

# VT2303R

## 产品描述

VT2303R 为高性能、低成本 OOK/GFSK 收发器，可用于 315MHz、433MHz、470MHz、868MHz 和 915MHz 频段的无线应用。该芯片内置一个高集成的 sub-1GHz 收发器和一个基带调制解调器，可编程 GFSK 数据速率范围是 2kbps~250kbps，OOK 数据速率范围是 0.5kbps~20kbps。数据处理特性包括 64 字节 TX/RX FIFO 和数据包处理如 CRC 生成、正向纠错、数据白化和曼彻斯特编码。

VT2303R 针对低功耗应用优化性能。在 433MHz 频段，其 RX 模式工作电流为 5.9mA，提供+20dBm TX 输出功率时的电流损耗为 71mA。低功耗低中频接收器可在 433MHz 频段数据速率为 2kbps 时实现-120dBm 的灵敏度。E 类功率放大器可在 433/868MHz 频段提供高达 +20dBm 输出功率。完全内置的小数分频合成器所支持的频率范围宽，分辨率高。芯片内部还内置有回路滤波器和晶振负载电容，减少外部元器件需求。

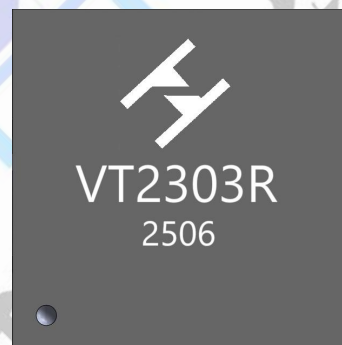
外部主控 MCU 可通过一个 3 线或 4 线的 SPI 接口访问 VT2303R。此芯片支持简短的 Strobe 命令，可在维持无线通信链接的同时减少主控 MCU 负载。

其它链接层特性包含 RSSI 用于通道评估、自动应答和自动重发、WOT 和 WOR 等等，便于实现基于微控制器的 ISM 频段无线链接应用。

VT2303R 采样 QFN-16(3mm\*3mm)封装。

## 应用领域

遥控门禁系统  
胎压监测设备  
遥控风扇、照明开关  
无线传感数据传输  
玩具遥控



## 特性

- 频率带宽: 315/433/470/868/915MHz
- OOK/GFSK 调制
- 支持 3 线或 4 线 SPI 接口
- 输入电压范围宽: 1.8V~3.6V
- 可编程数据速率:
  - OOK: 0.5kbps~20kbps
  - GFSK: 2kbps~250kbps
- 可编程 TX 输出功率: 0dBm~20dBm
- 低电流损耗
  - Deep Sleep 模式电流 0.4 $\mu$ A, 支持数据保存
  - RX 电流损耗 (AGC 开启 & 2kbps 数据速率) @ 433.92MHz: 5.9mA
  - RX 电流损耗 (AGC 开启 & 2kbps 数据速率) @ 868.3MHz: 6.9mA
  - TX 电流损耗 @ 433.92MHz: 43mA @ 13dBm POUT
  - TX 电流损耗 @ 868.3MHz: 47mA @ 13dBm POUT
- 高 RX 灵敏度 (433.92MHz: -120dBm @ 2kbps 无线数据速率)
- 高 RX 灵敏度 (868.3MHz): -120dBm @ 2kbps 无线数据速率
- 支持低成本 16MHz 晶振, 内置负载电容
- 可编程数字通道滤波器, 实现各种数据速率条件下最佳性能
- AGC (自动增益控制) 功能实现宽输入范围, 高达 +10dBm
- AFC (自动频率补偿) 功能用于补偿晶振老化造成的频漂
- 片上低功耗 RC 振荡器用于 WOR (从 RX 唤醒) 和 WOT (从 TX 唤醒) 功能
- 片上 8-bit RSSI (接收信号强度指示)
- 物理 TX/RX FIFO 缓冲器: TX 64 字节, RX 64 字节
- Simple FIFO/Block FIFO/Extend FIFO (高达 255 字节)/Infinite FIFO 模式
- 可编程载波检测阈值
- FIFO 模式和 Direct 模式支持帧同步识别
- 数据包处理
  - FEC (正向纠错)
  - 数据白化
  - 曼彻斯特编码
  - CRC-16 校验
- ATR (自动发送/接收)
  - 自动
  - 地址过滤
- 封装类型: 16-pin QFN (3x3mm)

## 目录

产品描述 .....	1
应用领域 .....	1
特性 .....	2
1、脚位定义及说明 .....	1
1.1 脚位示意图 .....	1
1.2 脚位说明 .....	1
2、绝对最大额定值 .....	2
3、工作条件 .....	2
4、电特性参数 .....	3
5、功能描述 .....	5
5.1 框架图 .....	5
5.2 晶体 .....	6
5.3 系统时钟 .....	6
5.4 频率合成器 .....	7
5.5 RF 接收器 .....	7
5.6 SPI 接口 .....	8
6、协议引擎 .....	10
6.1 特性 .....	10
6.2 协议引擎概述 .....	10
6.3 协议引擎封包格式 .....	11
7、射频功能操作 .....	14
7.1 FIFO 模式 .....	14
7.2 Direct 模式 .....	24
8、寄存器说明 .....	26
8.1 寄存器结构 .....	26
8.2 控制寄存器访问 .....	26
8.3 公用区控制寄存器 .....	26
8.4 Bank0 控制寄存器 .....	40
8.5 Bank1 控制寄存器 .....	50
8.6 Bank2 控制寄存器 .....	56
9、应用电路图 .....	57
10、封装外形 .....	58

## 1、脚位定义及说明

### 1.1 脚位示意图

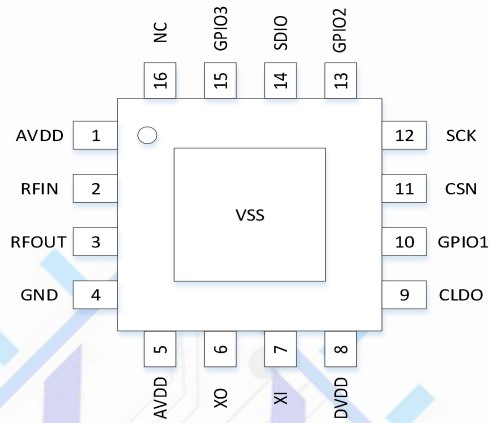


图 1 VT2303R 管脚示意图

### 1.2 脚位说明

表 1 VT2303R 管脚描述

管脚编号	管脚名称	I/O	管脚功能描述
1	AVDD	P	模拟正电源
2	RFIN	I	RF LAN 输入
3	RFOUT	O	RF 功率放大器输出
4	GND	P	接地
5	AVDD	P	模拟正电源
6	XO	A	晶振输出
7	XI	A	晶振输入
8	DVDD	P	数字正电源
9	CLDO	P	LDO 输出, 连接旁路电容
10	GPIO1	I/O	多功能I/O 1
11	CSN	I/O	SPI 芯片选择输入, 低有效
12	SCK	I/O	SPI 时钟输入
13	GPIO2	I/O	多功能I/O 2
14	SDIO	I/O	SPI 数据输入/ 输出
15	GPIO3	I/O	多功能I/O 3
16	NC	--	未连接, 保持浮空
—	VSS	—	裸露焊盘, 接地

## 2、绝对最大额定值

表 2 绝对最大额定值

参数	符号	条件	最小	最大	单位
电源电压	$V_{DD}$		-0.3	3.6	V
接口电压	$V_{IN}$		-0.3	$V_{DD} + 0.3$	V
结温	$T_J$		-40	125	°C
储藏温度	$T_{STG}$		-40	125	°C
焊接温度	$T_{SDR}$	持续时间不超过 30 秒		255	°C
ESD 等级		人体模型(HBM)	-3	3	kV
栓锁电流		@ 85 °C	-100	100	mA

## 3、工作条件

表 3 推荐工作条件

参数	符号	条件	最小	最大	单位	参数
运行电源电压	$V_{DD}$	-40°C到+85°C	1.8		3.6	V
运行温度	$T_{OP}$		-40		85	°C
电源电压斜率	$V_{SL}$		1			mV/us

## 4、电特性参数

表 4 直流电气参数

参数	符号	条件	最小	典型	最大	单位
深度睡眠电流	$I_{DS}$	Deep Sleep 模式		0.4	1.0	$\mu A$
轻睡眠电流	$I_{DS}$	Light Sleep 模式		0.6		mA
空闲电流	$I_{IL}$	Idle 模式, 晶振使能		1.6		$\mu A$
	$I_{SB}$	Standby 模式		3.9		mA
接收电流	315 频段	RX 模式@2kbps		5.5		mA
		RX 模式@250kbps		6.1		mA
	433 频段	RX 模式@2kbps		5.9		mA
		RX 模式@250kbps		6.5		mA
	868 频段	RX 模式@2kbps		6.9		mA
		RX 模式@250kbps		7.6		mA
	915 频段	RX 模式@2kbps		6.7		mA
		RX 模式@250kbps		7.5		mA
发射电流	315 频段	TX 模式@0dBm		17		mA
		TX 模式@10dBm		30		mA
		TX 模式@20dBm		74		mA
	433 频段	TX 模式@0dBm		19		mA
		TX 模式@10dBm		33		mA
		TX 模式@20dBm		71		mA
	868 频段	TX 模式@0dBm		19		mA
		TX 模式@10dBm		35		mA
		TX 模式@20dBm		88		mA
	915 频段	TX 模式@0dBm		17		mA
		TX 模式@10dBm		31		mA
		TX 模式@20dBm		84		mA



表 5 交流电气参数 1

参数	符号	条件	最小	典型	最大	单位
RF 频段	$F_{RF}$	315MHz 频段	290	—	335	MHz
		433MHz 频段	415	—	490	MHz
		470~510MHz 频段	470	—	510	MHz
		868MHz 频段	830	—	1000	MHz
		915MHz 频段	870	—	1050	MHz
数据速率	$D_R$	OOK 调制	0.5		20	kbps
		GFSK 调制	2		250	kbps
输出功率	$P_{OUT}$		0		20	dBm
发射启动时间	$T_{LSTX}$	Light Sleep 模式到 TX 模式		120		us
发射杂散	$P_{SE}$	$f < 1GHz$			-36	dBm
		$47MHz < f < 74MHz$			-54	dBm
		$87.5MHz < f < 118MHz$			-54	dBm
		$174MHz < f < 230MHz$			-54	dBm
		$470MHz < f < 862MHz$			-54	dBm
		二次谐波, 三次谐波			-30	dBm
接收启动时间	$T_{LSRX}$	Light Sleep 模式到 RX 模式		150		us
接收灵敏度 @BER=0.1%	$P_{SENS}$ @315MHz	OOK 模式, 2kbps		-119		dBm
		2kbps ( $f_{DEV}=8kHz$ )		-120		dBm
		10kbps ( $f_{DEV}=40kHz$ )		-112		dBm
		50kbps ( $f_{DEV}=18.75kHz$ )		-110		dBm
		125kbps ( $f_{DEV}=46.875kHz$ )		-106		dBm
		250kbps ( $f_{DEV}=93.75kHz$ )		-103		dBm
	$P_{SENS}$ @433MHz	OOK 模式, 2kbps		-119		dBm
		2kbps ( $f_{DEV}=8kHz$ )		-120		dBm
		10kbps ( $f_{DEV}=40kHz$ )		-113		dBm
		50kbps ( $f_{DEV}=18.75kHz$ )		-110		dBm
		125kbps ( $f_{DEV}=46.875kHz$ )		-106		dBm
		250kbps ( $f_{DEV}=93.75kHz$ )		-103		dBm
	$P_{SENS}$ @868MHz	OOK 模式, 2kbps		-118		dBm
		2kbps ( $f_{DEV}=8kHz$ )		-120		dBm
		10kbps ( $f_{DEV}=40kHz$ )		-112		dBm
		50kbps ( $f_{DEV}=18.75kHz$ )		-109		dBm
		125kbps ( $f_{DEV}=46.875kHz$ )		-105		dBm
		250kbps ( $f_{DEV}=93.75kHz$ )		-102		dBm
	$P_{SENS}$ @915MHz	OOK 模式, 2kbps		-118		dBm
		2kbps ( $f_{DEV}=8kHz$ )		-120		dBm
		10kbps ( $f_{DEV}=40kHz$ )		-112		dBm
		50kbps ( $f_{DEV}=18.75kHz$ )		-109		dBm
		125kbps ( $f_{DEV}=46.875kHz$ )		-105		dBm
		250kbps ( $f_{DEV}=93.75kHz$ )		-102		dBm

表 6 交流电气参数 2

参数	符号	条件	最小	典型	最大	单位
最大输入功率 @BER=0.1%	$P_{INMAX}$				10	dBm
镜像抑制	IR			25		dBm
RSSI 量程	$P_{RSSI}$	AGC 使能	-110		-10	us
晶振频率	$f_{XTAL}$		—	16	—	MHz
晶振等效串联电阻	ESR		—	—	100	$\Omega$
晶振电容负载	$C_{LOAD}$		8	12	16	pF
晶振容差	TOL		-20	—	+20	ppm
晶振启动时间	$t_{SU}$	49US XO	—	1	—	ms
		3225SMD XO	—	2	—	
SCK 频率	$f_{SCK}$		—	4	—	MHz
SCK 高电平时间	$t_{SCKH}$		62.5	—	—	ns
SCK 低电平时间	$t_{SCKL}$		62.5	—	—	ns
SDIO 输入设置时间	$t_{S\_SDIO}$		20	—	—	ns
SDIO 输入保持时间	$t_{H\_SDIO}$		20	—	—	ns
CSN 有效到 SCK 有效时间间隔	$t_{S\_CSN}$		30	—	—	ns
SCK 无效 CSN 无效时间间隔	$t_{H\_CSN}$		30	—	—	ns

## 5、功能描述

### 5.1 框架图

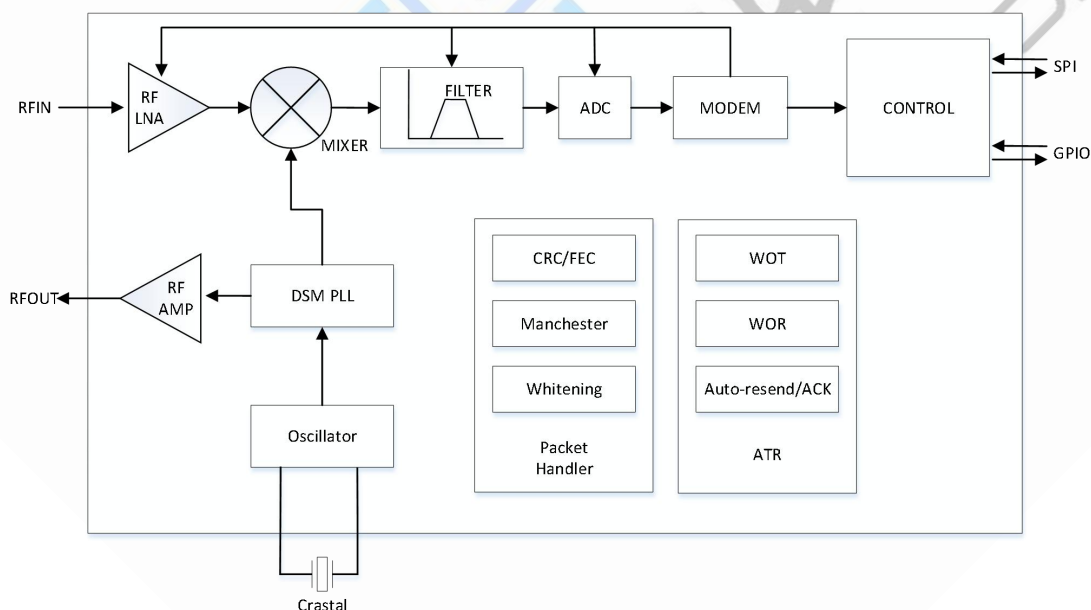


图 2 VT2303R 内部电路框架图



VT2303R 是一款数模混合设计的一体化接收机。该芯片集成 GFSK 接收机和发射机，仅需要外接少量器件即可实现稳定的数据传输功能。

## 5.2 晶体

为保证参考时钟的稳定性和相位噪声，VT2303R 采用两个管脚驱动晶体振荡器。晶体振荡所需的负载电容集成于芯片内，无需外置电容。推荐使用精度在为 $\pm 20$  ppm，等效电阻小于  $100\ \Omega$ ，负载电容为  $12\ \text{pF}$ ， $16\text{MHz}$  的晶体。需要注意的是，由于不同封装规格的晶体存在着寄生电容差异，请用户选用晶体时注意评估，避免由于晶体震荡频率偏离目标值过大而引起接收机性能降低。

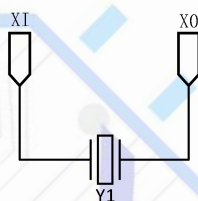


图 3 VT2303R 晶振驱动电路图

## 5.3 系统时钟

VT2303R 的主要系统时钟来自 XTAL 晶振。各种功能模块的所有内部操作时钟都来自此晶振。

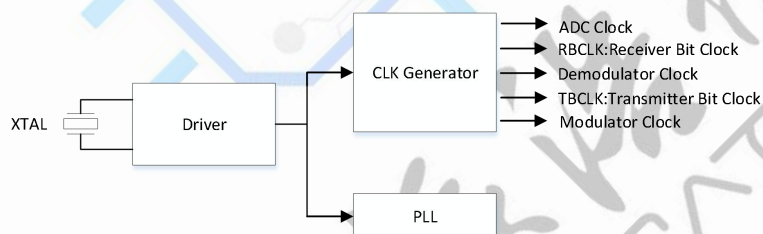


图 4 系统时钟示意图

## 5.4 频率合成器

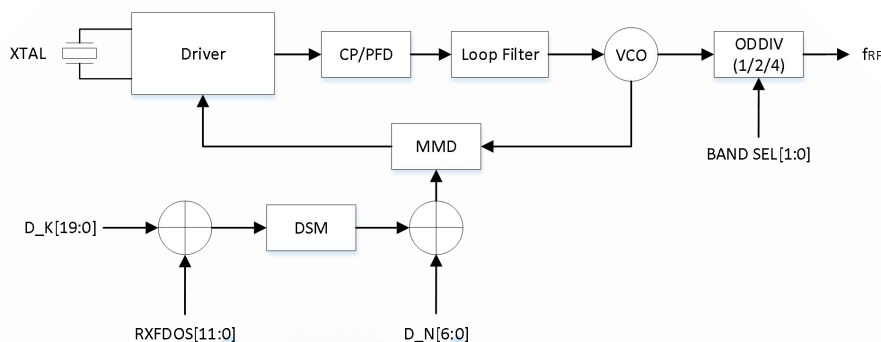


图 5 频率合成器结构示意图

RF 收发器频率由一个高分辨率的小数 N Delta Sigma 频率合成器产生。通过合理设置  $D_N[6:0]$  和  $D_K[19:0]$ , 可产生一个低噪声 LO 频率, 适用于各种无线电规范标准, 如 ETSI EN 和 FCC 等等。

$$D_N[6:0] = \text{FLOOR}\left(\frac{f_{RF} * \text{ODDIV}}{f_{XTAL}}\right)$$

$$D_K[19:0] = \text{FLOOR}\left(\frac{f_{RF} * \text{ODDIV}}{f_{XTAL}} - D_N[6:0] * 2^{20}\right)$$

$$\text{RXFDOS}[11:0] = \text{FLOOR}\left(\frac{f_{IF}}{f_{XTAL}} * 2^{17}\right)$$

