

图 A：收发器框图

一般说明

SX1280 和 SX1281 收发器提供 2.4 GHz 频段的超远距离通信，线性度可承受严重干扰。这使它们成为稳健可靠的无线解决方案的理想解决方案。它们是第一款同类 ISM 频段收发器 IC，用于集成飞行时间功能，开放应用解决方案，以跟踪和定位工厂中的人员，宠物，无人机或物体。这些长距离 2.4 GHz 产品包括多个物理层和调制，以便为视频和安全应用以高数据速率优化远程通信。由于高集成度和超低电流消耗，可以轻松设计用于可穿戴设备的非常小的产品，允许使用微型电池。

该无线电完全符合所有全球 2.4 GHz 无线电规则，包括 EN 300 440，FCC CFR 47 Part 15 和日本 ARIB STD-T66。

远距离 2.4 GHz 产品系列中的集成度，低功耗和测距功能可实现增强的连接性，并为新一代以前未连接的设备和应用提供附加功能。

主要特点

- 长距离 2.4 GHz 收发器
- 高灵敏度，低至 -132 dBm
- +12.5 dBm，高效率 PA
- 低能耗，片上 DC-DC
- LoRa，FLRC，(G) FSK 支持的调制
- 可编程比特率
- 出色的阻断免疫力
- 测距发动机，飞行时间功能
- BLE PHY 层兼容性
- 系统成本低

应用

- 家庭自动化和电器

- 安全系统
- 跟踪应用程序
- 可穿戴设备和运动/健身传感器
- 无线电遥控玩具和无人机
- 智能手表和信标
- 医疗保健

订购信息

零件号	交货	订单数量
SX1280IMLTRT	卷带	3 千件

SX1281IMLTRT	卷带	3 千件
--------------	----	------

QFN 24 封装，温度范围为-40 至 85°C

无铅，无卤素，符合 RoHS / WEEE 标准的产品

修订记录

版	ECO	日期	变更和/或修改
1.0 版	035543	2017 年 2 月	首次发布
1.1 版	037029	2017 年 5 月	添加了 LoRa 调制解调器的有效数据速率表
			修正 LoRa 中的播出时间公式
			在主机控制器接口章节中更正拼写错误
			使用可选的 TCXO 更新应用原理图
			更新参考设计 BOM
			删除热阻抗一章中的冗余信息

目录

一般说明.....	1
主要特点.....	1
申请.....	1
订购信息.....	2
修订历史.....	2
数字清单.....	8
表格列表.....	9
1. 介绍.....	15
1.1 模拟前端.....	15
1.2 配电.....	15
1.3 调制解调器.....	16
1.4 包处理.....	16
1.5 数字接口和控制.....	16
2. 引脚连接.....	17
2.1 收发器引脚.....	17
2.2 包视图.....	18
3. 规格.....	18
3.1 ESD 通知.....	19
3.2 绝对最小和最大额定值.....	19
3.3 工作范围.....	19
3.4 一般电气规范.....	19
3.5 接收器电气规范.....	20
3.5.1 接收器规格.....	21
3.5.2 LoRa 调制解调器.....	22
3.5.3 FLRC 调制解调器.....	23
3.5.4 FSK 调制解调器.....	24
3.6 变送器电气规范.....	25
3.7 晶体振荡器规格.....	26
3.8 数字引脚电平.....	26
4. 模拟前端.....	27
4.1 发射器.....	27
4.2 接收器.....	28

4.2.1 低功耗模式和高灵敏度模式.....	29
4.3 PLL	29
4.4 RC 振荡器.....	30
5. 配电.....	30
5.1 选择 DC-DC 转换器或 LDO 调节.....	30
5.2 灵活的 DIO 供应.....	31
6. 数字基带.....	32
6.1 概述.....	32
6.2 LoRa 调制解调器.....	32
6.2.1 LoRa 调制.....	33
6.2.2 传播因素.....	33
6.2.3 带宽.....	34
6.2.4 前向纠错编码率.....	34
6.2.5 测距发动机.....	35
6.3 FLRC 调制解调器.....	35
6.3.1 调制解调器带宽和数据速率.....	35
6.3.2 FEC 编码率.....	35
6.3.3 高斯滤波.....	36
6.4 FSK 调制解调器.....	37
6.4.1 调制解调器带宽和数据速率.....	37
6.4.2 调制解调器调制指数.....	39
6.5 调制解调器选择指南.....	39
7. 数据包引擎.....	41
7.1 GFSK 包.....	42
7.1.1 固定长度数据包.....	42
7.1.2 可变长度包.....	42
7.2 BLE 包格式.....	43
7.3 FLRC 包.....	44
7.3.1 FLRC 包格式.....	44
7.3.2 固定长度数据包格式.....	44
7.3.3 可变长度数据包格式.....	45
7.3.4 FLRC 播出时间.....	45
7.4 LoRa 包.....	46
7.4.1 LoRa 包格式.....	46

7.4.2 显式（可变长度）标头模式.....	46
7.4.3 隐式（固定长度）标头模式.....	47
7.4.4 LoRa 播出时间.....	47
7.5 LoRa 测距发动机包.....	49
7.5.1 测距数据包格式.....	49
7.5.2 测量主交换.....	50
7.5.3 测量从属交换.....	50
8 数据缓冲区.....	50
8.1 操作原理.....	51
8.2 接收操作.....	51
8.3 传输操作.....	52
8.4 使用数据缓冲区.....	52
9 数字接口和控制.....	53
9.1 BUSY 引脚通信.....	53
9.2 接口检测.....	53
9.3 SPI 接口.....	54
9.3.1 收发器处于活动模式时的 SPI 时序.....	54
9.3.2 收发器离开休眠模式时的 SPI 时序.....	55
9.3.3 SPI 时序.....	55
9.4 UART 接口.....	57
9.5 引脚共享.....	57
9.6 多功能数字输入/输出（DIO）.....	57
10 运作模式.....	58
10.1 启动.....	58
10.2 睡眠模式.....	58
10.3 待机模式.....	59
10.4 频率合成（FS）模式.....	59
10.5 接收（Rx）模式.....	59
10.6 发送（Tx）模式.....	59
10.7 收发器电路模式图形说明.....	60
10.8 活动模式切换时间.....	61
11 主机控制器接口.....	62
11.1 命令结构.....	62
11.2 GetStatus 命令.....	63

11.3 注册访问操作.....	64
11.3.1 WriteRegister 命令.....	64
11.3.2 ReadRegister 命令.....	64
11.4 数据缓冲操作.....	66
11.4.1 WriteBuffer 命令.....	66
11.4.2 ReadBuffer	66
11.5 无线电操作模式.....	67
11.5.1 SetSleep	67
11.5.2 SetStandby	67
11.5.3 SetFs	68
11.5.4 SetTx	68
11.5.5 SetRx	69
11.5.6 SetRxDutyCycle	70
11.5.7 SetLongPreamble	71
11.5.8 SetCAD	72
11.5.9 SetTxContinuousWave	72
11.5.10 SetTxContinuousPreamble	72
11.5.11 SetAutoTx	73
11.5.12 SetAutoFs	73
11.6 无线电配置.....	74
11.6.1 SetPacketType	74
11.6.2 GetPacketType	74
11.6.3 SetRfFrequency	75
11.6.4 SetTxParams	75
11.6.5 SetCadParams	76
11.6.6 SetBufferBaseAddress	77
11.6.7 SetModulationParams	77
11.6.8 SetPacketParams	78
11.7 通信状态信息.....	80
11.7.1 GetRxBufferStatus	80
11.7.2 GetPacketStatus	80
11.7.3 GetRssiInst	82
11.8 IRQ 处理.....	83
11.8.1 SetDiolrqParams	84
11.8.2 GetIrqStatus	84
11.8.3 ClearIrqStatus	84
12.命令列表.....	85
13.收发器操作.....	87
13.1 GFSK 包.....	87
13.1.1 通用收发器设置.....	87
13.1.2 Tx 设置和操作.....	92

13.1.3 Rx 设置和操作.....	93
13.2 BLE 包.....	96
13.2.1 通用收发器设置.....	96
13.2.2 Tx 设置和操作.....	100
13.2.3 Rx 设置和操作.....	101
13.2.4 BLE 特定功能.....	103
13.3 FLRC 包.....	103
13.3.1 通用收发器设置.....	103
13.3.2 Tx 设置和操作.....	108
13.3.3 Rx 设置和操作.....	109
13.4 LoRa 包.....	112
13.4.1 LoRa 的常见收发器设置.....	112
13.4.2 Tx 设置和操作.....	115
13.4.3 Rx 设置和操作.....	116
13.5 测距设置.....	118
13.5.1 测距设备设置.....	118
13.5.2 作为状态机的测距操作.....	121
13.6 其他功能.....	122
13.6.1 SetRegulatorMode 命令.....	122
13.6.2 上下文保存.....	123
14.参考设计和应用原理图.....	124
14.1 参考设计.....	124
14.1.1 应用设计原理图.....	124
14.1.2 参考设计 BOM	124
14.1.3 参考设计 PCB	125
14.2 带可选 TCXO 的应用设计.....	125
14.3 低压降稳压器的应用设计.....	126
14.4 睡眠模式消耗.....	126
15.包装信息.....	128
15.1 包装外形图.....	128
15.2 土地格局.....	129
15.3 回流曲线.....	129
15.4 热阻.....	129
词汇表.....	130

数字清单

图 2-1：收发器引脚位置.....	
15 图 4-1：收发器框图，模拟前端突出显示.....	24
图 5-1：收发器方框图，突出显示的功率.....	
图 5-2：单独的 DIO 电源.....	28
图 6-1：收发器方框图，调制解调器突出显示.....	29
图 6-2：FSK 调制参数.....	35
图 6-3：收发器调制解调器的灵敏度性能.....	37
图 7-1：收发器框图，数据包引擎突出显示.....	38
图 7-2：固定长度数据包格式.....	39
图 7-3：可变长度数据包格式.....	40
图 7-4：BLE 包格式.....	40
图 7-5：PDU 标头格式.....	
图 7-6：FLRC 固定长度数据包格式.....	
图 7-7：FLRC 可变长度数据包格式.....	
图 7-6：FLRC 固定长度数据包格式.....	40
图 7-7：FLRC 可变长度数据包格式.....	41
图 7-6：FLRC 固定长度数据包格式.....	40
图 7-7：FLRC 可变长度数据包格式.....	42
图 7-8：LoRa 可变长度数据包格式.....	43
图 7-9：LoRa 固定长度数据包格式.....	
图 7-10：测距包格式.....	
图 7-11：测距主数据包交换.....	图 7-12：测距从属数据包交换.....
图 8-1：数据缓冲图.....	48
图 9-1：收发器方框图，数字接口突出显示.....	50
图 9-2：SPI 时序图.....	
图 9-3：SPI 时序转换.....	
图 10-1：收发器电路模式.....	57
图 10-2：活动模式下的切换时间定义.....	58
图 13-1：.....	

测距状态机..... 120 图 14-1：
收发器应用设计原理图..... 122 图 14-2：远程参考设计 PCB 布局..... 图 14-3：带有可选 TCXO 的应用原理图..... 124 图 14-4：具有低压差稳压器原理图的应用原理图..... 124
图 15-1：QFN 4x4 封装外形图..... 126 图 15-2：QFN 4x4mm 焊盘图案..... 127

表格列表

表 1-1：产品组合和调制解调器功能..... 12 表 2-1：收发器引脚说明.....
14 表 3-1：最低和最高额定值..... 16 表 3-2：工作范围..... 16 表 3-3：一般电气规范..... 17 表 3-4：接收器规格..... 18 表 3-5：LoRa 调制解调器规格..... 19
表 3-6：FLRC 调制解调器规格..... 20 表 3-7：FSK 调制解调器规格..... 21 表 3-8：变送器电气规格..... 22 表 3-9：晶体振荡器规格..... 22 表 3-10：数字水平和时间..... 23 表 4-1：接收机增益手动设置程序..... 25 表 4-2：接收器增益手动设置..... 表 5-1：调节类型与电路模式..... 27 表 6-1：使用 LoRa 时的接收器灵敏度..... 30 表 6-2：使用 LoRa 时的有效数据速率..... 表 6-3：有效的 FLRC 数据速率和带宽组合..... 32 表 6-4：基于 FEC 使用的有效 FLRC 数据速率以及产生的灵敏度 33 调节类型与电路模式..... 表 6-1：使用 LoRa 时的接收灵敏度。..... 表 6-2：使用 LoRa 时的有效数据速率..... 31 表 6-3：有效的 FLRC 数据速率和带宽组合..... 32 表 6-4：基于有效的 FLRC 数据速率 FEC 使用后产生的敏感性 33 调节类型与电路模式..... 表 6-1：使用 LoRa 时的接收灵敏

19: SetFs 数据传输.....	
65 表 11-20 : SetTx SPI 数 据 传 输.....	65 表 11-21: SetTx
UART 数 据 传 输	
65.....	65 表 11-20: SetTx
SPI 数据传输.....	65 表 11-21:
SetTx UART 数 据 传 输	
65.....	65 表 11-20: SetTx
SPI 数据传输.....	65 表 11-21:
SetTx UART 数据传输.....	65
表 11-22: SetTx 超时定义。	
66 表 11-23 : SetTx 超 时 持 续 时 间.....	66 表 11-24: SetRx SPI
数据传输.....	66 表 11-25:
SetRx UART 数据传输.....	66
表 11-26 : SetRx 超 时 持 续 时 间.....	67 表 11-27: 占空比操作
SPI 数据传输.....	67 表 11-28: 责任循环操作 UART
数 据 传 输	67 表 11-29: Rx 持续时间定
义	68 表 11-30:
SetLongPreamble 数据传输.....	68 表 11-
31: SetCAD 数据传输.....	
69 表 11-32 : SetTxContinuousWave 数 据 传 输.....	69 表 11-33: SetTxContinuousPreamble
数据传输.....	69 表 11-34: SetAutoTx SPI 数据传
输	表 11-35: SetAutoTx UART 数据传
输	表 11-36: SetAutoFs SPI 数 据 传
输.....	70 表 11-37: SetAutoFs UART
数 据 传 输	70 表 11-38:
SetPacketType SPI 数据传输.....	71 表 11-
39: SetPacketType UART 数据传输.....	71 表
11-40: PacketType 定义.....	
71 表 11-41: GetPacketType SPI 数据传输.....	表
11-42: GetPacketType UART 数据传输.....	表
11-43: SetRfFrequency SPI 数据传输.....	
72 表 11-44 : SetRfFrequency UART 数 据 传 输.....	72 表 11-45: SetTxParams SPI 数据传

输.....	72 表 11-46: SetTxParams UART
数据传输.....	73 表 11-47: 斜坡时间定
义.....	73 表 11-48: CAD SPI
数据传输.....	73 表 11-49:
CAD UART 数据传输.....	73 表
11-50: CadSymbolNum 定义	
74 表 11-51: SetBufferBaseAddress SPI 数 据 传	
输	74 表 11-52: SetBufferBaseAddress UART
数据传输.....	74 表 11-53: SetModulationParams SPI
数据传输.....	74 表 11-54: SetModulationParams
UART 数 据 传 输	74 表 11-55:
SetModulationParams 参 数 定 义	表 11-56:
SetPacketParams SPI 数据传输.....	75 表 11-
57: SetPacketParams UART 数据传输.....	75 表
11-58: SetPacketParams 参数定义.....	76 表 11-59:
GetRxBufferStatus SPI 数据传输.....	77 表 11-
60: GetRxBufferStatus UART 数据传输.....	77 表
11-61: GetPacketStatus SPI 数据传输	
77 表 11-62: GetPacketStatus UART 数 据 传	
输	78 表 11-63: packetStatus 定
义	78 表 11-64: RSSI 和
SNR 数据包状态.....	78 表 11-65:
状态包状态字节.....	79 表 11-66:
错误包状态字节.....	79
表 11-67: 同步数据包状态字节.....	表
11-68: GetRssiInst SPI 数据传输.....	
80 表 11-69: GetRssiInst UART 数 据 传	
输	80 表 11-70: RssiInst 定
义.....	80 表 11-71: IRQ
注册... ..	80 表
11-72: IRQ 掩码定义 SPI 数据传输.....	81 表
11-73: IRQ 掩码定义 UART 数据传输.....	81 表 11-
74: GetIrqStatus SPI 数据传输.....	82
表 11-75: GetIrqStatus UART 数 据 传	
输	82 表 11-76: ClearIrqStatus SPI 数
据传输.....	82 表 11-77: ClearIrqStatus
UART 数 据 传 输	

82.....	82 表 11-75 : GetIrqStatus UART 数据传
输.....	82 表 11-76 : ClearIrqStatus SPI 数
据传输.....	82 表 11-77 : ClearIrqStatus
UART 数 据 传 输
82.....	82 表 11-75 : GetIrqStatus UART 数据传
输.....	82 表 11-76 : ClearIrqStatus SPI 数
据传输.....	82 表 11-77 : ClearIrqStatus
UART 数据传输.....	82ClearIrqStatus UART
数据传输.....	82ClearIrqStatus UART 数据传
输.....	82
表 12-1: 收发器可用命令.....	83 表 13-1 :
GFSK 模式下的调制参数.....	85
表 13-2: GFSK 模式下的调制参数.....	86
表 13-3: GFSK 模式下的调制参数.....	表 13-4: GFSK 数据包中的前导
码长度定义。	87 表 13-5: GFSK 数据包中的同步
字长定义.....	88 表 13-6: 在 GFSK 数据包中同步字组
合.....	88 表 13-7: GFSK 数据包中的数据包类型
定义.....	88 表 13-8: GFSK 数据包中的有效载
荷长度定义.....	表 13-9: GFSK 数据包中的 CRC 定
义.....	89 表 13-10: GFSK 数据包中
的白化启用.....	89
表 13-11: GFSK 数据包中的同步字定义.....	89
表 13-12: CRC 初始化寄存器..
90 表 13-13: CRC 多项式定义.....
表 13-14 : GFSK 数 据 包 中 的 PacketStatus	
[3]	91 表 13-15: GFSK 数据包中的
PacketStatus [2]	93 表 13-16: GFSK
模式包中的 PacketStatus [4]	93 表 13-17:
BLE 和 GFSK 模式下的调制参数.....	94
表 13-18: BLE 和 GFSK 模式下的调制参数.....	95
表 13-19: BLE 和 GFSK 模式下的调制参数.....	95 表 13-20:
BLE 包中的连接状态定义.....	表 13-21: BLE 包中的
CRC 定义.....	表 13-22: BLE 数据包测试
模式下的 Tx 测试数据包有效负载.....	96 表 13-23: BLE 包中的
白化启用.....	97 表 13-24: BLE 包中的同
步字定义.....	97 表 13-25: CRC 初始化寄存

器.....	97	表 13-26： BLE 包中的
PacketStatus3	98	
表 13-27 : BLE 模式下的		
PacketStatus2	99	表 13-28：
BLE 模式下的 PacketStatus4	100	
表 13-29 : SetAutoTx 模		
式.....	101	
表 13-30： FLRC 模式下的调制参数：带宽和比特率.....	102	表 13-31： FLRC 模
式下的调制参数：编码率.....	103	
表 13-32： FLRC 模式下的调制参数： BT	103	表
13-33： FLRC 数据包中的 AGC 前导码长度定义.....	103	表 13-
34：在 FLRC 数据包中同步字长定义.....	104	表 13-35：
FLRC 数据包中的同步字组合	104	表 13-36：
FLRC 数据包中的数据包头类型定义.....	105	表 13-
37： FLRC 数据包中的有效载荷长度定义... ..	105	表 13-38： FLRC
数据包中的 CRC 定义.....	105	
表 13-39： CRC 初始化寄存器.....	105	
表 13-40： CRC 多项式定义.....	106	
表 13-41： FLRC 数据包中的白化定义.....	106	
表 13-42：在 FLRC 数据包中同步字定义.....	106	
表 13-43 : FLRC 数 据 包 中 的		
PacketStatus3	107	
表 13-44 : FLRC 数 据 包 中 的		
PacketStatus2	109	
表 13-45 : FLRC 数 据 包 中 的		
PacketStatus3	109	
表 13-46 : FLRC 数 据 包 中 的		
PacketStatus4	110	表 13-47： LoRa
模式下的调制参数.....	111	
表 13-48： LoRa 模式下的调制参数.....	112	
表 13-49： LoRa 模式下的调制参数.....	112	表 13-
50： LoRa 或 Ranging 中的前导码定义.....	113	表 13-51： LoRa 或
测距数据包中的数据包头类型定义.....	113	表 13-52： LoRa 数据包中的有效载荷长度
定义	113	表 13-53： LoRa 数据包中的 CRC 启
用.....	113	表 13-54： LoRa 或测距数
据包中的 IQ 交换.....	113	表 13-55：测距设备调制
参 数	117	表 13-56：从属测距请求地址定
义	117	表 13-57：寄存器地址位定

义..... 117 表 13-58：主测距请求地址定义。

..... 117 表 13-59：寄存器中的校准值

..... 表 13-60：测量角色值..... 118 表 13-61：注册结果地址..... 119 表 13-62：测距结果类型选择..... 119 表 13-63：功率调节选择 SPI 数据传输.....表 13-64：功率调节选择 UART 数据传输

..... 121 表 13-65：RegModeParam 定义

..... 121 表 13-66：SetSaveContext 数据传输..... 121121 表 13-64：功率调节选择 UART 数据传输..... 121 表 13-65：RegModeParam 定义..... 121 表 13-66：SetSaveContext 数据传输

..... 121121 表 13-64：功率调节选择 UART 数据传输..... 121 表 13-65：RegModeParam 定义

..... 121 表 13-66：SetSaveContext 数据传输..... 121

表 14-1：参考设计 BOM表 14-2：最小化睡眠模式消耗的主机设置..... 125

1.简介

SX1280 和 SX1281 是半双工收发器，能够在全局 2.4 GHz ISM 频段内实现低功耗运行。收音机包括 5 个主要部分，将在以下章节中介绍。

1.1 模拟前端

该无线电具有高效率+12.5 dBm 发射器和高线性度接收链，均可通过公共天线端口引脚访问。RF 和基带（低 IF）之间的频率转换由以 52 MHz 晶振为参考的数字 PLL 控制。发送和接收链都通过数据转换器连接到随后的数字块。有关详细信息，请参见第 24 页的第 4 章“模拟前端”。

1.2 配电

提供两种形式的电压调节，可以是集成的低压输出（LDO）或高效降压（降压）直流到直流转换器。这允许设计者根据应用的设计优先级在高能效或无线电小型化之间进行选择。有关详细信息，请参见第 27 页的第 5 章“配电”。

1.3 调制解调器

LoRa 系列的三个调制解调器提供了一系列调制选项，每个调制解调器都有包含许多 MAC 层功能的数据包选项。有关每种调制格式的描述以及与该调制相关的性能优势，请参阅以下相应部分：

- LoRa 调制解调器和数据包：第 30 页第 6.2 节“LoRa 调制解调器”
- FLRC 调制解调器和数据包：第 32 页的第 6.3 节“FLRC 调制解调器”
- FSK 调制解调器和数据包：第 35 页上的第 6.4 节“FSK 调制解调器”

远程 2.4 GHz 产品系列还具有测距引擎，这是一种长距离测距功能，可以在一对 LoRa 无线电之间进行飞行时间测量。对于远程 2.4 GHz 产品系列中的每个部件号，每个调制解调器和测距引擎的可用性如下所示。**表 1-1：产品组合和调制解调器功能**

产品参考	SX1280	SX1281
洛拉	✓	✓
FLRC	✓	✓
GFSK	✓	✓
测距发动机	✓	

1.4 包处理

无线电可以在全自动模式下操作，其中可以在没有外部主微控制器干预的情况下执行用于发送或接收的分组的处理。有关更多详细信息，请参见第 38 页的第 7 章“数据包引擎”。

在发送和接收模式下，收发器的有效负载接口是本数据表第 48 页第 8 节“数据缓冲器”中描述的数据包数据缓冲器。

1.5 数字接口和控制

有关收发器的所有数字通信的规范和处理，请参见第 50 页的“数字接口和控制”。这包括 SPI 和 UART 接口的描述，可用于配置收发器以及用于向外部主机微控制器发送中断的数字输入/输出 (DIO)。

- 有关 SPI 接口的信息，请参见第 51 页上的第 9.3 节“SPI 接口”
- 关于 UART 接口，请参见第 54 页上的第 9.4 节“UART 接口”
- 对于 DIO，请参见第 54 页上的第 9.6 节“多功能数字输入/输出 (DIO)”

2.引脚连接

2.1 收发器引脚

表 2-1: 收发器引脚排列

销 数	销 名称	类型 (I =输入 O = Ouptut)	SPI 描述	UART 描述
0	GND	-	裸露的接地垫	
1	VR_PA	-	PA 的调节供应	
2	VBAT_IN	一世	监管供应投入。连接到引脚 12。	
3	NRESET	一世	复位信号，低电平有效，内部上拉电阻为 50kΩ	
4	XTA	-	参考振荡器连接或 TCXO 输入	
五	GND	-	地面	
6	XTB	-	参考振荡器连接	
7	忙	∅	收发器忙指示灯	
8	DIO1	I / O	可选的多功能数字 I / O.	
9	DIO2	I / O	可选的多功能数字 I / O.	
10	DIO3	I / O	可选的多功能数字 I / O.	
11	VBAT_IO	一世	供应数字 IO 接口 (1.8 V 至 3.7 V)。必须≤VBAT。	
12	DCC_FB	∅	来自内部稳压器的稳压输出电压	
13	GND	-	地面	
14	DCC_SW	∅	DC-DC 切换器输出	
15	VBAT	一世	供应 RFIC (1.8 V 至 3.7 V)。必须≥VBAT_IO。	
16	MISO_TX	∅	SPI 从机输出	UART 发送引脚
17	MOSI_RX	一世	SPI 从机输入	UART 接收引脚
18	SCK_RTSN	一世	SPI 时钟	UART 请求发送
19	NSS_CTS	一世	SPI Slave Select	UART 清除发送
20	GND	-	地面	
21	GND	-	地面	
22	RFIO	I / O	RF 发送输出和接收输入	
23	GND	-	地面	

24	GND	-	地面
----	-----	---	----

2.2 包视图

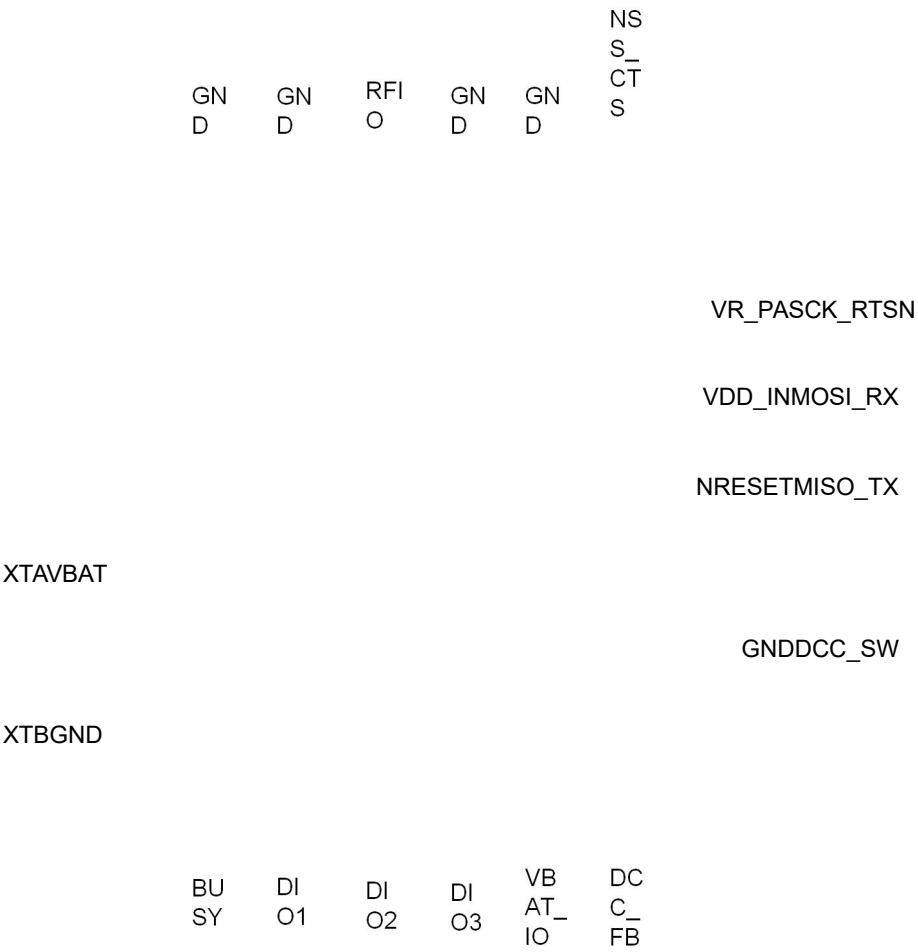


图 2-1: 收发器引脚位置

3.规格

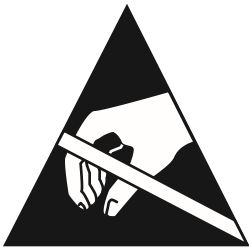
以下规格给出了 VBAT_IO = VBAT = 3.3 V，温度= 25°C，晶体振荡器频率= 52 MHz，RF 中心频率= 2.4 GHz 的典型工作条件。使用参考设计匹配所有 RF 阻抗，请参见第 122 页上的第 14.1 节“参考设计”。针对单音干扰器给出阻塞，ACR 和同信道抑制，并参考灵敏度等级+6dB。电流源是 VBAT 和 VBAT_IO 上的电流之和。除非另有说明，否则降压转换器（DC-DC）被认为是接通的。

3.1 ESD 通知

SX1280 / SX1281 收发器是高性能射频设备。

他们都满足：

- 所有引脚上的 JEDEC 标准 JESD22-A114（人体模型）的 2 级
- 所有引脚上的 JEDEC 标准 JESD22-C101（带电设备型号）的 III 类



3.2 绝对最小和最大额定值

表 3-1：最小和最大额定值

符号	描述	最低限度	典型	最大值	单元
VBATmr	VBAT 和 VBAT_IO 上的电源电压	-0.5	-	3.9	V
TMR	温度	-55	-	115	°C 下
PMR	射频输入电平	-	-	10	dBm 的

3.3 工作范围

表 3-2：工作范围

符号	描述	最低限度	典型	最大值	单元
VBATop	供电电压 VBAT 和 VBAT_IO	1.8	-	3.7	V
最佳	温度偏低	-40	-	85	°C 下
哒	在数字端口上加载电容	-	-	10	pF 的
ML	射频输入功率	-	-	10	dBm 的

3.4 通用电气规范

表 3-3：一般电气规范

符号	描述	最低限度	典型	最大值	单元
IDDSL	在保持数据缓冲区的休眠模式下提供电流	-	0.215	1.0	μA

	在休眠模式下提供电流，保留寄存器内容（保存上下文），不保留数据缓冲区	-	0.25	1.0	μA
	在保持指令 RAM 的休眠模式下提供电流	-	0.4	1.0	μA
	在休眠模式下提供电流，数据缓冲区，指令 RAM，数据 RAM 保留。RC 正在运行	-	1.2	1.8	μA
IDDSTDBYRC	在 STDBY_RC 模式下提供电流	-	760	-	μA
IDDSTDBYXOSC	在 STDBY_XOSC 模式下提供电流	-	1.2	-	嘛
IDDFS	FS 模式下的电源电流	-	2.8	-	嘛
FR	合成器频率范围	2400	-	2500	兆赫
fSTEP 的	合成器频率步长 (52 MHz 参考)	-	198	-	赫兹
2.45 GHz 的相位噪声					
PHN	1 MHz 偏移	-	-117	-	dBc 的/赫兹
	10 MHz 偏移	-	-133	-	dBc 的/赫兹
FXOSC	晶振频率	-	52	-	兆赫
TS_FS	启用 xosc 的频率合成器唤醒时间	-	54	-	微秒
TS_HOP	频率合成器跳跃时间在目标频率的 10 kHz 范围内				
	1 MHz	-	20	-	微秒
	10 MHz	-	三十	-	微秒
	100 MHz	-	50	-	微秒
TS_OS	晶振从 STDBY_RC 模式唤醒时间	-	100	-	微秒

有关数字规格，请参见 第 58 页的表 10-2：“所有可能转换的切换时间（TswMode）”。

3.5 接收器电气规范

对于具有 10 个字节有效载荷的分组，所有接收器灵敏度编号都是针对 1% 的打包器错误率（PER）给出的。

给出最大 AGC 增益的值，这是最高的低功率增益。

除非另有说明，否则连续波（CW）干扰源用于所有阻塞和拒绝测量。

3.5.1 接收器规格

表 3-4： 接收器规格

符号	描述	最小值	典型	最大值	单元
IIP3	3 阶输入截取，用于最大低功耗增益设置				
	带内干扰<6 MHz	-	-25	-	dBm 的
	带内干扰@ 6 MHz	-	-6	-	dBm 的
	带内干扰@ 10 MHz	-	-4	-	dBm 的
	带内干扰@ 20 MHz	-	-4	-	dBm 的
IMR	镜头拒绝 (CW 音调 1%PER)	-	三十	-	D b

