXT中的DLS_E设置为0时，该寄存器用于选择每个字符传输和接收的数据位数。 0x0: 每个字符5个数据位 0x1: 每个字符6个数据位 0x2: 每个字符7个数据位 0x3: 每个字符8个数据位
bit2	<b>STOP: 停止位数量</b> 如果UART_16550_COMPATIBLE == NO，则仅在UART不忙（USR[0]为0）时可写；否则，始终可写且始终可读。该位用于选择每个字符传输和接收的停止位数量。如果设置为0，则传输1个停止位。如果设置为1且数据位为5（LCR[1:0]设置为0），则传输1.5个停止位；否则，传输2个停止位。注意，无论选择多少个停止位，接收器仅检查第一个停止位。 0x0: 1个停止位 0x1: 当数据长度（LCR[1:0]）为0时，传输1.5个停止位，否则传输2个停止位
bit3	<b>PEN: 奇偶校验使能</b> 如果UART_16550_COMPATIBLE == NO，则仅在UART不忙（USR[0]为0）时可写；否则，始终可写且始终可读。该位用于启用或禁用传输和接收串行字符的奇偶校验生成和检测。 0x0: 禁用奇偶校验 0x1: 启用奇偶校验



bit4	<p>EPS: 偶校验选择</p> <p>如果UART_16550_COMPATIBLE == NO, 则仅在UART不忙 (USR[0]为0) 时可写; 否则, 始终可写且始终可读。该位用于选择偶校验或奇校验 (当PEN设置为1时启用)。如果设置为1, 则传输或检查偶数个逻辑“1”; 如果设置为0, 则传输或检查奇数个逻辑“1”。</p> <p>0x0: 传输或检查奇校验</p> <p>0x1: 传输或检查偶校验</p>
bit5	<p>SP: 强制奇偶校验</p> <p>如果UART_16550_COMPATIBLE == NO, 则仅在UART不忙 (USR[0]为0) 时可写; 否则, 始终可写且始终可读。该位用于强制奇偶校验值。当PEN、EPS和强制奇偶校验都设置为1时, 奇偶校验位将被传输和检查为逻辑0。如果PEN和强制奇偶校验设置为1, 而EPS为逻辑0, 则奇偶校验位将被传输和检查为逻辑1。如果该位设置为0, 则禁用强制奇偶校验。</p> <p>0x0: 强制奇偶校验 禁用</p> <p>0x1: 强制奇偶校验 使能</p>
bit6	<p>BC: 断点控制位用于向接收设备发送断点条件。如果设置为1, 则串行输出被强制为低电平 (逻辑0) 状态。当未处于环回模式 (由MCR[4]决定) 时, sout线将被强制拉低, 直到清除断点位。如果启用了SIR模式且处于活动状态 (MCR[6]设置为1), 则sir_out_n线将连续脉冲输出。在环回模式下, 断点条件将内部环回至接收器, 并且sir_out_n线被强制拉低。</p> <p>0x0: 串行输出释放用于数据传输</p> <p>0x1: 串行输出被强制为低电平状态</p>
bit7	<p>DLAB: 除数锁存器访问位如果UART_16550_COMPATIBLE == NO, 则仅在UART不忙 (USR[0]为0) 时可写; 否则, 始终可写且始终可读。该位用于启用对除数锁存寄存器 (DLL和DLH/LPDLL和 LPDLH) 的读写操作, 以设置UART的波特率。在初始波特率设置完成后, 必须清除该位, 才能访问其他寄存器。</p> <p>0x0: 仅在UART不忙时, 除数锁存寄存器可写</p> <p>0x1: 除数锁存寄存器始终可读可写</p>

## 9. 调制解调寄存器(MCR)

偏移地址: 0x10

RW	RW	RW	RW	RW	RW
[5]	[4]	[3]	[2]	[1]	[0]
AFCE	LoopBack	OUT2	OUT1	RTS	DTR

bit0	<p><b>DTR: 数据就绪控制</b> 用于直接控制数据终端就绪 (dtr_n) 输出。该输出用于通知调制解调器或数据集, UART已准备好建立通信。 注意, 在环回模式下 (MCR[4]设置为1), dtr_n输出将被保持在非活动高电平, 而该位置的值将被内部环回到输入。 0x0: dtr_n 解除有效 (逻辑 1) 0x1: dtr_n 有效 (逻辑 0)</p>
bit1	<p><b>RTS: 请求发送 (RTS) 控制直接控制 RTS 输出信号:</b> 非自动流控: MCR[1]=1 时 rts_n=0 自动流控: rts_n 受接收 FIFO 阈值门控 (高于阈值时 rts_n=1) 0x0: rts_n 解除有效 (逻辑 1) 0x1: rts_n 有效 (逻辑 0)</p>
bit2	<p><b>OUT1: 用户指定的Output1 (out1_n) 输出控制</b> 用于直接控制用户指定的Output1 (out1_n) 输出。写入该位置的值将被反转并输出到 out1_n。注意, 在环回模式下 (MCR[4]设置为1), out1_n输出将被保持在非活动高电平, 而该位置的值将被内部环回到输入。 0x0: out1_n 解除有效 (逻辑 1) 0x1: out1_n 有效 (逻辑 0)</p>
bit3	<p><b>OUT2: 用户指定的Output2 (out2_n) 输出控制</b> 用于直接控制用户指定的Output2 (out2_n) 输出。写入该位置的值将被反转并输出到 out2_n。注意, 在环回模式下 (MCR[4]设置为1), out2_n输出将被保持在非活动高电平, 而该位置的值将被内部环回到输入。 0x0: out2_n 解除有效 (逻辑 1) 0x1: out2_n 有效 (逻辑 0)</p>
bit4	<p><b>LoopBack: 回环模式使能</b> 用于将UART置于诊断模式, 以便进行测试。如果在UART模式下运行 (SIR_MODE != Enabled 或未激活, 即MCR[6]设置为0), sout线上的数据将被拉高, 而串行数据输出将被内部环回到sin线。在这种模式下, 所有中断功能正常。此外, 在环回模式下, 调制解调器控制输入 (dsr_n、cts_n、ri_n、dcd_n) 将被断开, 而调制解调器控制输出 (dtr_n、rts_n、out1_n、out2_n) 将被内部环回到输入。 0x0: 环回模式禁用 0x1: 环回模式启用</p>
bit5	<p><b>AFCE: 自动控制流使能</b> 仅当 AFCE_MODE == Enabled 时可写, 始终可读。当 FIFO 使能且 AFCE 置位时, 自动流控功能激活。 0x0: 自动流控禁用 0x1: 自动流控使能</p>

## 10. 线路状态寄存器(LSR)

偏移地址：0x14

RO	RO	RO	RO	RO	RO	RO	RO	RO
[8]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
ADDR_RCVD	RFE	TEMT	THRE	BI	FE	PE	OE	DR

bit0	<p>DR: 数据就绪控制, 用于指示接收器中是否至少有一个字符在RBR或接收FIFO中。</p> <p>在非FIFO模式下, 当读取RBR时, 该位被清除; 在FIFO模式下, 当接收FIFO为空时, 该位被清除。</p> <p>0x0: 数据未就绪</p> <p>0x1: 数据就绪</p>
bit1	<p>OE: 溢出错误位, 用于指示是否发生溢出错误。</p> <p>如果在非FIFO模式下, 当接收器接收到一个新字符而前一个字符尚未从RBR中读取时, 该位被设置, 此时RBR中的数据会被覆盖。</p> <p>在FIFO模式下, 当FIFO已满且接收器接收到一个新字符时, 会发生溢出错误。此时, FIFO中的数据会被保留, 而接收移位寄存器中的数据会丢失。</p> <p>读取LSR寄存器会清除该位 (如果LSR_STATUS_CLEAR == 1), 或者读取LSR或RBR寄存器会清除该位 (如果LSR_STATUS_CLEAR == 0)。</p> <p>0x0: 无溢出错误</p> <p>0x1: 溢出错误</p>
bit2	<p>PE: 奇偶校验错误位如果启用了奇偶校验 (PEN位, 即LCR[3]被设置), 该位用于指示接收器中是否发生奇偶校验错误。</p> <p>在FIFO模式下, 奇偶校验错误与接收到的字符相关联, 当带有奇偶校验错误的字符到达FIFO顶部时, 该错误会被揭示。</p> <p>读取LSR寄存器会清除该位 (如果LSR_STATUS_CLEAR == 1), 或者读取LSR或RBR寄存器会清除该位 (如果LSR_STATUS_CLEAR == 0)。</p> <p>0x0: 无奇偶校验错误</p> <p>0x1: 奇偶校验错误</p>
bit3	<p>FE: 帧错误位, 用于指示接收器中是否发生帧错误。</p> <p>在FIFO模式下, 帧错误与接收到的字符相关联, 当带有帧错误的字符位于FIFO顶部时, 该错误会被揭示。当帧错误发生时, UART会尝试重新同步, 它假设错误是由于下一个字符的起始位引起的, 然后继续接收其他位 (数据、奇偶校验和停止位)。</p> <p>读取LSR寄存器会清除该位 (如果LSR_STATUS_CLEAR == 1), 或者读取LSR或RBR寄存器会清除该位 (如果LSR_STATUS_CLEAR == 0)。</p> <p>0x0: 无帧错误</p> <p>0x1: 帧错误</p>

bit4	<p>BI: 通信中断状态，用于指示在串行输入数据中检测到中断序列。</p> <p>如果处于UART模式，当串行输入（sin）保持在逻辑“0”状态的时间超过起始位 + 数据位 + 奇偶校验位 + 停止位的总和时，该位被设置。</p> <p>在FIFO模式下，与中断条件相关的字符会通过FIFO传输，并在该字符位于FIFO顶部时被揭示。读取LSR寄存器会清除该位（如果LSR_STATUS_CLEAR == 1），或者读取LSR或RBR寄存器会清除该位（如果LSR_STATUS_CLEAR == 0）。在非FIFO模式下，该中断指示会立即发生，并持续到LSR寄存器被读取为止。</p> <p>0x0: 未检测到中断序列 0x1: 检测到中断序列</p>
bit5	<p>THRE: 发送保持寄存器空位如果THRE_MODE_USER禁用或THRE模式禁用（IER[7]设置为0），无论是否实现/启用FIFO，该位指示发送保持寄存器（THR）或发送FIFO是否为空。每当数据从THR或发送FIFO传输到发送移位寄存器且没有新数据写入THR或发送FIFO时，该位被设置。如果启用了THRE中断，则会触发THRE中断。如果THRE_MODE_USER启用且FIFO模式启用（IER[7]设置为1且FCR[0]设置为1），则该位的功能切换为指示发送FIFO是否已满，不再控制THRE中断，而是由FCR[5:4]的阈值设置控制。</p> <p>0x0: THRE中断控制禁用 0x1: THRE中断控制启用</p>
bit6	<p>TEMT: 发送器空位如果处于FIFO模式（FIFO_MODE != NONE且FCR[0]设置为1），则当发送移位寄存器和FIFO均为空时，该位被设置。如果处于非FIFO模式或FIFO未启用，则当发送保持寄存器和发送移位寄存器均为空时，该位被设置。</p> <p>0x0: 发送器不为空 0x1: 发送器为空</p>
bit7	<p>RFE: 接收FIFO错误位仅在FIFO模式启用（FIFO_MODE != NONE且FCR[0]设置为1）时相关。该位用于指示FIFO中是否存在至少一个奇偶校验错误、帧错误或中断指示。</p> <p>当LSR寄存器被读取且带有错误的字符位于接收FIFO顶部且FIFO中没有后续错误时，该位被清除。</p> <p>0x0: RX FIFO中无错误 0x1: RX FIFO中存在错误</p>
bit8	<p>ADDR_RCVD: 地址接收位如果启用了9位数据模式（LCR_EXT[0] = 1），该位用于指示接收到的数据的第9位是否被设置为1。此位还可以用于指示接收到的字符是地址还是数据：</p> <p>1: 表示该字符是地址 0: 表示该字符是数据</p> <p>在FIFO模式下，由于第9位与接收到的字符相关联，因此当第9位被设置为1的字符位于FIFO顶部时，该位会被揭示。</p> <p>读取LSR寄存器会清除该位。</p>

