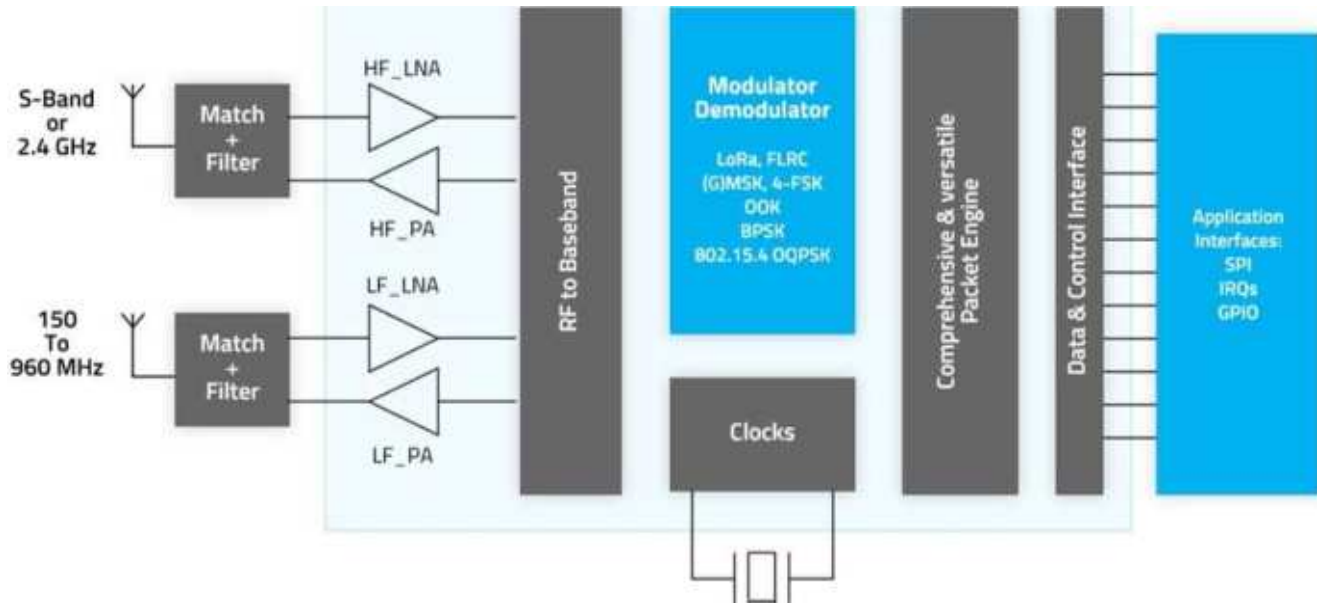




## LR2021/LR2022/LR2012 Datasheet

Long Range, Low Power, Multi-band LoRa® Transceiver



### Characteristics

- Receiver (Rx) Performance
  - LoRa® sub-GHz: -141.5 dBm @LoRa® 125 kHz
  - Bluetooth® LE PHY<sup>1, 2</sup>: -100 dBm @1 Mbps  
-97 dBm @2 Mbps
  - FLRC sub-GHz<sup>1</sup>: -111 dBm @260 kbps  
-104.5 dBm @0.975 Mbps  
-100.5 dBm @1.95 Mbps
  - O-QPSK 802.15.4 2.4 GHz<sup>1</sup>: -103 dBm @250 kbps DSSS
  - O-QPSK sub-GHz<sup>1</sup>: -111 dBm @100 kbps DSSS
- Transmitter (Tx) Performance
  - Up to +22 dBm @sub-GHz
  - Up to +12 dBm @2.4 GHz<sup>1, 2</sup>
  - 32 dB dynamic in 0.5 dB steps
- Low current consumption
  - Rx current 5.7 mA @sub-GHz
  - Tx current 105 mA @+22 dBm  
27 mA @+14 dBm  
12 mA @+10 dBm sub-GHz
- Low sleep current 470 nA
- 40 to +85 °C
- 1.8 to 3.7 V supply

1. LR2021 unique feature  
2. LR2022 unique feature

### Packaging and Ordering

- 5mm x 5mm QFN package
- MOQ 3000 pieces, Tape & Reel

### Generation 4 LoRa® IP

- Multi-SF receiver
- Multi-region front-end
- Improved CAD
- Increased frequency offset tolerance

### PHY Compatibility for Protocols

- LoRa / LoRaWAN® at 2.4 GHz<sup>1, 2</sup> and sub-GHz
- Amazon Sidewalk
- Sigfox BPSK
- Wi-SUN FSK
- Wireless M-BUS (device and collector)
- Bluetooth® LE 5.0 PHY compatible<sup>1, 2</sup>
- 2-FSK based protocols
- Z-Wave®, Z-Wave LR<sup>1</sup>
- IEEE® 802.15.4™ Thread® and Zigbee™ O-QPSK<sup>1</sup>

### Key Features

- 150–960 MHz bands
- LoRa up to 125 kbps
- 1.5–2.5 GHz bands<sup>1, 2</sup>
- FLRC up to 2.6 Mbps<sup>1</sup>
- LR-FHSS for SATCOM<sup>1, 2</sup> and sub-GHz

## Applications

- Security and building automation
- Smart city and Logistics
- Smart utility
- Smart Home
- Industrial Automation
- Smart Agriculture

## Regulatory Compliance

- ETSI EN 300 220-1
- FCC Part 15.247/249
- All LoRaWAN supported regions
- Improved Pout for STD-T108

## General Description

The LR2021 is Semtech's first chip in the LoRa Plus™ family. Incorporating a fourth-generation LoRa IP, the transceiver supports both Terrestrial and SATCOM networks in the sub-GHz, 2.4 GHz ISM bands, and licensed L/S-bands. The transceiver is designed to be backward compatible with previous LoRa devices, ensuring LoRaWAN compatibility.

The device additionally features expanded physical layer modulations for fast long-range communication (FLRC) and is compatible with various low-power wireless protocols including Amazon Sidewalk, Wireless M-BUS, Wi-SUN FSK, and Z-Wave when integrated with third-party stack offerings.

The LR2021 features an RF and analog architecture that supports a multi-region, single-SKU design, reducing external BOM costs, PCB footprint, and power consumption, while offering improved performance compared to previous LoRa transceivers.

This document describes the functionality of the LR2021, as well as that of the entire product family including the LR2022 and LR2012 devices, defined as follows:

| Pin-2-Pin Compatible                             | LR2021<br>Multi-Protocol | LR2022<br>LoRaWAN / Dual-band | LR2012<br>LoRaWAN® / Sub-GHz |
|--------------------------------------------------|--------------------------|-------------------------------|------------------------------|
| Sub-GHz LoRa   Wi-SUN FSK<br>wM-Bus   2 Mbps FSK | ✓                        | ✓                             | ✓                            |
| Sub-GHz RTTof (Ranging)                          | ✓                        | ✓                             | ✓                            |
| Sub-GHz FLRC 2.6 Mbps                            | ✓                        | —                             | —                            |
| 2.4GHz LoRa                                      | ✓                        | ✓                             | —                            |
| 2.4GHz RTTof (Ranging)                           | ✓                        | ✓                             | —                            |
| 2.4GHz FLRC 2.6 Mbps                             | ✓                        | —                             | —                            |
| SATCOM S-band <sup>1</sup> (LR-FHSS)             | ✓                        | ✓                             | —                            |
| ZigBee   Z-Wave   Thread                         | ✓                        | —                             | —                            |

| If You Need                               | Select    |
|-------------------------------------------|-----------|
| FLRC, Z-Wave, Zigbee,<br>Thread, 802.15.4 | LR2021    |
| L/S-band, 2.4GHz                          | LR2021/22 |
| Otherwise                                 | LR2012    |

Product features unavailable in the LR2022 and LR2012 are described in this document but are not available in the corresponding IC.

## Ordering codes

| P/N    | Ordering Code | MOQ             | Features                                                                              |
|--------|---------------|-----------------|---------------------------------------------------------------------------------------|
| LR2021 | LR2021IMLTRT  | T&R/3000 pieces | Full feature set, dual band                                                           |
| LR2022 | LR2022IMLTRT  | T&R/3000 pieces | Same as LR2021, excluding support for FLRC, O-QPSK (802.15.4), and Z-Wave             |
| LR2012 | LR2012IMLTRT  | T&R/3000 pieces | Sub-GHz only, excluding support for FLRC, O-QPSK (802.15.4), Bluetooth LE, and Z-Wave |

QFN 32 package, Pb-free, Halogen-free, RoHS/WEEE compliant product.

## DISCLAIMER 1

**Long Range-Frequency Hopping Spread Spectrum** (LR-FHSS) is a high link-budget, high-performance technology combining the benefits of a modulation employing low energy per bit and advanced frequency hopping schemes to achieve improved coexistence, spectral efficiency and sensitivity. Semtech Corp. holds patents directed to aspects of the LR-FHSS technology.

Your use of LR-FHSS software made available by Semtech Corp. or its affiliates does not grant any rights to their patents for LR-FHSS technology. Rights under Semtech patents may be available via various mechanisms, including by purchasing Semtech SX1261, SX1262, SX1268, LR1110, LR1120, LR1121, LR2012, LR2021 or LR2022 semiconductor devices, or their authorized counterparts from Semtech, or its affiliates, or their respective licensees.

## DISCLAIMER 2

Semtech's products are designed to be used in connection with qualified **Bluetooth®** products and applications but are not certified or qualified Bluetooth products.

## Revision History

| Version | Date         | Changes and/or Modifications                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0.2     | June 2025    | Preliminary release                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 1.1     | October 2025 | Final datasheet: Clarifications throughout                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 2.0     | March 2026   | <p>Expanded LR2021 datasheet to cover LR2022 and LR2012 specifications, also modified:</p> <ul style="list-style-type: none"> <li>• <a href="#">Figure 1-9</a> description, to specify +12 dBm is the power setting.</li> <li>• FLRC BW values in <a href="#">Table 18-1</a>, <a href="#">Table 18-3</a> and <a href="#">Section 18.4.1</a></li> <li>• <a href="#">Section 9.3</a></li> <li>• Added commands: <a href="#">9.9.12 SetLoraTxSync</a></li> <li>• Added <a href="#">11.1.3 (G)FSK Limitations</a></li> <li>• Modified <a href="#">Figure 23-1: LR2021/LR2022 Reference Design Schematic</a></li> <li>• Added expected degradation using SIMO with metallic RF shield in <a href="#">Section 23.1</a></li> <li>• Electrical spec conditions have been clarified to "no shield, SIMO"</li> <li>• Modified IDDSL1 value in <a href="#">Table 3-4</a></li> <li>• Modified BRFLRC Max value in <a href="#">Table 3-8</a></li> <li>• Modified ZW_R2_S, ZW_R3_S, ZWLR1 Typ values in <a href="#">Table 3-10</a></li> <li>• Added 0.5 dB to all sensitivity numbers in <a href="#">Table 3-13</a> and <a href="#">Table 3-18</a></li> <li>• Added <a href="#">Section 22.3 Firmware Patch RAM (PRAM)</a></li> </ul> |

| Version | Date       | Changes and/or Modifications (Continued)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|---------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2.1     | April 2026 | <p>Modified Expected degradation values in <a href="#">Table 23-1</a></p> <p>Modified footnote in <a href="#">Table 18-2</a></p> <p>Added FLRC_260_CR05_915_S effective bit rate in <a href="#">Table 3-12</a></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 2.2     | July 2026  | <p>Removed 4-FSK from “PHY compatibility for protocols” on front page</p> <p>Added freq_offset to <a href="#">9.9.9 GetLoraPacketStatus</a> and <a href="#">18.4.4 GetFlrcPacketStatus</a></p> <p>Added load pull specifications to <a href="#">Table 3-22</a></p> <p>Modified <a href="#">7.3.4 SetRssiCalibration</a></p> <p>Added <a href="#">Table 7-6: Recommended Values for Reference EVK HF path 2.4 GHz (loaded by default)</a>, <a href="#">Table 7-7: Recommended Values for Reference EVK LF path 868–915 MHz (loaded by default)</a>, and <a href="#">Table 7-8: Recommended Values for Reference EVK LF path 490 MHz (loaded by the application)</a></p> <p>Updated PA optimal values in <a href="#">Table 7-19</a>, <a href="#">Table 7-20</a>, <a href="#">Table 7-21</a></p> <p>Added <a href="#">5.3.3 Extending the Data FIFOs</a></p> <p>Removed “N 6.4 kbps” from <a href="#">12.1 WM-BUS Modulation</a></p> <p>Modified <a href="#">9.6 Fast CAD</a></p> <p>Modified <a href="#">6.5.2 GetTemp</a></p> <p>Modified <a href="#">23.4 Reference Design</a></p> <p>Modified <a href="#">Table 23-2: Register Usage</a></p> <p>Modified <a href="#">19.3.1 SetOqpskParams</a></p> <p>Modified <a href="#">20.3.4 GetZwavePacketStatus</a></p> <p>Added note to <a href="#">Table 3-12: FLRC Sub-GHz 1% PER (LR2021)</a>, <a href="#">Table 3-13: FLRC 2.4 GHz 1% PER (Packet Size = 31 bytes) (LR2021)</a></p> <p>Modified <a href="#">18.1 FLRC Modulation</a>, <a href="#">Table 18-3: Receiver Performance of the FLRC Modem</a>, <a href="#">18.1.3 Gaussian Filtering</a>, <a href="#">18.4.1 SetFlrcModulationParams</a>,</p> <p>Modified link budget improvement value in <a href="#">18.1 FLRC Modulation</a></p> <p>Added <a href="#">22.4 Regulatory Compliance</a></p> <p>Modified LDRO activation criteria in <a href="#">9.9.1 SetLoraModulationParams</a></p> |

## Contents

|                                                                |    |
|----------------------------------------------------------------|----|
| 1. Chip Description .....                                      | 20 |
| 1.1 LR2021 .....                                               | 20 |
| 1.2 LR2022 .....                                               | 21 |
| 1.3 LR2012 .....                                               | 22 |
| 1.4 Receiver RF Path .....                                     | 23 |
| 1.5 Transmitter RF Path .....                                  | 24 |
| 1.5.1 Low Frequency Power Amplifier (LF PA) (LR20xx) .....     | 25 |
| 1.5.2 High Frequency Power Amplifier (HF PA) (LR2021/22) ..... | 28 |
| 1.5.3 Direct-Tie Implementation (LR20xx) .....                 | 29 |
| 1.6 Frequency Synthesizer .....                                | 29 |
| 1.7 Modem .....                                                | 30 |
| 1.8 Power Distribution .....                                   | 31 |
| 1.8.1 Power-On Reset and Brown-Out Reset .....                 | 31 |
| 1.8.2 Low Battery Detector .....                               | 31 |
| 1.8.3 Over Current Protection .....                            | 31 |
| 1.8.4 Power Supply Mode .....                                  | 32 |
| 1.9 Clock Sources .....                                        | 33 |
| 1.9.1 32 kHz RC Oscillator .....                               | 33 |
| 1.9.2 32 MHz Crystal .....                                     | 33 |
| 1.9.3 32 MHz TCXO .....                                        | 34 |
| 1.9.4 External Clock Reference Inputs .....                    | 34 |
| 1.9.5 Output Clock .....                                       | 34 |
| 2. Pin Connection .....                                        | 35 |
| 2.1 I/O Description .....                                      | 35 |
| 3. Specifications (LR20xx) .....                               | 38 |
| 3.1 Absolute Maximum Ratings .....                             | 38 |
| 3.2 Operating Range .....                                      | 38 |
| 3.3 ESD Protection and Latch-up .....                          | 39 |
| 3.4 Electrical Specifications .....                            | 39 |
| 3.4.1 Power Consumption .....                                  | 40 |
| 3.4.2 General Specifications .....                             | 43 |
| 3.4.3 Receive Mode Specifications .....                        | 45 |
| 3.4.4 Transmit Mode Specifications .....                       | 67 |
| 3.5 Reference Oscillator Crystal Specification .....           | 70 |
| 3.6 SIMO Inductor Specification .....                          | 71 |
| 3.7 Digital I/O and Interface Specifications .....             | 71 |
| 3.7.1 Digital I/O Specifications .....                         | 71 |
| 3.7.2 SPI Interface Specifications .....                       | 72 |
| 4. Chip Modes (LR20xx) .....                                   | 73 |
| 4.1 Chip Mode State Diagram .....                              | 73 |

|                                                                                |     |
|--------------------------------------------------------------------------------|-----|
| 4.2 Reset .....                                                                | 75  |
| 4.3 Start-up and Boot Calibration Sequence .....                               | 75  |
| 4.4 Chip Modes .....                                                           | 76  |
| 4.4.1 Sleep .....                                                              | 76  |
| 4.4.2 Standby RC / XOSC.....                                                   | 76  |
| 4.4.3 Frequency Synthesis (FS) .....                                           | 76  |
| 4.4.4 Receive (Rx) .....                                                       | 76  |
| 4.4.5 Transmit (Tx).....                                                       | 77  |
| 5. Host Interface .....                                                        | 78  |
| 5.1 Hardware Interface (LR20xx) .....                                          | 78  |
| 5.1.1 DIO Pins .....                                                           | 78  |
| 5.1.2 SPI Interface.....                                                       | 78  |
| 5.2 Busy Line Control (LR20xx) .....                                           | 78  |
| 5.3 Data FIFO (LR20xx) .....                                                   | 79  |
| 5.3.1 Rx Data FIFO .....                                                       | 79  |
| 5.3.2 Tx Data FIFO.....                                                        | 79  |
| 5.3.3 Extending the Data FIFOs .....                                           | 80  |
| 5.4 Host-Command Interface (LR20xx) .....                                      | 81  |
| 5.4.1 Read Write Command .....                                                 | 81  |
| 5.5 DIO Rx / Tx Triggers (LR20xx) .....                                        | 84  |
| 5.6 Host Commands Summary .....                                                | 85  |
| 5.6.1 Direct Radio FIFO Read/Write Commands (LR20xx).....                      | 85  |
| 5.6.2 Register / Memory Access Operations (LR20xx) .....                       | 85  |
| 5.6.3 System Configuration / Status Operations (LR20xx) .....                  | 86  |
| 5.6.4 Radio Configuration / Status Operations (LR20xx except FLRC Commands)... | 88  |
| 5.7 Host Interrupts (LR20xx) .....                                             | 99  |
| 6. Chip Control Commands (LR20xx) .....                                        | 101 |
| 6.1 Direct Radio FIFO Read/Write .....                                         | 101 |
| 6.1.1 ReadRadioRxFifo.....                                                     | 101 |
| 6.1.2 WriteRadioTxFifo .....                                                   | 101 |
| 6.2 Register / Memory Access Operations .....                                  | 102 |
| 6.2.1 WriteRegMem32 .....                                                      | 102 |
| 6.2.2 WriteRegMemMask32 .....                                                  | 102 |
| 6.2.3 ReadRegMem32 .....                                                       | 103 |
| 6.3 Chip Mode Control .....                                                    | 103 |
| 6.3.1 SetSleep.....                                                            | 103 |
| 6.3.2 SetStandby.....                                                          | 104 |
| 6.3.3 SetFs .....                                                              | 104 |
| 6.3.4 SetAdditionalRegToRetain.....                                            | 104 |
| 6.3.5 SetRx.....                                                               | 105 |
| 6.3.6 SetTx .....                                                              | 106 |
| 6.3.7 SetRxTxFallbackMode .....                                                | 106 |
| 6.3.8 SetRxDutyCycle.....                                                      | 107 |
| 6.3.9 SetAutoRxTx .....                                                        | 108 |

|                                              |     |
|----------------------------------------------|-----|
| 6.3.10 StopTimeoutOnPreamble .....           | 108 |
| 6.3.11 SetLoraCadParams.....                 | 109 |
| 6.3.12 SetLoraCAD.....                       | 110 |
| 6.3.13 SetCad.....                           | 111 |
| 6.3.14 SetCadParams.....                     | 111 |
| 6.3.15 SetCca .....                          | 112 |
| 6.3.16 ResetRxStats .....                    | 112 |
| 6.3.17 SetDefaultRxTxTimeout.....            | 112 |
| 6.3.18 SetRegMode .....                      | 113 |
| 6.4 Chip Auto Calibration .....              | 114 |
| 6.4.1 Calibrate .....                        | 114 |
| 6.4.2 CalibFE .....                          | 115 |
| 6.5 Measurements .....                       | 116 |
| 6.5.1 GetVbat .....                          | 116 |
| 6.5.2 GetTemp.....                           | 117 |
| 6.6 Low Battery Detector Configuration ..... | 118 |
| 6.6.1 SetLbdCfg.....                         | 118 |
| 6.6.2 GetRandomNumber .....                  | 119 |
| 6.7 Chip Status .....                        | 120 |
| 6.7.1 GetStatus.....                         | 120 |
| 6.7.2 GetVersion.....                        | 121 |
| 6.7.3 ClearErrors.....                       | 121 |
| 6.7.4 GetErrors .....                        | 122 |
| 6.8 DIO Control .....                        | 123 |
| 6.8.1 SetDioFunction .....                   | 123 |
| 6.8.2 SetDioRfSwitchConfig.....              | 124 |
| 6.8.3 SetDioIrqConfig.....                   | 124 |
| 6.9 Interrupt Control .....                  | 125 |
| 6.9.1 ClearIrq .....                         | 125 |
| 6.9.2 GetAndClearIrqStatus.....              | 125 |
| 6.10 Data FIFO .....                         | 126 |
| 6.10.1 ConfigFifoIrq .....                   | 126 |
| 6.10.2 GetFifoIrqFlags .....                 | 127 |
| 6.10.3 ClearFifoIrqFlags.....                | 127 |
| 6.10.4 GetAndClearFifoIrqFlags.....          | 128 |
| 6.10.5 GetRxFifoLevel.....                   | 128 |
| 6.10.6 GetTxFifoLevel .....                  | 129 |
| 6.10.7 ClearRxFifo.....                      | 129 |
| 6.10.8 ClearTxFifo .....                     | 129 |
| 6.11 Clock Control .....                     | 130 |
| 6.11.1 ConfigLfClock .....                   | 130 |
| 6.11.2 ConfigClkOutputs.....                 | 130 |
| 6.11.3 SetTcxoMode.....                      | 131 |
| 6.11.4 SetXoscCpTrim.....                    | 132 |

|                                                                    |     |
|--------------------------------------------------------------------|-----|
| 6.12 Temperature Compensation .....                                | 132 |
| 6.12.1 SetTempCompCfg .....                                        | 133 |
| 6.12.2 SetNtcParams .....                                          | 133 |
| 7. Radio Frequency Front End Control .....                         | 134 |
| 7.1 Overview (LR2021/LR2022) .....                                 | 134 |
| 7.2 Overview (LR2012) .....                                        | 134 |
| 7.3 RF Front End Commands (LR20xx) .....                           | 134 |
| 7.3.1 SetRfFrequency .....                                         | 134 |
| 7.3.2 SetRxPath .....                                              | 135 |
| 7.3.3 GetRssiInst .....                                            | 135 |
| 7.3.4 SetRssiCalibration .....                                     | 136 |
| 7.3.5 ResetRxStats .....                                           | 139 |
| 7.3.6 SetTimestampSource .....                                     | 139 |
| 7.3.7 GetTimestampValue .....                                      | 140 |
| 7.3.8 GetCcaResult .....                                           | 140 |
| 7.3.9 SetAgcGainManual .....                                       | 141 |
| 7.3.10 GetRxPktLength .....                                        | 141 |
| 7.4 Power Amplifier Control Commands (LR20xx) .....                | 142 |
| 7.4.1 SetPaConfig .....                                            | 142 |
| 7.4.2 SelPa .....                                                  | 146 |
| 7.4.3 SetTxParams .....                                            | 147 |
| 8. Modem Configuration (LR20xx) .....                              | 148 |
| 8.1 Packet Type Selection .....                                    | 149 |
| 8.1.1 SetPacketType .....                                          | 149 |
| 8.1.2 GetPacketType .....                                          | 150 |
| 9. LoRa Modulation (LR20xx) .....                                  | 151 |
| 9.1 LoRa Modulation Principle .....                                | 151 |
| 9.1.1 Spreading Factor (SF) .....                                  | 151 |
| 9.1.2 LoRa Bandwidth (BWLORA) .....                                | 151 |
| 9.1.3 Coding Rate (CR) .....                                       | 151 |
| 9.1.4 Low Data Rate Optimization (LDRO) .....                      | 151 |
| 9.1.5 LoRa Symbol Rate (Rs) .....                                  | 152 |
| 9.2 LoRa Packet Handler .....                                      | 152 |
| 9.2.1 Preamble .....                                               | 152 |
| 9.2.2 Header (explicit packets only) .....                         | 152 |
| 9.2.3 Payload .....                                                | 153 |
| 9.3 Multi-SF Detection .....                                       | 153 |
| 9.4 Channel Activity Detection (CAD) .....                         | 154 |
| 9.5 Multi-SF CAD .....                                             | 154 |
| 9.6 Fast CAD .....                                                 | 155 |
| 9.7 Extended LoRa Rx Bandwidth .....                               | 155 |
| 9.8 Intra-packet Frequency Hopping Compatibility with SX127x ..... | 156 |
| 9.9 LoRa Commands .....                                            | 157 |
| 9.9.1 SetLoraModulationParams .....                                | 157 |



|                                                      |     |
|------------------------------------------------------|-----|
| 9.9.2 SF6 Compatibility with SX127x .....            | 158 |
| 9.9.3 SetLoraPacketParams .....                      | 159 |
| 9.9.4 SetLoRaSynchTimeout .....                      | 160 |
| 9.9.5 SetLoraSyncword .....                          | 160 |
| 9.9.6 SetLoraSideDetConfig .....                     | 161 |
| 9.9.7 SetLoraSideDetSyncword .....                   | 161 |
| 9.9.8 GetLoraRxStats .....                           | 162 |
| 9.9.9 GetLoraPacketStatus .....                      | 163 |
| 9.9.10 SetLoraAddress .....                          | 164 |
| 9.9.11 SetLoraHopping .....                          | 164 |
| 9.9.12 SetLoraTxSync .....                           | 165 |
| 9.9.13 SetLoraSideDetCad .....                       | 165 |
| 10. RTToF (Round-Trip Time of Flight) (LR20xx) ..... | 166 |
| 10.1 RTToF Engine .....                              | 166 |
| 10.2 RTToF Commands .....                            | 166 |
| 10.2.1 SetRangingAddr .....                          | 166 |
| 10.2.2 SetRangingReqAddr .....                       | 166 |
| 10.2.3 GetRangingResult .....                        | 167 |
| 10.2.4 GetRangingStats .....                         | 168 |
| 10.2.5 SetRangingTxRxDelay .....                     | 169 |
| 10.2.6 SetRangingParams .....                        | 169 |
| 11. (G)FSK Modulation (LR20xx) .....                 | 170 |
| 11.1 (G)FSK Modulation Principle .....               | 170 |
| 11.1.1 Modem Bandwidth and Bit Rates .....           | 170 |
| 11.1.2 (G)FSK Modulation Index .....                 | 171 |
| 11.1.3 (G)FSK Limitations .....                      | 171 |
| 11.2 (G)FSK Packet Handler .....                     | 172 |
| 11.2.1 Fixed-Length Packet .....                     | 172 |
| 11.2.2 Variable-Length Packet .....                  | 172 |
| 11.3 (G)FSK Commands .....                           | 173 |
| 11.3.1 SetFskModulationParams .....                  | 173 |
| 11.3.2 SetFskPacketParams .....                      | 175 |
| 11.3.3 SetFskWhiteningParams .....                   | 178 |
| 11.3.4 SetFskCrcParams .....                         | 178 |
| 11.3.5 SetFskSyncword .....                          | 179 |
| 11.3.6 SetFskAddress .....                           | 179 |
| 11.3.7 GetFskRxStats .....                           | 180 |
| 11.3.8 GetFskPacketStatus .....                      | 181 |
| 12. Wireless M-Bus (WM-BUS) (LR20xx) .....           | 182 |
| 12.1 WM-BUS Modulation .....                         | 182 |
| 12.2 WM-BUS Packet Handler .....                     | 183 |
| 12.3 WM-BUS Radio Commands .....                     | 183 |
| 12.3.1 SetWmbusParams .....                          | 183 |
| 12.3.2 GetWmbusRxStats .....                         | 184 |

|        |                                                           |     |
|--------|-----------------------------------------------------------|-----|
| 12.3.3 | GetWmbusPacketStatus .....                                | 185 |
| 12.3.4 | SetWmbusFilteringAddress .....                            | 186 |
| 13.    | Wi-SUN FSK (LR20xx) .....                                 | 187 |
| 13.1   | Wi-SUN Modulation .....                                   | 187 |
| 13.2   | Wi-SUN Packet Handler .....                               | 187 |
| 13.3   | Wi-SUN Radio Commands .....                               | 188 |
| 13.3.1 | SetWisunMode .....                                        | 188 |
| 13.3.2 | SetWisunPacketParams .....                                | 189 |
| 13.3.3 | GetWisunRxStats .....                                     | 190 |
| 13.3.4 | GetWisunPacketStatus .....                                | 191 |
| 13.3.5 | SetWisunPacketLen .....                                   | 191 |
| 14.    | Bluetooth Low Energy Physical Layer (LR2021/LR2022) ..... | 192 |
| 14.1   | Bluetooth LE Modulation .....                             | 192 |
| 14.2   | Bluetooth LE Packet Handler .....                         | 192 |
| 14.3   | Bluetooth LE Radio Commands .....                         | 192 |
| 14.3.1 | SetBleModulationParams .....                              | 192 |
| 14.3.2 | SetBleChannelParams .....                                 | 193 |
| 14.3.3 | SetBleTxPduLen .....                                      | 193 |
| 14.3.4 | SetBleTx .....                                            | 194 |
| 14.3.5 | GetBleRxStats .....                                       | 194 |
| 14.3.6 | GetBlePacketStatus .....                                  | 195 |
| 15.    | BPSK Modem (LR20xx) .....                                 | 196 |
| 15.1   | (D)BPSK Modulation .....                                  | 196 |
| 15.2   | BPSK Radio Commands .....                                 | 196 |
| 15.2.1 | SetBpskModulationParams .....                             | 196 |
| 15.2.2 | SetBpskPacketParams .....                                 | 197 |
| 16.    | OOK Modem (LR20xx) .....                                  | 198 |
| 16.1   | OOK Modulation .....                                      | 198 |
| 16.2   | OOK Packet Handler .....                                  | 198 |
| 16.3   | OOK Commands .....                                        | 198 |
| 16.3.1 | SetOokModulationParams .....                              | 198 |
| 16.3.2 | SetOokPacketParams .....                                  | 199 |
| 16.3.3 | SetOokCrcParams .....                                     | 200 |
| 16.3.4 | SetOokSyncword .....                                      | 200 |
| 16.3.5 | SetOokAddress .....                                       | 201 |
| 16.3.6 | GetOokRxStats .....                                       | 201 |
| 16.3.7 | GetOokPacketStatus .....                                  | 202 |
| 16.3.8 | SetOokDetector .....                                      | 203 |
| 16.3.9 | SetOokwhiteningParams .....                               | 204 |
| 17.    | LR-FHSS (LR20xx) .....                                    | 205 |
| 17.1   | LR-FHSS Modulation .....                                  | 205 |
| 17.2   | LR-FHSS Commands .....                                    | 206 |
| 17.2.1 | LrFhssBuildFrame .....                                    | 206 |
| 17.2.2 | LrFhssSetSyncword .....                                   | 207 |

|                                             |     |
|---------------------------------------------|-----|
| 17.2.3 ReadLrFhssHoppingTable .....         | 208 |
| 17.2.4 WriteLrFhssHoppingTable .....        | 208 |
| 18. FLRC Modem (LR2021) .....               | 209 |
| 18.1 FLRC Modulation .....                  | 209 |
| 18.1.1 Modem Bandwidth and Bit Rates.....   | 209 |
| 18.1.2 FEC Coding Rate .....                | 210 |
| 18.1.3 Gaussian Filtering .....             | 211 |
| 18.1.4 FLRC Frequency Tolerance .....       | 211 |
| 18.2 FLRC Packet Handler .....              | 212 |
| 18.2.1 Fixed-Length Packet Format .....     | 212 |
| 18.2.2 Variable-length Packet Format.....   | 212 |
| 18.3 FLRC Time-on-Air .....                 | 213 |
| 18.4 FLRC Radio Commands .....              | 214 |
| 18.4.1 SetFlrcModulationParams .....        | 214 |
| 18.4.2 SetFlrcPacketParams .....            | 215 |
| 18.4.3 GetFlrcRxStats .....                 | 216 |
| 18.4.4 GetFlrcPacketStatus .....            | 217 |
| 18.4.5 SetFlrcSyncword .....                | 218 |
| 19. OQPSK_15_4 (LR2021) .....               | 219 |
| 19.1 OQPSK_15_4 Modulation .....            | 219 |
| 19.2 OQPSK_15_4 Packet Handler .....        | 219 |
| 19.3 OQPSK_15_4 Radio Commands .....        | 219 |
| 19.3.1 SetOqpskParams.....                  | 219 |
| 19.3.2 GetOqpskRxStats .....                | 220 |
| 19.3.3 GetOqpskPacketStatus .....           | 221 |
| 19.3.4 SetOqpskPacketLen .....              | 221 |
| 19.3.5 SetOqpskAddress.....                 | 222 |
| 20. Z-Wave (LR2021) .....                   | 223 |
| 20.1 Z-Wave Modulation .....                | 223 |
| 20.2 Z-Wave Packet Handler .....            | 223 |
| 20.3 Z-Wave Radio Commands .....            | 224 |
| 20.3.1 SetZwaveParams .....                 | 224 |
| 20.3.2 SetZwaveHomeldFiltering .....        | 225 |
| 20.3.3 GetZwaveRxStats.....                 | 225 |
| 20.3.4 GetZwavePacketStatus .....           | 226 |
| 20.3.5 SetZwaveBeamFiltering.....           | 227 |
| 20.3.6 SetZwaveScanConfig.....              | 228 |
| 20.3.7 SetZwaveScan .....                   | 229 |
| 21. Test Commands (LR20xx) .....            | 230 |
| 21.1 Regulatory Overview .....              | 230 |
| 21.1.1 ETSI .....                           | 230 |
| 21.1.2 FCC.....                             | 230 |
| 21.2 SetTxTestMode .....                    | 231 |
| 22. Known Limitations and Workarounds ..... | 232 |

|                                                                      |     |
|----------------------------------------------------------------------|-----|
| 22.1 OOK Detection Threshold Adjustment (LR20xx)                     | 232 |
| 22.1.1 Description                                                   | 232 |
| 22.1.2 Workaround                                                    | 232 |
| 22.2 RTTOF Accuracy for SX1280-compatible Bandwidths (LR2021/LR2022) | 232 |
| 22.3 Firmware Patch RAM (PRAM)                                       | 233 |
| 22.3.1 Loading the Patch RAM                                         | 233 |
| 22.3.2 Checking the Patch RAM                                        | 233 |
| 22.4 Regulatory Compliance                                           | 234 |
| 22.4.1 CN470 SRRC Compliance for China                               | 234 |
| 22.4.2 IN865 Compliance for India                                    | 234 |
| 23. Application Information                                          | 235 |
| 23.1 Power Management                                                | 235 |
| 23.2 RF Shield Impact                                                | 235 |
| 23.3 Oscillator Recommendations                                      | 236 |
| 23.4 Reference Design                                                | 237 |
| 23.5 Unused Pins                                                     | 240 |
| 23.6 Configurations via Register Change                              | 240 |
| 23.7 OCP registers                                                   | 241 |
| 23.8 Compatibility of Generation 1–4 LoRa Radios                     | 241 |
| 24. Package Information                                              | 242 |
| 24.1 Package Outline Drawing                                         | 242 |
| 24.2 Package Marking                                                 | 243 |
| 24.3 Land Pattern                                                    | 244 |
| 24.4 Reflow Profiles                                                 | 244 |
| 24.5 Thermal Information                                             | 245 |
| 24.6 Tape & Reel Information                                         | 245 |