3.5.2 LoRa 调制解调器

表 3-5: LoRa 调制解调器规格

符号	描述	最低限度	典型	最大值	单元
IDDRXLP_L	低功耗模式下的电源电流				
	对于 BW = 203 kHz	-	5.5	-	嘛
	对于 BW = 406 kHz	-	6	-	嘛
	对于 BW = 812 kHz	-	7	-	嘛
	对于 BW = 1625 kHz	-	7.5	-	嘛
IDDRXHS_L	高灵敏度模式的电源电流				
	对于 BW = 203 kHz	-	6.2	-	嘛
	对于 BW = 406 kHz	-	6.7	-	嘛
	对于 BW = 812 kHz	-	7.7	-	嘛
	对于 BW = 1625 kHz	-	8.2	-	嘛
RB_L	LoRa 比特率可编程范围, CR = 4/5				
	SF5, BW = 1625 kHz	-	202	-	kb / s 的
	SF6, BW = 1625 kHz	-	122	-	kb / s 的
	SF7, BW = 1625 kHz	-	71	-	kb / s 的
	SF12, BW = 203 kHz	-	0.476	-	kb / s 的
BW_L	LoRa 带宽可编程范围	203	-	1625	千赫
RFSLP_L	LoRa 接收器灵敏度, CR = 4/5, 低功耗模式使能 ¹				
	SF7, BW = 1625 kHz,	-	-106	-	dBm 的
	SF12, BW = 203 kHz	-	-130	-	dBm 的
RFSHS_L	LoRa 接收器灵敏度, CR = 4/5, 高灵敏度模式启用 ¹				
	SF7, BW = 1625 kHz,	-	-109	-	dBm 的
	SF12, BW = 203 kHz	-	-132	-	dBm 的
CCR_L	同信道拒绝 LoRa				
	SF7	-	7.5	-	D b
	SF12	-	19.5	-	D b
BI_L	阻断 SF12 的免疫力				

	+/- 1 MHz	-	60	-	D b
	+/- 2 MHz	-	63	-	D b
	+/- 10 MHz	-	81	-	D b

表 3-5: LoRa 调制解调器规格

符号	描述	最低限度	典型	最大值	单元
ACR_L	相邻信道抑制在 1.5 BW 的 CW				
	SF = 12, BW = 203 kHz	-	37	-	D b
	SF = 7, BW = 1.6 MHz	-	37	-	D b

1.请参见第 26 页上的第 4.2.1 节“低功耗模式和高灵敏度模式”。

3.5.3 FLRC 调制解调器

表 3-6: FLRC 调制解调器规范

符号	描述	最低限度	典型	最大值	单元
IDDRX_FL	供电电流				
	BW = 300kHz, BR = 260kb / s	-	6.5	-	嘛
	BW = 1200kHz, BR = 1300kb / s	-	8.6	-	嘛
RB_FL	FLRC 调制解调器可编程比特率	260	-	1300	kb / s 的
BW_FL	可编程通道带宽范围	300	-	2400	千赫
RFS_FL	FLRC 接收器灵敏度				
	260 kSymb / s, 130 kb / s BW = 300 kHz CR = 1/2	-	-106	-	dBm 的
	2.6 MSymb / s, 1.3 Mb / s, BW = 2.4 MHz, CR = 1/2	-	-97	-	dBm 的
CCR_FL	同信道拒绝 FLRC	-	-10	-	D b
BI_FL	最大低功率增益设置的阻塞级别				
	+/- 1 MHz	-	41	-	D b
	+/- 2 MHz	-	44	-	D b
	+/- 10 MHz	-	62	-	D b
	+/- 20 MHz	-	69	-	D b
ACR_FL	在 1.5 邻信道抑制 BW 为 CW				
	260 kb / s, BW = 300 kHz	-	44	-	D b

1.3 Mb / s, BW = 2.4 MHz	-	49	-	D b
--------------------------	---	----	---	-----

注意：上表中列出的所有数据速率都是原始位。所有值均以 $\beta T = 0.5$ 给出。

3.5.4 FSK 调制解调器

表 3-7：FSK 调制解调器规范

符号	描述	最小	典型值	最大值	单元
供电电流为低功率模式，解调运行 ¹					
IDDRX_FSK_250_LP	BW = 300 kHz, BR = 250 kb / s	-	4.8	-	嘛
IDDRX_FSK_1000_LP	BW = 1200 kHz, BR = 1000 kb / s	-	5.3	-	嘛
IDDRX_FSK_2000_LP	BW = 2400 kHz, BR = 2000 kb / s	-	5.7	-	嘛
供电电流为高灵敏度模式，解调运行 ¹					
IDDRX_FSK_250_HS	BW = 300 kHz, BR = 250 kb / s	-	5.5	-	嘛
IDDRX_FSK_1000_HS	BW = 1200kHz, BR = 1000kb / s	-	6	-	嘛
IDDRX_FSK_2000_HS	BW = 2400kHz, BR = 2000kb / s	-	6.4	-	嘛
BR_FSK	FSK 调制解调器可编程比特率	125	-	2000	kb / s 的
BW_FSK	可编程通道带宽范围 DSB	300	-	2400	千赫
FSK 接收器灵敏度 BER 0.1%					
RFS_FSK1 低 功耗模式	250 kb / s, $\beta = 0.5$, BW = 300 kHz	-	-100	-	dBm 的
	1 Mb / s, $\beta = 0.5$, BW = 1200 kHz	-	-94	-	dBm 的
FSK 接收器灵敏度 BER 0.1%					
RFS_FSK1_HS 高灵敏度模式	250 kb / s, $\beta = 0.5$, BW = 300 kHz	-	-102	-	dBm 的
	1 Mb / s, $\beta = 0.5$, BW = 1200 kHz	-	-96	-	dBm 的
FSK 接收器灵敏度 PER 1%					
RFS_FSK2 低 功耗模式	250 kb / s, $\beta = 0.5$, BW = 300 kHz	-	-93	-	dBm 的
	1 Mb / s, $\beta = 0.5$, BW = 1200 kHz	-	-88	-	dBm 的
RFS_FSK2_HS	FSK 接收器灵敏度 PER 1%				

高灵敏度模式	250 kb / s, $\beta = 0.5$, BW = 300 kHz	-	-94	-	dBm 的
	1 Mb / s, $\beta = 0.5$, BW = 1200 kHz	-	-90	-	dBm 的
CCR_FSK	同频道拒绝	-	-10	-	D b
BI_FSK	最大低功率增益设置的阻塞级别, BR = 250 kb / s, BW = 300 kHz				
	+/- 1 MHz	-	41	-	D b
	+/- 2 MHz	-	44	-	D b
	+/- 10 MHz	-	62	-	D b
	+/- 20 MHz	-	69	-	D b

表 3-7: FSK 调制解调器规范

符号	描述	最小值	典型	最大值	单元
ACR_FSK	CW 的相邻信道抑制为 1.5 BW				
	BW = 300 kHz	-	34	-	D b
	BW = 1200 kHz	-	34	-	D b

1.请参见第 26 页上的第 4.2.1 节“低功耗模式和高灵敏度模式”。

注意：上表中列出的所有值均以调制指数 $\beta = 0.5$ 给出。

3.6 变送器电气规范

表 3-8: 变送器电气规范

符号	描述	最低限度	典型	最大值	单元
IDD_T13	12.5 dBm	-	24	-	嘛
IDD_T10	10 dBm	-	18	-	嘛
IDD_T0	0 dBm	-	10	-	嘛
RFOPMIN	最小射频输出功率	-	-18	-	dBm 的
RFOPMAX	最大射频输出功率	-	12.5	-	dBm 的
FDA	可编程 FSK 频率偏差	62.5	-	1000	千赫

3.7 晶体振荡器规格

表 3-9：晶体振荡器规范

符号	描述	最低限度	典型	最大值	单元
FXOSC	晶振频率	-	52	-	兆赫
CLOAD	晶体负载电容	-	10	-	pF 的
COXTAL	晶体分流电容	-	2	五	pF 的
RSXTAL	水晶系列电阻	-	10	50 ¹	Ω
CMXTAL	晶体运动电容	3	3.5 ²	4	FF

1. 如果 COXTAL 限制在<3 pF，则可以使用高达 90Ω的 RSXTAL。
2. 可以使用其他 CMXTAL 值，注意较小的值会减少启动时间，而较大的值会降低频率精度和相位噪声。

3.8 数字引脚电平

表 3-10：数字电平和时序

符号	描述	最低限度	典型	最大值	单元	条件
V IH	数字输入电平高	0.8	-	-	VBAT_IO	-
V IL	数字输入电平低	-	-	0.2	VBAT_IO	-
V OH	数字输出电平高	0.9	-	-	VBAT_IO	I max = 2.5 mA
V OL	数字输出电平低	-	-	0.1	VBAT_IO	I max = -2.5 mA
我漏了	数字输入漏电流 (NSS, MOSI, SCK)	-1	-	1	μA	-

4.模拟前端

模拟前端具有与集成匹配电路的单个天线端口连接，允许无线电的半双工操作，无需外部 RF 切换。

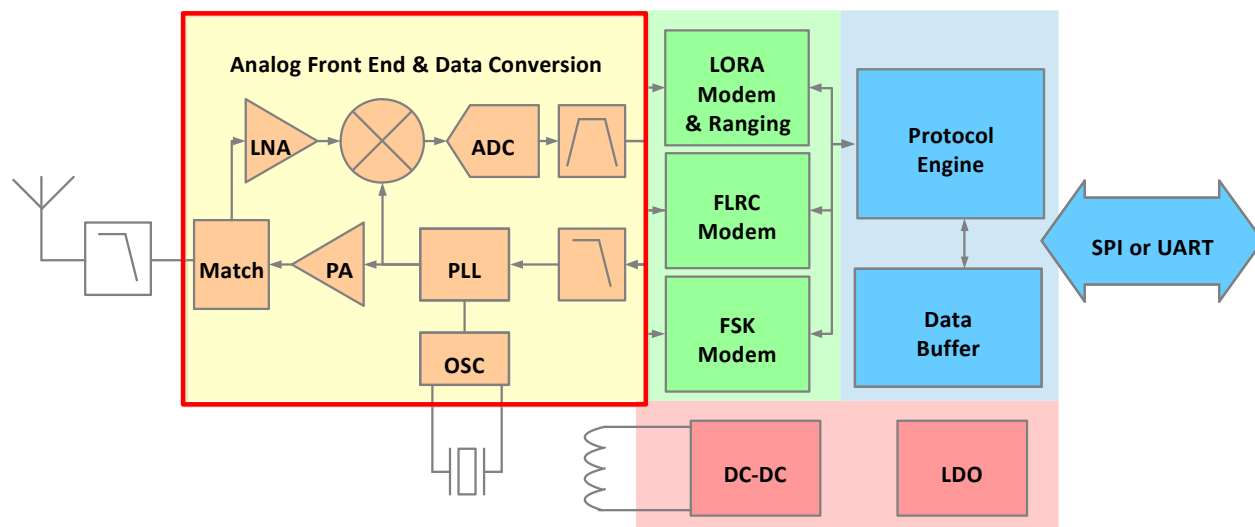


图 4-1：收发器框图，模拟前端突出显示

4.1 发射器

发送链包括来自调制解调器组的调制输出，其直接调制小数 N 分频 PLL。可以启用比特流的可选预滤波，以降低相邻信道中的功率，这也取决于所选择的调制类型。

使用 `SetTx (periodBase, periodBaseCount)` 命令启用发送器。发出此命令后，发送器发送存储在数据缓冲区中的数据包。然后，发射机返回到 STDBY_RC 模式，可以是分组发送结束时，或由中断定时器，的时间基准定义的预定义超时时段之后 `periodBase` 秒，和时钟的预设数量 `periodBaseCount` 如在第 12 章“第 83 页上的“命令列表”。

发射机的射频输出功率可在 -18 dBm 至 +12 dBm 的范围内以 1 dB 的增量进行控制，最终功率步长则为 0.5 dB 增量，最大发射机输出功率为 12.5 dBm。RF 输出功率（PRF）和斜坡时间由命令 `SetTxParam (功率, 斜坡时间)` 确定。输出功率使用以下公式设置：

$$P_{RF \text{ dBm 的}} = -18 + \text{功率}$$

最大输出功率 $P_{RF \text{ [dBm]}}$ 为 12.5 dBm。

这对应于参考设计的天线馈电点处的 RF 输出功率（请参见第 122 页的章节 14.1.1“应用设计原理图”）。RF 功率放大器的突然切换会引起不希望的杂散光谱发射。因此，精密 DAC 用作收发器 PA 电源的参考，允许平滑过渡到发送模

式。在数据包传输之前，PA 斜坡上升的时间相应地，*rampTime* 可以在 2 到 20μs 之间变化。在某些应用中并且出于监管测试目的，在发射模式下产生连续波（CW）音或启用连续调制输出可能是有用的。这两个功能可以通过 *SetTxContinuousWave()* 和 *SetTxContinuousPreamble()* 函数。后者使用任何配置的调制设置提供交替的逻辑“1”和“0”调制数据流。

4.2 接收者

LoRa，FLRC 或 FSK 系统作为半双工低 IF / 零 IF 收发器运行。接收的 RF 信号首先由 LNA 通过片上阻抗匹配网络放大。之后执行单端到差分转换以改善接收器的二阶线性度。然后通过正交混频器将信号下变频到基带或中频，以获得 I 和 Q 信号。然后对这些信号进行低通滤波并最终数字化。

接收链采用默认启用的自动增益控制（AGC），用于确保选择最佳前端增益以接收给定的检测信号功率。可以禁用此功能并手动设置 RF 前端的增益。为此，必须配置以下寄存器：

表 4-1：接收器增益手动设置的步骤

寄存器	位	值	评论
0x89F	第 7 位	1	启用手动增益控制
0x895	位 0	0	启用手动增益控制
0x89E	位 0: 3	1 至 13	手动增益设置（见下表）

然后可以根据下表中指示的设置设置增益：

表 4-2：接收器增益手动设置

设置	增益[dB]
13	马克斯
12	最大-2
11	最大-4
10	最大-6
9	最大-8
8	最大-12
7	最大-18
6	最大-24
五	最大-30
4	最大-36
3	最大-42

2	最大-48
1	最大-54

有关读取和写入控制寄存器的步骤，请参见第 83 页上的“命令列表”。

通过发出带有 *periodBase* 振荡器时基和 *periodBaseCount* 时钟周期数的 *SetRx (periodBase, periodBaseCount)* 命令来指定接收模式的超时（参见第 10.5 节“接收 (Rx) 模式”），进行到接收模式的转换。56）将退出到 STDBY_RC 模式。定时接收过程可以在收发器中完全自动化。有关每种调制格式的此过程和处理，请参见第 92 页上的第 13.1.3 节“Rx 设置和操作”。

当接收到信号或分组时，收发器使用接收信号强度指示器（RSSI）报告信号强度。使用 *GetPacketStatus ()* 请求返回此信息，如第 84 页上的第 12 章“命令列表”中所述。

4.2.1 低功耗模式和高灵敏度模式

在接收模式下，SX1280 可以在两种不同的操作方式之一下运行。低功耗模式可实现 SX1280 的最高效率，从而优化器件的性能，以实现接收器电流消耗。这是默认启用的，并限制接收器 LNA 的增益访问 LNA 增益的最高三个步骤。相反，高灵敏度模式可用于解锁最低噪声，最高灵敏度增益步骤，以略微增加接收器电流消耗。通过将地址 0x891 的位 7: 6 设置为 0x3，可以使能高灵敏度模式。一旦启用，接收器的噪声系数可提高多达 3 dB，以获得 500µA 的额外电流消耗。

4.3 PLL

分数 N 三阶Σ-Δ PLL 作为频率合成器的 LO（本地振荡器），用于接收器和发送链。该 PLL 能够快速自动校准与低开关时间。根据所选的调制类型，在 PLL 带宽内或外部自动执行调制。

该 PLL 频率从它使用外部 52 MHz 的晶体参考晶体振荡器电路的。PLL 和参考频率决定了无线电的 RF 中心频率。使用默认的晶体参考频率，FXosc 和 FRF 值，设置为 2.4 GHz。所有其他参考振荡器和 PLL 设置将根据所选的调制解调器设置自动优化。

要设置收发器的 RF 中心频率，请使用 *SetRFFrequency ()* 命令。频率作为 24 位操作数 *rfFrequency* 传递，如下所示：

$$F_{RF} = \frac{FX_{osc}}{182} \cdot rfFrequency$$

所述 PLL 可单独通过使用启用 *SetFS ()* 命令，该命令调谐 PLL 到发射频率。这是一种中间模式，可在从睡眠或待机模式转换为发送或接收模式时自动启用。

4.4 RC 振荡器

提供两个 RC 振荡器：64 kHz 和 13 MHz RC 振荡器。在休眠模式下，收发器可选择使用 64 kHz RC 振荡器来唤醒收发器以执行定期或占空比操作。所有 SPI 或 UART 通信均使能 13 MHz RC 振荡器，以便在不启动晶体振荡器的情况下配置器件。

两个振荡器的存在允许在休眠模式下实现超低功耗，仅运行 64 kHz 振荡器，而一旦启动通信，启动速度更快的 1.3 MHz 振荡器就可以实现与外部主处理器的高效高速通信。可选择使用晶体振荡器代替休眠模式以外的所有模式中的 RC 振荡器，如第 10 章“操作模式”（第 55 页）中所述。

5. 配电

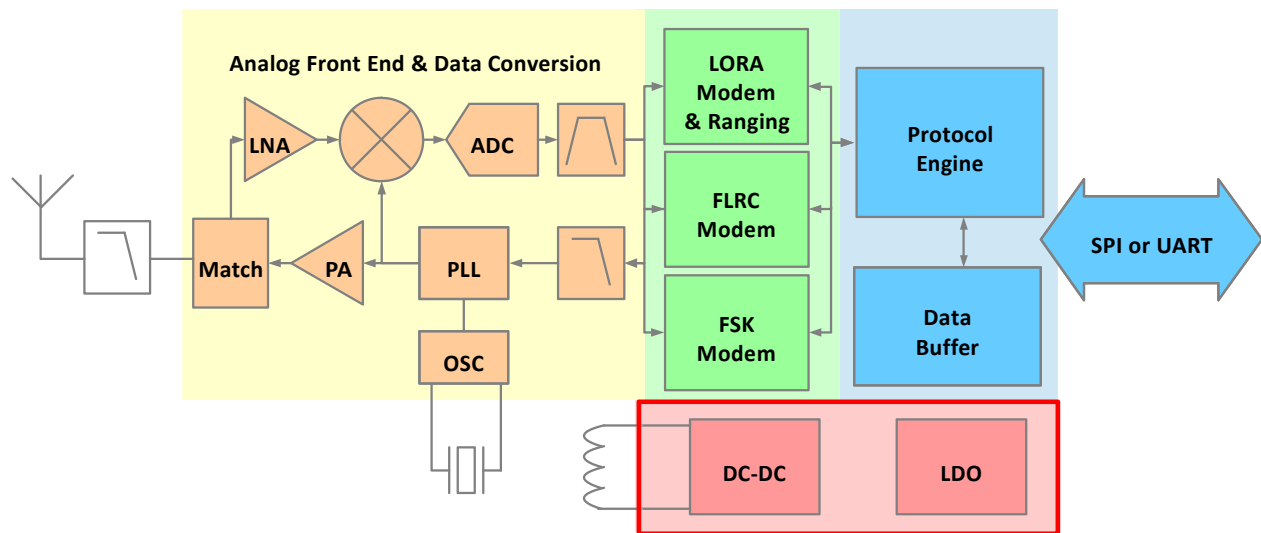


图 5-1：收发器框图，突出显示的功率分布

5.1 选择 DC-DC 转换器或 LDO 调节

根据应用的设计优先级，可提供两种形式的电压调节（DC-DC 降压转换器或线性稳压器）。默认情况下，线性 LDO 稳压器用于所有模式。或者，可以在 FS、Rx 和 Tx 模式下启用高效 DC-DC 降压转换器（DC-DC）。

收发器的所有规格均在启用 DC-DC 稳压器的情况下给出。对于成本和尺寸受限的应用，LDO-only 操作是可能的，这样就无需在引脚 12 和 14 之间使用 15μH 电感，从而具有以下优点：

- 减少物料清单
- 减少电路板空间

相反，无线电的能量消耗将增加。下表说明了不同模式和用户设置的功率调节选项。

表 5-1: 调节类型与电路模式

电路模式	睡觉	STDBY_RC	STDBY_XOSC	FS	RX	TX
调节器类型= 0	-	我愿意	我愿意	我愿意	我愿意	我愿意
调节器类型= 1	-	我愿意	DC-DC	DC-DC	DC-DC	DC-DC

用户可以使用命令 *SetRegulatorType (regulatorType)* 指定 DC-DC 的使用。此操作只能在 STDBY_RC 模式下执行。

5.2 灵活的 DIO 供应

收发器有两个电源引脚，一个用于收发器的核心，称为 VBAT，另一个用于主控制器接口 (SPI / UART , DIO , BUSY)，称为 VBAT_IO。两种电源都可以在应用中连接在一起。如果使用低压微控制器（通常使用 1.8 V 的 IO 焊盘）来控制收发器，用户可以：

- 使用 3.3V 的 VBAT 以获得最佳 RF 性能
- 将 VBAT_IO 直接连接到用于微控制器的相同电源•将包括 SPI 或 UART 的数字 IO 直接连接到微控制器 DIO。

在任何时候，VBAT_SX1280_DIO 必须低于或等于 VBAT。

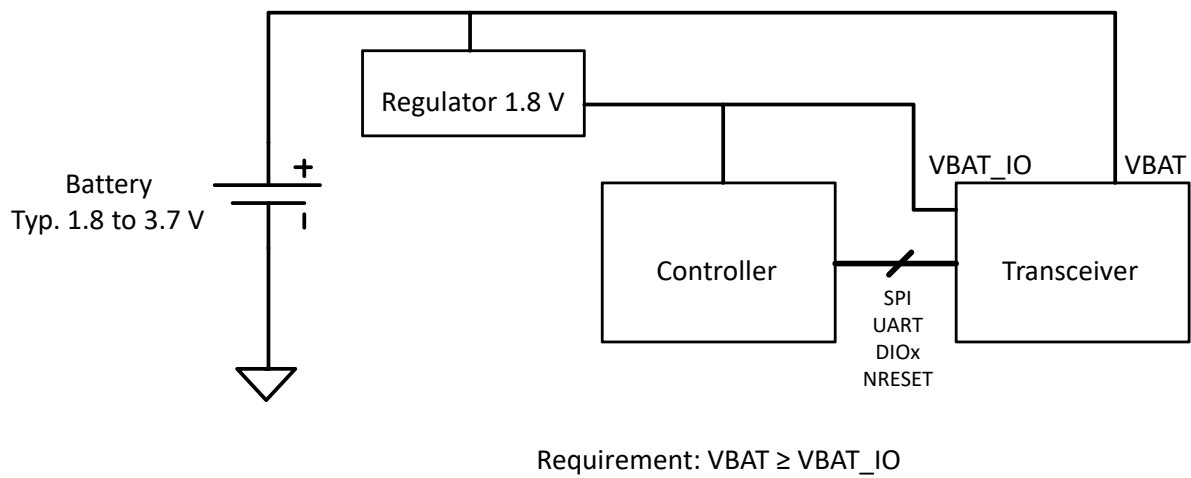


图 5-2: 单独的 DIO 电源

6. 数字基带

6.1 概述

收发器具有三个调制解调器，这三个调制解调器都在电路的数字基带部分中实现。对应于每个调制解调器的是独立的分组格式和分组配置选项。

所有调制解调器都使用数字自动频率校正 (AFC)。该过程完全自动化并且对用户透明。每个调制解调器的频率容差在相应的章节中详细说明。控制调制解调器配置的接口和存储分组的存储器对于所有调制解调器也是通用的，为调制和解调数据提供简单的统一接口。

每个调制解调器的可用调制解调器和相应的数据包类型显示在下面突出显示的块中：

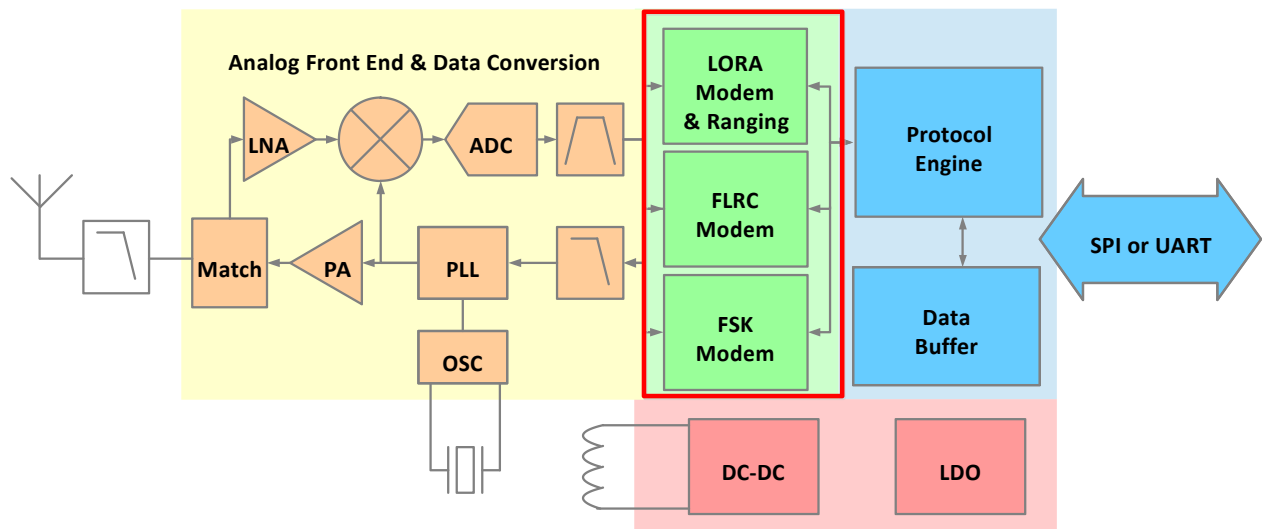


图 6-1：收发器框图，调制解调器突出显示

与可用的每个物理层调制相关联，还存在一系列相应的分组格式。

注意：因此，必须注意确保 仅在定义 要使用的数据包类型 `SetPacketType ()` 之后使用命令 `SetModulationParam ()` 设置调制参数。

6.2 LoRa 调制解调器

的洛拉调制解调器提供了基于两个长距离通信洛拉扩频调制，并采用了测距引擎，其提供该设施以测量时间飞行，从而提供推断出一对收发器之间的范围内的可能性。

6.2.1 LoRa 调制

的洛拉调制解调器使用扩频调制和前向纠错（FEC）技术来增加相对于传统的无线电通信链路的范围和健壮性 FSK 或 OOK 基于调制。

LoRa 调制解调器的一个重要方面是其出色的抗干扰性。它具有高达 19.5 dB 的同信道抑制能力。这种对干扰的抗扰性允许 LoRa 调制系统在重频谱使用的频带中或在使用 LoRa 的混合通信网络中共存，以在传统调制方案失败时扩展范围和鲁棒性。

当用于通信时，LoRa 数据包与此调制解调器兼容。有关此格式及其用法的完整详细信息，请参见第 111 页上的第 13.4 节“LoRa 数据包”。

6.2.2 传播因素

该 LORA 调制解调器使用基于线性调频扩频调制。对于任何扩频设备，LoRa 调制通过多个信息芯片表示有效载荷信息的每个符号。扩频因子（SF）确定符号率（Rs）和码片率（Rc）之间的比率：

$$SF = \frac{R_s}{R_c}$$

注意，由于不同的扩展因子彼此正交，因此必须在链路的发送侧和接收侧预先知道扩频因子。

下表显示使用 LoRa 调制解调器时的接收器灵敏度。接收器灵敏度给出：

- Packer 错误率（PER）为 1%，
- 包含 10 个字节的有效负载
- 25°C，3.3。V，CR = 4/5

表 6-1：使用 LoRa 时的接收器灵敏度

带宽 (kHz)	[Receiver 灵敏度 (dB 1)]							
	SF5	SF6	SF7	SF8	SF9	SF10	SF11	SF12
203	-109	-111	-115	-118	-121	-124	-127	-130
406	-107	-110	-113	-116	-119	-122	-125	-128
812	-105	-108	-112	-115	-117	-120	-123	-126
1625	-99	-103	-106	-109	-111	-114	-117	-120

下表显示了使用 LoRa 调制解调器时可以获得的有效数据速率：

表 6-2：使用 LoRa 时的有效数据速率

带宽 (kHz)	有效数据率 (kbps)							
	SF5	SF6	SF7	SF8	SF9	SF10	SF11	SF12
203	31.72	19.03	11.1	6.34	3.57	1.98	1.09	0.595
406	63.44	38.06	22.2	12.69	7.14	3.96	2.18	1.19
812	126.88	76.13	44.41	25.38	14.27	7.93	4.36	2.38
1625	253.91	152.34	88.87	50.78	28.56	15.87	8.73	4.76

6.2.3 带宽

在 LoRa 系统中，带宽设置设置双侧调制带宽，这相当于码片速率。信号带宽的增加相当于更高的有效数据速率。这意味着符号周期由下式给出：

$$T = \frac{2^{SF}}{BW}$$

符号周期是计算 LoRa 数据包播出时间的重要参数，如第 43 页第 7.4 节“LoRa 数据包”所示。通过设置扩频因子和 LoRa 调制的带宽来定义信号的灵敏度和空中时间之间的折衷。我们为 LoRa 调制解调器定义原始数据速率 R_b ，相当于：

$$R_b = \frac{2^{SF}}{T}$$

6.2.4 前向纠错编码率

为了进一步提高链路的鲁棒性，LoRa 调制解调器采用循环错误编码来执行前向错误检测和纠正。尽管前向纠错（FEC）在存在突发干扰的情况下不会显著提高调制解调器的灵敏度，但是在存在干扰的情况下提高链路的可靠性特别有效。可以响应于信道条件改变编码速率，从而改变干扰的鲁棒性。编码率可以可选地包括在分组报头中以供接收器使用。增加的空中时间开销与 FEC 的纠错能力成正比。由此产生的有效比特率，包括影响 FEC，可根据以下公式计算：

$$R = R_b \times \frac{4}{4 + CR}$$

其中 CR 是编程的编码率。LoRa 调制解调器允许的设置数据速率范围为 71 kb/s 至 202 kb/s，带宽为 1625 kHz，BW = 200 kHz 时低至 476 bps。

6.2.5 测距发动机

测距引擎使用 LoRa 调制解调器在一对收发器无线电之间执行飞行时间测量。有关测距功能操作的完整详细信息，请参见第 116 页上的第 13.5 节“测距设置”。飞行时间要求使用测距引擎数据包格式，如第 7.5 节“LoRa 测距引擎数据包”（第 46 页）中所述。

6.3 FLRC 调制解调器

快速远程通信（FLRC）调制解调器基于 GMSK 的相干解调，结合前向纠错和交织技术，以提高接收机灵敏度。用户可以访问这些参数，与相同数据速率下的 FSK 调制相比，允许高速通信，链路预算提高 8 到 10 dB。

与 FLRC 调制解调器一起使用的可用数据包类型是第 102 页第 13.3 节“FLRC 数据包”中描述的 FLRC 数据包

6.3.1 调制解调器带宽和数据速率

这些更高的数据速率范围从 260 kb / s 到 1.3 Mb / s。为了支持这些原始数据速率，支持 0.3 MHz 至 2.4 MHz 的调制带宽。请注意，并非所有带宽和数据速率组合都受支持。因此，使用 *SetModulationParam ()* 命令对原始数据速率进行编程，第一个参数选择原始数据速率和双边带调制带宽的有效组合之一。**表 6-3：有效的 FLRC 数据速率和带宽组合**

符号	原始比特率 RB [Mb / s 的]	带宽 BW [MHz DSB]
FLRC_BR_1_300_BW_1_2	1.3	1.2
FLRC_BR_1_040_BW_1_2	1.04	1.2
FLRC_BR_0_650_BW_0_6	0.65	0.6
FLRC_BR_0_520_BW_0_6	0.52	0.6
FLRC_BR_0_325_BW_0_3	0.325	0.3
FLRC_BR_0_260_BW_0_3	0.26	0.3

6.3.2 FEC 编码率

所述 FLRC 调制解调器可以任选地通过参数控制使用前向纠错 *codingRate* (CR)。应用于分组的卷积编码需要添加在纠错过程中使用的冗余信息。纠错使得分组有效载荷信息对来自相同频带或信道中的其他无线电服务的干扰突发具有鲁棒性。开销在下面表示为原始比特率和有效比特率的表，其考虑了 FEC 开销的影响。

表 6-4：基于 FEC 使用的有效 FLRC 数据速率和产生的灵敏度

符号	原始程序 数据速率 Rb [Mb / s]	程序 编码率 CR	有效 数据速率 Rbeff [Mb / s]	PER 1%时的灵敏度[dBm]
FLRC_BR_1_300_BW_1_2	1.3	1	1.3	-96
	1.3	3/4	0.975	-100
	1.3	1/2	0.65	-99
FLRC_BR_1_040_BW_1_2	1.04	1	1.04	-97
	1.04	3/4	0.78	-100
	1.04	1/2	0.52	-101
FLRC_BR_0_650_BW_0_6	0.65	1	0.65	-99
	0.65	3/4	0.488	-103
	0.65	1/2	0.325	-104
FLRC_BR_0_520_BW_0_6	0.52	1	0.52	-100
	0.52	3/4	0.39	-104
	0.52	1/2	0.26	-104
FLRC_BR_0_325_BW_0_3	0.325	1	0.325	-101
	0.325	3/4	0.195	-106
	0.325	1/2	0.130	-106
FLRC_BR_0_260_BW_0_3	0.26	1	0.26	-103
	0.26	3/4	0.195	-105
	0.26	1/2	0.130	-106

6.3.3 高斯滤波

在发射模式中，还可以使用由参数 BT 控制的可选高斯滤波器。该滤波功能用于减少发送的 FLRC 信号的旁瓣发射。滤波参数 BT 的有效值以减少滤波努力的顺序为：0.5,1 或 OFF。过滤器 BT 也由 SetModulationParam () 命令配置。

6.3.3.1 FLRC 频率容差。

调制解调器配置有通过第 84 页上的第 12 章“命令列表”中所述的 SetPacketParam () 和 SetModulationParam () 命令设置的数据速率参数。在接待处，接待过程分为三个阶段。第一个依赖于一群相关者都在寻找有效的传入前导码。运行的相关器数量是带宽和数据速率的函数，以确保对于 1.3 Mb / s 和 1.04 Mb / s 的数据速率，可以适应 +/- 30 ppm 的频率漂移。对于较低的数据速率，这降低到发射器和接收器之间频率未对准的 +/- 10ppm。