## 11. 调制解调状态寄存器(MSR)

偏移地址: 0x18

RO	RO	RO	RO	RO	RO	RO	RO
[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
DCD	RI	DSR	CTS	DDCD	TERI	DDSR	DCTS

bit0	<p>DCTS: 清除发送变化量。</p> <p>自上次读取 MSR 后 <code>cts_n</code> 发生状态变化时置位, 读取 MSR 自动清零。回环模式: 反映 MCR[1] (RTS) 变化</p> <p>0x0: 自上次读取MSR以来, <code>dscr_n</code>未发生变化</p> <p>0x1: 自上次读取MSR以来, <code>dscr_n</code>发生了变化</p>
bit1	<p>DDSR: 数据设备就绪变化量。</p> <p>用于指示自上次读取MSR寄存器以来, 调制解调器控制线<code>dscr_n</code>是否发生了变化。读取MSR寄存器会清除DDSR位。在环回模式下 (MCR[4]设置为1), DDSR反映MCR[0] (DTR) 的变化。</p> <p>0x0: 自上次读取MSR以来, <code>dscr_n</code>未发生变化</p> <p>0x1: 自上次读取MSR以来, <code>dscr_n</code>发生了变化</p>
bit2	<p>TERI: 振铃指示尾沿</p> <p>用于指示自上次读取MSR寄存器以来, <code>ri_n</code>输入是否从低电平 (活动) 变为高电平 (非活动)。</p> <p>读取MSR寄存器会清除TERI位。在环回模式下 (MCR[4]设置为1), TERI反映MCR[2] (Out1) 从高电平变为低电平的变化</p> <p>0x0: 自上次读取MSR以来, <code>ri_n</code>未发生变化</p> <p>0x1: 自上次读取MSR以来, <code>ri_n</code>发生了变化</p>
bit3	<p>DDCD: 载波检测变化指示。</p> <p>用于指示自上次读取MSR寄存器以来, 调制解调器控制线<code>dcd_n</code>是否发生了变化。</p> <p>读取MSR寄存器会清除DDCD位。在环回模式下 (MCR[4]设置为1), DDCD反映MCR[3] (Out2) 的变化。</p> <p>0x0: 自上次读取MSR以来, <code>dcd_n</code>未发生变化</p> <p>0x1: 自上次读取MSR以来, <code>dcd_n</code>发生了变化</p>
bit4	<p>CTS: 清除发送状态。</p> <p>用于指示调制解调器控制线<code>cts_n</code>的当前状态。该位是<code>cts_n</code>的反相。当<code>cts_n</code>输入被断言 (即逻辑0) 时, 表示调制解调器或数据集已准备好与UART交换数据。在环回模式下 (MCR[4]设置为1), CTS与MCR[1] (RTS) 相同。</p> <p>0x0: <code>cts_n</code> 无效 (逻辑1)</p> <p>0x1: <code>cts_n</code> 有效 (逻辑0)</p>

bit5	<p>DSR: 数据设备就绪状态</p> <p>用于指示调制解调器控制线ri_n的当前状态。该位是ri_n的反相。当ri_n输入被断言（即逻辑1）时，表示调制解调器或数据集已接收到电话振铃信号。在环回模式下（MCR[4]设置为1，RI与MCR[2]（Out1）相同）。</p> <p>0x0: dsr_n 无效（逻辑1）</p> <p>0x1: dsr_n 有效（逻辑0）</p>
bit6	<p>RI: 振铃指示</p> <p>如果处于FIFO模式（FIFO_MODE != NONE且FCR[0]设置为1），则当发送移位寄存器和FIFO均为空时，该位被设置。如果处于非FIFO模式或FIFO未启用，则当发送保持寄存器和发送移位寄存器均为空时，该位被设置。</p> <p>0x0: ri_n 无效（逻辑1）</p> <p>0x1: ri_n 有效（逻辑0）</p>
bit7	<p>DCD: 数据载波检测状态用于指示调制解调器控制线dcd_n的当前状态。该位是dcd_n的反相。当dcd_n输入被断言（即逻辑0）时，表示调制解调器或数据集已检测到载波。在环回模式下（MCR[4]设置为1），DCD与MCR[3]（Out2）相同。</p> <p>0x0: dcd_n 无效（逻辑1）</p> <p>0x1: dcd_n 有效（逻辑0）</p>

## 12. UART 状态寄存器(USR)

偏移地址: 0x7C

RW	RW	RW	RW	RW	RW
[22:20]	[18:16]	[14:12]	[10:8]	[6:4]	[2:0]
qspi0_pin5_sel	qspi0_pin4_sel	qspi0_pin3_sel	qspi0_pin2_sel	qspi0_pin1_sel	qspi0_pin0_sel

bit0	<p>BUSY: UART忙.</p> <p>仅当UART_16550_COMPATIBLE == NO时有效。该位用于指示UART是否正在进行串行传输。当该位被清除时，表示UART处于空闲或非活动状态。</p> <p>该位在以下任何一种情况下被设置为1（忙）：</p> <ul style="list-style-type: none"> <li>串行接口正在进行传输</li> <li>发送保持寄存器（THR）中有数据，且未使用FIFO访问模式（FAR = 0），波特率除数非零（{DLH, DLL}不为0），且除数锁存器访问位为0（LCR.DLAB = 0）</li> <li>接口正在进行接收</li> <li>接收缓冲寄存器（RBR）中有数据，且未使用FIFO访问模式（FAR = 0）</li> </ul> <p>0x0: UART空闲或非活动</p> <p>0x1: UART忙（正在传输数据）</p>
------	--



bit1	<p>TFNF: 发送FIFO未满。 仅当FIFO_STAT == YES时有效。该位用于指示发送FIFO是否未满。当TX FIFO满时, 该位被清除。 0x0: 发送FIFO满 0x1: 发送FIFO未满</p>
bit2	<p>TFE: 发送FIFO空 仅当FIFO_STAT == YES时有效。该位用于指示发送FIFO是否完全空。当TX FIFO不再空时, 该位被清除。 0x0: 发送FIFO非空 0x1: 发送FIFO空</p>
bit3	<p>RFNE: 接收FIFO非空。 仅当FIFO_STAT == YES时有效。该位用于指示接收FIFO中是否包含一个或多个条目。当RX FIFO为空时, 该位被清除。 0x0: 接收FIFO为空 0x1: 接收FIFO非空</p>
bit4	<p>RFF: 接收FIFO满。 仅当FIFO_STAT == YES时有效。该位用于指示接收FIFO是否完全满。当RX FIFO不再满时, 该位被清除。 0x0: 接收FIFO未满 0x1: 接收FIFO满</p>

### 13. 传输控制寄存器(TCR)

偏移地址: 0xAC

RW	RW	RW	RW
[4:3]	[2]	[1]	[0]
XFER_MODE	DE_POL	RE_POL	RS485_EN

bit0	<p>RS485_EN: RS485传输使能, 该字段用于启用或禁用RS485模式。 0x0: RS232模式, 该寄存器中的其他字段以及DE_EN、RE_EN和TAT寄存器均保留 0x1: RS485模式, 该寄存器中的其他字段均适用。</p>
bit1	<p>RE_POL: 接收使能极性, 该字段用于定义RE信号的极性。 仅当FIFO_STAT == YES时有效。该位用于指示发送FIFO是否未满。当TX FIFO满时, 该位被清除。 0x0: RE信号为低电平有效 0x1: RE信号为高电平有效</p>

bit2	DE_POL: 驱动使能极性, 该字段用于定义DE信号的极性: 0x0: DE信号为低电平有效 0x1: DE信号为高电平有效
bit[4:3]	XFER_MODE: 传输模式。 该字段用于定义UART的传输模式, 具体模式如下: 模式0: 全双工模式; 发送和接收可以同时进行。用户可以在任何时间点启用DE_EN、RE_EN。在此模式下, TAT寄存器中编程的翻转时序不适用。 模式1: 半双工基本模式; DE/RE 互斥使能; 只会启用DE或RE中的其中一个, 通过编程实现。硬件在从接收端(RE)切换到发送端(DE)或从发送端切换到接收端时, 会考虑在TAT寄存器中编程设置的周转时间。对于传输, 如果硬件正在接收任何传输数据, 则会在开始传输之前等待。 模式2: 半双工自动模式; 默认 RE 使能(接收态) TX FIFO 写入 → 自动切 DE(发送态) TX FIFO 空 → 自动切 RE(接收态) DE/RE 严格互补输出 0x0: 模式0 0x1: 模式1 0x2: 模式2

#### 14. 驱动使能寄存器(DE\_EN)

偏移地址: 0xB0

DE_EN	[0]	RW
-------	-----	----

描述: 驱动器使能控制, 本寄存器位直接控制 DE(驱动器使能)信号的断言/解除断言。写 0: 解除断言 DE 信号(驱动器禁用)写 1: 断言 DE 信号(驱动器使能)

#### 15. 接收使能寄存器(RE\_EN)

偏移地址: 0xB4

RE_EN	[0]	RW
-------	-----	----

描述: RE 使能控制, 该寄存器位用于控制 RE(Receiver Enable, 接收使能)信号的断言(assertion)和解除断言(de-assertion)。写 0: 解除断言 RE 信号(即 RE 信号为低电平)写 1: 断言 RE 信号(即 RE 信号为高电平)

#### 16. 接收使能寄存器(RE\_EN)

偏移地址: 0xC0



DLF	[3:0]	RW
-----	-------	----

描述：本寄存器设置波特率发生器的分数分频值，与 DLH/DLL 寄存的整数分频值共同组成完整分频系数：

·计算关系:

总分频值 = (DLH << 8 | DLL) + DLF / 2<sup>DLF\_SIZE</sup>

## 17. 线路扩展控制寄存器(LCR\_EXT)

偏移地址：0xCC

RW	RW	RW	RW
[3]	[2]	[1]	[0]
TRANSMIT_MODE	SEND_ADDR	ADDR_MATCH	DLS_E

bit0	DLS_E:该位用于启用传输和接收传输的9位数据。 0x0:失能9位传输和接收 0x1:使能9位传输和接收
bit1	ADDR_MATCH:地址匹配模式，该位用于在接收期间启用地址匹配功能。 0x0:正常模式； UART将开始接收数据，9位字符将被形成并写入接收RXFIFO。用户负责读取数据并区分地址和数据。 0x1:地址匹配模式； UART会等待接收到的字符的第9位设置为1,然后进一步检查地址是否与“接收地址匹配寄存器”中编程的内容匹配。如果找到匹配项，则后续字符将被视为有效数据，UART开始接收数据。
bit2	SEND_ADDR:发送地址控制bit0x0:发送一个9位字符，其中第9位设置为0,其余8位从通过8位宽的THR/STHR寄存器编程的TXFIFO中获取。 0x1:发送一个9位字符，其中第9位设置为1,其余8位与“发送地址寄存器”中编程的内容匹配。
bit3	TRANSMIT_MODE:传输模式。 0x0:在这种模式下，发送保持寄存器(THR)和影子发送保持寄存器(STHR)为8位宽。用户需要将地址编程到发送地址寄存器(TAR)中，将数据编程到THR/STHR寄存器中。SEND_ADDR位用作控制开关，以指示UART何时发送地址。 0x1:在这种模式下，发送保持寄存器(THR)和影子发送保持寄存器(STHR)为9位宽。用户需要确保正确地将地址或数据写入THR/STHR寄存器。若为地址，第9位设置为1;若为数据，第九位设置为0

## 6.6 TIMER

## 6.6.1 简介

BL1820 包含 8 个独立 16 位 Timer (Timer0~Timer7)，支持定时中断、PWM 输出功能，可用于定时任务、电机控制、LED 调光等场景。