## List of Figures

|                                                                                     |     |
|-------------------------------------------------------------------------------------|-----|
| Figure 1-1: LR2021 Block Diagram.....                                               | 20  |
| Figure 1-2: LR2022 Block Diagram.....                                               | 21  |
| Figure 1-3: LR2012 Block Diagram.....                                               | 22  |
| Figure 1-4: Power Amplifiers.....                                                   | 24  |
| Figure 1-5: LF PA Tx Power and IDD Tx vs Targeted Power (915 MHz, 3.3 V SIMO) ..... | 25  |
| Figure 1-6: LF PA Tx Power and IDD Tx vs Targeted Power (490 MHz, 3.3 V SIMO) ..... | 26  |
| Figure 1-7: LF PA Tx Power vs VBAT for +14, +17, +22 dBm Targeted Output Power..... | 27  |
| Figure 1-8: PA IDD Tx vs VBAT for +14, +17, +22 dBm Targeted Output Power .....     | 27  |
| Figure 1-9: HF PA Tx Power and IDD Tx vs Targeted Power (2445 MHz, 3.3 V SIMO)..... | 28  |
| Figure 1-10: Switchless Dual Band Operation .....                                   | 29  |
| Figure 1-11: Modem Top View.....                                                    | 30  |
| Figure 1-12: POR and BRN Functions .....                                            | 31  |
| Figure 1-13: SIMO Configuration and Connection .....                                | 32  |
| Figure 1-14: LDO Configuration and Connection .....                                 | 32  |
| Figure 1-15: NTC Connection.....                                                    | 33  |
| Figure 1-16: TCXO Circuit Diagram .....                                             | 34  |
| Figure 2-1: LR2021/22 Pin-out in QFN32 Package .....                                | 36  |
| Figure 2-2: LR2012 Pin-out in QFN32 Package .....                                   | 37  |
| Figure 3-1: SPI Timing Diagram .....                                                | 72  |
| Figure 4-1: Chip Modes .....                                                        | 73  |
| Figure 5-1: Write Command.....                                                      | 78  |
| Figure 5-2: Write Command.....                                                      | 82  |
| Figure 5-3: Read Command.....                                                       | 83  |
| Figure 5-4: Direct Read Timing Diagram.....                                         | 83  |
| Figure 5-5: Direct Write Timing Diagram.....                                        | 83  |
| Figure 5-6: Tx Trigger Example .....                                                | 84  |
| Figure 5-7: Tx Trigger Followed by FIFO Write Example .....                         | 84  |
| Figure 8-1: Modem Control Commands.....                                             | 148 |
| Figure 9-1: LoRa Signal Bandwidth .....                                             | 151 |
| Figure 9-2: LoRa Packet Format .....                                                | 152 |
| Figure 9-3: LoRa Fast CAD Principle.....                                            | 155 |
| Figure 11-1: FSK Modulation Parameters .....                                        | 170 |
| Figure 11-2: Fixed-Length Packet Format.....                                        | 172 |
| Figure 11-3: Variable-Length Packet Format.....                                     | 172 |
| Figure 14-1: Bluetooth Low Energy Packet .....                                      | 192 |
| Figure 17-1: LR-FHSS Spectral Plot Example.....                                     | 205 |
| Figure 18-1: FLRC Fixed-length Packet Format.....                                   | 212 |
| Figure 18-2: FLRC Variable-length Packet Format.....                                | 212 |
| Figure 23-1: LR2021/LR2022 Reference Design Schematic.....                          | 237 |
| Figure 23-2: LR2012 Reference Design Schematic .....                                | 238 |
| Figure 23-3: LR2021/LR2022 Reference Design PCB Layout .....                        | 239 |
| Figure 23-4: LR2012 Reference Design PCB Layout.....                                | 239 |
| Figure 24-1: Package Outline Drawing .....                                          | 242 |
| Figure 24-2: Package Marking .....                                                  | 243 |
| Figure 24-3: Land Pattern.....                                                      | 244 |
| Figure 24-4: Tape & Reel Information.....                                           | 245 |

## List of Tables

|                                                                                       |    |
|---------------------------------------------------------------------------------------|----|
| Table 2-1: Pin-out Description.....                                                   | 35 |
| Table 3-1: Absolute Maximum Ratings (LR20xx) .....                                    | 38 |
| Table 3-2: Operating Range (LR20xx) .....                                             | 38 |
| Table 3-3: ESD and Latch-up (LR20xx) .....                                            | 39 |
| Table 3-4: Basic Modes Power Consumption (LR20xx) .....                               | 40 |
| Table 3-5: Transmit Mode Power Consumption (LR20xx unless specified otherwise) .....  | 40 |
| Table 3-6: Receive Mode Power Consumption (LR20xx).....                               | 41 |
| Table 3-7: 2.4 GHz Specific Receive Mode Power Consumption (LR2021/LR2022) .....      | 42 |
| Table 3-8: General Specifications (LR20xx unless specified otherwise) .....           | 43 |
| Table 3-9: WM-Bus Receiver Specification (LR20xx) .....                               | 45 |
| Table 3-10: Z-Wave 1% PER (LR2021).....                                               | 48 |
| Table 3-11: Wi-SUN FSK 10% PER (LR20xx) .....                                         | 49 |
| Table 3-12: FLRC Sub-GHz 1% PER (LR2021) .....                                        | 51 |
| Table 3-13: FLRC 2.4 GHz 1% PER (Packet Size = 31 bytes) (LR2021) .....               | 53 |
| Table 3-14: FSK Sensitivity @0.1% BER (LR20xx).....                                   | 55 |
| Table 3-15: Bluetooth LE Physical Layer (LR2021/LR2022) .....                         | 57 |
| Table 3-16: OQPSK_15_4 (LR2021) .....                                                 | 59 |
| Table 3-17: LoRa Sub-GHz 64B Payload (LR20xx) .....                                   | 59 |
| Table 3-18: LoRa 2.4 GHz 64B Payload (LR2021/LR2022) .....                            | 64 |
| Table 3-19: OOK 1% PER (LR20xx) .....                                                 | 65 |
| Table 3-20: General Receiver Specifications (LR20xx unless specified otherwise) ..... | 66 |
| Table 3-21: Brownout and Battery Specification (LR20xx).....                          | 67 |
| Table 3-22: Transmit Mode Specification (LR20xx unless specified otherwise).....      | 67 |
| Table 3-23: Start-up/ Mode Switching Times Specifications (LR20xx) .....              | 68 |
| Table 3-24: 32 MHz Crystal Specifications. (LR20xx) For example NDK_NX2016SA .....    | 70 |
| Table 3-25: TCXO Regulator Specifications (LR20xx).....                               | 70 |
| Table 3-26: Digital I/O Specifications (LR20xx).....                                  | 71 |
| Table 3-27: SPI Timing Requirements (LR20xx) .....                                    | 72 |
| Table 4-1: Chip Mode Conditions (LR20xx) .....                                        | 74 |
| Table 5-1: Radio FIFO Switch Driver Functions.....                                    | 80 |
| Table 5-2: Radio FIFO Base Address Registers.....                                     | 80 |
| Table 5-3: Radio FIFO Size Registers.....                                             | 81 |
| Table 5-4: Radio FIFO Memory Commands.....                                            | 85 |
| Table 5-5: Register Memory Commands.....                                              | 85 |
| Table 5-6: System Configuration Commands .....                                        | 86 |
| Table 5-7: Common Radio Commands .....                                                | 88 |
| Table 5-8: LoRa Packet Radio Commands .....                                           | 90 |
| Table 5-9: FSK Packet Radio Commands .....                                            | 91 |
| Table 5-10: FLRC Packet Radio Commands .....                                          | 92 |
| Table 5-11: BPSK Packet Radio Commands.....                                           | 92 |
| Table 5-12: LR-FHSS Packet Radio Commands.....                                        | 93 |
| Table 5-13: Bluetooth Low Energy Packet Radio Commands.....                           | 94 |
| Table 5-14: WM-Bus Packet Radio Commands .....                                        | 94 |
| Table 5-15: Wi-SUN Packet Radio Commands .....                                        | 95 |
| Table 5-16: Z-Wave Packet Radio Commands.....                                         | 96 |
| Table 5-17: OQPSK_15_4 Packet Radio Commands.....                                     | 97 |

|                                                            |     |
|------------------------------------------------------------|-----|
| Table 5-18: RTToF Packet Radio Commands .....              | 97  |
| Table 5-19: OOK Packet Radio Commands.....                 | 98  |
| Table 5-20: Host Interrupts .....                          | 99  |
| Table 6-1: ReadRadioRxFifo Command .....                   | 101 |
| Table 6-2: WriteRadioTxFifo Command .....                  | 101 |
| Table 6-3: WriteRegMem32 Command.....                      | 102 |
| Table 6-4: WriteRegMemMask32 Command .....                 | 102 |
| Table 6-5: ReadRegMem32 Command.....                       | 103 |
| Table 6-6: ReadRegMem32 Response.....                      | 103 |
| Table 6-7: SetSleep Command .....                          | 103 |
| Table 6-8: SetStandby Command .....                        | 104 |
| Table 6-9: SetFs Command.....                              | 104 |
| Table 6-10: SetAdditionalRegToRetain Command .....         | 104 |
| Table 6-11: SetRx Command .....                            | 105 |
| Table 6-12: SetTx Command .....                            | 106 |
| Table 6-13: SetRxTxFallbackMode Command .....              | 106 |
| Table 6-14: SetRxDutyCycle Command .....                   | 107 |
| Table 6-15: SetAutoRxTx Command.....                       | 108 |
| Table 6-16: StopTimeoutOnPreamble Command .....            | 108 |
| Table 6-17: SetLoraCadParams Command .....                 | 109 |
| Table 6-18: CadExitMode Parameter Definition .....         | 109 |
| Table 6-19: Lora CAD Recommended det_peak values .....     | 109 |
| Table 6-20: SetLoRaCAD Command .....                       | 110 |
| Table 6-21: SetCad Command .....                           | 111 |
| Table 6-22: SetCadParams Command.....                      | 111 |
| Table 6-23: SetCca Command .....                           | 112 |
| Table 6-24: ResetRxStats Command .....                     | 112 |
| Table 6-25: SetDefaultRxTxTimeout Command .....            | 112 |
| Table 6-26: SetRegMode Command.....                        | 113 |
| Table 6-27: Calibrate Command.....                         | 114 |
| Table 6-28: Blocks_to_calibrate Parameter Definition ..... | 114 |
| Table 6-29: CalibFE Command.....                           | 115 |
| Table 6-30: GetVBat Request Command.....                   | 116 |
| Table 6-31: GetVBat Response.....                          | 116 |
| Table 6-32: GetTemp Command .....                          | 117 |
| Table 6-33: GetTemp Response .....                         | 117 |
| Table 6-34: SetLbdCfg Command .....                        | 118 |
| Table 6-35: GetRandomNumber Request Command.....           | 119 |
| Table 6-36: GetRandomNumber Response.....                  | 119 |
| Table 6-37: GetStatus Command .....                        | 120 |
| Table 6-38: Status (Stat) Definition .....                 | 120 |
| Table 6-39: GetVersion Command .....                       | 121 |
| Table 6-40: GetVersion Response .....                      | 121 |
| Table 6-41: ClearError Command.....                        | 121 |
| Table 6-42: GetErrors Command.....                         | 122 |
| Table 6-43: GetErrors Response.....                        | 122 |
| Table 6-44: SetDioFunction Command.....                    | 123 |
| Table 6-45: SetDioRfSwitchConfig Command .....             | 124 |
| Table 6-46: SetDioIrqConfig Command .....                  | 124 |
| Table 6-47: ClearIrq Command .....                         | 125 |
| Table 6-48: GetAndClearIrqStatus Command .....             | 125 |

|                                                                                             |     |
|---------------------------------------------------------------------------------------------|-----|
| Table 6-49: GetAndClearIrqStatus Response .....                                             | 125 |
| Table 6-50: ConfigFifoIrq Command .....                                                     | 126 |
| Table 6-51: GetFifoIrqFlags Request Command .....                                           | 127 |
| Table 6-52: GetFifoIrqFlags Read Command .....                                              | 127 |
| Table 6-53: ClearFifoIrqFlags Command .....                                                 | 127 |
| Table 6-54: GetAndClearFifoIrqFlags Command .....                                           | 128 |
| Table 6-55: GetAndClearFifoIrqFlags Response .....                                          | 128 |
| Table 6-56: GetRxFifoLevel Command .....                                                    | 128 |
| Table 6-57: GetRxFifoLevel Response .....                                                   | 128 |
| Table 6-58: GetTxFifoLevel Command .....                                                    | 129 |
| Table 6-59: GetTxFifoLevel Response.....                                                    | 129 |
| Table 6-60: ClearRxFifo Command .....                                                       | 129 |
| Table 6-61: ClearTxFifo Command .....                                                       | 129 |
| Table 6-62: ConfigLfClock Command .....                                                     | 130 |
| Table 6-63: ConfigClkOutputs Command .....                                                  | 130 |
| Table 6-64: SetTcxoMode Command .....                                                       | 131 |
| Table 6-65: TCXO Supply Voltage Programming Values .....                                    | 131 |
| Table 6-66: SetXoscCpTrim Command .....                                                     | 132 |
| Table 6-67: XTA Capacitor Value .....                                                       | 132 |
| Table 6-68: XTB Capacitor Value .....                                                       | 132 |
| Table 6-69: SetTempCompCfg Command .....                                                    | 133 |
| Table 6-70: SetNtcParams Command.....                                                       | 133 |
| Table 7-1: SetRfFrequency Command.....                                                      | 134 |
| Table 7-2: SetRxPath Command .....                                                          | 135 |
| Table 7-3: GetRssiInst Command .....                                                        | 135 |
| Table 7-4: GetRssiInst Response .....                                                       | 135 |
| Table 7-5: SetRssiCalibration Command .....                                                 | 136 |
| Table 7-6: Recommended Values for Reference EVK HF path 2.4 GHz (loaded by default) .       | 137 |
| Table 7-7: Recommended Values for Reference EVK LF path 868–915 MHz (loaded by default)     | 137 |
| Table 7-8: Recommended Values for Reference EVK LF path 490 MHz (loaded by the application) | 137 |
| Table 7-9: ResetRxStats Command .....                                                       | 139 |
| Table 7-10: SetTimestampSource Command .....                                                | 139 |
| Table 7-11: GetTimestampValue Command .....                                                 | 140 |
| Table 7-12: GetTimestampValue Response .....                                                | 140 |
| Table 7-13: GetCcaResult Command.....                                                       | 140 |
| Table 7-14: GetCcaResult Response.....                                                      | 140 |
| Table 7-15: SetAgcGainManual Command.....                                                   | 141 |
| Table 7-16: GetRxPktLength Command .....                                                    | 141 |
| Table 7-17: GetRxPktLength Response.....                                                    | 141 |
| Table 7-18: SetPaConfig Command .....                                                       | 142 |
| Table 7-19: Optimal Values for 915 MHz Semtech Reference Design .....                       | 143 |
| Table 7-20: Optimal Values for 490 MHz Semtech Reference Design .....                       | 144 |
| Table 7-21: Optimal Values for 2445 MHz Semtech Reference Design (LR2021/LR2022)....        | 145 |
| Table 7-22: SelPa Command.....                                                              | 146 |
| Table 7-23: SetTxParams Command .....                                                       | 147 |
| Table 7-24: Tx Ramp_time Parameter Definition.....                                          | 147 |
| Table 8-1: SetPacketType Command.....                                                       | 149 |
| Table 8-2: GetPacketType Command .....                                                      | 150 |
| Table 8-3: GetPacketType Response .....                                                     | 150 |
| Table 9-1: SetLoraModulationParams Command .....                                            | 157 |
| Table 9-2: LoRa Modulation sf Parameter Definition.....                                     | 157 |



|                                                                              |     |
|------------------------------------------------------------------------------|-----|
| Table 9-3: LoRa Modulation bw Parameter Definition.....                      | 157 |
| Table 9-4: LoRa Modulation cr Parameter Definition .....                     | 158 |
| Table 9-5: SetLoRaPacketParams Command.....                                  | 159 |
| Table 9-6: SetLoraSynchTimeout Command .....                                 | 160 |
| Table 9-7: SetLoraSyncword Command .....                                     | 160 |
| Table 9-8: SetLoraSideDetConfig Command .....                                | 161 |
| Table 9-9: SetLoraSideDetSyncword Command.....                               | 161 |
| Table 9-10: GetLoraRxStats Command.....                                      | 162 |
| Table 9-11: GetLoraRxStats Response.....                                     | 162 |
| Table 9-12: GetLoraPacketStatus Command.....                                 | 163 |
| Table 9-13: GetLoraPacketStatus Response.....                                | 163 |
| Table 9-14: SetLoraAddress Command .....                                     | 164 |
| Table 9-15: SetLoRaHopping Command.....                                      | 164 |
| Table 9-16: SetLoraTxSync Command .....                                      | 165 |
| Table 9-17: SetLoraSideDetCad Command .....                                  | 165 |
| Table 10-1: SetRangingAddr Command .....                                     | 166 |
| Table 10-2: SetRangingReqAddr Command .....                                  | 166 |
| Table 10-3: GetRangingResult Command.....                                    | 167 |
| Table 10-4: GetRangingResult Response.....                                   | 167 |
| Table 10-5: GetRangingStats Request Command.....                             | 168 |
| Table 10-6: GetRangingStats Response.....                                    | 168 |
| Table 10-7: SetRangingTxRxDelay Command.....                                 | 169 |
| Table 10-8: SetRangingParams Command .....                                   | 169 |
| Table 11-1: Oversampling Rate Limitations .....                              | 171 |
| Table 11-2: (G)FSK Limitations Summary .....                                 | 171 |
| Table 11-3: SetFskModulationParams Command .....                             | 173 |
| Table 11-4: Correlation of Bandwidth, rx_bw and Intermediate Frequency ..... | 174 |
| Table 11-5: SetFskPacketParams Command .....                                 | 175 |
| Table 11-6: (G)FSK Packet pbl_detect Parameter Definition .....              | 175 |
| Table 11-7: (G)FSK Packet AddrComp Parameter Definition .....                | 176 |
| Table 11-8: (G)FSK Packet pkt_format Parameter Definition .....              | 176 |
| Table 11-9: (G)FSK packet CRC Parameter Definition .....                     | 177 |
| Table 11-10: (G)FSK packet dc-free Parameter Definition .....                | 177 |
| Table 11-11: SetFskWhiteningParams Command .....                             | 178 |
| Table 11-12: SetFskCrcParams Command.....                                    | 178 |
| Table 11-13: SetFskSyncword Command .....                                    | 179 |
| Table 11-14: SetFskAddress Command .....                                     | 179 |
| Table 11-15: GetFskRxStats Command Request .....                             | 180 |
| Table 11-16: GetFskRxStats Response.....                                     | 180 |
| Table 11-17: GetFskPacketStatus Request Command .....                        | 181 |
| Table 11-18: GetFskPacketStatus Response.....                                | 181 |
| Table 12-1: Supported WM-BUS Modulation .....                                | 182 |
| Table 12-2: SetWmbusParams Command.....                                      | 183 |
| Table 12-3: GetWmbusRxStats Request.....                                     | 184 |
| Table 12-4: GetWmbusRxStats Response .....                                   | 184 |
| Table 12-5: GetWmbusPacketStatus Request.....                                | 185 |
| Table 12-6: GetWmbusPacketStatus Response .....                              | 185 |
| Table 12-7: SetWmbusFilteringAddress Command.....                            | 186 |
| Table 13-1: Supported Wi-SUN Modulation.....                                 | 187 |
| Table 13-2: SetWisunMode Command .....                                       | 188 |
| Table 13-3: SetWisunPacketParams Command .....                               | 189 |

|                                                                  |     |
|------------------------------------------------------------------|-----|
| Table 13-4: GetWisunRxStats Request Command .....                | 190 |
| Table 13-5: GetWisunRxStats Response.....                        | 190 |
| Table 13-6: GetWisunPacketStatus Request Command .....           | 191 |
| Table 13-7: GetWisunPacketStatus Response.....                   | 191 |
| Table 13-8: SetWisunPacketLen Command .....                      | 191 |
| Table 14-1: SetBleModulationParams Command .....                 | 192 |
| Table 14-2: SetBleChannelParams Command .....                    | 193 |
| Table 14-3: SetBleTxPduLen Command .....                         | 193 |
| Table 14-4: SetBleTx Command .....                               | 194 |
| Table 14-5: GetBleRxStats Command.....                           | 194 |
| Table 14-6: GetBleRxStats Response.....                          | 194 |
| Table 14-7: GetBlePacketStatus Command.....                      | 195 |
| Table 14-8: GetBlePacketStatus Response.....                     | 195 |
| Table 15-1: SetBpskModulationParams Command.....                 | 196 |
| Table 15-2: SetBpskPacketParams Command .....                    | 197 |
| Table 16-1: SetOokModulationParams Command.....                  | 198 |
| Table 16-2: SetOokPacketParams Command .....                     | 199 |
| Table 16-3: SetOokCrcParams Command .....                        | 200 |
| Table 16-4: SetOokSyncword Command.....                          | 200 |
| Table 16-5: SetOokAddress Command.....                           | 201 |
| Table 16-6: GetOokRxStats Command .....                          | 201 |
| Table 16-7: GetOokRxStats Response .....                         | 201 |
| Table 16-8: GetOokPacketStatus Command .....                     | 202 |
| Table 16-9: GetOokPacketStatus Response .....                    | 202 |
| Table 16-10: SetOokDetector Command .....                        | 203 |
| Table 16-11: SetOokwhiteningParams Command.....                  | 204 |
| Table 17-1: LrFhssBuildFrame Command.....                        | 206 |
| Table 17-2: LrFhssSetSyncword Command .....                      | 207 |
| Table 17-3: ReadLrFhssHoppingTable Command.....                  | 208 |
| Table 17-4: ReadLrFhssHoppingTable Command.....                  | 208 |
| Table 18-1: Valid FLRC Bit Rate and Bandwidth Combinations ..... | 209 |
| Table 18-2: Effective FLRC Bit Rates Based on FEC Usage .....    | 210 |
| Table 18-3: Receiver Performance of the FLRC Modem .....         | 211 |
| Table 18-4: SetFlrcModulationParams Command .....                | 214 |
| Table 18-5: SetFlrcPacketParams Command.....                     | 215 |
| Table 18-6: GetFlrcRxStats Request Command .....                 | 216 |
| Table 18-7: GetFlrcRxStats Response .....                        | 216 |
| Table 18-8: GetFlrcPacketStatus Command .....                    | 217 |
| Table 18-9: GetFlrcPacketStatus Response .....                   | 217 |
| Table 18-10: SetFlrcSyncword Command.....                        | 218 |
| Table 19-1: SetOqpskParams Command .....                         | 219 |
| Table 19-2: GetOqpskRxStats Command .....                        | 220 |
| Table 19-3: GetOqpskRxStats Response.....                        | 220 |
| Table 19-4: GetOqpskPacketStatus Command.....                    | 221 |
| Table 19-5: GetOqpskPacketStatus Response.....                   | 221 |
| Table 19-6: SetOqpskPacketLen Command .....                      | 221 |
| Table 19-7: SetOqpskAddress Command .....                        | 222 |
| Table 20-1: Supported Z-Wave Modulation .....                    | 223 |
| Table 20-2: SetZwaveParams Command.....                          | 224 |
| Table 20-3: SetZwaveHomelidFiltering Command.....                | 225 |
| Table 20-4: GetZwaveRxStats Request Command .....                | 225 |

|                                                                |     |
|----------------------------------------------------------------|-----|
| Table 20-5: GetZwaveRxStats Response .....                     | 225 |
| Table 20-6: GetZwavePacketStatus Request Command .....         | 226 |
| Table 20-7: GetZwavePacketStatus Response .....                | 226 |
| Table 20-8: SetZwaveBeamFiltering Command .....                | 227 |
| Table 20-9: SetZwaveScanConfig Command .....                   | 228 |
| Table 20-10: SetZwaveScan Command .....                        | 229 |
| Table 21-1: ETSI Test Signals .....                            | 230 |
| Table 21-2: SetTxTestMode Command .....                        | 231 |
| Table 23-1: Sensitivity for SIMO with Metallic Shielding ..... | 235 |
| Table 23-2: Register Usage.....                                | 240 |
| Table 24-1: Package Thermal Information .....                  | 245 |

# 1. Chip Description

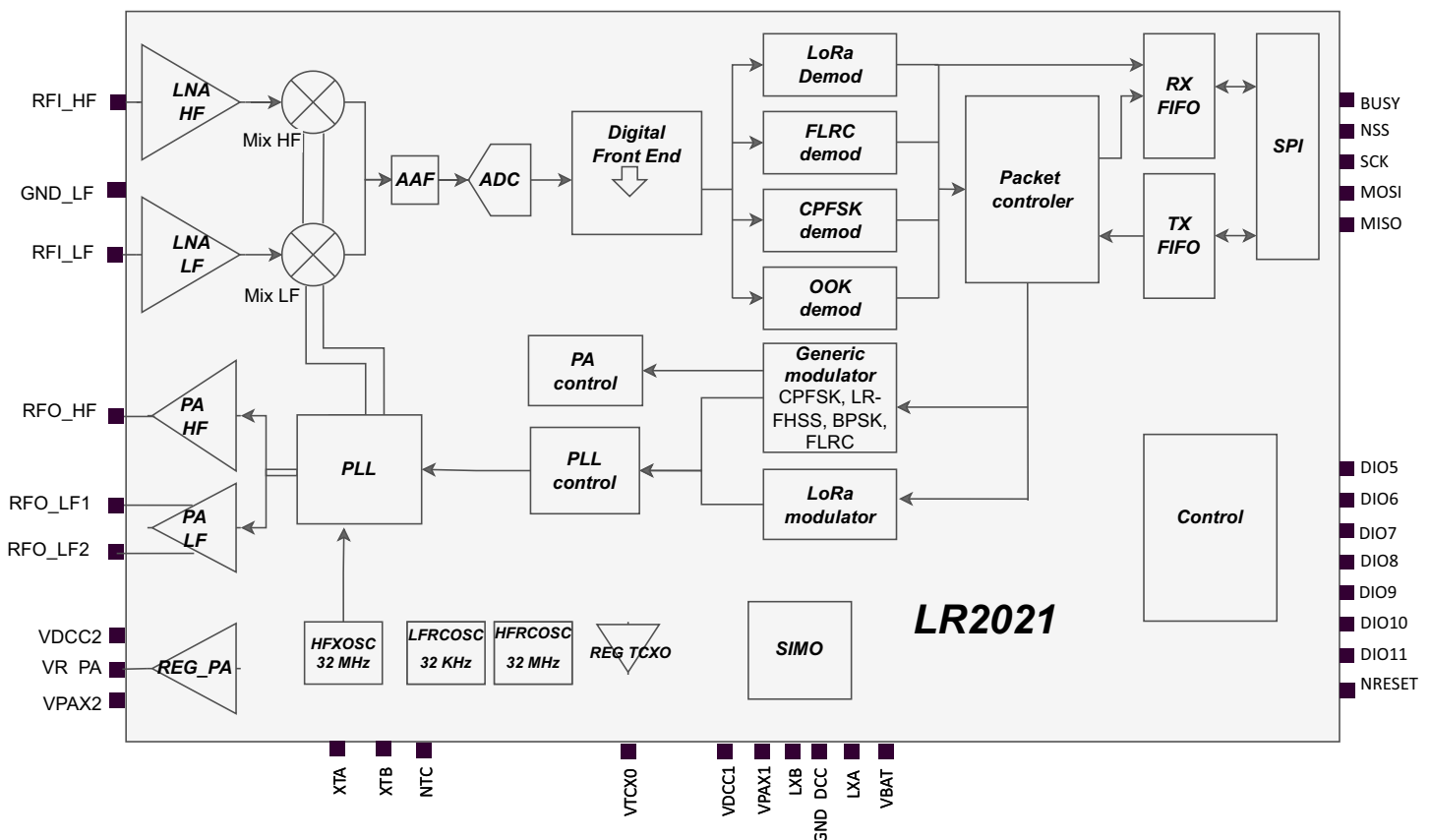
## 1.1 LR2021

The LR2021 is a half-duplex radio frequency (RF) transceiver, that handles various constant envelope modulation schemes, including LoRa, (G)FSK, (G)MSK, FLRC, 4-FSK, O-QPSK and LR-FHSS (transmit only). Additionally, the chip supports OOK and low bit rate DBPSK (transmit only), providing modulation options to suit diverse wireless communication needs.

Designed to be fully compatible with the SX126x, SX127x, SX128x and LR11xx families, the LR2021 offers direct integration and interoperability within existing systems using these families of ICs.

- Covering a broad frequency range from 150 MHz to 2500 MHz, the LR2021 enables communication across a wide spectrum of frequencies. Specifically, it supports signals in the sub-GHz ISM bands, with continuous coverage from 150 MHz to 960 MHz facilitated by a dedicated Rx (Receiver) and Tx (Transmitter) low-frequency path.
- The LR2021 also supports operation in the S and L bands and the 2.4 GHz ISM band using a dedicated Rx (Receiver) and Tx (Transmitter) high-frequency path. This allows the chip to handle communications in the 2.4 GHz range, a frequency commonly used for various wireless applications.

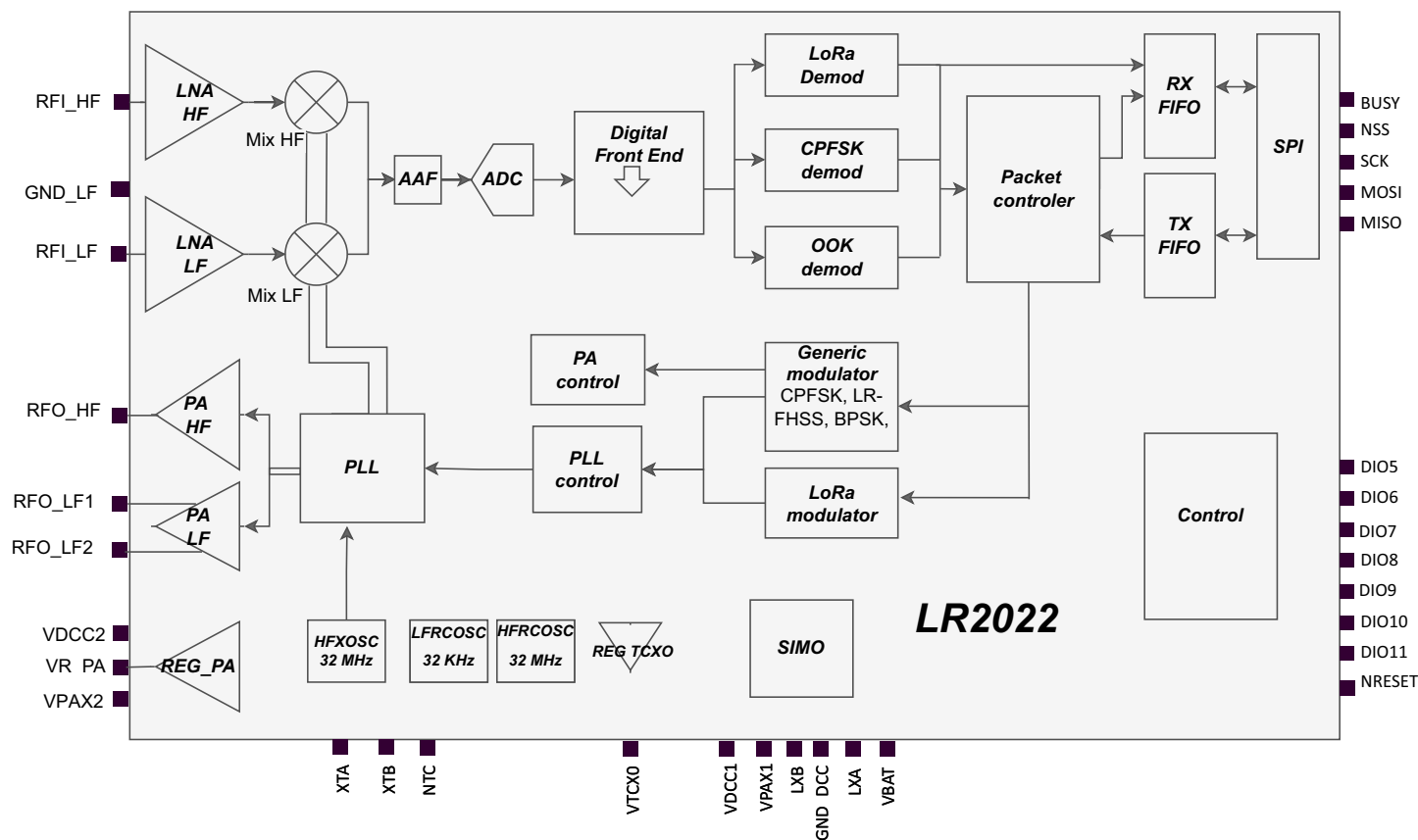
**Figure 1-1: LR2021 Block Diagram**



## 1.2 LR2022

The LR2022 provides all the features of the LR2021 except it does not support FLRC, O-QPSK, or Z-Wave.