一旦检测到有效的前导码，调制解调器继续检查同步字以确保接收的分组用于该无线电。解调过程的最后阶段是分组数据本身的解调。

可接受的频率容差如下表所示。

表 6-5: FLRC 调制解调器的接收器性能

数据速率[Mb / s]	带宽[MHz]	频率容差[kHz]
1.3	1.2	+/- 150
1.04	1.2	+/- 150
0.65	0.6	+/- 150
0.52	0.6	+/- 150
0.325	0.3	+/- 75
0.260	0.3	+/- 75

6.4 FSK 调制解调器

的 FSK 调制解调器是常规的 FSK 调制解调器，其还提供可选的高斯滤波。使用此调制解调器，支持 FSK，GFSK，MSK 和 GMSK 调制格式。

该调制器还用于提供物理层与蓝牙低功耗的兼容性，因此两种帧类型与 FSK 调制解调器兼容：BLE 帧和 GFSK 帧，有关这些帧格式及其使用的更多信息，请参见第 13.2 节“BLE 包”第 94 页上的第 13.1 节和第 85 页上的第 13.1 节“GFSK 数据包”。

6.4.1 调制解调器带宽和数据速率

的 FSK 调制解调器能够 2-FSK 调制的在一定范围内从 125 千字节/秒至 2 Mb / s 的数据速率。数据速率由第 74 页上的第 11.6.7 节“SetModulationParams”中的 SetModulationParams () 命令控制。FSK 双边带 (DSB) 占用带宽与其他调制参数一起定义，如下图所示：

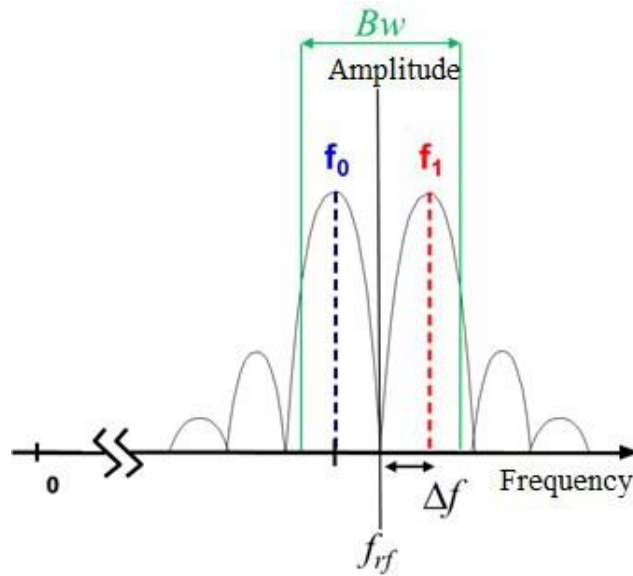


图 6-2: FSK 调制参数

其中 Δf 是频率偏差, f_{rf} 是 RF 中心频率。

在接收模式下, 带宽配置为最低接收器带宽, 可以容纳 FSK 信号的信号带宽定义为:

$$B = 2 \Delta f + R$$

- 20 dB

可提供 0.3 MHz 至 2.4 MHz 范围内的可编程带宽, 但并非所有原始数据速率和带宽组合均有效。有效组合的范围如下表所示。

表 6-6: 有效 FSK 数据速率和带宽组合以及产生的灵敏度

符号	原始比特率 R_b [kb / s]	带宽 BW [kHz DSB]	灵敏度[dBm]
FSK_BR_2_000_BW_2_4	2.0	2.4	-83
FSK_BR_1_600_BW_2_4	1.6	2.4	-84
FSK_BR_1_000_BW_2_4	1.0	2.4	-87
FSK_BR_1_000_BW_1_2	1.0	1.2	-88
FSK_BR_0_800_BW_2_4	0.8	2.4	-87
FSK_BR_0_800_BW_1_2	0.8	1.2	-89
FSK_BR_0_500_BW_1_2	0.5	1.2	-90
FSK_BR_0_500_BW_0_6	0.5	0.6	-89

FSK_BR_0_400_BW_1_2	0.4	1.2	-91
FSK_BR_0_400_BW_0_6	0.4	0.6	-90
FSK_BR_0_250_BW_0_6	0.25	0.6	-92
FSK_BR_0_250_BW_0_3	0.25	0.3	-93
FSK_BR_0_125_BW_0_3	0.125	0.3	-95

请注意，由于 FSK 调制解调器中没有纠错码，因此没有有效数据速率的概念。

6.4.2 调制解调器调制指数

除了原始比特率和带宽之外，设计人员还可以灵活地在 0.35 到 2 的范围内更改调制指数。调制指数 β 是描述“1”和“0”接近度的品质因数。给定数据速率的频率。这会影响解调器识别每个逻辑电平的容易程度，并由下式给出：

$$\beta = \frac{2 \Delta f}{R_b}$$

其中 Δf 是频率偏差， R_b 是编程数据速率。

6.5 调制解调器选择指南

收发器中三个调制解调器的相对接收性能如下图所示。蓝线表示在与用于测量调制解调器性能的设置相当的设置下的无差错通信的香农限制。在这里，我们看到传统的 FSK 调制解调器，如用于传统和蓝牙通信，产生 2.4 GHz 操作的传统灵敏度数字。

与此相反，由于使用扩频技术，LoRa 调制可以访问较低的有效数据速率。这显着提高了灵敏度，使其在理论极限的 10 到 11 dB 范围内。

基于相干 MSK 解调器的 FLRC 调制解调器提供对更有效数据速率的访问 - 相对于香农界限保持相同的灵敏度改进。因此，对于希望获得更长距离而不会受到更长时间广播惩罚的链路，FLRC 调制解调器提供了所需的设计灵活性。

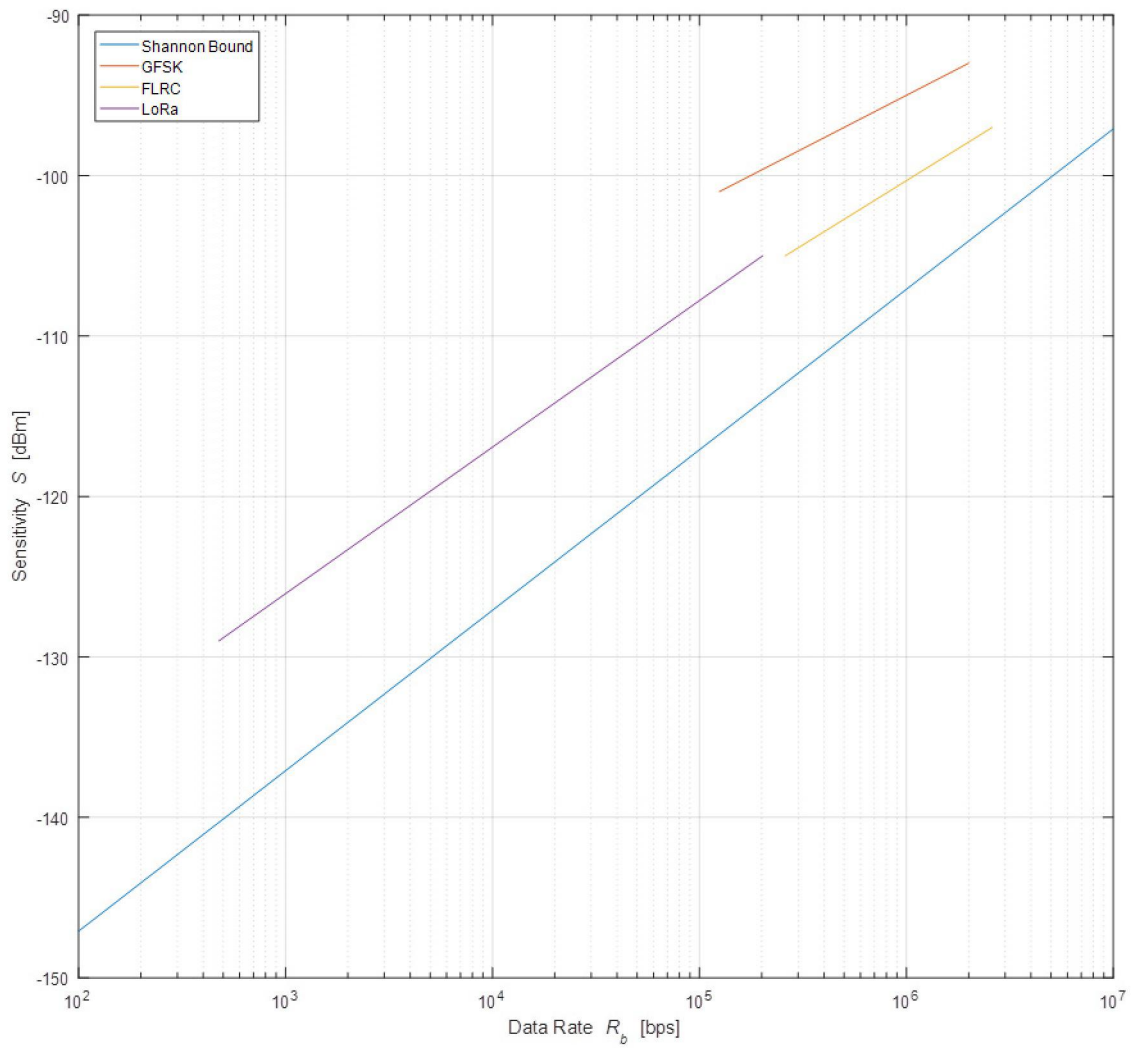


图 6-3: 收发器调制解调器的灵敏度性能

7.数据包引擎

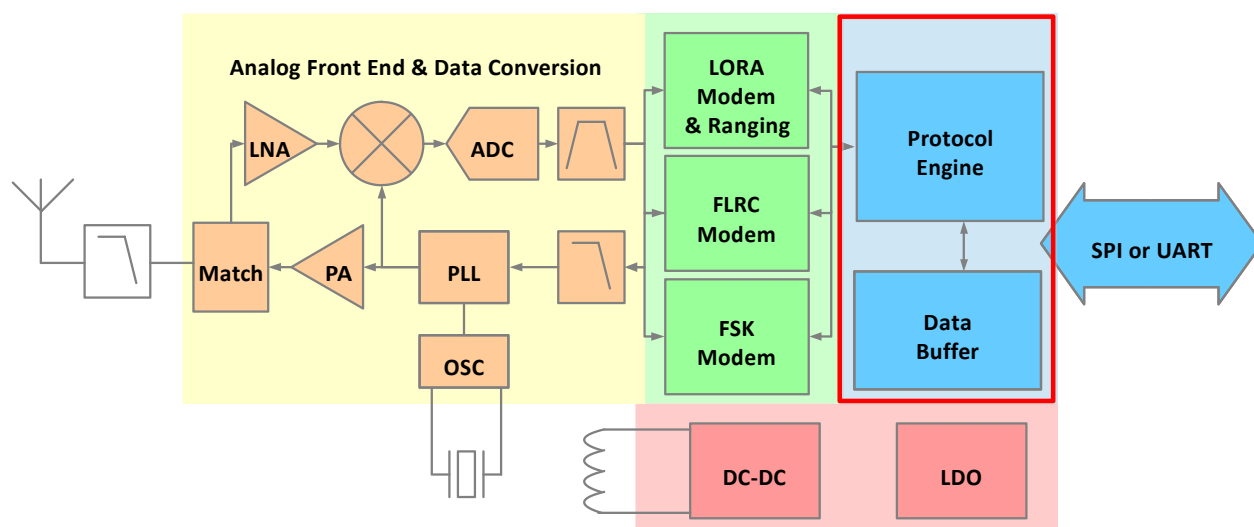


图 7-1: 收发器框图，突出显示的数据包引擎

收发器设计用于基于分组的操作。分组控制器以半双工模式工作，即一次发送或接收。使用第 84 页上的第 12 章“命令列表”中概述的命令 *SetPacketParam ()* 配置数据包控制器。

鉴于数据包引擎的操作取决于所选的数据包类型，必须在配置数据包参数之前使用 *SetPacketType ()* 命令选择数据包类型。

在接收模式中，分组控制器块负责数据比特流的组装和恢复以及其在数据缓冲器中的存储。第 48 页的“数据缓冲区”中详细介绍了数据缓冲区。它还对接收的比特流执行比特流解码操作，例如去白化和 CRC 校验。

在发送模式中，分组处理器构造分组并将其发送到调制器以进行传输。它还可以执行所选择的分组类型所需的所有编码和解码，包括数据白化，CRC 校验，交织，卷积编码和 FEC。

分组控制器块支持五个不同的分组帧，即 GFSK 帧，蓝牙低功耗（BLE），快速远程通信（FLRC）和 LoRa 分组（其又包括通信和测距模式分组类型）。

警告！

收发器实现实现蓝牙低能量通信帧所必需的蓝牙低功耗物理和链路层的各方面。在本节中，给出了数据包格式，数据传输和接收的详细信息。在这种操作模式中，不同于 1 Mb/s 的比特率也可用于解决使用相同分组格式的其他应用。值得注意的是，报头，有效载荷，数据包类型和数据包长度不会被收发器解释，而只会存储在数据缓冲区中。

7.1 GFSK 数据包

GFSK 分组格式提供传统的分组格式，用于专有的非归零（NRZ）编码，远程，低能量通信链路中的应用。数据包格式具有内置功能，用于 CRC 检查有效负载和动态有效负载大小。可选地，可以启用基于伪随机数生成（PRNG）的白化变换。GFSK 数据包与 FSK 调制一起使用。

GFSK 帧中有两种主要的数据包格式：固定长度和可变长度数据包。

7.1.1 固定长度数据包

如果分组长度是固定的并且在链路的两侧都是已知的，那么不需要通过空中传输分组长度的知识。相反，可以将数据包长度写入 `packetLength` 参数，该参数确定数据包长度（以字节为单位）（0 到 255）。

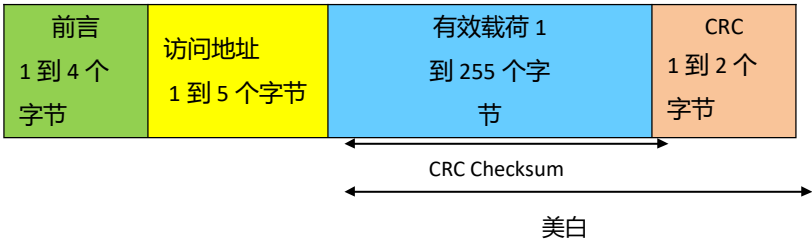


图 7-2：固定长度数据包格式

使用 `PreambleLen` 参数，前导码长度以半字节增量设置为 0.5 到 4 个字节。对于 1 Mb / s 通信，建议至少使用 1 个字节的前导码。对于所有其他数据速率，至少需要 2 个字节。所述 CRC 操作，数据包长度和前同步码长度是使用所定义的 `SetPacketParam ()` 命令中定义第 12 章“命令列表” 83 页上。

7.1.2 可变长度包

在分组具有不确定或可变大小的情况下，必须在分组内发送关于分组长度的信息。可变长度数据包的格式如下所示。

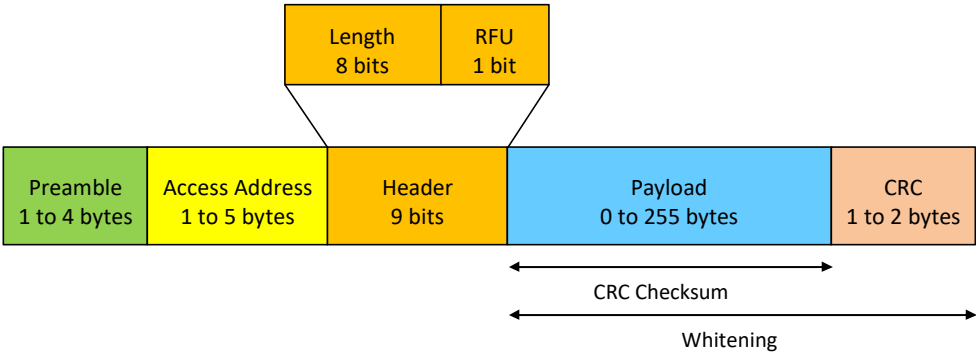


图 7-3：可变长度数据包格式

7.2 BLE 包格式

所有通信模式共用的 BLE 数据包格式如下图所示。它包括单字节的前同步码，后跟 4 字节的访问码，它与所使用的 RF 信道号，协议数据单元（PDU）和 3 个 CRC 字节相关。

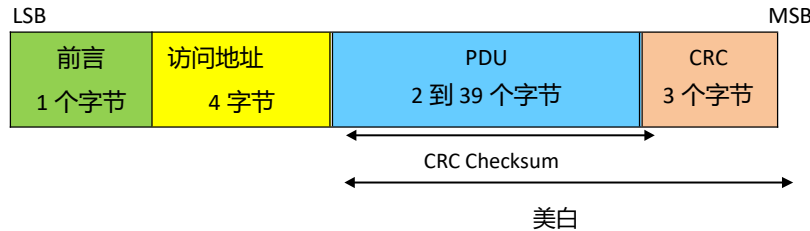


图 7-4：BLE 数据包格式

该 PDU 有两种格式：广告信道 PDU 和数据信道 PDU。在这两种情况下，PDU 由 2 字节的报头和有效载荷数据，0 到 31 字节的广告有效载荷和 0 到 37 字节的数据信道有效载荷组成。广告 PDU 格式用于周期性地广播并向三个配对信道之一上的任何监听（发起者）设备发起连接请求。一旦建立通信，发起者就成为主设备而广告商成为从设备。

该广告标题包含：

- 4 位表示 7 种 PDU 数据包中的一种
- 2 位保留供将来使用（RFU）
- TxAdd（ress）和 RxAdd（ress）位，用于指示广告信道数据包类型
- 6 位长度字段，用于指示有效负载的长度
- 2 个保留位

该数据信道报头中包含：

- LLID 指示数据包是控件还是数据类型
- NESN 是下一个预期序列号
- SN 是当前序列号
- MD 代表更多数据
- Length

（MIC

是有效负载+消息完整性检查
）长度（以字节为单位）

PDUtype 4 位	RFU 2 位	TxAdd 1 位	RxAdd 1 位	长度 6 位	RFU 2 位
----------------	------------	--------------	--------------	-----------	------------

LLID	NESN	SN	MD	RFU	长度	RFU
2 位	1 位	1 位	1 位	3 位	5 位	3 位

广告标题

数据通道标头

图 7-5: PDU 标头格式

请注意，标头不是由收发器生成的，必须在外部计算并作为有效负载的一部分传递到数据缓冲区。

7.3 FLRC 数据包

FLRC 是一个连贯的 MSK 调制解调器，允许比 LoRa 调制解调器更高的数据速率通信。然而，与传统的基于 FSK 的调制相比，FLRC 调制解调器具有更高的灵敏度和更好的链路预算。卷积编码和解码也用于进一步增强链路预算和抗干扰性。

7.3.1 FLRC 数据包格式

虽然是专有的，但 FLRC 数据包在其构造中是常规的。它具有标题，同步字和 CRC 结构。与 GFSK 模式类似，两种数据包格式可用于固定和可变长度数据包。

7.3.2 固定长度数据包格式

固定包长度格式如下图所示。该数据包包含以下元素：

- 可变长度 AGC 前同步码 - 对于 1.3 Mb / s 数据速率，可以减少到 1 个字节，对于所有其他数据速率，需要 2 个字节
- 21 位定时恢复前导码
- 4 字节的同步字
- 固定长度有效载荷，长度可为 6 到 127 个字节
- 一个 CRC 的长度为 2, 3 或 4 个字节字段
- 最后，数据包由短的 6 位尾随零序列终止

同步字大小，有效负载长度和 CRC 长度由命令 `SetPacketParam ()` 配置，如第 84 页上的第 12 章“命令列表”中所述。所述 CRC 是先前分组以外的前导码总体上执行。

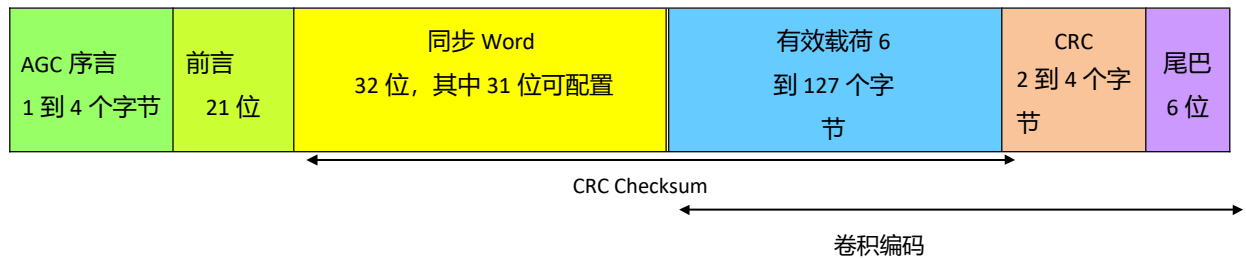


图 7-6: FLRC 固定长度数据包格式

7.3.3 可变长度数据包格式

可变格式分组在形式和功能上与固定分组长度格式相同，但增加了应用 CRC 和卷积编码的报头。标题结构是固定的，具有 2 位类型声明，请参见下图中的映射。其后是有效载荷长度。

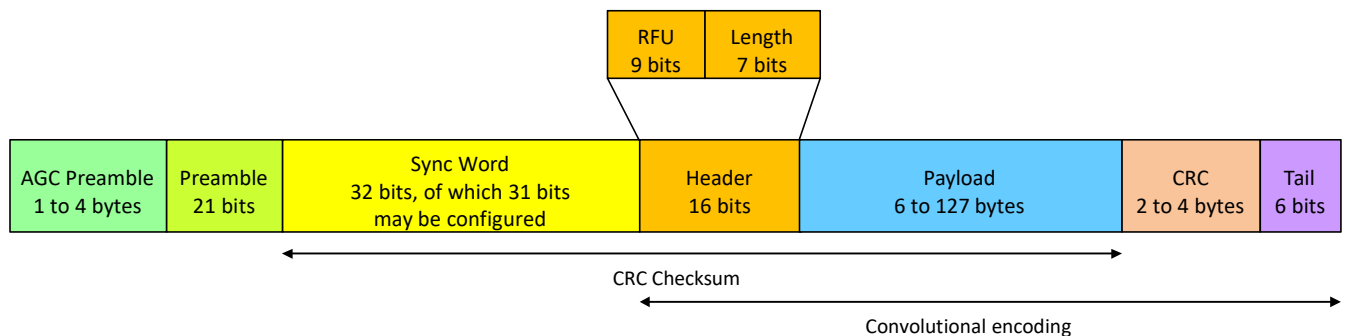


图 7-7: FLRC 可变长度数据包格式

7.3.4 FLRC 播出时间

在 FLRC 数据包中传输的总比特数如图 7-6 中所定义：FLRC 固定长度数据包格式。因此，总的空中时间的计算是有效载荷比特数（补偿卷积编码的影响）和标题比特数的组合。表示数据包的每个字段中的位数 n ，标头位数是：

$$\tilde{N}_{\text{未编码}} = N_{AGCPreamble} + N_{\text{序言}} + N_{SYNCWORD} + N_{\text{头}} \times VL$$

其中 $n_{\text{标头}}$ 是 16 位，如果是可变长度则 $VL = 1$ ，否则 $= 0$ 。和有效编

码位数：

$$\tilde{N}_{\text{编码}} = \left\lceil \frac{\tilde{N}_{\text{净荷}} + N_{CRC} + N_{\text{尾}}}{\text{CR}} \right\rceil \times \text{CR}$$

其中 CR 是编程为编码率的值，请参见表 13-31：FLRC 模式下的调制参数：编码率。如果 $CR = 1/2$ 或 $3/4$ ， n_{tail} 取决于 CR ： $n_{tail} = 6 \text{ bit}$ ；在其他情况下， $n_{tail} = 0$ 。
给定 FLRC 数据速率的比特周期很简单：

$$t_{bit} = \frac{I}{R \cdot b}$$

FLRC 中的 R_b 值可以在表 6-3 中找到：有效的 FLRC 数据速率和带宽组合。
并且总包传输时间由下式给出：

$$T_{\text{总}} = T_{\text{比特}} \times (N_{\text{未编码}} + N_{\text{编码}})$$

7.4 LoRa 包

LoRa 调制解调器采用两种类型的包格式，显式和隐式。显式分组包括短头，其包含关于字节数，编码率以及 CRC 是否附加到分组的信息。

7.4.1 LoRa 数据包格式

LoRa 分组以前导序列开始，该序列用于使接收器与输入信号同步。默认情况下，数据包配置有 12 个符号的长序列。该前导码长度是可编程的并且可以扩展；例如，在接收密集型应用中减少接收器占空比。可编程前导码长度可配置为 8 到 65535 个符号。LoRa 调制解调器自动添加 4.25 个符号，使实际前导码长度范围从 12.25 到 65539.25 个符号。这允许传输接近任意长的前导序列。

接收器进行周期性重启的前导码检测过程。因此，前导码长度应该与发射机前导码长度相同地配置。在前导码长度未知或可能变化的情况下，应在接收侧编程最大前导码长度。

可选的头部可以包括在 LoRa 分组中。显式（可变长度）和隐式（固定长度）头部模式分别指示包头的包含或排除。

7.4.2 显式（可变长度）标头模式

这是默认的操作模式，包括具有以下功能的标头：

- 有效负载长度，以字节为单位
- 前向纠错码率
- 有效负载存在可选的 16 位 CRC

始终使用最强纠错码对 8 符号长报头进行编码，以允许接收应用于有效载荷的任何 FEC 值的分组。标头还具有单独的 CRC，允许接收器丢弃无效的标头。

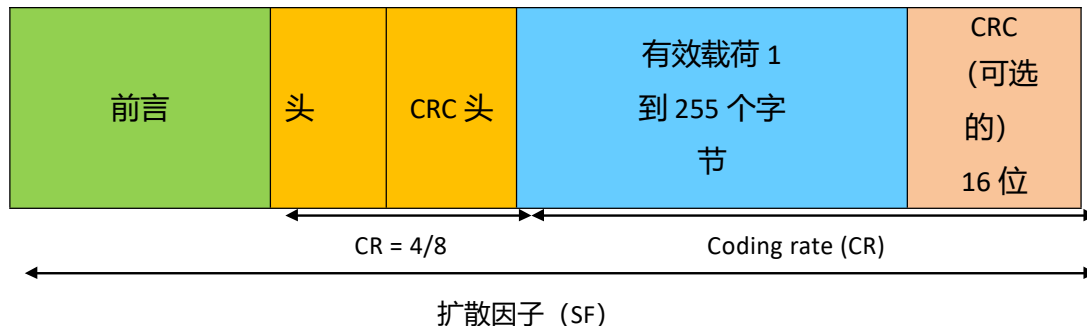


图 7-8: LoRa 可变长度数据包格式

7.4.3 隐式（固定长度）标头模式

在有效载荷，编码率和 CRC 存在被预先知道的情况下，可以通过移除头部（隐式模式）来减少分组持续时间。这里，有效载荷长度，错误编码率和有效载荷 CRC 的存在必须在无线电链路的两侧手动配置。在标题部分之后是以指定的错误率编码的预配置长度的有效载荷。注意，在所有头部操作模式中，可以可选地将 CRC 附加到分组以检测损坏或无效的分组有效载荷。

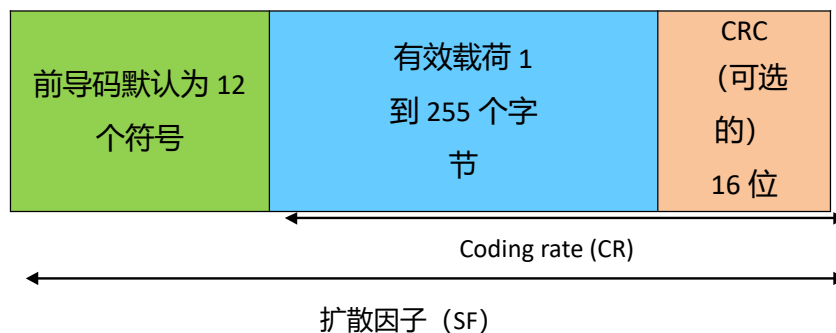


图 7-9: LoRa 固定长度数据包格式

7.4.4 LoRa 播出时间

LoRa 调制解调器的数据包格式如图 7-8 所示：LoRa 可变长度数据包格式和图 7-9：LoRa 固定长度数据包格式。获得 Time Air (ToA) 的等式是：

$$ToA = \frac{2^{SF}}{BW} * N_{symbol} \text{ 有:}$$

- SF: 扩散因子 (5 到 12)

- BW: 带宽 (以 kHz 为单位)
- ToA: 以毫秒为单位的播出时间
- N 符号: 符号数

符号数的计算根据调制的参数而不同。

对于 CR 是传统编码率 (即, 不长交错):

$$N_{symbol} = N_{symbol_preamble} + 6.25 + 8 + \text{ceil} \left(\frac{\max(8 * N_{byte_payload} + N_{bit_CRC} - 4 * SF + N_{symbol_header}, 0)}{4 * SF} \right) * (CR + 4)$$

如果 SF < SF7

$$N_{symbol} = N_{symbol_preamble} + 4.25 + 8 + \text{ceil} \left(\frac{\max(8 * N_{byte_payload} + N_{bit_CRC} - 4 * SF + 8 + N_{symbol_header}, 0)}{4 * SF} \right) * (CR + 4)$$

如果 SF7 ≤ SF ≤ SF10

$$N_{symbol} = N_{symbol_preamble} + 4.25 + 8 + \text{ceil} \left(\frac{\max(8 * N_{byte_payload} + N_{bit_CRC} - 4 * SF + 8 + N_{symbol_header}, 0)}{4 * (SF - 2)} \right) * (CR + 4)$$

如果 SF > SF10

附:

- 如果 CRC 激活, 则 N_bit_CRC = 16, 否则为 0
- 如果标头是可变的, 则 N_symbol_header = 20, 如果固定则为 0 • 对于相应的编码率 4 / 5, 4 / 6, 4 / 7 或 4 / 8,

CR 为 1, 2, 3 或 4 **对于长交错情况:**

$$N_{symbol} = N_{symbol_preamble} + 6.25 + 8 + \text{ceil} \left(\frac{\max(8 * N_{byte_payload} + N_{bit_CRC} - 8 * \text{floor}(\frac{4 * SF - N_{symbol_header}}{8}), 0) * CR}{4 * SF} \right)$$

如果 SF < SF7

$$N_{symbol} = N_{symbol_preamble} + 4.25 + 8 + \text{ceil} \left(\frac{\max(8 * N_{byte_payload} + N_{bit_CRC} - 8 * \text{floor}(\frac{4 * SF - 8 - N_{symbol_header}}{8}), 0) * CR}{4 * SF} \right)$$

如果 SF7 ≤ SF ≤ SF10

$$N_{symbol} = N_{symbol_preamble} + 4.25 + 8 + \text{ceil} \left(\frac{\max(8 * N_{byte_payload} + N_{bit_CRC} - 8 * \text{floor}(\frac{4 * SF - 8 - N_{symbol_header}}{8}), 0) * CR}{4 * (SF - 2)} \right)$$

如果 $SF > SF_{10}$

附：

- 如果 CRC 激活，则 $N_{\text{bit_CRC}} = 16$ ，否则为 0
- 如果标题是可变的，则 $N_{\text{symbol_header}} = 20$ ，如果标题是固定的则为 0
- 对于各自的编码率 $4/5$ LI, $4/6$ LI 或 $4/8$ LI，CR 为 5,6 或 8

7.5 LoRa 测距发动机包

测距操作包括在配置为测距主设备的收发器和配置为测距从设备的收发器之间的交换或交换序列。在每次交换中，主设备生成一个测距数据包，通过无线方式发送并由从设备接收。然后，从设备与输入的测距数据包同步并发送测距响应。

当主人接收到时，与测距响应的同步允许扣除主人和奴隶之间的飞行时间。这反过来在概念上被转换为距离。应该注意的是，报告的距离将代表无线电波行进的路径，而不是主机和从机之间的最短路径距离。

测距引擎数据包结构与 LoRa 数据包显式标头模式非常相似（请参见第 43 页上的第 7.4.2 节“显式（可变长度）标头模式”）。标头中的一个保留位被简单地设置为指示分组是测距请求。标头包含 32 位范围的从属 ID。从站将拒绝任何没有匹配 ID 的测距请求。

为了给系统提供一些灵活性，从设备还可以检查测距 ID 的一部分，特别是最不重要的 8,16,24 或 32 位。

主站报告的飞行时间以原始格式（报告单个测距测量的结果）或过滤格式提供。过滤应用非线性过滤功能来聚合多个测距交换结果并提高准确性。有关滤波和测距参数的配置，请参见第 116 页上的第 13.5 节“测距设置”以获取更多信息。