## 6.6.2 主要特性

- 计数器：16 位递减计数器，支持自由运行（加载 0xFFFF）、自定义运行（加载 TimerNLoadCount）；
- 时钟：可选择 48 MHz/24 MHz 时钟源，支持预分频（分频比 2~4096，PWMTT\_CLK\_SEL 配置）；
- PWM：支持独立 PWM、互补 PWM 输出，占空比 0%~100%可调，支持反相；
- 中断：计数器下溢（计数到 0）时触发中断，支持中断屏蔽。

## 6.6.3 功能描述

### [6.6.3.1] 定时器模式

1. **定时功能**：通过 TimerNLoadCount 设置初始值，计数器以预分频后的时钟递减，计数到 0 时触发中断，需软件清除中断标志；
2. **PWM 模式**：通过 TimerNCompare 设置比较值，计数器递减至比较值时翻转输出电平，实现 PWM 波形；占空比 = (比较值/初始值) × 100%（递减计数）。
3. **定时计数器**：

定时器向下计数。使用加载计数寄存器将初始值加载到定时器中，计数到 0 下溢后产生中断或重新加载初始计数。定时器在复位或禁用后启用，定时器会重新加载初始计数。当定时器倒计时至 0 时，会重新加载初始计数，可通过修改模式来改变初始计数的值。不同模式操作如下：

- 用户自定义计数模式：定时器加载 TimerNLoadCount 寄存器的当前值
- 自由运行模式：定时器加载到最大值（0xFFFF）

### [6.6.3.2] 预分频配置

Timer 预分频由 PWMTT\_CLK\_SEL 寄存器控制，每 4 位对应 1 个 Timer，分频比配置如下：

PWMTT_CLK_SEL	分频比	PWMTT_CLK_SEL	分频比
0000	2	1000	512

PWMTT_CLK_SEL	分频比	PWMTT_CLK_SEL	分频比
0001	4	1001	1024
0010	8	1010	2048
0011	16	1011	4096
0100	32	1100~1111	保留
0101	64		
0110	128		
0111	256		

#### [6.6.3.2] 定时器中断

- 中断产生及条件

Timer 会在预分频的一个时钟下降沿进入中断，要检测和服务中断，系统时钟必须处于活动状态。Timer 使能用于激活定时器时钟，并确保定时器运行时为组件提供活动系统时钟。

- 中断清除

如果定时器被使能，则中断保持有效，直到通过读取 TimerNEOI（单独中断结束寄存器）和 TimersEOI（全局中断结束寄存器）之一时将其清除。如果定时器被禁止，定时器中断被清除。可以通过读取 TimerNEOI 寄存器来清除单个定时器中断。也可以通过读取 TimersEOI 寄存器或禁用定时器来一次清除所有活动的定时器中断。

- 屏蔽中断

使用 TimerNControlReg 寄存器屏蔽每个单独的定时器中断。

### 6.6.4 寄存器映射（Timer0）

Timer 定时器基地址：0x41008000，其他有关 Timer 寄存器配置，在此基地址加上相应的偏移地址。

说明：寄存器模块 Timer1 到 Timer7 的寄存器映射和功能与 Timer0 完全一致，只是对应不同的定时器实例。因此，以下内容仅以 Timer0 为例进行详细描述。

#### 1. Timer0 装载值寄存器（Timer0\_Load\_Count）

偏移地址：0x0

Timer0_Load_Count	[15:0]	RW
-------------------	--------	----

设置 Timer0 的装载值。从这个值开始向下计数。写入此寄存器的任何值都会加载到 Timer0 中。

## 2. Timer0 当前值寄存器 (Timer0\_Current\_Value)

偏移地址：0x4

Timer0_Current_Value	[15:0]	RO
----------------------	--------	----

Timer0 的当前值。此寄存器默认值为 0xFFFF。

## 3. Timer0 控制寄存器 (Timer0\_Control\_Reg)

偏移地址：0x8

RO	RW	RW	RW	RW
[4]	[3]	[2]	[1]	[0]
Timer0_ON100PWM_EN	Timer0_PWM	Timer0_INTERRUPT_MASK	Timer0_MODE	Timer0_ENABLE

bit0	Timer0_ENABLE: Timer0使能/失能 0x0: 失能 0x1: 使能
bit1	Timer0_MODE: Timer0模式选择 0x0: 自由运行模式 0x1: 用户定义运行模式
bit2	Timer0_INTERRUPT_MASK: Timer0中断屏蔽 0x0: 不屏蔽(开启中断) 0x1: 屏蔽 (关闭中断)
bit3	Timer0_PWM: Timer0的PWM配置 0x0: 失能PWM模式 0x1: 使能PWM模式
bit4	Timer0_ON100PWM_EN: PWM占空比 0x0: 禁用定时器0%和100%PWM占空比模式 0x1: 启用定时器0%和100%PWM占空比模式

## 4. Timer0 清除中断寄存器 (Timer0\_EOI)

偏移地址：0x0C

Timer0_EOI	[0]	RO
------------	-----	----

从该寄存器读取会返回全零（0），并清除来自 Timer0 的中断。

## 5. Timer0 中断状态寄存器 (Timer0\_Int\_Status)

偏移地址：0x10

Timer0_Int_Status	[0]	RO
-------------------	-----	----

此寄存器表示 Timer0 的中断状态，0x0 表示 Timer0 中断未激活，0x1 表示 Timer0 中断 激活。

## 6. Timer1-Timer7 寄存器

地址偏移：Timer1 起始地址 0x14，Timer2 起始地址 0x28，Timer3 起始地址 0x3C，...，Timer6 起始地址 0x78，Timer7 起始地址 0x8C。

对于 Timer1-Timer7 来讲，是在 Timer0 寄存器的地址上加上（ $n * 0x14$ ）。例如：Timer6 的 Timer6\_Load\_Count 偏移地址为  $\text{Timer0\_Load\_Count} + 6 * 0x14 = 0x78$ 。

**功能定义：**所有寄存器（Timerx\_Load\_Count、Timerx\_Current\_Value、Timerx\_Control\_Reg、Timerx\_EOI、Timerx\_Int\_Status）的功能、位定义、权限、复位值及描述均与 Timer0 对应寄存器完全一致（详见「上述 Timer0 寄存器描述」章节）

## 7. Timers 清除中断寄存器 (Timers\_EOI)

偏移地址：0xA8

Timers_EOI	[7:0]	RO
------------	-------	----

从该寄存器读取会返回全零（0），并清除所有 Timer 的中断。

## 8. Timers 原始中断寄存器 (Timers\_Raw\_Int\_Status)

偏移地址：0xC

Timers_EOI	[7:0]	RO
------------	-------	----

该寄存器一次性给出全部 8 路定时器（Timer0~Timer7）的原始中断状态。对应 位给 1 即为对应 Timer 已产生中断请求，且尚未被任何屏蔽逻辑过滤。

## 9. Timer0\_Load\_Count2 寄存器

偏移地址：0xB0

Timer0_Load_Count2	[16:0]	RW
--------------------	--------	----

## 10. Timer1\_Load\_Count2-Timer1\_Load\_Count2

在 Timer0\_Load\_Count2 寄存器的地址基础上，加上  $n \times 0x04$ ，例如：  
Timer6\_Load\_Count2 寄存器的地址为  $0xB0 + 6 \times 0x04$ 。

注：

- 当置 Timerx\_Load\_Count2 时，占空比 =  $\text{Timerx\_Load\_Count2} / (\text{Timerx\_Load\_Count} + \text{Timerx\_Load\_Count2})$ 。

Timerx\_Current\_Value 的初值为 0xFFFF，当使能 PWM0~100% 功能时，如果其中一个 Timerx\_Load\_Count 为 0，那么 Timerx\_Current\_Value 值不会更新，也不会减一，而是保持不变。

如果只配置 Timerx\_Load\_Count 为非 0 值，则计数器会从 0xFFFF 减到 0 后，再使用配置的值重新递减。

- 在 Timerx\_ENABLE 使能(从 0 变为 1)，或 Timerx\_Load\_Count 减到 0 时，TimerX\_Current\_Value 更新为用户配置值。

## 6.7 GPADC

### 6.7.1 简介

GPADC（通用 ADC）为 10 位逐次逼近型模数转换器，支持外部引脚电压、内部电源电压、温度传感器采样，最高采样率 3 MHz。为提高采样精度，减少软件负担，内部集成采样点数字平均模块，可以对最多 256 个采样点进行均值平滑。

### 6.7.2 主要特性

- 分辨率：10 位；
- 采样率：3 MHz/2 MHz/1 MHz/48 KHz（可配置）；
- 输入通道：8 路外部模拟输入（ADC0~ADC7）+ 内部电压（AVDD、VDD、VPTAT）；
- 滤波：支持 1~256 点均值滤波；
- 模式：单次采样、连续采样（支持 DMA 传输）；
- 中断：采样完成中断、DMA 传输完成中断。

### 6.7.3 功能描述

#### [6.7.3.1] 时钟配置

ADC 采样时钟公式:

$$[f_{\text{adc\_clk}}] = \frac{\text{GPADC\_SOURCE\_CLK}}{(\text{GPADC\_CLK\_DIV} + 1) \times 2}$$

- GPADC\_SOURCE\_CLK: ADC 时钟源 (0=RC\_24M, 1=XTAL\_24M) ;
- GPADC\_CLK\_DIV: 分频系数 (0~31, 如配置 1 MHz 采样率时, GPADC\_CLK\_DIV=11) 。

### [6.7.3.2] 采样模式

通过寄存器 GPADC\_CONT\_MODE 配置采样模式, 所有采样模式都可以使能采样均值滤波功能。采样均值滤波功能使能时, 多个原始采样点均值滤波输出一个有效的采样点。

**单次采样:** 软件或硬件 (PWM 上升沿) 触发, 采样 1 次后进入空闲状态;

单次采样模式可由软件条件或硬件条件进行触发:

- 软件条件: 寄存器 GPADC\_TRIG 写“1”时, 触发采样
- 硬件条件: 寄存器 GPADC\_TIMER\_SEL 对应的 PWM 信号输出上升沿触发采样

单次采样模式适用于采样速率较低的场合, 软件在需要的时候触发一次单次采样, 使用简单方便, 并可完全规避因响应不及时而导致的采样值覆盖风险。

**连续采样:** 触发后持续采样, 采样值可通过 DMA 存储至指定内存 (8 KB DMA 内存, 分 4 块, 每块 2 KB)。连续采样模式可由软件条件或硬件条件进行触发:

- 软件条件: 寄存器 GPADC\_TRIG 写“1”时, 触发采样
- 硬件条件: 寄存器 GPADC\_TIMER\_SEL 对应的 PWM 信号输出上升沿触发采样

连续采样模式适用于采样速率较高的场合, 软件需要在指定时间内读取采样值, 否则会有采样值覆盖风险, 或者启用 GPADC DMA, 将采样值存储至指定内存区域。

### [6.7.3.3] GPADC DMA

寄存器 GPADC\_DMA\_EN 使能后, GPADC 工作在 DMA 模式。GPADC DMA 模式时, 不关心寄存器 GPADC\_CONT\_MODE 设置, 始终工作在连续采样模式, 并且每个采样点均会被存储到指定内存区域。GPADC DMA 数据存储有两种格式, Word 格式和 Half-word 格式, 通过寄存器 GPADC\_STORE\_MODE 进行选择。

GPADC DMA 内存总共 8KB, 地址偏移为 0x2000\_2000。DMA 内存平均分为四块, 每块 2KB。GPADC DMA 所使用的内存区域通过 GPADC\_START\_BK 和 GPADC\_BK\_NUM 设置。

GPADC\_BK\_START 用于选择起始内存块, GPADC\_BK\_NUM 用于选择内存块数量。

### [6.7.3.4] 均值滤波器

ADC 采样点因电路噪声、环境变化或外部干扰可能会产生采样偏差。数字均值滤波器可以通过滑动窗口对采样点进行平滑滤波, 从而减小采样偏差, 提高采样点的可靠性和稳定性。通过寄存器 GPADC\_AVG\_WIN<3:0>可设置均值滤波器窗口大小, 窗口最大为 256 个采样点。