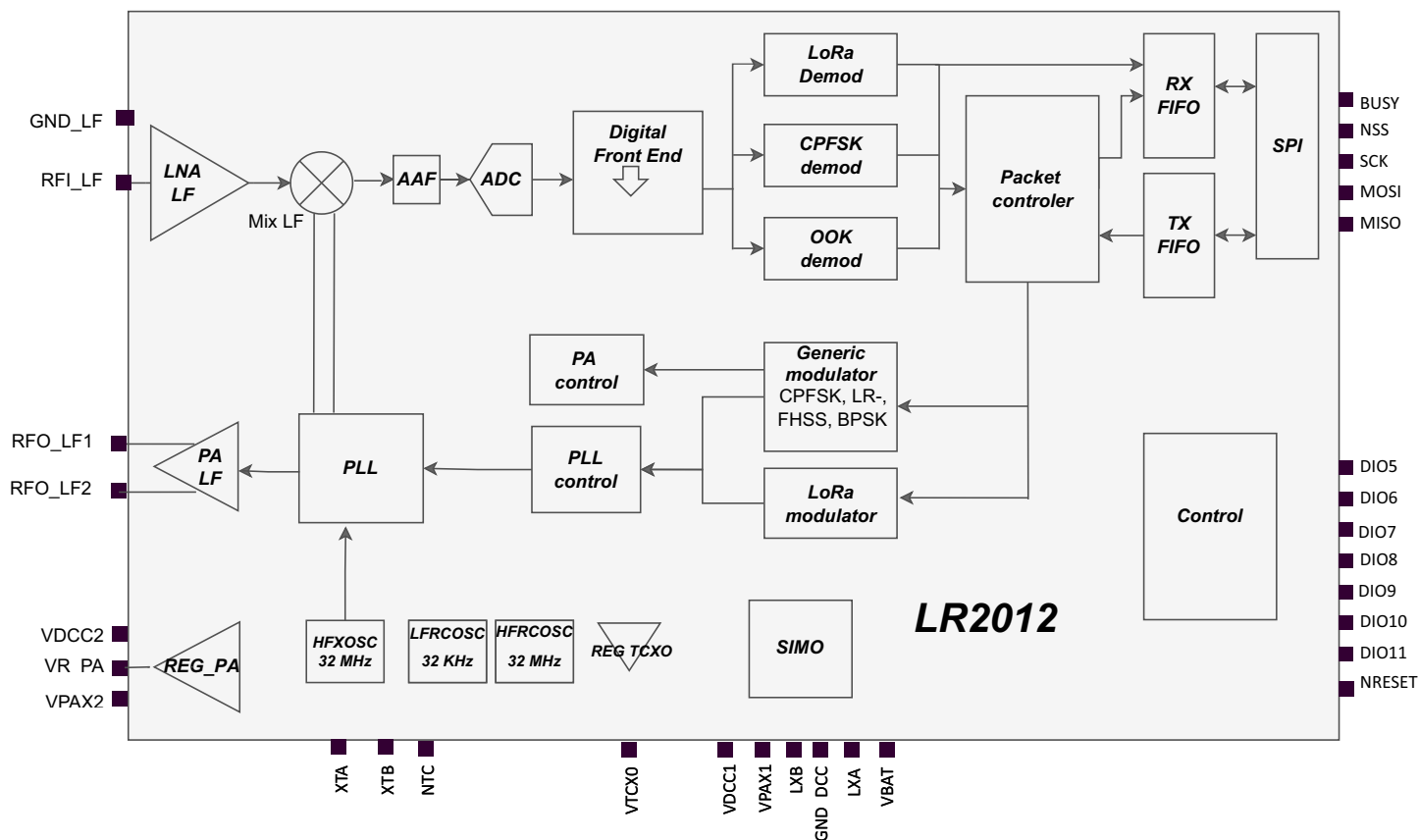
**Figure 1-2: LR2022 Block Diagram**



## 1.3 LR2012

The LR2012 only supports sub-GHz features, and does not support FLRC, O-QPSK, Z-Wave, or Bluetooth Low Energy.

Figure 1-3: LR2012 Block Diagram



## 1.4 Receiver RF Path

The LR2021/LR2022 features two distinct receiver inputs, RFI\_HF/LF, specifically designed for receiving signals in the sub-GHz ISM bands and 2.4 GHz bands. The LR2012 features only the RFI\_LF receiver supporting the sub-GHz ISM band. To achieve the desired receiver gain, a discrete external matching network partially contributes to the amplification process. This external matching network ensures that the receiver is appropriately tuned for optimal signal reception.

In addition to the sub-GHz ISM bands, the LR2021/LR2022 supports reception in the 1.5–2.5 GHz frequency range through the RFI\_HF input. This input is particularly suitable for signals in S and L Bands, used in satellite communication, as well as Bluetooth Low Energy compatible applications. Similar to the RFI\_LF input, the RFI\_HF input also benefits from an external matching network to achieve the desired impedance adaptation.

Optimal performance is achieved without a switch. A switch can be used when an external PA, LNA or FEM is employed.

After amplification in the receive chain, the RF signals are down-converted to an intermediate frequency, preparing them for further processing.

The down-converted digital signal is digitalized and then passes through a digital front end responsible for digital decimation filtering and control of the Automatic Gain Control (AGC), ensuring signal conditioning and optimization.

After conditioning, bits are extracted by the selected demodulator.

The demodulated data is further processed by the packet handler, which parses, when applicable, header information, checks CRC (Cyclic Redundancy Check), and applies optional address filtering. Once processed, the data is stored in a receiver FIFO (First-In-First-Out) buffer, accessible to the host through SPI commands, facilitating smooth and efficient data transfer between the LR2021/LR2022/LR2012 and the host system.

Automatic Gain Control (AGC) is a feedback mechanism that automatically adjusts the receiver's gain to ensure proper reception of a wanted signal despite the limited dynamic of the Front-End (FE). AGC aims to meet the following objectives ordered by priority:

1. Avoid saturation of the FE.
2. Maintain sufficient Signal-to-Noise Power Ratio (SNR) within the reception channel.
3. Minimize the number of step changes.

The receiver gain comprises 11 standard LNA gain settings (G1–G11) and eight boosted configurations for each of the two highest gain settings G12 and G13 (G12/G13\_boost0 to G12/G13\_boost7), covering the full dynamic range of the front-end. The AGC uses these steps to select the optimal operating point; the [SetRssiCalibration](#) command stores the corresponding per-step gain and noise figure correction values, ensuring accurate RSSI reporting across all signal levels.

AGC is enabled by default. If necessary, it can be disabled using [SetAgcGainManual\(GainStep\)](#) and re-enabled using [SetAgcGainManual\(0\)](#). Even if gain control is disabled, power estimation continues to be computed and can be read using [GetRssiInst](#) command.

The gain tables in the LR2021/LR2022/LR2012 are currently optimized for the reference design in [Section 23. "Application Information" on page 235](#) and don't need to be changed for designs mimicking the Semtech reference design. For designs employing a low noise amplifier (LNA), contact your Semtech representative.

## 1.5 Transmitter RF Path

During a transmission, the data intended for transmission is stored in the transmitter First-In-First-Out (FIFO) buffer through SPI communication. Once the transmission commences, the packet handler efficiently packs the data before it undergoes modulation using one of the available modulators: LoRa, (G)(M)FSK, OOK, 4-FSK, FLRC or LR-FHSS.

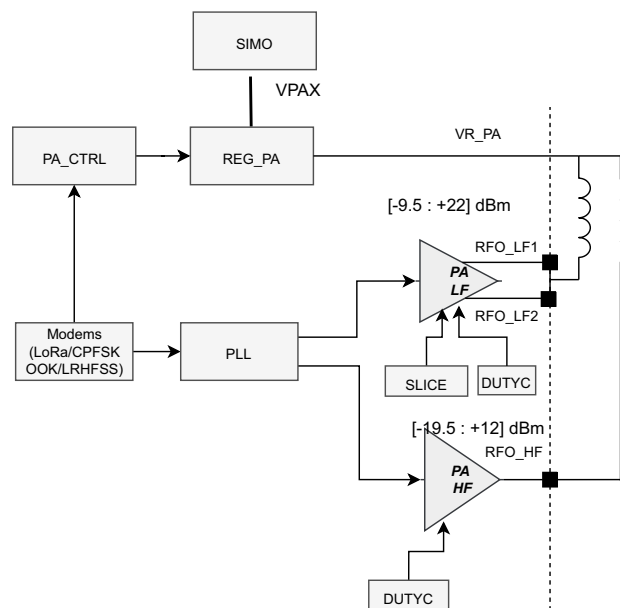
After modulation, the signal is further amplified using one of the two main Tx channels available in the circuit:

- ♦ High-Frequency Transmitter (PA\_HF): This channel is dedicated to high-frequency transmission needs up to +12 dBm, such as in the 1.5–2.5 GHz frequency range used for satellite communication and other licensed bands. The amplified signal is available on pin RFO\_HF. (Available on the LR2021/22).
- ♦ Low-Frequency Transmitter (PA\_LF): This channel is specifically designed for low-frequency transmission up to +22 dBm, covering sub-GHz ISM frequency bands and various other applications. The amplified signal is available on pins RFO\_LF1 and RFO\_LF2. These two pins must be connected together at PCB level to provide a singled ended configuration.

To facilitate proper biasing and control of the output power for each amplifier, a choke inductor is externally connected between the respective output of the amplifiers and VR\_PA (Voltage Regulator\_PA). Additionally, off-chip impedance matching and harmonic filtering are carried out using external components to optimize the performance of the amplifiers.

An internal regulator provides the output power of the power amplifiers and guarantees a smooth ramping to avoid spectral splatter. It achieves this by adjusting the voltage on VR\_PA according to the required output power, enabling efficient power management.

**Figure 1-4: Power Amplifiers**



For both the high and low-frequency paths, there are two possible methods to connect the Tx and Rx paths to the antenna:

- ♦ RF Switch Method: An RF switch can be used to connect both Tx and Rx paths to the antenna. While this ensures improved performance by maintaining separate paths for Tx and Rx, it involves the inclusion of an additional RF switch component.
- ♦ Direct-Tie Method: This method directly connects the paths to the antenna without using an RF switch, thus eliminating the cost of an additional component.

Semtech proposes an optimized direct-tie implementation for both the LF and HF paths, more details are given in [1.5.3 Direct-Tie Implementation \(LR20xx\)](#).

The developer should configure the PA corresponding to the output power and current consumption requirements of the application. No automatic frequency limitation is performed during the PA selection: the frequency selection has to be performed according to the capability offered by the external matching network component values.



### 1.5.1 Low Frequency Power Amplifier (LF PA) (LR20xx)

LF PA is designed for sub-GHz operation (150–960 MHz), optimized to deliver between -9.5 dBm and +22 dBm in 0.5 dB steps.

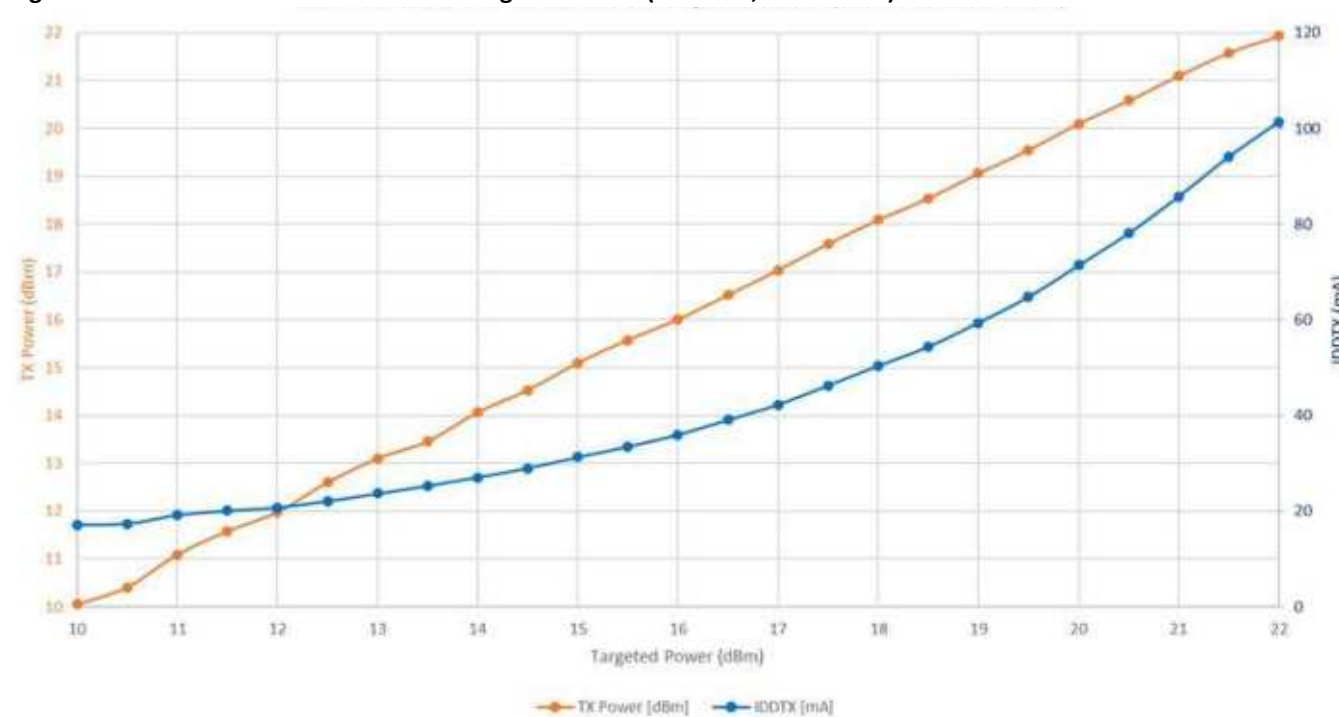
The PA output power is controlled by three main parameters, configurable through SPI commands:

- **tx\_power**: Programmed power level ([SetTxParams](#))
- **pa\_lf\_duty\_cycle**: PA conduction angle and bias setting ([SetPaConfig](#))
- **pa\_lf\_slices**: Size control for LF PA cell ([SetPaConfig](#))

The various output power steps can thus be obtained through combinations of these 3 parameters, allowing to optimize the output power, current consumption or PA harmonics level, depending on the application PCB implementation. Therefore a defined set of optimized PA parameters can be defined to reach the targeted Tx power for each application PCB. Refer to [Table 7-19](#) & [Table 7-20](#) for optimal values for 490 MHz and 915 MHz respectively.

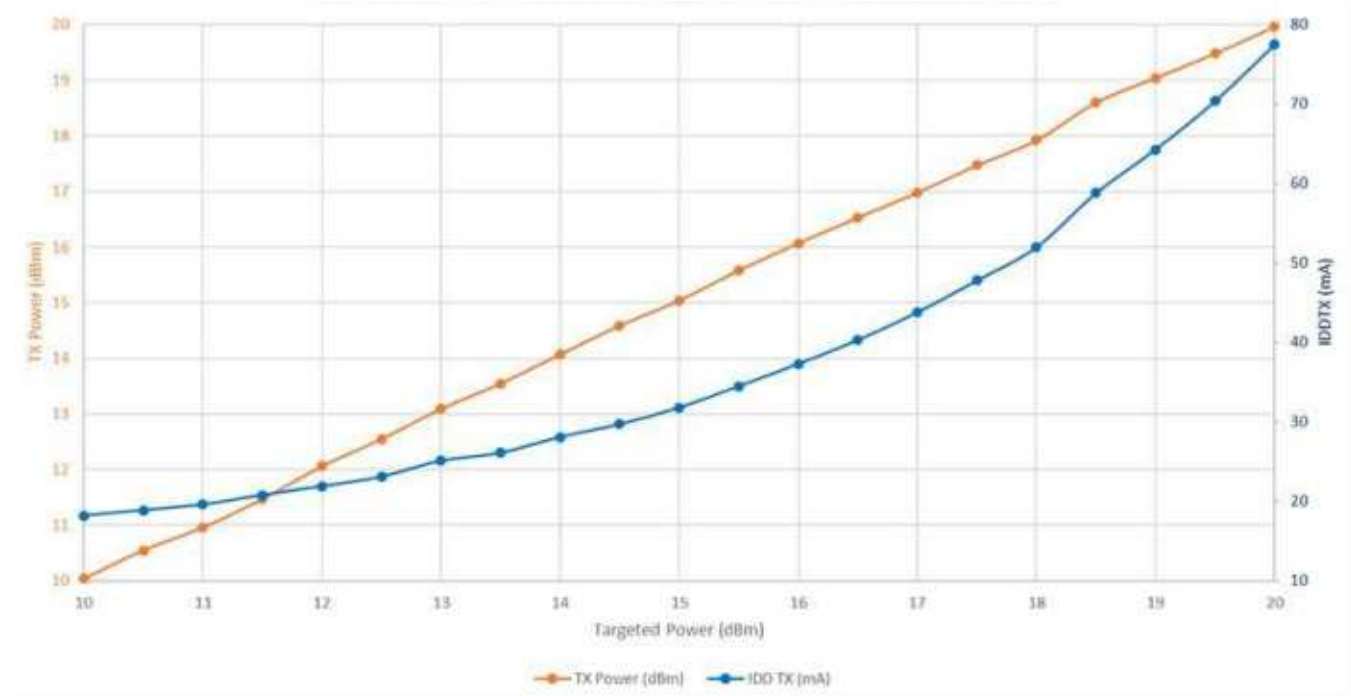
[Figure 1-5](#) and [Figure 1-6](#) below show the relationship between the LF PA output power and current consumption versus the targeted power at 3.3 V in SIMO configuration with these optimized settings for both Semtech 915 MHz reference design, and 490 MHz reference design respectively.

**Figure 1-5: LF PA Tx Power and IDD Tx vs Targeted Power (915 MHz, 3.3 V SIMO)**



The Tx Power (orange curve) demonstrates good linearity from 10 dBm to 22 dBm target power, with the actual output power closely tracking the targeted power. Current consumption (blue curve) shows a non-linear relationship range, as the current increases progressively at higher power levels, from approximately 18 mA at +10 dBm to 120 mA at +22 dBm.

**Figure 1-6: LF PA Tx Power and IDD Tx vs Targeted Power (490 MHz, 3.3 V SIMO)**



The LR2021/LR2022/LR2012 uses an internal SIMO converter and voltage regulation system to supply VR\_PA.

Figure 1-7 shows that the +14 dBm and +17 dBm power targets remain stable across the full supply voltage range (1.8 V-3.7 V), while the +22 dBm target is limited by the supply voltage: output is clamped at +20 dBm for VBAT < 2.2 V and achieves the full +22 dBm for VBAT ≥ 2.2 V.

**Figure 1-7: LF PA Tx Power vs VBAT for +14, +17, +22 dBm Targeted Output Power**

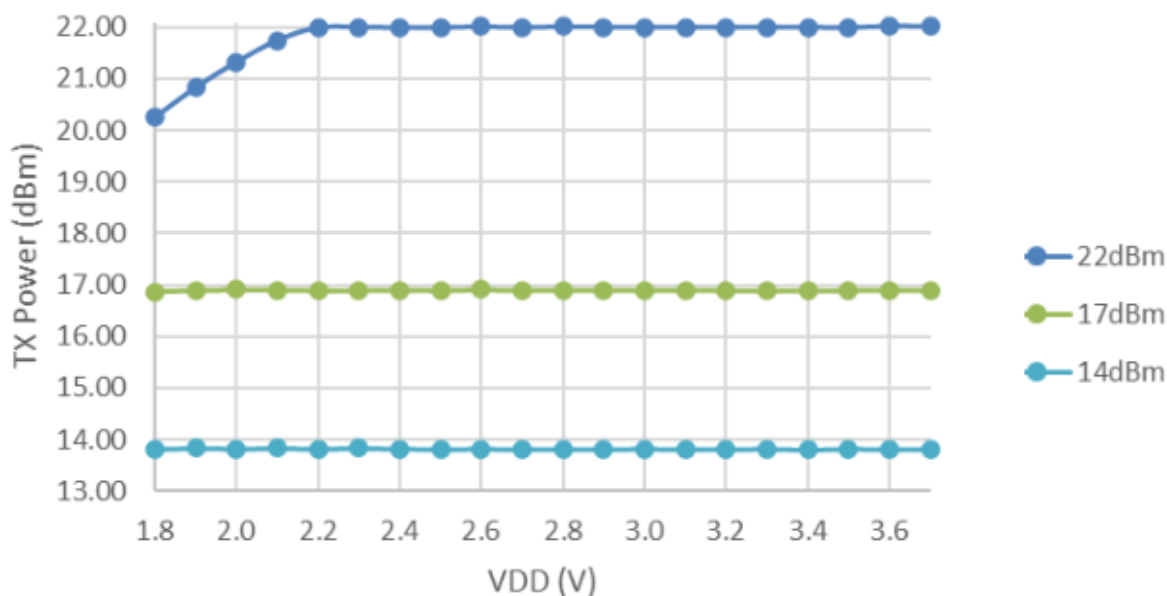
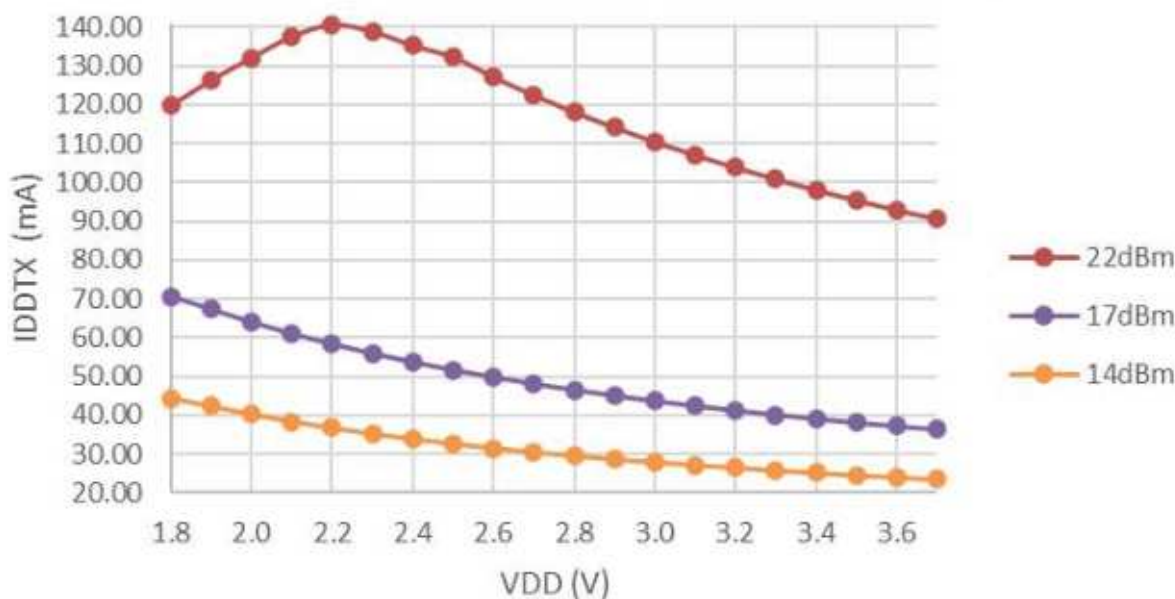


Figure 1-8 shows LF PA current consumption decreasing for +14 dBm and +17 dBm, with increasing VBAT due to improved regulator efficiency at higher supply voltages. The +22 dBm setting exhibits a peak at approximately 2.2 V (~140 mA) where the regulator operates with reduced efficiency while output is clamped at +20 dBm, then decreases to ~95 mA at higher voltages where improved regulator efficiency enables full +22 dBm delivery.

**Figure 1-8: PA IDD Tx vs VBAT for +14, +17, +22 dBm Targeted Output Power**



## 1.5.2 High Frequency Power Amplifier (HF PA) (LR2021/22)

HF PA is optimized for 2.4 GHz operation (1500–2500 MHz), capable of delivering up to 12 dBm in 0.5 dB steps.

HF PA output power is controlled by only two parameters, configurable through SPI commands:

- **tx\_power**: Programmed power level ([SetTxParams](#))
- **pa\_lf\_duty\_cycle**: PA conduction angle and bias setting ([SetPaConfig](#))

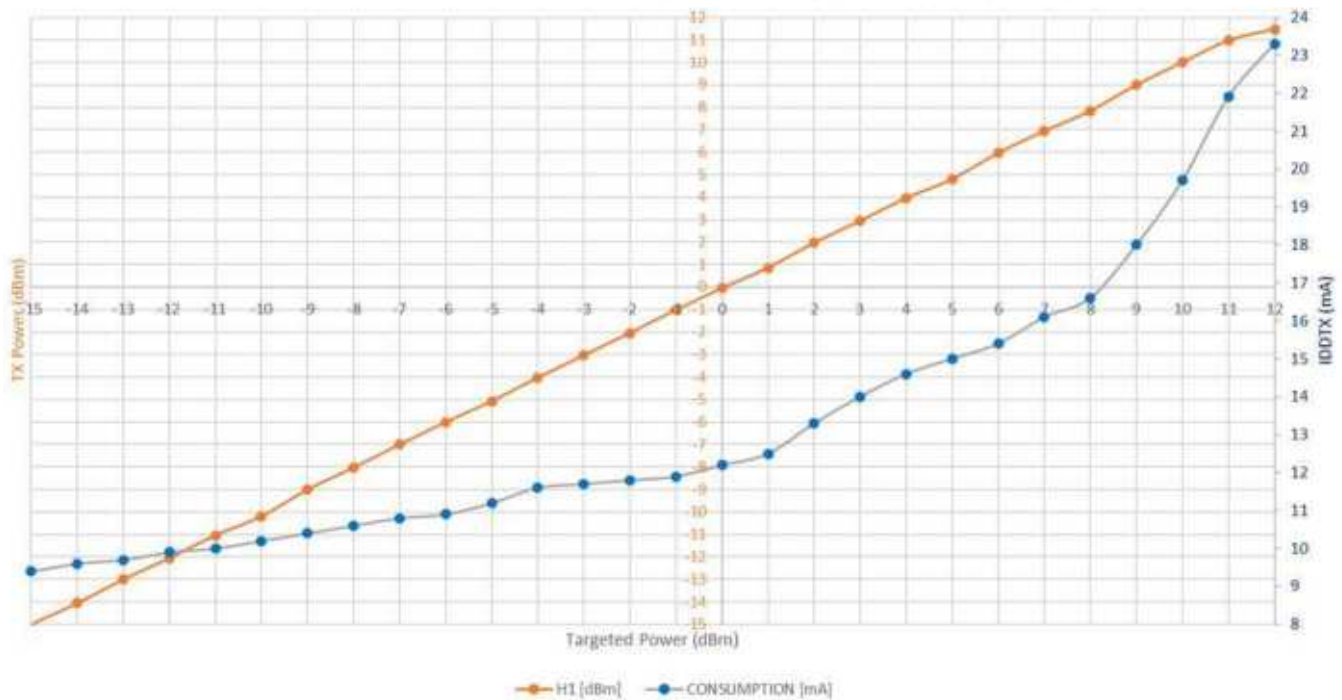
As for LF PA, the various output power steps can be obtained through combination of these parameters. Therefore a defined set of optimized PA parameters can be defined to reach the targeted Tx Power for each application PCB. Refer to [Table 7-21 for Optimal Values for 2445 MHz Semtech Reference Design \(LR2021/LR2022\)](#).

[Figure 1-9](#) shows that HF PA demonstrates linear power transfer, from -15 dBm to +12 dBm power setting, with actual output tracking programmed target. Current consumption ranges from 10 mA at -15 dBm to 24 mA at +12 dBm power setting, showing near-linear scaling.

The 24 mA consumption at maximum power (+12 dBm power setting) indicates good PA efficiency.

To obtain higher efficiency at 10 dBm or less, contact Semtech for assistance.

**Figure 1-9: HF PA Tx Power and IDD Tx vs Targeted Power (2445 MHz, 3.3 V SIMO)**

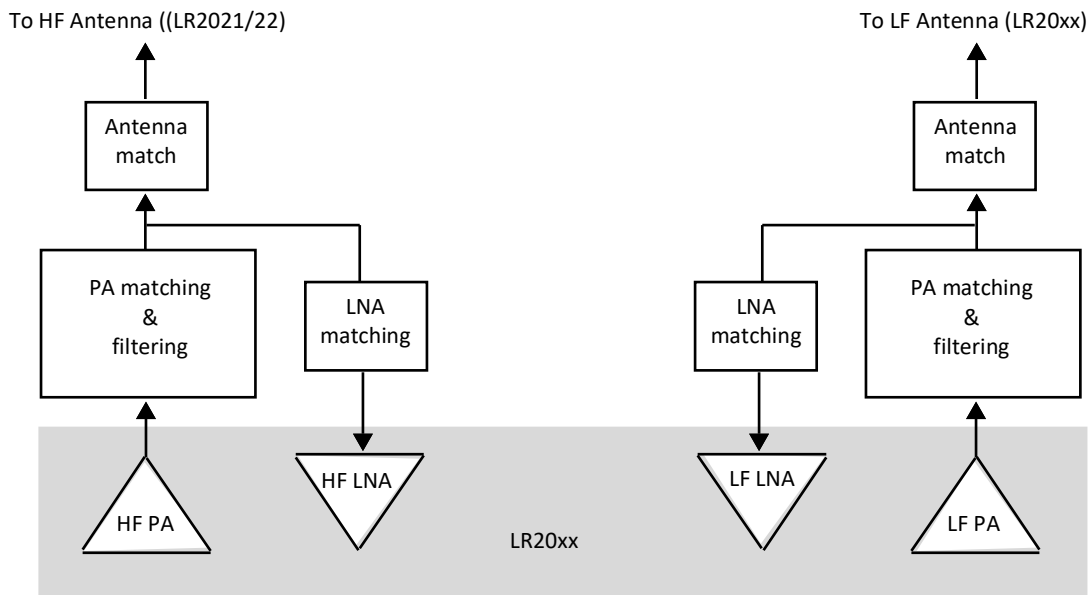


### 1.5.3 Direct-Tie Implementation (LR20xx)

Low cost applications can be implemented using a simple “direct-tie” matching configuration for both HF (LR2021/22) and LF (LR20xx) paths, eliminating the need for an RF switch.

In the direct-tie setup, the Power Amplifier (PA) and the Receiver (Rx) stages are directly connected. However, to safeguard the chip from potential damage caused by current flow in the Rx stage, series capacitors are essential between the PA and Rx stages. These capacitances act as protective elements, preventing any adverse effects on the chip during operation.

**Figure 1-10: Switchless Dual Band Operation**



This matching network shows a situation where both HF and LF paths are used and can be simplified if only one frequency band is used, or if a maximum output power on LF path is reduced. In the latter case, reducing harmonic filtering capability of the proposed matching is possible.

## 1.6 Frequency Synthesizer

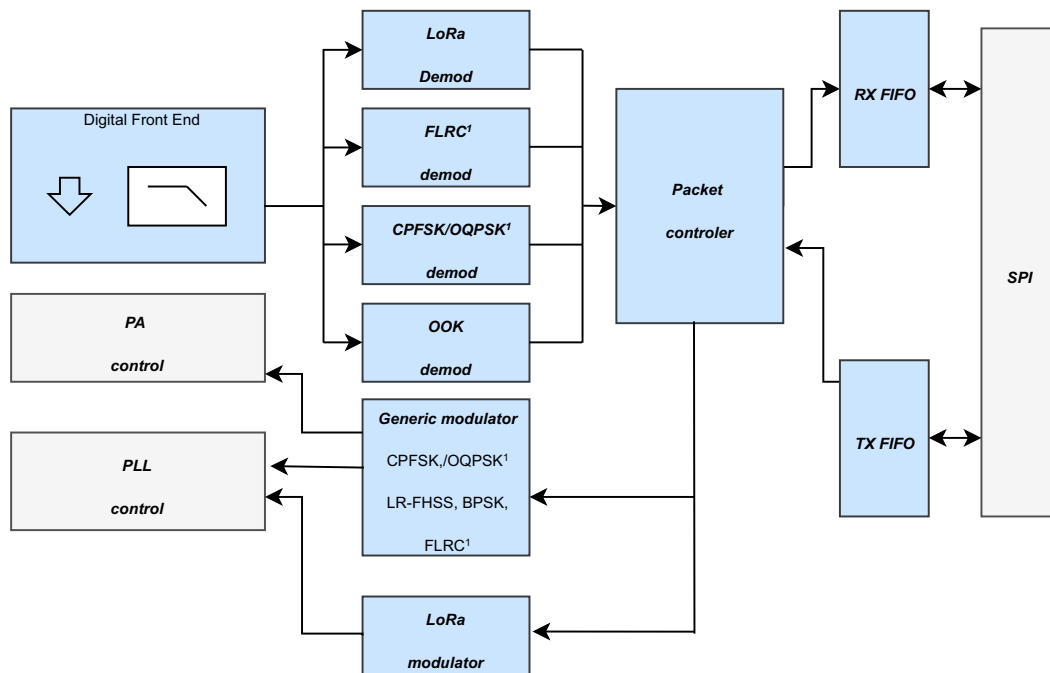
The frequency synthesizer covers the 150–2500 MHz frequency range. The chip employs a dual-path modulation technique known as “2-point modulation” to ensure an extended modulation bandwidth. This approach effectively addresses the limitations posed by the PLL bandwidth, mitigating issues related to distortion and Inter-Symbol Interference (ISI).