7.5.1 测距包格式

下图显示了测距交换中使用的专用帧：

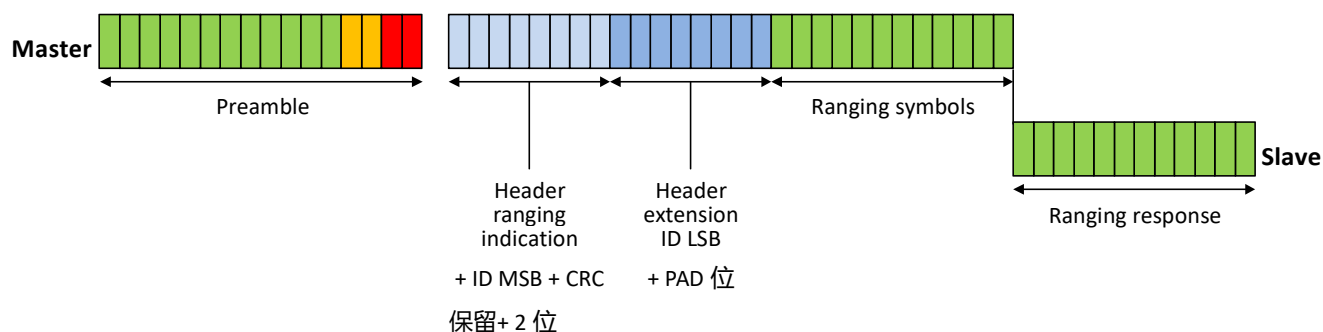


图 7-10：测距数据包格式

7.5.2 测量主交换

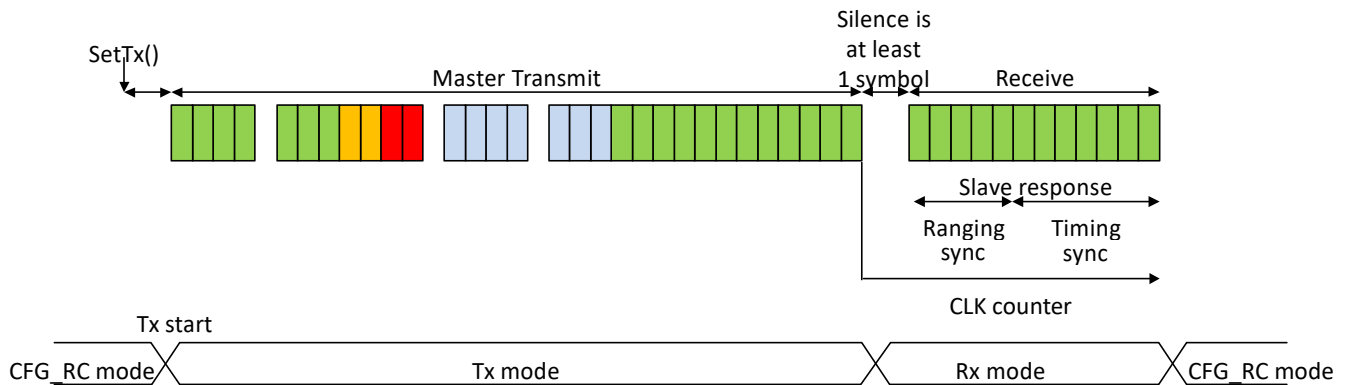


图 7-11：测距主数据包交换

7.5.3 测试从属交换

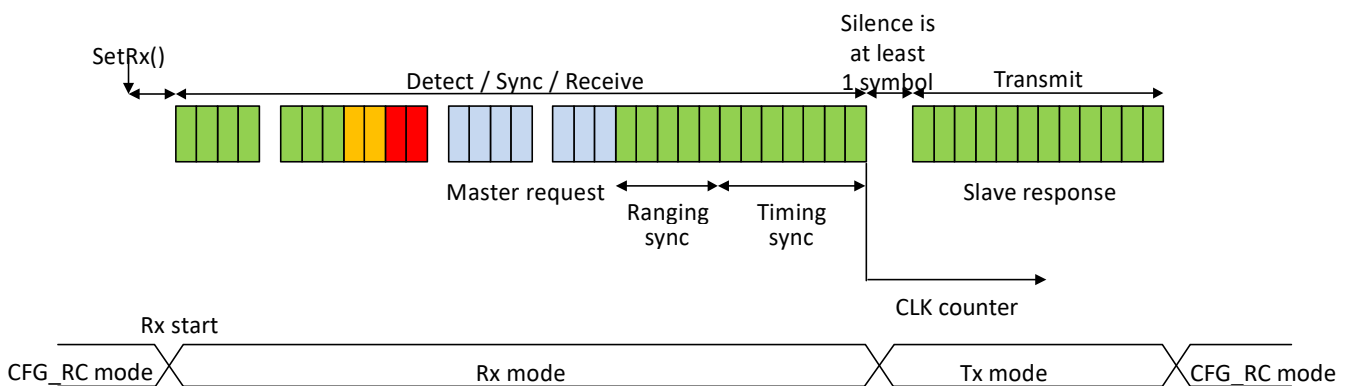


图 7-12：测距从属数据包交换

8.数据缓冲区

收发器配有 256 字节 RAM 数据缓冲区，除睡眠模式外，所有模式均可访问。该 RAM 区域可由用户完全定制，并允许访问数据以进行传输或从最后的数据包接收。所有对数据缓冲区的访问都可以通过第 9 章“数字接口和控制”的 SPI 或 UART 接口进行。

8.1 操作原理

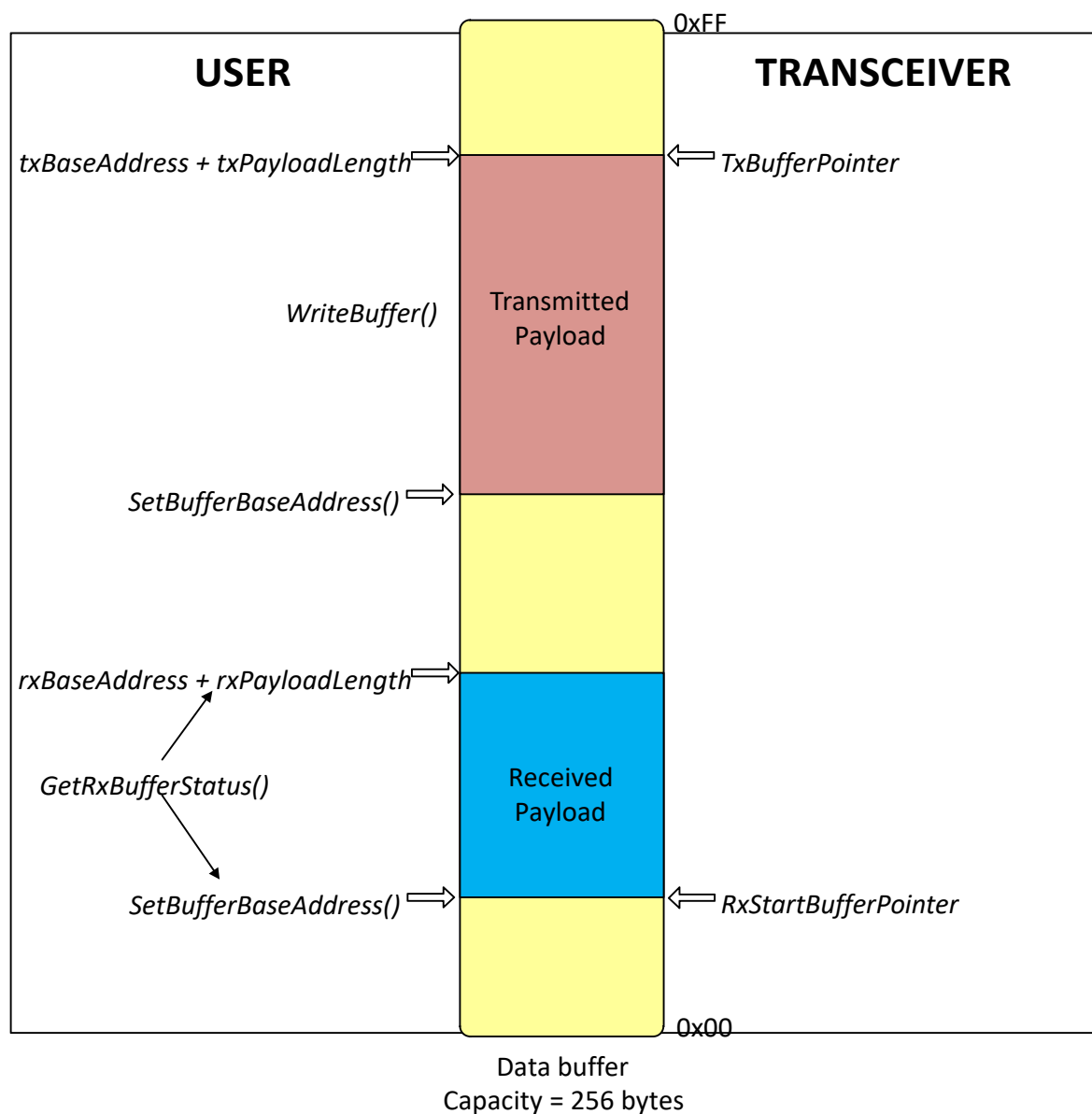


图 8-1：数据缓冲图

数据缓冲区可以配置为存储发送和接收有效负载。

8.2 接收操作

在接收模式下，`rxBaseAddress` 指定内存中将写入接收的数据包有效内容数据的缓冲区偏移量。然后，以接收模式写入的最后一个字节的缓冲区偏移量存储在 `RxDataPointer` 中，该接收器在接收开始时初始化为值 `rxBaseAddress`。

可以通过命令 `GetRxbufferStatus()` 读取指向最后一个数据包的第一个字节的指针和数据包长度。

在单模式下，每次收发器进入 Rx 时，*RxDataPointer* 都会自动初始化为 *rxBaseAddress*。在连续模式中，指针从前一个位置开始递增。有关更多详细信息，请参见第 56 页上的第 10.5 节“接收 (Rx) 模式”。

8.3 传输操作

在每次转换到发送模式时，指针 *TxDataPointer* 被初始化为 *txBaseAddress*，并且每次通过空中发送字节时递增。一旦发送的字节数等于函数 *SetPacketParam ()* 中定义的 *payloadlength* 参数，此操作就会停止。

8.4 使用数据缓冲区

既，*rxBaseAddress* 和 *txBaseAddress* 使用命令 *SetBufferBaseAddress ()* 设置。

默认情况下，*rxBaseAddress* 和 *txBaseAddress* 在地址 0x00 处初始化。

由于数据缓冲区的连续性，Tx 和 Rx 的基址可在 256 字节存储区中完全配置。每个指针都可以在缓冲区内的任何位置独立设置。要在发送或接收模式下利用最大数据缓冲区大小，可以通过在内存底部设置基址 *txBaseAddress* 和 *rxBaseAddress* (0x00)，在每种模式下使用整个数据缓冲区。

当器件进入深度休眠模式（意味着无法访问）时，数据缓冲区被清除。数据保留在所有其他操作模式中。

通过将数据缓冲区保持在保留状态，可以将数据值保持在休眠模式。但是指针位置将丢失。为了从睡眠保留中检索数据，用户必须使用基地址的默认值（例如，对于 Rx 为 0x00，对于 Tx 为 0x80），或者在进入休眠模式之前存储 *PayloadLengthRx* 和 *RxStartBufferPointer*。

使用命令 *WriteBuffer ()* 和 *ReadBuffer ()* 通过 SPI 或 UART 加入数据缓冲区。在此函数中，参数偏移量定义了要写入或读取的第一个数据的地址指针。偏移零定义数据缓冲区的第一个位置。

因此，在进行任何读取或写入操作之前，必须将此偏移量初始化为缓冲区的相应开头。在读取或写入数据缓冲区时，地址指针将自动递增。

存在两种获得偏移值的可能性：

- 首先使用 *rxBaseAddress* 值，因为用户在接收有效负载之前对其进行了定义。
- 其次，偏移可以用价值来初始化 *RxStartBufferPointer* 通过返回 *GetRxbufferStatus* 命令。

重要的是要注意，即使 CRC 无效，所有接收的数据也将被写入数据缓冲区，允许用户定义的后处理损坏的数据。同样重要的是要注意，在接收时，如果数据包大小超过为 Rx 分配的缓冲存储器，它将覆盖数据缓冲区的发送部分。

9. 数字接口和控制

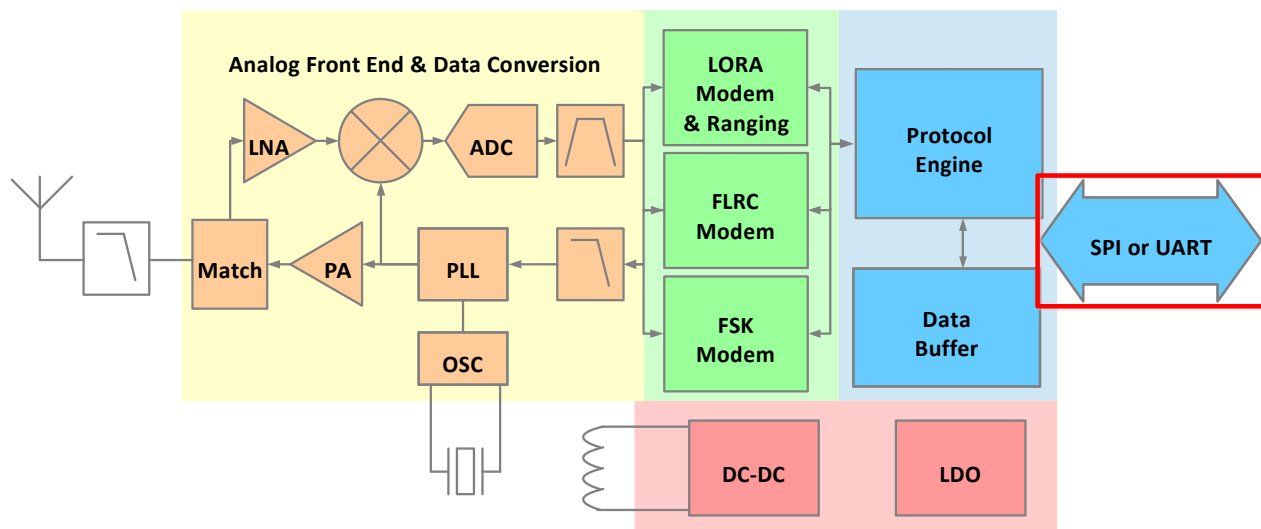


图 9-1：收发器框图，数字接口突出显示

收发器通过串行接口（SPI 或 UART）和一组通用输入/输出（DIO）进行控制。收发器使用协议引擎来处理通信和收发器控制（模式切换，API 等.....）。通过 SPI 或 UART，应用程序将命令发送到内部芯片或直接访问数据存储空间。所有寄存器均可通过 SPI 或 UART 访问。

9.1 BUSY 引脚通信

在所有通信中，BUSY 引脚用作忙信号，表示只有在该信号为低电平时，收发器才准备好接收新命令。请参见第 14 页的第 2 节“引脚连接”。

9.2 接口检测

两个接口都被启用，直到第一个接收到有效的事务，这将禁用未使用的接口。

为了允许 UART 接收，RTSN 需要被驱动为低电平。但是，由于它与 SCK 共用，因此最初使用高阻抗驱动器将引脚 18 驱动为低电平。如果检测到 UART 接口，则引脚 18 由片内 UART 直接驱动；否则引脚配置为输入并由外部 SPI 主控制器驱动。

9.3 SPI 接口

所述 SPI 接口使经由对应于同步全双工帧访问配置寄存器 $CPOL = 0$ 和 $CPHA = 0$ 在摩托罗拉/飞思卡尔命名法。仅实现从属方。

发送一个地址字节后跟一个数据字节用于写访问，而发送一个地址字节并接收一个读字节用于读访问。在 NSS 销在帧的开始变低并进入数据字节后高。

MOSI 由主机在 SCK 的下降沿产生，并在 SCK 的上升沿由从机（即该 SPI 接口）进行采样。MISO 由从机在 SCK 的下降沿产生。

始终通过 NSS 引脚变为低电平来启动传输。当 NSS 高时，MISO 是高阻抗。

SPI 在外部 SCK 时钟上运行，允许高达 20 MHz 的高速

主机通过提升 NSS 信号来终止 SPI 事务，它不会明确地将命令长度作为参数发送。主机不得在事务的字节内引发 NSS。

如果主机发送需要参数的命令，则必须在引发 NSS 之前发送所有参数。如果不是，收发器将使用未知值来丢失参数。

9.3.1 收发器处于活动模式时的 SPI 时序

未处于休眠模式时，认为收发器处于活动模式。在工作模式下，收发器可以立即处理标准 SPI 命令，即第一次 SPI 事务不需要额外的延迟。主要原因是，与处于睡眠模式时的行为相反，收发器不必经历启动过程。

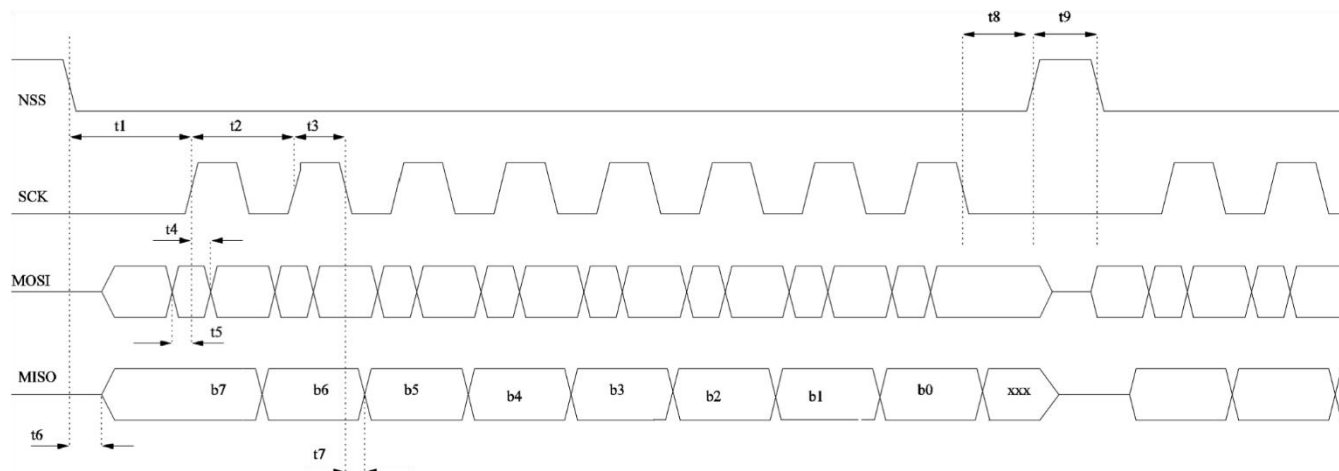


图 9-2: SPI 时序图

9.3.2 收发器离开休眠模式时的 SPI 时序

收发器离开睡眠模式的方法是等待 NSS 的下降沿。在下降沿，所有必要的内部稳压器都接通；收发器在能够接受第一个 SPI 命令之前开始初始化。这意味着 NSS 下降沿和 SCK 第一个上升沿之间的延迟必须考虑唤醒序列和收发器初始化。

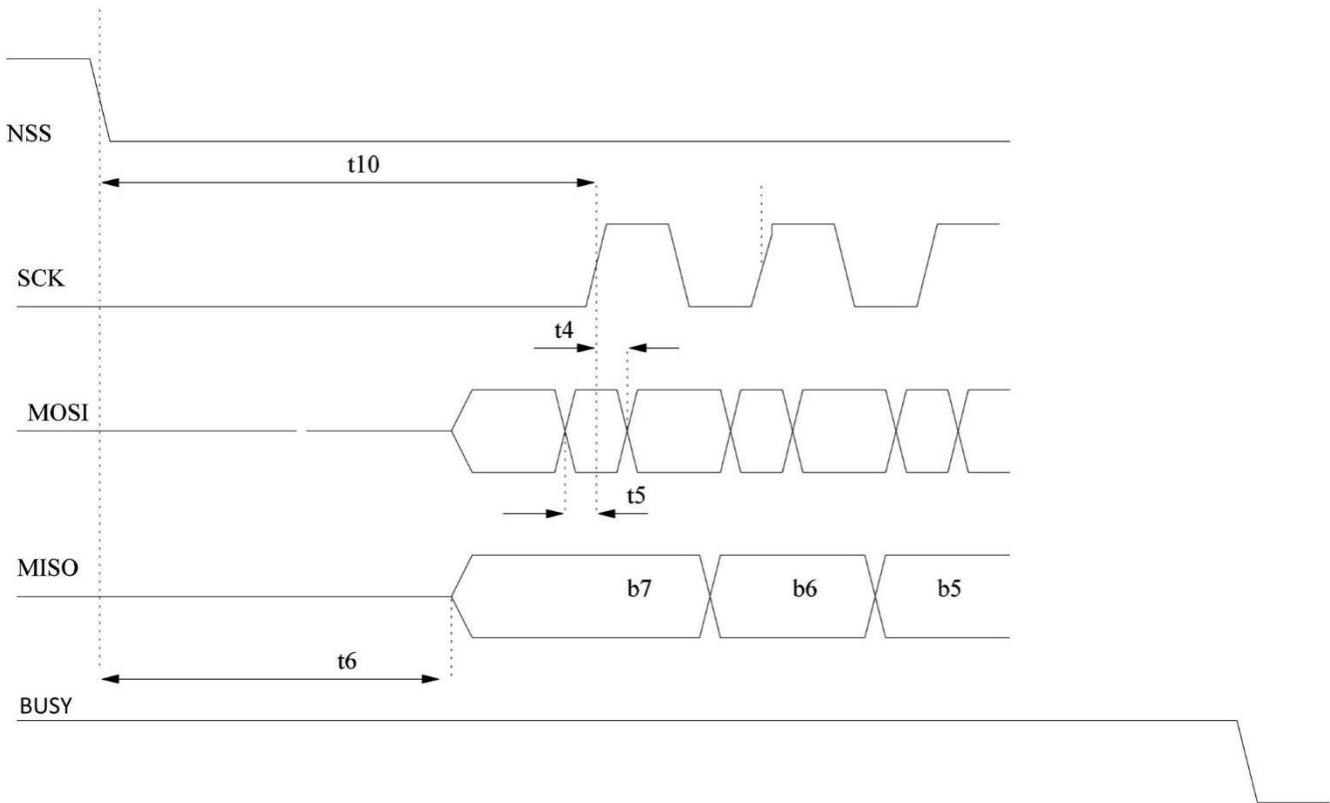


图 9-3：SPI 时序转换

在休眠模式和初始化阶段，BUSY 引脚上映射的忙信号设置为高电平，向主机指示收发器无法接受新命令。收发器处于 STDBY_RC 模式后，忙信号变为低电平，主机可以开始发送命令。请注意，在开机或硬重置后启动也是如此。

有关 SPI 时序的值，请参见第 53 页上的第 9.3.3 节“SPI 时序”。

9.3.3 SPI 时序

以下规格给出了 $V_{BAT_IO} = V_{BAT} = 3.3\text{ V}$ ，温度 = 25°C ，晶体振荡器频率 = 52 MHz 的典型工作条件。

所有时间在下表中给出，最大负载上限为 10 pF 。

表 9-1：SPI 时序要求

符号	描述	最低限度	典型	最大值	单元
----	----	------	----	-----	----

T1	NSS 下降到 SCK 设置时间	25	-	-	NS
T2	SCK 期间	50	-	-	NS
T3	SCK 时间很长	25	-	-	NS
T4	MOSI 到 SCK 有时间	五	-	-	NS
T5	MOSI 到 SCK 的设置时间	五	-	-	NS
T6	NSS 陷入 MISO 延迟	0	-	15	NS
T7	SCK 下降到 MISO 延迟	0	-	15	NS
T8	SCK 到 NSS 上升沿保持时间	25	-	-	NS
T9	NSS 很高兴	100	-	-	NS
T10	从休眠模式切换到 STDBY_RC 模式时, NSS 下降到 SCK 建立时间	125	-	-	微秒

9.4 UART 接口

收发器 **UART** 支持以下设置：

- 波特率：921.6 k, 460.6 k, 115.2 k, 57.6 k, 38.4 k, 19.2 k, 9.6k
- RTS / CTS 流量控制
- 奇偶校验控制：无，奇数，偶数
- 8 位字
- 1,2 个停止位
- Rx 满，Tx 为空，错误（奇偶校验，无停止位）中断。

最初，**UART** 配置为以 115.2 kb / s 运行，其中 1 个停止位，偶校验，CTS 流控制和最低有效位 (**LSB**) 首先到达。在启动时，必须将 CSTN 驱动为低电平以启动通信。然后可以配置其他兼容的 **UART** 通信设置。

在 **UART** 事务中，主机必须提供命令长度。一旦收到所需的字节，设备就开始处理事务。后续字节被视为属于新事务。

9.5 引脚共享

SPI 和 **UART** 之间的引脚通过以下方式共享：

- **NSS** (IN) / **CTSN** (IN)
- **SCK** (IN) / **RTSN** (OUT)
- **MOSI** (IN) / **RX** (IN)
- **MISO** (OUT) / **TX** (OUT)

9.6 多功能数字输入/输出 (DIO)

收发器提供 3 个可配置为中断的 **DIO**。

BUSY 引脚用作中断，始终为输出。当处理完当前命令并且设备已准备好接受新的命令时，将激活忙碌中断。

此外，可以选择 3 个 **DIO** 中的任何一个作为收发器的外部中断源。

注意：可以使用 **SetDioIrqParams ()** 命令将 3 个 **DIO** 中的任何一个映射到来自收发器的任何中断信号。有关完整详细信息，请参见第 81 页上的第 11.8.1 节“**SetDioIrqParams**”。

当应用程序收到中断时，它可以通过读取器件 **IRQ** 寄存器来确定原因。可以使用 **ClearIrq ()** 命令清除中断。

使用 [SPI](#) 接口时，将在每个不需要发送数据的事务上发送状态。

使用 [UART](#) 接口时，可以通过 `GetDeviceStatus ()` 命令检索状态。

10.运作模式

收发器具有六种工作模式，每种工作模式下启用的模拟前端和数字模块如下表所示：

表 10-1：SX1280 工作模式

模式	启用块
睡觉	可选寄存器，备用稳压器，RC32K 振荡器，数据缓冲器，数据 RAM
STDBY_RC	顶部稳压器 (LDO)，RC13M 振荡器
STDBY_XOSC	顶部稳压器 (DC-DC)，RC13M 振荡器， XOSC
FS	频率合成器，Tx 频率
TX	频率合成器和发射器，调制解调器
RX	频率合成器和接收器，调制解调器

10.1 启动

上电时，收发器进入启动状态。BUSY 引脚设置为高电平，表示收发器处于忙碌状态且无法接受命令。当数字电压和 RC 时钟可用时，收发器可以启动并启动校准阶段，该阶段包括：

- 借助 52 MHz 晶振校准 RC13 MHz。这是正确建立 [UART](#) 通信所必需的
- 借助 52 MHz 晶振校准 RC 64K。
- 校准 [PLL](#) 调制路径
- 校准 [ADC](#)

校准终止后，收发器进入 STDBY_RC 模式。收发器现已就绪，BUSY 引脚变为低电平，表示器件已准备好接受来自主机的命令。

校准的所有结果都存储在数据存储器中。当收发器从休眠模式唤醒并保留数据存储器内容时，将从存储器中检索校准数据，而不重复完整的过程。

10.2 睡眠模式

在此模式下，只有启动和睡眠控制器 (SCC) 块以及可选的 RC64K 和定时器正在运行，存储器可能处于保留状态。收发器可以从 STDBY_RC 状态进入该模式，并且如果发生以下事件之一，则可以离开睡眠模式：

- NSS 引脚 (19) 变为低电平。
- **RTC** 定时器生成计数结束 (对应于 Duty 循环操作)。请参见第 67 页上的第 11.5.6 节“SetRxDutyCycle”。

10.3 待机模式

在待机模式下，主机应在进入 Rx 或 Tx 模式之前配置收发器。默认情况下，在此状态下，系统由 13 MHz RC 振荡器提供时钟，以降低所有其他模式的功耗，除了休眠模式，晶振开启。但是，如果应用程序是时间关键的，则可以将 **XOSC** 块打开或保持打开状态。

STDBY 模式下的晶体振荡器 (**STDBY_XOSC**) 或 13 MHz RC 振荡器 (**STDBY_RC**) 选择由命令 *SetStandby* (*oscillatorMode*) 命令中的模式参数决定。

如果 **XOSC** 与 DC-DC 电源调节一起使用，DC-DC 将自动以 **STDBY_XOSC** 模式供电。

10.4 频率合成 (FS) 模式

在 FS 模式下，**PLL** 和相关稳压器接通。PLL 锁定后，**BUSY** 引脚变为低电平。

可以通过使用 *SetFs* () 命令请求无线电保持此模式。

由于收发器使用低 IF 架构，因此 Rx 和 Tx 频率不同。Rx 频率等于 Tx 频率减去中频 (IF) 偏移 (默认为 1.3 MHz)。

在 FS 模式下，**PLL** 调谐的频率对应于发射频率。

10.5 接收 (Rx) 模式

在 Rx 模式下，**LNA**，**MIXER**，**PLL** 和选定的调制解调器 (LoRa / **FSK** / **FLRC**) 打开。在连续模式下，设备保持 Rx 模式并查找传入的数据包，直到主机请求不同的模式。在单模式下，设备自动返回 **STDBY_RC** 模式。

通过发出带有 *periodBase* 振荡器时基和 *periodBaseCount* 时钟周期数的 *SetRx* (*periodBase*, *periodBaseCount*) 命令来指定接收模式退出到 **STDBY_RC** 模式的超时，从而转换到接收模式。定时接收过程可以在收发器中完全自动化。有关每种调制格式的此过程和处理，请参见第 92 页上的第 13.1.3 节“Rx 设置和操作”。

10.6 发送 (Tx) 模式

在 Tx 模式下，功率放大器 (**PA**) 在上升之后传输数据缓冲器。发送数据包后，设备自动返回 **STDBY_RC**。

10.7 收发器电路模式图形说明

所有器件工作模式和每种模式选择转换的状态如下图所示：

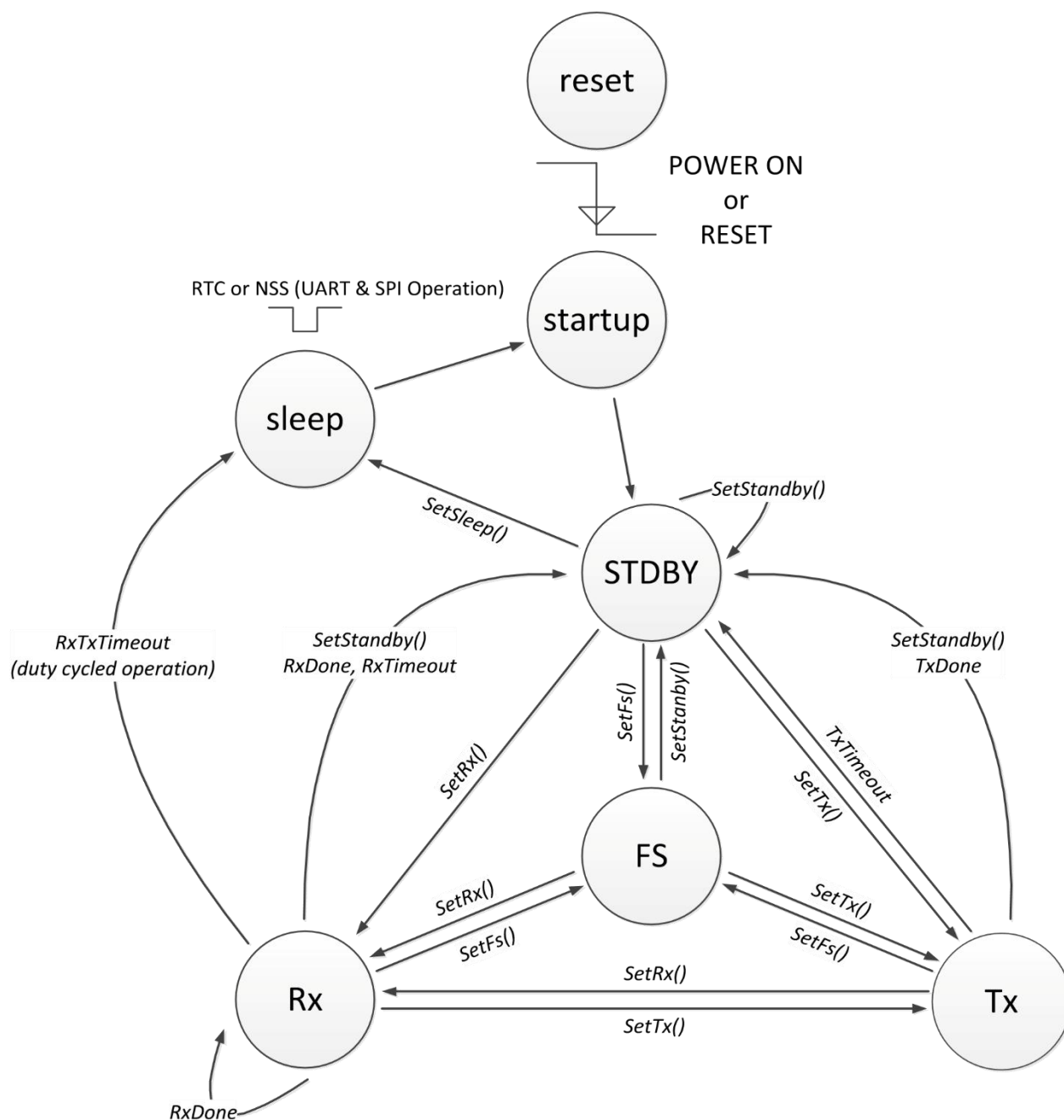


图 10-1: 收发器电路模式

10.8 活动模式切换时间

在与收发器的每次交易（寄存器读/写操作或模式切换）中，BUSY 引脚在事务期间和收发器正在处理命令时设置为高电平。一旦收发器准备好接收新命令或达到稳定模式，BUSY 引脚将回到零。在下图中，切换时间定义为结束 SPI 事务的 NSS 上升沿与 BUSY 引脚下降沿之间的时间。

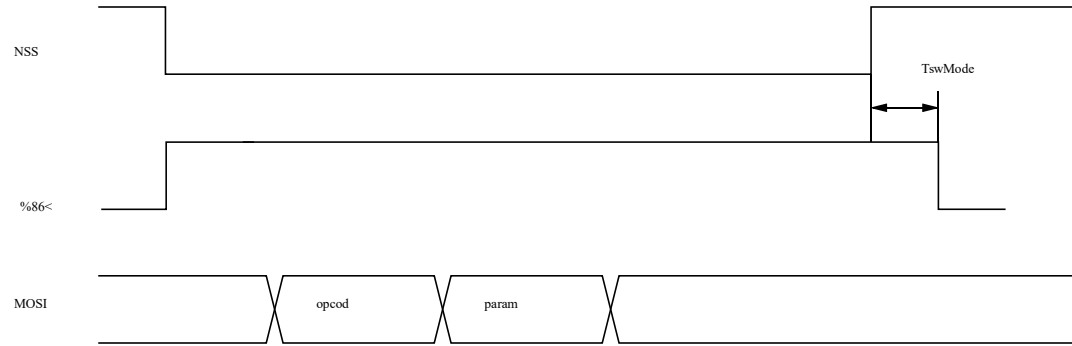


图 10-2：活动模式下的切换时间定义

过渡	TswMode 典型值[μs]
SLEEP 到 STDBY_RC	1700
SLEEP 到 STDBY_RC	250
STDBY_RC 到 STDBY_XOSC	53
STDBY_RC 到 FS	83
STDBY_RC 到 Rx	115
STDBY_RC 到 Tx	102
STDBY_XOSC 到 FS	40
STDBY_XOSC 到 Tx	54
STDBY_XOSC 到 Rx	68
FS 到 Rx	34
FS 到 Tx	27
Rx 到 FS	13
Rx 到 Tx	39
Tx 到 FS	31
Tx 到 Rx	60