## 5.5 RF 接收器

VT2303R 采用完全集成的低中频接收器架构。接收到的 RF 信号先经过一个低噪声放大器(LNA)进行放大, 接着通过一个正交混频器将频率向下转换为中频。混频器输出信号经过通道选择滤波器进行滤波, 滤除不必要的带外干扰和图像信号。经过滤波后, 中频信号经过一个模拟可编程增益放大器(PGA)进行放大。接着使用一个 10 位  $\Sigma\Delta\text{A/D}$  转换器将放大后的中频信号数字化。

VT2303R 内置一个自动增益控制(AGC)单元, 可根据数字调制解调器产生的 RSSI 来调节接收器增益。此 AGC 功能使得 VT2303R 可在灵敏度级别至 +10dBm 输入功率范围之内工作。

VT2303R 采用完全内置的小数 N 分频合成器, 包含 RFVCO、回路滤波器、内带负载电容的数字控制晶振。在 PCB 上装 VCO 负载电感可降低 VCO 谐振频率, 从而实现低至 5.4mA 的 RX 模式电流损耗。小数 N 分频合成器架构允许用户将其潜在应用扩展至更广泛的频率范围。

传输会话采用 VCO 直接调制架构。与传统的直接上变频发送器不同, 这里利用小数 N 分频合成器的优势直接将 GFSK 调制信号接入 VCO。因此, 布局面积和电流损耗都比传统直接上变频发送器的小很多。精细的分辨率可产生低 FSK 误差的 GFSK 信号。调制好的信号接入一个 E 类功率放大器(PA), 最大输出功率可达 +20dBm。

## 5.6 SPI 接口

VT2303R 通过一个3线SPI接口(CSN, SCK, SDIO)或一个4线串行接口(SDO位于 GPIO<sub>On</sub>)与主控 MCU 通信, 数据速率高达 4Mbps。一笔 SPI 传输其实就是一个 $(8+8 \times n)$ 位的序列, 包含一个 8 位的命令和  $n \times 8$  位数据, 其中  $n$  可以是 0 或者任何自然数。若  $n$  大于地址边界, 则会返回地址 0。主控 MCU 要访问 VT2303R 时应将 CSN(SPI 芯片选择)引脚拉低。用户可通过 SPI 接口访问控制寄存器并发出 Strobe 命令。当写数据到 RF 芯片时, SPI 数据会在 SCK 信号上升沿时存入对应寄存器。若从 RF 芯片寄存器读取数据, 当输入目标寄存器地址后, 每一个位数据会在 SCK 信号下降沿时传出。

有两种命令, 一种是只有 1 个字节的命令, 即 Cmd0; 另一种是 1 个字节命令加  $n$  个字节数据, 即 CmdD。

表 7 SPI 命令格式

命令 (8 位)								数据 (8 位)							
C7	C6	C5	C4	C3	C2	C1	C0	D7	D6	D5	D4	D3	D2	D1	D0

表 8 命令说明

	C6	C5	C4	C3	C2	C1	C0	说明	Cmd0	CmdD
0	1	A5	A4	A3	A2	A1	A0	写入控制寄存器		✓
1	1	A5	A4	A3	A2	A1	A0	读取控制寄存器		✓
0	0	1	x	x	x	B1	B0	设置寄存器存储区	✓	
0	0	0	1	x	x	x	0	写同步码命令		✓
1	0	0	1	x	x	x	0	读同步码命令		✓
0	0	0	1	x	x	x	1	TX FIFO 写命令		✓
1	0	0	1	x	x	x	1	RX FIFO 读命令		✓
1	0	0	1	1	1	1	1	读取芯片 ID 命令		✓
0	0	0	0	1	0	0	0	软件复位命令	✓	
0	0	0	0	1	0	0	1	TX FIFO 地址指针复位命令	✓	
1	0	0	0	1	0	0	1	RX FIFO 地址指针复位命令	✓	
0	0	0	0	1	0	1	0	Deep Sleep 模式	✓	
0	0	0	0	1	0	1	1	Idle 模式	✓	
0	0	0	0	1	1	0	0	Light Sleep 模式	✓	
0	0	0	0	1	1	0	1	Standby 模式	✓	
0	0	0	0	1	1	1	0	TX 模式	✓	
1	0	0	0	1	1	1	0	RX 模式	✓	

A5~A0: 控制寄存器地址;

x: 硬件上无关, 但建议软件设置为 0;

B1~B0: Bank 编号

注: 1.此芯片支持多字节读/写操作, 每次读或写操作后地址自动递增。

2.在单个 CSN 使能周期内, 每个读/写命令之后允许软件读/写多个字节。

3.在 Sleep 模式下, GPIO 引脚维持上一个工作模式时的电平状态。

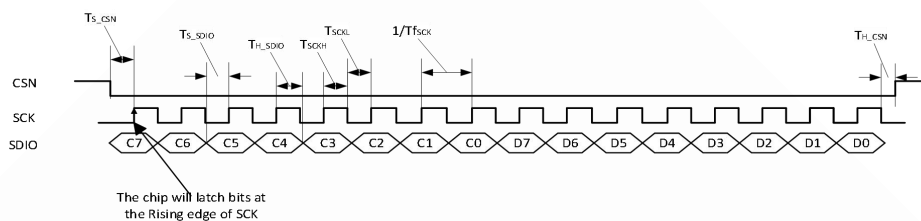


图 6 3 线 SPI 接口写入 1 字节数据

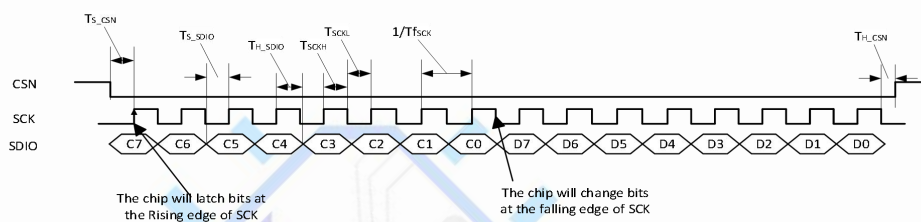


图 7 3 线 SPI 接口读取 1 字节数据

## 6、协议引擎

协议引擎是基于封包的数据链路层，具备自动封包组装与定时、自动应答及封包重传功能。该引擎可实现超低功耗与高性能通信，其特性显著提升了双向/单向系统的能效，且无需增加主控制器复杂度。

### 6.1 特性

协议引擎主要特性：

- 1 至 32 字节动态负载长度
- 自动封包处理
- 自动封包事务管理
  - 含负载的自动应答
  - 自动重传
- 支持 1:3 星型网络的 3 条数据通道

### 6.2 协议引擎概述

协议引擎使用自定义协议实现自动封包处理与定时：

发送时：组装封包并为传输数据提供时钟信号

接收时：持续在解调信号中搜索有效地址。发现有效地址后处理剩余封包并通过 CRC 校验。若校验通过，负载移入 RX FIFO 空闲槽位。所有高速比特处理与定时均由协议引擎控制。

协议引擎的自动封包事务管理简化了可靠双向数据链路的实现：

封包事务是两台收发器间的封包交换，一方作主接收器 (PRX)，另一方作主发射器 (PTX)。

事务由 PTX 发送封包启动，当 PTX 收到 PRX 的应答封包 (ACK) 时完成。PRX 可在 ACK 封包中附加用户数据实现双向通信。

自动事务处理流程：

- 1、PTX 向 PRX 发送数据包启动事务，协议引擎自动将 PTX 切换至接收模式等待 ACK 包。
- 2、PRX 收到包后，协议引擎自动组装并发送 ACK 包至 PTX，随后返回接收模式。
- 3、若 PTX 未立即收到 ACK 包，协议引擎在可编程延迟后自动重发原始数据包，并维持接收模式等待 ACK 包。

可配置最大重传次数/重传间隔等参数，所有处理均无需 MCU 介入。



## 6.3 协议引擎封包格式

本节描述协议引擎数据包的格式：该数据包包含前导码(Preamble)、同步码(SYNCword)、连接码(Trailer)、有效载荷头码(PLH)、有效载荷长度(PLEN)、数据(DATA)及校验码(CRC)。表 9 展示了该数据包格式（MSB 居左）。

表 9 数据包格式

Preamble	SYNCword	Trailer	PLH	PLEN	DATA	CRC
1~256 bytes	1~8 bytes	4 bits (optional)	1~2 bytes (optional)	1 byte (optional)	Max. 255 bytes	2 bytes (optional)

注：1. 前导码格式依据同步码最高位进行反转

若 MSB=0, 前导码 = 0101...01

若 MSB=1, 前导码 = 1010...10

2. 连接码格式依据同步码最低位进行反转

若 LSB=0, 连接码 = 1010

若 LSB=1, 连接码 = 0101

3. 连接码包含 4 位，可由用户自选，通过 TRAILER\_EN 位控制。

### 6.3.1 前导码(Preamble)

数据包开头是 1~256 个字的前导码，TX 模式下此长度由 TXPMLEN[7:0]决定，每个字的长度又由前导码模式决定。有两种前导码模式，通过 PMLP\_EN 位切换。

PMLP\_EN=0——自动生成前导码样式

开始位等于同步码的最高位，接着进行 1/0 翻转；此模式下，1 个字的长度=1 个字节。

PMLP\_EN=1—前导码样式依寄存器设定

前导码样式由 PMLPAT[15:0]定义，长度由 PMLPLEN 位决定。

PMLPLEN=0, 前导码样式=PMLPAT[7:0]；1 个字的长度为 1 个字节。

PMLPLEN=1, 前导码样式=PMLPAT[15:0]；1 个字的长度为 2 个字节。

### 6.3.2 同步码(SYNCword)

同步码长度由(SYNCLLEN[1:0],SYNCLLENLB)+1 设置，TX/RX 模式下可为 1~8 个字节。当 RX 端接收到匹配的同步码数据包时，数据域部分将被存入 FIFO。同步码低字节不可设置为 0x55 或 0xAA。注意，1 个字节同步码容易导致接收端误动作，建议同步码长度至少 2 个字节。

### 6.3.3 连接码(Trailer)

连接码固定是 4 位，可由用户自选，通过 TRAILER\_EN 位使能。

### 6.3.4 有效载荷头码(PLH)

PLH 可由用户自选，通过 PLH\_EN 位使能。有效载荷头码长度是 1 或 2 个字节，由 PLHLEN 位决定。当 PLHLEN 位为 0 时，数据包中的头码仅包含 PID[1:0]和 PLHA[5:0]位域。此时 PID[1:0]位于头码域的 7~6 位。当 PLHLEN 位设为 1 时，地址域的长度扩展到 14 位，由 PLHA[5:0]和 PLHEA[7:0]构成。

PLHA[5:0]与 PLHEA[7:0](若 PLHLEN=1)有两种用途，由 PLHAC\_EN 位控制。若 PLHAC\_EN



位为 0，PLHA[5:0]与 PLHEA[7:0]可用作软件标志位，实际功能可由用户自定义。若 PLHAC\_EN 位为 1，芯片将对比自身{PLHA[5:0],PLHEA[7:0]}和接收到的 {PLHA[5:0],PLHEA[7:0]}位域值。若匹配，则接收到的数据会移入 RXFIFO，否则紧接着的数据会被丢弃。PLHA[5:0]位域的目的是用于支持广播功能，当 PLHA[5:0]等于 0 时，允许 VT2303R 不执行地址过滤机制。

### 6.3.5 有效载荷长度 (PLEN)

PLEN 域可由用户自选，一旦通过 PLEN\_EN 位使能后，PLEN 域固定为 1 个字节。当 PLEN\_EN 位置 1 时，数据域的长度是可变的，由每个 TX/RX 数据包中的 PLEN 域决定。在 TX 模式下，发送器从 TXFIFO 发出由 TXDLEN 定义字节数量的数据，且 TXDLEN 值自动载入 PLEN 位域。在 RX 模式，接收器获得 PLEN 位域值，并接收由 PLEN 定义的字节数量的数据到 RXFIFO。

### 6.3.6 数据 (DATA)

TX 模式下的 TX 数据长度取决于 TXDLEN[7:0]。在 ExtendFIFO 模式下，最大长度可为 255 个字节。在 Infinite FIFO 模式下，长度无限制可超过 255 字节。若 PLEN\_EN 位等于 1，TX 数据包的 PLEN 域使能，此时 PLEN 值等于 TXDLEN[7:0]。在 RX 模式下，若 PLEN\_EN 等于 0，RX 数据长度由 RXDLEN[7:0]决定；若 PLEN\_EN 等于 1，RX 数据长度由 PLEN 位域决定。

### 6.3.7 校验码(CRC)

CRC 域可由用户自选，通过 CRC\_EN 位使能。建议始终设置 CRC\_EN 位为 1 以检查数据的正确性。共有两种 CRC 格式，通过 CRCFMT 位选择。

CRCFMT=0:

$$CCITT-16-CRC\ G(X)=X^{16}+X^{12}+X^5+1$$

CRCFMT=1:

$$IBC-16-CRC\ G(X)=X^{16}+X^{15}+X^2+1$$

### 6.3.8 正向纠错 (FEC)

该数据编码/解码功能由 FEC\_EN 位使能。使用(7,4)汉明码对每 4 位数据进行大于等于 1 位的错误检测。正向纠错之后，每笔数据的数据长度为(4+3)×2=14 位。

表 10 汉明码功能表格

位	7	6	5	4	3	2	1
传送位	D3	D2	D1	P2	D0	P1	P0
P0	Y	N	Y	N	Y	N	Y
P1	Y	Y	N	N	Y	Y	N
P2	Y	Y	Y	Y	N	N	N

### 6.3.9 数据白化 (Data Whitening)

数据白化/去白功能通过 WHT\_EN 位使能。使用 PN7/PN9 码与发送的数据进行异或运算。数据白化种子值由 WHTSD[8:0]设置。

### 6.3.10 曼彻斯特码 (Manchester Code)

曼彻斯特编码/解码功能由 MCH\_EN 位使能。每一个位在经过曼彻斯特编码后变成两位，解码后再恢复至一位。

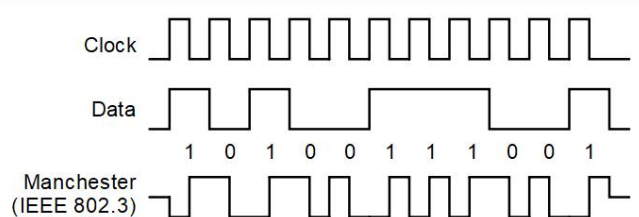


图 8 Direct 曼彻斯特码示例

## 7、射频功能操作

本章描述 VT2303R 射频收发器的操作模式以及用于控制无线电的参数。

可将 VT2303R 配置为掉电 (Power down)、深度睡眠 (Deep Sleep)、轻睡眠 (Light Sleep)、待机 (Standby)、空闲模式 (Idle)、接收 (RX) 或发射 (TX) 模式。VT2303R 内置一个状态机，用于控制芯片各操作模式之间的转换。该状态机接收来自用户定义的寄存器值和内部信号的输入。所有工作模式以及对应的关键功能开启/关闭状态表 11 所示。

表 11 各模式对应功能及命令

模式	寄存器保存	3.3V	LIRC	稳压器	X0	Standby+VC0	TX	RX	Strobe 命令
RX	Yes	ON	OFF	ON	ON	ON	OFF	ON	1000_1110
TX	Yes	ON	OFF	ON	ON	ON	ON	OFF	0000_1110
Idle	Yes	ON	ON	OFF	OFF	OFF	OFF	OFF	0000_1011
Standby	Yes	ON	OFF	ON	ON	ON	OFF	OFF	0000_1101
Light Sleep	Yes	ON	OFF	ON	ON	OFF	OFF	OFF	0000_1100
Deep Sleep	Yes	ON	OFF	OFF	OFF	OFF	OFF	OFF	0000_1010
Power Down	No	OFF	OFF	OFF	OFF	OFF	OFF	OFF	—

### 7.1FIFO 模式

#### 7.1.1FIFO 状态机

若 DIR\_EN 位为 0，芯片工作于 FIFO 模式，芯片的模式切换通过主控 MCU 发送 Strobe 命令来实现，且 TX/RX 数据来自数据包处理硬件。FIFO 模式下，RF 发送器要发送的数据来自 FIFO 且由主控 MCU 预先写好。共有四个 FIFO 模式可支持各种应用。这些模式为 Simple FIFO 模式、Block FIFO 模式、Extend FIFO 模式和 Infinite FIFO 模式。

使用 FIFO 之前，需先发送 TX FIFO 地址指针复位命令以及 RX FIFO 地址指针复位命令，以复位 FIFO 指针和缓冲器。在这之后，FIFO 将复位至初始状态。