## 1.7 Modem

The modem architecture contains several key components:

- **Digital Front End (DIG\_FE) Block:** This block handles tasks such as decimation, channel filtering, and Received Signal Strength Indicator (RSSI) measurement. It prepares the incoming signals for further processing by the modems.
- **Generic Modulator:** Responsible for generating frequency and phase information from data obtained through the packet handler output during the Transmit (Tx) operation. This modulator is used across all supported modulations, except for LoRa, which uses its dedicated modulator.
- **LoRa Modem:** Introduces significant enhancements, including improved robustness to interference, multi Spreading Factors (SF) support, including SF12, and a novel CAD (Channel Activity Detection) mechanism. This new modem provides improved frequency drift tolerance for LDRO cases and a more reliable link via a new FEC mechanism. Additionally, this modem extends the frequency capture range, providing enhanced capabilities for LoRa communications.
- **FSK Modem:** Supports various modulations such as (G)FSK and LR FHSS, and communication protocols including WI-SUN FSK, WM-BUS, Bluetooth LE, among others.
- **FLRC Modem (LR2021):** Compatible with the SX1280 FLRC modulation scheme, this modem offers flexibility for high speed and long distance communication scenarios. Compared to the SX1280, this FLRC modem supports bit rates up to 2.6 Mbps and also operates in the sub-GHz bands.
- **OOK Modem:** OOK modulation is applied by switching the power amplifier on and off. It uses digital control and ramping features to enhance the transient power response of the OOK transmitter. Moreover, modulation shaping can be applied to further improve the narrow band response of the LR20xx transmitter.
- **Generic Packet Handler:** It supports both standard and proprietary frame formats. In the Receive (Rx) operation, its task is to extract payload information from the incoming bit-stream, perform address filtering and conduct essential checks, such as Cyclic Redundancy Check (CRC). Due to the high configurability of the packet handler, predefined settings are available for specific frame types associated with different protocols, such as Bluetooth Low Energy (including Bluetooth Low Energy coded), Wi-SUN FSK, and WM-BUS. However, custom frame formats and backward compatibility with previous Semtech chips remain achievable through dedicated low-level functions, offering developers greater flexibility in their wireless designs.
- **Automatic Gain Control (AGC):** Automatically adjusts the gain to maintain consistent and optimal signal reception.

**Figure 1-11: Modem Top View**



## 1.8 Power Distribution

### 1.8.1 Power-On Reset and Brown-Out Reset

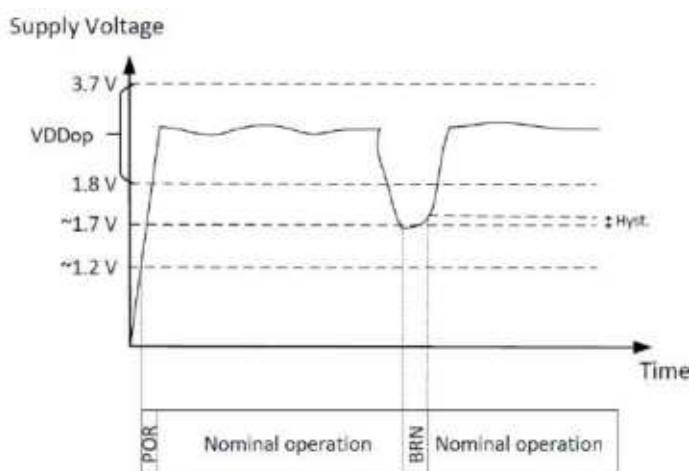
The LR2021/LR2022/LR2012 is equipped with a Power-On Reset (POR) and a Brown-Out Reset (BRN).

POR maintains the internal blocks in a reset state until a safe supply voltage level is attained, particularly during events like battery insertion.

BRN acts as a safeguard, initiating a device reset whenever the supply voltage falls below the safe operating threshold, *BRNTRHRS*. This prevents unintended operation during unsafe voltage conditions.

The POR/BRN detector is engineered with a hysteresis, *BRNHYS*, adding a level of tolerance to the reset conditions and enhancing stability during the start-up process. Refer to [Figure 1-12](#) for an illustration of the POR and BRN functions.

**Figure 1-12: POR and BRN Functions**



### 1.8.2 Low Battery Detector

The Low Battery Detector (LBD) detects when the supply voltage drops below the typical threshold, *LBO*. When this condition occurs, the LBD triggers an interrupt signal. The threshold can be changed with the *SetLbdCfg* command.

The LBD IRQ (Interrupt Request) is activated through the command *SetDiolrqConfig*, offering a direct and efficient way to notify the host system of the low battery condition.

### 1.8.3 Over Current Protection

The switching-class PA provides the best efficiency possible. One consequence of such a PA is that the consumption is dependent on the antenna load. To avoid drawing excessive current under severely detuned load conditions, the chip incorporates a protection mechanism.

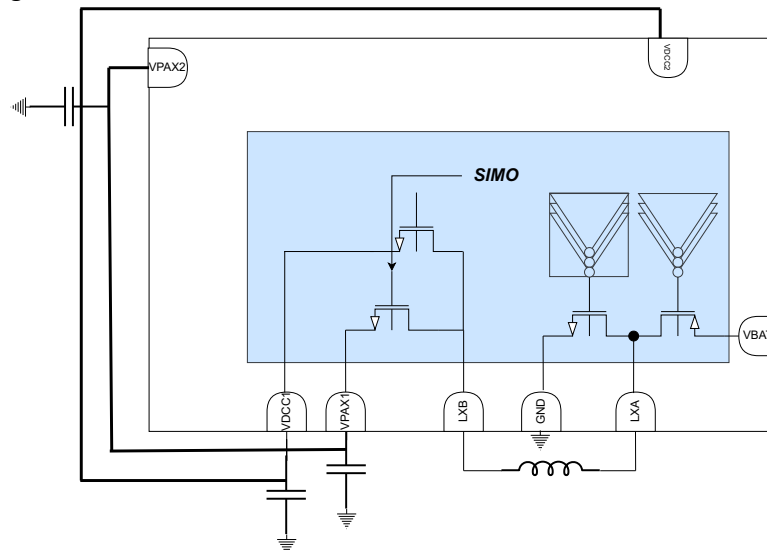
The Over Current Protection (OCP) block offers configurable current clamping values depending on the selected PA (PA\_LF or PA\_HF). The primary objective is to set the OCP threshold just below the battery's recommended limit, while also maintaining a safety margin of at least 25% above the PA output stage current. Refer to [Section 23.7](#) for OCP register configurations.

## 1.8.4 Power Supply Mode

Two power modes are available, SIMO converter for low power applications and, LDO for noise sensitive, low-cost or small size applications. The SIMO delivers two phases:

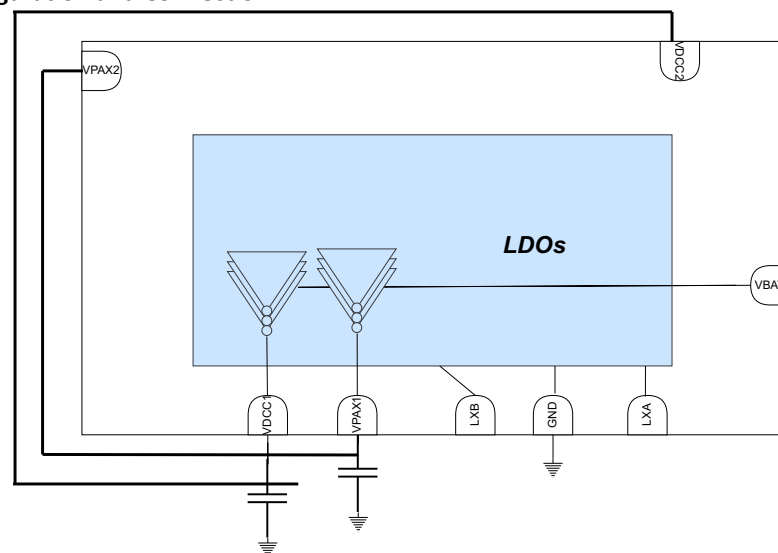
- **VDCC:** To supply the LDOs for the analogue and digital block of the circuit. Typical value for VDCC = 1.55 V, max IDCC = 20 mA. The regulated supply is available on pin VDCC1 and is connected to the rest of the chip externally via PCB and pin VDCC2.
- **VPAX:** To specifically supply the PA (through the REG PA). The regulated supply is available on VPAX1 and is connected to the PA regulator externally via PCB and pin VPAX2.

**Figure 1-13: SIMO Configuration and Connection**



The LDO mode simplifies connections for low cost applications where PA efficiency is not the primary concern, as shown in [Figure 1-14](#).

**Figure 1-14: LDO Configuration and Connection**



Refer to [Section 23.1](#) for additional information and guidance on SIMO mode recommendations.



## 1.9 Clock Sources

The low frequency RC oscillator is used for internal sequencing and as a timeout mechanism associated with several commands.

During RF operations, a high precision 32 MHz clock reference is necessary. This reference can be sourced from either an external crystal oscillator or a Temperature-Compensated Crystal Oscillator (TCXO), offering further options for meeting the specific needs of RF communication.

### 1.9.1 32 kHz RC Oscillator

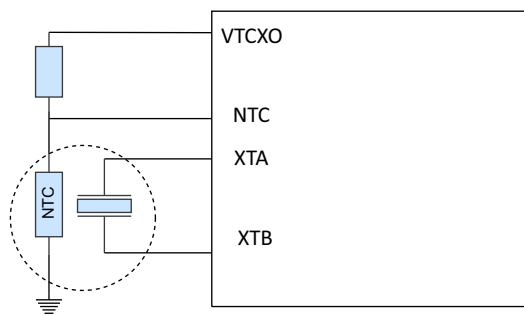
The 32 kHz RC oscillator is optionally used by the circuit in Sleep mode to wake-up the device when performing periodic or duty cycled operations. Several commands make use of this 32 kHz RC oscillator (RTC) to generate time-based events like time-outs.

### 1.9.2 32 MHz Crystal

A 32 MHz crystal oscillator provides the 32 MHz clock reference. This crystal oscillator serves as a reliable and cost-effective means of ensuring precise timing for the chip's operations.

The optional XTAL temperature compensation mechanism measures the XTAL temperature change and compensates on chip for the induced frequency shift. When the temperature compensation mechanism is used, the VTCXO pin can be used to power an external temperature sensor (R and NTC) monitoring the XTAL temperature. The NTC output is then fed into the chip via pin NTC and measured by an increase in ADC. The resulting temperature information is used by the chip to automatically compensate, to some extent, the frequency shift due to XTAL heating. This function is particularly useful when high power PA is desired on a small PCB footprint. Refer to AN1200.102 for further details.

**Figure 1-15: NTC Connection**



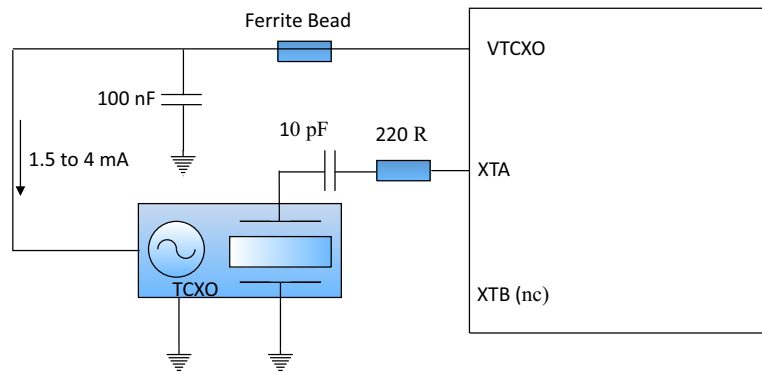
### 1.9.3 32 MHz TCXO

A Temperature Compensated Crystal Oscillator (TCXO) can be used to enhance performance in environments characterized by extreme temperature variations, when achieving precise frequency accuracy becomes challenging.

When utilizing a TCXO, the following considerations should be taken into account:

- **Connection and Components:** The TCXO should be connected to the XTA pin, see [Figure 1-16](#). The XTB pin should be unconnected. The conditions required by [ATCXO](#) should be met.
- **TCXO Power Supply:** The TCXO can be supplied by the internal regulator via the VTCXO pin. The regulated [VTCXO](#) voltage can be programmed using the [SetTcxoMode](#) command.

**Figure 1-16: TCXO Circuit Diagram**



### 1.9.4 External Clock Reference Inputs

An external low-frequency clock reference (32 kHz) can be received through the DIO11 pin, facilitated by the [ConfigLfClock](#) command. This feature allows designers to inject an accurate external clock signal into the device, enabling precise timing in various scenarios, for example Wake On Radio applications where timing needs to be accurate in synchronous protocols.

This external clock is used for all operations of the chip requiring a low frequency oscillator, such as wakeup from sleep, time-outs or control of duty cycled operation.

### 1.9.5 Output Clock

The chip can provide the application with both low and high frequency reference clocks.

- The high frequency reference clock output is derived from the system clock (the RC oscillator or the crystal/TCXO depending on the mode), which ensures reliable and accurate timing. Designers have the flexibility to choose one of the DIO pins (DIO5 to DIO11) to make this high frequency clock signal available to the application. Furthermore, the HF clock output frequency can be programmed as any 2-divider from 32 MHz down to 976.562 kHz through [ConfigClkOutputs](#). When the reference clock is provided at one of the pins of the chip, RF performances may be impacted depending on the chosen frequency, PCB layout etc.
- The low frequency reference clock is derived from the internal RC oscillator, operating at 32 kHz. This internal 32 kHz RC oscillator serves as a reliable and consistent source for the low frequency reference clock. The low frequency reference clock is made readily available on pins DIO7 to DIO11. This clock is available in all modes including Sleep mode, provided that the RC32 kHz oscillator is kept on during Sleep mode.

## 2. Pin Connection

### 2.1 I/O Description

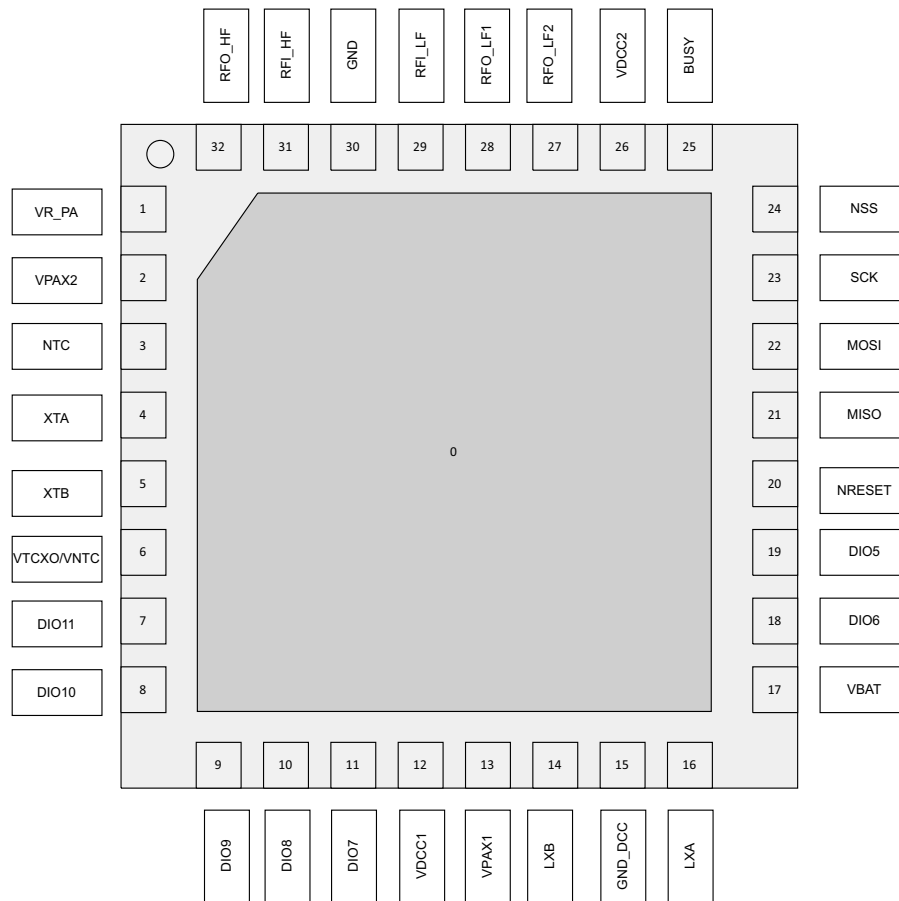
Table 2-1: Pin-out Description (Sheet 1 of 2)

| Pin | Name<br>LR2021/22 | Name<br>LR2012 | Description                                                                         |
|-----|-------------------|----------------|-------------------------------------------------------------------------------------|
| 0   | GND               | GND            | Exposed Ground pad                                                                  |
| 1   | VR_PA             | VR_PA          | Regulated power supply for all Power Amplifiers                                     |
| 2   | VPAX2             | VPAX2          | Regulated supply for Power Amplifier regulator                                      |
| 3   | NTC               | NTC            | Negative Temperature Coefficient (NTC) resistor connection                          |
| 4   | XTA               | XTA            | 32 MHz crystal oscillator connection; can be used to input external reference clock |
| 5   | XTB               | XTB            | 32 MHz crystal oscillator connection                                                |
| 6   | VTCXO/VNTC        | VTCXO/VNTC     | External TCXO supply voltage (REG_TCXO) / NTC supply                                |
| 7   | DIO11             | DIO11          | Multi-purpose digital I/O                                                           |
| 8   | DIO10             | DIO10          | Multi-purpose digital I/O                                                           |
| 9   | DIO9              | DIO9           | Multi-purpose digital I/O                                                           |
| 10  | DIO8              | DIO8           | Multi-purpose digital I/O                                                           |
| 11  | DIO7              | DIO7           | Multi-purpose digital I/O                                                           |
| 12  | VDCC1             | VDCC1          | SIMO converter output voltage. To be connected to VDCC2                             |
| 13  | VPAX1             | VPAX1          | SIMO converter output voltage for PAs supply. To be connected to VPAX2              |
| 14  | LXB               | LXB            | SIMO coil connection                                                                |
| 15  | GND_DCC           | GND_DCC        | Ground for SIMO                                                                     |
| 16  | LXA               | LXA            | SIMO coil connection                                                                |
| 17  | VBAT              | VBAT           | Main battery supply                                                                 |
| 18  | DIO6              | DIO6           | Multi-purpose digital I/O                                                           |
| 19  | DIO5              | DIO5           | Multi-purpose digital I/O                                                           |
| 20  | NRESET            | NRESET         | Reset signal, active low                                                            |
| 21  | MISO              | MISO           | SPI MISO                                                                            |
| 22  | MOSI              | MOSI           | SPI MOSI                                                                            |
| 23  | SCK               | SCK            | SPI clock                                                                           |
| 24  | NSS               | NSS            | SPI chip select                                                                     |

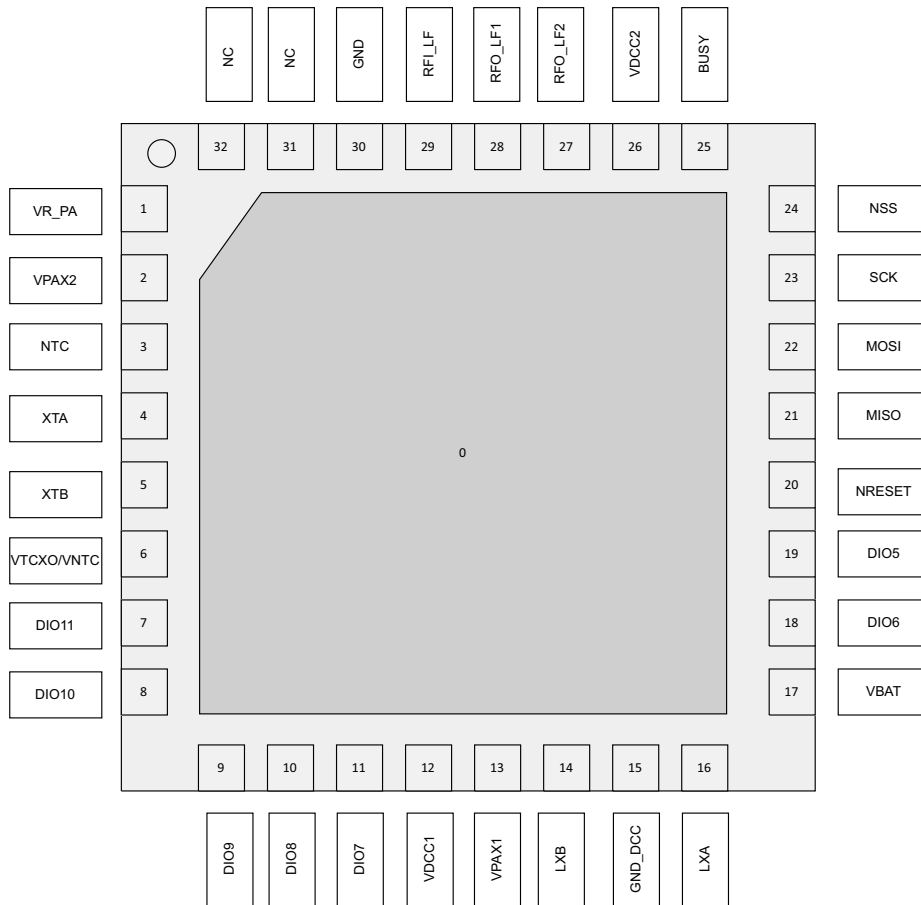
**Table 2-1: Pin-out Description (Sheet 2 of 2)**

| Pin | Name<br>LR2021/22 | Name<br>LR2012 | Description                                     |
|-----|-------------------|----------------|-------------------------------------------------|
| 25  | BUSY              | BUSY           | Busy indicator signal                           |
| 26  | VDCC2             | VDCC2          | Regulated supply input                          |
| 27  | RFO_LF2           | RFO_LF2        | RF transmitter output for LF PA                 |
| 28  | RFO_LF1           | RFO_LF1        | RF transmitter output for LF PA                 |
| 29  | RFI_LF            | RFI_LF         | RF LF receiver input                            |
| 30  | GND               | GND            | Ground                                          |
| 31  | RFI_HF            | NC             | RF HF receiver input / NC for LR2012            |
| 32  | RFO_HF            | NC             | RF transmitter output for HF PA / NC for LR2012 |

**Figure 2-1: LR2021/22 Pin-out in QFN32 Package**



**Figure 2-2: LR2012 Pin-out in QFN32 Package**



### 3. Specifications (LR20xx)

The specifications refer to all LR20xx chips unless specified otherwise in the table title.

#### 3.1 Absolute Maximum Ratings

Stresses above the values listed below may cause permanent device failure. Exposure to absolute maximum ratings for extended periods may affect device reliability and reduce product life time.

**Table 3-1: Absolute Maximum Ratings (LR20xx)**

| Symbol | Description    | Min  | Typ | Max | Unit |
|--------|----------------|------|-----|-----|------|
| VDDmr  | Supply voltage | -0.5 | -   | 3.9 | V    |
| Tmr    | Temperature    | -55  | -   | 125 | °C   |
| Pmr    | RF Input level | -    | -   | +10 | dBm  |

#### 3.2 Operating Range

Operating ranges define the limits for functional operation and parametric characteristics of the device as described in this section. Functionality outside these limits is not guaranteed.

**Table 3-2: Operating Range (LR20xx)**

| Symbol | Description                                                                              | Min | Typ | Max             | Unit |
|--------|------------------------------------------------------------------------------------------|-----|-----|-----------------|------|
| VDDop  | Supply voltage                                                                           | 1.8 | -   | 3.7             | V    |
| Top    | Ambient Operating Temperature                                                            | -40 | -   | 85              | °C   |
| Tmaxj  | Max Operating Junction Temperature                                                       | -   | -   | 105             | °C   |
| Clop   | Load capacitance on digital ports                                                        | -   | -   | 20              | pF   |
| ML     | RF Input power: Sub-GHz & HF paths                                                       | -   | -   | +6 <sup>1</sup> | dBm  |
| VSWR   | Voltage Standing Wave Ratio to guarantee PA stability.<br>Any mismatch causes power loss | -   | -   | 10:1            | -    |

1. RSSI values over 0 dBm are not reported, making RSSI clamp at 0 dBm.

### 3.3 ESD Protection and Latch-up

The LR2021/LR2022/LR2012 is a high performance radio frequency device presenting very good ESD robustness on all pins. The chip should be handled with all the necessary ESD precautions to avoid any permanent damage.

**Table 3-3: ESD and Latch-up (LR20xx)**

| Symbol  | Description                                             | Min | Typ | Max | Unit |
|---------|---------------------------------------------------------|-----|-----|-----|------|
| ESD_HBM | Human Body Model<br>JEDEC JS-001, Class 2 HBM           | -   | -   | 2.0 | kV   |
| ESD_CDM | ESD Charged Device Model<br>JEDEC ESDA JS-002, Class 2a | -   | -   | 500 | V    |
| LU      | Latch-up<br>JEDEC standard JESD78 B, class I level A    | -   | -   | 100 | mA   |

### 3.4 Electrical Specifications

The tables below give the electrical specifications of the LR2021/LR2022/LR2012 transceiver under the following conditions, unless otherwise specified:

**Note: All references to the HF path supporting 2.4 GHz ISM band and licensed L/S bands are applicable only to LR2021/LR2022**

- VDD = 3.3 V as supplied on VBAT, Temperature = 25 °C, FXOSC = 32 MHz.
- All performance given with SIMO used to power the chip, except for sleep and standby RC modes. The sensitivity is increased in LDO mode.
- FRF = 915/869 MHz for sub-GHz path.
- FRF = 2.45 GHz for HF path. (LR2021/LR2022)
- All RF impedances matched using multi-band 868–915 MHz reference design for LF and dedicated 2.4 GHz reference design.
- Transmit mode output power defined in 50  $\Omega$  load.
- Default gain step defined:
  - Non boosted: G13 with rx\_boost = 0 for LF path
  - Non boosted: G13 with rx\_boost = 4 for HF path
  - Boosted gain: G13 with rx\_boost = 7 for LF and HF
- Receiver performance tabulated with non boosted gain, unless otherwise specified.
  - Measured without RF shield on the reference design. Refer to [Section 23.2](#) for additional information.
- Bandwidth are defined as double side band.
- FSK sensitivity is given as Bit Error Rate (BER) = 0.1%, 2-level FSK modulation without pre-filtering, BR = 4.8 kbps, FDA = 5 kHz, BWF = 20 kHz.
- All FLRC sensitivity results are with the Gaussian filter (BT) = 0.5, measured with 10 bytes of payload.
- LoRa sensitivity is given as Packet Error Rate (PER) = 1%, packet of 64 bytes, preamble of 8 symbols, error correction code CR=4/5, CRC on payload enabled, explicit header, only in sub-GHz frequency range. Non boosted gain.
- LoRa/(G)FSK FLRC blocking Immunity, ACR, and co-channel rejection are given for a single tone interferer and referenced to sensitivity 3 dB, blocking tests are performed with unmodulated interferer.
- All power consumption numbers were taken with an XTAL, and indifferently at FRF = 915 MHz or 2.4 GHz, unless specified otherwise.

### 3.4.1 Power Consumption

The following tables give the total consumptions of all blocks in the specified modes.

**Table 3-4: Basic Modes Power Consumption (LR20xx)**

| Symbol   | Description                        | Conditions                       | Min | Typ  | Max | Unit |
|----------|------------------------------------|----------------------------------|-----|------|-----|------|
| IDDSL0   | Supply current in SLEEP mode       | No retention without RTC         | -   | 0.47 | -   | μA   |
| IDDSL1   |                                    | Data retention without RTC       | -   | 0.65 | -   | μA   |
| IDDSL165 |                                    | Data retention without RTC 65 °C | -   | -    | 3.5 | μA   |
| IDDSL2   |                                    | No retention with RTC            | -   | 1.28 | -   | μA   |
| IDDSL3   |                                    | Data retention with RTC          | -   | 1.4  | -   | μA   |
| IDDSBRLD | Supply current in STDBY_RC         | Internal HFRC(32 MHz) ON         | -   | 1.21 | -   | mA   |
| IDDSBX   | Supply current in STDBY_XOSC       | HF XOSC ON                       | -   | 1.79 | -   | mA   |
| IDDFSDC  | Supply current in Synthesizer mode | Frequency at 915 MHz             | -   | 3    | -   | mA   |

**Table 3-5: Transmit Mode Power Consumption (LR20xx unless specified otherwise)**

| Symbol   | Description                    | Conditions           | Min | Typ | Max | Unit |
|----------|--------------------------------|----------------------|-----|-----|-----|------|
| IDDTXLF1 | 868/915 MHz<br>LF PA           | +22 dBm              |     | 105 |     | mA   |
| IDDTXLF2 |                                | +20 dBm              | -   | 72  | -   | mA   |
| IDDTXLF3 |                                | +17 dBm              | -   | 43  | -   | mA   |
| IDDTXLF4 |                                | +16 dBm              | -   | 36  | -   | mA   |
| IDDTXLF5 |                                | +14 dBm              | -   | 27  | -   | mA   |
| IDDTXLF6 |                                | +13 dBm              | -   | 24  | -   | mA   |
| IDDTXLF7 |                                | +10 dBm              | -   | 17  | -   | mA   |
| IDDTXLF8 |                                | +10 dBm <sup>1</sup> | -   | 12  | -   | mA   |
| IDDTX8   | 434/490 MHz<br>LF PA           | +22 dBm              |     | 109 | -   | mA   |
| IDDTX9   |                                | +20 dBm              | -   | 78  | -   | mA   |
| IDDTX10  |                                | +17 dBm              | -   | 43  | -   | mA   |
| IDDTX11  |                                | +16 dBm              | -   | 37  | -   | mA   |
| IDDTX12  |                                | +14 dBm              | -   | 28  | -   | mA   |
| IDDTX13  |                                | +13 dBm              | -   | 25  | -   | mA   |
| IDDTXHF  | 2.4 GHz, HF PA (LR2021/LR2022) | +12 dBm              |     | 23  |     | mA   |

1. For 10 dBm, only PA\_HF is used with dedicated matching.



Table 3-6: Receive Mode Power Consumption (LR20xx)

| Symbol      | Description                                       | Conditions                                   | Min | Typ  | Max | Unit |
|-------------|---------------------------------------------------|----------------------------------------------|-----|------|-----|------|
| IDDRXLFFD1  | Supply current in Receive mode                    | BRF = 0.6 kbps, FDA = 0.8 kHz, BWF = 4 kHz   | -   | 5.5  | -   | mA   |
| IDDRXLFFD2  |                                                   | BRF = 1.2 kbps, FDA = 5 kHz, BWF = 20 kHz    | -   | 5.5  | -   | mA   |
| IDDRXLFFD3  |                                                   | BRF = 4.8 kbps, FDA = 5 kHz, BWF = 20 kHz    | -   | 5.6  | -   | mA   |
| IDDRXLFFD4  |                                                   | BRF = 38.4 kbps, FDA = 40 kHz, BWF = 160 kHz | -   | 5.7  | -   | mA   |
| IDDRXLFFD5  |                                                   | BRF = 50 kbps, FDA = 12.5 kHz, BWF = 100 kHz | -   | 5.6  | -   | mA   |
| IDDRXLFFD6  |                                                   | BRF = 100 kbps, FDA = 50 kHz, BWF = 200 kHz  | -   | 5.7  | -   | mA   |
| IDDRXLFFD7  |                                                   | BRF = 250 kbps, FDA = 125 kHz, BWF = 500 kHz | -   | 5.9  | -   | mA   |
| IDDRXLFD2   |                                                   | MSK 2 Mbps                                   | -   | 7.9  | -   | mA   |
| IDDRXLFFLD1 |                                                   | FLRC 260 kbps                                | -   | 6.6  | -   | mA   |
| IDDRXLFFLD2 |                                                   | FLRC 2600 kbps                               | -   | 10.4 | -   | mA   |
| IDDRXLFL4   |                                                   | LoRa SF7 125 kHz                             | -   | 5.7  | -   | mA   |
| IDDRXLFL5   |                                                   | LoRa SF7 250 kHz                             | -   | 5.7  | -   | mA   |
| IDDRXLFL8   |                                                   | LoRa SF7 406 kHz                             | -   | 6    | -   | mA   |
| IDDRXLFL6   |                                                   | LoRa SF7 500 kHz                             | -   | 5.9  | -   | mA   |
| IDDRXLFL9   |                                                   | LoRa SF7 812 kHz                             | -   | 6.9  | -   | mA   |
| IDDRXLFL7   |                                                   | LoRa SF7 1000 kHz                            | -   | 6.5  | -   | mA   |
| IDDELTA1    | Additional consumption of Rx boosted mode LF path | Boosted LF vs default LF                     | -   | 1.2  | -   | mA   |