表 10-2：所有可能转换的切换时间 (TswMode)

数据保留没有数据保
留

在 FS 中，当 PLL 锁定时，BUSY 引脚将变为低电平。

在 Rx 中，一旦 Rx 上升并准备好接收数据，BUSY 引脚将变为零。

在 Tx 中，当 PA 上升并且前导码的传输开始时，BUSY 引脚将变为低电平。

11.主机控制器接口

通过 SPI 或 UART 接口，主机可以向收发器发出命令或访问数据存储空间以直接检索或写入数据。在正常操作中，除数据缓冲区外，需要减少数量的直接数据写操作。用户通过 API（指令集）与收发器交互。

收发器使用 BUSY 引脚指示收发器的状态及其在完成内部处理时接收命令的能力。在主机发送命令之前，因此有必要检查 BUSY 引脚的状态以确保收发器处于处理它的状态。

支持两种类型的交易：

- 配置事务（STDBY）：向主机提供直接寄存器访问，即它用于写入或读取收发器配置寄存器或数据缓冲区。
- 命令事务（CMD）：需要更复杂的非原子操作，例如数据包传输/接收或模式切换。

11.1 命令结构

如果命令未传递任何参数，则主机仅发送 SPI 和 UART 接口的操作码（1 个字节）。

如果命令需要参数：

- 在 SPI 传输的情况下，操作码字节紧跟在参数字节之后，NSS 上升沿终止命令。

表 11-1：SPI 接口命令序列

字节	0	[1: N]
来自主机的数据	操作码	参数
要托管的数据	状态	状态

- 在 UART 传输的情况下，操作码字节后跟长度字节（即参数字节数），然后是参数字节。

表 11-2：UART 接口命令序列

字节	0	1	[2: n]的
来自主机的数据	操作码	长度	参数

- 在 UART 缓冲区写操作的情况下，在发送地址的 **LSB** 之后，主机必须发送一个字节，用于定义将要传输的数据字节数。然后必须传输该数量的数据字节。
- 在 UART 缓冲区读操作的情况下，在发送了地址的 **LSB** 之后，主机必须发送一个字节来定义将要接收的数据字节数。然后收发器将该数量的字节发送给主机。
- 在 UART 直接寄存器操作的情况下，在发送操作码和地址后，UART 必须发送要读/写的字节数。

11.2 GetStatus 命令

主机可以直接通过 *GetStatus ()* 命令检索收发器状态：可以随时发出此命令。在 SPI 接口的情况下，设备返回同一事务的状态；在 UART 帧的情况下，状态由命令后的写入事务返回。使用 SPI 接口时，*GetStatus ()* 命令不是必需的，因为设备也会在命令字节上返回状态信息。返回的状态字节如下表所述：

表 11-3：状态字节定义

7: 5	4: 2	1	0
电路模式	命令状态	保留的	忙
0x0: 保留	0x0: 保留		
0x1: 保留	0x1: 收发器已成功处理命令 ¹		
0x2: STDBY_RC	0x2: 主机 ² 可以使用数据		收发器处理时该位为 1 命令或做内部 操作。它反映了 BUSY 引脚 状态。
0x3: STDBY_XOSC	0x3: 命令超时 ³		
		-	
0x4: FS	0x4: 命令处理错误 ⁴		
0x5: Rx	0x5: 无法执行命令 ⁵		
0x6: Tx	0x6: 命令 Tx 完成 ⁶		

- 命令已正确终止
- 已成功接收数据包并可检索数据
- 主持人的交易花了太长时间才完成并引发了内部监督。监视器机制可以被主机禁用，它旨在防止死锁情况。在这种情况下，主机应该重新发送命令。
- 由于操作码无效或者提供的参数数量不正确，收发器无法处理命令。
- 命令已成功处理，但收发器无法执行命令；例如，它无法进入指定的设备模式或发送请求的数据，
- 当前数据包的传输已终止

表 11-4 给出了 *GetStatus ()* 命令的 SPI 事务，表 11-5 给出了 UART 事务。

表 11-4: GetStatus 数据传输 (SPI)

字节	0
来自主机的数据	操作码= 0xC0
要托管的数据	状态

表 11-5: GetStatus 数据传输 (UART)

字节	0	1
来自主机的数据	操作码= 0xC0	-
要托管的数据	-	状态

11.3 注册访问操作

11.3.1 WriteRegister 命令

命令 *WriteRegister ()* 从特定地址开始在数据存储空间中写入一个字节块。在每个数据字节之后地址自动递增，以便将数据存储连续的存储器位置中。SPI 数据传输是在描述表 11-6 和 UART 数据传输是在描述表 11-7。

表 11-6: WriteRegister 数据传输 (SPI)

字节	0	1	2	3	4	...	n
来自主机的数据	操作码= 0x18	地址[15: 8]	地址[7: 0]	数据@地址	数据@地址+1	...	data @ address + (n-3)
要托管的数据	状态	状态	状态	状态	状态	...	状态

表 11-7: 写入寄存器数据传输 (UART)

字节	0	1	2	3	4	五	...	n
主机 UART Tx	操作码= 0x18	地址[15: 8]	地址[7: 0]	长度= (n-4)	数据@地址	数据@地址+1	...	数据@地址+第 (n-4)

11.3.2 ReadRegister 命令

命令 *ReadRegister ()* 从给定地址开始读取数据块。每个字节后地址自动递增。SPI 数据传输如表 11-8 所示，UART 数据传输如表 11-9 所示。在 UART 情况下，要读取的数据数由长度参数提供。

注意，当使用 SPI 时，主机必须在发送 2 个字节的地址后发送 NOP，以便在发送的下一个 NOP 上开始接收数据字节。

表 11-8：读取寄存器数据传输（SPI）

字节	0	1	2	3	4	五	...	ñ
来自主机的数据	操作码= 0x19	地址[15: 8]	地址[7: 0]	NOP	NOP	NOP	...	NOP
要托管的数据	状态	状态	状态	状态	数据@地址	数据@地址+1	...	数据@地址+第 (n-4)

表 11-9：读取寄存器数据传输（UART）

字节	0	1	2	3	4	五	...	ñ
主机 UART TX	操作码 = 0x19	地址[15: 8]	地址[7: 0]	长度	---	----	...	----
芯片 UART TX	---	----	---	----	数据@地址	数据@地址+1	...	数据@地址+第 (n-4)

11.4 数据缓冲操作

11.4.1 WriteBuffer 命令

此函数用于写入要传输的数据有效负载。地址自动递增，当地址超过 255 时，由于数据缓冲区的循环特性，它将绕到 0。地址从作为函数参数给出的偏移量开始。表 11-10 描述了 SPI 数据传输，表 11-11 描述了 UART 数据传输。

表 11-10：WriteBuffer SPI 数据传输

字节	0	1	2	3	...	ñ
来自主机的数据	操作码= 0x1A	抵消	数据@偏移	数据@偏移+1	...	数据@偏移+ (N-2)
要托管的数据	状态	状态	状态	状态	...	状态

表 11-11：WriteBuffer UART 数据传输

字节	0	1	2	3	4	..	ñ
主机 UART Tx	操作码= 0x1A	抵消	长度= n-3	数据@地址	数据@地址+1	...	数据@地址+ (N-3)
芯片 UART Tx	---	----	---	---	----	---	---

11.4.2 ReadBuffer

此函数允许从偏移量开始读取 (n-3) 字节的有效负载。

注意在发送偏移后使用 SPI 时要发送的 NOP。

表 11-12：ReadBuffer SPI 数据传输

字节	0	1	2	3	4	...	ñ
来自主机的数据	操作码= 0x1B	抵消	NOP	NOP	NOP	...	NOP
要托管的数据	状态	状态	状态	数据@偏移	数据@偏移+1	...	数据@偏移+ (N-3)

表 11-13：ReadBuffer UART 数据传输

字节	0	1	2	3	4	...	ñ
主机 UART Tx	操作码= 0x1B	抵消	长度	---	----	...	NOP
设备 UART Tx	---	----	---	数据@偏移	数据@偏移+1	...	数据@偏移+ (N-4)

11.5 无线电操作模式

本章介绍可用于收发器的命令集。该事务仅针对 SPI 提供，但使用 UART 时可以使用相同的命令。

11.5.1 SetSleep

所述 *SetSleep* () 命令用于设置收发器到休眠模式具有最低电流消耗可能的。该命令只能在 STDBY 模式下发送 (STDBY_RC 或 STDBY_XOSC)。在 NSS 的上升沿之后，除了备用调节器 (如果需要) 和 *sleepConfig* 参数中指定的块之外，所有块都将关闭。

表 11-14: SetSleep SPI 数据传输

字节	0	1
来自主机的数据	操作码= 0x84	sleepConfig

在 UART 事务中，主机发送与 SPI 事务相同的字节。所述 *sleepConfig* 参数被定义为：

表 11-15: 睡眠模式定义。

sleepConfig [7: 4]	sleepConfig [2]	sleepConfig [1]	sleepConfig [0]
没用过	0: 指令 Ram 在休眠模式下刷新 (相当于重置)	0: 在休眠模式下刷新数据缓冲区	0: 睡眠模式期间刷新数据 RAM
	1: 保持模式下的指令 RAM	1: 保留模式下的数据缓冲区	1: 数据 Ram 处于保留模式

如果 NSS 的上升沿或收发器 **RTC** 已被编程为唤醒自身，收发器模式将从休眠模式转换为 STDBY_RC

当收发器进入休眠模式时，寄存器的内容将丢失。为避免这种情况，必须执行 *SaveContext* 命令，并且 *SetSleep* 命令必须使用 *sleepConfig* [0] 位，当设置为 1 时，允许将寄存器内容存储在数据存储单元中。保留数据存储单元，并且在从 SLEEP 转换到 STANDBY_RC 时，寄存器填充先前存储在数据存储单元中的值。这减少了主机和收发器之间的交互，在经常使收发器进入睡眠状态以节省功率的系统中非常有用。

11.5.2 SetStandby

命令 *SetStandby* () 用于将器件设置为 STDBY_RC 或 STDBY_XOSC 模式，这是中间功耗级别。在该模式中，收发器可以配置用于将来的 RF 操作。

上电或应用复位后，收发器将以 13 MHz RC 时钟进入 STDBY_RC 模式。

表 11-16: SetStandby SPI 数据传输

字节	0	1
----	---	---

来自主机的数据	操作码= 0x80	StandbyConfig
---------	-----------	---------------

表 11-17: SetStandby UART 数据传输

字节	0	1	2
来自主机的数据	操作码= 0x80	0x01	StandbyConfig

该 *StandbyConfig* 字节定义在接下来的表中给出：

表 11-18: StandbyConfig 定义

StandbyConfig	值	描述
STDBY_RC	0	器件在 RC 13MHz 上运行，设置 STDBY_RC 模式
STDBY_XOSC	1	设备在 XTAL 52MHz 上运行，设置 STDBY_XOSC 模式

11.5.3 SetFs

命令 *SetFs ()* 用于将器件设置为频率合成模式，其中 *PLL* 锁定到载波频率。该模式用于 *PLL* 的测试目的，可以视为中间模式。从 STDBY_RC 模式进入 Tx 模式或 Rx 模式时自动到达。SPI 和 UART 的此命令的数据传输相同。

表 11-19: SetFs 数据传输

字节	0
来自主机的数据	操作码= 0xC1

11.5.4 SetTx

命令 *SetTx ()* 将设备设置为发送模式。

在使用此命令之前，应清除 IRQ 状态，请参见第 82 页上的第 11.8.3 节“ClearIrqStatus”。

表 11-20: SetTx SPI 数据传输

字节	0	1	2	3
来自主机的数据	操作码= 0x83	periodBase	periodBaseCount [15: 8]	periodBaseCount [7: 0]

表 11-21: SetTx U. ART 数据传输

字节	0	1	2	3	4
来自主机的数据	操作码= 0x83	0x03	periodBase	periodBaseCount [15: 8]	periodBaseCount [7: 0]

从 STDBY_RC 模式开始，振荡器接通，然后是 *PLL*，然后 *PA*（功率放大器）接通，*PA* 调节器根据 *SetTxParam ()* 命令定义的加速时间开始斜坡上升。一旦完成加速，分组处理就开始分组传输。一旦发送了数据包的最后一位，*PA* 调

节器就会降低，PA 关闭，收发器返回 STDBY_RC 模式并生成 IRQ TxDone。如果未生成 TxDone IRQ，则触发 TIMEOUT IRQ。在 TIMEOUT IRQ 或 TxDone IRQ 之后，收发器返回 STDBY_RC 模式。

超时持续时间由公式计算：

超 时 持 续 时 间 = periodBase * periodBaseCount 其中

periodBase 是下一个表中定义的 RTC 的步骤。

表 11-22: SetTx 超时定义。

periodBase	超时步骤
为 0x00	15.625μs
0x01	62.5μs
0x02	1 毫秒
x03	4 毫秒

periodBaseCount 是一个 16 位参数，用于定义超时期间使用的步骤数，如下表所示：

表 11-23: SetTx 超时持续时间

periodBaseCount [15: 0]	超时持续时间
为 0x0000	没有超时，Tx 单模式，设备将保持在 Tx 模式，直到数据包被传输，并在完成后以 STDBY_RC 模式返回。
其他	超时有效，器件保持 Tx 模式，在定时器计数结束时或发送数据包时自动返回 STDBY_RC 模式。

11.5.5 SetRx

命令 SetRx () 将设备设置为接收器模式。

在使用此命令之前，应清除 IRQ 状态，请参见第 82 页上的第 11.8.3 节“ClearIrqStatus”。

表 11-24: SetRx SPI 数据传输

字节	0	1	2	3
来自主机的数据	操作码= 0x82	periodBase	periodBaseCount [15: 8]	periodBaseCount [7: 0]

表 11-25: SetRx UART 数据传输

字节	0	1	2	3	4
主机	Opcode = 0x82	0x03 的数据	periodBase	periodBaseCount [15: 8]	periodBaseCount [7: 0]

此命令将收发器设置为 Rx 模式，等待接收一个或多个数据包。接收器模式以超时运行，为最终用户提供最大的灵活性。超时持续时间的参数是：

超 时 持 续 时 间 = periodBase * periodBaseCount 其 中

periodBase 是 RTC 的步骤，如表 11-22 中所定义。periodBaseCount 是超时期间使用的步骤

数，如下表中所定义：

表 11-26：SetRx 超时持续时间

TickNum (15:0)	超时持续时间
为 0x0000	没有超时。Rx 单模式。器件将保持 Rx 模式，直到发生接收，器件完成后返回 STDBY_RC 模式
为 0xFFFF	Rx 连续模式。设备保持 Rx 模式，直到主机发送命令更改操作模式。设备可以接收多个数据包。每次接收到分组时，向主机提供“分组接收”指示，并且设备将继续搜索新分组。
其他	超时有效。器件保持 Rx 模式，在定时器计数结束时或收到数据包时自动返回 STDBY_RC 模式。一旦检测到数据包，就会自动禁用定时器以允许完全接收数据包。

11.5.6 SetRxDutyCycle

此命令将收发器设置为嗅探模式，以便它定期查找新数据包（占空比操作）。

表 11-27：占空比操作 SPI 数据传输

字节	0	1	2	3	五	6
来自主机的数据	操作码= 0x94 的	PeriodBase	rxPeriodBase 计数 [15: 8]	rxPeriodBase 算[7: 0]	sleepPeriodBase 计数 [15: 8]	sleepPeriodBase 计数 [7: 0]

表 11-28：占空比操作 UART 数据传输

字节	0	1	2	3	4	6	7
来自主机的数据	操作码= 0x94	0x05	PeriodBase	rxPeriodBase 计数 [15: 8]	rxPeriodBase 计数[7: 0]	sleepPeriodBase 计数 [15: 8]	sleepPeriodBase 计数 [7: 0]

在 STDBY_RC 模式下发送此命令后，上下文（Rx 配置）将保存到数据 RAM 中，并且收发器将启动由以下步骤定义的循环：

- 输入 Rx 并侦听由 PeriodBase 和 rxPeriodBaseCount 定义的一段时间的数据包。

- 收发器查找由 0101 组成的前导码....如果检测到前导码，收发器将查找同步字和有效负载。
- 如果在 Rx 窗口期间没有收到数据包，收发器将进入休眠模式（保存上下文）一段时间，由 PeriodBase 和 sleepPeriodBaseCount 定义。
- 在休眠窗口结束时，收发器将退出休眠模式，恢复上下文并在上下文恢复后进入 Rx 模式，并在 Rx 窗口期间侦听数据包，依此类推。在任何时候，主机都可以停止该过程。如果有的话，循环终止
- 在 Rx 窗口期间检测到数据包，收发器通过 RxDone 标志中断主机并返回 STDBY_RC 模式。
- 主机在 Rx 窗口期间发出 SetStandby () 命令（在睡眠期间，窗口设备无法接收命令）。

请注意，要使用 RxDone 中断，必须在进入 Duty 循环操作之前启用相应的 IRQ。要启用 RxDone IRQ，请参阅命令 SetDioIrqParams () 在第 11.8.1“SetDioIrqParams”第 81 页。

睡眠模式持续时间定义如下：

$$\text{睡眠持续时间} = \text{PeriodBase} * \text{sleepPeriodBaseCount}$$

Rx 模式持续时间由。定义

$$\text{RX 持续时间} = \text{PeriodBase} * \text{rxPeriodBaseCount}$$

PeriodBase 被定义为 periodBase 在 表 11-22 。

rxPeriodBaseCount 和 sleepPeriodBaseCount 是 16 位参数，用于定义用于定义 Rx 持续时间和睡眠持续时间的步数。表 11-29 给出了 rxPeriodBaseCount 的一些特定值。

表 11-29：Rx 持续时间定义。

rxPeriodBaseCount [15： 0]	超时持续时间
为 0x0000	收发器等待直到找到数据包。一旦找到，收发器在向主机发送 RxDone IRQ 后进入 STDBY_RC 模式
其他	当器件返回休眠模式时，器件将保持 Rx 模式直到定时器结束 睡眠时间

注意！ 必须在 SetRxDutyCycle 之前发出命令 SetLongPreamble。

11.5.7 SetLongPreamble

命令（操作码 0x98）将收发器设置为长前导码模式，并且只能与 LoRa 模式和 GFSK 模式一起使用。在此模式下，命令 SetTx， SetRx 和 SetRxDutyCycle 的行为 被修改为：

- 仅在 GFSK 中， SetTx 参数不再定义超时，而是定义 GFSK 数据包的前导码部分的时间。因此，在 GFSK 模式下不会产生 TxTimeout 中断。在 LoRa 中， SetTx 行为不会更改。
- 仅在使用 LongPreamble 模式的 GFSK 中，前导码检测模式被激活。然后，命令 SetRx 可以为前导码检测生成中断。

- 在 GFSK 和 LORA，行为 *RxDutyCycle* 被修改，使得如果检测到前同步码时，Rx 窗口由 $SleepPeriod + 2 * RxPeriod$ 延长。

表 11-30: SetLongPreamble 数据传输

字节	0	1
来自主机的数据	操作码= 0x9B	启用

11.5.8 SetCAD

命令 *SetCAD* () (通道活动检测) 只能在 LoRa 包类型中使用。通道活动检测是 LoRa 特定的操作模式，其中设备搜索 LoRa 信号。搜索完成后，设备返回 STDBY_RC 模式。搜索的长度是通过 *SetCadParams* () 命令配置的。

在搜索周期结束时，设备始终发送 CadDone IRQ。如果检测到有效信号，它还会生成 CadDetected IRQ。

这种操作模式在需要 Listen before Talk 的所有应用程序中特别有用。

UART 数据传输和 SPI 数据传输是相同的。

表 11-31: SetCAD 数据传输

字节	0
来自主机的数据	操作码= 0xC5

11.5.9 SetTxContinuousWave

命令 *SetTxContinuousWave* () 是一个测试命令，用于以选定的频率和输出功率生成连续波 (RF 音)。设备保持 Tx 连续波，直到主机发送模式配置命令。此命令适用于所有数据包类型。

表 11-32: SetTxContinuousWave 数据传输

字节	0
来自主机的数据	操作码= 0xD1

UART 数据传输和 SPI 数据传输是相同的。

11.5.10 SetTxContinuousPreamble

命令 *SetTxContinuousPreamble* () 是一个测试命令，用于在 GFSK，BLE 或 FLRC 调制中生成交替的'0'和'1'以及 LoRa 中的符号 0 的无限序列。设备保持 Tx 连续波，直到主机发送模式配置命令。

UART 数据传输和 SPI 数据传输是相同的。

表 11-33: SetTxContinuousPreamble 数据传输

字节	0
----	---

来自主机的数据

操作码= 0xD2

11.5.11 SetAutoTx

BLE 要求收发器能够在数据包接收后 125μs 发回响应。这是通过发送命令 *SetAutoTx ()* 来执行的，该命令允许收发器在数据包接收结束后的用户可编程时间（时间）发送数据包。必须以 STDBY_RC 模式发出 *SetAutoTx ()*。*SetAutoTx ()* 的数据传输在表 11-34 中描述。

表 11-34：SetAutoTx SPI 数据传输

字节	0	1	2
来自主机的数据	操作码= 0x98	时间[15: 8]	时间[7: 0]

表 11-35：SetAutoTx UART 数据传输

字节	0	1	2	3
来自主机的数据	操作码= 0x98	0x02	时间[15: 8]	时间[7: 0]

时间以μs 表示。数据包接收结束与下一个数据包传输开始之间的延迟定义如下：

$$TxDelay = \text{时间} + \text{偏移}$$

此处偏移是收发器切换模式所需的时间，等于 33μs。

一旦发出此命令，每次收发器进入 Rx 模式时，它将自动切换到 Tx 并在预定义的时间 TxDelay 中发送数据包。要进行正常操作（在 Rx 之后进入 STDBY_RC），需要使用带有 0x00 的命令 *SetAutoTx* 作为时间参数。

11.5.12 SetAutoFs

此功能修改芯片行为，使 Rx 或 Tx 操作后的状态为 FS 而不是 STDBY（参见第 57 页第 10.7 节“收发器电路模式图形说明”）。此功能用于减少连续 Rx 和/或 Tx 操作之间的切换时间（请参见表 10-2：所有可能转换的切换时间（TswMode））。

- 要激活 AutoF 功能，请使用带参数 *true* 的命令 *SetAutoFs*
- 要取消激活 AutoF 功能，请使用命令 *SetAutoFs* with *false*。

表 11-36：SetAutoFs SPI 数据传输

字节	0	1
来自主机的数据	操作码= 0x9E	启用= 0x01, 禁用= 0x00

表 11-37：SetAutoFs UART 数据传输

字节	0	1	2
----	---	---	---

来自主机的数据	操作码= 0x9E	为 0x1	启用
---------	-----------	-------	----

11.6 无线电配置

11.6.1 SetPacketType

命令 *SetPacketType* () 从 6 种不同的数据包类型中选择设置收发器无线电帧。尽管其中一些使用相同的物理调制解调器，但它们并不都共享相同的参数。

注意！ 命令 *SetPacketType* () 必须是无线电配置序列的第一个。

表 11-38: SetPacketType SPI 数据传输

字节	0	1
来自主机的数据	操作码= 0x8A	packetType

表 11-39: SetPacketType UART 数据传输

字节	0	1	2
来自主机的数据	操作码= 0x8A	0x01	packetType

该命令的参数在表 11-40 中定义。

表 11-40: PacketType 定义

packetType	值	调制解调器操作模式
PACKET_TYPE_GFSK	0x00 [默认]	GFSK 模式
PACKET_TYPE_LORA	0x01	LoRa 模式
PACKET_TYPE_RANGING	0x02	测距发动机模式
PACKET_TYPE_FLRC	0x03	FLRC 模式
PACKET_TYPE_BLE	0x04	BLE 模式
保留的	> = 5	保留的

通过发送 *SetPacketType* () 命令来执行从一种操作模式到另一种操作模式的转换。来自先前模式的参数不在内部保留。从一帧到另一帧的切换必须在 STD BY_RC 模式下执行。

11.6.2 GetPacketType

命令 *GetPacketType* () 返回无线电的当前操作数据包类型。

表 11-41: GetPacketType SPI 数据传输

字节	0	1	2
----	---	---	---

来自主机的数据	操作码= 0x03	NOP	NOP
要托管的数据	状态	状态	packetType

表 11-42: GetPacketType UART 数据传输

字节	0	1	2
来自主机的数据	操作码= 0x03	0x01	-
要托管的数据	-	-	packetType

11.6.3 SetRfFrequency

命令 *SetRfFrequency* () 用于设置 RF 频率模式的频率。

表 11-43: SetRfFrequency SPI 数据传输

字节	0	1	2	3
来自主机的数据	操作码= 0x86	rfFrequency [23:16]	rfFrequency [15: 8]	rfFrequency [7: 0]

表 11-44: SetRfFrequency UART 数据传输

字节	0	1	2	3	4
来自主机的数据	操作码= 0x86	×03	rfFrequency [23:16]	rfFrequency [15: 8]	rfFrequency [7: 0]

的 LSB 的 rfFrequency 等于 PLL 步骤即 $52e6 / 2^{18}$ 赫兹，其中 52e6 是以 Hz 为晶体 frequency。*SetRfFrequency* () 定义 Tx 频率。IF 降低 Rx 频率。IF 默认设置为 1.3 MHz。

11.6.4 SetTxParams

该命令将使用参数 TX 输出功率的功率和使用的 Tx 斜坡时间参数 rampTime。此命令可用于所有 packetType。

表 11-45: SetTxParams SPI 数据传输

字节	0	1	2
来自主机的数据	操作码= 0x8E	功率	rampTime

表 11-46: SetTxParams UART 数据传输

字节	0	1	2	3
来自主机的数据	操作码= 0x8E	0x02	功率	rampTime

输出功率 (Pout) 由参数功率定义。

$$P = -18 + \text{功率 dBm 输出}$$

$P_{\text{outMax}} = -18 \text{ dBm}$ (功率 = 0) P

$\text{outMax} = 13 \text{ dBm}$ (功率 = 31)

根据表 11-47，使用 rampTime 参数定义所需的功率放大器斜坡时间。

表 11-47: RampTime 定义

rampTime	值	斜坡时间 (μs)
RADIO_RAMP_02_US	为 0x00	2
RADIO_RAMP_04_US	为 0x20	4
RADIO_RAMP_06_US	0x40 的	6
RADIO_RAMP_08_US	0x60	8
RADIO_RAMP_10_US	0x80 的	10
RADIO_RAMP_12_US	0xA0	12
RADIO_RAMP_16_US	将 0xC0	16
RADIO_RAMP_20_US	取 0xE0	20

11.6.5 SetCadParams

命令 *SetCadParams* () 定义检测到通道活动 (CAD) 的符号数。

表 11-48: CAD SPI 数据传输

字节	0	1
来自主机的数据	操作码 = 0x88	cadSymbolNum

表 11-49: CAD UART 数据传输

字节	0	1	2
来自主机的数据	操作码 = 0x88	0x01	cadSymbolNum

要使用的符号数量在下表中定义。

表 11-50: CadSymbolNum 定义

cadSymbolNum	值	用于 CAD 的符号数
LORA_CAD_01_SYMBOL	为 0x00	1
LORA_CAD_02_SYMBOLS	为 0x20	2
LORA_CAD_04_SYMBOLS	0x40 的	4

LORA_CAD_08_SYMBOLS	0X60	8
LORA_CAD_16_SYMBOLS	0x80 的	16

注意：对于符号 1 和 2，错误检测的风险较高。

11.6.6 SetBufferBaseAddress

此命令修复所有数据包类型的 Tx 和 Rx 模式下数据包处理操作的基址。

表 11-51：SetBufferBaseAddress SPI 数据传输

字节	0	1	2
来自主机的数据	操作码= 0x8F	txBaseAddress	rxBaseAddress

表 11-52：SetBufferBaseAd 连衣裙 UART 数据传输

字节	0	1	2	3
来自主机的数据	操作码= 0x8F	0x02	txBaseAddress	rxBaseAddress

11.6.7 SetModulationParams

命令 *SetModulationParams* () 用于配置无线电的调制参数。此函数传递的参数将根据帧类型进行解释，在调用此函数之前应将其设置为所需类型。

表 11-53：SetModulationParams SPI 数据传输

字节	0	1	2	3
来自主机的数据	操作码= 0x8B	PARAM [0]	PARAM [1]	PARAM [2]

表 11-54：SetModulationParams UART 数据传输

字节	0	1	2	3	4
主机	Opcode = 0x8B 的数据	×03	PARAM [0]	PARAM [1]	PARAM [2]

在 GFSK，FLRC 和 BLE 中，比特率和带宽由 param [0] 参数定义为一对值，请参见第 85 页的表 13-1：“GFSK 模式下的调制参数”。调制指数与比特率结合使用，以计算用于发送或接收的频率偏差。调制索引由 param [1] 参数定义。的 BT 表示高斯滤波器可被用来过滤在发射机侧的调制码流。BT 由 param [2] 参数定义。参数的含义取决于所选的数据包类型，并将在专用于所选数据包类型的章节中定义。

在 LoRa 分组类型中，SF 对应于用于 LoRa 调制的扩频因子。SF 由 param [0] 参数定义。该 BW 对应于在其上 LORA 信号被扩展的带宽。LoRa 中的 BW 由 param [1] 参数定义。

LoRa 有效载荷配有前向纠错机制，该机制具有多级编码。编码率（CR）由 LoRa 中的 param [2] 参数定义。

SetModulationParams () 参数的定义总结在下表中：

表 11-55：SetModulationParams 参数定义

参数	BLE 和 GFSK	FLRC	LoRa 和测距引擎
modParam1	BitrateBandwidth	BitrateBandwidth	SpreadingFactor
modParam2	ModulationIndex	CodingRate	带宽
modParam3	ModulationShaping	ModulationShaping	CodingRate

11.6.8 SetPacketParams

该命令用于设置数据包处理块的参数。

表 11-56：SetPacketParams SPI 数据传输

字节	0	1	2	3	4	五	6	7
来自主机的数据	操作码= 0x8C	SetPacketParam1	SetPacketParam2	SetPacketParam3	SetPacketParam4	SetPacketParam5	SetPacketParam6	SetPacketParam7

表 11-57：SetPacketParams UART 数据传输

字节	0	1	2	3	4	五	6	7	8
来自主机的数据	操作码= 0x8C	0x07	SetPacketParam1	SetPacketParam2	SetPacketParam3	SetPacketParam4	SetPacketParam5	SetPacketParam6	SetPacketParam7

数据包参数含义取决于所选的数据包类型。表 11-58 给出了根据数据包类型的参数含义。

表 11-58：SetPacketParams 参数定义。

参数	GFSK 和 FLRC	BLE	LoRa 和测距引擎
SetPacketParam1	前导长度	ConnectionState	前导长度
SetPacketParam2	SyncWordLength	CrcLength	HeaderType
SetPacketParam3	SyncWordMatch	BleTestPayload	PayloadLength
SetPacketParam4	HeaderType	美白	CRC
SetPacketParam5	PayloadLength	不曾用过	InvertIQ /啁啾反转
SetPacketParam6	CrcLength	不曾用过	不曾用过
SetPacketParam7	美白	不曾用过	不曾用过

这些参数的用法和定义在不同的数据包类型部分中描述。

11.7 通信状态信息

这些命令返回有关收发器状态，接收的数据包长度，接收功率以及指示数据包是否已被正确接收的若干标志的信息。返回的参数对于除 LoRa 之外的所有帧都是通用的。

11.7.1 GetRxBufferStatus

此命令返回最后接收的数据包的长度（payloadLengthRx）和接收的第一个字节的地址（rxBufferOffset），它适用于所有调制解调器。地址是相对于数据缓冲区的第一个字节的偏移量。

表 11-59：GetRxBufferStatus SPI 数据传输

字节	0	1	2	3
来自主机的数据	操作码= 0x17	NOP	NOP	NOP
要托管的数据	状态	状态	rxPayloadLength	rxStartBufferPointer

表 11-60：GetRxBufferStatus UART 数据传输

字节	0	1	2	3
来自主机的数据	操作码= 0x17	0x02	-	-
要托管的数据	-	-	rxPayloadLength	rxStartBufferPointer

请注意，在具有固定标头的 LoRa 数据包类型中（请参见第 44 页上的第 7.4.3 节“隐式（固定长度）标头模式”），GetRxBufferStatus 始终为 rxPayloadLength 返回 0x00。实际上，在这种配置中，数据包中不存在标头，因此无法从中提取有效负载大小。但是，可以通过直接寄存器读取来恢复无线电中配置的有效载荷大小。因此，读取寄存器 0x901 将返回有效负载大小。有关寄存器读取的数据传输，请参见第 62 页上的第 11.3.2 节“ReadRegister 命令”。

11.7.2 GetPacketStatus

此命令用于检索有关上次接收的数据包的信息。返回的参数取决于帧。

表 11-61：GetPacketStatus SPI 数据传输

字节	0	1	2	3	4	五	6
来自主机的数据	操作码= 0x1D	NOP	NOP	NOP	NOP	NOP	NOP
要托管的数据	状态	状态	packetStatus [7: 0]	packetStatus [15: 8]	packetStatus [23:16]	packetStatus [31:24]	packetStatus [39:32]

表 11-62：GetPacketStatus UART 数据传输

字节	0	1	2	3	4	五	6
----	---	---	---	---	---	---	---

来自主机的数据	操作码= 0x1D	0x05	-	-	-	-	-
要托管的数据	-	-	packetStatus [0: 7]	packetStatus [8:15]	packetStatus [16:23]	packetStatus [24:31]	packetStatus [32:39]