图 9 展示了 FIFO 模式及其运作方式。

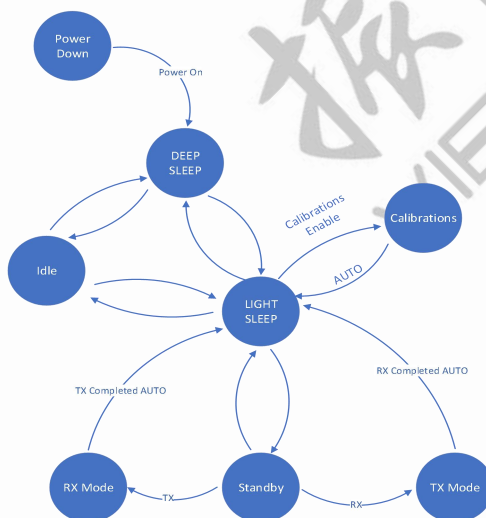


图 9 VT2303R FIFO 模式状态图

VT2303R 初始状态为 Power Down 模式。芯片完成内部上电复位后先进入 Deep Sleep 模式，等待来自主控 MCU 的 Strobe 命令。若接收到 Light Sleep 命令，芯片将使能内部 LDO、起振 XO 并进入 Light Sleep 模式。在此模式下，若有需要主控 MCU 可让 VT2303R 执行校准功能。若要进行正常的 TRX 操作，主控 MCU 可发送 RX 或 TX 命令给芯片。当芯片接收到 TX 或 RX 命令后，会先进入 Standby 模式并持续一段时间，此时间称为 TX/RX 设置时间。经过这段设置时间后，芯片将进入 RX 或者 TX 模式。芯片保持 TX/RX 状态直到 TX/RX 操作完成。这之后芯片自动返回 Light Sleep 模式。

针对低功耗周期性无线传输，该芯片支持低功耗 Idle 模式，在此模式下 LIRC 和唤醒定时器开启。合理设置定时器并发送 Idle 命令，芯片将关闭 LDO 和 XO 并进入 Idle 模式。当 ATR\_EN 设置为“1”后，唤醒定时器开始计数。芯片保持 Idle 模式直到达到定时器定时时间，与此同时芯片通过 GPIO 发送一个中断请求以唤醒主控 MCU。接着，主控 MCU 可让芯片进入 Light Sleep 模式，接着再执行 TX/RX 相关操作。当 TX/RX 操作完成，主控 MCU 可发送 Idle 命令给芯片使其再次进入 Idle 模式。

### 7.1.2 FIFO 模式时序图

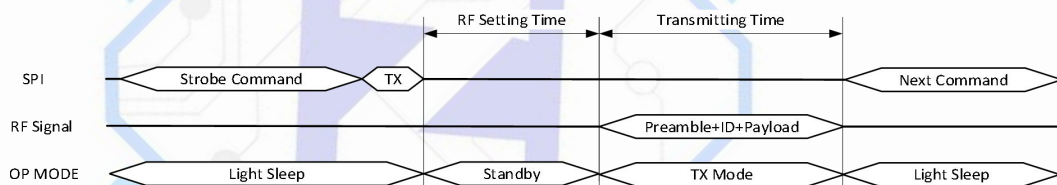


图 10 FIFO 模式 TX 时序图

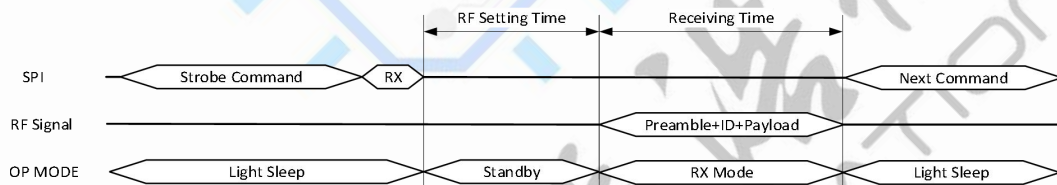


图 11 FIFO 模式 RX 时序图

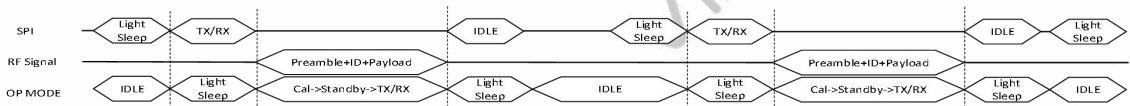


图 12 FIFO 模式周期性 TX/RX 时序图

### 7.1.3 Simple FIFO 模式

此 FIFO 模式用于 TX/RX 数据长度小于等于 64 字节的一般应用。注意，数据长度不可超过 64 个字节。要使用 Simple FIFO 模式，主控 MCU 需通过 SPI 写 FIFO 命令将要发送的数据写入 FIFO。发送的顺序为先写的字节先发送，每个字节里的最高位先发送。用户需预先确定好所有发送数据包格式，包括前导码、同步码以及数据包编码如正向纠错、CRC 和数据白

化。当 FIFO 数据填写完成后，将 TXFFSA[5:0]位域清零，并将 TXDLEN[7:0]/RXDLEN[7:0]位域设置为所需的发送/接收长度，单位为字节。接着发送 TX 命令开始数据传输。当前发送完成后，数据会被保存在 FIFO 中并等待下一次传输。

编程步骤：

- 1.通过 SPI 复位 TX FIFO 命令复位 TX FIFO。
- 2.通过 SPI 复位 RX FIFO 命令复位 RX FIFO。
- 3.TXFFSA[5:0]必须清零。
- 4.通过 SPI 写 FIFO 命令填写 TX FIFO。
- 5.设置 TXDLEN[7:0]/RXDLEN[7:0]控制 TX/RX 长度，单位为字节。
- 6.发送 TX 命令给发送器，发送 RX 命令给接收器。
- 7.通过 TX/RX 完成 IRQ 告知 TX/RX 完成状态。
- 8.重新发送含相同数据的 TX 数据包会自动将 TXFF\_RPTR 清零。

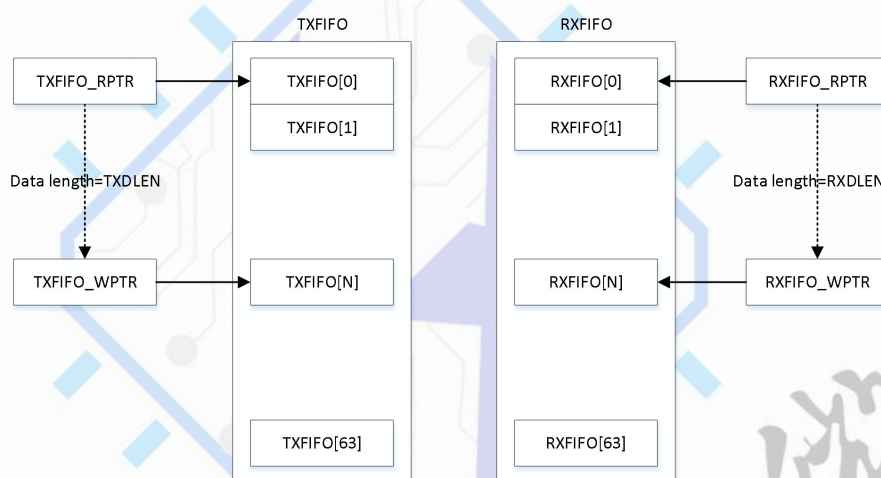


图 13 Simple FIFO 模式 FIFO 操作示意图

### 7.1.3 Block FIFO 模式

Block FIFO 模式可支持多键代码应用。用户需先将所有按键代码写入 FIFO。当有按键被按下时，主控 MCU 会侦测对应哪个按键，将 TXFFSA[5:0]设置为对应按键码的起始地址，并设置 TXDLEN[7:0]指示按键码长度，接着发送 TX 命令开始传输。此模式最大 FIFO 长度限制为 64 个字节。

编程步骤：

- 1.通过 SPI 写 FIFO 命令将按键 0~n 的代码写入 TXFIFO。
- 2.当有按键按下时，主控 MCU 会将 TXFFSA[5:0]设置为对应按键代码的起始地址。
- 3.设置 TXDLEN[7:0]表示按键代码长度。
- 4.将 RXDLEN[7:0]设置为按键代码长度，然后借由 SPI 命令进入 RX 模式。
- 5.发送 TX 命令给发送器，发送 RX 命令给接收器。
- 6.通过 TX/RX 完成 IRQ 告知 TX/RX 完成状态。



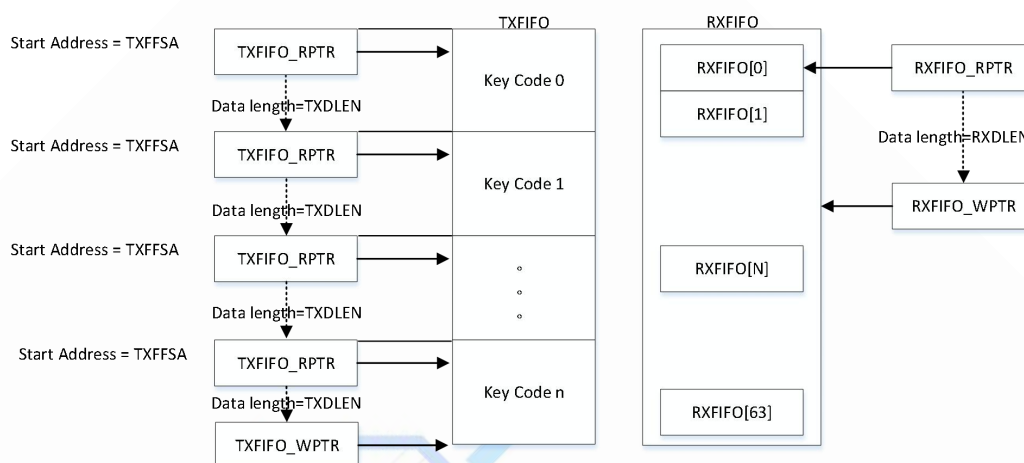


图 14 Block FIFO 模式 FIFO 操作示意图

#### 7.1.4 Extend FIFO 模式

Extend FIFO 模式适用于传输有效载荷数据长度较长的数据包。最大长度为 255 个字节。由于 FIFO 的物理长度为 64 字节，想要扩展每个数据包的可用发送长度，主控 MCU 和 FIFO 控制器之间需要一个握手机制。

设置 FFMG[1:0] 决定 FIFO 数据长度边距，然后设置 FFMG\_EN 位使能边距检测功能，当 TX FIFO 数据长度小于所选边距时通知 MCU。MCU 收到此提醒时应尽快往 TX FIFO 写数据以避免 TX FIFO 中数据长度为零(无数据)迫使传输中断。

编程步骤：

1. 设置 FFMG\_EN 使能 FIFO 长度边距检测功能(也叫 FIFO 低阈值检测功能)，设置 FFMG[1:0] 选择长度边距为 4、8、16 或 32 字节。

2. 设置 FIFOLTIE 位为 1 使能 FIFO 低阈值 IRQ。

3. 设置 GPIO[S] 位域(n=1~3)为“101”，则 IRQ 可从 GPIO1~GPIO3 输出。

4. TX: 若检测到 FIFO 低阈值 IRQ 信号，MCU 会往 TX FIFO 写数据，写入长度小于等于(64-长度边距)个字节。接着 MCU 将 FIFO 低阈值 IRQ 标志位 FIFOLTIF 清零。MCU 重复这一步骤直到所有 TX 数据都完全写入 TX FIFO。

5. RX: 若检测到 FIFO 低阈值 IRQ 信号，MCU 会从 RX FIFO 读取数据。接着 MCU 将 FIFO 低阈值 IRQ 标志位 FIFOLTIF 清零。MCU 重复这一步骤直到接收到 RX 完成 IRQ，并从 RX FIFO 读取剩余数据。



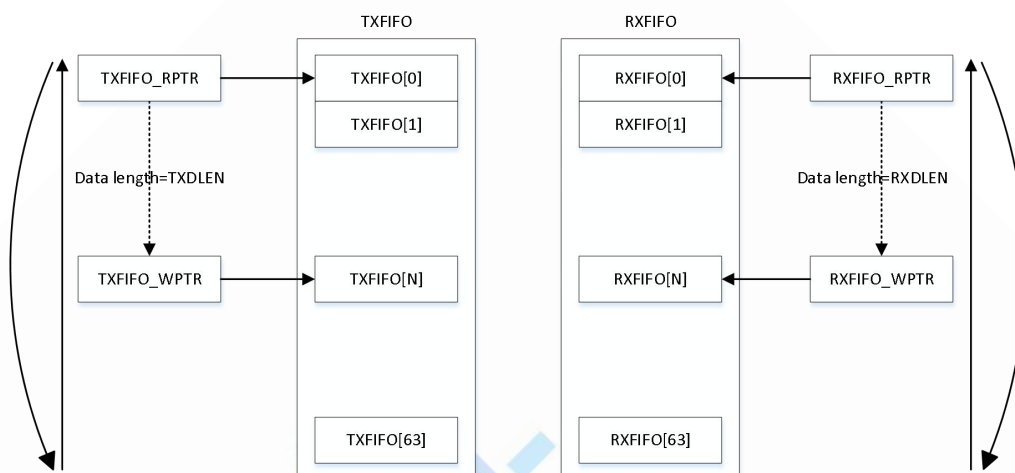


图 15 Extend FIFO 模式 FIFO 操作示意图

### 7.1.5 Infinite FIFO 模式

编程步骤:

1. 设置 FFINF\_EN 为 1 使能 Infinite FIFO 模式。
2. 此模式下的握手机制以及 IRQ 功能都与 Extend FIFO 模式下的一样。

3.TX: 当接收到 FIFO 低阈值 IRQ 时, MCU 继续往 TX FIFO 写入 TX 数据, 写入长度小于等于(64-长度边距)字节。接着 MCU 将 FIFO 低阈值 IRQ 标志位 FIFOLTIF 清零。MCU 重复这一步骤直到它想结束 Infinite FIFO 模式。若要想结束此模式, 当收到 IRQ 并往 TXFIFO 写入数据后, 若剩余待发送数据的长度小于 192 字节并大于 64 字节, MCU 应将 FFINF\_EN 位清零并将 TXDLEN[7:0]设置为剩余发送数据的长度。当所有目标数据都完全发送完后, 数据包发送终止。

4.RX: 当接收到 FIFO 低阈值 IRQ 时, MCU 从 RX FIFO 读取数据。接着 MCU 将 FIFO 低阈值 IRQ 标志位 FIFOLTIF 清零。MCU 重复这一步骤直到它想结束 Infinite FIFO 模式。若要想结束此模式, 当收到 IRQ 并从 RXFIFO 读取数据后, 若剩余待接收数据的长度小于 192 字节并大于 64 字节, MCU 应将 FFINF\_EN 位清零并将 RXDLEN[7:0]设置为剩余接收数据的长度。当所有目标数据都完全接收完后, 数据包接收终止。

### 7.1.6 接收数据包判断

在正常 RX 工作模式下, 数据包接收遵循图 16 所示判断标准。

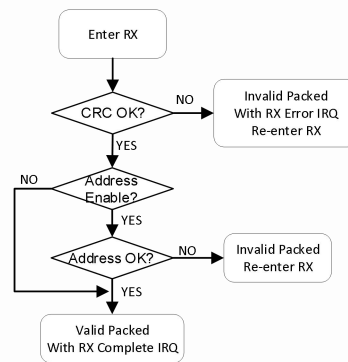


图 16 接收数据包判断示意图

如图 17 所示，VT2303R 针对连续 RX 模式以及自动应答模式采用额外的接收器数据包判断机制。这些特殊链接层函数的目的是为了在处理 TRX 数据包事务时减少 MCU 负载。

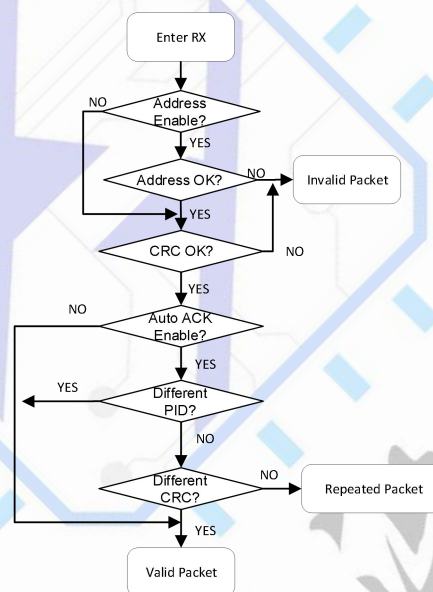


图 17 连续 RX 模式以及自动应答模式数据包判断示意图

### 7.1.7 连续 RX 模式

VT2303R 还支持特殊的连续 RX 工作模式。MCU 可设置 `RXCON_EN` 位为 1 来使能此模式并发送 RX 命令给芯片以启动此模式。若接收到一个有效的 RX 数据包，芯片将会向 MCU 发出 RX 完成中断请求。经过 `DLY_TXS[2:0]` 定义的一段时间后，芯片重复 RX 操作继续侦听后续传入的数据包。若接收到的是无效数据包，芯片只会重复 RX 操作但不会向 MCU 发出 RX 完成中断请求。MCU 发送 `LightSleep` 命令给芯片可停止连续 RX 模式。在连续 RX 模式下，只可使用 Simple FIFO 模式。在 MCU 从 RX FIFO 读取数据之前，为了防止接收的数据包数据长度域因后续传入数据包而损坏，用户需将 `RXPL2F_EN` 和 `PLLEN_EN` 都设置为 1 从而将 `PLLEN` 信息保存到 RX FIFO 中。由于 `PLLEN` 域的存在，最大数据包数据长度变为 63 个字节。若 MCU 从 RX FIFO 读取数据之前有新传入的数据，则发生 FIFO 溢出错误，此时 VT2303R 会将 `RXERRIF` 位置高并向 MCU 发出 RX 错误中断请求。此时，MCU 应该退出连续 RX 模式并将 `RXFIFO` 指针复位。

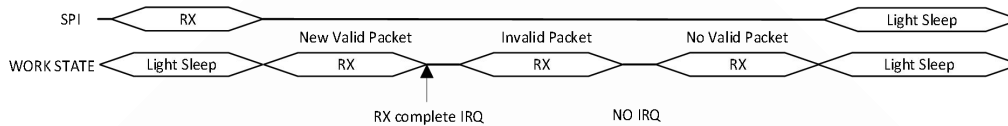


图 18 连续 RX 模式流程示意图

### 7.1.8 ARK 模式：自动重发和自动应答

VT2303R 支持自动重发和自动应答机制，通过设置 `ARK_EN` 为 1 来使能。此机制支持简单的双向通信，但只可工作在 SimpleFIFO 模式下。

设置 `ARK_EN` 为 1 使芯片进入自动重发和自动应答模式。当接收到来自 MCU 的 TX 命令时会触发自动重发功能，当收到 RX 命令时会触发自动应答功能。自动重发模式下主机发送给从机的数据包格式如下图所示。

Preamble	SYNCword	Trailer	PLH	PLEN	DATA	CRC
1~256 bytes	1~8 bytes	4 bits (optional)	1~2 bytes (optional)	1 byte (optional)	Max. 255 bytes	2 bytes (optional)