### [6.7.3.5] 数据格式

GPADC 原始采样值为 10-bit 位宽，数据格式由模拟 ADC 决定，默认为二进制自然码格式。GPADC 原始采样值可以通过寄存器 GPADC\_FORMAT 进行转换，可根据需要转换为常用的二进制补码形式。

ADC 采样值存储到寄存器或内存中时，通常是字节对齐的，为方便软件使用，需要对数据进行位宽扩展。通过寄存器 GPADC\_SIGNEXT 可选择进行有符号数扩展或无数数扩展，有符号数扩展按照原始采样值最高位扩展其他高位，无数数扩展不关心原始采样值最高位，直接在高位补零。扩展位宽根据其他设置，16 位和 32 位可选。

### [6.7.3.6] 中断

GPADC 共有两个中断源，DMA 中断和 SAMPLE 中断。两个中断源可通过寄存器 GPADC\_IE<1:0>单独使能。SAMPLE 中断为采样完成中断，单次采样模式和连续采样模式均会触发采样完成中断。其中，单次采样模式只会触发一次采样完成中断，连续采样模式则会在每个采样点采样完成后，触发采样完成中断，此时软件需要连续响应采样完成中断。

## 6.7.4 寄存器映射

GPADC 基地址：0x40040000，其他有关 GPADC 的寄存器配置，在此基地址加上相应的偏移地址。

### 1. GPADC 使能寄存器(gpadc\_en)

偏移地址：0x0

RW	RW
[1]	[0]
da_gpadc_en	da_gpadc_pga_en

bit0	da_gpadc_pga_en: PGA使能/失能 0x0: 失能 0x1: 使能
bit1	da_gpadc_en: ADC使能/失能 0x0: 失能 0x1: 使能

### 2. GPADC 控制寄存器(gpadc\_ctrl)



偏移地址：0x4

RW	RW	RW	RW	RW
[14:13]	[12]	[11:8]	[7:4]	[3:0]
rg_gpadc_att_sel	rg_gpadc_diff_en	rg_gpadc_pga_gc	rg_gpadc_vin_sel	rg_gpadc_vip_sel

bit[3:0]	rg_gpadc_vip_sel: GPADC 正输入信号选择 0x0: GPIO[2] 0x1: GPIO[6] 0x2: GPIO[7] 0x3: GPIO[10] 0x4: GPIO[15] 0x5: GPIO[17] 0x6: GPIO[19] 0x7: GPIO[21] 0x8: internal 0.9 V reference voltage 0x9: internal 0.9 V reference voltage 0xA: internal 0.9 V reference voltage 0xB: internal 0.9 V reference voltage
bit[7:4]	rg_gpadc_vin_sel: GPADC 负输入信号选择 0x0: GPIO[2] 0x1: GPIO[6] 0x2: GPIO[7] 0x3: GPIO[10] 0x4: GPIO[15] 0x5: GPIO[17] 0x6: GPIO[19] 0x7: GPIO[21] 0x8: temperature sensor 0x9: battery voltage 0xA: core input voltage 0xB: core supply voltage
bit[11:8]	rg_gpadc_pga_gc: GPADC PGA 增益控制，增益 = 3dB ×rg_gpadc_pga_gc 取值：0 ~ 15（根据实际需求配置）
bit12	rg_gpadc_diff_en: GPADC 差分模式使能 0x0: 失能差分输入模式 0x1: 使能差分输入模式
bit[14:13]	rg_gpadc_att_sel: GPADC 输入电压分压控制 0x0: 1 (high input impedance) 0x1: 1/2 0x2: 1/3 0x3: 1/4

### 3. GPADC 配置寄存器(gpadc\_cfg)

偏移地址：0x8

RW	RW	RW	RW
[10:9]	[8:7]	[6:3]	[2:0]
rg_gpadc_pga_cm1_ibias	rg_gpadc_pga_cm3_ibias	rg_gpadc_pga_rs	rg_gpadc_pga_vcm_sel

bit[2:0]	rg_gpadc_pga_vcm_sel: PA（功率放大器）和 OTP（一次性可编程）模块的 LDO 0x0: 关闭 PA/OTP LDO 0x1~0x7: 根据设计配置不同模式或电压等级默认为0x1
bit[6:3]	rg_gpadc_pga_rs: VCO（压控振荡器）LDO 0x0: 快速充电关闭 0x1~0xF: 根据实际设计配置不同充电速度或模式默认为0x1
bit[8:7]	rg_gpadc_pga_cm3_ibias: LO（本振）LDO 使能控制 0x0: 关闭 0x1: 开启默认为0x1
bit[10:9]	rg_gpadc_pga_cm1_ibias: LNA（低噪声放大器）LDO 使能控制 0x0: 关闭 0x1: 开启默认为0x1

#### 4. GPADC 动作寄存器(gpadc\_action)

偏移地址：0x60

WO	RO	RW	RW	RW	RW	RW	RW
[31]	[26:16]	[15:12]	[6:4]	[3]	[2]	[1]	[0]
gpadc trig	gpadc wp	gpadc avg _win	gpadc time r_sel	gpadc timer mode	gpadc cont mode	gpadc sam p_en	gpadc dma _en

bit[0]	gpadc_dma_en: DMA使能/失能，当设置为 '1' 时，启用 DMA 采样，同时平均模块也会生效。 0x0: 失能 0x1: 使能
bit[1]	gpadc_samp_en: 开启/关闭ADC采样 0x0: 失能 0x1: 使能

bit[2]	gpadc_cont_mode: GPADC采样模式，只有当gpadc_samp_en为1时有效 0x0: 单次采样模式，在gpadc_is为0时，硬件自动开始一次采样 0x1: 连续采样模式，硬件自动采样，忽略gpadc_is状态
bit[3]	gpadc_timer_mode: 定时器触发模式控制。当设置为 '1' 时，定时器 PWM 信号的上升沿会触发一次 ADC 采样事件。 0x0: 定时器触发模式禁用 0x1: 定时器触发模式启用
bit[6:4]	gpadc_timer_sel: 定时器选择，用于指定哪个定时器信号将被用来触发 ADC 采样。 0x0~0x7: 选择对应的定时器。
bit[15:12]	gpadc_timer_sel: 设定 GPADC 平均计算窗口大小。实际平均窗口大小为 2 的 gpadc_avg_win 次幂。例如，如果 gpadc_avg_win 设置为 8，则平均窗口大小为 256（最大）。
bit[26:16]	gpadc_wp: GPADC 写指针，指示当前数据写入的位置。
bit[31]	gpadc_trig: 写入 可触发一次 ADC 采样操作。该操作仅在 gpadc_samp_en 位设置为 '1' 时有效。

## 5. GPADC 配置寄存器(gpadc\_cfg)

偏移地址：0x64

RW	RW	RW	RW	RW	RW
[15:8]	[7:6]	[5:4]	[2]	[1]	[0]
gpadc_int_interval	gpadc_bk_num	gpadc_start_bk	gpadc_signext	gpadc_format	gpadc_store_mode

bit[0]	gpadc_store_mode:GPADC 存储模式： 0x0: 32 位模式 0x1: 16 位模式
bit[1]	gpadc_format: GPADC 采样值格式 0x0: 二进制补码格式（signed two's complement） 0x1: 原始二进制格式（binary code）
bit[2]	gpadc_signext: GPADC 采样值符号扩展控制： 0x0: 无符号扩展（unsigned ext） 0x1: 有符号扩展（signed ext）
bit[5:4]	gpadc_start_bk: GPADC RAM 块起始选择，每个块大小为 2KB。 注：GPADC 可用 RAM 范围为 0x2000_2000~0x2000_3FFF，共 8KB

bit[7:6]	gpadc_bk_num: GPADC RAM 块数量选择, 实际可用 RAM 容量为 (gpadc_bk_num + 1)*2KB
bit[15:8]	gpadc_int_interval: GPADC 中断间隔控制。当 (addr[7:0]   gpadc_interval) == 8。触发中断

## 6. GPADC 中断控制寄存器(gpadc\_int\_ctrl)

偏移地址: 0x68

RW	RW1C	RW1C
[17:16]	[9:8]	[1:0]
gpadc_ie	gpadc_mis	gpadc_is

bit[1:0]	gpadc_is:中断状态: bit[0]: DMA 中断状态 bit[1]: GPADC 采样完成中断状态
bit[9:8]	gpadc_mis: 屏蔽中断状态: bit[0]: DMA 屏蔽中断状态 bit[1]: GPADC 采样完成屏蔽中断状态
bit[17:16]	gpadc_ie:中断使能控制: bit[0]: DMA 中断使能控制 bit[1]: GPADC 采样完成中断使能控制

## 4. GPADC 采样值(gpadc\_samp)

偏移地址: 0x6C

gpadc_samp	[31:0]	RO
------------	--------	----

GPADC 的采样值, 只有当 gpadc\_is[0]置位时有效。

注: 读取 gpadc\_samp 后, 硬件会自动清除 gpadc\_mis[0]和 gpadc\_is[0]

## 5. GPADC 时钟控制寄存器(gpadc\_clk\_ctrl)

偏移地址: 0x70

[19:16]	[15:8]	[5]	[4]	[3]	[2]	[1]	[0]
gpadc_clk_pre	gpadc_clk_div	gpadc_clk_out_pol	gpadc_clk_out_inv	gpadc_clk_samp_inv	gpadc_clk_en	gpadc_clk_autocg	gpadc_clk_sel
RW	RW	RW	RW	RW	RW	RW	RW

bit[0]	gpadc_clk_sel: 时钟源选择: 0x0: RC24M 0x1: XTAL 24M
bit[1]	gpadc_clk_autocg: ADC采样时钟自动门控使能: 0x0: 禁用自动时钟门控 0x1: 启用自动时钟门控
bit[2]	gpadc_clk_en: 启用采样时钟, 忽略自动时钟门控设置 0x0: 开启 0x2: 关闭
bit[3]	gpadc_clk_samp_inv: GPADC模块ADC采样边沿控制: 0x0: 下降沿(默认) 0x1: 上升沿
bit[4]	gpadc_clk_out_inv: GPADC时钟输出反转选择: 0x0: 正常输出 0x1: 反转输出
bit[5]	gpadc_clk_out_pol: GPADC时钟输出初始极性
bit[15:8]	gpadc_clk_div: ADC时钟分频值, 实际频率为 $\text{gpadc\_clk\_source} / ((\text{gpadc\_clk\_div} + 1) / 2)$
bi[19:16]	gpadc_clk_pre: ADC采样预延迟, 当GPADC使能后, 经过0到15次采样后开始输出采样结果

## 6.8 IIC

### 6.8.1 简介

IC 总线使用连接设备的“SCL”（串行时钟总线）和“SDA”（串行数据总线）来传输信息。主机在 SCL 线上输出串行时钟信号，数据在 SDA 线上进行传输，每传输一个字节（最高位 MSB 开始传输），后面跟随一个应答位。一个 SCL 时钟脉冲传输一个数据位。

### 6.8.2 主要特性

- 支持 IC Master 和 IC Slave 两种模式
- IIC 支持以下三个速度模式
- Standard mode(0 to 100 Kb/s)
- Fast mode( $\leq 400$  Kb/s)

- Fast mode plus( $\leq 1000$  Kb/s)

### 6.8.3 功能描述

#### [6.8.3.1] 数据传输及信号

IIC 传输协议包含四个部分：起始信号 (start) 或重复起始信号 (restart), 从机地址及读写位, 数据位, 停止信号 (stop)。

- 起始信号(start): 当总线处于空闲状态下 (SCL 和 SDA 线同时为高), SDA 线上出现由高到低的信号, 表明总线上产生了起始信号

- 重复起始信号 (restart): 当两个起始信号之间没有停止信号时, 即产生了重复起始信号。主机采用这种方法与另一个从机或相同的从机以不同传输方向进行通信 (例如: 从写入设备到从设备读出)而不释放总线。