`GetPacketStatus ()` 命令返回的值取决于数据包类型，并在表 11-63 中进行了总结。

在 LoRa 和/或测距引擎的情况下，命令只返回 2 个字节。

表 11-63: packetStatus 定义

参数	BLE, GFSK, FLRC	LoRa 和测距引擎
packetStatus [7: 0]	RFU	rssiSync
packetStatus [15: 8]	rssiSync	SNR
packetStatus [16 点 23 分]	错误	-
packetStatus [24:31]	状态	-
packetStatus [32:39]	同步	-

rssiSync 和 snr 的含义在下表中给出。请注意，snr 仅在 LoRa 和 Ranging Engine 数据包类型中可用。

表 11-64: RSSI 和 SNR 数据包状态

值	描述
rssiSync	RSSI 值在检测到同步地址时锁存。 实际的信号功率- rssiSync 2 (DBM)
SNR	收到的最后一个数据包的 SNR 估计。在两个恭维格式中乘以 4。 实际 SNR 是 SNR 4 (dB) 的

表 11-65: 状态包状态字节

PStatus3	符号	描述
第 7: 6	保留的	保留的
第 5 位	rxNoAck	收到的数据包的 NO_ACK 字段。仅适用于动态长度数据包的 Rx。
第 4: 1	保留的	保留的
位 0	pktSent	表示数据包传输已完成。不表示数据包有效性。 仅适用于 Tx。

表 11-66: 错误包状态字节

错误	符号	描述
----	----	----

第 7 位	保留的	保留的
第 6 位	SyncError	当前数据包的同步地址检测状态 仅在启用同步地址检测时适用于 Rx。
第 5 位	LengthError	当接收数据包的长度大于 PAYLOAD_LENGTH 参数中定义的最大长度时置位。 仅适用于动态长度数据包的 Rx。
第 4 位	CrcError	CRC 检查当前数据包的状态。无论如何，数据包在 FIFO 中都可用。 仅在启用 CRC 校验时适用于 Rx
第 3 位	AbortError	中止状态指示 Rx / Tx 中的当前数据包是否已中止。 适用于 Rx 和 Tx。
第 2 位	headerReceived	指示是否已收到当前数据包的标头。 仅适用于动态长度数据包的 Rx
第 1 位	packetReceived	表示数据包接收已完成。不表示数据包有效性。 仅适用于 Rx。
位 0	packetCtrlBusy	表示数据包控制器正忙。适用于 Rx / Tx

表 11-67：同步数据包状态字节

同步	符号	描述
第 7：3 点	保留的	保留的
第 2：0 位	syncAdrsCode	检测到同步地址的代码 000：同步地址检测错误 001：检测到 sync_adrs_1' 010：sync_adrs_2'，检测到 100：检测到 sync_adrs_3'

11.7.3 GetRssiInst

该命令在接收数据包期间返回瞬时 RSSI 值。该命令对所有帧都有效。在 LoRa 操作中，瞬时 RSSI 在接收到的每个符号处更新。

表 11-68：GetRssiInst SPI 数据传输

字节	0	1	2
来自主机的数据	操作码= 0x1F	NOP	NOP
要托管的数据	状态	状态	rssiInst

表 11-69：GetRssiInst UART 数据传输

字节	0	1	2
来自主机的数据	操作码= 0x1F	0x01	-

要托管的数据	-	-	rssilnst
--------	---	---	----------

表 11-70: Rssilnst 定义

参数	描述
rssilnst	信号功率是 - rssilnst 2 (DBM)

11.8 IRQ 处理

总共有 16 种可能的中断源，具体取决于所选的帧和收发器模式。它们中的每一个都可以启用或屏蔽。此外，它们中的每一个都可以映射到 DIO1, DIO2 或 DIO3。

表 11-71: IRQ 寄存器

位	IRQ	描述	包
0	TxDone	Tx 完成	所有
1	RxDone	Rx 完成	所有
2	SyncWordValid	同步。字有效	GFSK / BLE / FLRC
3	SyncWordError	同步。字错误	FLRC
4	HeaderValid	标题有效	LoRa /测距引擎
五	HeaderError	标题错误	LoRa /测距引擎
6	CrcError	CRC 错误	GFSK / BLE / FLRC / LoRa
7	RangingSlaveResponseDone	测距响应完成 (Slave)	测距发动机
8	RangingSlaveRequestDiscard	丢弃测距请求 (Slave)	LoRa /测距引擎
9	RangingMasterResultValid	测距结果有效 (主)	测距发动机

表 11-71: IRQ 寄存器

位	IRQ	描述	包
10	RangingMasterTimeout	测距超时 (主)	测距发动机
11	RangingMasterRequestValid	测距请求有效 (从属)	测距发动机
12	CadDone	渠道活动检查完成	LoRa /测距引擎
13	CadDetected	检测到渠道活动	LoRa /测距引擎
14	RxTxTimeout	Rx 或 Tx 超时	所有
15	PreambleDetected	检测到前导码	全部如果已激活 SetLongPreamble

一个名为 IRQ_reg 的专用 16 位寄存器用于记录 IRQ 源。每个位置对应一个 IRQ 源，如上表所述。一组用户命令用于配置 IRQ 掩码，DIO 映射和 IRQ 清除，如下段所述。

11.8.1 SetDioIrqParams

此命令用于启用 IRQ 并将 IRQ 连接到 DIO 引脚。

表 11-72: IRQ 掩码定义 SPI 数据传输

字节	0	1	2	3	4	五	6	7	8
来自主机的数据	OpCode = 0x8D	irqMask [15: 8]	IRQMASK [7: 0]	dio1Mask [15: 8]	dio1Mask [7: 0]	dio2Mask [15: 8]	dio2Mask [7: 0]	dio3Mask [15: 8]	dio3Mask [7: 0]

表 11-73: IRQ 掩码定义 UART 数据传输

字节	0	1	2	3	4	五	6	7	8	9
来自主机的数据	操作码 = 0x8D	0x08 的	IRQMASK [15: 8]	IRQMASK [7: 0]	dio1Mask [15: 8]	dio1Mask [7: 0]	dio2Mask [15: 8]	dio2Mask [7: 0]	dio3Mask [15: 8]	dio3Mask [7: 0]

如果标志寄存器中的相应位置 1，则在 IRQ 寄存器中标记中断。例如，只有当 IrqMask 的位 0 设置为 1 时，TxDone 才能将 IRQ 寄存器的第 0 位置 1。

如果 dioXMask 和 irqMask 中的相应位置 1，则中断会导致 DIO 置 1。例如，如果 irqMask 的第 0 位设置为 1 并且 dio1Mask 的第 0 位设置为 1，则 IRQ 源 TxDone 的上升沿将记录在 IRQ 寄存器中，并将同时出现在 DIO1 上。

一个 IRQ 可以映射到所有 DIO，一个 DIO 可以映射到所有 IRQ（执行 OR 操作），但某些 IRQ 源仅在某些操作模式和帧类型上可用。

11.8.2 GetIrqStatus

该命令返回 IRQ 寄存器的值。

表 11-74: GetIrqStatus SPI 数据传输

字节	0	1	2	3
来自主机的数据	操作码= 0x15	NOP	NOP	NOP
要托管的数据	状态	状态	irqStatus [15: 8]	irqStatus [7: 0]

表 11-75: GetIrqStatus UART 数据传输

字节	0	1	2	3
来自主机的数据	操作码= 0x15	0x02	-	-
要托管的数据	-	-	irqStatus [15: 8]	irqStatus [7: 0]

11.8.3 ClearIrqStatus

该命令清除 IRQ 寄存器中的 IRQ 标志。

表 11-76: ClearIrqStatus SPI 数据传输

字节	0	1	2
来自主机的数据	操作码= 0x97	IRQMASK [15: 8]	IRQMASK [7: 0]

表 11-77: ClearIrqStatus UART 数据传输

字节	0	1	2	3
来自主机的数据	操作码= 0x97	0x02	IRQMASK [15: 8]	IRQMASK [7: 0]

要清除 IRQ 寄存器中的 IRQ 标志，应将 irqMask 的位设置为 1，该位 对应于要清除的 IRQ 标志的相同位置。例如，如果 irqMask 的位 0 设置为 1，则清除 IRQ 寄存器的位 0 的 IRQ 标志。

如果 DIO 映射到单个 IRQ 源，则如果 IRQ 寄存器中的相应位清零，则 DIO 清零。如果 DIO 与多个 IRQ 源进行“或”运算，则 DIO 保持设置为 1，直到映射到 IRQ 寄存器中的 DIO 的所有位都清零。

12.命令列表

下表给出了命令列表和相应的操作码。

表 12-1: 收发器可用命令

命令	操作码	参数	返回
的 getStatus	将 0xC0	-	状态
WriteRegister	为 0x18	地址[15: 8], 地址[7: 0], 数据[0: n]	-
ReadRegister	0x19	地址[15: 8], 地址[7: 0]	数据[0: N-1]
WriteBuffer	0x1A 的	偏移量, 数据[0: N]	-
ReadBuffer	0x1B	抵消	数据[0: N-1]
SetSleep	的 0x84	sleepConfig	-
SetStandby	0x80 的	standbyConfig	-
SetFs	0xC1	-	-
SetTx	0x83	periodBase, periodBaseCount [15: 8], periodBaseCount [7: 0]	-
SetRx	为 0x82	periodBase, periodBaseCount [15: 8], periodBaseCount [7: 0]	-
SetRxDutyCycle	0x94 的	rxPeriodBase, rxPeriodBaseCount [15: 8], rxPeriodBaseCount [7: 0], sleepPeriodBase , sleepPeriodBaseCount [15: 8], sleepPeriodBaseCount [7: 0]	-

SetCad	0xC5	-	-
SetTxContinuousWave	0xD1	-	-
SetTxContinuousPreamble	0xD2	-	-
SetPacketType	0x8A	packetType	-
GetPacketType	×03	-	packetType
SetRfFrequency	0x86 可以	rfFrequency [23:16], rfFrequency [15: 8], rfFrequency [7: 0]	-
SetTxParams	为 0x8E	功率, rampTime	-
SetCadParams	均为 0x88	cadSymbolNum	-
SetBufferBaseAddress	值为 0x8F	txBaseAddress, rxBaseAddress	-
SetModulationParams	0x8B	modParam1, modParam2, modParam3	-

表 12-1: 收发器可用命令

命令	操作码	参数	返回
SetPacketParams	0x8C	packetParam1, packetParam2, packetParam3, packetParam4, packetParam5, packetParam6, packetParam7	-
GetRxBufferStatus	0×17	-	payloadLength, rxBufferOffset
GetPacketStatus	0x1D	-	packetStatus [39:32], packetStatus [31:24], packetStatus [23:16], packetStatus [15: 8], packetStatus [7: 0]
GetRssiInst	为 0x1F	-	rssiInst
SetDioIrqParams	0x8D	irqMask [15: 8], irqMask [7: 0], dio1Mask [15: 8], dio1Mask [7: 0], dio2Mask [15: 8], dio2Mask [7: 0], dio3Mask [15: 8], dio3Mask [7: 0]	-
GetIrqStatus	为 0x15	-	irqStatus [15: 8], irqStatus [7: 0]
ClrIrqStatus	0x97	irqMask [15: 8], irqMask [7: 0]	-
SetRegulatorMode	0x96	regulatorMode	-
SetSaveContext	0xD5	-	-
SetAutoFS	0x9E	0x00: 禁用或 0x01: 启用	-

SetAutoTx	0x98 在全局	时间	-
SetPerfCounterMode	为 0x9c	perfCounterMode	-
SetLongPreamble	0x9B	启用	-
SetUartSpeed	0x9D	uartSpeed	-
SetRangingRole	0xA3 执行	0x00 =从机或 0x01 =主机	-

13.收发器操作

13.1 GFSK 数据包

13.1.1 通用收发器设置

上电或硬复位后，收发器运行一个简短的校准程序，然后进入 STDBY_RC 模式，由 BUSY 引脚上的低电平状态指示。从此状态开始或接收 GFSK 格式 FSK 数据包所需的步骤（顺序很重要）如下所示：

1. 如果不处于 STDBY_RC 模式，则通过发送命令进入此模式：

SetStandby (STDBY_RC)

2. 通过发送命令定义 GFSK 数据包：

SetPacketType (PACKET_GFSK)

3. 通过发送命令定义 RF 频率：

SetRfFrequency (rfFrequency)

的 LSB 的 rfFrequency 等于 PLL 步骤即 $52e6 / 2^{18}$ 赫兹。SetRfFrequency () 定义 Tx 频率。

在 Rx 中，频率降低中频 (IF)。IF 默认设置为 1.3 MHz。

4. 通过发送命令，指示数据包处理程序将读取的地址 (Tx 中的 txBaseAddress) 或写入 (Rx 中的 rxBaseAddress) 数据有效负载的第一个字节：

SetBufferBaseAddress (txBaseAddress, rxBaseAddress)

请注意，txBaseAddress 和 rxBaseAddress 相对于数据存储器映射的开头是偏移的。

5. 通过发送命令定义调制参数

SetModulationParams (modParam1, modParam2, modParam3)

比特率和带宽通过 modParam1 设置进行配置。

表 13-1: GFSK 模式下的调制参数

参数	符号	值	比特率 (Mb / s)	带宽 (MHz DSB)
modParam1	GFSK_BLE_BR_2_000_BW_2_4	0x04	2	2.4
modParam1	GFSK_BLE_BR_1_600_BW_2_4	0x28	1.6	2.4
modParam1	GFSK_BLE_BR_1_000_BW_2_4	0x4C	1	2.4
modParam1	GFSK_BLE_BR_1_000_BW_1_2	0x45	1	1.2
modParam1	GFSK_BLE_BR_0_800_BW_2_4	0x70	0.8	2.4

表 13-1: GFSK 模式下的调制参数

参数	符号	值	比特率 (Mb / s)	带宽 (MHz DSB)
modParam1	GFSK_BLE_BR_0_800_BW_1_2	0x69	0.8	1.2
modParam1	GFSK_BLE_BR_0_500_BW_1_2	0x8D	0.5	1.2
modParam1	GFSK_BLE_BR_0_500_BW_0_6	0x86 可以	0.5	0.6
modParam1	GFSK_BLE_BR_0_400_BW_1_2	0xB1	0.4	1.2
modParam1	GFSK_BLE_BR_0_400_BW_0_6	和 0xAA	0.4	0.6
modParam1	GFSK_BLE_BR_0_250_BW_0_6	0xCE	0.25	0.6
modParam1	GFSK_BLE_BR_0_250_BW_0_3	0xC7	0.25	0.3
modParam1	GFSK_BLE_BR_0_125_BW_0_3	0xEF	0.125	0.3

表 13-2: GFSK 模式下的调制参数

参数	符号	值	Modindex
modParam2	MOD_IND_0_35	为 0x00	0.35
modParam2	MOD_IND_0_5	0x01	0.5
modParam2	MOD_IND_0_75	0x02	0.75
modParam2	MOD_IND_1_00	×03	1
modParam2	MOD_IND_1_25	0x04	1.25
modParam2	MOD_IND_1_50	0x05	1.5
modParam2	MOD_IND_1_75	0x06	1.75
modParam2	MOD_IND_2_00	0x07	2
modParam2	MOD_IND_2_25	0x08 的	2.25
modParam2	MOD_IND_2_50	×09	2.5

modParam2	MOD_IND_2_75	的 0x0A	2.75
modParam2	MOD_IND_3_00	0x0B 中	3
modParam2	MOD_IND_3_25	0x0C	3.25
modParam2	MOD_IND_3_50	0x0D	3.5
modParam2	MOD_IND_3_75	为 0x0E	3.75
modParam2	MOD_IND_4_00	为 0x0F	4

表 13-3: GFSK 模式下的调制参数

参数	符号	值	BT
modParam3	BT_OFF	为 0x00	没有过滤
modParam3	BT_1_0	为 0x10	1
modParam3	BT_0_5	为 0x20	0.5

6. 通过发送命令定义要使用的数据包设置:

SetPacketParams (param [0], param [1], param [2], param [3], param [4], param [5], param [6])

- packetParam1 = PreambleLength
- packetParam2 =定义用于同步字 (SyncWordLength) 的字节数。
- packetParam3 =定义 SyncWordMatch 使用的相关器数量
- packetParam4 = HeaderType
- packetParam5 = PayloadLength
- packetParam6 = CrcLength
- packetParam7 =美白

表 13-4: GFSK 数据包中的前导码长度定义

参数	符号	值	前导码长度, 以位为单位
packetParam1	PREAMBLE_LENGTH_04_BITS	为 0x00	4
packetParam1	PREAMBLE_LENGTH_08_BITS	为 0x10	8
packetParam1	PREAMBLE_LENGTH_12_BITS	为 0x20	12
packetParam1	PREAMBLE_LENGTH_16_BITS	的 0x30	16
packetParam1	PREAMBLE_LENGTH_20_BITS	0x40 的	20
packetParam1	PREAMBLE_LENGTH_24_BITS	为 0x50	24
packetParam1	PREAMBLE_LENGTH_28_BITS	0x60	28

packetParam1	PREAMBLE_LENGTH_32_BITS	0x70	32
--------------	-------------------------	------	----

使用 AGC 时的最小前导码长度对于 1 Mb / s 的比特率应为 8 比特。对于其他比特率，前导码比特的最小数量必须至少为 16 比特。

表 13-5: GFSK 数据包中的同步字长定义

参数	符号	值	以字节为单位同步字大小
packetParam2	SYNC_WORD_LEN_1_B	为 0x00	1
packetParam2	SYNC_WORD_LEN_2_B	0x02	2
packetParam2	SYNC_WORD_LEN_3_B	0x04	3
packetParam2	SYNC_WORD_LEN_4_B	0x06	4
packetParam2	SYNC_WORD_LEN_5_B	0x08 的	五

由于有 3 个相关器，收发器可以同时搜索多个同步字：

表 13-6: GFSK 数据包中的同步字组合

参数	符号	值	同步 Word 组合使用
packetParam3	RADIO_RX_MATCH_SYNCWORD_OFF	为 0x00	禁用同步字
packetParam3	RADIO_RX_MATCH_SYNCWORD_1	为 0x10	SyncWord1
packetParam3	RADIO_RX_MATCH_SYNCWORD_2	为 0x20	SyncWord2
packetParam3	RADIO_RX_MATCH_SYNCWORD_1_2	的 0x30	SyncWord1 或 SyncWord2
packetParam3	RADIO_RX_MATCH_SYNCWORD_3	0x40 的	SyncWord3
packetParam3	RADIO_RX_MATCH_SYNCWORD_1_3	为 0x50	SyncWord1 或 SyncWord3
packetParam3	RADIO_RX_MATCH_SYNCWORD_2_3	0x60	SyncWord2 或 SyncWord3
packetParam3	RADIO_RX_MATCH_SYNCWORD_1_2_3	0x70	SyncWord1, SyncWord2 或 SyncWord3

表 13-7: GFSK 数据包中的数据包类型定义

参数	符号	值	包长度模式
packetParam4	RADIO_PACKET_FIXED_LENGTH	为 0x00	固定长度模式
packetParam4	RADIO_PACKET_VARIABLE_LENGTH	为 0x20	可变长度模式

有效载荷长度由 param [4] 参数定义。Tx 中的数据包处理程序使用此参数来发送确切的字节数。在 Rx 可变长度模式下，数据包处理程序将过滤掉大小大于 Payloadlength 的所有数据包。

表 13-8: GFSK 数据包中的有效载荷长度定义

参数	符号	值	描述
----	----	---	----

packetParam5	PAYLOAD_LENGTH	[0 ... 255]	有效载荷长度，以字节为单
--------------	----------------	-------------	--------------

使用 GFSK 数据包，CRC 可以在 1 或 2 个字节上计算或忽略。这是使用参数 param [5] 定义的。**表 13-9：GFSK 数据包中的 CRC 定义**

参数	符号	值	CRC 类型
packetParam6	RADIO_CRC_OFF	为 0x00	没有 CRC
packetParam6	RADIO_CRC_1_BYTES	为 0x10	CRC 字段使用 1 个字节
packetParam6	RADIO_CRC_2_BYTES	为 0x20	CRC 字段使用 2 个字节

可以在参数参数[6]中启用白化。

表 13-10：GFSK 数据包中的白化启用

参数	符号	值	美白模式
packetParam7	WHITENING_ENABLE	为 0x00	白皙化
packetParam7	WHITENING_DISABLE	0x08 的	白皙的

7. 定义同步字值

另外，用户可以定义同步字的 32 位（SyncWord1，SyncWord2，SyncWord3）。这是通过发送 *WriteRegister ()* 命令来执行的，下一个表给出了同步字的地址。

表 13-11：GFSK 数据包中的同步字定义

同步 Word	字节	地址
SyncWord1	SyncWord1 (39:32)	0x09CE
	SyncWord1 (31:24)	0x09CF
	SyncWord1 (23:16)	0x09D0
	SyncWord1 (15: 8)	0x09D1
	SyncWord1 (7: 0)	0x09D2

表 13-11：GFSK 数据包中的同步字定义

同步 Word	字节	地址
SyncWord2	SyncWord2 (39:32)	0x09D3
	SyncWord2 (31:24)	0x09D4
	SyncWord2 (23:16)	0x09D5
	SyncWord2 (15: 8)	0x09D6
	SyncWord2 (7: 0)	0x09D7

SyncWord3	SyncWord3 (39:32)	0x09D8
	SyncWord3 (31:24)	0x09D9
	SyncWord3 (23:16)	0x09DA
	SyncWord3 (15: 8)	0x09DB
	SyncWord3 (7: 0)	0x09DC

用于 CRC 的种子也需要针对某些应用进行修改。这是通过使用命令 `WriteReg ()` 直接寄存器访问来执行的。

表 13-12: CRC 初始化寄存器

参数	字节	地址
CrcInit	CRC 初始值 MSB	0x9c8
	CRC 初始值 LSB	0x9c9

的 CRC 多项式，也可以通过使用命令直接寄存器访问改性 `WRITEREG ()` 。

表 13-13: CRC 多项式定义

参数	字节	地址	描述
CrcPolynomial	CRC 多项式 MSB	0x9C6	定义 16 位 CRC 多项式的 LSB 字节 或定义 8 位 CRC 多项式 例如，编程以下多项式 $P_{16}(x) = x^{16} + x^{12} + x^5 + 1$
	CRC 多项式 LSB	0x9C7	初始化 <code>crc_polynomial (15: 0) = 0x1021</code> 编程以下多项式 $P_8(x) = x^8 + x^2 + x + 1$ 初始化 <code>crc_polynomial (7: 0) = 0x07</code>

13.1.2 Tx 设置和操作

1. 通过发送命令定义输出功率和斜坡时间:

```
SetTxParams (电力, ramptime)
```

2. 通过发送命令将有效负载发送到数据缓冲区:

```
WriteBuffer (偏移量, *数据)
```

其中* data 是指向有效载荷的指针，offset 是有效载荷的第一个字节在 FIFO 中的地址。偏移量将对应于正常操作中的 txBaseAddress。

3. 使用命令配置 DIO 和中断源 (IRQ):

SetDioIrqParams (IrqMask, Dio1Mask, Dio2Mask, Dio3Mask)

在典型的 Tx 操作中，用户可以选择一个或多个 IRQ 源：

- TxDone IRQ 表示数据包传输结束。收发器将处于 STDBY_RC 模式。
- RxTxTimeout（可选）以防止死锁。如果发生超时，收发器将自动返回 STDBY_RC 模式。

4. 配置完成后，将收发器设置为发射器模式，使用命令开始传输：

SetTx (periodBase, periodBaseCount [15: 8], periodBaseCount [7: 0])

如果需要超时，请将 periodBaseCount 设置为非零值。此超时可用于避免死锁。

等待 IRQ TxDone 或 RxTxTimeout

一旦发送了数据包或发生了超时，收发器就会自动进入 STDBY_RC 模式。

5. (可选) 使用以下命令检查数据包状态以确保数据包已正确发送：

GetPacketStatus ()

在这种情况下，只有参数 packetStatus [3]才有用。

表 13-14: GFSK 数据包中的 PacketStatus [3]

PacketStatus [3]	符号	描述
第 7: 1 点	保留的	保留的
位 0	PktSent	表示数据包传输已完成。仅表示传输过程完成而不是数据包有效性。仅适用于 Tx。

6. 通过发送命令清除 TxDone 或 RxTxTimeout IRQ：

ClrIrqStatus (irqMask [15: 8], irqMask [7: 0])

此命令将重置 irqMask 中相应位位置设置为 1 的标志。

13.1.3 Rx 设置和操作

1. 使用命令配置 DIO 和中断源 (IRQ)

SetDioIrqParams (irqMask, dio1Mask, dio2Mask, dio3Mask)

在典型的 GFSK Rx 操作中，可以选择一个或多个 IRQ 源：

- RxDone 表示已检测到数据包。此 IRQ 并不意味着数据包有效（大小或 CRC 正确）。用户必须检查数据包状态以确保收到有效的打包。
- SyncWordValid 表示已检测到同步字。
- CrcError 表示收到的数据包有 CRC 错误

- RxTxTimeout 表示在 SetRx () 命令中由 timeout 参数定义的给定时间帧内未检测到数据包。

根据需要将这些 IRQ 映射到一个或多个 DIO。

2. 配置完成后，将收发器设置为接收器模式，使用命令开始接收：

SetRx (periodBase, periodBaseCount [15: 8], periodBaseCount [7: 0]) 根据 *periodBaseCount*，可能有 3 种可能的 Rx 行为：

- *periodBaseCount* 设置为 0，然后没有 Timeout，Rx Single 模式，设备将保持 Rx 模式，直到发生接收并且设备完成后返回 STDBY_RC 模式
- *periodBaseCount* 设置为 0xFFFF，Rx 连续模式，设备保持 Rx 模式，直到主机发送命令改变操作模式。设备可以接收多个数据包。每次接收到分组时，分组接收指示被提供给主机，并且设备将继续搜索新分组。
- *periodBaseCount* 设置为另一个值，然后 Timeout 处于活动状态。设备保持 Rx 模式；它会在计时器计数结束时或收到数据包时自动返回 STDBY_RC 模式。一旦检测到数据包，就会自动禁用定时器以允许完全接收数据包。

3. 在典型情况下，使用超时并等待 IRQs RxDone 或 RxTxTimeout。

如果 IRQs RxDone 上升，如果使用单模式（超时设置为不同于 0xFFFF 的值），收发器将进入 STDBY_RC 模式。如果使用连续模式（超时设置为 0xFFFF），收发器将保留在 Rx 中并继续侦听新数据包。

4. 使用以下命令检查数据包状态以确保已正确接收数据包：

GetPacketStatus ()

该命令返回以下参数：

- *RssiSync*：检测到同步字时的 RSSI 值。实际信号功率为 $-RssiSync / 2$ (dBm)
- *packetStatus2*：提供有关收到的最后一个数据包的信息，如下表所述
- *packetStatus3*：在 Tx 中用于指示传输结束
- *packetStatus4*：指示哪个相关器检测到同步字

表 13-15：GFSK 数据包中的 PacketStatus [2]

PacketStatus [2]	符号	描述
第 7 位	保留的	保留的
第 6 位	SyncError	同步当前数据包的地址检测状态仅在启用同步地址检测时适用于 Rx。
第 5 位	LengthError	当接收数据包的长度大于 PAYLOAD_LENGTH 参数中定义的最大长度时置位。仅适用于

动态长度数据包的 Rx。		
第 4 位	CrcError	CRC 检查当前数据包的状态。无论如何，数据包在 FIFO 中都可用。 仅在启用 CRC 校验时适用于 Rx
第 3 位	AbortError	中止状态指示 Rx / Tx 中的当前数据包是否已中止。适用于 Rx 和 Tx。
第 2 位	HeaderReceived	指示是否已收到当前数据包的标头。 仅适用于动态长度数据包的 Rx
第 1 位	PacketReceived	表示数据包接收已完成。不表示数据包有效性。 仅适用于 Rx。
位 0	PacketCtrlBusy	表示数据包控制器正忙。适用于 Rx / Tx

表 13-16: GFSK 模式包中的 PacketStatus [4]

PacketStatus [4]	符号	描述	值
第 7: 3 点	保留的	保留的	
第 2: 0 位	SyncAdrsCode	检测到同步地址的代码	000: 同步地址检测错误 001: 检测到 sync_adrs_1' 010: sync_adrs_2', 检测到 100: 检测到 sync_adrs_3'

5. 完成所有检查后，通过发送命令清除 IRQ:

ClrIrqStatus (IRQMASK)

此命令将重置 irqMask 中相应位位置设置为 1 的标志。注意，DIO 可以映射到多个 IRQ 源（与 IRQ 源进行 OR 运算）。一旦所有相应的 IRQ 标志都设置为零，DIO 将变为零。

6. 获取发出命令的接收有效负载的数据包长度和起始地址:

GetRxbufferStatus ()

此命令返回最后接收的数据包的长度 (payloadLength) 和接收的第一个字节的地址 (rxBufferOffset)。它适用于所有调制解调器。

7. 使用以下命令读取数据缓冲区:

ReadBuffer (offset, payloadLength)

其中 offset 等于 rxBufferOffset，接收的有效负载长度为 payloadLength。

13.2 BLE 包

13.2.1 通用收发器设置

上电或硬复位后，收发器运行校准程序，并进入由 BUSY 引脚上的低电平状态指示的 STDBY_RC 模式。从这个状态开始，步骤是：

1. 如果不是 STDBY_RC 模式，则使用命令进入该模式：

SetStandby (STDBY_RC)

2. 通过发送命令定义 BLE 包：

SetPacketType (PACKET_BLE)

3. 通过发送命令定义 RF 频率：

SetRfFrequency (rfFrequency)

rfFrequency 的 LSB 等于 PLL 步长，即 $52\text{ MHz} / 2^{18}$ 。SetRfFrequency () 定义 Tx 频率。

在 Rx 中，频率降低中频 (IF)。IF 默认设置为 1.3 MHz。

4. 通过发送命令，指示数据包处理程序将读取的地址 (Tx 中的 txBaseAddress) 或写入 (Rx 中的 rxBaseAddress) 数据有效负载的第一个字节：

SetBufferBaseAddress (txBaseAddress, rxBaseAddress)

5. 通过发送命令定义调制参数：

SetModulationParams (modParam1, modParam2, modParam3)

- param [0]：比特率和带宽定义。
- param [1]：调制索引定义。
- param [2]：脉冲整形定义

在 BLE 不同比特率的情况下，调制索引和 BT 可以与标准一起使用。

表 13-17: BLE 和 GFSK 模式下的调制参数

参数	符号	值	BR [Mb / s]	BW [MHz DSB]
modParam1	BLE_BR_2_000_BW_2_4	0x04	2	2.4
modParam1	BLE_BR_1_600_BW_2_4	0x28	1.6	2.4

modParam1	BLE_BR_1_000_BW_2_4	0x4C	1	2.4
modParam1	BLE_BR_1_000_BW_1_2	0x45	1	1.2
modParam1	BLE_BR_0_800_BW_2_4	0x70	0.8	2.4
modParam1	BLE_BR_0_800_BW_1_2	0x69	0.8	1.2
modParam1	BLE_BR_0_500_BW_1_2	0x8D	0.5	1.2
modParam1	BLE_BR_0_500_BW_0_6	0x86 可以	0.5	0.6
modParam1	BLE_BR_0_400_BW_1_2	0xB1	0.4	1.2

表 13-17: BLE 和 GFSK 模式下的调制参数

参数	符号	值	BR [Mb / s]	BW [MHz DSB]
modParam1	BLE_BR_0_400_BW_0_6	和 0xAA	0.4	0.6
modParam1	BLE_BR_0_250_BW_0_6	0xCE	0.25	0.6
modParam1	BLE_BR_0_250_BW_0_3	0xC7	0.25	0.3
modParam1	BLE_BR_0_125_BW_0_3	0xEF	0.125	0.3

表 13-18: BLE 和 GFSK 模式下的调制参数

参数	符号	值	Modindex
modParam2	MOD_IND_0_35	为 0x00	0.35
modParam2	MOD_IND_0_5	0x01	0.5
modParam2	MOD_IND_0_75	0x02	0.75
modParam2	MOD_IND_1	×03	1
modParam2	MOD_IND_1_25	0x04	1.25
modParam2	MOD_IND_1_5	0x05	1.5
modParam2	MOD_IND_1_75	0x06	1.75
modParam2	MOD_IND_2	0x07	2
modParam2	MOD_IND_2_25	0x08 的	2.25
modParam2	MOD_IND_2_5	×09	2.5
modParam2	MOD_IND_2_75	的 0x0A	2.75
modParam2	MOD_IND_3	0x0B 中	3
modParam2	MOD_IND_3_25	0x0C	3.25
modParam2	MOD_IND_3_5	0x0D	3.5

modParam2	MOD_IND_3_75	为 0x0E	3.75
modParam2	MOD_IND_4	为 0x0F	4

表 13-19: BLE 和 GFSK 模式下的调制参数

参数	符号	值	BT
modParam3	BT_DIS	为 0x00	没有过滤
modParam3	BT_1	为 0x10	1
modParam3	BT_0_5	为 0x20	0.5

6. 通过发送命令定义要使用的数据包格式：

SetPacketParams (packetParam [0], packetParam [1], packetParam [2], packetParam [3])

- packetParam1 = ConnectionState
- packetParam2 = CrcLength
- packetParam3 = BleTestPayload
- packetParam4 =美白

请注意，虽然此命令最多可以接受 7 个参数，但在 BLE 模式下，SetPacketParams 只能接受 4 个参数。

表 13-20: BLE 包中的连接状态定义

参数	符号	值	BLE 州
packetParam1	BLE_MASTER_SLAVE	为 0x00	主从
packetParam1	BLE_ADVERTISER	0x02	广告
packetParam1	BLE_Tx_TEST_MODE	0x04	测试 Tx 模式
packetParam1	BLE_RX_TEST_MODE	0x06	测试 Rx 模式
packetParam1	BLE_RXTx_TEST_MDOE	0x08 的	专用 Rx Tx 测试模式