PID	PLHA	PLHEA
2 bits	6 bits	8 bits

处于自动应答模式下的从机，使用下图所示的数据包格式发送应答数据包给主机。注意，此应答数据包不包含有效载荷数据域。

Preamble	SYNCword	Trailer	PLH	PLEN	CRC
1~256 bytes	1~8 bytes	4 bits (optional)	1~2 bytes (optional)	1 byte (optional)	2 bytes (optional)



PID	PLHA	PLHEA
2 bits	6 bits	8 bits

若在 ARK 模式下要使用地址域，则自动重发端(主机)设置的地址要与自动应答端(从机)的相同。

设置好 `ARKNM[3:0]`、`ARK_EN` 和 `ARKRXAP[7:0]` 之后，MCU 发出 TX 命令以开始自动重发进程。VT2303R 开始发送 TXFIFO 中的数据并在 TX 操作完成后进入 RX 模式。RX 周期为 250  $\mu$ s(默认)的倍数，此倍数等于  $(\text{ARKRXAP}[7:0]+1)$ 。若 VT2303R 在 RX 周期内收到来自从机的 CRC 校验正确的有效应答包，芯片将返回 LightSleep 模式并向 MCU 发送 TX 完成中断请求。否则，VT2303R 会判断是否达到  $(\text{ARKNM}[3:0]+1)$  定义的自动重发次数。若未达到，芯片将进入 TX 模式继续发送相同的 TX 数据，此时自动重发次数加一。

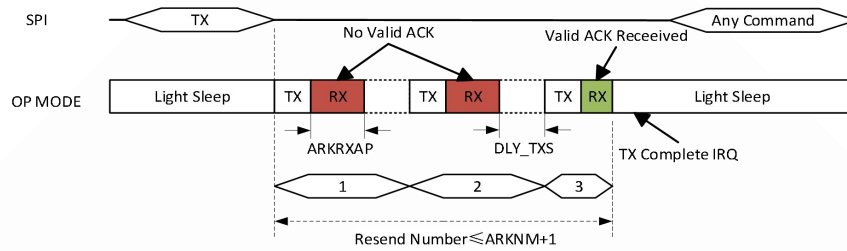


图 19 达到 ARKNM 次数之前收到 ACK 的数据包流程图

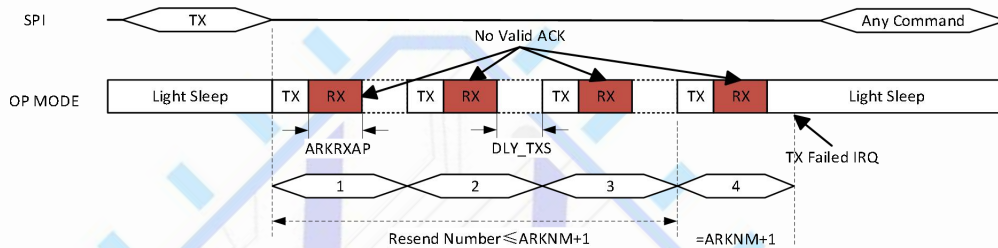


图 20 达到 ARKNM 次数之前未收到 ACK 的数据包流程图

从机自动应答方面，MCU 通过发送 RX 命令开启自动应答进程，通过发送 LightSleep 命令停止自动应答进程。在自动应答模式下，从机端会启用 PID/CRC 过滤功能以检查接收到的数据包。若新传入的数据包的 PID/CRC 与上一个数据包的 PID/CRC 相同，则新接收到的数据包将被视为重复的数据包。

在自动应答过程中，若接收到 PID/CRC 值不同且 CRC/地址检测正确的有效数据包，芯片会向 MCU 发出 RX 完成中断请求，并自动发送应答包给主机。若芯片接收到相同 PID/CRC 且 CRC/地址检测正确的数据包，会将其视为重复的数据包。接着，芯片不会向 MCU 发送 RX 完成中断请求但仍会自动发送应答包给主机。若芯片接收到的数据包 CRC/地址检测错误，不发送中断请求并且重新执行 RX 操作继续侦听后续数据包。

当前 RX 完成到重新下一次 RX 操作之间的时间间隔由 ARKRXAP[7:0]决定。在一般应用里，MCU 接收到 RX 完成中断请求后需在此时间内从接收器 FIFO 读取数据。另外，在接收到 RX 完成中断请求后，MCU 若想退出 ARK 模式需等待同样一段时间。

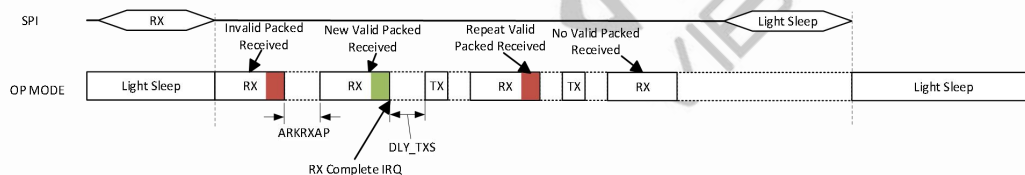


图 21 自动应答流程

### 7.1.9 ATR 模式：自动发送/接收

VT2303R 支持特殊的 ATR 工作模式，可减少外部主机负载。此芯片含两种 ATR 功能，一种是 WOR，另一种是 WOT。这两种功能都只能工作在 SimpleFIFO 模式下。这两种工作模式需要搭配使用一个以低频运行的 Idle 模式定时器。其低频时钟来自内部 LIRC 或外部 ROSCI 时钟，由 ATR1 寄存器中的 ATRCLKS 位选择。ATRCT 定时器有两种工作模式，通过 ATRCTM



位选择。若 ATRCTM 位清零则选择单次模式，此模式下当芯片进入 Idle 状态，每当发生 ATR 事务时 ATRCT 定时器重新启动，每当接收到 LightSleep 命令时，ATRCT 定时器停止并退出 ATR 模式。若 ATRCTM 位置高则选择连续模式，此模式下一旦接收到 Idle 命令 ATRCT 定时器开始工作，并持续工作直到 ATR\_EN 位或 ATRCTM 位清零。

进入 ATR 模式后，只有 Idle 命令、LightSleep 命令、设置寄存器存储区命令和控制寄存器读/写命令可被 VT2303R 识别。

### 7.1.10 WOT (自动唤醒 TX) 功能

当设置 ATR\_EN 位为 1、ATRM[1:0]为“00”使能 WOT 功能后，芯片将周期性地从 Idle 模式唤醒并在不与主控 MCU 互动的情况下发送 TXFIFO 中的数据。当接收到来自 MCU 的 Idle 命令时芯片开始 WOT 进程，当接收到来自 MCU 的 LightSleep 命令时，芯片停止 WOT 进程。ATRCYC[15:0]位用于设置 WOT 功能的唤醒周期。若达到定时器定时时间，唤醒定时器会触发芯片离开 Idle 状态并进入激活状态去发送数据，同时 ATRCYC[15:0]的值会被载入定时计数器。完成 TX 操作后，芯片返回 Idle 模式并保持此状态直到下一次达到唤醒定时器定时时间。在激活状态下，芯片默认只执行一次唤醒传输。用户可结合 ARK 功能扩展唤醒传输机制。重复发送的次数由(ARKNM[3:0]+1)决定。在重复发送数据包的间隔内会插入一个 RX 时隙，此时隙由 ARK8 寄存器的 ARKRXAP[7:0]位控制。若芯片在此间隙期间收到 ACK 信号，会向主控 MCU 发出 TX 完成中断请求。

Note: 1.VCO Calibration time:~152μs@433MHz/~96μs@868MHz

2.Both new TX and RX need to first go through“LightSleep→VCO Cal.→Standby”

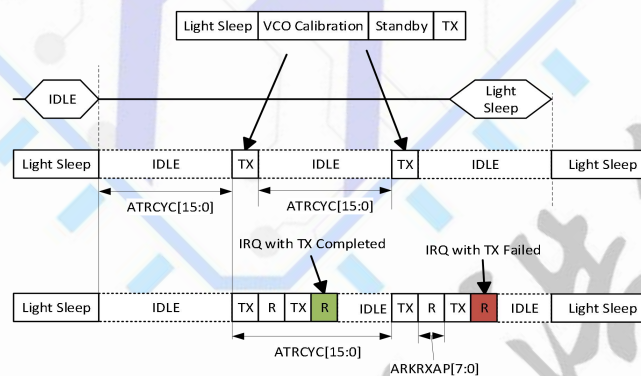


图 22 WOT 流程

### 7.1.11 WOR (自动唤醒 RX) 功能

当设置 ATR\_EN 位为 1、ATRM[1:0]为“01”使能 WOR 功能后，芯片将周期性地从 Idle 模式唤醒并在不与主控 MCU 互动的情况下侦听传入的数据。当接收到来自 MCU 的 Idle 命令时芯片开始 WOR 进程，当接收到来自 MCU 的 LightSleep 命令时，芯片停止 WOR 进程。ATRCYC[15:0]位用于设置 WOR 功能的唤醒周期。若达到定时器定时时间，唤醒定时器会触发芯片离开 Idle 状态并进入激活状态去侦听传入的数据，同时 ATRCYC[15:0]的值会被载入定时器的计数器。接收有效周期由 ATRRXAP[7:0]位决定，是 250us(默认)的倍数，最少为 250us。若在 RX 有效周期内未接收到数据包，芯片将返回 Idle 模式并等待下一轮 WOR 进程。

若检测到“前导码+同步码”，则有效周期将自动延长。延长时间由(ATRRXEP[15:0]+1)定义。延长时间是 250 μs(默认)的倍数，最少为 250us。一旦接收到同步码，接收周期将会自动延长直到整个数据包被完全接收。当 RX 接收完成且 CRC 校验正确时，若 RXCMPIE 为 1，VT2303R 会发送 RX 完成中断请求告知 MCU 并停留在 LightSleep 模式。MCU 可从 RXFIFO 读

取传入的数据并发出 Idle 命令从而开始新一轮 WOR 进程。若想退出 WOR 模式，MCU 还需发送 LightSleep 命令给 VT2303R。

Note: 1.VCO Calibration time:~152us@433MHz/~96us@868MHz

2.Both new RX and TX need to first gothrough “LightSleep→VCOCal.→Standby”

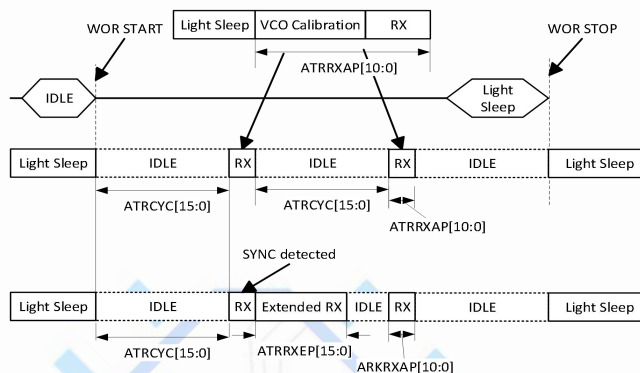


图 23 WOR 未接收到传入数据

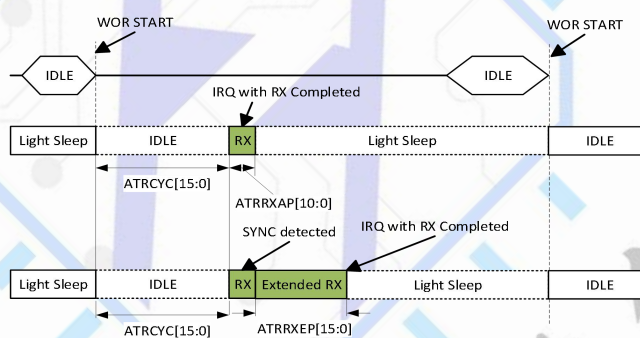


图 24 WOR 接收到传入数据

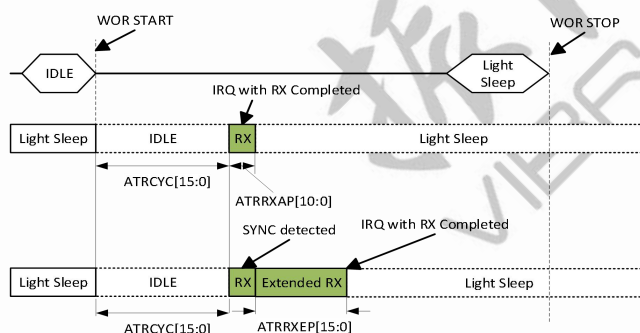


图 25 WOR 接收到传入数据后停止 WOR

在 WOR 有效周期内，芯片默认只执行一次 RX 操作。用户可结合 ARK 功能扩展唤醒接收机制。在 WOR+ARK 模式下，在重复接收数据包的间隔内会插入一个 TX 时隙作为应答。TX 持续时间取决于发送数据速率。芯片保持 RX 模式的最大时间由 ATTRXAP 和 ATTRXEP 共同决定。若在达到定时器定时时间之前接收到 CRC 校验正确且包含不同 PID/CRC 值的有效数



据包，芯片会发送一个 RX 完成中断请求给 MCU 并自动进入 TX 模式。若接收到 CRC 校验正确但 PID/CRC 值相同的重复数据包，芯片只会自动进入 TX 模式，不会向 MCU 发送中断请求。当 TX 操作完成后，芯片将再次返回 RX 模式并侦听传入的数据包直到无数据包传入为止。

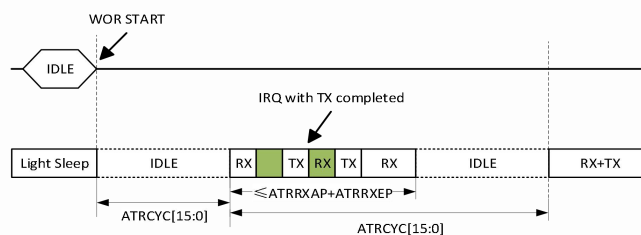


图 26 WOR+ARK 流程

### 7.1.12 WTM (唤醒定时模式)

VT2303R 可被设置作为一个可编程定时器从 GPIO 口输出周期性波形。用户可使用此信号唤醒 CPU。设置 ATR\_EN 为 1 和 ATRM=10/11 以使能 WTM 模式。当接收到来自 MCU 的 Idle 命令时，芯片开始 WTM 进程，当接收到来自 MCU 的 LightSleep 命令时，芯片停止 WTM 进程。在整个 WTM 进程内芯片都处于 Idle 模式。

## 7.2 Direct 模式

若设置 DIR\_EN 为 1, TX 数据由主控 MCU 直接发送给 VT2303R，RX 数据由 VT2303R 直接发送给主控 MCU。

### 7.2.1 Direct 状态机

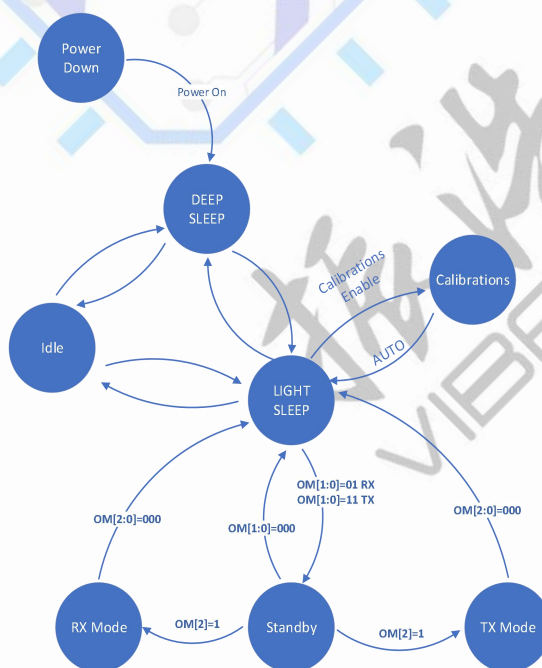


图 27 VT2303R Direct 模式状态图

为了简化 VT2303R 与主控 MCU 之间的数据位时钟同步，设置 GPIO3S[3:0]，VT2303R 便可从 GPIO3 输出 TBCLK/RBCLK 时钟。TBCLK 和 RBCLK 都是 50/50 占空比周期。在发送模式下，

主控 MCU 在 TBCLK 信号上升沿时输出位数据，VT2303R 在 TBCLK 信号的下降沿时采样 TX 位数据。在接收模式下，主控 MCU 在 RBCLK 信号上升沿时接收数据，VT2303R 在 RBCLK 信号下降沿时输出位数据。主控 MCU 可设置 GPIO1S[2:0]/GPIO2S[2:0]选择 GPIO1/GPIO2 用于 TX/RX 位数据传输。

若要在 Direct 模式下进行 TX 操作，主控 MCU 需设置 OM[1:0]，即 RTX\_SEL 和 SX\_EN 位，为“11”以选中 TX 模式并先让 VT2303R 进入 Standby 模式。接着设置 OM[2]，即 RTX\_EN 位，为“1”使 VT2303R 开始发送 TX 数据。一旦主控 MCU 将 OM[2:0]位设置为“000”，VT2303R 将返回 LightSleep 模式。

若要在 Direct 模式下进行 RX 操作，主控 MCU 需先设置 OM[1:0]位为“01”，接着设置 OM[2]为“1”使 VT2303R 开始接收数据。当芯片接收到匹配的同步码时，会输出 RBCLK 时钟，接收数据位，即有效载荷部分，然后再发送给主控 MCU。