- 停止信号 (stop): 当 SCL 线为高时, SDA 线上出现由低到高的信号, 被定义为停止信号。主机向总线发出停止信号结束数据传送。

- 从机地址与读写 bit7 位从设备寻址模式: 在 7 比特从设备地址模式中, 第一个字节的前 7bit(A6:A0) 就是从设备地址, 最低位 (LSB) 为 R/W 位。当 R/W 位被设置为 0 时, 表示写操作, 即主设备向从设备写数据。当 R/W 位被设置为 1 时, 表示读操作, 即主设备从从设备读取数据。

- 10 位从设备寻址模式: 在 10 比特从设备地址模式中, 需要传输 2 个字节数据来设置 10bit 地址。传输的第一个字节有 3 部分定义: 前 5 个 bit(bit7:3, 协议固定为 11110), 用于通知从设备这是一个 10bit 模式的传输; 随后的 2 个 bit(A9: A8) 表示从设备地址, 最后 1bit (LSB)是 R/W 位。传输的第二个字节 (A7:A0)

则全部用于表示从设备地址。这样, A9:A0 就表示了 10bit 从设备地址。

- 数据位: 数据传输过程中, 一个 SCL 时钟脉冲传输一个数据位, 且 SDA 线只有在 SCL 为低时才可以改变。

#### [6.8.3.2] 数据传输应答

IIC 每次传输完 8 位数据后, 需要从设备反馈一个应答位, 以确认从设备是否正常接收了数据。当主设备发送了 8 位数据后, 会再产生一个时钟, 此时主设备放开 SDA 的控制, 读取 SDA 电平, 在上拉电阻的影响下, 此时 SDA 默认为高, 必须将从设备拉低, 以确认收到数据。应答信号为低电平时, 规定为有效应答位(ACK 简称应答

位), 表示接收器已经成功地接收了该字节; 应答信号为高电平时, 规定 为非应答位 (NACK), 一般表示接收器接收该字节没有成功。

### [6.8.3.3] IIC 中断

PADC\_BK\_START 用于选择起始内存块, GPADC\_BK\_NUM 用于选择内存块数量。下列详细介绍 IC 中断及产生条件, 可结合寄存器 IC\_INTR\_MASK 使用。

●RX\_UNDER: 当处理器通过 IC\_DATA\_CMD 寄存器读取接收缓冲器为空时置 位。

●RX\_OVER:当接收缓冲器被填满, 而且还有数据从外设发送过来时被置位; 缓冲器被填满后接收的数据将会丢失。

●RX\_FULL: 当接收缓冲器达到或者超过 IC\_RX\_TL 寄存器中规定的阈值时被置位; 当数据低于阈值时标志位将被自动清除。

TX\_OVER:当发送缓冲器被填满, 而且处理器试图发送另外的命令写 IC\_DATA\_CMD 寄存器时被置位。

TX\_EMPTY:当发送缓冲器等于或者低于 IC\_TX\_TL 寄存器中规定的阈值时 被置位; 当数据高于阈值时标志位将被自动清除。

RD\_REQ:当 IIC 模块作为从机时并且另外的主机试图从本模块读取数据时被 置位。

TX\_ABRT:当 IIC 模块无法完成处理器下达的命令时被置位, 有如下几种原

因:

- 1.发送地址字节后没有从机应答地址
- 2.识别成功后主机发送的数据从机没有应答
- 3.当 IIC 模块只能作为从机时试图发送主机命令
4. 当模块的 RESTART 功能被关闭, 而处理试图完成的功能必须要 RESTART 功能开启才能完成
- 5.高速模块主机代码被应答
- 6.START BYTE 被应答
- 7.模块仲裁失败(无论标志位什么时候被置位, 发送缓冲器和接收缓冲器的内容 都会被刷新)

RX\_DONE:当 IIC 模块作为从机发送数据时, 如果主机没有应答则置位; 这种情况发生在 IIC 模块发送最后一个字节数据时, 表明传输结束。

ACTIVITY:表明 IIC 模块正在活动, 这个标志位将会一直保持直到用以下 4 种方 式

清除。

1. 关闭 IIC
2. 读取 IC\_CLR\_ACTIVITY 寄存器
3. 读取 IC\_CLR\_INTR 寄存器
4. 系统重启

·STOP\_DET: 表明 IIC 总线上产生了 STOP 信号，无论模块作为主机还是从机

·START\_DET: 表明 IC 总线上产生了 START 信号，无论模块作为主机还是从机

### 6.8.4 寄存器映射

IIC 基地址：0x41005000，其他有关 IIC 寄存器配置，在此基地址加上相应的偏移地址。

#### 1. IIC 控制寄存器 (IC\_CON)

偏移地址：0x0

RW	RW	RW	RW	RW	RW
[6]	[5]	[4]	[3]	[2:1]	[0]
IC_SLAVE_DISABLE	IC_RESTART_EN	IC_10BITADDR_MASTER	IC_10BITADDR_SLAVE	SPEED	MASTER_MODE

bit0	MASTER_MODE: 主机模式 0x0: 主机模式失能 0x1: 主机模式使能
bit[2:1]	SPEED: 传输速度模式 0x1: Standard模式 0x2: Fast/Fast Plus快速模式
bit3	IC_10BITADDR_SLAVE: 从机寻址位 0x0: ADDR_7BITs 从机7位寻址模式 0x1: ADDR_10BITs 从机10位寻址模式
bit4	IC_10BITADDR_MASTER: 主机寻址位 0x0: ADDR_7BITs 主机7位寻址模式 0x1: ADDR_10BITs 主机10位寻址模式
bit5	IC_RESTART_EN: 主机重复开始信号 0x0: 失能 0x1: 使能



bit6	IC_SLAVE_DISABLE: 从机模式 0x0: 失能从机 0x1: 使能从机
------	--

## 2. IIC 目标地址寄存器 (IC\_TAR)

偏移地址: 0x4

IC_TAR	[9:0]	RW
--------	-------	----

定义主设备通信的目标从设备地址，仅需单次写入即可生成 START BYTE，当 IC\_TAR = IC\_SAR 时启用回环，仅支持单向单工回环（主→从）；不支持全双工回环（FIFO 共享冲突）；主设备不可向自身发送数据。

## 3. IIC 从机地址寄存器 (IC\_SAR)

偏移地址: 0x8

IC_SAR	[9:0]	RW
--------	-------	----

定义 I<sup>2</sup>C 作为设备时的响应地址，仅当 I<sup>2</sup>C 接口禁用时（IC\_ENABLE[0]=0）可写，其他时间写入无效。

禁止设置以下保留地址范围：

- 0x00 ~ 0x07（系统保留）
- 0x78 ~ 0x7F（10-bit 地址保留区） 设置保留地址将无效。

## 4. IIC 数据寄存器 (IC\_DATA\_CMD)

偏移地址: 0x10

WO	RW
CMD	DAT
[8]	[7:0]

bit[7:0]	DAT: 传输/接收的数据
bit[8]	CMD:主机读写数据 0x1: 主机读数据命令 0x2: 主机写数据命令

## 5. IIC 标准速度 SCL 高电平计数寄存器 (IC\_SS\_SCL\_HCNT)

偏移地址: 0x14

IC_SS_SCL_HCNT	[15:0]	RW
----------------	--------	----

设置标准速度模式（Standard Speed）下 SCL 时钟高电平周期的计数。

## 6. IIC 标准速度 SCL 低电平计数寄存器 (IC\_SS\_SCL\_LCNT)

偏移地址：0x18

IC_SS_SCL_LCNT	[15:0]	RW
----------------	--------	----

设置标准速度模式（Standard Speed）下 SCL 时钟低电平周期的计数。

## 7. IIC 快速模式 SCL 高电平计数寄存器 (IC\_FS\_SCL\_HCNT)

偏移地址：0x1C

IC_FS_SCL_LCNT	[15:0]	RW
----------------	--------	----

设置快速模式（Fast Speed）下 SCL 时钟低电平周期的计数。

## 8. IIC 快速模式 SCL 低电平计数寄存器 (IC\_FS\_SCL\_LCNT)

偏移地址：0x20

IC_FS_SCL_LCNT	[15:0]	RW
----------------	--------	----

设置快速模式（Fast Speed）下 SCL 时钟低电平周期的计数。

## 9. IIC 中断状态标志寄存器 (IC\_INTR\_STAT)

偏移地址：0x2C

RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
[10]	[9]	[8]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
R STAR T DET	R STOP _DET	R ACT IVITY	R RX DONE	R TX ABRT	R RD REQ	R TX E MPTY	R TX OVER	R RX FULL	R RX OVER	R RX U NDER

bit0	R_RX_UNDER: 读取空接收缓冲区时触发（禁用I <sup>2</sup> C时自动清除） 0x0: 中断未激活 0x1: 中断激活
bit1	R_RX_OVER: 接收溢出。 若启用IC_RX_FULL_HLD_BUS_EN且 IC_CON=1, 则永不触发 0x0: 中断未激活 0x1: 中断激活
bit2	R_RX_FULL: 接收缓冲区达到阈值。 接收数据 ≥ IC_RX_TL阈值时触发 0x0: 中断未激活 0x1: 中断激活

bit3	R_TX_OVER: 发送溢出，发送缓冲区满时，继续写入触发 0x0: 中断未激活 0x1: 中断激活
bit4	R_TX_EMPTY: 发送缓冲区空，行为由IC_CON.TX_EMPTY_CTRL配置 0x0: 中断未激活 0x1: 中断激活
bit5	R_RD_REQ: 从机收到主机读请求 0x0: 中断未激活 0x1: 中断激活
bit6	R_TX_ABRT: 发送中止 0x0: 中断未激活 0x1: 中断激活
bit7	R_RX_DONE: 传输完成 0x0: 中断未激活 0x1: 中断激活
bit8	R_ACTIVITY: 总线活动状态 0x0: 中断未激活 0x1: 中断激活
bit9	R_STOP_DET: 检测到PC停止条件 (STOP) 0x0: 中断未激活 0x1: 中断激活
bit10	R_START_DET: 检测到PC起始条件 (START/RESTART) 0x0: 中断未激活 0x1: 中断激活

## 10. IIC 中断屏蔽寄存器 (IC\_INTR\_MASK)

偏移地址: 0x30

RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
[10]	[9]	[8]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
M STAR T DET	M STO P DET	M ACT IVITY	M RX DONE	M TX ABRT	M RD REQ	M TX E MPTY	M TX OVER	M RX FULL	M RX OVER	M RX U NDER

bit0	M_RX_UNDER: 屏蔽RX_UNDER中断 (复位默认屏蔽) 0x0: 屏蔽 0x1: 取消屏蔽
bit1	M_RX_OVER: 屏蔽RX_OVER中断 (复位默认屏蔽) 0x0: 屏蔽 0x1: 取消屏蔽

bit2	M_RX_FULL: 屏蔽RX_FULL中断（复位默认屏蔽） 0x0: 屏蔽 0x1: 取消屏蔽
bit3	M_TX_OVER: 屏蔽TX_OVER中断（复位默认屏蔽） 0x0: 屏蔽 0x1: 取消屏蔽
bit4	M_TX_EMPTY: 屏蔽TX_EMPTY中断（复位默认屏蔽） 0x0: 屏蔽 0x1: 取消屏蔽
bit5	M_RD_REQ: 屏蔽RD_REQ中断（复位默认屏蔽） 0x0: 屏蔽 0x1: 取消屏蔽
bit6	R_TX_ABRT: 屏蔽TX_ABRT中断（复位默认屏蔽） 0x0: 屏蔽 0x1: 取消屏蔽
bit7	M_RX_DONE: 屏蔽RX_DONE中断 0x0: 屏蔽 0x1: 取消屏蔽
bit8	M_ACTIVITY: 屏蔽ACTIVITY中断 0x0: 屏蔽 0x1: 取消屏蔽
bit9	M_STOP_DET: 屏蔽STOP_DET中断 0x0: 屏蔽 0x1: 取消屏蔽
bit10	M_START_DET: 屏蔽START_DET中断 0x0: 屏蔽 0x1: 取消屏蔽