**Table 3-7: 2.4 GHz Specific Receive Mode Power Consumption (LR2021/LR2022)**

| Symbol      | Description                                   | Conditions               | Min | Typ  | Max | Unit |
|-------------|-----------------------------------------------|--------------------------|-----|------|-----|------|
| IDDRXHFD1   | Supply current in Receive mode<br>FRF=2.4 GHz | MSK 2 Mbps               | -   | 9.2  | -   | mA   |
| IDDRXHFD2   |                                               | MSK 1 Mbps               | -   | 7.6  | -   | mA   |
| IDDRXHFFLD1 |                                               | FLRC 260 kbps            | -   | 7.3  | -   | mA   |
| IDDRXHFFLD2 |                                               | FLRC 2600 kbps           | -   | 10.9 | -   | mA   |
| IDDRXHFL4   |                                               | LoRa SF7 125 kHz         | -   | 6.4  | -   | mA   |
| IDDRXHFL5   |                                               | LoRa SF7 250 kHz         | -   | 6.4  | -   | mA   |
| IDDRXHFL8   |                                               | LoRa SF7 406 kHz         | -   | 6.8  | -   | mA   |
| IDDRXHFL6   |                                               | LoRa SF7 500 kHz         | -   | 6.6  | -   | mA   |
| IDDRXHFL9   |                                               | LoRa SF7 812 kHz         | -   | 7.5  | -   | mA   |
| IDDRXHFL7   |                                               | LoRa SF7 1000 kHz        | -   | 7.1  | -   | mA   |
| IDDRXHFL10  |                                               | O-QPSK 250 kbps          |     | 10.5 |     | mA   |
| IDDDelta2   | Boosted mode Additional<br>current in HF      | Boosted HF vs default HF |     | 0.4  |     | mA   |

### 3.4.2 General Specifications

Table 3-8: General Specifications (LR20xx unless specified otherwise) (Sheet 1 of 2)

| Symbol   | Description                                                         | Conditions                            | Min  | Typ    | Max  | Unit   |
|----------|---------------------------------------------------------------------|---------------------------------------|------|--------|------|--------|
| FRANGE   | PLL addressable frequency range                                     |                                       | 150  |        | 2500 | MHz    |
| FSTEP1   | Synthesizer frequency step                                          | FXOSC/ 2 <sup>25</sup>                | -    | 0.95   | -    | Hz     |
| PHNLF1   | Synthesizer phase noise <sup>1</sup><br>sub-GHz band 915 MHz        | 100 kHz offset                        |      | -104   | -    | dBc/Hz |
| PHNLF2   |                                                                     | 300 kHz offset                        | -    | -112.5 | -    | dBc/Hz |
| PHNLF3   |                                                                     | 1 MHz offset                          | -    | -121   | -    | dBc/Hz |
| PHNLF4   |                                                                     | 10 MHz offset                         | -    | -133.5 | -    | dBc/Hz |
| PHNLF5   | Synthesizer phase noise <sup>1</sup><br>sub-GHz band 490 MHz        | 100 kHz offset                        | -    | -108   | -    | dBc/Hz |
| PHNLF6   |                                                                     | 300 kHz offset                        | -    | -117   | -    | dBc/Hz |
| PHNLF7   |                                                                     | 1 MHz offset                          | -    | -126   | -    | dBc/Hz |
| PHNLF8   |                                                                     | 10 MHz offset                         | -    | -135   | -    | dBc/Hz |
| PHNH1    | Synthesizer phase noise <sup>1</sup><br>2.45 GHz<br>(LR2021/LR2022) | 100 kHz offset                        | -    | -93    | -    | dBc/Hz |
| PHNH2    |                                                                     | 300 kHz offset                        |      | -102.5 | -    | dBc/Hz |
| PHNH3    |                                                                     | 1 MHz offset                          | -    | -113.5 | -    | dBc/Hz |
| PHNH4    |                                                                     | 10 MHz offset                         | -    | -122.5 | -    | dBc/Hz |
| FRRF1    | Max. frequency range, LF inputs                                     | RFI_LF                                | 150  | -      | 960  | MHz    |
| FRRF2    | Max. frequency range HF                                             | RFI_HF                                | 1500 | -      | 2500 | MHz    |
| FRRF3    | Max.frequency range, LF                                             | RFO_LF1/2                             | 150  | -      | 960  | MHz    |
| FRRF4    | Max.frequency range, HF                                             | RFO_HF                                | 1500 | -      | 2500 | MHz    |
| HFOSCCP1 | HF Crystal oscillator supported<br>off-chip capacitance             | Single ended on XTA and XTB           | 0.2  | 1      | 3    | pF     |
| HFOSCCP2 |                                                                     | Differential XTA to XTB               | -    | 0.1    | 0.3  | pF     |
| OSCTRM1  | HF Crystal oscillator trimming<br>step                              | Normal                                | -    | 1      | -    | ppm    |
| OSCTRM2  |                                                                     |                                       |      |        |      |        |
| LFCLKRF  | Rise/fall time for external LF clock                                |                                       | -    | -      | 5    | ns     |
| BRFSK1   | Bit rate, FSK                                                       | For frequency range 433 MHz - 2.5 GHz | 0.5  | -      | 2000 | kbps   |
| BRFSK2   | Programmable                                                        | For frequency range 150 MHz - 433 MHz | 0.5  | -      | 1000 | kbps   |
| FDAFSK1  | Frequency deviation, FSK                                            | For frequency range 433 MHz - 2.5 GHz | 0.6  | -      | 500  | kHz    |
| FDAFSK2  | Programmable                                                        | For frequency range 150 MHz - 433 MHz | 0.6  | -      | 330  | kHz    |
| BRFLRC   | Effective bit rate                                                  | For frequency range 150 MHz - 470 MHz | 0.13 | -      | 1.3  | Mbps   |
|          |                                                                     | For frequency range 470 MHz - 2.5 GHz | 0.13 | -      | 2.6  | Mbps   |
| MODIND1  | Modulation index                                                    | GFSK                                  | 0.3  | -      | 6.2  | -      |
| MODIND2  | Programmable                                                        | FSK                                   | 0.3  | -      | 8.3  | -      |

**Table 3-8: General Specifications (LR20xx unless specified otherwise) (Sheet 2 of 2)**

| Symbol       | Description                                            | Conditions                                | Min   | Typ      | Max    | Unit     |
|--------------|--------------------------------------------------------|-------------------------------------------|-------|----------|--------|----------|
| BT           | 3-dB bandwidth-symbol time product, Programmable       | Gaussian filtering enabled                | 0.3   | -        | 1      | -        |
| BRLORA1      | Bit rate, LoRa                                         | SF12, BWLORA = 31.25 kHz, CR = 4/8        | -     | 45.8     | -      | bps      |
| BRLORA2      |                                                        | SF5, BWLORA = 500 kHz, CR = 4/5           | -     | 62.5     | -      | kbps     |
| BRLORA3      |                                                        | SF5, BWLORA = 1000 kHz, CR = 4/5          | -     | 125      | -      | kbps     |
| BRLORA4      |                                                        | SF12, BWLORA = 125 kHz, CR = 4/5          | -     | 292      | -      | bps      |
|              |                                                        | SF5, BWLORA = 812 kHz, CR = 4/5           | -     | 101.5    | -      | kbps     |
| BWLORA1      | Signal BW, LoRa                                        | For frequency range 434 MHz - 2500 MHz    | 7.825 | -        | 1000   | kHz      |
| BWLORA2      |                                                        | For frequency range 150 MHz - 433 MHz     | 7.825 | -        | 500    | kHz      |
| SF           | Spreading factor coefficient, LoRa chips/symbol = 2^SF | Programmable                              | 5     | -        | 12     | -        |
| BRB1         | Bit rate, BPSK                                         | Programmable, only 2 BR available Tx only | 0.1   | -        | 0.6    | kbps     |
| BRLRFHSS     | Bit rate, LR-FHSS                                      | Raw bit rate                              | -     | 488      | -      | bps      |
| BROOK        | Bit rate, OOK                                          | Programmable                              | 1.2   | -        | 32.768 | kbps     |
| BROOK_R      | Bit rate, OOK in Rx mode                               | Programmable                              | 1.2   | -        | 2      | Mbps     |
| BWF          | DSB channel filter BW, FSK/GMSK/OOK                    | Programmable                              | 3.4   | -        | 2666   | kHz      |
| LOSS_S_BOOST | Sensitivity loss when Rx boost is off                  | HF<br>LF                                  |       | 0.5<br>1 |        | dB<br>dB |

1. Phase Noise specifications are given for the recommended PLL bandwidth to be used for the specific modulation/ bit rate.

### 3.4.3 Receive Mode Specifications

The following table provides the receiver specification for Wireless M-Bus @80% PER 20 bytes. G13 rx\_boost=7 for sensitivity measurements.

**Table 3-9: WM-Bus Receiver Specification (LR20xx) (Sheet 1 of 3)**

| Symbol             | Description                         | Conditions                      | Min | Typ       | Max | Unit |
|--------------------|-------------------------------------|---------------------------------|-----|-----------|-----|------|
| WM_N2P4_S          | Sensitivity                         | BR = 2.4 kbps,<br>FDA = 2.4 kHz |     | -124      |     | dBm  |
| WM_N2P4_FREQERR    | Max frequency error tolerated       | FRF = 169 MHz                   |     | +/- 2.5   |     | kHz  |
| WM_N2P4_BERR       | Bit rate error tolerated            |                                 |     | +/- 0.01  |     | %    |
| WM_N2P4_FDEVERR    | Frequency deviation error tolerated |                                 |     | +/- 0.72  |     | kHz  |
| WM_N2P4_B_5P       | Blocker +/- 5% RF                   | Wanted at sensi + 3 dB          |     | -30       |     | dBm  |
| WM_N2P4_B_2M       | Blocker +/- 2 MHz                   | Wanted at sensi + 3 dB          |     | -42       |     | dBm  |
| WM_N2P4_B_10M      | Blocker +/- 10 MHz                  | Wanted at sensi + 3 dB          |     | -32       |     | dBm  |
| WM_N4P8_S          | Sensitivity                         | BR = 4.8 kbps,<br>FDA = 2.4 kHz |     | -124      |     | dBm  |
| WM_N4P8_FREQERR    | Max frequency error tolerated       | FRF = 169 MHz                   |     | +/- 2.5   |     | kHz  |
| WM_N4P8_BERR       | Bit rate error tolerated            |                                 |     | +/- 0.01  |     | %    |
| WM_N4P8_FDEVERR    | Frequency deviation error tolerated |                                 |     | +/- 0.72  |     | kHz  |
| WM_N4P8_B_5P       | Blocker +/- 5% RF                   | Wanted at sensi + 3 dB          |     | -29       |     | dBm  |
| WM_N4P8_B_2M       | Blocker +/- 2 MHz                   | Wanted at sensi + 3 dB          |     | -42       |     | dBm  |
| WM_N4P8_B_10M      | Blocker +/- 10 MHz                  | Wanted at sensi + 3 dB          |     | -32       |     | dBm  |
| WM_F2P4_S          | Sensitivity                         | BR = 2.4 kbps,<br>FDA = 5.5 kHz |     | -123      |     | dBm  |
| WM_F2P4_FREQERR    | Max frequency error tolerated       | FRF = 434 MHz                   |     | +/-16     |     | ppm  |
| WM_F2P4_BERR       | Bit rate error tolerated            |                                 |     | +/- 0.01  |     | %    |
| WM_F2P4_FDEVERR    | Frequency deviation error tolerated |                                 |     | -0.7/+1.5 |     | kHz  |
| WM_F2P4_B_5P       | Blocker +/- 5% RF                   | Wanted at sensi + 3 dB          |     | -21       |     | dBm  |
| WM_F2P4_B_2M       | Blocker +/- 2 MHz                   | Wanted at sensi + 3 dB          |     | -42       |     | dBm  |
| WM_F2P4_B_10M      | Blocker +/- 10 MHz                  | Wanted at sensi + 3 dB          |     | -27       |     | dBm  |
| WM_C2O2M50_S       | Sensitivity                         | BR = 50 kbps,<br>FDA = 25 kHz   |     | -113      |     | dBm  |
| WM_C2O2M50_FREQERR | Max frequency error tolerated       | FRF = 868 MHz                   |     | +/-25     |     | ppm  |
| WM_C2O2M50_BERR    | Bit rate error tolerated            |                                 |     | +/- 0.01  |     | %    |
| WM_C2O2M50_FDEVERR | Frequency deviation error tolerated |                                 |     | +/-6.25   |     | kHz  |
| WM_C2O2M50_B_5P    | Blocker +/- 5% RF                   | Wanted at sensi + 3 dB          |     | -21       |     | dBm  |

Table 3-9: WM-Bus Receiver Specification (LR20xx) (Sheet 2 of 3)

| Symbol                | Description                         | Conditions                        | Min | Typ      | Max | Unit |
|-----------------------|-------------------------------------|-----------------------------------|-----|----------|-----|------|
| WM_C2O2M50_B_2M       | Blocker +/- 2 MHz                   | Wanted at sensi + 3 dB            |     | -40      |     | dBm  |
| WM_C2O2M50_B_10M      | Blocker +/- 10 MHz                  | Wanted at sensi + 3 dB            |     | -24      |     | dBm  |
| WM_C1C2M2O100_S       | Sensitivity                         | BR = 100 kbps,<br>FDA = 45 kHz    |     | -111     |     | dBm  |
| WM_C1C2M2O100_FREQERR | Max frequency error tolerated       | FRF = 868 MHz                     |     | +/-25    |     | ppm  |
| WM_C1C2M2O100_BERR    | Bit rate error tolerated            |                                   |     | +/- 0.01 |     | %    |
| WM_C1C2M2O100_FDEVERR | Frequency deviation error tolerated |                                   |     | +/-11.25 |     | kHz  |
| WM_C1C2M2O100_B_5P    | Blocker +/- 5% RF                   | Wanted at sensi + 3 dB            |     | -19      |     | dBm  |
| WM_C1C2M2O100_B_2M    | Blocker +/- 2 MHz                   | Wanted at sensi + 3 dB            |     | -40      |     | dBm  |
| WM_C1C2M2O100_B_10M   | Blocker +/- 10 MHz                  | Wanted at sensi + 3 dB            |     | -27      |     | dBm  |
| WM_R2_4P8_S           | Sensitivity                         | CR = 4.8 kcps, FDA = 6 kHz        |     | -122     |     | dBm  |
| WM_R2_4P8_FREQERR     | Max frequency error tolerated       | FRF = 868 MHz                     |     | +/-20    |     | ppm  |
| WM_R2_4P8_BERR        | Bit rate error tolerated            |                                   |     | +/- 2    |     | %    |
| WM_R2_4P8_FDEVERR     | Frequency deviation error tolerated |                                   |     | +/-1.2   |     | kHz  |
| WM_R2_4P8_B_5P        | Blocker +/- 5% RF                   | Wanted at sensi + 3 dB            |     | -15      |     | dBm  |
| WM_R2_4P8_B_2M        | Blocker +/- 2 MHz                   | Wanted at sensi + 3 dB            |     | -38      |     | dBm  |
| WM_R2_4P8_B_10M       | Blocker +/- 10 MHz                  | Wanted at sensi + 3 dB            |     | -21      |     | dBm  |
| WM_T2O2M32_S          | Sensitivity                         | CR = 32.768 kcps,<br>FDA = 50 kHz |     | -113     |     | dBm  |
| WM_T2O2M32_FREQERR    | Max frequency error tolerated       | FRF = 868 MHz                     |     | +/-25    |     | ppm  |
| WM_T2O2M32_BERR       | Bit rate error tolerated            |                                   |     | +/- 1.5  |     | %    |
| WM_T2O2M32_FDEVERR    | Frequency deviation error tolerated |                                   |     | -10/+30  |     | kHz  |
| WM_T2O2M32_B_5P       | Blocker +/- 5% RF                   | Wanted at sensi + 3 dB            |     | -19      |     | dBm  |
| WM_T2O2M32_B_2M       | Blocker +/- 2 MHz                   | Wanted at sensi + 3 dB            |     | -42      |     | dBm  |
| WM_T2O2M32_B_10M      | Blocker +/- 10 MHz                  | Wanted at sensi + 3 dB            |     | -24      |     | dBm  |
| WM_T1T2M2O100_S       | Sensitivity                         | CR = 100 kcps,<br>FDA = 50 kHz    |     | -110     |     | dBm  |
| WM_T1T2M2O100_FREQERR | Max frequency error tolerated       | FRF = 868 MHz                     |     | +/-60    |     | ppm  |
| WM_T1T2M2O100_BERR    | Bit rate error tolerated            |                                   |     | +/- 1    |     | %    |
| WM_T1T2M2O100_FDEVERR | Frequency deviation error tolerated |                                   |     | -10/+30  |     | kHz  |
| WM_T1T2M2O100_B_5P    | Blocker +/- 5% RF                   | Wanted at sensi + 3 dB            |     | -19      |     | dBm  |

Table 3-9: WM-Bus Receiver Specification (LR20xx) (Sheet 3 of 3)

| Symbol              | Description                         | Conditions                        | Min | Typ     | Max | Unit |
|---------------------|-------------------------------------|-----------------------------------|-----|---------|-----|------|
| WM_T1T2M2O100_B_2M  | Blocker +/- 2 MHz                   | Wanted at sensi + 3 dB            |     | -41     |     | dBm  |
| WM_T1T2M2O100_B_10M | Blocker +/- 10 MHz                  | Wanted at sensi + 3 dB            |     | -28     |     | dBm  |
| WM_S1S232_S         | Sensitivity                         | CR = 32.768 kcps,<br>FDA = 50 kHz |     | -113    |     | dBm  |
| WM_S1S232_FREQERR   | Max frequency error tolerated       | FRF = 868 MHz                     |     | +/-60   |     | ppm  |
| WM_S1S232_BERR      | Bit rate error tolerated            |                                   |     | +/- 2   |     | %    |
| WM_S1S232_FDEVERR   | Frequency deviation error tolerated |                                   |     | -10/+30 |     | kHz  |
| WM_S1S232_B_5P      | Blocker +/- 5% RF                   | Wanted at sensi + 3 dB            |     | -21     |     | dBm  |
| WM_S1S232_B_2M      | Blocker +/- 2 MHz                   | Wanted at sensi + 3 dB            |     | -37     |     | dBm  |
| WM_S1S232_B_10M     | Blocker +/- 10 MHz                  | Wanted at sensi + 3 dB            |     | -23     |     | dBm  |

The following table provides the receiver specification for Z-Wave without boost (unless otherwise noted) @1% PER 20 bytes.

**Table 3-10: Z-Wave 1% PER (LR2021)**

| Symbol          | Description                   | Conditions                                              | Min | Typ   | Max | Unit |
|-----------------|-------------------------------|---------------------------------------------------------|-----|-------|-----|------|
| ZW_R1_S         | Sensitivity 2-FSK             | BR = 9.6 kbps, FDA = 20 kHz, Manchester, G13 rx_boost=7 | -   | -114  | -   | dBm  |
| ZW_R1_CCA_THRES | CCA threshold                 |                                                         |     | -80   |     | dBm  |
| ZW_R1_FREQERR   | Max frequency error tolerated | FRF = 908.42 MHz                                        |     | +/-27 |     | ppm  |
| ZW_R1_B1        | Blocking +- 1 MHz             | Wanted signal at -92 dBm                                |     | -32   |     | dBm  |
| ZW_R1_B2        | Blocking +- 2 MHz             | Wanted signal at -92 dBm                                |     | -34   |     | dBm  |
| ZW_R1_B5        | Blocking +- 5 MHz             | Wanted signal at -92 dBm                                |     | -25   |     | dBm  |
| ZW_R1_B10       | Blocking +- 10 MHz            | Wanted signal at -92 dBm                                |     | -20   |     | dBm  |
| ZW_R1_FDEVERR   | Frequency deviation variation |                                                         |     | +/-20 |     | %    |
| ZW_R2_S         | Sensitivity 2-FSK             | BR = 40 kbps, FDA = 20 kHz, G13 rx_boost=7              | -   | -110  | -   | dBm  |
| ZW_R2_CCA_THRES | CCA threshold                 |                                                         |     | -80   |     | dBm  |
| ZW_R2_FREQERR   | Max frequency error tolerated | FRF = 908.4 MHz                                         |     | +/-27 |     | ppm  |
| ZW_R2_B1        | Blocking +- 1 MHz             | Wanted signal at -89 dBm                                |     | -32   |     | dBm  |
| ZW_R2_B2        | Blocking +- 2 MHz             | Wanted signal at -89 dBm                                |     | -31   |     | dBm  |
| ZW_R2_B5        | Blocking +- 5 MHz             | Wanted signal at -89 dBm                                |     | -23   |     | dBm  |
| ZW_R2_B10       | Blocking +- 10 MHz            | Wanted signal at -89 dBm                                |     | -20   |     | dBm  |
| ZW_R2_FDEVERR   | Frequency deviation variation |                                                         |     | +/-20 |     | %    |
| ZW_R3_S         | Sensitivity 2-GFSK, BT = 0.6  | BR = 100 kbps, FDA = 24 kHz, BT = 0.6, G13 rx_boost=7   | -   | -110  | -   | dBm  |
| ZW_R3_CCA_THRES | CCA threshold                 |                                                         |     | -80   |     | dBm  |
| ZW_R3_FREQERR   | Max frequency error tolerated | FRF = 916 MHz                                           |     | +/-27 |     | ppm  |
| ZW_R3_B1        | Blocking +- 1 MHz             | Wanted signal at -86 dBm                                |     | -33   |     | dBm  |
| ZW_R3_B2        | Blocking +- 2 MHz             | Wanted signal at -86 dBm                                |     | -29   |     | dBm  |
| ZW_R3_B5        | Blocking +- 5 MHz             | Wanted signal at -86 dBm                                |     | -21   |     | dBm  |
| ZW_R3_B10       | Blocking +- 10 MHz            | Wanted signal at -86 dBm                                |     | -20   |     | dBm  |
| ZW_R3_FDEVERR   | Frequency deviation variation |                                                         |     | +/-20 |     | %    |
| ZWLR1           | Sensitivity O-QPSK            | BR = 100 kbps 2 Mcps, FRF=912 MHz, G13 rx_boost=7       | -   | -108  | -   | dBm  |
| ZW_LR_CCA_THRES | CCA threshold                 |                                                         |     | -60   |     | dBm  |
| ZW_LR_B2        | Blocking +- 2 MHz             | Wanted signal at -89 dBm                                |     | -38   |     | dBm  |
| ZW_LR_B5        | Blocking +- 5 MHz             | Wanted signal at -89 dBm                                |     | -30   |     | dBm  |
| ZW_LR_B10       | Blocking +- 10 MHz            | Wanted signal at -89 dBm                                |     | -34   |     | dBm  |



The following table provides the receiver specification for Wi-SUN FSK @10% PER 250 bytes, FRF = 915 MHz. G13 rx\_boost=7 for sensitivity measurements.

**Table 3-11: Wi-SUN FSK 10% PER (LR20xx) (Sheet 1 of 2)**

| Symbol      | Description                       | Conditions                               | Min | Typ    | Max | Unit |
|-------------|-----------------------------------|------------------------------------------|-----|--------|-----|------|
| WS_1a_S     | Sensitivity 2-FSK                 |                                          | -   | -113.5 | -   | dBm  |
| WS_1a_ACRM  | Lower adjacent channel rejection  | SR = 50 kbps,<br>FDA = 12.5 kHz, no FEC  |     | 31     |     | dB   |
| WS_1a_ACR   | Upper adjacent channel rejection  |                                          |     | 32     |     | dB   |
| WS_1a_ALCRM | Lower alternate channel rejection |                                          |     | 48     |     | dB   |
| WS_1a_ALCR  | Upper alternate channel rejection |                                          |     | 48     |     | dB   |
| WS_1b_S     | Sensitivity 2-FSK                 |                                          | -   | -110.5 | -   | dBm  |
| WS_1b_ACRM  | Lower adjacent channel rejection  | SR = 50 kbps,<br>FDA = 25 kHz, no FEC    |     | 48     |     | dB   |
| WS_1b_ACR   | Upper adjacent channel rejection  |                                          |     | 48     |     | dB   |
| WS_1b_ALCRM | Lower alternate channel rejection |                                          |     | 50     |     | dB   |
| WS_1b_ALCR  | Upper alternate channel rejection |                                          |     | 51     |     | dB   |
| WS_2a_S     | Sensitivity 2-FSK                 |                                          | -   | -110   | -   | dBm  |
| WS_2a_ACRM  | Lower adjacent channel rejection  | SR = 100 kbps,<br>FDA = 25 kHz, no FEC   |     | 31     |     | dB   |
| WS_2a_ACR   | Upper adjacent channel rejection  |                                          | -   | 32     | -   | dB   |
| WS_2a_ALCRM | Lower alternate channel rejection |                                          | -   | 47     | -   | dB   |
| WS_2a_ALCR  | Upper alternate channel rejection |                                          | -   | 52     | -   | dB   |
| WS_2b_S     | Sensitivity 2-FSK                 |                                          | -   | -108   | -   | dBm  |
| WS_2b_ACRM  | Lower adjacent channel rejection  | SR = 100 kbps,<br>FDA = 50 kHz, no FEC   |     | 48     |     | dB   |
| WS_2b_ACR   | Upper adjacent channel rejection  |                                          |     | 49     |     | dB   |
| WS_2b_ALCRM | Lower alternate channel rejection |                                          |     | 41     |     | dB   |
| WS_2b_ALCR  | Upper alternate channel rejection |                                          |     | 55     |     | dB   |
| WS_3_S      | Sensitivity 2-FSK                 |                                          | -   | -108   | -   | dBm  |
| WS_3_ACRM   | Lower adjacent channel rejection  | SR = 150 kbps,<br>FDA = 37.5 kHz, no FEC |     | 15     |     | dB   |
| WS_3_ACR    | Upper adjacent channel rejection  |                                          |     | 15     | -   | dB   |
| WS_3_ALCRM  | Lower alternate channel rejection |                                          | -   | 44     | -   | dB   |
| WS_3_ALCR   | Upper alternate channel rejection |                                          |     | 51     |     | dB   |

**Table 3-11: Wi-SUN FSK 10% PER (LR20xx) (Sheet 2 of 2)**

| Symbol      | Description                       | Conditions                                               | Min | Typ    | Max | Unit |
|-------------|-----------------------------------|----------------------------------------------------------|-----|--------|-----|------|
| WS_4a_S     | Sensitivity 2-FSK                 |                                                          |     | -107.5 |     | dBm  |
| WS_4a_ACRM  | Lower adjacent channel rejection  | SR = 200 kbps,<br>FDA = 50 kHz, no FEC                   |     | 45     |     | dB   |
| WS_4a_ACR   | Upper adjacent channel rejection  |                                                          |     | 50     |     | dB   |
| WS_4a_ALCRM | Lower alternate channel rejection |                                                          |     | 41     |     | dB   |
| WS_4a_ALCR  | Upper alternate channel rejection |                                                          |     | 53     |     | dB   |
| WS_4b_S     | Sensitivity 2-FSK                 |                                                          |     | -104.5 |     | dBm  |
| WS_4b_ACRM  | Lower adjacent channel rejection  | SR = 200 kbps,<br>FDA = 100 kHz, no FEC                  |     | 45     |     | dB   |
| WS_4b_ACR   | Upper adjacent channel rejection  |                                                          |     | 52     |     | dB   |
| WS_4b_ALCRM | Lower alternate channel rejection |                                                          |     | 50     |     | dB   |
| WS_4b_ALCR  | Upper alternate channel rejection |                                                          |     | 54     |     | dB   |
| WS_5_S      | Sensitivity 2-FSK                 |                                                          |     | -106   |     | dBm  |
| WS_5_ACRM   | Lower adjacent channel rejection  | SR = 300 kbps,<br>FDA = 75 kHz, no FEC                   |     | 30     |     | dB   |
| WS_5_ACR    | Upper adjacent channel rejection  |                                                          |     | 30     |     | dB   |
| WS_5_ALCRM  | Lower alternate channel rejection |                                                          |     | 35     |     | dB   |
| WS_5_ALCR   | Upper alternate channel rejection |                                                          |     | 51     |     | dB   |
| WS_FDEVERR  | Frequency deviation variation     | Mode Wi-SUN 1a, 1b,2a,<br>2b,3, 4, 4b,5<br>FRF = 915 MHz |     | +/-30  |     | %    |
| WS_BRERR    | Bit rate error                    |                                                          |     | +/-100 |     | ppm  |
| WS_FERR     | Max frequency error tolerated     |                                                          |     | +20    |     | ppm  |

The following table provides the receiver specification for FLRC Sub-GHz @1% PER.

**Table 3-12: FLRC Sub-GHz 1% PER (LR2021)<sup>1</sup> (Sheet 1 of 2)**

| Symbol               | Description                   | Conditions                                                                                       | Min | Typ    | Max | Unit |
|----------------------|-------------------------------|--------------------------------------------------------------------------------------------------|-----|--------|-----|------|
| FLRC_2600_CR05_915_S | Sensitivity 1% PER 915 MHz    | BRF = 2600 kbps, CR = 3/4,<br>BWF = 2666 kHz<br>Effective bit rate= 1.95 Mbps,<br>G13 rx_boost=7 |     | -100.5 |     | dBm  |
| FLRC_2600_CR05_FERR  | Max frequency error tolerated | FRF = 915 MHz                                                                                    |     | +/-150 |     | kHz  |
| FLRC_2600_CR05_PRE   | Min preamble length           |                                                                                                  |     | 24     |     | bits |
| FLRC_2080_CR05_915_S | Sensitivity 1% PER 915 MHz    | BRF = 2080 kbps, CR = 3/4,<br>BWF = 2666 kHz<br>Effective bit rate= 1.56 Mbps,<br>G13 rx_boost=7 |     | -101.5 |     | dBm  |
| FLRC_2080_CR05_FERR  | Max frequency error tolerated | FRF = 915 MHz                                                                                    |     | +/-150 |     | kHz  |
| FLRC_2080_CR05_PRE   | Min preamble length           |                                                                                                  |     | 20     |     | bits |
| FLRC_1300_CR05_915_S | Sensitivity 1% PER 915 MHz    | BRF = 1300 kbps, CR = 3/4,<br>BWF = 1200 kHz<br>Effective bit rate= 975 kbps,<br>G13 rx_boost=7  |     | -104.5 |     | dBm  |
| FLRC_1300_CR05_FERR  | Max frequency error tolerated | FRF = 915 MHz                                                                                    |     | +/-100 |     | kHz  |
| FLRC_1300_CR05_PRE   | Min preamble length           |                                                                                                  |     | 16     |     | bits |
| FLRC_1040_CR05_915_S | Sensitivity 1% PER 915 MHz    | BRF = 1040 kbps, CR = 3/4,<br>BWF = 1200 kHz<br>Effective bit rate= 780 kbps,<br>G13 rx_boost=7  |     | -105   |     | dBm  |
| FLRC_1040_CR05_FERR  | Max frequency error tolerated | FRF = 915 MHz                                                                                    |     | +/-100 |     | kHz  |
| FLRC_1040_CR05_PRE   | Min preamble length           |                                                                                                  |     | 16     |     | bits |
| FLRC_650_CR05_915_S  | Sensitivity 1% PER 915 MHz    | BRF = 650 kbps, CR = 3/4,<br>BWF = 600 kHz<br>Effective bit rate= 487 kbps,<br>G13 rx_boost=7    |     | -107   |     | dBm  |
| FLRC_650_CR05_FERR   | Max frequency error tolerated | FRF = 915 MHz                                                                                    |     | +/-70  |     | kHz  |
| FLRC_650_CR05_PRE    | Min preamble length           |                                                                                                  |     | 16     |     | bits |

**Table 3-12: FLRC Sub-GHz 1% PER (LR2021)<sup>1</sup> (Sheet 2 of 2)**

| Symbol              | Description                   | Conditions                                                                                  | Min | Typ    | Max | Unit |
|---------------------|-------------------------------|---------------------------------------------------------------------------------------------|-----|--------|-----|------|
| FLRC_520_CR05_915_S | Sensitivity 1% PER 915 MHz    | BRF = 520 kbps, CR = 3/4, BWF = 600 kHz<br>Effective bit rate= 390 kbps,<br>G13 rx_boost=7  |     | -108.5 |     | dBm  |
| FLRC_520_CR05_FERR  | Max frequency error tolerated | FRF = 915 MHz                                                                               |     | +/-50  |     | kHz  |
| FLRC_520_CR05_PRE   | Min preamble length           |                                                                                             |     | 16     |     | bits |
| FLRC_325_CR05_915_S | Sensitivity 1% PER 915 MHz    | BRF = 325 kbps, CR = 3/4, BWF = 300 kHz<br>Effective bit rate= 243 kbps,<br>G13 rx_boost=7  |     | -110   |     | dBm  |
| FLRC_325_CR05_FERR  | Max frequency error tolerated | FRF = 915 MHz                                                                               |     | +/-30  |     | kHz  |
| FLRC_325_CR05_PRE   | Min preamble length           |                                                                                             |     | 16     |     | bits |
| FLRC_260_CR05_915_S | Sensitivity 1% PER 915 MHz    | BRF = 260 kbps, CR = 3/4, BWF = 300 kHz,<br>Effective bit rate= 195 kbps,<br>G13 rx_boost=7 |     | -111   |     | dBm  |
| FLRC_260_CR05_FERR  | Max frequency error tolerated | FRF = 915 MHz                                                                               |     | +/-25  |     | kHz  |
| FLRC_260_CR05_PRE   | Min preamble length           |                                                                                             |     | 16     |     | bits |
| FLRC_CCR            | Co-channel rejection, FLRC    | BRF = 650 kbps, CR = 3/4                                                                    | -   | -2     | -   | dB   |

1. Using CR1/2 following the FLRP Physical Layer guidelines allows to improve receiver sensitivity.

The following table provides the receiver specification for FLRC @1% PER 31 bytes.