在 Direct 模式下，对传输的数据长度无限制。

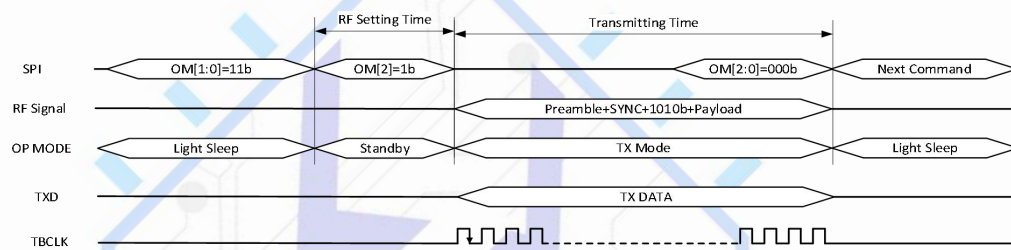


图 28 Direct 模式 TX 时序图

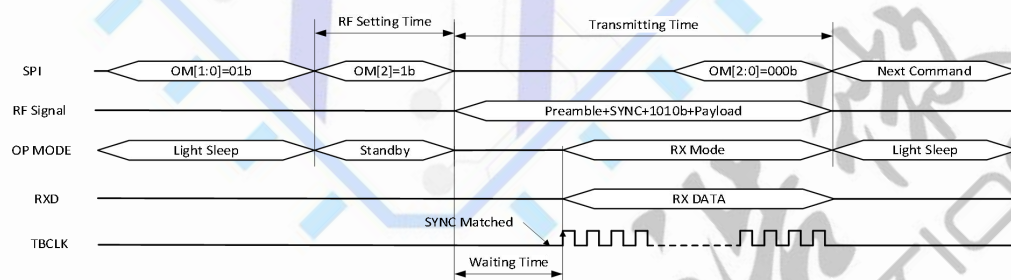


图 29 Direct 模式 RX 时序图

## 8、寄存器说明

### 8.1 寄存器结构



图 30 寄存器结构示意图

公用区：包含 32 字节空间。无论存储区指针为何值，访问地址 00h~1Fh 都意味着访问公用区。

Bank0~1：每个 Bank 包含 32 字节空间。通过存储区指针选择 Bank。

存储区指针，即 BANK[1:0]，定义在公用区，可通过设置寄存器存储区命令直接设置，且可通过控制寄存器命令进行读/写。

### 8.2 控制寄存器访问



图 31 命令后加 N 字节数据

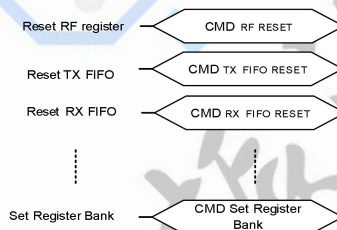


图 32 仅命令

### 8.3 公用区控制寄存器

#### 8.3.1 公共区域寄存器 MAP

上电复位后，所有控制寄存器被设置为初始值。软件复位后，除了位于 RC1、IO1、IO2 和 IO3 寄存器的 FSYCK\_EN、FSYCK\_DIV[1:0]、PWRON、GPIO1S[2:0]、GPIO2S[2:0]、PADDS[1:0]、GPIO3S[3:0]、GPIOPU[3:1]、SPIPU、SDO\_TEN 位之外，其它控制寄存器也将恢复至初始值，而前述控制位在软件复位后保持不变。

表 12 公共区寄存器位说明

地址	名称	位							
		7	6	5	4	3	2	1	0
00h	CFG1	OOK_EN	AGC_EN	RXCON_EN	DIR_EN	—	—	BANK[1:0]	
01h	RC1	PWRON	FSYCK_RD	XCLK_RDY	XCLK_EN	FSYCK_DIV[1:0]	FSYCK_EN	RST_LL	
02h	IRQ1	RXTO	RXFFOW	—	RXCRCF	RXDETS[1:0]	IRQCPOR	IRQPOR	
03h	IRQ2	ARKTFIE	ATRCTIE	FIFOLTIE	RXERRIE	RXDETIE	CALCMPIE	RXCMPIE	TXCMPIE
04h	IRQ3	ARKTFIF	ATRCTIF	FIFOLTIF	RXERRIF	RXDETIF	CALCMPIF	RXCMPIF	TXCMPIF
06h	IO1	PADDS[1:0]		GPIO2S[2:0]			GPIO1S[2:0]		
07h	IO2	—				GPIO3S[3:0]			
08h	IO3	SDO_TEN	SPIPU	—	—	GPIOPU[3:1]			—
09h	FIFO1	—	—	TXFFSA[5:0]					
0Ah	FIFO2	—	—	—	RXPL2F_EN	FFINF_EN	FFMG_EN	FFMG[1:0]	
0Bh	PKT1	TXPMLN[7:0]							
0Ch	PKT2	PID[1:0]		TRAILER_EN	WHTFMT[0]	SYNCLEN[1:0]		—	
0Dh	PKT3	MCH_EN	FEC_EN	CRC_EN	CRCFMT	PLEN_EN	PLHAC_EN	PLHLEN	PLH_EN
0Eh	PKT4	WHT_EN	WHTSD[6:0]						
0Fh	PKT5	TXDLEN[7:0]							
10h	PKT6	RXDLEN[7:0]							
11h	PKT7	RXPID[1:0]		DLY_RXS[2:0]			DLY_TXS[2:0]		
12h	PKT8	—	PLHA[5:0]						
13h	PKT9	PLHEA[7:0]							
14h	MOD1	DTR[7:0]							
15h	MOD2	RXFDOS[11:8]				DITHER[1:0]		—	DTR[8]
16h	MOD3	RXFDOS[7:0]							
17h	DM1	—	—	MDIV[5:0]					
18h	DM2	—	—	SDR[5:0]					
19h	DM3	CSF_SW_EN	FD_MOD[6:0]						
1Ah	DM4	—				CFO_DSEL	—	PH_DIFF_MOD	—
1Bh	DM5	FD_HOLD[7:0]							
1Eh	DM8	M_RATIO[7:0]							

### 8.3.2 CFG1：配置控制寄存器 1

Bit	7	6	5	4	3	2	1	0
Name	OOK_EN	AGC_EN	RXCON_EN	DIR_EN	—	—	BANK[1:0]	
R/W	R/W	R/W	R/W	R/W	—	—	R/W	
Reset	0	0	0	0	0	0	0	0

Bit 7    OOK\_EN: OOK 模式使能  
          0: GFSK 模式  
          1: OOK 模式

- Bit 6 AGC\_EN: AGC 使能  
0: 除能  
1: 使能
- Bit 5 RXCON\_EN: RX 连续模式使能  
0: 除能  
1: 使能  
设置此位只影响正常 RX 模式以及无 ARK 功能的 ATR RX 模式。
- Bit 4 DIR\_EN: Direct 模式使能  
0: TX/RX 数据来自数据包处理硬件  
1: TX/RX 数据直接来自/ 发至外部 MCU
- Bit 3~2 保留, 必须设为“00”
- Bit 1~0 BANK[1:0]: 控制寄存器 Bank 选择  
00: Bank 0  
01: Bank 1  
10: Bank 2  
11: 保留  
此选择可通过设置寄存器存储区命令和控制寄存器命令设置。

### 8.3.3 RC1: 复位 / 时钟控制寄存器 1

Bit	7	6	5	4	3	2	1	0
Name	PWRON	FSYCK_RDY	XCLK_RDY	XCLK_EN	FSYCK_DIV[1:0]		FSYCK_EN	RST_LL
R/W	R/W	R	R	R/W	R/W		R/W	R/W
POR	1	—	—	—	0	0	0	—
Reset	—	0	0	1	—	—	—	0

- Bit 7 PWRON: 3.3V 上电标志位  
此位仅在上电复位后被设置为“1”，它不受 Strobe 命令软件复位影响。此标志位被置高后需通过软件清零。软件可先检查此标志位状态再决定在 Light Sleep 模式时是否进行自动校准。
- Bit 6 FSYCK\_RDY: FSYCK 时钟就绪标志位(只读)  
0: 未就绪  
1: 就绪  
此位用于指示 FSYCK 时钟是否就绪。当 FSYCK\_EN=0、发生上电复位或接收到 Deep Sleep 或者 Idle 命令时，此位都会自动清零。
- Bit 5 XCLK\_RDY: XCLK 时钟就绪标志位(只读)  
0: 未就绪  
1: 就绪  
此位用于指示 XCLK 去抖计数器是否记满、XCLK 时钟是否就绪。当离开 Deep Sleep 状态时，需要一段时间后此标志位才被置高。当 XCLK\_EN=0、RST\_LL=1、发生上电复位或接收到软件复位命令、Deep Sleep 命令或者 Idle 命令时，此标志位都会自动清零。
- Bit 4 XCLK\_EN: XCLK 时钟使能  
0: 除能  
1: 使能  
此位置高将使能 XCLK 至基带模块的路径。若有需要可将此位清零以减少功耗。



当写数据到 FIFO 时，XCLK 时钟必须使能。

Bit 3~2 FSYCK\_DIV[1:0] : FSYCK 时钟( XCLK 分频) 选择

00 : 1/1 XCLK

01 : 1/2 XCLK

10 : 1/4 XCLK

11 : 1/8 XCLK

Bit 1 FSYCK\_EN: FSYCK 时钟使能

0: 除能

1: 使能

Bit 0 RST\_LL: 低电压(1.2V)逻辑复位控制

0: 不复位

1: 复位

### 8.3.4 IRQ1: 中断控制寄存器 1

Bit	7	6	5	4	3	2	1	0
Name	RXTO	RXFFOW	—	RXCRCF	RXDETS[1:0]		IRQCPOR	IRQPOR
R/W	R	R	—	R/W	R/W		R/W	R/W
Reset	0	0	0	0	1	0	0	1

Bit 7 RXTO: RX 溢出标志位

0 : RX 溢出未发生

1 : RX 溢出发生

当 RX 溢出发生时，此标志位将被硬件置高。当接收到 Light Sleep 命令、芯片进入 RX 连续模式、WOR/WOT 唤醒发生或芯片进入 ARK TX/RX 模式时，此标志位都会自动清零。

Bit 6 RXFFOW: RX FIFO 覆写标志位

0 : RX FIFO 覆写未发生

1 : RX FIFO 覆写发生

当 RX FIFO 覆写情况发生时，此标志位将被硬件置高。当接收到 RX FIFO 复位命令或 RX 命令时，此标志位都会自动清零。

Bit 5 保留，必须设为“0”

Bit 4 RXCRCF: RX CRC 错误标志位

Bit 3~2 RXDETS[1:0] : RX 检测选择

00: 检测载波(Carry)

01: 保留

10/11: 检测同步码(SYNCWORD)

Bit 1 IRQCPOR: IRQ 标志位清零极性选择

0: 写 0 时对应 IRQ 标志位清零

1: 写 1 时对应 IRQ 标志位清零

Bit 0 IRQPOR: IRQ 信号极性选择

0: 低有效

1: 高有效

当 IRQ3 寄存器里的 IRQ 标志位置高且对应的 IRQ 功能使能时，IRQ 信号的有效电平由此位决定。



### 8.3.5 IRQ2：中断控制寄存器 2

Bit	7	6	5	4	3	2	1	0
Name	ARKTFIE	ATRCTIE	FIFOLTIE	RXERRIE	RXDETIE	CALCMPIE	RXCMPPIE	TXCMPPIE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit7 ARKTFIE: ARKTX 失败 IRQ 使能

0: 除能

1: 使能

Bit6 ATRCTIE: ATR 周期定时器 IRQ 使能

0: 除能

1: 使能

Bit5 FIFOLTIE: FIFO 低阈值 IRQ 使能

0: 除能

1: 使能

Bit4 RXERRIE: RX 错误 IRQ 使能

0: 除能

1: 使能

Bit3 RXDETIE: RX 事件检测 IRQ 使能

0: 除能

1: 使能

Bit2 CALCMPIE: 校准完成 IRQ 使能

0: 除能

1: 使能

Bit1 RXCMPPIE: RX 完成 IRQ 使能

0: 除能

1: 使能

Bit0 TXCMPPIE: TX 完成 IRQ 使能

0: 除能

1: 使能

### 8.3.6 IRQ3：中断控制寄存器 3

Bit	7	6	5	4	3	2	1	0
Name	ARKTFIF	ATRCTIF	FIFOLTIF	RXERRIF	RXDETIF	CALCMPIF	RXCMPPIF	TXCMPPIF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

当此寄存器中的各个标志位置高时，产生对应的中断请求。这些标志位通过写 0 还是写 1 清零由 IRQCPOR 位决定。

Bit7 ARKTFIF: ARKTX 失败 IRQ 标志位

0: 无请求

1: 中断请求

Bit6 ATRCTIF: ATR 周期定时器 IRQ 标志位

0: 无请求

1: 中断请求

当 ATRCT 定时器计满时，此标志位将置高。

Bit5 FIFOLTIF: FIFO 低阈值 IRQ 标志位

0: 无请求

1: 中断请求

在 BurstTX 模式下，若此位置 1 则表示 TXFIFO 里数据长度小于 FFMG 设置的阈值且还有待写入 FIFO 的 TX 数据。在 BurstRX 模式下，若此位置 1 则表示 RXFIFO 剩余的空间小于 FFMG 设置的阈值且待接收的 RX 数据长度大于 FFMG 设置的阈值。

Bit4 RXERRIF: RX 错误 IRQ 标志位

0: 无请求

1: 中断请求

所谓 RX 错误情况包含 RX 失败、CRC 失败(CRC\_EN=1)或 RXFIFO 覆写。

Bit3 RXDETIF: RX 事件检测 IRQ 标志位

0: 无请求

1: 中断请求

RX 事件包括载波、前导码和同步码，实际触发中断源取决于 RXDETS[1:0]设。

Bit2 CALCMPIF: 校准完成 IRQ 标志位

0: 无请求

1: 中断请求

当 ACAL\_EN=0 时，LIRC 校准可由自己的使能位使能，当校准完成后会触发中断请求。当 ACAL\_EN=1 时，VCO 和 RC 校准都使能，两者都完成后会触发中断请求。

Bit1 RXCMPIF: RX 完成 IRQ 标志位

0: 无请求

1: 中断请求

当 RX 操作完成且无错误发生，此标志位将被硬件置高。

Bit0 TXCMPIF: TX 完成 IRQ 标志位

0: 无请求

1: 中断请求

### 8.3.7 IO1: I/O 控制寄存器 1

Bit	7	6	5	4	3	2	1	0
Name	PADDS[1:0]		GPIO2S[2:0]			GPIO1S[2:0]		
R/W	R/W		R/W			R/W		
ROR	0	1	0	0	0	0	0	0

Bit7~6 PADDS[1:0]: PAD 驱动强度选择(仅通过 POR 复位)

00: 0.5mA

01: 1mA

10: 5mA

11: 10mA

Bit5~3 GPIO2S[2:0]: GPIO2 引脚功能选择(仅通过 POR 复位)

000/111: 无功能，输入

001: SDO, 4 线 SPI 数据，输出

010: TRXD, Direct 模式 TXD/RXD, 输入/输出

011: TXD, Direct 模式 TXD, 输入

100: RXD, Direct 模式 RXD, 输出

101: IRQ, 中断请求, 输出

110: ROSCi, ATR 时钟外部输入

Bit2~0 GPIO1S[2:0]: GPIO1 引脚功能选择(仅通过 POR 复位)

000/111: 无功能, 输入 001: SDO, 4 线 SPI 数据, 输出

010: TRXD, Direct 模式 TXD/RXD, 输入/输出

011: TXD, Direct 模式 TXD, 输入

100: RXD, Direct 模式 RXD, 输出

101: IRQ, 中断请求, 输出

110: ROSCi, ATR 时钟外部输入

### 8.3.8 IO2: I/O 控制寄存器 2

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	GPIO3S[3:0]			
R/W	—	—	—	—	R/W			
ROR	0	0	0	0	0	0	0	0

Bit3~0 GPIO3S[3:0]: GPIO3 引脚功能选择(仅通过 POR 复位)

0000/0111: 无功能, 输入

0001: SDO, 4 线 SPI 数据, 输出

0010: TRXD, Direct 模式 TXD/RXD, 输入/输出

0011: TXD, Direct 模式 TXD, 输入

0100: RXD, Direct 模式 RXD, 输出

0101: IRQ, 中断请求, 输出

0110: ROSCi, ATR 时钟外部输入

1000: TBCLK, TX 位(数据)时钟, 输出

1001: RBCLK, RX 位(还原)时钟, 输出

1010: FSYCK, 即 XCLK1/1, 1/2, 1/4, 1/8 输出

1011: LIRCCLK, 内部 LIRC 去抖时钟, 输出

1100: EPA\_EN, 外部功率放大器(PA)使能, 输出

1101: ELAN\_EN, 外部 LNA 使能, 输出

1110: TRBCLK, TX 模式的 TBCLK 或 RX 模式的 RBCLK, 输出

1111: PDB, 掉电开关及 XO 使能, 输出

### 8.3.9 IO3: I/O 控制寄存器 3

Bit	7	6	5	4	3	2	1	0
Name	SDO_TEN	SPIPU	—	—	GPIO3S[3:1]			—
R/W	R/W	R/W	—	—	R/W			—
ROR	0	1	1	1	1	1	1	1

Bit3~0 GPIO3S[3:0]: GPIO3 引脚功能选择(仅通过 POR 复位)

Bit7 SDO\_TEN: SDO 三态使能(仅通过 POR 复位)

0: 除能

1: 使能

Bit6 SPIPU: 3 线 SPI 上拉使能(仅通过 POR 复位)

0: 除能

1: 使能

此位置 1 仅控制 CSN、SCK 和 SDIO 引脚的上拉功能。注意，4 线 SPI 的 SDO 引脚上拉功能需通过 GPIOPU[3:1]对应位设置。

Bit5~4 保留，必须设为“11”

Bit3~1 GPIOPU[3:1]: GPIO 引脚功能上拉使能控制(仅通过 POR 复位)

这些位分别控制 GPIO3~GPIO1 引脚的上拉功能。

Bit0 保留，必须设为“1”

### 8.3.10 FIFO1: FIFO 控制寄存器 1

Bit	7	6	5	4	3	2	1	0
Name	—	—	TXFFSA[5:0]					
R/W	—	—	R/W					
Reset	0	0	0	0	0	0	0	0

Bit7~6 保留，必须设为“00”

Bit5~0 TXFFSA[5:0]: TX FIFO 起始地址，用于 Block FIFO 模式

### 8.3.11 FIFO2: FIFO 控制寄存器 2

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	RXPL2F_EN	FFINF_EN	FFMG_EN	FFMG[1:0]	
R/W	—	—	—	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	1

Bit7~5 保留，必须设为“00”

Bit4 RXPL2F\_EN: RX 有效载荷(Payload)长度字节载入 FIFO 使能

0: 除能

1: 使能

当此位置高，指示有效载荷长度的字节将被加入数据包并载入 RXFIFO。在 RX 连续模式下(RXCON\_EN=1)，RXFIFO 将支持多笔有效载荷，此位也必须置高。

Bit3 FFINF\_EN: FIFO 无限制长度模式使能

0: 除能

1: 使能

Bit2 FFMG\_EN: FIFO 长度边距检测使能

0: 除能

1: 使能

Bit1~0 FFMG[1:0]: FIFO 长度边距选择

TXFIFO 剩余数据长度阈值:

00: 4 字节

01: 8 字节

10: 16 字节

11: 32 字节

RXFIFO 剩余空间阈值:

00: 4 字节

01: 8 字节

10: 16 字节

11: 32 字节

当 FFMG\_EN 位置高使能 FIFO 长度边距检测功能，且已通过这些位选择所需检

测的 FIFO 长度边距后，当所选的情况发生时，FIFOLTIF 标志位将被置高。此时，若对应的中断功能已使能，将产生中断。

### 8.3.12 PKT1：数据包控制寄存器 1

Bit	7	6	5	4	3	2	1	0
Name	TXPMLEN[7:0]							
R/W	R/W							
Reset	0	0	0	0	0	0	0	1

Bit 7~0 TXPMLEN[7:0]: TX 前导码长度

发送前导码长度=(TXPMLEN[7:0]+1)个字，1 字的长度 (1 个字节或 2 个字节) 由 PMLP\_EN 和 PMLPLEN 的设定决定。

### 8.3.13 PKT2：数据包控制寄存器 2

Bit	7	6	5	4	3	2	1	0
Name	PID[1:0]		TRAILER_EN	WHTFMT[0]	SYNCLEN[1:0]		—	—
R/W	R/W		R/W	R/W	R/W		—	—
Reset	0	0	1	0	0	1	0	0

Bit 7~6 PID[1:0]: TX 数据包 ID

当 PLH\_EN 位置高使能头码选项时，此 ID 会被放入有效载荷头码域的最高两位。

Bit 5 TRAILER\_EN: 连接码域使能

0: 除能

1: 使能

Bit 4 WHTFMT[0]: 数据白化格式选择 bit0

WHTFMT[1:0]=

00: VT230X,  $G(X)=X^7+X^6+X^5+X^4+1$

01: PN7,  $G(X)=X^7+X^4+1$

10: PN9-CCITT,  $G(X)=X^9+X^5+1$

11: PN9-IBM,  $G(X)=X^9+X^5+1$  WHTFMT[1]位于 PKT10 寄存器。

Bit 3~2 SYNCLEN[1:0]: TX/RX 模式同步码长度设置 bit1~0

SYNCWORD 长度=(SYNCLEN[1:0], SYNCLENLB)+1, SYNCLENLB 位于 PKT10 寄存器。

Bit 1~0 保留，必须设为“00”

### 8.3.14 PKT3：数据包控制寄存器 3

Bit	7	6	5	4	3	2	1	0
Name	MCH_EN	FEC_EN	CRC_EN	CRCFMT	PLEN_EN	PLHAC_EN	PLHLEN	PLH_EN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	0	0	0	0	0

Bit 7 MCH\_EN: 曼彻斯特编码使能

0: 除能

1: 使能

Bit 6 FEC\_EN: FEC 使能

0: 除能

1: 使能



- Bit5 CRC\_EN: CRC 域使能  
0: 除能  
1: 使能
- Bit4 CRCFMT: CRC 格式选择  
0: CCITT-16-CRC,  $G(X)=X^{16}+X^{12}+X^5+1$   
1: IBC-16-CRC,  $G(X)=X^{16}+X^{15}+X^2+1$
- Bit3 PLEN\_EN: 有效载荷长度域使能  
0: 除能  
1: 使能
- Bit2 PLHAC\_EN: 有效载荷头码地址校准使能控制  
0: 除能, PLHA[5:0]/PLHEA[7:0]位域可由用户自定义作为标志位使用  
1: 使能, PLHA[5:0]/PLHEA[7:0]位域必须包含相同地址, 否则数据包将被视为无效数据包  
注意, 是否包含 PLHEA[7:0]取决于 PLHLEN 位设定。
- Bit1 PLHLEN: 有效载荷头码长度选择  
0: 1 字节  
1: 2 字节
- Bit0 PLH\_EN: 有效载荷头码域使能  
0: 除能  
1: 使能

### 8.3.15 PKT4: 数据包控制寄存器 4

Bit	7	6	5	4	3	2	1	0
Name	WHT_EN	WHTSD[6:0]						
R/W	R/W	R/W						
Reset	0	0	1	1	0	1	1	0

- Bit 7 WHT\_EN: 数据白化使能  
0: 除能  
1: 使能

- Bit 6~0 WHTSD[6:0]: 数据白化种子 bit 6~0  
WHTSD[8:7]位于 PKT15 寄存器。

### 8.3.16 PKT5: 数据包控制寄存器 5

Bit	7	6	5	4	3	2	1	0
Name	TXDLEN[7:0]							
R/W	R/W							
Reset	0	1	0	0	0	0	0	0

- Bit 7~0 TXDLEN[7:0]: TX 数据长度(单位: 字节; 仅用于 Burst 模式)

### 8.3.17 PKT6: 数据包控制寄存器 6

Bit	7	6	5	4	3	2	1	0
Name	RXDLEN[7:0]							
R/W	R/W							
Reset	0	1	0	0	0	0	0	0

Bit 7~0 RXDLEN[7:0]: RX 数据长度(单位: 字节; 仅用于 Burst 模式 )

当 PLEN\_EN 位清零时, 接收的数据长度取决于此位域。当此寄存器被读取时, 所读出的数值表示 RXFIFO 中的数据长度。此寄存器被读取出的默认值是 00h。

### 8.3.18 PKT7: 数据包控制寄存器 7

Bit	7	6	5	4	3	2	1	0
Name	RXPID[1:0]		DLY_RXS[2:0]			DLY_TXS[2:0]		
R/W	R		R/W			R/W		
Reset	0	0	1	0	0	0	0	0

Bit7~6 RXPID[1:0]: 接收数据包 PID(只读)

Bit5~3 DLY\_RXS[2:0]: RX 模块使能后稳定时间选择

000: 4 $\mu$ s  
001: 8 $\mu$ s  
010: 12 $\mu$ s  
011: 16 $\mu$ s  
100: 20 $\mu$ s  
101: 32 $\mu$ s  
110: 64 $\mu$ s  
111: 100 $\mu$ s

这些位用于选择 RX 使能后到 RX 稳定前的等待时间。此时间应该大于 RXDCOCTurbo 模式默认延迟时间(6 $\mu$ s)。

Bit2~0 DLY\_TXS[2:0]: 进入 TX 模式前的 TX 启动(延迟)时间

000: 0 $\mu$ s  
001: 10 $\mu$ s  
010: 20 $\mu$ s  
011: 40 $\mu$ s  
100: 60 $\mu$ s  
101: 80 $\mu$ s  
110: 100 $\mu$ s  
111: 120 $\mu$ s

该时间用于 ARK 模式下发送器和接收器间的时序调整。

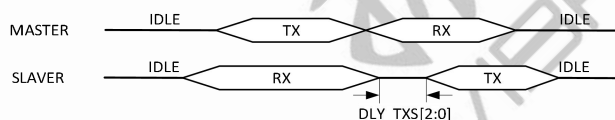


图 33 DLY\_TXS 示意图

### 8.3.19 PKT8: 数据包控制寄存器 8

Bit	7	6	5	4	3	2	1	0
Name	—	—	PLHA[5:0]					
R/W	—	—	R/W					
Reset	0	0	0	0	0	0	0	0

Bit7~6 保留，必须设为“00”

Bit5~0 PLHA[5:0]: 有效载荷头码地址，用于支持广播模式

RX 模式下若此地址为 0 表示不执行地址核对。

写：写数据到 TXPLHA[5:0]。读：从 RXPLHA[5:0]读数据。

### 8.3.20 PKT9: 数据包控制寄存器 9

Bit	7	6	5	4	3	2	1	0
Name	PLHEA[7:0]							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

Bit7~0 PLHEA[7:0]: 有效载荷头码扩展地址，用于支持广播模式

RX 模式下若此地址为 0 表示不执行地址核对。

### 8.3.21 MOD1: 调制器控制寄存器 1

Bit	7	6	5	4	3	2	1	0
Name	DTR[7:0]							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

Bit7~0 DTR[7:0]

DTR[8:0]: 数据速率分频器，DTR[8]位于 MOD2 寄存器。

数据速率= $f_{XTAL}/[32 \times (DTR[8:0]+1)]$ ，这里的数据速率表示 TBCLK。注意，DTR[8:0]只能是奇数。

### 8.3.22 MOD2: 调制器控制寄存器 2

Bit	7	6	5	4	3	2	1	0
Name	RXFDOS[11:8]				DITHER[1:0]		—	DTR[8]
R/W	R/W				R/W		—	R/W
Reset	1	0	0	1	0	0	0	0

Bit7~4 RXFDOS[11:8]

RXFDOS[11:0]: RX 中频偏移，RXFDOS[7:0]位于 MOD3 寄存器。

要先写 RXFDOS[11:8] 再写 RXFDOS[7:0] 才可完全更新 RXFDOS[11:0]。

$RXFDOS[11:0] = \text{floor}\{(f_{if}/f_{XTAL}) \times 2^{17}\}$

Bit3~2 DITHER[1:0]: 高频振动值

Bit1 保留，必须设为“0”

Bit0 DTR[8]

DTR[8:0]: 数据速率分频，DTR[7:0]位于 MOD1 寄存器。

数据速率= $f_{XTAL}/[32 \times (DTR[8:0]+1)]$ ，这里的数据速率表示 TBCLK。注意，DTR[8:0]只能是奇数。

### 8.3.23 MOD3: 调制器控制寄存器 3

Bit	7	6	5	4	3	2	1	0
Name	RXFDOS[7:0]							
R/W	R/W							
Reset	1	0	0	1	1	0	1	0

Bit7~0 RXFDOS[7:0]

RXFDOS[11:0]: RX 中频偏移, RXFDOS[11:8]位于 MOD2 寄存器

要先写 RXFDOS[11:8] 再写 RXFDOS[7:0] 才可完全更新 RXFDOS[11:0]。

$$\text{RXFDOS}[11:0] = \text{floor}\{(\text{fir}/\text{f}_{\text{XTAL}}) \times 2^{17}\}$$

### 8.3.24 DM1: 解调器控制寄存器 1

Bit	7	6	5	4	3	2	1	0
Name	—	—	MDIV[5:0]					
R/W	—	—	R/W					
Reset	0	0	0	0	0	0	1	1

Bit7~6 保留, 必须设为“00”

Bit5~0 MDIV[5:0]: 解调器工作时钟分频

$$\text{DMCLK} = \text{ADCLK} / (\text{MDIV}[5:0] + 1)$$

### 8.3.25 DM2: 解调器控制寄存器 2

Bit	7	6	5	4	3	2	1	0
Name	—	—	SDR[5:0]					
R/W	—	—	R/W					
Reset	0	0	0	0	0	0	1	1

Bit7~6 保留, 必须设为“00”

Bit5~0 SDR[5:0]: 相位提取后的解调器工作时钟

$$\text{SDR}[5:0] + 1 = \text{DMCLK} / (8 \times \text{DATA\_RATE}), \text{ 这里的 DATA\_RATE 表示 RBCLK.}$$

### 8.3.26 DM3: 解调器控制寄存器 3

Bit	7	6	5	4	3	2	1	0
Name	CSF_SW_EN	FD_MOD[6:0]						
R/W	R/W	R/W						
Reset	1	1	1	0	0	0	0	0

Bit7 CSF\_SW\_EN: 通道选择滤波器自动频宽切换使能

0: 除能

1: 使能

Bit6~0 FD\_MOD[6:0]: 频率偏移调节器

$$\text{FD\_MOD} = \text{round}((h / (\text{SDR}[5:0] + 1)) \times 128); \text{ h=解调系数}$$

$$\text{SDR}[5:0] + 1 = \text{DMCLK} / (8 \times \text{DATA\_RATE})$$

### 8.3.27 DM4: 解调器控制寄存器 4

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	CFO_DSEL	—	PH_DIFF_MO	—
R/W	—	—	—	—	R/W	—	R/W	—
Reset	0	0	0	0	1	0	0	0

Bit7~4 保留, 必须设为“0000”

Bit3 CFO\_DSEL: CFO 校准域选择

0: 模拟域

1: 数字域

Bit2 保留, 必须设为“0”

Bit1 PH\_DIFF\_MOD: 相位差提取模式设置

0: 相位提取范围 $[-\pi/2, \pi/2]$

1: 相位提取范围 $[-\pi, \pi]$  Bit0 保留, 必须设为“0”

### 8.3.28 DM5: 解调器控制寄存器 5

Bit	7	6	5	4	3	2	1	0
Name	FD_HOLD[7:0]							
R/W	R/W							
Reset	0	0	1	1	0	0	0	0

Bit7~0 FD\_HOLD[7:0]: 前导码 + SYNCWORD 检测频率偏差阈值

### 8.3.29 DM8: 解调器控制寄存器 8

Bit	7	6	5	4	3	2	1	0
Name	M_RATIO[7:0]							
R/W	R/W							
Reset	0	1	0	0	0	0	0	0

Bit7~0 M\_RATIO[7:0]: 用于 CFO 计算

$$M\_RATIO = \text{round}(1/(MDIV[5:0]+1) \times 28)$$



## 8.4 Bank0 控制寄存器

上电复位后，所有控制寄存器被设置为初始值。软件复位后，除了位于 XO3 寄存器的 LIRC\_EN 和 LIRC\_OW 位之外，其它控制寄存器也将恢复至初始值。而前述控制位在软件复位后保持不变。

表 13 BANK0 寄存器位说明

地址	名称	位							
		7	6	5	4	3	2	1	0
20h	OM	PWR_SOFT	BAND_SEL[1:0]		—	ACAL_EN	RTX_EN	RTX_SEL	SX_EN
22h	SX1	—	D_N[6:0]						
23h	SX2	D_K[7:0]							
24h	SX3	D_K[15:8]							
25h	SX4	—	—	—	—	D_K[19:16]			
26h	STA1	—	—	—	CD_FLAG	—	OMST[2:0]		
28h	RSSI2	—				RSSI_CTHD[3:0]			
29h	RSSI3	RSSI_NEGDB[7:0]							
2Ah	RSSI4	RSSI_SYNC_OK[7:0]							
2Bh	ATR1	ATRCLK_DIV[1:0]		ATRCLKS	ATRTU	ATRCTM	ATRM[1:0]		ATR_EN
2Ch	ATR2	ATRCYC[7:0]							
2Dh	ATR3	ATRCYC[15:8]							
2Eh	ATR4	ATRRXAP[7:0]							
2Fh	ATR5	ATRRXEP[7:0]							
30h	ATR6	ATRRXEP[15:8]							
31h	ATR7	ARKNM[3:0]				—	ATR_WDLY[1:0]		ARK_EN
32h	ATR8	ARKRXAP[7:0]							
33h	ATR9	ATRCT[7:0]							
34h	ATR10	ATRCT[15:8]							
35h	ATR11	ATRCYCM	—				ATRRXAP[10:8]		
36h	PKT10	—	WHTFMT[1]	CRCBYTE0	CRCBIT0	CRCINV	SYNCLLEN	PMLPLEN	PMLP_EN
37h	PKT11	PMLPAT[7:0]							
38h	PKT12	PMLPAT[15:8]							
39h	PKT13	CRCSD[7:0]							
3Ah	PKT14	CRCSD[15:8]							
3Bh	PKT15	WHTSD[8:7]		OOKDT_TS[3:0]				OOKDT_	OOKDT_
3Ch	XO1	XSHIFT[1:0]		—	XO_TRIM[4:0]				
3Dh	XO2	—		—	XO_SW	—	—		
3Eh	XO3	LIRCCAL_EN	LIRC_OW	LIRC_OP[4:0]					LIRC_EN
3Fh	TX2	—	—	CT_PAD[5:0]					

注：未列于此表格的地址是预留给将来使用。建议不要通过任何方式修改这些地址的初始值。

下面寄存器说明表格里的复位值指的是 Strobe 命令软件复位后的结果。

### 8.4.1 OM: 工作模式控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	PWR_SOFT	BAND_SEL[1:0]		—	ACAL_EN	RTX_EN	RTX_SEL	SX_EN
R/W	R/W	R/W		—	R/W	R/W	R/W	R/W
Reset	0	0	1	0	0	0	0	0

Bit7 PWR\_SOFT: RF 工作模式选择

- 0: RF 正常工作模式
- 1: RF 工程模式

Bit6~5 BAND\_SEL[1:0]: 频段选择(当 PWR\_SOFT=0)

- 00: 315MHz 频段, ODDIV=4
- 01: 433MHz 频段, ODDIV=2
- 10: 470~510MHz 频段, ODDIV=2
- 11: 868/915MHz 频段, ODDIV=1Bit4 保留, 必须设为“0”

Bit3 ACAL\_EN: 自动校准使能

- 0: 除能
- 1: 使能

当此位置高, VCO 和 RC 校准都使能。当 VCO 和 RC 校准都完成后, 此位由硬件自动清零。

Bit2 RTX\_EN: RX 或 TX 模式使能

- 0: 除能
- 1: 使能

RX 或 TX 模式由 RTX\_SEL 位选择之后, 此位置高将会使能所选的模式。

Bit1 RTX\_SEL: RX 或 TX 模式选择

- 0: RX 模式
- 1: TX 模式

Bit0 SX\_EN: 合成器使能(Standby 模式使能控制)

- 0: 除能
- 1: 使能

此位置高将使能 PFD、CP 和 VCO 功能。

### 8.4.2 SX1: 小数 N 分频合成器控制寄存器 1

Bit	7	6	5	4	3	2	1	0
Name	—	D_N[6:0]						
R/W	—	R/W						
Reset	0	0	1	1	0	1	1	0

Bit7 保留, 必须设为“0”

Bit6~0 D\_N[6:0]: RF 通道整数代码

$$D\_N[6:0] = \text{floor}(f_{RF} \times \text{ODDIV} / f_{XTAL})$$

例如, 默认 XO=16MHz 且 RF 频段=433.92MHz:

$$\rightarrow (433.92\text{MHz} \times 2) / 16\text{MHz} = 54.24$$

$$\rightarrow D\_N = 54$$

$$\rightarrow \text{Dec2Hex}(54) = 36$$

$$\rightarrow \text{Dec2Bin}(54) = 0110110$$

### 8.4.3 SX2: 小数 N 分频合成器控制寄存器 2

Bit	7	6	5	4	3	2	1	0
Name	D_K[7:0]							
R/W	R/W							
Reset	0	0	0	0	1	0	1	0

Bit7~0 D\_K[7:0]: RF 通道小数代码低字节

### 8.4.4 SX3: 小数 N 分频合成器控制寄存器 3

Bit	7	6	5	4	3	2	1	0
Name	D_K[15:8]							
R/W	R/W							
Reset	0	0	0	0	1	0	1	0

Bit7~0 D\_K[15:8]: RF 通道小数代码中间字节

### 8.4.5 SX4: 小数 N 分频合成器控制寄存器 4

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	D_K[19:16]			
R/W	—	—	—	—	R/W			
Reset	0	0	0	0	0	0	1	1