表 13-21: BLE 包中的 CRC 定义

参数	符号	值	包长模式
packetParam2	BLE_CRC_OFF	为 0x00	没有 CRC
packetParam2	BLE_CRC_3B	为 0x10	CRC 字段使用 3 字节

表 13-22: BLE 数据包测试模式下的 Tx 测试数据包有效负载

参数	符号	值	有效载荷内容
----	----	---	--------

基于 9 次多项式的伪随机二进制序列			
packetParam3	BLE_PRBS_9	为 0x00	$P7(x) = x^9 + x^5 + 1$ PRBS9 序列'1111111110000011110 1' (按传输顺序)
packetParam3	BLE_EYELONG_1_0	0x04	重复'11110000' (按传输顺序) 序列
packetParam3	BLE_EYESHORT_1_0	0x08 的	重复'10101010' (按传输顺序) 序列
基于 15 次多项式的伪随机二进制序列			
packetParam3	BLE_PRBS_15	0x0C	$P15(x) = x^{15} + x^{14} + x^{13} + x^{12} + x^2 + x + 1$
packetParam3	BLE_ALL_1	为 0x10	重复'11111111' (按传输顺序) 序列
packetParam3	BLE_ALL_0	0x14	重复'11111111' (按传输顺序) 序列
packetParam3	BLE_EYELONG_0_1	为 0x18	重复'00001111' (按传输顺序) 序列
packetParam3	BLE_EYESHORT_0_1	为 0x1C	重复'01010101' (按传输顺序) 序列

表 13-23: BLE 数据包中的白化启用

参数	符号	值	美白模式
packetParam4	BLE_WHITENING_ENABLE	为 0x00	白皙化
packetParam4	BLE_WHITENING_DISABLE	0x08 的	白皙的

7. 定义同步字值

除了这些参数，用户还需要定义 32 位同步字 SyncWord1。这是通过发送 `WriteRegister ()` 命令来执行的，下一个表给出了同步字的地址。

表 13-24: BLE 数据包中的同步字定义

同步 Word	字节	地址
SyncWord1	SyncWord1 (31:24)	0x09CF
	SyncWord1 (23:16)	0x09D0
	SyncWord1 (15: 8)	0x09D1
	SyncWord1 (7: 0)	0x09D2

用于 CRC 的种子也需要针对某些应用进行修改。这是通过发送函数 `WriteRegister ()` 通过直接寄存器访问来执行的。

表 13-25: CRC 初始化寄存器

参数	字节	地址
CrcInit	CRC 初始值	0x9C7 (MSB)
		0x9C8
		0x9C9 (LSB)

13.2.2 Tx 设置和操作

1. 通过发送命令定义输出功率和斜坡时间：

SetTxParam (电力, ramptime)

2. 与其他调制解调器相反，在 SX1280 芯片的数据缓冲区中以 BLE 模式写入的有效负载必须包含 BLE 头。要在有效负载开头添加的 BLE 标头必须与第 94 页上的第 13.2.1 节“通用收发器设置”中步骤 6 中选择的 BLE 模式相对应。请参见图 7-5：标头定义的 PDU 标头格式。

通过发出命令将有效负载发送到数据缓冲区：

WriteBuffer (偏移量, 数据)

其中 data 是包含要发送的 BLE 头的有效载荷，offset 是有效载荷的第一个字节在 FIFO 中的地址。

3. 使用命令配置 DIO 和中断源 (IRQ)：

SetDioIrqParams (irqMask, dio1Mask, dio2Mask, dio3Mask)

在典型的 Tx 操作中，可以选择一个或多个 IRQ 源：

- TxDone IRQ 表示数据包传输结束。收发器将处于 STDBY_RC 模式。
- RxTxTimeout (可选) 以确保不会发生死锁。如果发生超时，收发器将自动返回 STDBY_RC 模式。

4. 配置完成后，将收发器设置为发射器模式，使用命令开始传输：

SetTx (periodBase, periodBaseCount [15: 8], periodBaseCount [7: 0])

如果需要超时，请将 *periodBaseCount* 设置为不等于零的值。此超时可用于避免死锁。

等待 IRQ TxDone 或 RxTxTimeout

一旦发送数据包或发生超时，收发器将自动进入 STDBY_RC 模式

5. (可选) 检查数据包状态以确保数据包已正确发送。通过发出命令：

GetPacketStatus ()

在这种情况下，只有参数 *packetStatus3* 才有用。

表 13-26: BLE 包中的 PacketStatus3

PacketStatus3	符号	描述
第 7: 1 点	保留的	保留的
位 0	PktSent	表示数据包传输已完成。不显示数据包有效性。 仅适用于 Tx。

6. 通过发送命令清除 TxDone 或 RxTxTimeout IRQ：

此命令将重置 *irqMask* 中相应位位置设置为 1 的标志。

13.2.3 Rx 设置和操作

1. 使用命令配置 DIO 和中断源 (IRQ):

SetDioIrqParams (irqMask, dio1Mask, dio2Mask, dio3Mask)

在典型的 BLE Rx 操作中, 可以选择一个或多个 IRQ 源

- RxDone 表示已检测到数据包。此 IRQ 并不意味着数据包有效 (大小或 CRC 正确)。用户必须检查数据包状态以确保收到有效的打包。
- SyncWordValid 表示已检测到同步字。
- CrcError 表示收到的数据包有 CRC 错误
- RxTxTimeout 表示在 SetRx () 命令中由 timeout 参数定义的给定时间包中未检测到数据包。

将这些 IRQ 映射到一个 DIO (DIO1 或 DIO2 或 DIO3)。

2. 配置完成后, 将收发器设置为接收器模式, 使用命令开始接收:

SetRx (periodBase, periodBaseCount [15: 8], periodBaseCount [7: 0]) 根据

periodBaseCount, 可能有 3 种可能的 Rx 行为:

- *periodBaseCount* 设置为 0, 然后没有 Timeout, Rx Single 模式, 设备将保持 Rx 模式, 直到发生接收并且设备完成后返回 STDBY_RC 模式
- *periodBaseCount* 设置为 0xFFFF, Rx 连续模式, 设备保持 Rx 模式, 直到主机发送命令改变操作模式。设备可以接收多个数据包。每次接收到分组时, 分组接收指示被提供给主机, 并且设备将继续搜索新分组。
- *periodBaseCount* 设置为另一个值, 然后 Timeout 处于活动状态。收发器保持 Rx 模式; 它会在计时器计数结束时或收到数据包时自动返回 STDBY_RC 模式。一旦检测到数据包, 就会自动禁用定时器以允许完全接收数据包。

3. 在典型情况下, 使用超时并等待 IRQ RxDone 或 RxTxTimeout。

如果 IRQ RxDone 被置位, 则如果使用单模式 (超时设置为不同于 0xFFFF 的值), 收发器将进入 STDBY_RC 模式。如果使用连续模式 (超时设置为 0xFFFF), 收发器将保留在 Rx 中并继续侦听新数据包。

4. 检查数据包状态以确保使用以下命令正确接收数据包:

GetPacketStatus ()

该命令返回以下参数:

- *RssiSync*: 检测到同步字时的 RSSI 值;

- *packetStatus2*: 提供有关收到的最后一个数据包的信息，如下表所述;
- *packetStatus3*: 在 BLE 数据包中，此状态在 Tx 模式下指示是否已发送数据包;

表 13-27: BLE 模式下的 PacketStatus2

PacketStatus2	符号	描述
第 7 位	保留的	保留的
第 6 位	SyncError	同步当前数据包的地址检测状态仅在启用同步地址检测时适用于 Rx。

表 13-27: BLE 模式下的 PacketStatus2

PacketStatus2	符号	描述
第 5 位	lengthError	当接收数据包的长度大于 PAYLOAD_LENGTH 参数中定义的最大长度时置位。 仅适用于动态长度数据包的 Rx。
第 4 位	CrcError	CRC 检查当前数据包的状态。无论如何，数据包在 FIFO 中都可用。 仅在启用 CRC 时适用于 Rx
第 3 位	AbortError	中止状态指示 Rx / Tx 中的当前数据包是否已中止。 适用于 Rx 和 Tx。
第 2 位	HeaderReceived	指示是否已收到当前数据包的标头。 仅适用于动态长度数据包的 Rx
第 1 位	PacketReceived	表示数据包接收已完成。不表示数据包有效性。 仅适用于 Rx。
位 0	PacketCtrlBusy	表示数据包控制器正忙。适用于 Rx / Tx

- *packetStatus4*: 指示哪个相关器检测到同步字。如果是 BLE，则仅使用 sync_adrs_1。

表 13-28: BLE 模式下的 PacketStatus4

PacketStatus4	符号	描述
第 7: 3 点	保留的	保留的
第 2: 0 位	SyncSdrsCode	检测到同步地址的代码 0x0: 同步地址检测错误 0x1: sync_adrs_1'检测到其他 人: 保留

5. 完成所有检查后，通过发送命令清除 IRQ:

ClrIrqStatus (IRQMASK)

此命令将重置 *irqMask* 中相应位位置设置为 1 的标志。注意 aa DIO 可以映射到多个 IRQ 源（与 IRQ 源进行 OR 运算）。一旦 IRQ 标志设置为零，DIO 将变为零。

6. 获取发出命令的接收有效负载的数据包长度和起始地址：

GetRxbufferStatus ()

此命令返回最后接收的数据包的长度 (payloadLength) 和接收的第一个字节的地址 (rxBufferOffset) 它适用于所有调制解调器。地址是相对于数据缓冲区的第一个字节的偏移量。

7. 使用以下命令读取数据缓冲区：

ReadBuffer (offset , payloadLength) 其

中 offset 等于 rxBufferOffset。

13.2.4 BLE 特定功能

13.2.4.1 SetAutoTx ()

另外一个命令可用于简化 BLE 数据包的实现。BLE 要求收发器能够在数据包接收后 125µs 发回响应。这是通过发送命令 SetAutoTx () 来执行的，该命令允许收发器在数据包接收结束后的用户可编程时间 (时间) 之后发送数据包。

必须以 STDBY_RC 模式发出 SetAutoTx (时间)。表 13-

29: SetAutoTx 模式

字节	0	1	2
来自主机的数据	操作码= 0x98	时间 (15: 8)	时间 (7: 0)

时间以µs 为单位。数据包接收结束与下一个数据包传输开始之间的延迟定义如下：

$$TxDelay = 时间 - 偏移量$$

其中 offset 是收发器切换模式所需的时间，等于 33µs。

一旦发出此命令，每次收发器收到一个数据包，它将自动切换到 Tx 并在预定义的时间后发送数据包。

如果用户想要正常操作 (在 Tx 之后进入 STDBY_RC)，他需要发送命令 SetAutoTx ()，并将 time 参数设置为零。

13.3 FLRC 数据包

13.3.1 通用收发器设置

上电或硬复位后，收发器运行校准程序，并进入由 BUSY 引脚上的低电平状态指示的 STDBY_RC 模式。从这个状态开始，步骤是

1. 如果不处于 STDBY_RC 模式，则通过发送命令进入此模式：

SetStandby (STDBY_RC)

2. 通过发送命令定义 GFSK 数据包类型:

SetPacketType (PACKET_FLRC)

3. 通过发送命令定义 RF 频率:

SetRfFrequency (rfFrequency)

rfFrequency 的 LSB 等于 PLL 步长, 即 $52e6 / 2^{18}$ Hz。SetRfFrequency () 定义 Tx 频率。

在 Rx 中, 频率降低中频 (IF)。IF 默认设置为 1.3 MHz。

4. 通过发送命令, 指示数据包处理程序将读取的地址 (Tx 中的 txBaseAddress) 或写入 (Rx 中的 rxBaseAddress) 数据有效负载的第一个字节:

SetBufferBaseAddress (txBaseAddress, rxBaseAddress)

请注意, txBaseAddress 和 rxBaseAddress 是数据存储器映射开头的偏移量。

5. 通过发送命令定义调制参数:

SetModulationParams (modParam1, modParam2, modParam3)

比特率和带宽通过 param [0] 链接。在误差校正机构中使用的编码率中所定义 PARAM [1] 和 BT 中定义 PARAM [2]。

表 13-30: FLRC 模式下的调制参数: 带宽和比特率

参数	符号	值	比特率[Mb / s]	带宽[MHz DSB]
modParam1	FLRC_BR_1_300_BW_1_2	0x45	1.3	1.2
modParam1	FLRC_BR_1_000_BW_1_2	0x69	1.04	1.2
modParam1	FLRC_BR_0_650_BW_0_6	0x86 可以	0.65	0.6
modParam1	FLRC_BR_0_520_BW_0_6	和 0xAA	0.52	0.6
modParam1	FLRC_BR_0_325_BW_0_3	0xC7	0.325	0.3
modParam1	FLRC_BR_0_260_BW_0_3	将 0xEB	0.26	0.3

表 13-31: FLRC 模式下的调制参数: 编码率

参数	符号	值	编码率
modParam2	FLRC_CR_1_2	为 0x00	$\frac{1}{2}$
modParam2	FLRC_CR_3_4	0x02	$\frac{3}{4}$
modParam2	FLRC_CR_1_0	0x04	1
modParam2	保留的	大于或等于 3	保留的

表 13-32: FLRC 模式下的调制参数: BT

参数	符号	值	BT
modParam3	BT_DIS	为 0x00	没有过滤
modParam3	BT_1	为 0x10	1
modParam3	BT_0_5	为 0x20	0.5

6. 通过发送命令定义要使用的数据包格式:

SetPacketParams (packetParam1, packetParam2, packetParam3, packetParam4, packetParam5, packetParam6, packetParam7)

- packetParam1 = AGCPreambleLength
- packetParam2 = SyncWordLength
- packetParam3 = SyncWordMatch
- packetParam4 = PacketType
- packetParam5 = PayloadLength
- packetParam6 = CrcLength
- packetParam7 =美白

表 13-33: FLRC 数据包中的 AGC 前导码长度定义

参数	符号	值	前导码长度, 以位为单位
packetParam1	PREAMBLE_LENGTH_4_BITS	为 0x00	保留的
packetParam1	PREAMBLE_LENGTH_8_BITS	为 0x10	8
packetParam1	PREAMBLE_LENGTH_12_BITS	为 0x20	12
packetParam1	PREAMBLE_LENGTH_16_BITS	的 0x30	16
packetParam1	PREAMBLE_LENGTH_20_BITS	0x40 的	20

表 13-33: FLRC 数据包中的 AGC 前导码长度定义

参数	符号	值	前导码长度, 以位为单位
packetParam1	PREAMBLE_LENGTH_24_BITS	为 0x50	24
packetParam1	PREAMBLE_LENGTH_28_BITS	0x60	28
packetParam1	PREAMBLE_LENGTH_32_BITS	0x70	32

使用 AGC 时的最小前导码长度应为 8 比特, 比特率为 1 Mb / s。对于其他比特率, 前导码比特的最小数量必须至少为 16 比特。

用于同步字的字节数由 packetParam2 定义。用户可以依赖于检测数据包开始所需的内置 21 位前导码，或者在多个设备的情况下添加 4 个额外的同步字以进行地址检测。

表 13-34: FLRC 数据包中的同步字长定义

参数	符号	值	以字节为单位同步字大小
packetParam2	FLRC_SYNC_NOSYNC	为 0x00	21 位前导码
packetParam2	FLRC_SYNC_WORD_LEN_P32S	0x04	21 位前同步码+ 32 位同步 Word

通过 3 个相关器，收发器可以在此时搜索多个同步字。同步字检测的组合由参数 PacketParam3 定义。

表 13-35: FLRC 数据包中的同步字组合

参数	符号	值	同步 Word 组合使用
packetParam3	RX_DISABLE_SYNC_WORD	为 0x00	禁用同步字
packetParam3	RX_MATCH_SYNC_WORD_1	为 0x10	SyncWord1
packetParam3	RX_MATCH_SYNC_WORD_2	为 0x20	SyncWord2
packetParam3	RX_MATCH_SYNC_WORD_1_2	的 0x30	SyncWord1 或 SyncWord2
packetParam3	RX_MATCH_SYNC_WORD_3	0x40 的	SyncWord3
packetParam3	RX_MATCH_SYNC_WORD_1_3	为 0x50	SyncWord1 或 SyncWord3
packetParam3	RX_MATCH_SYNC_WORD_2_3	0x60	SyncWord2 或 SyncWord3
packetParam3	RX_MATCH_SYNC_WORD_1_2_3	0x70	SyncWord1 或 SyncWord2 或 SyncWord3

有效负载长度由 packetParam4 参数定义。Tx 中的数据包处理程序使用此参数来发送确切的字节数。在 Rx 中，在可变长度模式下，数据包处理程序将过滤掉大小大于有效负载长度的所有数据包。请注意，最小有效负载长度为 6 个字节。

表 13-36: FLRC 数据包中的数据包类型定义

参数	符号	值	包长度模式
packetParam4	PACKET_FIXED_LENGTH	为 0x00	固定长度模式
packetParam4	PACKET_VARIABLE_LENGTH	为 0x20	可变长度模式

表 13-37: FLRC 数据包中的有效载荷长度定义

参数	符号	值	描述
packetParam5	PAYLOAD_LENGTH	[6 ... 127]	有效载荷长度，以字节为单

在 FLRC 模式下，CRC 可以在 2,3 或 4 个字节上计算或忽略。这是使用参数 *param [5]* 定义的。

表 13-38: FLRC 数据包中的 CRC 定义

参数	符号	值	CRC 类型
packetParam6	CRC_OFF	为 0x00	没有 CRC
packetParam6	CRC_1_BYTE	为 0x10	CRC 字段使用 1 个字节
packetParam6	CRC_2_BYTE	为 0x20	CRC 字段使用 2 个字节
packetParam6	CRC_3_BYTE	的 0x30	CRC 字段使用 3 个字节

用于 CRC 的种子也需要针对某些应用进行修改。这是通过发送函数 *WriteRegister ()* 通过直接寄存器访问来执行的。

表 13-39: CRC 初始化寄存器

参数	字节	地址
CrcInit	CRC 初始值 MSB	0x9c8
	CRC 初始值 LSB	0x9c9

的 CRC 多项式也可以通过直接寄存器访问使用命令改性 *WriteRegister ()* 。表 13-40: CRC 多项式定义

参数	字节	地址	描述
CrcPolynomial	CRC 多项式 MSB	0x9C6	定义 16 位 CRC 多项式的 LSB 字节或定义 8 位 CRC 多项式 例如，编程以下多项式 $P_{16}(x) = x^{16} + x^{12} + x^5 + 1$
	CRC 多项式 LSB	0x9C7	初始化 <code>crc_polynomial [15: 0] = 0x1021</code> 编程以下多项式 $P_8(x) = x^8 + x^2 + x + 1$ 初始化 <code>crc_polynomial [7: 0] = 0x07</code>

在 FLRC 数据包类型中，无法启用白化。您必须始终将 packetParam7 的值设置为 *disabled*。

表 13-41: FLRC 数据包中的白化定义

参数	符号	值	描述
----	----	---	----

packetParam7	美白	0x08 的	美白禁用
--------------	----	--------	------

7. 定义同步字值

除了这些参数之外，用户还需要定义同步字（SyncWord1, SyncWord2, SyncWord3）。这是通过发送 WriteRegister () 命令来执行的。下表给出了同步字的地址..

表 13-42：FLRC 数据包中的同步字定义

同步 Word	字节	地址
SyncWord1	SyncWord1 [31:24]	0x09CF
	SyncWord1 [23:16]	0x09D0
	SyncWord1 [15: 8]	0x09D1
	SyncWord1 [7: 0]	0x09D2
SyncWord2	SyncWord2 [31:24]	0x09D4
	SyncWord2 [23:16]	0x09D5
	SyncWord2 [15: 8]	0x09D6
	SyncWord2 [7: 0]	0x09D7
SyncWord3	SyncWord3 [31:24]	0x09D9
	SyncWord3 [23:16]	0x09DA
	SyncWord3 [15: 8]	0x09DB
	SyncWord3 [7: 0]	0x09DC

13.3.2 Tx 设置和操作

1. 通过发送命令定义输出功率和斜坡时间:

SetTxParam (电力, rampTime)

2. 通过发送命令将有效负载发送到数据缓冲区:

WriteBuffer (偏移量, 数据)

其中 data 是要发送的有效负载，offset 是有效负载的第一个字节在缓冲区中的地址。在正常操作中，偏移量将对应于 txBaseAddress。

3. 通过发送命令配置 DIO 和中断源 (IRQ):

SetDioIrqParams (irqMask, dio1Mask, dio2Mask, dio3Mask)

在典型的 Tx 操作中，可以选择一个或多个 IRQ 源：

- TxDone IRQ 表示数据包传输结束。收发器将处于 STDBY_RC 模式。
- RxTxTimeout（可选）以确保不会发生死锁。如果发生超时，收发器将自动返回 STDBY_RC 模式。

4. 配置完成后，将收发器设置为发射器模式，使用命令开始传输：

SetTx (periodBase, periodBaseCount [15: 8], periodBaseCount [7: 0])

如果需要超时，请将 periodBaseCount 值设置为零。此超时可用于避免死锁。

等待 IRQ TxDone 或 RxTxTimeout。一旦发送数据包或发生超时，收发器将自动进入 STDBY_RC 模式。

5. (可选) 检查数据包状态，以确保使用以下命令正确发送数据包：

GetPacketStatus ()

在这种情况下，只有参数 packetStatus3 才有用。

表 13-43：FLRC 数据包中的 PacketStatus3

PacketStatus3	符号	描述
第 7: 6	rxpid	接收到的数据包的 PID 字段 仅适用于动态长度数据包的 Rx
第 5 位	rx_no_ack	收到的数据包的 NO_ACK 字段仅适用于 动态长度数据包的 Rx。
第 4 位	rxpiderr	当前数据包的 PID 检查状态 rxpid (N) = rxpid (N-1) 和 crc_checksum (N) = crc_checksum (N-1) 当 rxpid_filter_enable = '1' 时，仅适用于动态长度数据包的 Rx
第 3: 1 位	保留的	保留的
位 0	PktSent	表示数据包传输已完成。仅表示传输过程完成而不是数据包有效性。仅适用于 Tx。

6. 通过发送命令清除 TxDone 或 RxTxTimeout IRQ：

ClrIrqStatus (IRQMASK)

此命令将重置 irqMask 中相应位位置设置为 1 的标志。

13.3.3 Rx 设置和操作

1. 通过发送命令配置 DIO 和中断源（IRQ）：

SetDioIrqParams (IrqMask, Dio1Mask, Dio2Mask, Dio3Mask)

在典型的 FLRC Rx 操作中，可以选择一个或多个 IRQ 源：

- RxDone 表示已检测到数据包。此 IRQ 并不意味着数据包有效（大小或 CRC 正确）。用户必须检查数据包状态以确保收到有效的打包。
- SyncWordValid 表示已检测到同步字。
- CrcError 表示收到的数据包有 CRC 错误
- RxTxTimeout 表示在 SetRx () 命令中由 timeout 参数定义的给定时间帧内未检测到数据包。

将这些 IRQ 映射到一个 DIO（DIO1 或 DIO2 或 DIO3）。

2. 配置完成后，将收发器设置为接收器模式，使用命令开始接收：

SetRx (periodBase, periodBaseCount [15: 8], periodBaseCount [7: 0]) 根据

periodBaseCount，可能有 3 种可能的 Rx 行为：

- periodBaseCount 设置为 0，然后没有超时，Rx Single 模式，设备将保持 Rx 模式，直到发生接收，设备完成后返回 STDBY_RC 模式。
- periodBaseCount 设置为 0xFFFF，Rx 为连续模式，设备保持 Rx 模式，直到主机发送命令改变操作模式。设备可以接收多个数据包。每次接收到分组时，都会向主机发送分组接收指示，并且设备将继续搜索新分组。
- periodBaseCount 设置为另一个值，然后 Timeout 处于活动状态。设备保持 Rx 模式；它会在计时器计数结束时或收到数据包时自动返回 STDBY_RC 模式。一旦检测到数据包，就会自动禁用定时器以允许完全接收数据包。

3. 通常，使用超时并等待 IRQ RxDone 或 RxTxTimeout。

如果 IRQ RxDone 上升，如果使用单模式（超时设置为不同于 0xFFFF 的值），收发器将进入 STDBY_RC 模式。如果使用连续模式（超时设置为 0xFFFF），收发器将保留在 Rx 中并继续侦听新数据包。

4. 通过发送命令检查数据包状态以确保已正确接收数据包：

GetPacketStatus ()

该命令返回以下参数：

- RssiSync: 检测到同步字时的 RSSI 值。实际信号功率为 -RssiSync / 2 (dBm)
- packetStatus2: 提供有关收到的最后一个数据包的信息，如下表所述
- packetStatus3: 在 FLRC 数据包中，此状态在 Tx 模式下指示是否已发送数据包
- packetStatus4: 指示哪个相关器检测到同步字

表 13-44: FLRC 数据包中的 PacketStatus2

PacketStatus2	符号	描述
第 7 位	保留的	保留的
第 6 位	SyncError	同步当前数据包的地址检测状态仅在启用同步地址检测时适用于 Rx。
第 5 位	LengthError	当接收数据包的长度大于 PAYLOAD_LENGTH 参数中定义的最大长度时置位。 仅适用于动态长度数据包的 Rx。
第 4 位	CrcError	CRC 检查当前数据包的状态。无论如何，数据包在 FIFO 中都可用。 仅在启用 CRC 时适用于 Rx
第 3 位	AbortError	中止状态指示 Rx / Tx 中的当前数据包是否已中止。 适用于 Rx 和 Tx。
第 2 位	HeaderReceived	指示是否已收到当前数据包的标头。 仅适用于动态长度数据包的 Rx
第 1 位	PacketReceived	表示数据包接收已完成。不表示数据包有效性。 仅适用于 Rx。
位 0	PacketCtrlBusy	表示数据包控制器正忙。适用于 Rx / Tx

表 13-45: FLRC 数据包中的 PacketStatus3

PacketStatus3	符号	描述
第 7: 6	rxpid	接收到的数据包的 PID 字段 仅适用于动态长度数据包的 Rx
第 5 位	rx_no_ack	收到的数据包的 NO_ACK 字段仅适用于 动态长度数据包的 Rx。
第 4 位	rxpiderr	当前数据包的 PID 检查状态 $rxpid(N) = rxpid(N-1)$ 和 $crc_checksum(N) = crc_checksum(N-1)$ 当 rxpid_filter_enable = '1' 时，仅适用于动态长度数据包的 Rx
第 3: 1 位	保留的	保留的
位 0	PktSent	表示数据包传输已完成。不表示数据包有效性。仅适用于 Tx。

表 13-46: FLRC 数据包中的 PacketStatus4

PacketStatus4	符号	描述
第 7: 3 点	保留的	保留的

		检测到同步地址的代码
		000: 同步地址检测错误
第 2: 0 位	SyncSdrsCode	001: 检测到 sync_adrs_1'
		010: sync_adrs_2', 检测到
		100: 检测到 sync_adrs_3'

5. 完成所有检查后，通过发送命令清除 IRQ:

ClrIrqStatus (IRQMASK)

此命令将重置 *irqMask* 中相应位位置设置为 1 的标志。注意 aa **DIO** 可以映射到多个 IRQ 源（与 IRQ 源进行 OR 运算）。一旦 IRQ 标志设置为零，**DIO** 将变为零。

6. 通过发送命令获取数据包长度和接收的有效负载的起始地址:

GetRxBufferStatus ()

此命令返回最后接收的数据包的长度 (*payloadLength*) 和接收的第一个字节的地址 (*rxBufferOffset*) 它适用于所有调制解调器。地址是相对于数据缓冲区的第一个字节的偏移量。

7. 使用以下命令读取数据缓冲区:

ReadBuffer (offset , payloadLength) 其

中 offset 等于 *rxBufferOffset* , 命令包含 *payloadLength*。

13.4 LoRa 包

13.4.1 LoRa 的通用收发器设置

上电或硬复位后，收发器运行校准程序，并进入由 BUSY 引脚上的低电平状态指示的 STDBY_RC 模式。从这个状态开始，步骤是

1. 如果不处于 STDBY_RC 模式，则通过发送命令进入此模式:

SetStandby (STDBY_RC)

2. 通过发送命令定义 LoRa 数据包类型:

SetPacketType (PACKET_LORA)

3. 通过发送命令定义 RF 频率:

SetRfFrequency (rfFrequency)

的 **LSB** 的 *rfFrequency* 等于 **PLL** 步骤即 $52e6 / 2^{18}$ 赫兹。*SetRfFrequency ()* 定义 Tx 频率。

在 Rx 中，频率降低中频（IF）。IF 默认设置为 1.65 MHz。

4. 通过发送命令，指示数据包处理程序将读取的地址（Tx 中的 *txBaseAddress*）或写入（Rx 中的 *rxBaseAddress*）数据有效负载的第一个字节：

SetBufferBaseAddress (txBaseAddress, rxBaseAddress)

请注意，*txBaseAddress* 和 *rxBaseAddress* 相对于数据存储映射的开头是偏移的。

5. 通过发送命令定义调制参数：

SetModulationParams (modParam1, modParam2, modParam3)

modParam1 定义信号 BW，*modParam2* 定义 SF，*modParam3* 定义编码率（CR）。

表 13-47：LoRa 模式下的调制参数

参数	符号	值	传播因素
modParam1	LORA_SF_5	为 0x50	五
modParam1	LORA_SF_6	0x60	6
modParam1	LORA_SF_7	0x70	7
modParam1	LORA_SF_8	0x80 的	8
modParam1	LORA_SF_9	的 0x90	9
modParam1	LORA_SF_10	0xA0	10
modParam1	LORA_SF_11	0xB0	11
modParam1	LORA_SF_12	将 0xC0	12

注意，在 *SetModulationParams* 命令之后：

- 如果选择的扩频因子是 SF5 或 SF6，则需要使用 *WriteRegister (0x925,0x1E)*
- 如果扩频因子为 SF7 或 SF-8，则必须使用命令 *WriteRegister (0x925,0x37)*
- 如果扩频因子为 SF9，SF10，SF11 或 SF12，则必须使用命令 *WriteRegister (0x925,0x32)*

表 13-48：LoRa 模式下的调制参数

参数	符号	值	带宽[kHz]
modParam2	LORA_BW_1600	的 0x0A	1625.0
modParam2	LORA_BW_800	为 0x18	812.5
modParam2	LORA_BW_400	0x26	406.25
modParam2	LORA_BW_200	0x34	203.125

表 13-49: LoRa 模式下的调制参数

参数	符号	值	编码率
modParam3	LORA_CR_4_5	0x01	4/5
modParam3	LORA_CR_4_6	0x02	4/6
modParam3	LORA_CR_4_7	x03	4/7
modParam3	LORA_CR_4_8	0x04	4/8
modParam3	LORA_CR_LI_4_5	0x05	五分之四*
modParam3	LORA_CR_LI_4_6	0x06	4/6 *
modParam3	LORA_CR_LI_4_7	0x07	4/8 *

*已经实现了一种新的交织方案，以增加对突发干扰和/或强多普勒事件的鲁棒性。该 FEC 一直保持同样的限制对复杂性的影响。长交错重新使用检测期间使用的存储器。当设置长交织时，该存储器用于编码和解码。

通过设置 *modParam3* = 0x5,0x6 或 0x7 (LORA_CR_LI_4_5, LORA_CR_LI_4_6 或 LORA_CR_LI_4_7) 来选择长交错。编码率在标头中发出信号。以前，只有值 0x0 到 0x4 有效。

编码率值分别为 4 / 5, 4 / 6, 4 / 8。所以 LORA_CR_LI_4_5 就像 LORA_CR_4_5，LORA_CR_LI_4_6 像 LORA_CR_4_6，LORA_CR_LI_4_7 像 LORA_CR_4_8 用不同的交织方案。

长交错与隐式报头兼容。与传统交织一样，在长交织中发生加扰。

注意：LORA_CR_LI_4_7 的最大有效负载长度存在限制。如果启用 CRC，则有效负载长度不应超过 253 个字节。

6. 通过发送命令定义要使用的数据包格式：

SetPacketParams (pktParam1, pktParam2, pktParam3, pktParam4, pktParam5)

- *packetParam1* = PreambleLength，它定义主要由 Tx 模式下的数据包处理使用的前导码长度（以符号表示）。
- *packetParam2* = HeaderType
- *packetParam3* = PayloadLength
- *packetParam4* = CRC
- *packetParam5* = InvertIQ / chirp invert
- *packetParam1* 定义以 LoRa 符号表示的前导码长度数。建议值为 12 个符号。**表 13-50: LoRa 或 Ranging 中的前导码定义**

参数	符号	值	符号中的前导码长度
packetParam1 (3: 0)	LORA_PBLE_LEN_MANT	[1:15]	序言长度=
packetParam1 (7: 4)	LORA_PBLE_LEN_EXP	[1:15]	LORA_PBLE_LEN_MANT * 2 ^ (LORA_PBLE_LEN_EXP)

数据包的类型由参数 PacketParam2 定义。对于固定长度的数据包，不会显示任何标头，并使用隐式标头。在可变长度分组中，使用显式报头模式。

表 13-51: LoRa 或测距包中的包类型定义

参数	符号	值	标题模式
packetParam2	EXPLICIT_HEADER	为 0x00	显瘦标题
packetParam2	IMPLICIT_HEADER	0x80 的	隐含的标题

有效负载长度在 *packetParam3* 中定义。

表 13-52: LoRa 包中的有效载荷长度定义

参数	符号	值	PayloadLength
packetParam3	PayloadLength	[1 ... 0.255]	PayloadLength

注意: LORA_CR_LI_4_7 的最大有效负载长度存在限制。如果启用 CRC, 则有效负载长度不应超过 253 个字节。

该 CRC 用法在定义 *packetParam4*。表 13-53:

LoRa 数据包中的 CRC 启用

参数	符号	值	CRC 模式
packetParam4	LORA_CRC_ENABLE	为 0x20	CRC ENABLE
packetParam4	LORA_CRC_DISABLE	为 0x00	CRC 禁用