**Table 3-13: FLRC 2.4 GHz 1% PER (Packet Size = 31 bytes) (LR2021)<sup>1</sup> (Sheet 1 of 2)**

| Symbol               | Description                   | Conditions                                                                                    | Min    | Typ | Max | Unit |
|----------------------|-------------------------------|-----------------------------------------------------------------------------------------------|--------|-----|-----|------|
| FLRC_2600_CR05_2G4_S | Sensitivity 1% PER 2.4 GHz    | BRF = 2600 kbps, CR = 3/4, BWF = 2666 kHz<br>Effective bit rate= 1.95 Mbps,<br>G13 rx_boost=7 | -99    |     |     | dBm  |
| FLRC_2600_CR05_FERR  | Max frequency error tolerated | FRF = 2.4 GHz                                                                                 | +/-150 |     |     | kHz  |
| FLRC_2600_CR05_PRE   | Min preamble length           |                                                                                               | 24     |     |     | bits |
| FLRC_2080_CR05_2G4_S | Sensitivity 1% PER 2.4 GHz    | BRF = 2080 kbps, CR = 3/4, BWF = 2666 kHz<br>Effective bit rate= 1.56 Mbps,<br>G13 rx_boost=7 | -100   |     |     | dBm  |
| FLRC_2080_CR05_FERR  | Max frequency error tolerated | FRF = 2.4 GHz                                                                                 | +/-150 |     |     | kHz  |
| FLRC_2080_CR05_PRE   | Min preamble length           |                                                                                               | 20     |     |     | bits |
| FLRC_1300_CR05_2G4_S | Sensitivity 1% PER 2.4 GHz    | BRF = 1300 kbps, CR = 3/4, BWF = 1200 kHz<br>Effective bit rate= 975 kbps,<br>G13 rx_boost=7  | -101.5 |     |     | dBm  |
| FLRC_1300_CR05_FERR  | Max frequency error tolerated | FRF = 2.4 GHz                                                                                 | +/-100 |     |     | kHz  |
| FLRC_1300_CR05_PRE   | Min preamble length           |                                                                                               | 16     |     |     | bits |
| FLRC_1040_CR05_2G4_S | Sensitivity 1% PER 2.4 GHz    | BRF = 1040 kbps, CR = 3/4, BWF = 1200 kHz<br>Effective bit rate= 780 kbps,<br>G13 rx_boost=7  | -102.5 |     |     | dBm  |
| FLRC_1040_CR05_FERR  | Max freq error tolerated      | FRF = 2.4 GHz                                                                                 | +/-100 |     |     | kHz  |
| FLRC_1040_CR05_PRE   | Min preamble length           |                                                                                               | 16     |     |     | bits |
| FLRC_650_CR05_2G4_S  | Sensitivity 1% PER 2.4 GHz    | BRF = 650 kbps, CR = 3/4, BWF = 600 kHz<br>Effective bit rate= 487 kbps,<br>G13 rx_boost=7    | -104   |     |     | dBm  |
| FLRC_650_CR05_FERR   | Max freq error tolerated      | FRF = 2.4 GHz                                                                                 | +/-70  |     |     | kHz  |
| FLRC_650_CR05_PRE    | Min preamble length           |                                                                                               | 16     |     |     | bits |
| FLRC_520_CR05_2G4_S  | Sensitivity 1% PER 2.4 GHz    | BRF = 520 kbps, CR = 3/4, BWF = 600 kHz<br>Effective bit rate= 390 kbps,<br>G13 rx_boost=7    | -105   |     |     | dBm  |
| FLRC_520_CR05_FERR   | Max freq error tolerated      | FRF = 2.4 GHz                                                                                 | +/-50  |     |     | kHz  |
| FLRC_520_CR05_PRE    | Min preamble length           |                                                                                               | 16     |     |     | bits |

**Table 3-13: FLRC 2.4 GHz 1% PER (Packet Size = 31 bytes) (LR2021)<sup>1</sup> (Sheet 2 of 2)**

| Symbol              | Description                   | Conditions                                                                                 | Min | Typ    | Max | Unit |
|---------------------|-------------------------------|--------------------------------------------------------------------------------------------|-----|--------|-----|------|
| FLRC_325_CR05_2G4_S | Sensitivity 1% PER 2.4 GHz    | BRF = 325 kbps, CR = 3/4, BWF = 300 kHz<br>Effective bit rate= 243 kbps,<br>G13 rx_boost=7 |     | -108   |     | dBm  |
| FLRC_325_CR05_FERR  | Max frequency error tolerated | FRF = 2.4 GHz                                                                              |     | +/-30  |     | kHz  |
| FLRC_325_CR05_PRE   | Min preamble length           |                                                                                            |     | 16     |     | bits |
| FLRC_260_CR05_2G4_S | Sensitivity 1% PER 2.4 GHz    | BRF = 260 kbps, CR = 3/4, BWF = 300 kHz,<br>G13 rx_boost=7                                 |     | -108.5 |     | dBm  |
| FLRC_260_CR05_FERR  | Max frequency error tolerated | FRF = 2.4 GHz                                                                              |     | +/-25  |     | kHz  |
| FLRC_260_CR05_PRE   | Min preamble length           |                                                                                            |     | 16     |     | bits |
| FLRC_CCR            | Co-channel rejection, FLRC    | BRF = 650 kbps, CR = 3/4                                                                   | -   | -2     | -   | dB   |

1. Using CR1/2 following the FLRP Physical Layer guidelines allows to improve receiver sensitivity.

The following table provides the receiver specification for FSK Sensitivity at 0.1% BER (measured with a 13% Packet Error Rate (PER) test with packets with 32-bit Syncword, 10-Byte payload and 2-Byte CRC).

**Table 3-14: FSK Sensitivity @0.1% BER (LR20xx) (Sheet 1 of 2)**

| Symbol        | Description                     | Conditions                                                       | Min | Typ     | Max | Unit |
|---------------|---------------------------------|------------------------------------------------------------------|-----|---------|-----|------|
| FSK_250_S     | Sensitivity                     | 250 kbps 100 kHz Bw555, G13 rx_boost=7                           |     | -106    |     | dBm  |
| FSK_250_FERR  | Max frequency error tolerated   | FRF = 915 MHz                                                    |     | +/- 60  |     | ppm  |
| FSK_100_S     | Sensitivity                     | 100 kbps 50 kHz Bw333, G13 rx_boost=7                            |     | -110.5  |     | dBm  |
| FSK_100_FERR  | Max frequency error tolerated   | FRF = 915 MHz                                                    |     | +/- 60  |     | ppm  |
| FSK_50_S      | Sensitivity                     | 50 kbps 12.5 kHz Bw138, G13 rx_boost=7                           |     | -116    |     | dBm  |
| FSK_50_LW     | Sensitivity                     | 50 kbps 25 kHz Bw138, in LoRaWAN conditions, G13 rx_boost=7      |     | -113    |     | dBm  |
| FSK_50_FERR   | Max frequency error tolerated   | FRF = 915 MHz                                                    |     | +/- 60  |     | ppm  |
| FSK_38P4_S    | Sensitivity                     | 38.4 kbps 40 kHz Bw185, G13 rx_boost=7                           |     | -113    |     | dBm  |
| FSK_38P4_FERR | Max frequency error tolerated   | FRF = 915 MHz                                                    |     | +/- 60  |     | ppm  |
| FSK_4P8_S     | Sensitivity                     | 4.8 kbps 5 kHz Bw27, G13 rx_boost=7                              |     | -121.5  |     | dBm  |
| FSK_4P8_FERR  | Max frequency error tolerated   | FRF = 915 MHz                                                    |     | +/- 2.5 |     | kHz  |
| FSK_1P2_S     | Sensitivity                     | 1.2 kbps 5 kHz Bw17, G13 rx_boost=7                              |     | -124.5  |     | dBm  |
| RXSIG         | Sigfox sensitivity              | BRF= 0.6 kbps, FDA = 0.8 kHz, BWF = 4 kHz<br>G13 rx_boost=7      |     | -128.5  |     | dBm  |
| FSK_1P2_FERR  | Max frequency error tolerated   | FRF = 915 MHz                                                    |     | +/- 2.5 |     | kHz  |
| FSK_OP600_S   | Sensitivity                     | FRF = 868 MHz                                                    |     | -126    |     | dBm  |
| FSK_ACR       | Adjacent channel rejection, FSK | Offset = max of +/- 1.5 BW or +/- 50 kHz.<br>4.8 kbps 5 kHz Bw27 | -   | 50      | -   | dB   |
| FSK_BI1_3     | Blocking immunity               | Offset = +/- 1 MHz, 4.8 kbps 5 kHz Bw27,<br>sensitivity + 3 dB   | -   | 65      | -   | dB   |
| FSK_BI2_3     | Blocking immunity               | Offset = +/- 2 MHz, 4.8 kbps 5 kHz Bw27,<br>sensitivity + 3 dB   | -   | 70      | -   | dB   |
| FSK_BI10_3    | Blocking immunity               | Offset = +/- 10 MHz, 4.8 kbps 5 kHz Bw27,<br>sensitivity + 3 dB  | -   | 79      | -   | dB   |
| FSK_B1_10     | Blocking immunity               | Offset = +/- 1 MHz, 4.8 kbps 5 kHz Bw27,<br>sensitivity + 10 dB  | -   | 63      | -   | dB   |
| FSK_BI2_10    | Blocking immunity               | Offset = +/- 2 MHz, 4.8 kbps 5 kHz Bw27,<br>sensitivity + 10 dB  | -   | 65      | -   | dB   |
| FSK_BI10_10   | Blocking immunity               | Offset = +/- 10 MHz, 4.8 kbps 5 kHz Bw27,<br>sensitivity + 10 dB | -   | 80      | -   | dB   |
| FSK_B1_30     | Blocking immunity               | Offset = +/- 1 MHz, 4.8 kbps 5 kHz Bw27,<br>sensitivity + 30 dB  | -   | 62      | -   | dB   |

Table 3-14: FSK Sensitivity @0.1% BER (LR20xx) (Sheet 2 of 2)

| Symbol      | Description       | Conditions                                                    | Min | Typ | Max | Unit |
|-------------|-------------------|---------------------------------------------------------------|-----|-----|-----|------|
| FSK_BI2_30  | Blocking immunity | Offset = +/- 2 MHz, 4.8 kbps 5 kHz Bw27, sensitivity + 30 dB  | -   | 62  | -   | dB   |
| FSK_BI10_30 | Blocking immunity | Offset = +/- 10 MHz, 4.8 kbps 5 kHz Bw27, sensitivity + 30 dB | -   | 72  | -   | dB   |
| FSK_B1_50   | Blocking immunity | Offset = +/- 1 MHz, 4.8 kbps 5 kHz Bw27, sensitivity + 50 dB  | -   | 65  | -   | dB   |
| FSK_BI2_50  | Blocking immunity | Offset = +/- 2 MHz, 4.8 kbps 5 kHz Bw27, sensitivity + 50 dB  | -   | 65  | -   | dB   |
| FSK_BI10_50 | Blocking immunity | Offset = +/- 10 MHz, 4.8 kbps 5 kHz Bw27, sensitivity + 50 dB | -   | 75  | -   | dB   |



The following table provides the receiver specification for Bluetooth LE Physical Layer @30.8% PER with 20 byte payloads.

**Table 3-15: Bluetooth LE Physical Layer (LR2021/LR2022) (Sheet 1 of 2)**

| Symbol           | Description                                              | Conditions                                 | Min | Typ  | Max | Unit |
|------------------|----------------------------------------------------------|--------------------------------------------|-----|------|-----|------|
| BLE_2M_S         | 30.8% PER Sensitivity Bluetooth LE 2 Mbps                | G13 rx_boost = 7                           |     | -97  |     | dBm  |
| BLE_2M_CCR       | Co-channel interference Bluetooth LE 2 Mbps              | Wanted signal at -67 dBm                   |     | 5    |     | dB   |
| BLE_2M_CI_P2M    | Adjacent interference 2 MHz                              | Wanted signal at -67 dBm                   |     | -7   |     | dB   |
| BLE_2M_CI_M2M    | Adjacent interference -2 MHz                             | Wanted signal at -67 dBm                   |     | -7   |     | dB   |
| BLE_2M_CI_P4M    | Adjacent interference 4 MHz                              | Wanted signal at -67 dBm                   |     | -37  |     | dB   |
| BLE_2M_CI_M4M    | Adjacent interference -4 MHz                             | Wanted signal at -67 dBm                   |     | -35  |     | dB   |
| BLE_2M_CI_S6M    | Adjacent interference >= 6 MHz                           | Wanted signal at -67 dBm                   |     | -51  |     | dB   |
| BLE_2M_CI_S6M    | Adjacent interference <= -6 MHz                          | Wanted signal at -67 dBm                   |     | -35  |     | dB   |
| BLE_2M_IMG       | Image frequency interference                             | Wanted signal at -67 dBm, image @-5.33 MHz |     | -34  |     | dB   |
| BLE_2M_IMG_ADJ_P | Adjacent interference to in-band image frequency + 2 MHz | Wanted signal at -67 dBm                   |     | -33  |     | dB   |
| BLE_2M_IMG_ADJ_M | Adjacent interference to in-band image frequency -2 MHz  | Wanted signal at -67 dBm                   |     | -45  |     | dB   |
| BLE_1M_S         | 30.8% PER Sensitivity Bluetooth LE 1 Mbps                | G13 rx_boost = 7                           |     | -100 |     | dBm  |
| BLE_1M_CCR       | Co-channel interference Bluetooth LE 1 Mbps              | Wanted signal at -67 dBm                   |     | 5    |     | dB   |
| BLE_1M_CI_P1M    | Adjacent interference 1 MHz                              | Wanted signal at -67 dBm                   |     | -7   |     | dB   |
| BLE_1M_CI_M1M    | Adjacent interference -1 MHz                             | Wanted signal at -67 dBm                   |     | -7   |     | dB   |
| BLE_1M_CI_P2M    | Adjacent interference 2 MHz                              | Wanted signal at -67 dBm                   |     | -35  |     | dB   |
| BLE_1M_CI_M2M    | Adjacent interference -2 MHz                             | Wanted signal at -67 dBm                   |     | -35  |     | dB   |
| BLE_1M_CI_P3M    | Adjacent interference 3 MHz                              | Wanted signal at -67 dBm                   |     | -36  |     | dB   |
| BLE_1M_CI_M3M    | Adjacent interference -3 MHz                             | Wanted signal at -67 dBm                   |     | -34  |     | dB   |
| BLE_1M_IMG       | Image frequency interference                             | Wanted signal at -67 dBm, image @-2.66 MHz |     | -34  |     | dB   |
| BLE_1M_IMG_ADJ_P | Adjacent interference to in-band image frequency +1 MHz  | Wanted signal at -67 dBm                   |     | -35  |     | dB   |
| BLE_1M_IMG_ADJ_M | Adjacent interference to in-band image frequency -1 MHz  | Wanted signal at -67 dBm                   |     | -41  |     | dB   |
| BLE_500K_S       | 30.8% PER Sensitivity Bluetooth LE 500 kbps              | G13 rx_boost = 7                           |     | -103 |     | dBm  |

Table 3-15: Bluetooth LE Physical Layer (LR2021/LR2022) (Sheet 2 of 2)

| Symbol             | Description                                              | Conditions                                 | Min | Typ     | Max | Unit |
|--------------------|----------------------------------------------------------|--------------------------------------------|-----|---------|-----|------|
| BLE_500K_CCR       | Co-channel interference Bluetooth LE 1 Mbps              | Wanted signal at -72 dBm                   |     | 4       |     | dB   |
| BLE_500K_CI_P1M    | Adjacent interference 1 MHz                              | Wanted signal at -72 dBm                   |     | -8      |     | dB   |
| BLE_500K_CI_M1M    | Adjacent interference -1 MHz                             | Wanted signal at -72 dBm                   |     | -8      |     | dB   |
| BLE_500K_CI_P2M    | Adjacent interference 2 MHz                              | Wanted signal at -72 dBm                   |     | -50     |     | dB   |
| BLE_500K_CI_M2M    | Adjacent interference -2 MHz                             | Wanted signal at -72 dBm                   |     | -51     |     | dB   |
| BLE_500K_CI_P3M    | Adjacent interference 3 MHz                              | Wanted signal at -72 dBm                   |     | -55     |     | dB   |
| BLE_500K_CI_M3M    | Adjacent interference -3 MHz                             | Wanted signal at -72 dBm                   |     | -37     |     | dB   |
| BLE_500K_IMG       | Image frequency interference                             | Wanted signal at -72 dBm, image @ 2.66 MHz |     | -37     |     | dB   |
| BLE_500K_IMG_ADJ_P | Adjacent interference to in-band image frequency + 1 MHz | Wanted signal at -72 dBm                   |     | -41     |     | dB   |
| BLE_500K_IMG_ADJ_M | Adjacent interference to in-band image frequency - 1 MHz | Wanted signal at -72 dBm                   |     | -48     |     | dB   |
| BLE_125K_S         | 30.8% PER Sensitivity Bluetooth LE 125 kbps              | G13 rx_boost = 7                           |     | -106.5  |     | dBm  |
| BLE_125K_CCR       | Co-channel interference Bluetooth LE 1 Mbps              | Wanted signal at -79 dBm                   |     | 4       |     | dB   |
| BLE_125K_CI_P1M    | Adjacent interference 1 MHz                              | Wanted signal at -79 dBm                   |     | -18     |     | dB   |
| BLE_125K_CI_M1M    | Adjacent interference -1 MHz                             | Wanted signal at -79 dBm                   |     | -18     |     | dB   |
| BLE_125K_CI_P2M    | Adjacent interference 2 MHz                              | Wanted signal at -79 dBm                   |     | -55     |     | dB   |
| BLE_125K_CI_M2M    | Adjacent interference -2 MHz                             | Wanted signal at -79 dBm                   |     | -43     |     | dB   |
| BLE_125K_CI_3M     | Adjacent interference 3 MHz                              | Wanted signal at -79 dBm                   |     | -47     |     | dB   |
| BLE_125K_CI_M3M    | Adjacent interference <= -3 MHz                          | Wanted signal at -79 dBm                   |     | -38     |     | dB   |
| BLE_125K_IMG       | Image frequency interference                             | Wanted signal at -79 dBm, image @ 2.66 MHz |     | -38     |     | dB   |
| BLE_125K_IMG_ADJ_P | Adjacent interference to in-band image frequency + 1 MHz | Wanted signal at -72 dBm                   |     | -43     |     | dB   |
| BLE_125K_IMG_ADJ_M | Adjacent interference to in-band image frequency - 1 MHz | Wanted signal at -72 dBm                   |     | -55     |     | dB   |
| BLE_FREQERR        | Maximum frequency error                                  | Wanted signal at -79 dBm                   |     | +/- 150 |     | kHz  |

The following table provides the receiver specification for OQPSK\_15\_4 @1% PER with 20 byte payloads. G13 rx\_boost=7 for sensitivity measurements.

**Table 3-16: OQPSK\_15\_4 (LR2021)**

| Symbol            | Description                         | Conditions                                | Min | Typ    | Max | Unit |
|-------------------|-------------------------------------|-------------------------------------------|-----|--------|-----|------|
| OQPSK_250_S       | RF@2.4 GHz, 250 kbps                |                                           |     | -103   |     | dBm  |
| OQPSK_250_ACR     | Adjacent interference               | Signal @sensitivity + 3 dB,<br>RF@2.4 GHz |     | 27     |     | dB   |
| OQPSK_250_ALT     | Alternate Adjacent interference     | Signal @sensitivity + 3 dB,<br>RF@2.4 GHz |     | 45     |     | dB   |
| OQPSK_250_FREQERR | Maximum frequency error tolerated   |                                           |     | +/- 40 |     | ppm  |
| OQPSK_250_SYMBERR | Maximum Symbol rate error tolerated |                                           |     | +/- 40 |     | ppm  |

The following table provides the receiver specification for LoRa sub-GHz 64B payload @1% PER. Sensitivities are provided for G13 rx\_boost=7 unless otherwise specified.

**Table 3-17: LoRa Sub-GHz 64B Payload (LR20xx) (Sheet 1 of 5)**

| Symbol             | Description      | Conditions                         | Min | Typ    | Max | Unit |
|--------------------|------------------|------------------------------------|-----|--------|-----|------|
| LORA_SUB_1000_SF5  | Sensitivity LoRa | BWLORA = 1000 kHz, SF = 5, CR 4/5  | -   | -112.5 | -   | dBm  |
| LORA_SUB_1000_SF6  |                  | BWLORA = 1000 kHz, SF = 6, CR 4/5  | -   | -115.5 | -   | dBm  |
| LORA_SUB_1000_SF7  |                  | BWLORA = 1000 kHz, SF = 7, CR 4/5  | -   | -118   | -   | dBm  |
| LORA_SUB_1000_SF8  |                  | BWLORA = 1000 kHz, SF = 8, CR 4/5  | -   | -120   | -   | dBm  |
| LORA_SUB_1000_SF9  |                  | BWLORA = 1000 kHz, SF = 9, CR 4/5  | -   | -123   | -   | dBm  |
| LORA_SUB_1000_SF10 |                  | BWLORA = 1000 kHz, SF = 10, CR 4/5 | -   | -126   | -   | dBm  |
| LORA_SUB_1000_SF11 |                  | BWLORA = 1000 kHz, SF = 11, CR 4/5 | -   | -129   | -   | dBm  |
| LORA_SUB_1000_SF12 |                  | BWLORA = 1000 kHz, SF = 12, CR 4/5 | -   | -131   | -   | dBm  |
| LORA_SUB_800_SF5   | Sensitivity LoRa | BWLORA = 800 kHz, SF = 5, CR 4/5   | -   | -113   | -   | dBm  |
| LORA_SUB_800_SF6   |                  | BWLORA = 800 kHz, SF = 6, CR 4/5   | -   | -115.5 | -   | dBm  |
| LORA_SUB_800_SF7   |                  | BWLORA = 800 kHz, SF = 7, CR 4/5   | -   | -118   | -   | dBm  |
| LORA_SUB_800_SF8   |                  | BWLORA = 800 kHz, SF = 8, CR 4/5   | -   | -121   | -   | dBm  |
| LORA_SUB_800_SF9   |                  | BWLORA = 800 kHz, SF = 9, CR 4/5   | -   | -124   | -   | dBm  |
| LORA_SUB_800_SF10  |                  | BWLORA = 800 kHz, SF = 10, CR 4/5  | -   | -126.5 | -   | dBm  |
| LORA_SUB_800_SF11  |                  | BWLORA = 800 kHz, SF = 11, CR 4/5  | -   | -130   | -   | dBm  |
| LORA_SUB_800_SF12  |                  | BWLORA = 800 kHz, SF = 12, CR 4/5  | -   | -132   | -   | dBm  |

**Table 3-17: LoRa Sub-GHz 64B Payload (LR20xx) (Sheet 2 of 5)**

| Symbol            | Description      | Conditions                        | Min | Typ    | Max | Unit |
|-------------------|------------------|-----------------------------------|-----|--------|-----|------|
| LORA_SUB_500_SF5  | Sensitivity LoRa | BWLORA = 500 kHz, SF = 5, CR 4/5  | -   | -116   | -   | dBm  |
| LORA_SUB_500_SF6  |                  | BWLORA = 500 kHz, SF = 6, CR 4/5  | -   | -119   | -   | dBm  |
| LORA_SUB_500_SF7  |                  | BWLORA = 500 kHz, SF = 7, CR 4/5  | -   | -122   | -   | dBm  |
| LORA_SUB_500_SF8  |                  | BWLORA = 500 kHz, SF = 8, CR 4/5  | -   | -124.5 | -   | dBm  |
| LORA_SUB_500_SF9  |                  | BWLORA = 500 kHz, SF = 9, CR 4/5  | -   | -127.5 | -   | dBm  |
| LORA_SUB_500_SF10 |                  | BWLORA = 500 kHz, SF = 10, CR 4/5 | -   | -130   | -   | dBm  |
| LORA_SUB_500_SF11 |                  | BWLORA = 500 kHz, SF = 11, CR 4/5 | -   | -133   | -   | dBm  |
| LORA_SUB_500_SF12 |                  | BWLORA = 500 kHz, SF = 12, CR 4/5 | -   | -135.5 | -   | dBm  |
| LORA_SUB_400_SF5  | Sensitivity LoRa | BWLORA = 400 kHz, SF = 5, CR 4/5  | -   | -117   | -   | dBm  |
| LORA_SUB_400_SF6  |                  | BWLORA = 400 kHz, SF = 6, CR 4/5  | -   | -120   | -   | dBm  |
| LORA_SUB_400_SF7  |                  | BWLORA = 400 kHz, SF = 7, CR 4/5  | -   | -122.5 | -   | dBm  |
| LORA_SUB_400_SF8  |                  | BWLORA = 400 kHz, SF = 8, CR 4/5  | -   | -125.5 | -   | dBm  |
| LORA_SUB_400_SF9  |                  | BWLORA = 400 kHz, SF = 9, CR 4/5  | -   | -128.5 | -   | dBm  |
| LORA_SUB_400_SF10 |                  | BWLORA = 400 kHz, SF = 10, CR 4/5 | -   | -130.5 | -   | dBm  |
| LORA_SUB_400_SF11 |                  | BWLORA = 400 kHz, SF = 11, CR 4/5 | -   | -134   | -   | dBm  |
| LORA_SUB_400_SF12 |                  | BWLORA = 400 kHz, SF = 12, CR 4/5 | -   | -136.5 | -   | dBm  |
| LORA_SUB_250_SF5  | Sensitivity LoRa | BWLORA = 250 kHz, SF = 5, CR 4/5  | -   | -119   | -   | dBm  |
| LORA_SUB_250_SF6  |                  | BWLORA = 250 kHz, SF = 6, CR 4/5  | -   | -122   | -   | dBm  |
| LORA_SUB_250_SF7  |                  | BWLORA = 250 kHz, SF = 7, CR 4/5  | -   | -124.5 | -   | dBm  |
| LORA_SUB_250_SF8  |                  | BWLORA = 250 kHz, SF = 8, CR 4/5  | -   | -127.5 | -   | dBm  |
| LORA_SUB_250_SF9  |                  | BWLORA = 250 kHz, SF = 9, CR 4/5  | -   | -130   | -   | dBm  |
| LORA_SUB_250_SF10 |                  | BWLORA = 250 kHz, SF = 10, CR 4/5 | -   | -133   | -   | dBm  |
| LORA_SUB_250_SF11 |                  | BWLORA = 250 kHz, SF = 11, CR 4/5 | -   | -135.5 | -   | dBm  |
| LORA_SUB_250_SF12 |                  | BWLORA = 250 kHz, SF = 12, CR 4/5 | -   | -138.5 | -   | dBm  |

**Table 3-17: LoRa Sub-GHz 64B Payload (LR20xx) (Sheet 3 of 5)**

| Symbol            | Description      | Conditions                        | Min | Typ    | Max | Unit |
|-------------------|------------------|-----------------------------------|-----|--------|-----|------|
| LORA_SUB_200_SF5  | Sensitivity LoRa | BWLORA = 200 kHz, SF = 5, CR 4/5  | -   | -119   | -   | dBm  |
| LORA_SUB_200_SF6  |                  | BWLORA = 200 kHz, SF = 6, CR 4/5  | -   | -122.5 | -   | dBm  |
| LORA_SUB_200_SF7  |                  | BWLORA = 200 kHz, SF = 7, CR 4/5  | -   | -125   | -   | dBm  |
| LORA_SUB_200_SF8  |                  | BWLORA = 200 kHz, SF = 8, CR 4/5  | -   | -128   | -   | dBm  |
| LORA_SUB_200_SF9  |                  | BWLORA = 200 kHz, SF = 9, CR 4/5  | -   | -131   | -   | dBm  |
| LORA_SUB_200_SF10 |                  | BWLORA = 200 kHz, SF = 10, CR 4/5 | -   | -133.5 | -   | dBm  |
| LORA_SUB_200_SF11 |                  | BWLORA = 200 kHz, SF = 11, CR 4/5 | -   | -136   | -   | dBm  |
| LORA_SUB_200_SF12 |                  | BWLORA = 200 kHz, SF = 12, CR 4/5 | -   | -139   | -   | dBm  |
| LORA_SUB_125_SF5  | Sensitivity LoRa | BWLORA = 125 kHz, SF = 5, CR 4/5  | -   | -122   | -   | dBm  |
| LORA_SUB_125_SF6  |                  | BWLORA = 125 kHz, SF = 6, CR 4/5  | -   | -125   | -   | dBm  |
| LORA_SUB_125_SF7  |                  | BWLORA = 125 kHz, SF = 7, CR 4/5  | -   | -127.5 | -   | dBm  |
| LORA_SUB_125_SF8  |                  | BWLORA = 125 kHz, SF = 8, CR 4/5  | -   | -130.5 | -   | dBm  |
| LORA_SUB_125_SF9  |                  | BWLORA = 125 kHz, SF = 9, CR 4/5  | -   | -133   | -   | dBm  |
| LORA_SUB_125_SF10 |                  | BWLORA = 125 kHz, SF = 10, CR 4/5 | -   | -136   | -   | dBm  |
| LORA_SUB_125_SF11 |                  | BWLORA = 125 kHz, SF = 11, CR 4/5 | -   | -138.5 | -   | dBm  |
| LORA_SUB_125_SF12 |                  | BWLORA = 125 kHz, SF = 12, CR 4/5 | -   | -141.5 | -   | dBm  |
| LORA_SUB_62_SF5   | Sensitivity LoRa | BWLORA = 62 kHz, SF = 5, CR 4/5   | -   | -124.5 | -   | dBm  |
| LORA_SUB_62_SF6   |                  | BWLORA = 62 kHz, SF = 6, CR 4/5   | -   | -126.5 | -   | dBm  |
| LORA_SUB_62_SF7   |                  | BWLORA = 62 kHz, SF = 7, CR 4/5   | -   | -129   | -   | dBm  |
| LORA_SUB_62_SF8   |                  | BWLORA = 62 kHz, SF = 8, CR 4/5   | -   | -132   | -   | dBm  |
| LORA_SUB_62_SF9   |                  | BWLORA = 62 kHz, SF = 9, CR 4/5   | -   | -135   | -   | dBm  |
| LORA_SUB_62_SF10  |                  | BWLORA = 62 kHz, SF = 10, CR 4/5  | -   | -137.5 | -   | dBm  |
| LORA_SUB_62_SF11  |                  | BWLORA = 62 kHz, SF = 11, CR 4/5  | -   | -140.5 | -   | dBm  |
| LORA_SUB_62_SF12  |                  | BWLORA = 62 kHz, SF = 12, CR 4/5  | -   | -143   | -   | dBm  |

Table 3-17: LoRa Sub-GHz 64B Payload (LR20xx) (Sheet 4 of 5)

| Symbol           | Description                                                                                           | Conditions                                      | Min | Typ    | Max | Unit |
|------------------|-------------------------------------------------------------------------------------------------------|-------------------------------------------------|-----|--------|-----|------|
| LORA_SUB_31_SF5  | Sensitivity LoRa                                                                                      | BWLORA = 31 kHz, SF = 5, CR 4/5                 | -   | -128   | -   | dBm  |
| LORA_SUB_31_SF6  |                                                                                                       | BWLORA = 31 kHz, SF = 6, CR 4/5                 | -   | -130.5 | -   | dBm  |
| LORA_SUB_31_SF7  |                                                                                                       | BWLORA = 31 kHz, SF = 7, CR 4/5                 | -   | -133   | -   | dBm  |
| LORA_SUB_31_SF8  |                                                                                                       | BWLORA = 31 kHz, SF = 8, CR 4/5                 | -   | -135.5 | -   | dBm  |
| LORA_SUB_31_SF9  |                                                                                                       | BWLORA = 31 kHz, SF = 9, CR 4/5                 | -   | -138   | -   | dBm  |
| LORA_SUB_31_SF10 |                                                                                                       | BWLORA = 31 kHz, SF = 10, CR 4/5                | -   | -142   | -   | dBm  |
| LORA_SUB_31_SF11 |                                                                                                       | BWLORA = 31 kHz, SF = 11, CR 4/5                | -   | -144.5 | -   | dBm  |
| LORA_SUB_31_SF12 |                                                                                                       | BWLORA = 31 kHz, SF = 12, CR 4/5                | -   | -147   | -   | dBm  |
| LORA_CCR_6       | Co-channel rejection, LoRa                                                                            | CW in band SF = 7, CR=4/5                       | -   | 7.25   | -   | dB   |
| LORA_CCR_7       |                                                                                                       | CW in band SF = 12, CR=4/5                      | -   | 22.5   | -   | dB   |
| LORA_ACR7        | Adjacent channel rejection, LoRa                                                                      | BWLORA = 125 kHz, SF = 7                        | -   | 61     | -   | dB   |
| LORA_ACR12       | Offset = +/- 1.5 x BWLORA                                                                             | BWLORA = 125 kHz, SF = 12                       | -   | 75     | -   | dB   |
| LORA_BI_SF12_1M  | Blocking immunity, LoRa, BWLORA = 125 kHz                                                             | SF =12, offset = +/- 1 MHz                      | -   | 79     | -   | dB   |
| LORA_BI_SF12_2M  |                                                                                                       | SF =12, offset = +/- 2 MHz                      | -   | 89     | -   | dB   |
| LORA_BI_SF12_10M |                                                                                                       | SF =12, offset = +/- 10 MHz                     | -   | 100    | -   | dB   |
| LORA_BI_SF7_1M   |                                                                                                       | SF =7, offset = +/- 1 MHz                       | -   | 64     | -   | dB   |
| LORA_BI_SF7_2M   |                                                                                                       | SF =7, offset = +/- 2 MHz                       | -   | 76     | -   | dB   |
| LORA_BI_SF7_10M  |                                                                                                       | SF =7, offset = +/- 10 MHz                      | -   | 86     | -   | dB   |
| LORA_FDRIFT1     | LoRa frequency drift tolerance<br>For sensitivity degradation below 2 dB                              | BWLORA = 125 kHz, SF10, LowDataRateOptimize = 0 | -   | -      | 200 | Hz/s |
| LORA_FDRIFT2     | LoRa frequency drift tolerance                                                                        | BWLORA = 125 kHz, SF12, LowDataRateOptimize = 1 | -   | -      | 100 | Hz/s |
| LORA_FERR_L1     | LoRa Maximum tolerated frequency offset between transmitter and receiver                              | All bandwidths                                  | -25 |        | 25  | %BW  |
| LORA_FERR_L2     | LoRa Maximum tolerated frequency offset between transmitter and receiver for at most 3 dB degradation | All bandwidths                                  | -33 |        | 33  | %BW  |

**Table 3-17: LoRa Sub-GHz 64B Payload (LR20xx) (Sheet 5 of 5)**

| Symbol       | Description                                                                                                                                                                             | Conditions                                                                     | Min  | Typ | Max | Unit  |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------|------|-----|-----|-------|
| LORA_FERR_L4 | LoRa Maximum tolerated frequency offset between transmitter and receiver for at most 1.5 dB degradation.<br>The tighter limit between this specification and FERR_L1 or FERR_L2 applies | SF12                                                                           | -100 | -   | 100 | ppm   |
| LORA_FERR_L5 | LoRa Maximum tolerated frequency offset between transmitter and receiver for at most 1.5 dB degradation                                                                                 | SF11                                                                           | -200 | -   | 200 | ppm   |
| LORA_PL_LR1  | Payload length range                                                                                                                                                                    | In all combinations of BW, SF and CR except Long interleaving with CRC enabled | 1    |     | 255 | Bytes |
| LORA_PL_LR2  |                                                                                                                                                                                         | Long interleaving and CRC enabled                                              | 1    |     | 253 | Bytes |

The following table provides the receiver specification for LoRa 2.4 GHz 64B payload @1% PER 20 Bytes. Sensitivities are provided for G13 rx\_boost=7 unless otherwise specified.