Bit7~4 保留, 必须设为“0000”

Bit3~0 D\_K[19:16]: RF 通道小数代码高字节

$$D\_K[19:0] = \text{floor}\{(\text{frF} \times \text{ODDIV} / \text{fXTAL} - D\_N[6:0]) \times 2^{20}\}$$

例如, 默认 XO=16MHz 且 RF 频段=433.92MHz:

$$\rightarrow (433.92\text{MHz} \times 2) / 16\text{MHz} = 54.24$$

$$\rightarrow D\_K = 0.12 \times 2^{20} = 251658$$

$$\rightarrow \text{Dec2Hex}(251658) = 3D70A$$

$$\rightarrow \text{Dec2Bin}(251658) = 0011\_1101\_0111\_0000\_1010$$

### 8.4.6 STA1: 状态控制寄存器 1

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	CD_FLAG	—	OMST[2:0]		
R/W	—	—	—	R	—	R		
Reset	0	0	0	0	0	0	0	0

Bit7~5 保留, 必须设为“000”

Bit4 CD\_FLAG: 载波检测标志位(只读)

当 DEMOD\_EN 拉高且载波检测没问题时, 此标志位将被硬件置高。这里的 DEMOD\_EN 高电平是内部信号, 在 Direct 模式(DIR\_EN=1)下由内部状态机产生, 或在 Burst 模式(DIR\_EN=0)下接收到 RX 命令后产生。此标志位在 RX\_EN 上升沿时被自动清零。这里说的 RX\_EN 上升沿在 Direct 模式下设置 RTX\_SEL=0 且 RTX\_EN=1 后由内部状态机产生, 或在 Burst 模式下接收到 RX 命令后产生。

Bit3 保留, 必须设为“0”

Bit2~0 OMST[2:0]: 工作模式状态指示(只读)

000: DeepSleep 模式

001: Idle 模式

- 010: LightSleep 模式
- 011: Standby 模式
- 100: TX 模式
- 101: RX 模式
- 110: VCO 校准模式
- 111: 未定义

#### 8.4.7 RSSI2: RSSI 控制寄存器 2

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	RSSI_CTHD[3:0]			
R/W	—	—	—	—	R/W			
Reset	0	0	0	0	1	0	1	0

Bit7~4 保留, 必须设为“0000”

Bit3~0 RSSI\_CTHD[3:0]: 载波检测 RSSI 阈值  
 $(\text{RSSI\_CTHD}[3:0] \times 2 + 1) + 74 = \text{载波检测 RSSI 阈值}$

#### 8.4.8 RSSI3: RSSI 控制寄存器 3

Bit	7	6	5	4	3	2	1	0
Name	RSSI_NEGDB[7:0]							
R/W	R							
Reset	0	0	0	0	0	0	0	0

Bit7~0 RSSI\_NEGDB[7:0]: RSSI 值(单位: -dB)  
 此值为实时测量值。

#### 8.4.9 RSSI4: RSSI 控制寄存器 4

Bit	7	6	5	4	3	2	1	0
Name	RSSI_SYNC_OK[7:0]							
R/W	R							
Reset	0	0	0	0	0	0	0	0

Bit7~0 RSSI\_SYNC\_OK[7:0]: 同步码检测正确时的 RSSI 快摄值

#### 8.4.10 ATR1: 自动 TX/RX 控制寄存器 1

Bit	7	6	5	4	3	2	1	0
Name	ATRCLK_DIV[1:0]		ATRCLKS	ATRTU	ATRCTM	ATRM[1:0]		ATR_EN
R/W	R/W		R/W	R/W	R/W	R/W		R/W
Reset	1	1	0	0	1	0	0	0

Bit7~6 ATRCLK\_DIV[1:0]: ATR 时钟频率分频

- 00: 1/1, ATRCLK=32768Hz
- 01: 1/4, ATRCLK=8192Hz
- 10: 1/8, ATRCLK=4096Hz
- 11: 1/16, ATRCLK=2048Hz

Bit5 ATRCLKS: ATRCLK 时钟源选择

- 0: 来自内部 LIRC 时钟
- 1: 来自 GPIO<sub>n</sub> 引脚的外部 ROSC<sub>i</sub> 时钟输入

Bit4 ATRTU: 自动 TRX 单位时间选择

0: 250μs

1: 1ms, 用于支持低数据速率应用

此位用于选择 ATRRX 有效周期 (ATTRXAP[10:0])、ATTRX 扩展周期 (ATTRXEP[15:0])以及 ARKRX 有效周期(ARKRXAP[7:0])的单位时间。

Bit3 ATRCTM: 自动 TRX 定时模式选择

0: 单次模式, 每次发生 ATR 事务时启动 ATRCT 定时器

1: 连续模式, 接收到 Idle 命令时启动 ATRCT 定时器, 当 ATR\_EN=0 或者 ATRCTM=0 时停止 ATRCT 定时器

Bit2~1 ATRM[1:0]: 自动 TRX 模式选择

00: ATRWOT 模式

01: ATRWOR 模式

10/11: ATRWTM 模式

Bit0 ATR\_EN: 自动 TRX 使能

0: 除能

1: 使能

当工作模式状态由 DeepSleep/LightSleep 模式切换到 Idle 模式时 ATR 功能启动。

#### 8.4.11 ATR2: 自动 TX/RX 控制寄存器 2

Bit	7	6	5	4	3	2	1	0
Name	ATRCYC[7:0]							
R/W	R/W							
Reset	1	1	1	1	1	1	1	1

Bit7~0 ATRCYC[7:0]: ATRCT 定时器界限值低字节

#### 8.4.12 ATR3: 自动 TX/RX 控制寄存器 3

Bit	7	6	5	4	3	2	1	0
Name	ATRCYC[15:8]							
R/W	R/W							
Reset	0	0	0	0	1	1	1	1

Bit7~0 ATRCYC[7:0]: ATRCT 定时器界限值高字节

ATRCYCM=0 兼容模式(默认):

唤醒周期=ATRCLK 周期×ATRCYC[15:0]+LIRCCLK 周期, ATRCYC[15:0]≠0。默认唤醒周期为 2 秒。

ATRCYCM=1 VT2303R 模式:

唤醒周期=ATRCLK 周期×(ATRCYC[15:0]+1), ATRCYC[15:0]≠0。默认唤醒周期为 2 秒。

#### 8.4.13 ATR4: 自动 TX/RX 控制寄存器 4

Bit	7	6	5	4	3	2	1	0
Name	STRRXAP[7:0]							
R/W	R/W							
Reset	0	0	1	0	0	1	1	1

Bit7~0 ATTRXAP[7:0]: ATRRX 有效周期低字节

ATTRRX 有效周期高字节 ATTRXAP[10:8]位于 ATR11 寄存器。



有效周期=单位时间 $\times$ (ATTRXAP[10:0]+1); 单位时间为 250 $\mu$ s 或 1ms, 由 ATRTU 位决定。由于默认的单位时间是 250 $\mu$ s, 默认的 ATTRX 有效周期为 10ms。

#### 8.4.14 ATR5: 自动 TX/RX 控制寄存器 5

Bit	7	6	5	4	3	2	1	0
Name	ATTRXEP[7:0]							
R/W	R/W							
Reset	1	0	0	0	1	1	1	1

Bit7~0 ATTRXEP[7:0]: ATR RX 扩展周期低字节

#### 8.4.15 ATR6: 自动 TX/RX 控制寄存器 6

Bit	7	6	5	4	3	2	1	0
Name	ATTRXEP[15:8]							
R/W	R/W							
Reset	0	0	0	0	0	0	0	1

Bit7~0 ATTRXEP[15:8]: ATR RX 扩展周期高字节

扩展周期=单位时间 $\times$ (ATTRXEP[15:0]+1); 单位时间为 250 $\mu$ s 或 1ms, 由 ATRTU 位决定。由于默认的单位时间是 250 $\mu$ s, 默认的 ATTRX 扩展周期为 100ms。

#### 8.4.16 ATR7: 自动 TX/RX 控制寄存器 7

Bit	7	6	5	4	3	2	1	0
Name	ARKNM[3:0]				—	ATR_WDLY[1:0]		ARK_EN
R/W	R/W				—	R/W		R/W
Reset	0	1	1	1	0	0	1	0

Bit7~4 ARKNM[3:0]: ARK 重复周期次数

最大重复周期次数=ARKNM[3:0]+1

Bit3 保留, 必须设为“0”

Bit2~1 ATR\_WDLY[1:0]: 自动唤醒延迟时间

00: 244 $\mu$ s

01: 488 $\mu$ s

10: 732 $\mu$ s

11: 976 $\mu$ s

Bit0 ARK\_EN: 自动重发/ACK 使能

0: 除能

1: 使能

#### 8.4.17 ATR8: 自动 TX/RX 控制寄存器 8

Bit	7	6	5	4	3	2	1	0
Name	ARKRXAP[7:0]							
R/W	R/W							
Reset	0	0	1	0	0	1	1	1

Bit7~0 ARKRXAP[7:0]: ARKRX 有效周期

有效周期=单位时间 $\times$ (ARKRXAP[7:0]+1); 单位时间为 250 $\mu$ s 或 1ms, 由 ATRTU 位决定。由于默认的单位时间是 250 $\mu$ s, 默认的 ARKRX 有效周期为 10ms。

#### 8.4.18 ATR9：自动 TX/RX 控制寄存器 9

Bit	7	6	5	4	3	2	1	0
Name	ATRCT[7:0]							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

Bit7~0 ATRCT[7:0]: ATR 周期低字节

#### 8.4.19 ATR10：自动 TX/RX 控制寄存器 10

Bit	7	6	5	4	3	2	1	0
Name	ATRCT[15:8]							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

Bit7~0 ATRCT[15:8]: ATR 周期高字节

读 ATRCT[15:0]将得到当前计数值。由于 8 位 SPI 数据长度的限制，读取 ATR9 寄存器时会快速抓取完整的 16 位数据放入读取寄存器缓冲器。用户须连续读取 ATR9 和 ATR10 寄存器(不被间断)才能得到正确数据。

写数据到 ATRCT[15:0]将更新计数值。先写 ATR9 寄存器再写 ATR10 寄存器才能触发 ATRCT 写功能。此定时器的更新机制用于双向 RF 系统调整主从端时隙。

#### 8.4.20 ATR11：自动 TX/RX 控制寄存器 11

Bit	7	6	5	4	3	2	1	0
Name	ATRCYCM	—	—	—	—	ATRRXAP[10:8]		
R/W	R/W	—	—	—	—	R/W		
Reset	0	0	0	0	0	0	0	0

Bit7 ATRCYCM: ATR 周期计算模式

0: 保留

1: VT2303R 模式

关于周期计算的更多细节，详见 ATR2 与 ATR3 寄存器。

Bit6~3 保留，必须设为“0000”

Bit2~0 ATRRXAP[10:8]: ATRRX 有效周期高字节

ATRRX 有效周期低字节 ATRRXAP[7:0]位于 ATR4 寄存器。

有效周期=单位时间×(ATRRXAP[10:0]+1); 单位时间为 250μs 或 1ms，由 ATRTU 位决定。由于默认的单位时间是 250μs，默认的 ATRRX 有效周期为 10ms。

#### 8.4.21 PKT10：数据包控制寄存器 10

Bit	7	6	5	4	3	2	1	0
Name	—	WHTFMT[1]	CRCBYTE0	CRCBIT0	CRCINV	SYNCLLENLB	PMLPLEN	PMLP_EN
R/W	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	1	0	0

Bit7 保留，必须设为“0”

Bit6 WHTFMT[1]: 数据白化格式选择 bit1

WHTFMT[1:0]=

00: VT230X,  $G(X)=X^7+X^6+X^5+X^4+1$

- 01: PN7,  $G(X)=X^7+X^4+1$   
 10: PN9-CCITT,  $G(X)=X^9+X^5+1$   
 11: PN9-IBM,  $G(X)=X^9+X^5+1$  WHTFMT[0]位于 PKT2 寄存器。
- Bit5 CRCBYTEO: CRC 字节顺序  
 0: CRC 高字节优先  
 1: CRC 低字节优先
- Bit4 CRCBITO: CRC 位顺序  
 0: CRC 高位优先  
 1: CRC 低位优先
- Bit3 CRCINV: CRC 位反序功能控制  
 0: 不反序  
 1: 反序
- Bit2 SYNCLENLB: TX/RX 模式 SYNCWORD 长度设置最低位  
 SYNCWORD 长度=(SYNCLEN[1:0],SYNCLENLB)+1, SYNCLEN[1:0]位于 PKT2 寄存器。
- Bit1 PMLPLEN: 前导码长度选择(当 PMLP\_EN=1)  
 0: 1 个字节-PMLPAT[7:0]  
 1: 2 个字节-PMLPAT[15:0], 低字节优先
- Bit0 PMLP\_EN: 前导码样式选择  
 0: 前导码样式=取 SYNCWORD 最高位, 接着进行 1/0 翻转  
 如: SYNCWORD 最高位=0, 前导码=01010101(b)  
 SYNCWORD 最高位=1, 前导码=10101010(b)  
 1: 前导码样式来自 PMLPAT[15:0]

#### 8.4.22 PKT11: 数据包控制寄存器 11

Bit	7	6	5	4	3	2	1	0
Name	PMLPAT[7:0]							
R/W	R/W							
Reset	0	1	0	1	0	1	0	1

Bit 7~0 PMLPAT[7:0]: 前导码低字节

#### 8.4.23 PKT12: 数据包控制寄存器 12

Bit	7	6	5	4	3	2	1	0
Name	PMLPAT[15:8]							
R/W	R/W							
Reset	0	1	0	1	0	1	0	1

Bit 7~0 PMLPAT[15:8]: 前导码高字节

#### 8.4.24 PKT13: 数据包控制寄存器 13

Bit	7	6	5	4	3	2	1	0
Name	CRCSD[7:0]							
R/W	R/W							
Reset	1	1	1	1	1	1	1	1

Bit 7~0 CRCSD[7:0]: CRC 种子低字节

#### 8.4.25 PKT14: 数据包控制寄存器 14

Bit	7	6	5	4	3	2	1	0
Name	CRCSD[15:8]							
R/W	R/W							
Reset	1	1	1	1	1	1	1	1

Bit 7~0 CRCSD[15:8]: CRC 种子高字节

#### 8.4.26 PKT15: 数据包控制寄存器 15

Bit	7	6	5	4	3	2	1	0
Name	WHTSD[8:7]		OOKDT_TS[3:0]				OOKDT_POR	OOKDT_EN
R/W	R/W		R/W				R/W	R/W
Reset	0	0	0	1	0	1	0	1

Bit7 MCH\_EN: 曼彻斯特编码使能

Bit7~6 WHTSD[8:7]: 数据白化种子 bit8~7

WHTSD[6:0]位于 PKT4 寄存器。注意, 仅 PN9-CCIT 和 PN9-IBM 数据白化多项式会用到完整的 WHTSD[8:0]9 个位。

Bit5~2 OOKDT\_TS[3:0]: OOK 占空比调谐时间选择

0000: 2 $\mu$ s  
0001: 4 $\mu$ s  
0010: 6 $\mu$ s  
0011: 8 $\mu$ s  
...(步进: 2 $\mu$ s)  
1100: 26 $\mu$ s  
1101: 28 $\mu$ s  
1110: 30 $\mu$ s  
1111: 32 $\mu$ s

Bit1 OOKDT\_POR: OOK 占空比调谐极性

0: 延长 0 个占空比  
1: 延长 1 个占空比

Bit0 OOKDT\_EN: OOK 占空比调谐使能

0: 除能  
1: 使能

#### 8.4.27 XO1: XO 控制寄存器 1

Bit	7	6	5	4	3	2	1	0
Name	XSHIFT[1:0]		—	XO_TRIM[4:0]				
R/W	R/W		—	R/W				
Reset	0	0	0	1	0	0	0	0

Bit7~6 XSHIFT[1:0]: 晶振负载电容粗调, 以匹配不同 C<sub>LOAD</sub>

Bit5 保留, 必须设为“1”

Bit4~0 XO\_TRIM[4:0]: 晶振负载电容微调

#### 8.4.28 XO2: XO 控制寄存器 2

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	XO_SW	—	—	—	—
R/W	—	—	—	R/W	—	—	—	—
Reset	0	0	0	0	0	0	1	1

Bit7~5 保留, 必须设为“000”

Bit4 XO\_SW: 晶振负载电容开关  
0: 约 12~16pF  
1: 约 8~12pF

Bit3~0 保留, 必须设为“0011”

#### 8.4.29 XO3: XO 控制寄存器 3

Bit	7	6	5	4	3	2	1	0
Name	LIRCCAL_EN	LIRC_OW	LIRC_OP[4:0]					LIRC_EN
R/W	R/W	R/W	R/W					R/W
ROR	—	0	—	—	—	—	—	0
Reset	0	—	0	1	1	0	1	—

Bit7~5 保留, 必须设为“000”

Bit7 LIRCCAL\_EN: LIRC 校准使能控制  
0: 除能  
1: 使能

Bit6 LIRC\_OW: LIRC 覆写控制(仅通过 POR 复位)  
0: LIRC\_OP[4:0]来自校准引擎  
1: LIRC\_OP[4:0]来自控制寄存器

Bit5~1 LIRC\_OP[4:0]: LIRC 调整  
写数据到 LIRC\_OP[4:0]之后, 此值在 LIRC\_OW 位置高后生效。当从 LIRC\_OP[4:0]读取数据时, 实际数据源取决于 LIRC\_OW 位的设置。

Bit0 LIRC\_EN: LIRC 使能控制(仅通过 POR 复位)  
0: 除能  
1: 使能

#### 8.4.30 TX2: TX 控制寄存器 2

Bit	7	6	5	4	3	2	1	0
Name	—	—	CT_PAD[5:0]					
R/W	—	—	R/W					
Reset	0	0	1	1	0	0	0	0

Bit7~6 保留, 必须设为“00”