## 11. IIC 原始中断状态寄存器 (IC\_RAW\_INTR\_STAT)

偏移地址: 0x34

RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
[10]	[9]	[8]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
START_DET	STOP_DET	ACTIVITY	RX_DONE	TX_ABRT	RD_REQ	TX_EMPTY	TX_OVER	RX_FULL	RX_OVER	RX_UNDER

bit0	<p><b>RX_UNDER:</b> 当试图通过读 IC_DATA_CMD 寄存器从空的接收缓冲区读取数据时，该位被置位。若模块被禁用（IC_ENABLE[0]=0），该位保持原值直到主/从状态机进入空闲，且当 ic_en 置 0 时中断被清除。</p> <p>0x0: 中断未激活 0x1: 中断激活</p>
bit1	<p><b>RX_OVER:</b> 接收溢出。当接收缓冲区已完全填满（达 IC_RX_BUFFER_DEPTH）且又接收到外部 I²C 设备的一个额外字节时，该位被置位。I²C 仍会发送 ACK，但超出 FIFO 容量的数据字节将丢失。若模块被禁用（IC_ENABLE[0]=0），该位保持原值直到主/从状态机进入空闲，且当 ic_en 置 0 时中断被清除。</p> <p>0x0: 中断未激活 0x1: 中断激活</p>
bit2	<p><b>RX_FULL:</b> 接收缓冲区达到阈值。接收数据 ≥ IC_RX_TL 阈值时触发。当接收缓冲区达到或超过 IC_RX_TL 寄存器设定的 RX_TL 阈值时，该位被置位。当缓冲区低于阈值时，硬件自动清除该位。若模块被禁用（IC_ENABLE[0]=0），RX FIFO 被清空并保持在复位状态，因此不会满；故一旦 IC_ENABLE 位 0 被写 0，该位即被清零，不受后续活动影响。</p> <p>0x0: 中断未激活 0x1: 中断激活</p>
bit3	<p><b>TX_OVER:</b> 发送溢出，当发送缓冲区已满（达到 IC_TX_BUFFER_DEPTH）且处理器仍试图通过写 IC_DATA_CMD 寄存器发送新的 I²C 命令时，该位被置位。若模块被禁用，该位保持原值直到主/从状态机进入空闲，且当 ic_en 置 0 时中断被清除。</p> <p>0x0: 中断未激活 0x1: 中断激活</p>
bit4	<p><b>TX_EMPTY:</b> 发送缓冲区空，TX_EMPTY 中断状态的行为取决于 IC_CON 寄存器中 TX_EMPTY_CTRL 位的设置。</p> <p>0x0: 中断未激活 0x1: 中断激活</p>
bit5	<p><b>RD_REQ:</b> 当 I²C 作为从机且另一主机试图从该从机读取数据时，该位被置 1。I²C 会拉低 SCL 保持总线等待，直到此中断得到服务。这意味着从机已被远程主机寻址并要求发送数据。处理器必须响应该中断，并将所需数据写入 IC_DATA_CMD 寄存器。处理器读取 IC_CLR_RD_REQ 寄存器后，该位自动清零。</p> <p>0x0: 中断未激活 0x1: 中断激活</p>
bit6	<p><b>TX_ABRT:</b> 该位表示 I²C 作为发送器时无法完成对发送 FIFO 内容的预期操作。这种情况可能发生在主机或从机模式下，称为“发送中止”。当该位为 1 时，IC_TX_ABRT_SOURCE 寄存器会给出中止原因。一旦任何由 IC_TX_ABRT_SOURCE 记录的事件导致发送中止，I²C 模块会立即清空/复位/置空 TX_FIFO。发送 FIFO 将保持此清空状态，直到软件读取 IC_CLR_TX_ABRT 寄存器；读取后，发送 FIFO 才重新准备接收来自 APB 外设的新数据。</p> <p>0x0: 中断未激活 0x1: 中断激活</p>

bit7	<b>RX_DONE:</b> 当 I <sup>2</sup> C 作为从机发送时，如果主机对某次发送的字节未给出 ACK，则该位被置 1。这通常发生在一次传输的最后一个字节，表明传输结束。 0x0: 中断未激活 0x1: 中断激活
bit8	<b>ACTIVITY:</b> 捕获 I <sup>2</sup> C 总线活动，一旦被置位将保持置位状态，直到通过以下四种方式之一清除： 失能I2C 读取IC_CLR_ACTIVITY寄存器 读取IC_CLR_INTR寄存器 系统重置 0x0: 中断未激活 0x1: 中断激活
bit9	<b>STOP_DET:</b> 检测到I <sup>2</sup> C停止条件（STOP）无论 I <sup>2</sup> C 当前处于主机还是从机模式。 0x0: 中断未激活 0x1: 中断激活
bit10	<b>START_DET:</b> 检测到I <sup>2</sup> C起始条件(START)或重新起始条件(RESTART)（主/从模式均有效） 0x0: 中断未激活 0x1: 中断激活

## 12. IIC 接收 FIFO 阈值寄存器 (IC\_RX\_TL)

偏移地址：0x38

IC_RX_TL	[7:0]	RW
----------	-------	----

接收 FIFO 阈值：此寄存器设置接收缓冲区触发中断的数据量阈值。

## 13. IIC 接收 FIFO 阈值寄存器 (IC\_TX\_TL)

偏移地址：0x3C

IC_TX_TL	[7:0]	RW
----------	-------	----

发送 FIFO 阈值：此寄存器设置发送缓冲区触发中断的数据量阈值。

## 14. IIC 清除全局中断寄存器 (IC\_CLR\_INTR)

偏移地址：0x40

IC_CLR_INTR	[0]	RO
-------------	-----	----

中断清除（全局）：读取此寄存器清除所有组合中断标志。

## 15. IIC 清除接收欠载中断寄存器 (IC\_CLR\_RX\_UNDER)

偏移地址：0x44

IC_CLR_RX_UNDER	[0]	RO
-----------------	-----	----

清除 RX\_UNDER 中断：读取此寄存器清除接收欠载中断

## 16. IIC 清除接收溢出中断寄存器 (IC\_CLR\_RX\_OVER)

偏移地址：0x48

IC_CLR_RX_UNDER	[0]	RO
-----------------	-----	----

清除 RX\_OVER 中断：读取此寄存器清除接收溢出中断

## 17. IIC 清除发送溢出中断寄存器 (IC\_CLR\_TX\_OVER)

偏移地址：0x4C

IC_CLR_RX_UNDER	[0]	RO
-----------------	-----	----

清除 TX\_OVER 中断：读取此寄存器清除接收溢出中断

## 18. IIC 清除读请求中断寄存器 (IC\_CLR\_RD\_REQ)

偏移地址：0x50

IC_CLR_RD_REQ	[0]	RO
---------------	-----	----

清除 RD\_REQ 中断：读取此寄存器清除从机读请求中断

## 19. IIC 清除发送中止中断寄存器 (IC\_CLR\_TX\_ABRT)

偏移地址：0x54

IC_CLR_TX_ABRT	[0]	RO
----------------	-----	----

清除 TX\_ABRT 中断：读取此寄存器清除发送中止中断并恢复 TX FIFO

## 20. IIC 清除接收完成中断寄存器 (IC\_CLR\_RX\_DONE)

偏移地址：0x58

IC_CLR_RX_DONE	[0]	RO
----------------	-----	----

清除 RX\_DONE 中断：读取此寄存器清除从机发送完成中断

## 21. IIC 清除总线活动中断寄存器 (IC\_CLR\_ACTIVITY)

偏移地址：0x5C

IC_CLR_ACTIVITY	[0]	RO
-----------------	-----	----

清除 ACTIVITY 中断：读取此寄存器清除总线活动标志

## 22. IIC 清除停止检测中断寄存器 (IC\_CLR\_STOP\_DET)

偏移地址：0x60

IC_CLR_STOP_DET	[0]	RO
-----------------	-----	----

清除 STOP\_DET 中断：读取此寄存器清除停止条件检测中断

### 23. IIC 清除接收完成中断寄存器 (IC\_CLR\_RX\_DONE)

偏移地址：0x64

IC_CLR_START_DET	[0]	RO
------------------	-----	----

清除 START\_DET 中断：读取此寄存器清除起始条件检测中断

### 24. IIC 使能控制寄存器 (IC\_ENABLE)

偏移地址：0x6C

IC_ENABLE	[0]	RO
-----------	-----	----

IIC 使能/失能：0x1 启用 I<sup>2</sup>C；0x0：禁用 I<sup>2</sup>C（清空 FIFO 并复位状态机）

### 25. IIC 发送 FIFO 数据量寄存器 (IC\_TXFLR)

偏移地址：0x74

IC_TXFLR	[7:0]	RO
----------	-------	----

发送 FIFO 数据量：实时反映发送缓冲区有效数据条目数量

### 26. IIC 清除接收完成中断寄存器 (IC\_CLR\_RX\_DONE)

偏移地址：0x78

IC_RXFLR	[7:0]	RO
----------	-------	----

接收 FIFO 数据量：实时反映接收缓冲区有效数据条目数量

## 6.8 WDT（看门狗定时器）

### 6.8.1 简介

WDT (Watchdog Timer, 看门狗定时器) 用于检测 and 解决软件错误导致的系统故障。当 WDT 计数器达到设定溢出时间且未被软件清零时，会触发中断或生成系统复位信号，确保芯片在软件异常时可恢复正常运行。

### 6.8.2 主要特性



- 计数器规格：20 位自由运行递增计数器；
- 溢出时间：可配置为 125 ms ~ 32 s（通过 `WDT_SET.WDT_CNT[7:0]` 寄存器调整）；
- 溢出动作：支持两种模式——中断（仅触发中断，需软件处理）、复位（触发系统复位，重置 CPU 及大部分外设）；
- 低功耗兼容性：支持在 Light Sleep 模式下工作，不支持 DeepSleep 模式（DeepSleep 模式下时钟关闭，计数器停止）。

## 6.8.3 功能描述

### [6.8.3.1] WDT 溢出中断模式

此模式下，WDT 按设定时间周期性触发中断，需在中断服务程序中清除中断标志，避免后续复位。配置步骤如下：

- 计算溢出时间：通过 `WDT_SET.WDT_CNT[7:0]` 配置，公式为：
$$\text{溢出时间} = \frac{32 - \text{配置值}}{8} \text{ s}$$
例：配置值为 0x00 时，溢出时间=32/8=4 s；配置值为 0xF8 时，溢出时间=(32-248)/8= -27 s（无效，需确保配置值≤256，溢出时间为正）。
- 使能中断模式：将 `WDT_CFG.WINT_INT_EN` 置 1，选择溢出后触发中断；
- 使能 NVIC 中断：在 NVIC 中断向量表中启用 WDT 中断（对应 IRQ16）；
- 启动 WDT：将 `WDT_CFG.WINT_EN` 置 1，启动计数器；
- 中断处理：中断触发后，读取 `WDT_INT_STATUS` 确认中断，通过 `WDT_INT_CLEAR` 清除中断标志，可重新配置 `WDT_SET.WDT_CNT` 更新溢出时间。

### [6.8.3.2] WDT 溢出复位模式

此模式下，WDT 计数器溢出后直接生成复位信号，复位 CPU 及大部分外设（部分外设可配置为不复位）。配置步骤如下：

- 计算溢出时间：同中断模式，通过 `WDT_SET.WDT_CNT[7:0]` 配置；
- 使能复位模式：将 `WDT_CFG.WINT_INT_EN` 置 0，选择溢出后触发复位；
- 启动 WDT：将 `WDT_CFG.WINT_EN` 置 1，启动计数器；

- 复位后处理：系统复位后，WDT\_RST\_STATUS 寄存器会置 1，需将 CPU\_RST\_CLR[4]置 1 清除该标志，避免误判复位原因。

## 6.8.4 寄存器映射

看门狗相关寄存器在 SYS\_CTRL 中。SYS\_CTRL 基地址：0x40000000，其他有关 SYS\_CTRL 的寄存器配置，在此基地址 加上相应的偏移地址。