**Table 3-18: LoRa 2.4 GHz 64B Payload (LR2021/LR2022) (Sheet 1 of 2)**

| Symbol             | Description      | Conditions                         | Min | Typ    | Max | Unit |
|--------------------|------------------|------------------------------------|-----|--------|-----|------|
| LORA_2G4_1000_SF5  | Sensitivity LoRa | BWLORA = 1000 kHz, SF = 5, CR 4/5  | -   | -110   | -   | dBm  |
| LORA_2G4_1000_SF6  |                  | BWLORA = 1000 kHz, SF = 6, CR 4/5  | -   | -113   | -   | dBm  |
| LORA_2G4_1000_SF7  |                  | BWLORA = 1000 kHz, SF = 7, CR 4/5  | -   | -115.5 | -   | dBm  |
| LORA_2G4_1000_SF8  |                  | BWLORA = 1000 kHz, SF = 8, CR 4/5  | -   | -118.5 | -   | dBm  |
| LORA_2G4_1000_SF9  |                  | BWLORA = 1000 kHz, SF = 9, CR 4/5  | -   | -121.5 | -   | dBm  |
| LORA_2G4_1000_SF10 |                  | BWLORA = 1000 kHz, SF = 10, CR 4/5 | -   | -123.5 | -   | dBm  |
| LORA_2G4_1000_SF11 |                  | BWLORA = 1000 kHz, SF = 11, CR 4/5 | -   | -126.5 | -   | dBm  |
| LORA_2G4_1000_SF12 |                  | BWLORA = 1000 kHz, SF = 12, CR 4/5 | -   | -129.5 | -   | dBm  |
| LORA_2G4_800_SF5   | Sensitivity LoRa | BWLORA = 800 kHz, SF = 5, CR 4/5   | -   | -111   | -   | dBm  |
| LORA_2G4_800_SF6   |                  | BWLORA = 800 kHz, SF = 6, CR 4/5   | -   | -114   | -   | dBm  |
| LORA_2G4_800_SF7   |                  | BWLORA = 800 kHz, SF = 7, CR 4/5   | -   | -116.5 | -   | dBm  |
| LORA_2G4_800_SF8   |                  | BWLORA = 800 kHz, SF = 8, CR 4/5   | -   | -119.5 | -   | dBm  |
| LORA_2G4_800_SF9   |                  | BWLORA = 800 kHz, SF = 9, CR 4/5   | -   | -122.5 | -   | dBm  |
| LORA_2G4_800_SF10  |                  | BWLORA = 800 kHz, SF = 10, CR 4/5  | -   | -125   | -   | dBm  |
| LORA_2G4_800_SF11  |                  | BWLORA = 800 kHz, SF = 11, CR 4/5  | -   | -128   | -   | dBm  |
| LORA_2G4_800_SF12  |                  | BWLORA = 800 kHz, SF = 12, CR 4/5  | -   | -130.5 | -   | dBm  |
| LORA_2G4_500_SF5   | Sensitivity LoRa | BWLORA = 500 kHz, SF = 5, CR 4/5   | -   | -112.5 | -   | dBm  |
| LORA_2G4_500_SF6   |                  | BWLORA = 500 kHz, SF = 6, CR 4/5   | -   | -116   | -   | dBm  |
| LORA_2G4_500_SF7   |                  | BWLORA = 500 kHz, SF = 7, CR 4/5   | -   | -118.5 | -   | dBm  |
| LORA_2G4_500_SF8   |                  | BWLORA = 500 kHz, SF = 8, CR 4/5   | -   | -121.5 | -   | dBm  |
| LORA_2G4_500_SF9   |                  | BWLORA = 500 kHz, SF = 9, CR 4/5   | -   | -124   | -   | dBm  |
| LORA_2G4_500_SF10  |                  | BWLORA = 500 kHz, SF = 10, CR 4/5  | -   | -127   | -   | dBm  |
| LORA_2G4_500_SF11  |                  | BWLORA = 500 kHz, SF = 11, CR 4/5  | -   | -129.5 | -   | dBm  |
| LORA_2G4_500_SF12  |                  | BWLORA = 500 kHz, SF = 12, CR 4/5  | -   | -132.5 | -   | dBm  |
| LORA_2G4_400_SF5   | Sensitivity LoRa | BWLORA = 400 kHz, SF = 5, CR 4/5   | -   | -114   | -   | dBm  |
| LORA_2G4_400_SF6   |                  | BWLORA = 400 kHz, SF = 6, CR 4/5   | -   | -117   | -   | dBm  |
| LORA_2G4_400_SF7   |                  | BWLORA = 400 kHz, SF = 7, CR 4/5   | -   | -119.5 | -   | dBm  |
| LORA_2G4_400_SF8   |                  | BWLORA = 400 kHz, SF = 8, CR 4/5   | -   | -123   | -   | dBm  |
| LORA_2G4_400_SF9   |                  | BWLORA = 400 kHz, SF = 9, CR 4/5   | -   | -125.5 | -   | dBm  |
| LORA_2G4_400_SF10  |                  | BWLORA = 400 kHz, SF = 10, CR 4/5  | -   | -128   | -   | dBm  |
| LORA_2G4_400_SF11  |                  | BWLORA = 400 kHz, SF = 11, CR 4/5  | -   | -131   | -   | dBm  |
| LORA_2G4_400_SF12  |                  | BWLORA = 400 kHz, SF = 12, CR 4/5  | -   | -134   | -   | dBm  |



**Table 3-18: LoRa 2.4 GHz 64B Payload (LR2021/LR2022) (Sheet 2 of 2)**

| Symbol            | Description                                    | Conditions                                             | Min | Typ    | Max | Unit |
|-------------------|------------------------------------------------|--------------------------------------------------------|-----|--------|-----|------|
| LORA_2G4_200_SF5  | Sensitivity LoRa                               | BWLORA = 200 kHz, SF = 5, CR 4/5                       | -   | -117   | -   | dBm  |
| LORA_2G4_200_SF6  |                                                | BWLORA = 200 kHz, SF = 6, CR 4/5                       | -   | -120   | -   | dBm  |
| LORA_2G4_200_SF7  |                                                | BWLORA = 200 kHz, SF = 7, CR 4/5                       | -   | -123   | -   | dBm  |
| LORA_2G4_200_SF8  |                                                | BWLORA = 200 kHz, SF = 8, CR 4/5                       | -   | -125.5 | -   | dBm  |
| LORA_2G4_200_SF9  |                                                | BWLORA = 200 kHz, SF = 9, CR 4/5                       | -   | -128.5 | -   | dBm  |
| LORA_2G4_200_SF10 |                                                | BWLORA = 200 kHz, SF = 10, CR 4/5                      | -   | -131   | -   | dBm  |
| LORA_2G4_200_SF11 |                                                | BWLORA = 200 kHz, SF = 11, CR 4/5                      | -   | -134   | -   | dBm  |
| LORA_2G4_200_SF12 |                                                | BWLORA = 200 kHz, SF = 12, CR 4/5                      | -   | -137   | -   | dBm  |
| LORA_2G4_CCR_6    | Co-channel rejection, LoRa                     | CW in band SF = 7, CR=4/5                              | -   | 7.25   | -   | dB   |
| LORA_2G4_CCR_7    |                                                | CW in band SF = 12, CR=4/5                             | -   | 22.5   | -   | dB   |
| LORA_2G4_ACR7     | Adjacent channel rejection, LoRa               | BWLORA = 800 kHz, SF = 7                               | -   | 53     | -   | dB   |
| LORA_2G4_ACR12    |                                                | Offset = +/- 1.5 x BWLORA<br>BWLORA = 800 kHz, SF = 12 | -   | 68     | -   | dB   |
| LORA_2G4_BI12_2M  | Blocking immunity<br>LoRa, Offset = +/- 2 MHz  | BWL = 802 kHz, SF12                                    | -   | 68     | -   | dB   |
| LORA_2G4_BI12_10M | Blocking immunity<br>LoRa, Offset = +/- 10 MHz | BWL = 802 kHz, SF12                                    | -   | 82     | -   | dB   |
| LORA_2G4_BI7_2M   | Blocking immunity<br>LoRa, Offset = +/- 2 MHz  | BWL = 802 kHz, SF7                                     | -   | 54     | -   | dB   |
| LORA_2G4_BI7_10M  | Blocking immunity<br>LoRa, Offset = +/- 10 MHz | BWL = 802 kHz, SF7                                     | -   | 67     | -   | dB   |

The following table provides the receiver specification for OOK @1% PER 20 Bytes.

**Table 3-19: OOK 1% PER (LR20xx)**

| Symbol        | Description                                                                                                   | Conditions                     | Min | Typ  | Max | Unit |
|---------------|---------------------------------------------------------------------------------------------------------------|--------------------------------|-----|------|-----|------|
| OOK_S2        | Sensitivity OOK                                                                                               | BR = 4.8 kbps, BW = 24 kHz     | -   | -114 | -   | dBm  |
| OOK_S3        |                                                                                                               | BR = 32.768 kbps, BW = 153 kHz | -   | -105 | -   | dBm  |
| OOK_S1        |                                                                                                               | BR = 1.562 kbps, BW = 7.4 kHz  | -   | -120 | -   | dBm  |
| OOK_FREQ_RG_1 | Max tolerated frequency offset between transmitter and receiver for 2 dB sensitivity degradation <sup>1</sup> | BR = 1.562 kbps, BW = 7.4 kHz  | -26 |      | 26  | kHz  |

1. For extended frequency error tolerance contact your Semtech representative.

**Table 3-20: General Receiver Specifications (LR20xx unless specified otherwise)**

| Symbol      | Description                                       | Conditions                                                                 | Min  | Typ | Max  | Unit |
|-------------|---------------------------------------------------|----------------------------------------------------------------------------|------|-----|------|------|
| BW          | Receiver bandwidth                                |                                                                            | 3.5  | -   | 3076 | kHz  |
| RX_HF_FREQR | High frequency path (HF) Rx input frequency range | (LR2021/LR2022) On pin RFI_HF                                              | 1500 | -   | 2600 | MHz  |
| RX_LF_FREQR | Low Frequency (LF) Rx input frequency range       | On pin RFI_LF                                                              | 150  | -   | 960  | MHz  |
| LF_N_IIP3   | 3rd order input intercept point LF path           | Unwanted tones @1 MHz and 1.96 MHz                                         | -    | -12 | -    | dBm  |
| HF_N_IIP3   | 3rd order input intercept point HF path           | Unwanted tones @1 MHz and 1.96 MHz                                         | -    | -12 | -    | dBm  |
| LF_W_IIP3   | 3rd order input intercept point LF path           | Unwanted tones @20 MHz and 39.4 MHz                                        | -    | -10 | -    | dBm  |
| HF_W_IIP3   | 3rd order input intercept point HF path           | Unwanted tones @20 MHz and 39.4 MHz                                        | -    | -10 | -    | dBm  |
| IMR_LF_UC   | Image attenuation, sub-GHz path                   | Without image calibration                                                  | -    | 35  | -    | dB   |
| IMR_LF_C    | Image attenuation, sub-GHz path after calibration | With image calibration BW < 500 kHz                                        | -    | 45  | -    | dB   |
| IMR_HF_UC   | Image attenuation, 2.4 GHz path                   | (LR2021/LR2022) Without image calibration on high frequency path           | -    | 25  | -    | dB   |
| IMR_HF_C    | Image attenuation, 2.4 GHz path after calibration | (LR2021/LR2022) With image calibration BW < 500 kHz on high frequency path | -    | 35  | -    | dB   |
| HF_M_IP     | Max input power at LNA input HF path              |                                                                            | -    | 6   | -    | dBm  |
| LF_M_IP     | Max input power at LNA input LF path              |                                                                            |      | 6   | -    | dBm  |

### 3.4.4 Transmit Mode Specifications

**Table 3-21: Brownout and Battery Specification (LR20xx)**

| Symbol   | Description                             | Conditions   | Min | Typ  | Max | Unit |
|----------|-----------------------------------------|--------------|-----|------|-----|------|
| BRNTRHRS | Brownout threshold voltage              |              | -   | 1.63 | -   | V    |
| BRNHYS   | Brownout detector hysteresis            |              | -   | 55   | -   | mV   |
| LBO      | Low battery indicator threshold voltage | Programmable | 1.8 | 1.88 | 2.1 | V    |
| LBOHYS   | Low battery indicator hysteresis        |              | -   | 51   | -   | mV   |

**Table 3-22: Transmit Mode Specification (LR20xx unless specified otherwise) (Sheet 1 of 2)**

| Symbol   | Description                                    | Conditions                                                                                  | Min | Typ    | Max | Unit  |
|----------|------------------------------------------------|---------------------------------------------------------------------------------------------|-----|--------|-----|-------|
| TXOPLF   | Maximum Tx power                               |                                                                                             | 19  | 22     | -   | dBm   |
| TXDRPLF  | Drop in maximum Tx power vs VDD (1.8 to 3.7 V) |                                                                                             | -   | 2      | -   | dB    |
| OCPLF    | Over current protection trigger value          | Active when PA_LF is used                                                                   | -   | 164    | -   | mA    |
| TXPRNGLF | Tx power range                                 | Programmable in steps of 0.5 dB from maximum Tx power                                       | -   | 63     | -   | steps |
| TXPRSTEP | Tx power programming step                      |                                                                                             | -   | 0.5    | -   | dB    |
| TXACCLF  | Tx output power step accuracy                  |                                                                                             | -   | +/- 2  | -   | dB    |
| TXRMPLF1 | Power amplifier ramping time                   | Programmable, lowest setting                                                                | -   | 2      | -   | μs    |
| TXRMPLF2 |                                                | Programmable, highest setting                                                               | -   | 340    | -   | μs    |
| OOKOOLF1 | OOK ON OFF ratio                               | @22 dBm                                                                                     | -   | 40     | -   | dB    |
| OOKOOLF2 |                                                | @12 dBm                                                                                     | -   | 30     | -   | dB    |
| TXOPHF   | Maximum Tx power                               | Highest power step setting                                                                  | -   | 12     | -   | dBm   |
| TXDRPHF  | Tx power drop versus supply voltage            | VDDop range, 1.8 to 3.7 V                                                                   | -   | 0.5    | -   | dB    |
| ZINTXLF  | TX LF impedance                                | Impedance to be presented to RFO_LF1 and RFO_LF2 connected together for maximum efficiency. |     |        |     |       |
|          |                                                | 490 MHz 17 dBm Targeted Power                                                               | -   | 13+j15 | -   | Ohm   |
|          |                                                | 868 MHz 14 dBm Targeted Power                                                               | -   | 11+j6  | -   | Ohm   |
|          |                                                | 915 MHz 22 dBm Targeted Power                                                               | -   | 11+j1  | -   | Ohm   |
| ZINTXHF  | TX HF impedance                                | Impedance to be presented to RFO_HF for maximum efficiency.                                 |     |        |     |       |
|          |                                                | 2450 MHz 12 dBm Targeted Power                                                              | -   | 29-j7  | -   | Ohm   |
| OCPHF    | Over current protection trigger value          | Active when PA_HF is used (LR2021/LR2022)                                                   | -   | 55     | -   | mA    |
| TXPRNGHF | RF output power range                          | Programmable in steps of 0.5 dB from maximum Tx power                                       | -   | 63     | -   | steps |

**Table 3-22: Transmit Mode Specification (LR20xx unless specified otherwise) (Sheet 2 of 2)**

| Symbol   | Description                  | Conditions                    | Min | Typ | Max | Unit |
|----------|------------------------------|-------------------------------|-----|-----|-----|------|
| TXRMPHF1 | Power amplifier ramping time | Programmable, lowest setting  | -   | 2   | -   | μs   |
| TXRMPHF2 |                              | Programmable, highest setting | -   | 340 | -   | μs   |
| OOKOOHF1 | OOK ON OFF ratio             | @12 dBm                       | -   | 30  | -   | dB   |

The following table gives the value of the Mode switching time. The transition time is defined as the time between the rising edge of the NSS ending the SPI transaction requesting a mode change and the falling edge of the BUSY signal.

**Table 3-23: Start-up/ Mode Switching Times Specifications (LR20xx) (Sheet 1 of 2)**

| Symbol   | Description                        | Timing Precisions                                                                              | Min | Typ             | Max | Unit |
|----------|------------------------------------|------------------------------------------------------------------------------------------------|-----|-----------------|-----|------|
| TSHFOSC  | HF Crystal oscillator wake-up time | From STDBY_RC <sup>1</sup>                                                                     | -   | 70 <sup>2</sup> | 400 | μs   |
| PLL_LOCK | Synthesizer lock time              |                                                                                                | -   | 32              | -   | μs   |
| TSPDRC   | Sleep to STDBY_RC                  | Cold start (without data retention) from default value, hence, calibrated at start-up          | -   | 6               | -   | ms   |
| TSPDRCW  | Sleep to STDBY_RC                  | Warm start (with data retention). Retrieve calibration from retained registers. No calibration | -   | 1               | -   | ms   |
| TSRCXO   | STDBY_RC to STDBY_XOSC             | Mainly OSC startup time                                                                        | -   | 43              | -   | μs   |
| TSRCFS   | STDBY_RC to FS                     | Mainly OSC startup time + PLL lock                                                             | -   | 76              | -   | μs   |
| TSRCRX   | STDBY_RC to Rx                     | Mainly OSC startup time + PLL lock + Rx Front end startup                                      | -   | 115             | -   | μs   |
| TSRCTX   | STDBY_RC to Tx                     | Mainly OSC startup time + PLL lock + PA ramping. Depends on selected ramp time                 | -   | 106             | -   | μs   |
| TSXORC   | STDBY_XOSC to STDBY_RC             |                                                                                                | -   | 41              | -   | μs   |
| TSXOFS   | STDBY_XOSC to FS                   | Synthesizer wake-up time from STDBY_XOSC<br>Mainly PLL lock                                    | -   | 44              | -   | μs   |
| TSXOTX   | STDBY_XOSC to Tx                   | Mainly PLL lock + PA ramping Depends on selected ramp time                                     | -   | 76              | -   | μs   |
| TSXORX   | STDBY_XOSC to Rx                   | Mainly PLL lock + Rx Front end startup                                                         | -   | 84              | -   | μs   |
| TSFSRC   | FS to STDBY_RC                     |                                                                                                | -   | 76              | -   | μs   |
| TSFSXO   | FS to STDBY_XOSC                   |                                                                                                | -   | 51              | -   | μs   |
| TSFSRX1  | FS to Rx                           | LoRa: Mainly PLL lock + Rx Front end start-up                                                  | -   | 53              | -   | μs   |
| TSFSRX2  |                                    | FSK: Mainly PLL lock + Rx Front end start-up                                                   | -   | 37              | -   | μs   |
| TSFSTX1  | FS to Tx                           | Depends on selected ramp time                                                                  | -   | 45              | -   | μs   |
| TSRXRC   | Rx to STDBY_RC                     |                                                                                                | -   | 106             | -   | μs   |
| TSRXXO   | Rx to STDBY_XOSC                   |                                                                                                | -   | 79              | -   | μs   |
| TSRXFS1  | Rx to FS                           | Rx in LoRa                                                                                     | -   | 10              | -   | μs   |
| TSRXFS2  |                                    | Rx in FSK                                                                                      | -   | 11              | -   | μs   |

**Table 3-23: Start-up/ Mode Switching Times Specifications (LR20xx) (Sheet 2 of 2)**

| Symbol | Description      | Timing Precisions | Min | Typ | Max | Unit |
|--------|------------------|-------------------|-----|-----|-----|------|
| TSRXTX | Rx to Tx         |                   | -   | 107 | -   | μs   |
| TSTXRC | Tx to STDBY_RC   |                   | -   | 116 | -   | μs   |
| TSTXXO | Tx to STDBY_XOSC |                   | -   | 90  | -   | μs   |
| TSTXFS | Tx to FS         |                   |     | 54  |     | μs   |
| TSTXRX | Tx to Rx         |                   |     | 91  |     | μs   |

1. Wake-up time till crystal oscillator frequency is within +/- 10 ppm for Normal mode

2. TSHFOSC maximum specification assume pre-trimming of the start-up current depending on the application board and crystal characteristics

### 3.5 Reference Oscillator Crystal Specification

The tables below give the electrical specifications of the LR2021/LR2022/LR2012 transceiver under the following conditions, unless otherwise specified:

V<sub>BAT</sub> = 3.3 V, Temperature = 25 °C, FXOSC = 32 MHz XTAL used.

**Table 3-24: 32 MHz Crystal Specifications. (LR20xx) For example NDK\_NX2016SA**

| Symbol   | Description                  | Conditions                     | Min | Typ  | Max    | Unit |
|----------|------------------------------|--------------------------------|-----|------|--------|------|
| FXOSCHF  | Crystal oscillator frequency |                                | -   | 32   | -      | MHz  |
| CLOADHF  | Crystal loading capacitance  | Differential                   | 9.5 | 10   | 10.5   | pF   |
| COXTALHF | Crystal shunt capacitance    |                                | 0.3 | 0.6  | 2      | pF   |
| RSXTALHF | Crystal series resistance    |                                | -   | 30   | 60     | Ω    |
| CMXTALHF | Crystal motional capacitance |                                | 1.3 | 1.89 | 2.5    | fF   |
| DRIVEHF  | Drive level                  |                                | -   | -    | 100    | μW   |
| FRTOLHF  | Crystal frequency accuracy   | Initial                        | -   | -    | +/- 10 | ppm  |
|          |                              | Over temperature (-20 to 70°C) | -   | -    | +/- 10 | ppm  |
|          |                              | Aging over 10 years            | -   | -    | +/- 10 | ppm  |

**Table 3-25: TCXO Regulator Specifications (LR20xx)**

| Symbol | Description                                                                | Conditions                                        | Min | Typ | Max | Unit   |
|--------|----------------------------------------------------------------------------|---------------------------------------------------|-----|-----|-----|--------|
| CVTCXO | External decoupling capacitor for REG_TCXO                                 | X5R type recommended                              | 70  | 100 | 130 | nF     |
| RVTCXO | Equivalent series resistance for CVTCXO                                    |                                                   | 0.1 | -   | 1   | Ω      |
| VTCXO  | Regulated voltage range for TCXO voltage supply,<br>VDDop > VTCXO + 200 mV | RegTcxoTune = 000                                 | -   | 1.6 | -   | V      |
|        |                                                                            | RegTcxoTune = 001                                 | -   | 1.7 | -   | V      |
|        |                                                                            | RegTcxoTune = 111                                 | -   | 3.3 | -   | V      |
| ILTCXO | Load current for TCXO regulator                                            |                                                   | -   | 1.5 | 4   | mA     |
| ATCXO  | Amplitude voltage for external TCXO applied to XTA pin, clipped-sine       | AC coupled through 10 pF DC-cut series with 220 Ω | 0.4 | 0.6 | 1.2 | Vpk-pk |

## 3.6 SIMO Inductor Specification

The SIMO requires an external inductor for operation. The specification is:

Value: 2.2  $\mu$ H

DCR (max): 0.5  $\Omega$

Isat (min): 200 mA

Resonant Frequency (min): 20 MHz

Suggested part numbers: LQM18PN2R2MGH and LSQNB160808T2R2M.

## 3.7 Digital I/O and Interface Specifications

The tables below give the electrical specifications of the LR2021/LR2022/LR2012 transceiver under the following conditions, unless otherwise specified:

VBAT = 3.3 V, Temperature = 25 °C, FXOSC = 32 MHz XTAL used.

### 3.7.1 Digital I/O Specifications

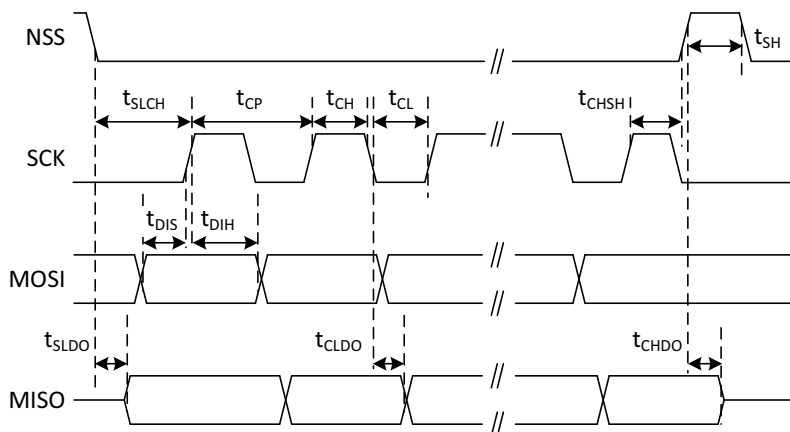
**Table 3-26: Digital I/O Specifications (LR20xx)**

| Symbol | Description                    | Condition                          | Min      | Typ | Max      | Unit       |
|--------|--------------------------------|------------------------------------|----------|-----|----------|------------|
| VIH    | Input High Voltage             |                                    | 0.7*VBAT | -   | VBAT+0.3 | V          |
| VIL    | Input Low Voltage              |                                    | -0.3     |     | 0.3*VBAT | V          |
| VOH    | Output High Voltage            | $I_{max} = 2.5$ mA                 | 0.9*VBAT |     | VBAT     | V          |
| VOL    | Output Low Voltage             | $I_{max} = -2.5$ mA                | 0        |     | 0.1*VBAT | V          |
| ZPUD   | Pull-up and Pull-down Resistor | Any DIO pin when pulled up or down |          | 40  |          | k $\Omega$ |

### 3.7.2 SPI Interface Specifications

The Serial Peripheral Interface (SPI) is used as host interface supporting mode 0 (CPOL/CPHA=0). The interface can only be operated in Subordinate mode only. A transfer is always started by a falling edge of the Subordinate select (NSS). MISO is high impedance when NSS is high. The SPI runs on the external SCK clock with a maximum frequency of 16 MHz.

**Figure 3-1: SPI Timing Diagram**



**Table 3-27: SPI Timing Requirements (LR20xx)**

| Symbol     | Description                                              | Condition                         | Min   | Typ | Max | Unit |
|------------|----------------------------------------------------------|-----------------------------------|-------|-----|-----|------|
| $t_{SLCH}$ | Subordinate select falling edge to SCK setup time        |                                   | 31.25 | -   | -   | ns   |
| $t_{CP}$   | SCK clock period                                         |                                   | 61.50 | -   | -   | ns   |
| $t_{CH}$   | SCK high time                                            |                                   | 31.25 | -   | -   | ns   |
| $t_{CL}$   | SCK low time                                             |                                   | 31.25 | -   | -   | ns   |
| $t_{CHSH}$ | SCK rising edge to Subordinate select rising edge        |                                   | 65    | -   | -   | ns   |
| $t_{SH}$   | Subordinate select high time                             | Maximum load capacitance of 10 pF | 125   | -   | -   | ns   |
| $t_{DIS}$  | MOSI data input setup time                               |                                   | 15    | -   | -   | ns   |
| $t_{DIH}$  | MOSI data input hold time                                |                                   | 15    | -   | -   | ns   |
| $t_{SLDO}$ | Subordinate select falling edge to MISO data output time |                                   | -     | -   | 10  | ns   |
| $t_{CLDO}$ | SCK falling edge to MISO data output time                |                                   | -     | -   | 18  | ns   |
| $t_{CHDO}$ | SCK rising edge to MISO data output time                 |                                   | -     | -   | 10  | ns   |



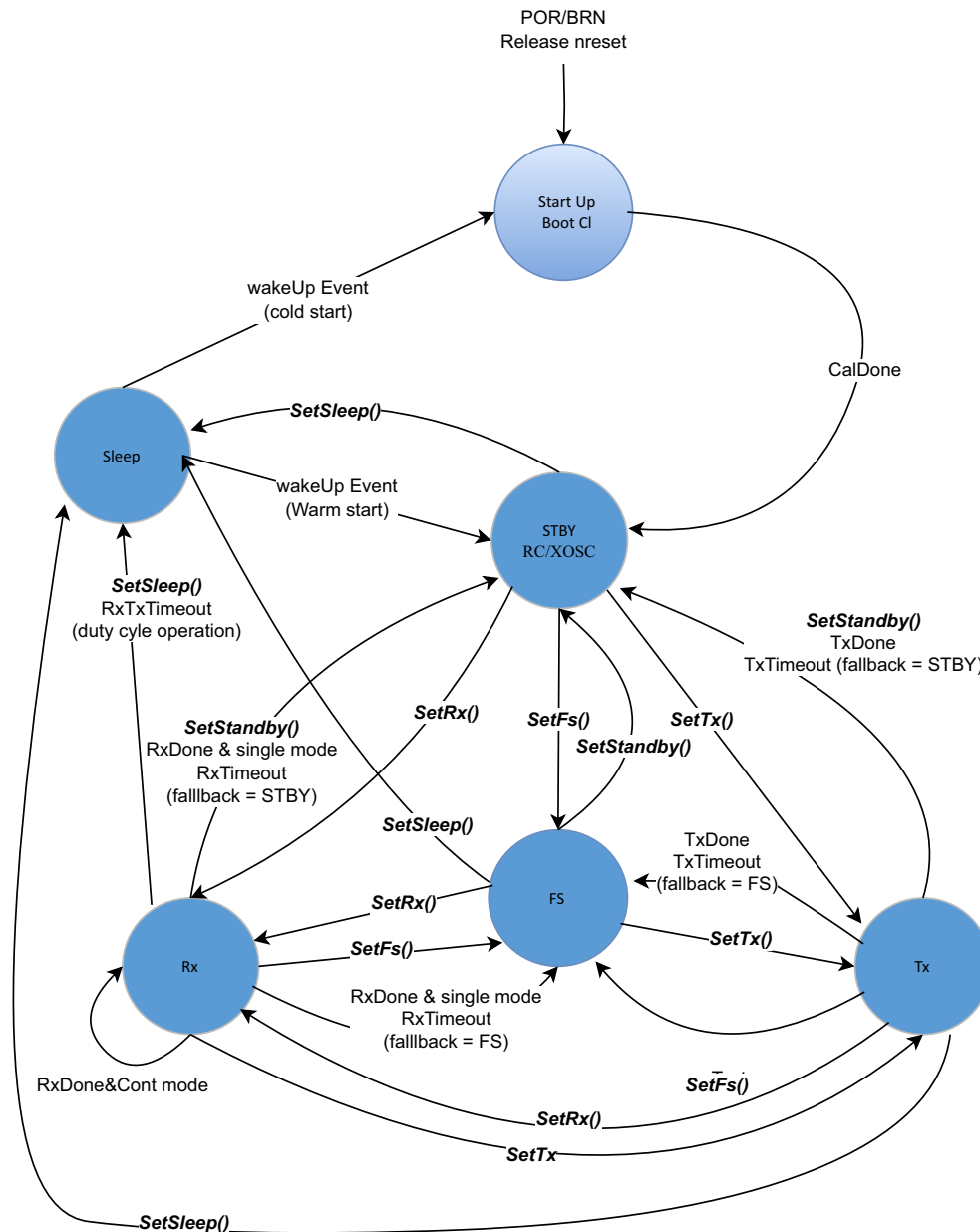
## 4. Chip Modes (LR20xx)

### 4.1 Chip Mode State Diagram

The chip modes define the different modes of operation of the device. These include low power modes, standby modes and operational radio modes.

**Note:** RC refers to the internal or external 32 kHz oscillator and XOSC refers to the 32 MHz precision clock from XTAL oscillator or TCXO.

Figure 4-1: Chip Modes



[Table 4-1](#) defines the chip mode and the conditions required to get out of those modes as well as the reachable mode.