Bit5~0 CT\_PAD[5:0]: TXPAD 线性功率控制



## 8.5 Bank1 控制寄存器

上电复位后，所有控制寄存器被设置为初始值。软件复位后，这些寄存器也将恢复至初始值。

表 14 BANK1 寄存器位说明

地址	名称	位							
		7	6	5	4	3	2	1	0
20h	AGC1	MPT_0DB_E	—	OFFSET_HBSEL[2:0]			OFFSET_LBSEL[2:0]		
21h	AGC2	—			DBFS_OFFSET[2:0]			AGC_CMP_THD[1:0]	
22h	AGC3	—					IF_DETOK_THD[2:0]		
23h	AGC4	GAIN_SEL[3:0]			—		AGC_ST[2:0]		
26h	AGC7	GAIN_STB[7:0]							
2Ch	FCF1	—	—	SFRATIO[1:0]		GFD_COMP[3:0]			
2Dh	FCF2	FSCALE[7:0]							
2Eh	FCF3	—				FSCALE[11:8]			
2Fh	FCF4	CF_B12[7:0]							
30h	FCF5	—						CF_B12[9:8]	
31h	FCF6	CF_B13[7:0]							
32h	FCF7	—						CF_B13[9:8]	
33h	FCF8	CF_A12[7:0]							
34h	FCF9	—						CF_A12[9:8]	
35h	FCF10	CF_A13[7:0]							
36h	FCF11	—						CF_A13[9:8]	
37h	FCF12	CF_B22[7:0]							
38h	FCF13	—						CF_B22[9:8]	
39h	FCF14	CF_B23[7:0]							
3Ah	FCF15	—						CF_B23[9:8]	
3Bh	FCF16	CF_A22[7:0]							
3Ch	FCF17	—						CF_A22[9:8]	
3Dh	FCF18	CF_A23[7:0]							
3Eh	FCF19	—						CF_A23[9:8]	

注：未列于此表格的地址是预留给将来使用。建议不要通过任何方式修改这些地址的初始值。

下面寄存器说明表格里的复位值指的是 Strobe 命令软件复位后的结果。

### 8.5.1 AGC1: AGC 控制寄存器 1

Bit	7	6	5	4	3	2	1	0
Name	MPT_0DB_E	—	OFFSET_HBSEL[2:0]			OFFSET_LBSEL[2:0]		
R/W	R/W	—	R/W			R/W		
Reset	0	0	0	1	1	0	0	0

Bit7 MPT\_0DB\_EN: 迫使曲线 1 的最大点=0dB

0: 除能

1: 使能

- Bit6 保留，必须设为“0”
- Bit5~3 OFFSET\_HBSEL[2:0]: AGC 最大点选择  
最大点= $-6+2\times\text{offset\_HB}$
- Bit2~0 OFFSET\_LBSEL[2:0]: AGC 最小点选择  
最小点=最大点-22+2×offset\_LB

表 15 OFFSET\_HBSEL 位说明

OFFSET_HBSEL[2:0]	offset_HB
000	0
001	1
010	2
011	3
100	-4
101	-3
110	-2
111	-1

表 16 OFFSET\_LBSEL 位说明

OFFSET_LBSEL[2:0]	offset_LB
000	0
001	1
010	2
011	3
100	-4
101	-3
110	-2
111	-1

### 8.5.2 AGC2: AGC 控制寄存器 2

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	DBFS_OFFSET[2:0]			AGC_CMP_THD[1:0]	
R/W	—	—	—	R/W			R/W	
Reset	0	0	0	1	0	1	0	0

- Bit7~5 保留，必须设为“000”
- Bit4~2 DBFS\_OFFSET[2:0]: GFSK 调制的对数偏移量
- Bit1~0 AGC\_CMP\_THD[1:0]: AGC 比较次数阈值  
00: 连续的 AGC 比较直到检测到 SYNCWORD  
01~11: 比较次数阈值

### 8.5.3 AGC3: AGC 控制寄存器 3

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	IF_DETOK_THD[2:0]		
R/W	—	—	—	—	—	R/W		
Reset	0	0	0	0	0	1	0	0

Bit7~3 保留, 必须设为“00000”

Bit2~0 IF\_DETOK\_THD[2:0]: IF 检测正常阈值

经过一段增益稳定时间(由 AGC7 寄存器决定)后, AGC 电路在开始检测 IF 信号强度前会先等待(IF\_DETOK\_THD×8)个 ADCLK 周期。

#### 8.5.4 AGC4: AGC 控制寄存器 4

Bit	7	6	5	4	3	2	1	0
Name	GAIN_SEL[3:0]				—	AGC_ST[2:0]		
R/W	R				—	R		
Reset	0	0	0	1	0	0	0	1

Bit7~4 GAIN\_SEL[3:0]: 增益曲线选择

Bit3 保留, 必须设为“0”

Bit2~0 AGC\_ST[2:0]: AGC 状态机的状态

100: AGC 完成

#### 8.5.5 AGC7: AGC 控制寄存器 7

Bit	7	6	5	4	3	2	1	0
Name	GAIN_STB[7:0]							
R/W	R/W							
Reset	0	0	1	1	0	0	0	0

Bit7~0 GAIN\_STB[7:0]: 增益稳定计数

增益稳定延迟计数(单位: ADCLK 周期)=GAIN\_STB[7:0]×2

#### 8.5.6 FCF1: 滤波器系数控制寄存器 1

Bit	7	6	5	4	3	2	1	0
Name	—	—	SFRATIO[1:0]		GFD_COMP[3:0]			
R/W	—	—	R/W		R/W			
Reset	0	0	0	0	0	1	1	0

Bit7 MCH\_EN: 曼彻斯特编码使能

Bit 7~6 保留, 必须设为“00”

Bit5~4 SFRATIO[1:0]: 平滑滤波器比率选择

00: 1/1

01: 1/16

10: 1/64

11: 1/128

Bit3~0 GFD\_COMP[3:0]: 101/010 位序列之高斯滤波补偿

[3]: 1/8; [2]: 1/16; [1]: 1/32; [0]: 1/64 举例:

0000: 无补偿

...

1111: 补偿后的值=(1+0.234375)倍=[1+(1/8+1/16+1/32+1/64)]倍

### 8.5.7 FCF2：滤波器系数控制寄存器 2

Bit	7	6	5	4	3	2	1	0
Name	FSCALE[7:0]							
R/W	R/W							
Reset	0	1	0	0	0	1	0	0

Bit7 FSCALE[7:0]：频率偏移比例参数低字节

### 8.5.8 FCF3：滤波器系数控制寄存器 3

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	FSCALE[11:8]			
R/W	—	—	—	—	R/W			
Reset	0	0	0	0	0	1	0	0

Bit7~4 保留，必须设为“0000”

Bit3~0 FSCALE[11:8]：频率偏移比例参数高字节

若数据速率范围在 250kbps~100kbps，则需进行此滤波。

$$FSCALE[11:0] = \text{round}\{(h \times f_s / f_{XTAL}) \times 2^{15}\}$$

$$h = (2 \times \text{频率偏移}) / \text{数据速率}$$

其中 h 为解调系数，通过频率偏移和数据速率计算。

### 8.5.9 FCF4：滤波器系数控制寄存器 4

Bit	7	6	5	4	3	2	1	0
Name	CF_B12[7:0]							
R/W	R/W							
Reset	1	0	0	0	0	1	0	1

### 8.5.10 FCF5：滤波器系数控制寄存器 5

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	CF_B12[9:8]	
R/W	—	—	—	—	—	—	R/W	
Reset	0	0	0	0	0	0	1	0

### 8.5.11 FCF6：滤波器系数控制寄存器 6

Bit	7	6	5	4	3	2	1	0
Name	CF_B13[7:0]							
R/W	R/W							
Reset	1	0	0	0	1	0	1	0

### 8.5.12 FCF7：滤波器系数控制寄存器 7

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	CF_B13[9:8]	
R/W	—	—	—	—	—	—	R/W	
Reset	0	0	0	0	0	0	0	0

### 8.5.13 FCF8：滤波器系数控制寄存器 8

Bit	7	6	5	4	3	2	1	0
Name	CF_A12[7:0]							
R/W	R/W							
Reset	0	0	0	1	0	0	1	0

### 8.5.14 FCF9：滤波器系数控制寄存器 9

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	CF_A12[9:8]	
R/W	—	—	—	—	—	—	R/W	
Reset	0	0	0	0	0	0	0	0

当数据速率在 49kbps~2kbps 范围内时，需要进行以下平滑滤波。

$$CF\_A12[9:0]=\text{mod}(210+[(SFRATIO[1:0]-1)\times 28],210)$$

### 8.5.15 FCF10：滤波器系数控制寄存器 10

Bit	7	6	5	4	3	2	1	0
Name	CF_A13[7:0]							
R/W	R/W							
Reset	0	0	1	0	1	0	1	1

### 8.5.16 FCF11：滤波器系数控制寄存器 11

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	CF_A13[9:8]	
R/W	—	—	—	—	—	—	R/W	
Reset	0	0	0	0	0	0	1	1

### 8.5.17 FCF12：滤波器系数控制寄存器 12

Bit	7	6	5	4	3	2	1	0
Name	CF_B22[7:0]							
R/W	R/W							
Reset	0	0	0	1	0	1	0	0

### 8.5.18 FCF13：滤波器系数控制寄存器 13

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	CF_B22[9:8]	
R/W	—	—	—	—	—	—	R/W	
Reset	0	0	0	0	0	0	0	1

### 8.5.19 FCF14：滤波器系数控制寄存器 14

Bit	7	6	5	4	3	2	1	0
Name	CF_B23[7:0]							
R/W	R/W							
Reset	0	0	1	0	0	0	0	1



### 8.5.20 FCF15: 滤波器系数控制寄存器 15

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	CF_B23[9:8]	
R/W	—	—	—	—	—	—	R/W	
Reset	0	0	0	0	0	0	0	0

### 8.5.21 FCF16: 滤波器系数控制寄存器 16

Bit	7	6	5	4	3	2	1	0
Name	CF_A22[7:0]							
R/W	R/W							
Reset	0	1	1	1	1	0	0	0

### 8.5.22 FCF17: 滤波器系数控制寄存器 17

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	CF_A22[9:8]	
R/W	—	—	—	—	—	—	R/W	
Reset	0	0	0	0	0	0	0	0

### 8.5.23 FCF18: 滤波器系数控制寄存器 18

Bit	7	6	5	4	3	2	1	0
Name	CF_A23[7:0]							
R/W	R/W							
Reset	0	0	1	0	1	0	0	0

### 8.5.24 FCF19: 滤波器系数控制寄存器 19

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	CF_A23[9:8]	
R/W	—	—	—	—	—	—	R/W	
Reset	0	0	0	0	0	0	0	0

FCF4~FCF19 寄存器定义了 8 组 IIR 系数, 不同 XTAL 时钟情况下这些寄存器的建议设定值如下面表格所示。

表 17 滤波器系数建议

fXTAL	16MHz	16MHz	16MHz	16MHz	16MHz
fS	250kbps	125kbps	50kbps	10kbps	2kbps
fD	93.75kHz	46.875kHz	18.75kHz	40kHz	8kHz
D_K[19:0] (H)	$f_{RF} \times ODDIV / f_{XTAL}$ , 取小数				
D_N[6:0] (H)	$f_{RF} \times ODDIV / f_{XTAL}$ , 取整数				
SFRATIO[1:0] (D)	0	0	0	1	3
FSCALE[11:0] (H)	294	119	4C	A4	20
CF_B12[9:0] (H)	2CA	01D	0	0	0
CF_B13[9:0] (H)	62	346	0	0	0
CF_A12[9:0] (H)	358	22	0	310	302
CF_A13[9:0] (H)	3E9	331	0	0	0
CF_B22[9:0] (H)	3B3	386	0	0	0
CF_B23[9:0] (H)	3E	12	0	0	0
CF_A22[9:0] (H)	3E9	8	0	0	0
CF_A23[9:0] (H)	39	8	0	0	0

## 8.6 Bank2 控制寄存器

上电复位后，所有控制寄存器被设置为初始值。软件复位后，这些寄存器也将恢复至初始值。

表 18 BANK2 寄存器位说明

地址	名称	位							
		7	6	5	4	3	2	1	0
21h	CFG0	保留							
2Eh	CFG1	保留							
2Fh	CFG2	保留							
33h	CFG3	保留							
34h	CFG4	保留							
39h	CFG5	保留							
3Bh	CFG6	保留							

注：未列于此表格的地址是预留给将来使用。建议不要通过任何方式修改这些地址的初始值。

表 19 BANK2 寄存器设置说明

地址	名称	频段	
		433MHz	868MHz
21h	CFG0	97h	
2Eh	CFG1	68h	
2Fh	CFG2	16h ( $\leq 50$ kbps: 06h)	96h ( $\leq 50$ kbps: 86h)
33h	CFG3	01h (OOK mode)/41h (GFSK mode)	
34h	CFG4	90h	
39h	CFG5	9Ch	
3Bh	CFG6	5Dh	

## 9、应用电路图

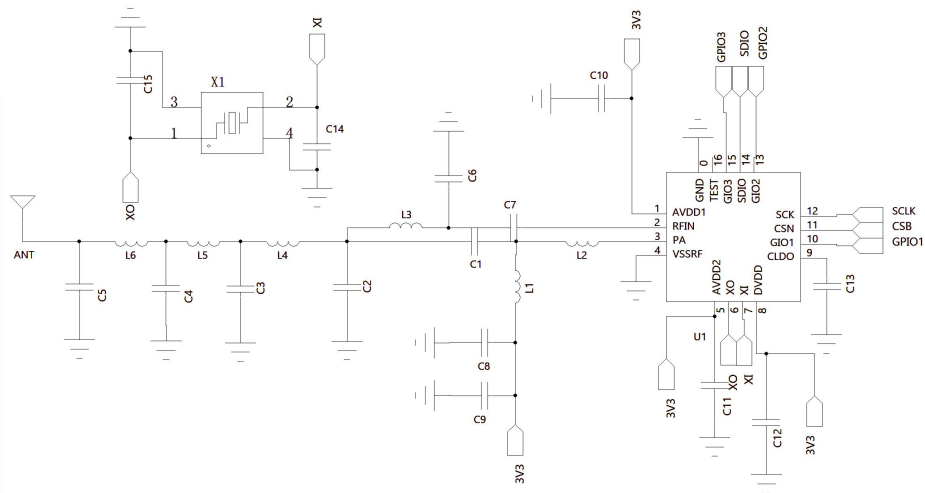


图 34 参考应用电路图

表 20 典型电路参考 BOM

标号	说明	值（匹配到 $\lambda/4$ 天线）					单位
		315MHz	433MHz	470MHz	868MHz	915MHz	
X1	晶体, $\pm 20$ ppm	16					MHz
L1	匹配网络电感, $\pm 10\%$ , 叠层电感	120	120	120	120	120	nH
L2	匹配网络电感, $\pm 10\%$ , 叠层电感	0R	2	0R	0R	0R	nH
L3	匹配网络电感, $\pm 10\%$ , 叠层电感	120	68	56	18	15	nH
L4	匹配网络电感, $\pm 10\%$ , 叠层电感	27	22	18	5.6	3.9	nH
L5	匹配网络电感, $\pm 10\%$ , 叠层电感	47	27	22	0R	0R	nH
L6	匹配网络电感, $\pm 10\%$ , 叠层电感	39	27	15	8.2	7.5	nH
C1	匹配网络电容, $\pm 0.25$ pF, NP0, 50 V	100	56	68	100	47	pF
C2	匹配网络电容, $\pm 0.25$ pF, NP0, 50 V	NC	NC	NC	NC	NC	pF
C3	匹配网络电容, $\pm 0.25$ pF, NP0, 50 V	12	6.2	10	5	5	pF
C4	匹配网络电容, $\pm 0.25$ pF, NP0, 50 V	12	7	8.2	NC	NC	pF
C5	匹配网络电容, $\pm 0.25$ pF, NP0, 50 V	6.8	6.8	3.3	2.4	1.5	pF
C6	匹配网络电容, $\pm 0.25$ pF, NP0, 50 V	NC	NC	NC	NC	NC	pF
C7	匹配网络电容, $\pm 0.25$ pF, NP0, 50 V	100	100	100	100	100	pF
C8	电源滤波电容, $\pm 20\%$ , X7R, 25 V	100	100	100	100	100	nF
C9	电源滤波电容, $\pm 20\%$ , X7R, 25 V	1	1	1	1	1	uF
C10	电源滤波电容, $\pm 20\%$ , X7R, 25 V	1	1	1	1	1	uF
C11	电源滤波电容, $\pm 20\%$ , X7R, 25 V	1	1	1	1	1	uF
C12	电源滤波电容, $\pm 20\%$ , X7R, 25 V	1	1	1	1	1	uF
C13	电源滤波电容, $\pm 20\%$ , X7R, 25 V	1	1	1	1	1	uF
C14	晶振驱动电容, $\pm 0.25$ pF, NP0, 50 V	NC	NC	NC	NC	NC	pF
C15	晶振驱动电容, $\pm 0.25$ pF, NP0, 50 V	NC	NC	NC	NC	NC	pF

## 10、封装外形

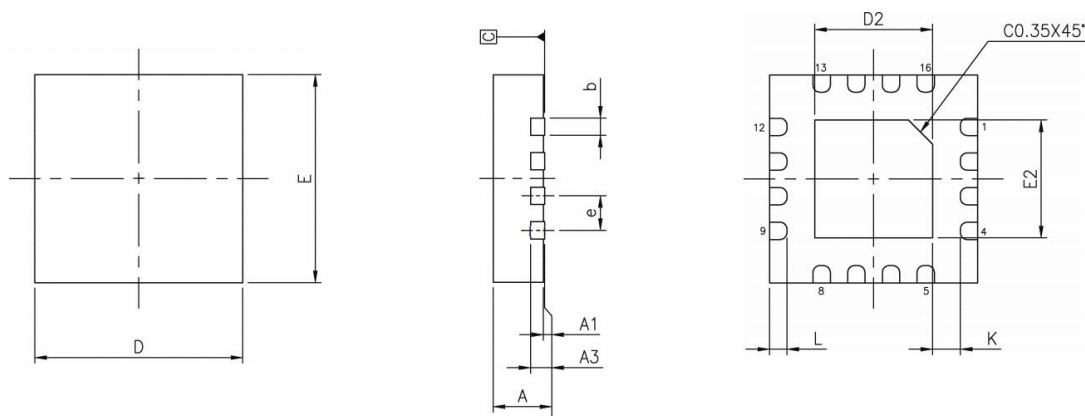


图 34 QFN16(3×3mm) 封装示意图

表 21 封装尺寸参数

符号	尺寸 (毫米 mm)		
	最小值	典型值	最大值
A	0.70	0.75	0.80
A1	0.00	0.02	0.05
A3	0.203 REF		
b	0.18	0.25	0.30
D	3.00 BSC		
E	3.00 BSC		
e	0.50 BSC		
D2	1.60	—	1.80
E2	1.60	—	1.80
L	0.20	—	0.40
K	0.20	—	—