### 1. 系统复位寄存器 (sys\_reset)

偏移地址：0x8

sys_reset	[31:0]	RW
-----------	--------	----

对应不同位写 1，则重置不同的模块。

bit0	复位系统，不包括aon域
bit1	复位CPU
bit2	复位看门狗
bit3	复位icache
bit4	复位qspi
bit5	复位radio
bit6	复位SDMA
bit7	复位otp_prg
bit8	复位AES
bit10	复位UART0
bit11	复位UART1
bit12	复位SPIM0
bit13	复位SPIM1
bit14	复位SPIS0
bit16	复位I2C0
bit19	复位GPIO
bit20	复位CAPTURE
bit21	复位TIMER_REG

bit22	复位rf_ramp
bit23	复位rf_reg
bit[31:24]	复位TIMER7~TIMER0

## 2. 看门狗定时器设置寄存器 (wdt\_set)

偏移地址：0x100

wdt_set	[7:0]	RW
---------	-------	----

用于设置 CPU 看门狗 (WDT) 计数器的高 8 位 [19:12]。看门狗定时长度计算 公式： WDT 定时时间 (秒) = 32 - (wdt\_cnt / 8)

## 3. 看门狗定时器配置寄存器 (wdt\_cfg)

偏移地址：0x104

RW	RW
[1]	[0]
wdt_int_en	wdt_en

bit0	wdt_en: 看门狗模块总使能位 0x0: 失能 0x1: 使能
bit1	wdt_int_en: 看门狗中断模式选择位 0x0: 看门狗超时而向系统请求软复位 0x1: 看门狗超时而向系统请求中断

## 4. 看门狗定时器状态寄存器 (wdt\_status)

偏移地址：0x108

0	RO	RO
[13]	[12]	[8:0]
wdt_rstn	wdt_rst_status	wdt_cnt_monitor

bit[8:0]	wdt_cnt_monitor: WDT 计数器实时监测值
----------	-------------------------------

bit[12]	wdt_rst_status: 看门狗复位记录位
bit[13]	wdt_rstn: 看门狗中断状态标志

## 6.9 Flash

### 6.9.1 简介

BL1820 仅 **Flash 版本 (BL1820FPG)** 集成 256 KB 片上 Flash，非 Flash 版本 (BL1820PPG/P8) 无片上 Flash，但均支持最大 64 MB 片外 Flash 扩展（通过 QSPI 接口，支持 XiP）。Flash 用于存储程序代码、配置数据，支持擦除、编程、读取操作。

### 6.9.2 片上 Flash（仅 BL1820FPG）

- 容量：256 KB；
- 地址范围：0x10000000 ~ 0x10040000（共 256 KB，16 KB/扇区）；
- 操作特性：
  - 擦除：支持扇区擦除（16 KB）、整片擦除；
  - 编程：支持字节编程、半字编程、字编程，编程前需先擦除对应扇区；
  - 保护：支持读保护（通过 OTP 配置），防止代码泄露。
- 通信引脚（复用 GPIO）：
  - QSPI\_CS: GPIO4（片选信号）；
  - QSPI\_CLK: GPIO6（时钟信号）；
  - QSPI\_D0: GPIO8（数据输入/输出 0）；
  - QSPI\_D1: GPIO9（数据输入/输出 1）；
  -

### 6.9.3 片外 Flash（所有版本可选）

- 接口：QSPI（四线/三线模式）；
- 最大容量：64 MB；
- 支持特性：
  - XiP（Execute-in-Place）：程序可直接从片外 Flash 执行，无需加载到 SRAM；
  - 时钟速率：最高 24 MHz（由 QSPI\_CLK 控制）；

## 7. 电气特性（续）

### 7.1 绝对最大额定值

绝对最大额定值是指短时间暴露并不会导致器件永久损伤的极限值，长时间在绝对最高额定值下使用可能会影响设备的可靠性。这只是一个器件过压级别，不保证在这些过压级别或超出推荐工作条件范围外器件的正常工作。

绝对最大额定值表：

Parameter	Symbol	Condition	Min	Typ	Max	Unit
VDD/VDDIO 供电电压	VDDMAX	-	-0.3	-	3.9	V
XTALIN/XTALOUT 引脚电压	VXTAL		-0.3	-	1.3	V
GPIO 引脚电压	VGPIOMAX	VDDIO $\leq$ 3.6 V	-0.3	-	VDDIO+0.3	V
		VDDIO > 3.6 V	-0.3	-	3.9	V
RFIO 引脚输入功率	PRFMAX	-	-	-	10	dBm
RFIO 引脚直流电压	VRFMAX	非零直流电压会损坏 RF 端口	0	0	0	V
VDD 引脚输入电流	IVDDMAX	-	-	-	100	mA

Parameter	Symbol	Condition	Min	Typ	Max	Unit
VDDIO 引脚输入电流	IVDDIOMAX	-	-	-	100	mA
VSS 引脚输出电流	IVSSMAX	-	-	-	200	mA
单路 I/O 引脚电流	IIOMAX	源极/漏极电流	-	-	20	mA
所有 I/O 引脚总电流	IIOMAX_TOTAL	源极/漏极总电流	-	-	90	mA
结温	TJMAX	-	-	-	125	°C
存储温度	TSMAX	-	-40	-	150	°C

## 7.2 推荐工作条件

推荐工作条件:

Parameter	Symbol	Condition	Min	Typ	Max	Unit
VDD 供电电压	VDD	$VDD \geq VDDIO$ (否则会产生泄漏电流)	$\max(1.7, VDDIO)$	3.0	3.6	V
VDDIO 供电电压	VDDIO	由外部芯片供电 (非 VDD)	1.7	3.0	3.6	V
工作温度	TA	-	-40	25	85	°C

## 7.3 复位特性

Parameter	Symbol	Condition	Min	Typ	Max	Unit
上电复位 (POR) 上升阈值电压	VPOR	电源电压上升沿	1.60	1.70	1.80	V
掉电复位 (PDR) 下降阈值电压	VPDR	电源电压下降沿	1.52	1.62	1.72	V

Parameter	Symbol	Condition	Min	Typ	Max	Unit
压						
POR 延迟时间	TPOR	上电后复位释放时间	23	30	37	ms
PDR 延迟时间	TPDR	掉电后复位触发时间	19	23	28	μs

## 7.4 射频特性

### 7.1.1 发射机规格 (TA=25°C, VDD=3.0 V, 传导模式)

Parameter	Symbol	Condition	Min	Typ	Max	Unit
频率范围	f_TX	2.4G 频段	2400	-	2483	MHz
输出功率	P_out	全功率范围	-20	-	10	dBm
功率控制步长	P_step	批次间功率校准	-	1.0	-	dB
2 次谐波功率	P_2harm	0 dBm 输出	-	-45.0	-	dBm
		4 dBm 输出	-	-40.0	-	dBm
		10 dBm 输出	-	-25.5	-	dBm
杂散发射（4 dBm 输出）	P_spur	30 MHz ~ 1000 MHz	-	-43.7	-	dBm
		1 GHz ~ 12.75 GHz	-	-31.0	-	dBm
		47 MHz ~ 74 MHz / 87.5 MHz ~ 108 MHz / 174 MHz ~ 230 MHz	-	-75	-	dBm
		470 MHz ~ 862 MHz	-	-44.0	-	dBm
平均频率偏差（1 Mbps BLE）	$\Delta f_{l\_avg\_1M}$	0 dBm 输出	244	248.6	251	KHz
		4 dBm 输出	242	248.1	253	KHz
		10 dBm 输出	244	249.3	257	KHz
邻道功率（1 Mbps BLE，2 MHz 偏移）	P_adj_1M	0 dBm 输出	-54	-51.4	-47	dBm
		4 dBm 输出	-50	-47.4	-44	dBm
		10 dBm 输出	-42	-37.4	-32	dBm

### 7.1.2 接收机规格（TA=25°C，VDD=3.0 V，传导模式，BER=0.1%）

Parameter	Symbol	Condition	Min	Typ	Max	Unit
-----------	--------	-----------	-----	-----	-----	------



Parameter	Symbol	Condition	Min	Typ	Max	Unit
频率范围	f_RX	2.4G 频段	2400	-	2483	MHz
接收灵敏度 (1 Mbps BLE)	PSENS_1M	0.1 % BER	-94	-96	-	dBm
同频信道 C/I 比	C/ICO_1M	0.1 % BER	3.8	-	21	dB
邻道 C/I 比 (1 MHz 偏移)	C/I1_1M	0.1 % BER	-23.6	-	15	dB
邻道 C/I 比 (2 MHz 偏移)	C/I2_1M	0.1 % BER	-26.8	-	-17	dB
最大输入信号电平???	PIN_MAX_1M	0.1 % BER	-10	-	0.0	dBm
带外阻塞 (30 MHz ~ 2000 MHz)	OOB	-	-30	-	-	dBm
带外阻塞 (2003 MHz ~ 2399 MHz/2484 MHz ~ 2997 MHz)	OOB	-	-35	-	-	dBm
带外阻塞 (3000 MHz ~ 12.75 GHz)	OOB	-	-30	-	-	dBm

## 7.2 GPIO 特性 (TA=25°C, VDDIO=3.0 V)

Parameter	Symbol	Condition	Min	Typ	Max	Unit
输入低电平	V_IL	-	-	-	0.3×VDDIO	V
输入高电平	V_IH	-	0.7×VDDIO	-	-	V
输出低电平(灌电流 20 mA)	V_OL	VDDIO=3 V	-	-	0.5	V
输出低电平 (灌电流 8 mA)	V_OL	VDDIO=1.7 V	-	-	0.4	V
输出高电平(拉电流 20 mA)	V_OH	VDDIO=3 V	2.5	-	-	V
输出高电平 (拉电流 8 mA)	V_OH	VDDIO=1.7 V	1.3	-	-	V

Parameter	Symbol	Condition	Min	Typ	Max	Unit
上拉电阻	R_PU	-	-	25	-	kΩ
下拉电阻	R_PD	-	-	25	-	kΩ
泄漏电流	I_LEAK _IO	-	-	1.9	-	nA

### 7.3 通用 ADC 特性 (TA=25°C, VIO=3.3 V)

Parameter	Symbol	Condition	Min	Typ	Max	Unit
采样频率	F_S	-	-	1	3	MHz
输入测量范围 (普通模式)	V_IN	-	0	-	V_IO	V
输入测量范围 (PGA 模式)	V_IN	-	0	-	0.9	V
PGA 增益范围	G_PGA	PGA 模式	0	-	27	dB
信噪失真比 (普通模式)	SINAD	Fs=1 MHz, VIO=3.3 V	-	60	-	dB
信噪失真比 (PGA 模式)	SINAD	Fs=1 MHz, VIO=3.3 V	50	57	-	dB
有效位数 (普通模式)	ENOB	Fs=1 MHz, VIO=3.3 V	9.2	9.5	-	bits
有效位数 (PGA 模式)	ENOB	Fs=1 MHz, VIO=3.3 V	8	9.0	-	bits
微分非线性	DNL	-	-	-	1	LSB
积分非线性	INL	-	-	-	2	LSB
输入采样电容	C_S	-	-	256	-	fF

Parameter	Symbol	Condition	Min	Typ	Max	Unit
内部参考电压	V_REF	25 mV 步长可编程	-	0.9	-	V

## 7.4 时钟特性

### 7.4.1 24 MHz 外部时钟源 (XTALIN 输入)

如果 XTALIN 和 XTALOUT 引脚没有连接晶体，晶体振荡器可以从 XTALIN 引脚接受 外部时钟源信号进行整形和放大提供给芯片内部使用。  
外部时钟信号波形可以为正弦波和 方波。

24 MHz 外部时钟源参数:

Parameter	Symbol	Condition	Min	Typ	Max	Unit
频率	f_HSE_EXT	BLE 模式下必须为 24 MHz	-	24	-	MHz
输入电压	V_P_HSE_EXT	通过 100 nF 隔直电容输入	0.5	-	1.3	V
相位噪声 (100 Hz 偏移)	PN_HSE_EXT_BLE	BLE 兼容模式	-	-115	-109	dBc/Hz
相位噪声 (1 kHz 偏移)	PN_HSE_EXT_BLE	-	-	-125	-119	dBc/Hz
相位噪声 (10 kHz 偏移)	PN_HSE_EXT_BLE	-	-	-135	-129	dBc/Hz
相位噪声 (100 kHz 偏移)	PN_HSE_EXT_BLE	-	-	-142	-136	dBc/Hz
相位噪声 (1 MHz 偏移)	PN_HSE_EXT_BLE	-	-	-146	-140	dBc/Hz