IQ 交换由 *PacketParam5* 定义。

表 13-54: LoRa 或测距数据包中的 IQ 交换

参数	符号	值	LoRa IQ 交换
packetParam5	LORA_IQ_STD	0x40 的	智商定义
packetParam5	LORA_IQ_INVERTED	为 0x00	智商换了

13.4.2 Tx 设置和操作

1. 通过发送命令定义输出功率和斜坡时间:

SetTxParam (power, rampTime)

2. 通过发送命令将有效负载发送到数据缓冲区:

WriteBuffer (偏移量, *数据)

其中 *data 是指向有效负载的指针, offset 是有效负载的第一个字节在缓冲区中的地址。在正常操作中, 偏移量将对应于 txBaseAddress。

3. 通过发送命令配置 DIO 和中断源 (IRQ):

SetDioIrqParams (irqMask, dio1Mask, dio2Mask, dio3Mask)

在典型的 Tx 操作中，用户可以选择一个或多个 IRQ 源：

- TxDone IRQ 表示数据包传输结束。收发器将处于 STDBY_RC 模式。
- RxTxTimeout (可选) 以确保不会发生死锁。如果发生超时，收发器将自动返回 STDBY_RC 模式。

4. 配置完成后，将收发器设置为发送器模式，通过发送命令开始发送：

SetTx (periodBase, periodBaseCount [15: 8], periodBaseCount [7: 0])

如果需要超时，请将 *periodBaseCount* 设置为非零值。此超时可用于避免死锁。

等待 IRQ TxDone 或 RxTxTimeout

一旦发送数据包或发生超时，收发器将自动进入 STDBY_RC 模式。

5. 通过发送命令清除 TxDone 或 RxTxTimeout IRQ：

ClrIrqStatus (irqStatus)

此命令将重置 *irqStatus* 中相应位位置 设置为 1 的标志。

13.4.3 Rx 设置和操作

1. 使用命令配置 DIO 和中断源 (IRQ)：

SetDioIrqParams (IRQMASK, dio1Mask, dio2Mask, dio3Mask)

在典型的 LoRa Rx 操作中，用户可以选择以下一个或多个 IRQ 源：

- RxDone 表示已检测到数据包。此 IRQ 并不意味着数据包有效 (大小或 CRC 正确)。用户必须检查数据包状态以确保已收到有效的打包。
- SyncWordValid 表示已检测到同步字。
- CrcError 表示收到的数据包有 CRC 错误
- RxTxTimeout 表示在 SetRx () 命令中由 timeout 参数定义的给定时间帧内未检测到数据包。

2. 配置完成后，将收发器设置为接收器模式，使用命令开始接收：

SetRx (periodBase, periodBaseCount [15: 8], periodBaseCount [7: 0]) 根据

periodBaseCount，可能有 3 种可能的 Rx 行为：

- *periodBaseCount* 设置为 0，然后没有 Timeout，Rx Single 模式，设备将保持 Rx 模式，直到发生接收，设备完成后返回 STDBY_RC 模式。

- *periodBaseCount* 设置为 0xFFFF, Rx 为连续模式, 设备保持 Rx 模式, 直到主机发送命令改变操作模式。设备可以接收多个数据包。每次接收到分组时, 分组接收指示被提供给主机, 并且设备将继续搜索新分组。
- *periodBaseCount* 设置为另一个值, 然后 Timeout 处于活动状态。设备保持 Rx 模式; 它会在计时器计数结束时或收到数据包时自动返回 STDBY_RC 模式。一旦检测到数据包, 就会自动禁用定时器以允许完全接收数据包。

3. 在典型情况下, 使用超时并等待 IRQ RxDone 或 RxTxTimeout。

如果 IRQ RxDone 被置位, 则如果使用单模式 (超时设置为不同于 0xFFFF 的值), 收发器将进入 STDBY_RC 模式。如果使用连续模式 (超时设置为 0xFFFF), 收发器将保留在 Rx 中并继续侦听新数据包。

4. 通过发送命令检查数据包状态以确保已正确接收数据包:

GetPacketStatus ()

该命令返回以下参数:

- *SnrPkt* 估计收到的最后一个数据包的 SNR。以 2 的补码格式乘以 4。实际 SNR, 以 dB 为单位 = $\text{SnrPkt} / 4$

5. 完成所有检查后, 通过发送命令清除 IRQ:

ClrIrqStatus (IRQMASK)

此命令将重置 *irqMask* 中相应位位置 设置为 1 的标志。注意 aa DIO 可以映射到多个 IRQ 源 (与 IRQ 源进行 OR 运算)。一旦 IRQ 标志设置为零, DIO 将变为零。

6. 通过发送命令获取收到的有效负载的数据包长度和起始地址:

GetRxBufferStatus ()

此命令返回最后接收的数据包的长度 (*payloadLengthRx*) 和接收的第一个字节的地址 (*rxStartBufferPointer*)。它适用于所有调制解调器。地址是相对于数据缓冲区的第一个字节的偏移量。

7. 使用以下命令读取数据缓冲区:

ReadBuffer (offset, payloadLengthRx)

其中 *offset* 等于 *rxStartBufferPointer*, *payloadLengthRx* 是要读取的缓冲区的大小。

8. 可选地, 可以从寄存器 0x0954 (MSB) 0x0955, 0x0956 (LSB) 读取频率错误指示符 (FEI)。FEI 表示为 20 位 2 的补码数。这必须从两个补码转换为带符号的 FEI 读数, 然后可以使用以下公式将其转换为以 Hz 为单位的频率误差:

$$\text{FrequencyError 赫兹} = 1.55 \text{ 倍} \frac{\text{SignedFeiReading}}{1600}$$

13.5 测距设置

基本测距操作是在配置为主无线电的单个无线电和配置为从属的单独 SX1280 之间交换特定格式的 LoRa 消息。

以下部分将介绍测距操作所需的 SX1280 配置设置。除非另有明确说明，否则必须在主无线电和从无线电上相同地再现这些配置步骤。

13.5.1 测距设备设置

主设备操作的范围特定设置将在下一节中给出。

1. 如果不处于 STDBY_RC 模式，则通过发送命令进入此模式：

SetStandby (STDBY_RC)

2. 通过发送命令将数据包类型设置为范围：

SetPacketType (PACKET_TYPE_RANGING)

3. 通过发送命令设置测距操作的调制参数：

SetModulationParams (modParamSF, modParamBW, modParamCR)

SetModulationParams 的三个参数的定义与 LoRa 设置的定义相同。

但是，对于测距操作，不允许使用 SF11 和 SF12。类似地，测距操作的带宽配置限于 406.25 kHz，812.5 kHz 和 1625 kHz 的值。

下表总结了 *SetModulationParams* 命令的可接受值（三个参数可以以任何方式组合）：

表 13-55：测距设备调制参数

modParamSF	modParamBW	modParamCR
LORA_SF_5	LORA_BW_400	LORA_CR_4_5
LORA_SF_6	LORA_BW_800	LORA_CR_4_6
LORA_SF_7	LORA_BW_1600	LORA_CR_4_7
LORA_SF_8		LORA_CR_4_8
LORA_SF_9		LORA_CR_4_5
LORA_SF_10		LORA_CR_4_6

4. 通过命令设置数据包参数：

SetPacketParams (preambleLength, headerType, payloadLength, crcMode, invertIq) 参

数的含义类似于 LoRa SetPacketParams 用法。

5. 设置命令使用的 RF 频率：

SetRfFrequency (rfFrequency)

rfFrequency 将作为多个 PLL 步骤（即 $52e6 / (2^{18})$ Hz）提供。SetRfFrequency () 定义 Tx 频率。
在 Rx 中，频率降低中频 (IF)。IF 默认设置为 1.65 MHz。

6. 通过以下方式设置 Tx 参数：

SetTxParams (txPower, rampTime)

7. 在测距操作期间，多个从设备和多个主设备可以在通信范围内。但是，测距操作必须仅使用一个从站和一个主站。为了帮助从设备区分主设备以响应范围内的其他主设备，并且为了寻址特定的从设备，测距请求包含地址字段，该设备在测距请求接收时由从设备检查。

仅在从站上，使用 WriteRegister 命令设置从站可以响应的地址。要写入的寄存器由下表给出：

表 13-56：从属测距请求地址定义

Slave Ranging req 地址	地址
RangingRangingAddress [31:24]	0x916
RangingRangingAddress [23:16]	0x917
RangingRangingAddress [15: 8]	0x918
RangingRangingAddress [7: 0]	0x919

从器件还需要通过发出 带有以下参数的 WriteRegister 命令来设置地址位的数量：

表 13-57：寄存器地址位定义

注册地址	领域	描述
0x931	7: 6	0x0: 8 位
		0x1: 16 位
		0x2: 24 位
		0x3: 位

Master 还需要使用相同的地址，因为这是发出测距请求的地址。它通过向 以下寄存器发出 WriteRegister 命令来设置：

表 13-58：主测距请求地址定义

主测距请求地址	地址
RangingRequestAddress [31:24]	0x912

RangingRequestAddress [23:16]	0x913
RangingRequestAddress [15: 8]	0x914
RangingRequestAddress [7: 0]	0x915

8. 使用以下命令设置应由无线电生成的 IRQ 进行测距操作：

```
SetDioIrqParams (irqMask, dio1Mask, dio2Mask, dio3Mask)
```

要激活的 IRQ 取决于 sx1280 的测距角色。对于 Master，典型的测距操作需要以下 IRQ：

- RangingMasterResultValid • RangingMasterResultTimeout

对于 Slave，典型的 IRQ 是：

- RangingSlaveResponseDone
- RangingSlaveRequestDiscarded

9. 测距过程需要校准值来补偿 RxTx 延迟偏移。通过在以下寄存器上调用 *WriteRegister* 命令来设置校准值：表

13-59：寄存器中的校准值

校准值	寄存器
校准[15: 8]	0x92C
校准[7: 0]	0x92D

10. 必须明确给出 SX1280 在测距操作中的作用，并发出以下命令：

```
SetRangingRole (role) 其中
```

角色值由下表提供：

表 13-60：范围角色值

范围角色	值
主	0x01
奴隶	为 0x00

11.最后，使用以下命令启动测距过程：

- 在 Slave 端：SetRx (periodBaseRx, periodCountRx)
- 在主方：SetTx (periodBaseTx, periodCountTx)

如果应用程序级别没有时序约束，建议使用连续模式（即 *PeriodCount* = 0xFFFF）。测距调制解调器自动管理从 Rx 到 Tx 到 Slave 的转换，以及从 Tx 到 Rx 到 Master 的转换。

12.测距结果只能从师父那里获得。当 Master 生成 IRQ RangingMasterResultValid 时，可以使用以下几种测距结果：

- 原始测距结果

- 平均测距结果
- 偏差的结果
- 过滤范围结果

通过阅读以下三个寄存器可以访问所有测距结果：

表 13-61：注册结果地址

注册地址	测距结果
0x961	RangingResult [23:16]
0x962	RangingResult [15： 8]
0x963	RangingResult [7： 0]

所有测距结果都存储在这三个寄存器中，并通过给定地址的 *ReadRegister* 命令读取。通过根据以下内容将寄存器 **RangingResMux** 的第 5 位和第 4 位设置在地址 0x924 来完成要读取的测距结果的选择：

表 13-62：测距结果类型选择

RangingResMux (5： 4)	测距输出类型	转换为距离
00	原始结果，它可能是负面的	距离[m] = RangingResult * 150 / (2 ^ 12 * BW) BW 为 MHz
01	平均结果，可以是负数	距离[m] = RangingResult * 20/100
10	偏离结果它不能是负面的	距离[m] = RangingResult * 20/100
11	过滤结果，不能是否定的	距离[m] = RangingResult * 20/100

由于测距结果寄存器的特殊用途，读取测距结果需要以下步骤：1。在 Xoscilator 模式下设置无线电

```
SetStandby (STDBY_XOSC) 2。
```

在 LoRa 内存中启用时钟：

```
WriteRegister (0x97F, ReadRegister (0x97F) | (1 << 1)) ;
```

3.像往常一样设置测距类型并读取测距寄存器：

```
WriteRegister (0x0924, (ReadRegister (0x0924) & 0xCF) | (((resultType) & 0x03) << 4)) ;
valLsb = ((ReadRegister (0x0961) << 16) | (ReadRegister (0x0962) << 8) | (ReadRegister (0x0963))) ;4.在配置模式下设置收音机：
```

```
SetStandby (STDBY_RC)
```

13.5.2 作为状态机的测距操作

可以使用以下简单状态机来总结测距操作：

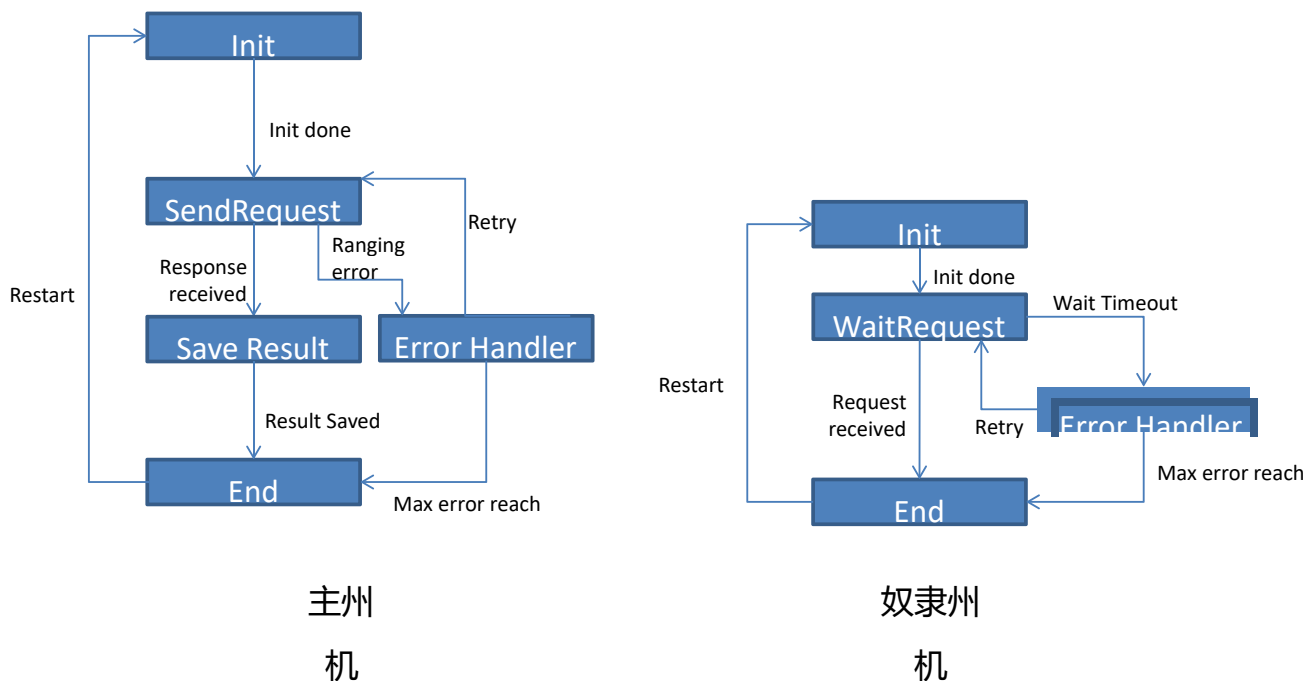


图 13-1: 测距状态机

这些状态机不代表芯片的内部行为，而是用户必须实施的步骤才能使用基本测距估计。

先前概述的步骤 1) 至 10) 中描述的无线电配置属于 Init 状态。

步骤 11) 中描述的 SetTx 和 SetRx 调用在状态 SendRequest 和 WaitRequest 中执行。

测距结果的读取在 Master 状态机的 SaveResult 状态下执行。它也是应用频率校正的地方。

注意，该频率校正需要事先评估从属侧的估计频率误差，这里没有表示。

错误处理程序状态取决于应用程序。在 RangingMasterResultTimeout 或 RangingSlaveRequestDiscarded IRQs 的情况下应该达到此状态。这里建议重试发送/接收测距请求，直到达到最大错误数。

注意：主机侧的测距响应接收和从机侧的响应广播由 SX1280 芯片自动处理。

13.6 其他功能

13.6.1 SetRegulatorMode 命令

默认情况下，仅使用 LDO。这在低成本应用中非常有用，其中 DC-DC 转换器所需的额外电感器的成本过高。仅使用 LDO 意味着 Rx 或 Tx 电流加倍。此命令允许用户指定 DC-DC 或 LDO 是否用于功率调节。

表 13-63：功率调节选择 SPI 数据传输

字节	0	1
来自主机的数据	操作码= 0x96	regModeParam

表 13-64：功率调节选择 UART 数据传输

字节	0	1	2
来自主机的数据	操作码= 0x96	0x01	regModeParam

表 13-65：RegModeParam 定义

RegModeParam 值	0	1
使用调节器	只有 LDO 用于所有模式	DC-DC 用于 STDBY_XOSC, FS, Rx 和 Tx 模式 LDO 用于 STDBY_RC

13.6.2 上下文保存

转换到休眠模式后，收发器寄存器的内容将丢失。无线电设备的配置可以利用自动恢复 的 *SetSaveContext ()* 命令。这将无线电寄存器值的当前上下文存储到协议引擎内的数据 RAM 中，以便在唤醒时进行恢复。操作顺序如下：

- 配置设备后，主机必须在 *SetSleep ()* 命令之前发送 *SaveContext ()*。
- 发出 *SetSleep ()* 命令后，必须保留数据 RAM。
- 当收发器从休眠状态唤醒时，它会检查数据 RAM 是否已保存，以及是否存在有效的寄存器转储（即已发出保存上下文命令）。如果满足两个条件，则将恢复寄存器。

请注意，在占空比循环操作中，会自动调用保存上下文。退出休眠模式时，用于有效负载数据的 Rx 缓冲区指针将重置为默认值 0x00。

表 13-66：SetSaveContext 数据传输

字节	0
来自主机的数据	操作码= 0xD5

SPI 和 UART 传输的 *SetSaveContext ()* 数据传输相同。

14.参考设计和应用原理图

14.1 参考设计

14.1.1 应用设计原理图

远程 2.4 GHz 应用电路如下所示：

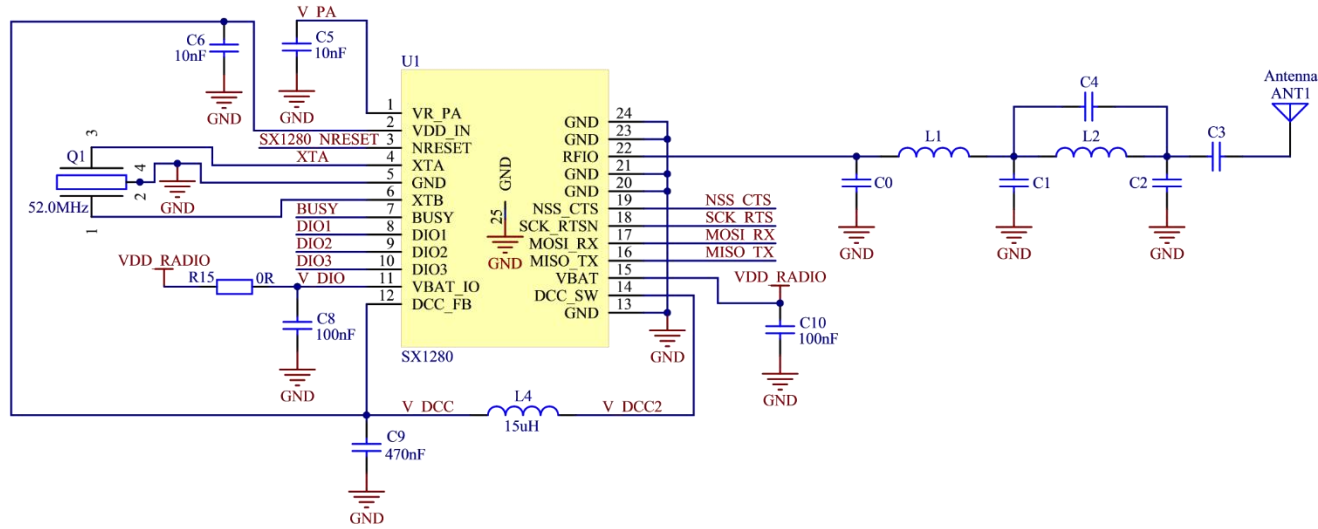


图 14-1：收发器应用设计原理图

14.1.2 参考设计 BOM

表 14-1：参考设计 BOM

参照标示 元件	MPN	GEOM	值	描述
U1	SX1280	VQFN24 4x4mm	SX1280	SX1280 2.4 GHz 高链路预算收发器 使用 LoRa 技术
R15	CRCW04020000Z0ED	0402	0Ω	厚膜电阻±1%，1 / 16W
C0	GRM1555C1HR80BA01D	0402	0.8 pF	多层陶瓷电容 C0G±0.1 pF，50 V.
C1	GRM1555C1H1R2BA01D	0402	1.2 pF	多层陶瓷电容 C0G±0.1 pF，50 V.
C2	GRM1555C1H1R2BA01D	0402	1.2 pF	多层陶瓷电容 C0G±0.1 pF，50 V.
C3	GRM1555C1H101JA01D	0402	100 pF	多层陶瓷电容 C0G±5%，50 V.
C4	GRM1555C1HR50WA01D	0402	0.5 pF	多层陶瓷电容器 C0G±0.05 pF，50 V.

C5	GRM155R71E103KA01D	0402	10 nF	多层陶瓷电容器 X7R±10%， 25 V
C6	GRM155R71E103KA01D	0402	10 nF	多层陶瓷电容器 X7R±10%， 25 V
C8	GRM155R71C104KA88D	0402	100 nF	多层陶瓷电容器 X7R±10%， 16 V
C9	GRM155R61A474KE15D	0402	470 nF	多层陶瓷电容器 X5R±10%， 10 V
C10	GRM155R71C104KA88D	0402	100 nF	多层陶瓷电容器 X7R±10%， 16 V
L1	LQW15AN3N0B80D	0402	3.0 nH	线绕电感器±0.1 nH
L2	LQW15AN2N5C00D	0402	2.5 nH	线绕电感器±0.2 nH
L4	MLZ2012M150W	0805	15μH	MLZ2012 多层屏蔽电感±5%
Q1	NX2016SA-52.000000MHZ	NX2016SA	52.000 MHz	水晶单元，NDK 参考号：EXS00A-CS07103， 托尔。±10 ppm，Load = 10 pF

14.1.3 参考设计 PCB

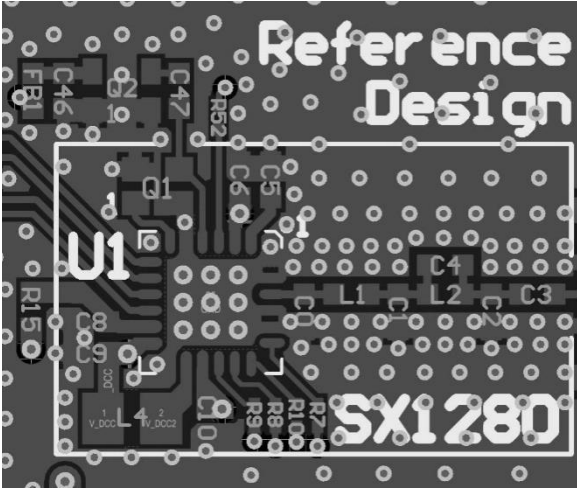


图 14-2：远程参考设计 PCB 布局

14.2 使用可选 TCXO 的应用程序设计

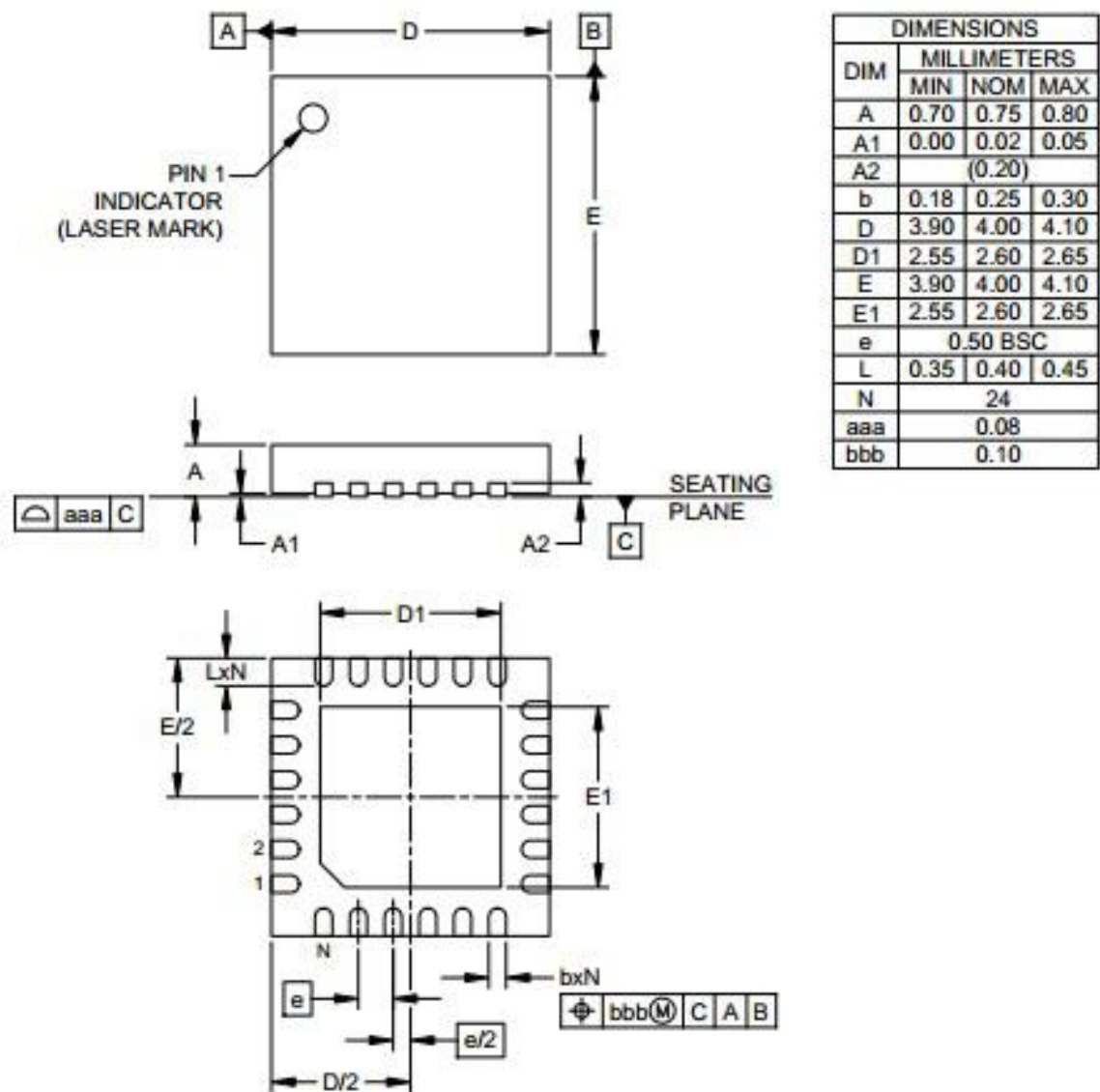
尽管收发器的调制参数被设计为容许与使用工业标准容差晶体参考振荡器组件相关的典型频率漂移，但是可以使用外部温度补偿晶体振荡器（TCXO）。下图显示了 TCXO 应如何交流耦合到收发器的 XTA 输入。有关 TCXO 特定组件的详细信息，请咨询晶体制造商。

NSS	逻辑'1'
SCK	逻辑'0'
MOSI	逻辑“1”或“0”
味噌	高阻抗
NRESET	逻辑'1'
忙	产量
DIO1	产量
DIO2	产量
DIO3	产量

15.包装信息

15.1 封装外形图

该收发器采用 4x4mm QFN 封装，间距为 0.5mm：



- NOTES:
1. CONTROLLING DIMENSIONS ARE IN MILLIMETERS (ANGLES IN DEGREES).
 2. COPLANARITY APPLIES TO THE EXPOSED PAD AS WELL AS THE TERMINALS.

图 15-1：QFN 4x4 封装外形图

15.2 土地格局

推荐的土地格局如下：

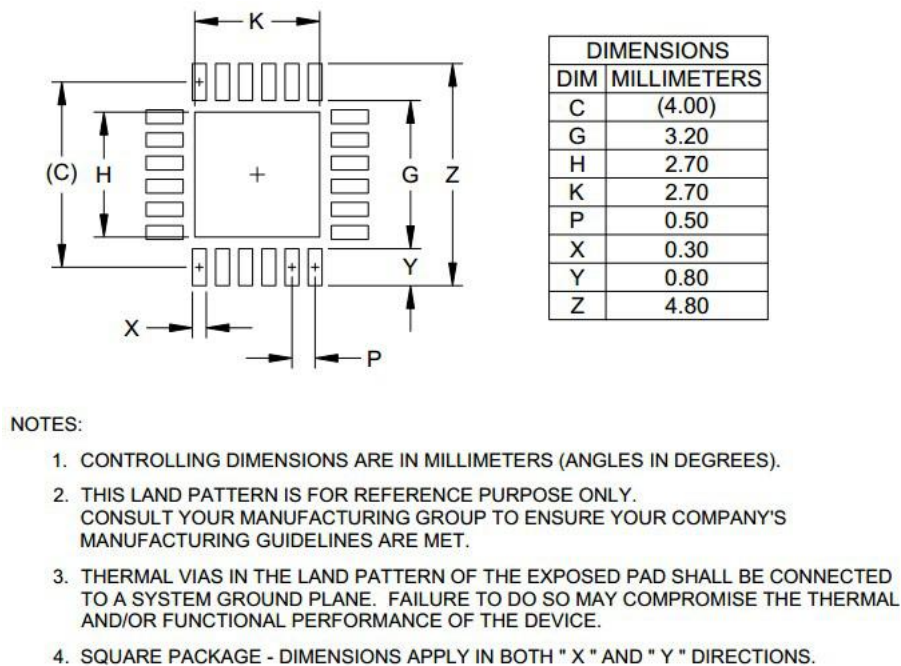


图 15-2：QFN 4x4mm 焊盘图案

15.3 回流曲线

回收过程说明可从 Semtech 网站获得，地址如下：http://www.semtech.com/quality/ir_reflow_profiles.html

该收发器采用 QFN24 4x4mm 封装，也称为 MLP 封装。

15.4 热阻

封装 QFN 24L 4X4 E-pad（器件：SX12801MLTRT）安装在 4 层 JEDEC PCB 上，在静止空气中具有 9 个散热通孔，能够在 25°C 的环境温度下耗散所需的功率 0.20W，同时保持最大结点模具温度低于 125°C。

θ_{JA} 在静止空气中相应的包的被计算为 50.3°C / W。

ψ 相应封装的 JT 计算为 0.3°C / W。

θ_{JC} 相应的包的被计算为 30.0°C / W。

θ_{JB} 对应的包的被计算为 13.3°C / W。

词汇表

首字母缩略词列表及其含义

缩写	含义
ACR	相邻信道拒绝
ADC	模数转换器
AFC	自动频率校正
AGC	自动增益控制
API	应用程序接口
β	调制指数
BLE	低功耗蓝牙
BR	比特率
BT	带宽 - 时间位周期乘积
BW	带宽
CAD	频道活动检测
CMD	命令交易
CPOL	时钟极性
CPHA	时钟相位
CR	编码率
CRC	循环冗余检查
CW	连续波
DIO	数字输入/输出
FEC	前向纠错
FLRC	快速远程通信
FSK	频移键控
GFSK	高斯频移键控
GMSK	高斯最小频移键控
IRQ	中断请求
我愿意	低压差
LLID	逻辑链路标识符
LNA	低噪声放大器
LO	本地振荡器

首字母缩略词列表及其含义

缩写	含义
LSB	最重要的一点
MD	更多数据
MIC	消息完整性检查
味噌	主输入从输出
MOSI	主输出从输入
MSB	最重要的一点
MSK	最小班次键控
NESN	下一个预期序列号
NOP	没有操作
NRZ	不归零
NSS	从机选择低电平有效
OOK	开关键控
PA	功率放大器
PDU	协议数据单元
每	数据包错误率
PID	产品标识
PLL	锁相环
PRNG	伪随机数生成
RFU	保留供将来使用
RTC	实时时钟
RTSN	请求发送
SCK	串行时钟
SF	传播因素
SN	序列号
SNR	信噪比
SPI	串行外设接口
STDBY	支持
TCXO	温度补偿晶体振荡器
UART	通用异步接收器/发送器

XOSC

晶体振荡器



重要的提醒

与本产品和本文所述的应用或设计相关的信息被认为是可靠的，但是此类信息仅作为指南提供，Semtech 对本文档中的任何错误或本文所述的应用或设计不承担任何责任。Semtech 保留随时更改产品或本文档的权利，恕不另行通知。买方在下订单前应获取最新的相关信息，并应验证此类信息是否最新且完整。Semtech 保证其产品的性能符合销售时适用的规格，所有销售均按照 Semtech 的标准销售条款和条件进行。

SEMTECH 产品并非设计，预期，授权或保证适合在生命支持应用程序，设备或系统中使用，或在核应用程序中使用，因为这些应用程序可能会导致失败，导致人身伤害，生命损失或严重财产损失或环境损害。在这些应用中包含 SEMTECH 产品的行为完全由客户自行承担。如果客户购买或使用 Semtech 产品进行任何此类未经授权的应用，客户应赔偿并使 Semtech 及其高级职员，员工，子公司，关联公司和经销商免受可能产生的所有索赔，成本损失和律师费的损害。

Semtech 名称和徽标是 Semtech Corporation 的注册商标。提及的所有其他商标和商品名称可能是 Semtech 或其各自公司的商标和名称。Semtech 保留更改或终止本文档中描述的任何产品的权利，恕不另行通知。Semtech 对其产品是否适用于任何特定用途不做任何明示或暗示的保证，陈述或保证。版权所有。

©Semtech 2017

联系信息

Semtech 公司

无线和传感产品

200 Flynn Road, Camarillo, CA 93012

电话：(805) 498-2111, 传真：(805) 498-3804

www.semtech.com