### 7.4.2 24 MHz 晶体振荡器 (XTALIN/XTALOUT)

24 MHz 晶体振荡器参数:

Parameter	Symbol	Condition	Min	Typ	Max	Unit
晶体频率	f_HSE	-	-	24	-	MHz
等效串联电阻 (ESR)	ESR_HSE	-	-	50	80	$\Omega$
负载电容	C_L_HSE	片内可编程电容	-	-	14	pF
频率容差	$\Delta f_{HSE}$	温度、工艺、老化影响	-50	-	50	ppm
起振时间	T_ST_HSE	频率稳定在 $\pm 50$ ppm 内	-	1.0	-	ms

### 7.4.3 24 MHz RC 振荡器 (内部)

Parameter	Symbol	Condition	Min	Typ	Max	Unit
频率	f_HSI	-	-	24	-	MHz
频率精度	f_HSI_ACC	-	-2	-	2	%
温度漂移	$\Delta f_{HSI\_TEMP}$	-40°C ~ 85°C	-3	-	3	%
电流消耗	I_HSI	启动后	-	67	-	$\mu A$
起振时间	T_HSI_ST	含 LDO, 频率稳定在 $\pm 50$ ppm 内 (BLE 模式)	-	3.6	5	$\mu s$

### 7.4.4 32 kHz RC 振荡器 (内部)

Parameter	Symbol	Condition	Min	Typ	Max	Unit
频率	f_LSI	-	-	32.768	-	kHz
频率精度	f_LSI_ACC	±0.5°C恒温，每 8 秒校准，平均间隔>7.5 ms	-700	-	700	ppm
起振时间	T_LSI_ST	频率稳定在 99%稳定值内（含 LDO）	-	500	-	μs

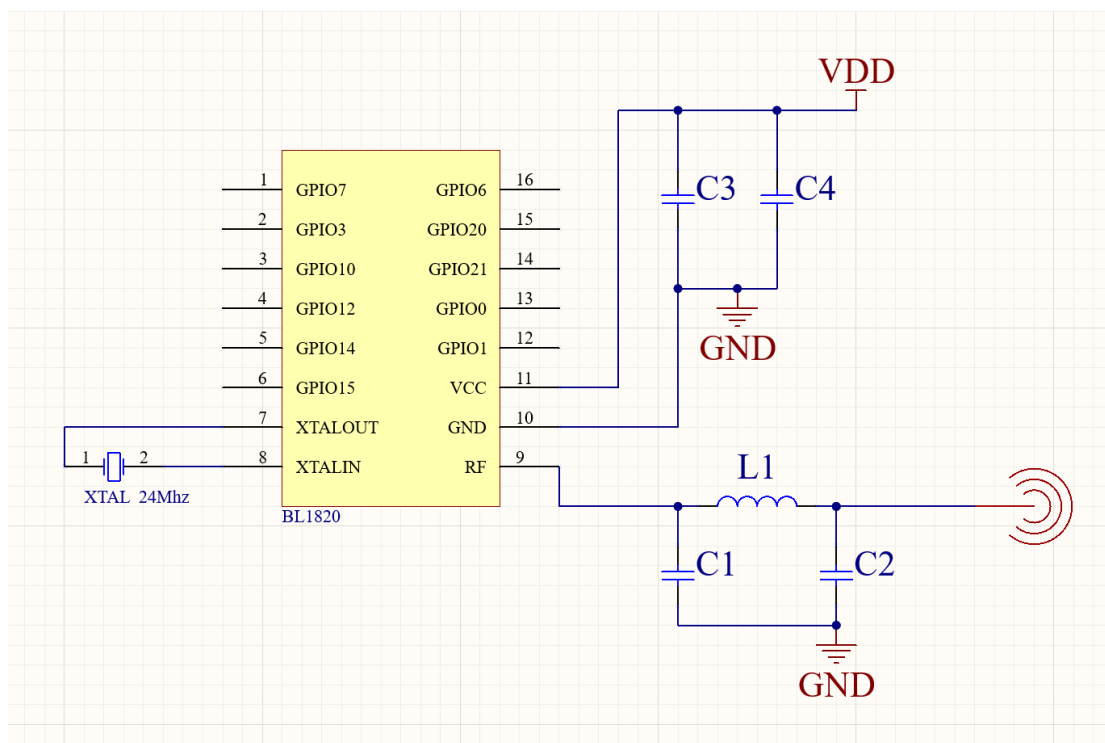
## 8. 参考设计

### 8.1 最小系统设计

BL1820 最小系统需包含以下关键电路，适用于快速原型验证：

- 电源电路：
  1. VDD：1.7 V ~ 3.6 V 供电，建议使用 3.3 V 线性电源，并联 10 μF 电解电容 +0.1 μF 陶瓷电容滤波；
  2. VDDIO：与 VDD 共压，需单独并联 0.1 μF 陶瓷电容滤波；
  3. GND：确保与电源地、射频地单点连接，减少干扰。
- 4. 时钟电路：

- 外部 24 MHz 晶体：XTALIN（引脚 8）、XTALOUT（引脚 7）连接 24 MHz $\pm$ 50 ppm 晶体，并联 12 pF ~ 14 pF 匹配电容（根据晶体负载电容选择）；
- 无晶体场景：XTALIN 引脚输入 24 MHz 方波/正弦波（0.5 V ~ 1.3 V），XTALOUT 悬空。
- 射频电路：
  - 2. RFIO：直接连接 PCB 天线或外接 SMA 接头，天线需远离电源、时钟电路，减少干扰；无需外部匹配网络（芯片内部集成 T/R 开关）。
- 复位电路：
  - RSTN：低电平复位，建议通过 10 k $\Omega$  上拉电阻接 VDDIO，复位按钮接地（可选，用于手动复位）。
- 调试电路：
  - 2. SWD 接口：GPIO0（TCLK）、GPIO1（TMS）通过 100  $\Omega$  电阻连接调试器，支持程序下载与调试。



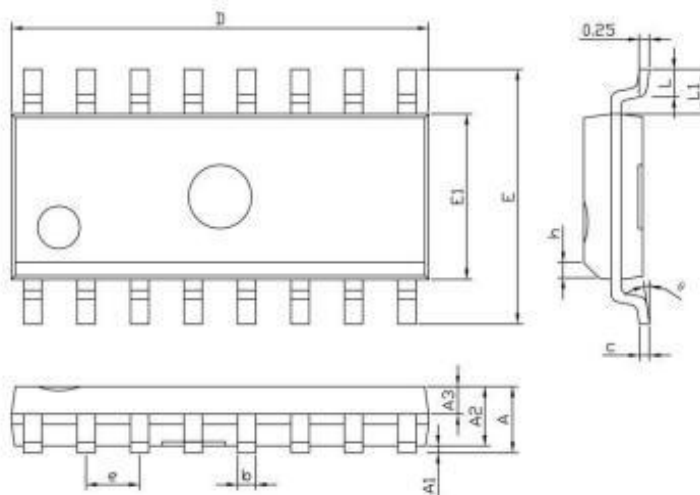
## 8.2 典型应用电路示例（无线传感器节点）

- 4. 功能：采集温湿度数据（通过 IIC 传感器），通过 2.4G 无线传输至网关；

- **外设连接：**
  - IIC 传感器（如 SHT30）：SDA（GPIO7）、SCL（GPIO6），并联 4.7 k $\Omega$  上拉电阻；
  - 按键：GPIO2 接按键（下拉电阻），用于唤醒/配置；
  - LED 指示灯：GPIO3 接 LED（串联 1 k $\Omega$  限流电阻），用于状态指示；
- **功耗优化：**
  - 空闲时进入 RTC 模式（保留 16K RAM，功耗 3  $\mu$ A），通过 RTC 定时唤醒采集数据；
  - 无线传输完成后立即关闭 2.4G 射频模块，仅保留 MCU 核心与 RTC。

## 9. 封装信息

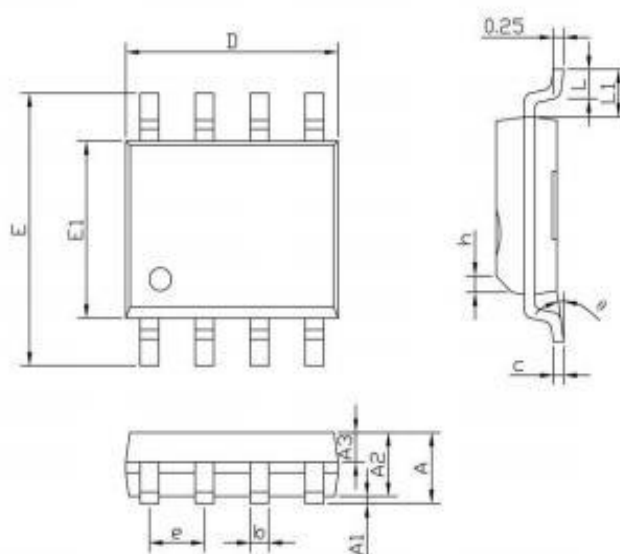
### 9.1 SOP16 封装 (BL1820PPG/FPG)



COMMON DIMENSIONS (UNITS OF MEASURE=MILLIMETER)			
SYMBOL	MIN	NOM	MAX
A	-	-	1.75
*A1	0.10	-	0.225
A2	1.30	1.40	1.50
A3	0.60	0.65	0.70
b	0.39	-	0.47
c	0.20	-	0.24
D	9.80	9.90	10.00
*E	5.80	6.00	6.20
*E1	3.80	3.90	4.00
*e	1.24	1.27	1.30
h	0.30	0.40	0.50
L	0.50	-	0.80
*L1	0.95	1.05	1.15
∅	0*	-	0*

备注：1.标注“\*”项为测量尺寸

### 9.2 SOP8 封装 (BL1820P8)



COMMON DIMENSIONS (UNITS OF MEASURE=MILLIMETER)			
SYMBOL	MIN	NOM	MAX
A	-	-	1.75
*A1	0.05	-	0.20
A2	1.30	1.40	1.50
A3	0.60	0.65	0.70
b	0.39	-	0.47
c	0.20	-	0.24
D	4.80	4.90	5.00
*E	5.80	6.00	6.20
*E1	3.80	3.90	4.00
*e	1.24	1.27	1.30
h	0.30	0.4	0.50
L	0.50	-	0.80
*L1	1.00	1.05	1.10
∅	0*	-	0*

备注：1.标注“\*”项为测量尺寸



## 10. 缩略语

Abbreviation	Full Name	中文释义
SoC	System on Chip	片上系统
RISC-V	Reduced Instruction Set Computing-V	精简指令集架构-V 版本
MCU	Microcontroller Unit	微控制器单元
UART	Universal Asynchronous Receiver/Transmitter	通用异步收发器
SPI	Serial Peripheral Interface	串行外设接口
IIC	Inter-Integrated Circuit	集成电路间总线
DMA	Direct Memory Access	直接存储器访问
ADC	Analog-to-Digital Converter	模数转换器
DAC	Digital-to-Analog Converter	数模转换器 (BL1820 无此模块)
WDT	Watchdog Timer	看门狗定时器
RTC	Real-Time Clock	实时时钟
OTP	One-Time Programmable	一次性可编程存储器
LDO	Low-Dropout Regulator	低压差线性稳压器
POR	Power-On Reset	上电复位
BOR	Brown-Out Reset	掉电复位
LVD	Low Voltage Detector	低电压检测器

<b>Abbreviation</b>	<b>Full Name</b>	<b>中文释义</b>
RSSI	Received Signal Strength Indicator	接收信号强度指示
XiP	Execute-in-Place	原地执行（从 Flash 直接运行程序）
NVIC	Nested Vectored Interrupt Controller	嵌套向量中断控制器