**Table 4-1: Chip Mode Conditions (LR20xx)**

| Chip Mode                  | Mode seen by Host            | Description                                                                                                             | Output Condition and Reachable Final Mode                                                                                                                                                                                                                                                                                                                               |
|----------------------------|------------------------------|-------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Startup + boot calibration | Sleep <sup>1</sup><br>busy=1 | Transition mode where all internal power supplies are setup and all initial calibration are performed                   | Automatic once all power supplies are established and initial calibration is completed                                                                                                                                                                                                                                                                                  |
| Sleep                      | Sleep <sup>1</sup>           | Lowest power mode<br>can be with or without data RAM in retention<br>Can be with or without RTC clock and timer running | After Reset: Standby RC via cold start<br>NSS falling: Standby RC via warm start if data RAM is in retention else Standby RC via cold start<br>RTC count reached if RC clock active: Standby RC via warm start if data RAM is in retention else Standby RC via cold start<br>RTC count reached during an Rx duty cycle operation. Final state is Rx after a warm start. |
| Standby RC <sup>2</sup>    | StandbyRC                    | Mode for chip configuration                                                                                             | After Reset: Standby RC via cold start<br>Command: all modes                                                                                                                                                                                                                                                                                                            |
| Standby XOSC               | StandbyXOSC                  | Mode for chip configuration                                                                                             | After Reset: Standby RC via cold start<br>Command: all modes                                                                                                                                                                                                                                                                                                            |
| FS                         | FS                           | Frequency synthesiser mode                                                                                              | After Reset: Standby RC via cold start<br>Command: all modes                                                                                                                                                                                                                                                                                                            |
| Rx                         | Rx                           | Rx mode for packet reception                                                                                            | After Reset: Standby RC via cold start<br>Command: all modes                                                                                                                                                                                                                                                                                                            |
| Tx                         | Tx                           | Tx mode for packet transmission                                                                                         | After Reset: Standby RC via cold start<br>Command: all modes                                                                                                                                                                                                                                                                                                            |

1. In Sleep mode at startup, the busy line is 1.

2. See [Section 4.4.2](#).

The transition from Rx to Tx or Tx to Rx is done under control of the chip via FS mode.

## 4.2 Reset

The LR2021/LR2022/LR2012 has three distinct reset sources that can trigger a reset of the device, initiating the start-up sequence for smooth operation:

- ◆ The Power-On-Reset (POR) is one of the reset sources available in the LR2021/LR2022/LR2012. It initiates a restart of the device when the power supply is first applied, ensuring a reliable and controlled start-up process.
- ◆ The Brown-Out Reset (BRN) is another reset source that triggers a device restart. It comes into play if the supply voltage falls below a certain threshold, providing an added layer of protection against improper operation under low-voltage conditions.
- ◆ The RESET procedure is as follows:
  1. Assert NRESET low for minimum 100  $\mu$ s.
  2. Release NRESET.
  3. Wait for boot calibration (see [Section 4.3](#)).
  4. Device enters STDBY\_RC mode.
  5. Ready for SPI commands.
  6. In the post-RESET state:
    - ◆ All registers return to default values
    - ◆ FIFO contents are cleared
    - ◆ DIO pins are in default configuration.

During each of the reset procedures mentioned above, the BUSY signal remains high, indicating that the LR2021/LR2022/LR2012 is undergoing the restart process.

After the reset event, the device enters its startup sequence.

## 4.3 Start-up and Boot Calibration Sequence

During power-up or after a reset event, the LR2021/LR2022/LR2012 initiates its start-up phase, where various internal processes are executed to prepare the device for operation. Throughout this start-up phase, the BUSY signal is set to high, indicating that the device is currently busy with its initialization and cannot accept any external commands.

The start-up sequence encompasses essential tasks such as initialization, configuration, and calibration to ensure the device's proper operation.

Once the calibration phase is completed, the LR2021/LR2022/LR2012 transitions into Standby RC mode. In this mode, the device is ready to accept external commands and perform its intended operations. As the device enters Standby RC mode, the BUSY signal returns to a low state, indicating that the LR2021/LR2022/LR2012 is now idle and capable of receiving commands from the developer.

Refer to [Section 6.4](#) for Calibration details.

## 4.4 Chip Modes

### 4.4.1 Sleep

Sleep mode can be activated using the command [SetSleep](#).

During Sleep mode, a real time clock (RTC) driven by LF\_CLK clock can be activated. A chip parameters retention mechanism can be optionally enabled in Sleep mode to save all previously programmed chip settings. Depending on Sleep mode configuration, the conditions and sequences to leave Sleep mode are affected.

The chip leaves Sleep mode if NSS is asserted low for 100us, or with any reset source as described in [Section 4.2](#).

### 4.4.2 Standby RC / XOSC

In Standby mode, the host must configure the chip before transitioning to any other operational mode. Standby mode serves as a low-power state where the device awaits instructions from the host before entering active modes of operation.

Within Standby mode, the chip offers two sub-modes that cater to specific power and switching requirements:

- ◆ Standby RC Mode: The system is clocked by the internal RC32M oscillator. By utilizing the internal RC32M oscillator, the device can maintain basic functionality while conserving energy, ideal for situations where power efficiency is a primary concern.
- ◆ Standby XOSC Mode: The system is clocked by the external XOSC32M crystal or TCXO (Temperature Compensated Crystal Oscillator).

The developer can elect to use either RC or XOSC mode to optimize power consumption and transition time to FS or Rx/Tx mode. Corresponding transition time values and power consumptions are tabulated in [Table 3-23](#) and [Table 3-4](#).

### 4.4.3 Frequency Synthesis (FS)

In FS mode, the Phase-Locked Loop (PLL) and its associated regulators are activated, enabling frequency synthesis for signal generation.

As the device operates in FS mode, the BUSY signal remains high throughout the PLL locking process. However, once the PLL achieves locking or times out due to unsuccessful locking attempts, the BUSY signal transitions from high to low. A PLL lock error condition can be detected by using the [GetErrors](#) command.

### 4.4.4 Receive (Rx)

In Rx (receiver) mode, the PLL (Phase-Locked Loop), Digital Front End, Analog Front End, and the selected modem are activated. Only one modem can be active at any given time.

Rx mode can operate in different sub-modes, each designed to meet specific application requirements:

- ◆ Continuous Mode: In continuous mode, the device remains in Rx mode, continuously searching for incoming packets until the host requests a different mode. This mode is well-suited for scenarios where continuous data reception is required.
- ◆ Single Mode: In single mode, the device automatically returns to a configured mode defined by [SetRxTxFallbackMode](#) (default Standby RC) after successfully receiving a packet. This mode offers a straightforward and efficient approach for receiving single packets without the need for additional manual intervention.
- ◆ Single with Timeout Mode: In single with timeout mode, the device also automatically returns to a configured mode (default Standby RC) after packet reception, but with the added functionality of a timeout feature. If a Syncword or preamble or LoRa header is detected during packet reception, the timeout is stopped, allowing the device to stay in Rx mode for extended periods if necessary.
- ◆ Rx Duty Cycle Mode: In Rx Duty Cycle mode, the device periodically switches into Rx mode to receive a packet before returning to Sleep mode. This cyclical behaviour continues until a packet is received. Before entering Duty Cycle mode, the low-frequency clock source for Rx/Tx (internal 32 kHz RC or external 32 kHz input) should be configured using [SetRxDutyCycle](#) command.

### 4.4.5 Transmit (Tx)

In Tx (transmitter) mode, the PLL (Phase-Locked Loop), analog front end, Power-Amplifier (PA), and the selected modem are activated. As with Rx mode, only one modem can be active at a time to ensure correct operation.

TX mode can operate in different sub-modes, each catering to specific transmission requirements:

- ◆ Single Mode: In single mode, the device automatically returns to a configured mode defined by [SetRxTxFallbackMode](#) (default Standby RC) after successfully transmitting a packet. This mode is suitable for transmitting single packets without the need for manual intervention once the transmission is complete.
- ◆ Single Mode with Timeout: In single mode with timeout, the device also automatically returns to a configured mode (default Standby RC) after packet transmission. Additionally, the device exits TX mode with a Timeout IRQ (if enabled) if the transmission is not successful within the given timeout time.
- ◆ Continuous Wave (CW) Mode: In CW mode, the carrier frequency is transmitted indefinitely until another command is issued to change the mode. This mode is useful for generating continuous RF signals without any data modulation, and it is often used for testing and signal generation purposes.
- ◆ Infinite Preamble Mode: In infinite preamble mode, an infinite preamble of the configured modulation is continuously output on the RF (Radio Frequency). This mode is employed for generating specific signal patterns at the start of the transmission.

## 5. Host Interface

### 5.1 Hardware Interface (LR20xx)

#### 5.1.1 DIO Pins

The LR2021/LR2022/LR2012 features 7 I/O pins, DIO5 to DIO11, which serve as general-purpose digital I/Os. These pins can pilot RF switches, trigger Rx or Tx operations or can be used as IRQs for host notification.

When the chip enters Standby RC mode after a POR/BRN or reset command, the DIO pins are at high impedance (HIZ) except DIO5 and 6 that are in pull up state.

The DIO configurations are defined using command [SetDioFunction](#) in Standby RC mode and only in that mode. When the chip goes back to Sleep mode, the DIO configuration is maintained.

In Sleep mode, the DIO cannot drive a high or low state, its value can only be set via **pull\_drive** parameter (none, pull-down, pull-up, pull-auto). See chapter [6.8.1 SetDioFunction](#) for more details.

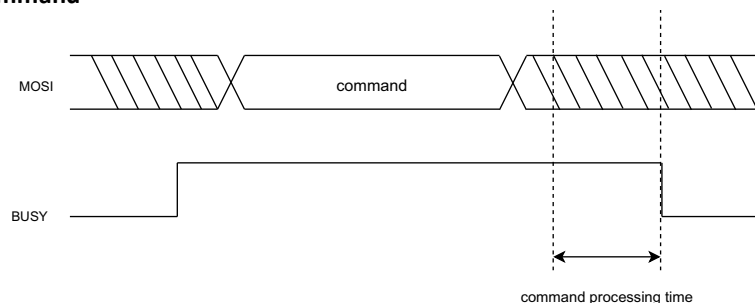
#### 5.1.2 SPI Interface

The chip is designed to be controlled through a synchronous full-duplex SPI (Serial Peripheral Interface) communication interface. 4 pins are dedicated to the SPI interface signals NSS, SCK, MOSI and MISO. See [Section 3.7.2](#) for details.

### 5.2 Busy Line Control (LR20xx)

The BUSY line indicates the status of the chip controller. When the BUSY line is held low, it indicates that the radio is ready to accept a command from the host controller. The BUSY pin is automatically asserted on the falling edge of the NSS. Once the chip has finished processing the command, the BUSY line is de-asserted to indicate that the chip has reached a stable mode device and is ready to accept another command.

**Figure 5-1: Write Command**



Inherently, the amount of time the BUSY line stays high depends on the nature of the command. For example, setting the device into TX mode from the STDBY\_RC mode takes much more time than simply changing some radio parameters because the internal state machine maintains the BUSY line high until the radio is effectively transmitting the packet.

- In FS mode, BUSY goes low when the PLL is locked.
- In Rx mode, BUSY goes low as soon as the chip is ready to receive data.
- In Tx mode, BUSY goes low when the PA has ramped-up and transmission of preamble starts.

The BUSY line must go low before sending an SPI command (either a “read” or “write” command) as otherwise, the behaviour of the circuit is not defined and could lead to un-executed commands or wrongly executed commands.

## 5.3 Data FIFO (LR20xx)

The LR2021/LR2022/LR2012 transceiver is equipped with two 256-byte data buffers, functioning as FIFOs (First In First Out) for data transmission and reception (Tx data FIFO and Rx data FIFO).

These data buffers are accessible in all operational modes except for the Sleep modes. However, after waking up from any Sleep mode, the contents of both data buffers are cleared, resulting in the loss of all data previously stored in them.

The LR2021/LR2022/LR2012 provides programmable FIFO level thresholds for the data buffers, using IRQs. This feature enhances data flow control and enables direct communication between the host application and the transceiver.

The LR2021/LR2022/LR2012 provides multiple methods for monitoring the FIFO level:

- ◆ IRQs (Interrupt Requests) mapped on the DIO (Digital Input/Output) pins: The FIFO level can trigger specific IRQs, which, when detected on the DIO pins, indicate different fill levels of the FIFO. This allows the Host controller to receive real-time notifications about the FIFO status, enabling timely data handling.
- ◆ Readout through dedicated APIs ([GetRxFifoLevel](#) and [GetTxFifoLevel](#)). The dedicated Application Programming Interface (API) allows the Host controller to directly read the FIFO level, providing an efficient way to access the FIFO level information when needed.

### 5.3.1 Rx Data FIFO

In receive mode, the LR2021/LR2022/LR2012's activated modem automatically adds received data to the Rx data FIFO. The level of the FIFO is continuously monitored to trigger specific interrupts. Status information in the [ConfigFifoIrq](#) command includes *rx\_low\_threshold*, *rx\_high\_threshold*, and FIFO flags *FifoEmpty*, *FifoFull*, *FifoOverflow*, and *FifoUnderflow*.

The level of the Rx FIFO can be directly read using the [GetRxFifoLevel](#) command. The [ReadRadioRxFifo](#) command reads received bytes sequentially from the Rx data FIFO. Additionally, the [GetRxPktLength](#) command reads the length of the last received packet.

In Rx continuous mode, all payload data from received packets are appended to the Rx FIFO. If the packet lengths are not known in advance, the payload length needs to be read after each RxDone interrupt to determine which data belongs to which packet. As long as the total number of received data fits into the FIFO and the developer is aware of the various payloads' lengths, there is no limit to the number of concurrent packets stored in the Rx FIFO.

For payloads exceeding 256 bytes, the Rx FIFO must be read out while receiving new data utilizing the threshold information available through the API or on the DIOs to avoid overflow. The Rx FIFO can be cleared using the [ClearRxFifo](#) command, discarding all data present in the FIFO.

All received data is written to the data buffer, even if the CRC (Cyclic Redundancy Check) is invalid, allowing for user-defined post-processing of corrupted data. However, in the case of address error or length error, data are not received.

### 5.3.2 Tx Data FIFO

In transmit mode, the LR2021/LR2022/LR2012 requires that the data to be sent is written into the Tx data FIFO using the [WriteRadioTxFifo](#) command. The level of the Tx FIFO is continuously monitored to trigger specific interrupts). Status information in the [ConfigFifoIrq](#) command includes *tx\_low\_threshold*, *tx\_high\_threshold* and FIFO flags *FifoUnderflow*, *FifoEmpty*, *FifoFull*, and *FifoOverflow*.

The fill level of the Tx FIFO can also be directly read using the [GetTxFifoLevel](#) command. For payloads superior to 256 bytes, the Tx FIFO has to be written when data is being transmitted, avoiding an underflow of the Tx FIFO, using the threshold informations available through the API or the DIOs.

The Tx FIFO can be cleared with the [ClearTxFifo](#) command, discarding all data still in the FIFO.

### 5.3.3 Extending the Data FIFOs

The default size of both the RX and TX FIFO is 256 bytes. The LR20xx drivers also expose a set of functions that move the FIFOs to a larger memory region, allowing both FIFOs to reach 1024 bytes in size.

**Note:** While using the extended FIFOs, if any CalibFe command is issued, the content of the FIFOs is not valid anymore. Any data in the FIFOs is overwritten with data used during calibration. Sequences such as "Fill TX FIFO CalibFE Transmit" or "RX data CalibFE Read RX FIFO" must be avoided.

**Table 5-1: Radio FIFO Switch Driver Functions**

| Function                                                                                              | Purpose                                                 |
|-------------------------------------------------------------------------------------------------------|---------------------------------------------------------|
| <code>lr20xx_radio_fifo_configure_1024_byte_tx_fifo(context)</code>                                   | Switch Tx FIFO to the 1024 byte region                  |
| <code>lr20xx_radio_fifo_configure_1024_byte_rx_fifo(context)</code>                                   | Switch Rx FIFO to the 1024 byte region                  |
| <code>lr20xx_radio_fifo_1024_byte_tx_fifo_store_retention_mem(context, slot_addr, slot_size)</code>   | Retain Tx FIFO switch across sleep                      |
| <code>lr20xx_radio_fifo_1024_byte_rx_fifo_store_retention_mem(context, slot_addr, slot_size)</code>   | Retain Rx FIFO switch across sleep                      |
| <code>lr20xx_workarounds_1024_byte_fifo_cfg_irq(context, ...)</code>                                  | Configure FIFO threshold IRQs with low thresholds > 256 |
| <code>lr20xx_workarounds_1024_byte_fifo_cfg_irq_store_retention_mem(context, slot_rx, slot_tx)</code> | Retain the workaround threshold registers across sleep  |

#### 5.3.3.1 How to switch to the 1024 byte FIFOs

The move is a configuration that is typically performed during radio initialization, right after reset/wakeup and before the first Tx or Rx. Each FIFO is configured independently.

##### 5.3.3.1.1 Step 1: Switch the FIFO(s) to the large region

Use the functions `lr20xx_radio_fifo_configure_1024_byte_tx_fifo` and `lr20xx_radio_fifo_configure_1024_byte_rx_fifo` to change the FIFOs to the larger location. After this, `lr20xx_radio_fifo_write_tx()` / `lr20xx_radio_fifo_read_rx()` operate on the larger FIFOs transparently.

Under the hood, each call writes two registers: the FIFO base address and the FIFO size. The writes are done via `WriteRegMem32(register_address, new_register_value);`

**Table 5-2: Radio FIFO Base Address Registers**

| Register             | Register address | New register value (extended base address) |
|----------------------|------------------|--------------------------------------------|
| Tx FIFO base address | 0x00F3002C       | 0x00804000                                 |
| Rx FIFO base address | 0x00F30028       | 0x00804400                                 |



Table 5-3: Radio FIFO Size Registers

| Register     | Register address | New register value (1024 bytes FIFO) | Size interpretation                                                                                                                     |
|--------------|------------------|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| Tx FIFO size | 0x00F30034       | 0x000003FC (1020 decimal)            | Size is bits 9:2 (8 bits) representing N + 1 words. Total 1024 bytes                                                                    |
| Rx FIFO size | 0x00F30030       | 0x000003E8 (1000 decimal)            | Size is bits 9:2 (8 bits) representing N + 1 words. There are an additional 4 bytes that are implicit in the Rx chain. Total 1024 bytes |

#### 5.3.3.1.2 Step 2 (optional): Configure threshold IRQs

If the application does not use the low-threshold IRQs, or if all the low thresholds stay within the range 0–256, this step is not needed.

If the application needs the low thresholds (tx\_fifo\_low\_threshold / rx\_fifo\_low\_threshold) to exceed 256, meaning the application is expecting to send or receive packets that are larger than 1 kB and must make use of the dynamic buffer reading/filling, then the application must use the workaround wrapper `lr20xx_workarounds_1024_byte_fifo_cfg_irq`, as the standard `lr20xx_radio_fifo_cfg_irq()` cannot express low thresholds above 256. The argument order mirrors the one of `lr20xx_radio_fifo_cfg_irq()`.

#### 5.3.3.1.3 Step 3 (optional): Preserve the configuration across sleep

The FIFO location registers and the workaround threshold registers are not retained by default when the chip enters sleep mode.

If the application sleeps between packets and wakes with retention (warm start), it can call the following functions, which ensures the configuration is restored automatically:

- `lr20xx_radio_fifo_1024_byte_tx_fifo_store_retention_mem`,
- `lr20xx_radio_fifo_1024_byte_rx_fifo_store_retention_mem`, and
- `lr20xx_workarounds_1024_byte_fifo_cfg_irq_store_retention_mem`.

After a cold start, the application must run Steps 1 and 2 again regardless.

## 5.4 Host-Command Interface (LR20xx)

### 5.4.1 Read Write Command

The LR2021/LR2022/LR2012 is controlled through a set of SPI commands and responses.

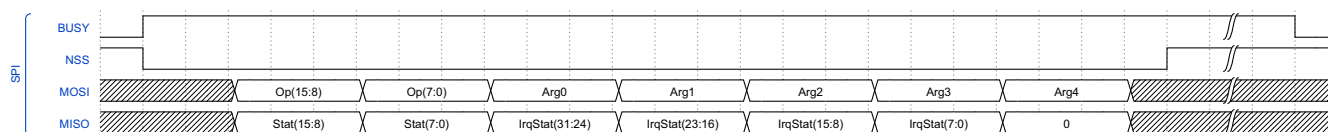
The BUSY signal serves as a handshake mechanism. It indicates whether the LR2021/LR2022/LR2012 is ready to accept a new command from the Host controller. When the BUSY signal is in a high state, it signifies that the LR2021/LR2022/LR2012 is occupied and cannot receive a new command. Conversely, when the BUSY signal is low, it indicates that the LR2021/LR2022/LR2012 is ready to process a new command.

Additionally, the LR2021/LR2022/LR2012 incorporates status variables called **stat**, which provide essential information about the device's current status. These status variables convey details such as the result of the last command sent, device interrupts, and the current operating mode.

To obtain the status registers, the Host controller can use the [GetStatus](#) command, which returns the relevant status information. Furthermore, the stat is always sent by the LR2021/LR2022/LR2012 when the Host issues a command.

## 5.4.1.1 Write Commands

**Figure 5-2: Write Command**



When the Host controller communicates with the LR2021/LR2022/LR2012 using SPI, the communication process involves sending an opcode (16 bits) followed by the required arguments. The LR2021/LR2022/LR2012 indicates its readiness to process the command by asserting the BUSY pin when the NSS (Subordinate Select) signal's falling edge is detected for the SPI communication. Once the LR2021/LR2022/LR2012 completes processing the command, it de-asserts the BUSY line to signal the end of the processing.

During the SPI communication, the first bytes shifted out on the MISO (Master In Slave Out) pin consist of the 16 status bits and the IrqStatus (31:0) values, providing information about the LR2021/LR2022/LR2012's status and interrupt status, followed by the '0's.

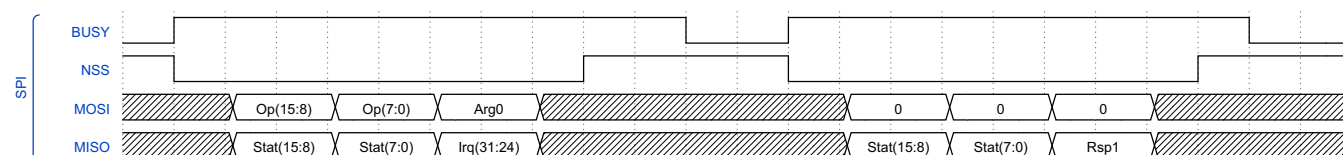
In the case of an ongoing read command, the 16 status bits are followed by the prepared data from the last read request.

## 5.4.1.2 Read Commands

In the SPI communication between the Host controller and the LR2021/LR2022/LR2012, the Host initiates the process by sending an opcode (16 bits) along with the required arguments. To indicate its readiness to process the command, the LR2021/LR2022/LR2012 asserts the BUSY pin when it detects the falling edge of the NSS (Subordinate Select) signal.

Once the LR2021/LR2022/LR2012 has finished preparing the requested data, it de-asserts the BUSY pin, signalling the completion of the operation. At this point, the Host controller can read back the data by sending 0x00 or any other appropriate value to shift out the data on the SPI.

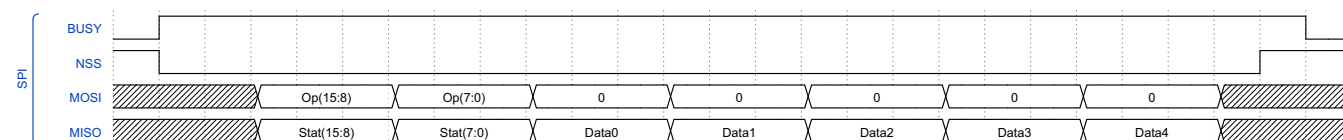
**Figure 5-3: Read Command**



## 5.4.1.3 Direct Read

The LR2021/LR2022/LR2012's radio FIFO read command is executed in a single SPI frame, which distinguishes it from other read commands where the data can only be read in a second SPI frame. To expedite SPI operations, data is read byte-by-byte from the radio FIFO immediately after the opcode has been parsed as a FIFO read operation.

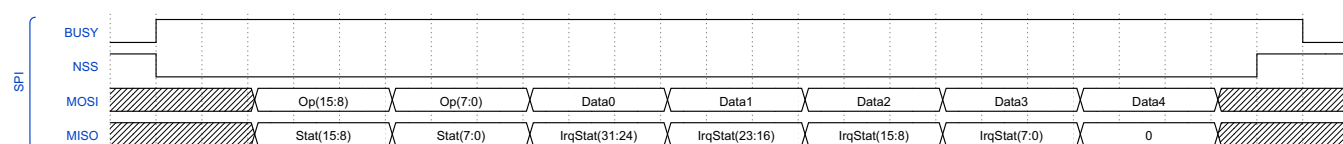
**Figure 5-4: Direct Read Timing Diagram**



## 5.4.1.4 Direct Write

Unlike other write commands that are processed only on the rising edge of the NSS (Subordinate Select) signal, the radio FIFO write command directly writes data continuously into the TX FIFO.

**Figure 5-5: Direct Write Timing Diagram**



## 5.5 DIO Rx / Tx Triggers (LR20xx)

When a DIO (Digital Input/Output) pin is configured as an Rx or Tx trigger, it starts an Rx or Tx sequence (behavior identical to *SetRx* or *SetTx* command) in the LR2021/LR2022/LR2012 transceiver when a DIO is asserted. During this sequence, the default timeouts (configured with the *SetDefaultRxTxTimeout* command) are applied.

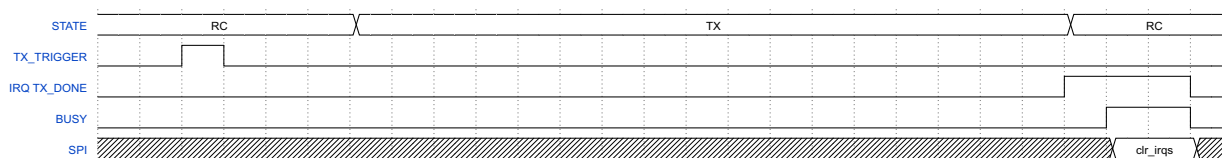
Unlike other commands, when triggered by a DIO pin, the BUSY signal is not raised. This is because a command can be sent immediately after the trigger, but only direct write or read commands are executed immediately. All other commands are postponed until after the *set\_rx* or *set\_tx* sequence is completed.

The BUSY signal is raised on the falling edge of NSS (Subordinate Select), and is reset to low after the postponed command is processed (normal command) or once the *SetTx/SetRx* command from the trigger is processed (direct\_read or direct\_write shorter than the *set\_tx/set\_rx* sequence).

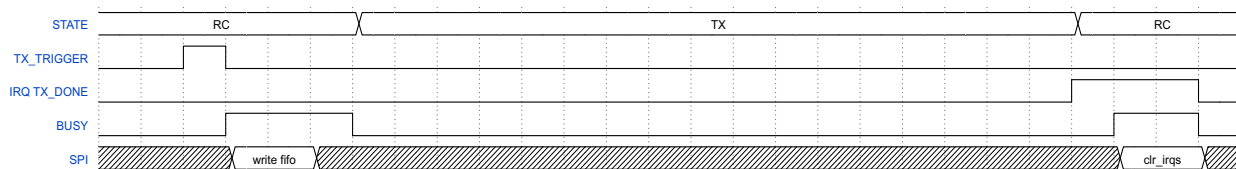
To determine when the LR2021/LR2022/LR2012 is actually in Rx mode, for example, the Host controller can send a *GetStatus* command after the trigger, and the BUSY signal goes low once the device is in Rx mode.

If a command initiating a mode change is already ongoing when a DIO trigger is received, the trigger is discarded, and the error bit CHIP\_BUSY is set to indicate that the LR2021/LR2022/LR2012 is currently busy processing another command (can be retrieved from the *GetErrors* command). For other commands the trigger takes precedence over the ongoing command.

**Figure 5-6: Tx Trigger Example**



**Figure 5-7: Tx Trigger Followed by FIFO Write Example**



## 5.6 Host Commands Summary

Host command opcodes are split into groups so the main API can easily redirect the commands to the dedicated API.

### 5.6.1 Direct Radio FIFO Read/Write Commands (LR20xx)

These commands use the direct read and direct write SPI frames.

**Table 5-4: Radio FIFO Memory Commands**

| Command                 | Opcode | Parameters | Description                            |
|-------------------------|--------|------------|----------------------------------------|
| <i>ReadRadioRxFifo</i>  | 0x0001 | ---        | Reads back data from the radio Rx FIFO |
| <i>WriteRadioTxFifo</i> | 0x0002 | Data[]     | Writes data into radio Tx FIFO         |

### 5.6.2 Register / Memory Access Operations (LR20xx)

**Table 5-5: Register Memory Commands**

| Command                  | Opcode | Parameters                           | Description                                                  |
|--------------------------|--------|--------------------------------------|--------------------------------------------------------------|
| <i>WriteRegMem32</i>     | 0x0104 | Addr(23:0)<br>Data[1:64]             | Writes data at given register/memory address.                |
| <i>WriteRegMemMask32</i> | 0x0105 | Addr(23:0), Mask(31:0)<br>Data(31:0) | Reads-Modifies-Writes data at given register/memory address. |
| <i>ReadRegMem32</i>      | 0x0106 | Addr(23:0)<br>Len(7:0)               | Reads data at given register/memory address.                 |

### 5.6.3 System Configuration / Status Operations (LR20xx)

Table 5-6: System Configuration Commands (Sheet 1 of 2)

| Command                     | Opcode | Parameters                                                                                                                                                   | Description                                                                        |
|-----------------------------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| <i>GetStatus</i>            | 0x0100 | ---                                                                                                                                                          | Returns status of device                                                           |
| <i>GetVersion</i>           | 0x0101 | ---                                                                                                                                                          | Returns version of firmware                                                        |
| <i>GetErrors</i>            | 0x0110 | ---                                                                                                                                                          | Returns error status of the device                                                 |
| <i>ClearErrors</i>          | 0x0111 | ---                                                                                                                                                          | Clears all error flags in the status                                               |
| <i>SetDioFunction</i>       | 0x0112 | Dio(7:0)<br>Func(3:0)<br>pull_drive(3:0)                                                                                                                     | Configures the function of the DIO                                                 |
| <i>SetDioRfSwitchConfig</i> | 0x0113 | Dio(7:0)<br>tx_hf, rx_hf, tx_lf, rx_lf,<br>standby                                                                                                           | Sets up the DIO output configuration for RF switch function                        |
| <i>ClearFifoIrqFlags</i>    | 0x0114 | RxFifoFlagsToClear(7:0)<br>TxFifoFlagsToClear(7:0)                                                                                                           | Clears specific FIFO IRQ flags                                                     |
| <i>SetDioIrqConfig</i>      | 0x0115 | Dio(7:0)<br>Irq(31:0)                                                                                                                                        | Configures IRQs which assert DIO pin                                               |
| <i>ClearIrq</i>             | 0x0116 | irqs_to_clear(31:0)                                                                                                                                          | Clears pending IRQs                                                                |
| <i>GetAndClearIrqStatus</i> | 0x0117 | ---                                                                                                                                                          | Reads out the pending IRQs and clear all pending IRQs                              |
| <i>ConfigLfClock</i>        | 0x0118 | lf_clock(1:0)                                                                                                                                                | Configures the used LF clock                                                       |
| <i>ConfigClkOutputs</i>     | 0x0119 | hf_clk_out_scaling(3:0)                                                                                                                                      | Configures HF clock configuration for DIO output                                   |
| <i>ConfigFifoIrq</i>        | 0x011A | rx_fifo_irq_enable(7:0)<br>tx_fifo_irq_enable(7:0)<br>rx_high_threshold(15:0)<br>tx_low_threshold(15:0)<br>rx_low_threshold(15:0)<br>tx_high_threshold(15:0) | Configures which FIFO status flags generate Fifolrqs and sets the level thresholds |
| <i>GetFifoIrqFlags</i>      | 0x011B | ---                                                                                                                                                          | Reads out the FIFO flags which generated Fifolrqs                                  |
| <i>GetRxFifoLevel</i>       | 0x011C | ---                                                                                                                                                          | Reads out the Rx FIFO level in bytes                                               |
| <i>GetTxFifoLevel</i>       | 0x011D | ---                                                                                                                                                          | Reads out the Tx FIFO level in bytes                                               |
| <i>ClearRxFifo</i>          | 0x011E | ---                                                                                                                                                          | Clears the Rx FIFO                                                                 |
| <i>ClearTxFifo</i>          | 0x011F | ---                                                                                                                                                          | Clears the Tx FIFO                                                                 |
| <i>SetTcxoMode</i>          | 0x0120 | tune(7:0)<br>start_time(31:0)                                                                                                                                | Configures the chip for a connected TCXO                                           |

Table 5-6: System Configuration Commands (Sheet 2 of 2)

| Command                         | Opcode | Parameters                                           | Description                                                                           |
|---------------------------------|--------|------------------------------------------------------|---------------------------------------------------------------------------------------|
| <i>SetRegMode</i>               | 0x0121 | simo_usage(7:0)                                      | Configures if SIMO may be used for XOSC, FS, Rx or Tx modes                           |
| <i>Calibrate</i>                | 0x0122 | blocks_to_calibrate(7:0)                             | Perform calibrations                                                                  |
| <i>CalibFE</i>                  | 0x0123 | Freq1(15:0)<br>Freq2(15:0)<br>Freq3(15:0)            | Performs front end calibrations (Image and ADC Offset (Self Reception Cancellation))  |
| <i>GetVbat</i>                  | 0x0124 | rfu(3:0)<br>Format<br>Resolution(2:0)                | Measures and returns raw supply measurement, or battery voltage                       |
| <i>GetTemp</i>                  | 0x0125 | rfu(1:0)<br>Source(1:0)<br>Format<br>Resolution(2:0) | Measures and returns raw temperature measurement, or the temperature in °C            |
| <i>GetRandomNumber</i>          | 0x0126 | Source(1:0)                                          | Returns a 32-bit random number                                                        |
| <i>SetSleep</i>                 | 0x0127 | sleep_config(7:0)<br>sleep_time(31:0)                | Puts device in Sleep Mode, with clock, data and time options                          |
| <i>SetStandby</i>               | 0x0128 | standby_mode(7:0)                                    | Puts device in standby Mode (XOSC or RC)                                              |
| <i>SetFs</i>                    | 0x0129 | ---                                                  | Puts device in Frequency Synthesis Mode                                               |
| <i>SetAdditionalRegToRetain</i> | 0x012A | slot(7:0)<br>addr(23:0)                              | Address of registers to save in the retention RAM context [slot] for some Sleep modes |
| <i>GetAndClearFifoIrqFlags</i>  | 0x012E | ---                                                  | Reads out and clears the FIFO flags                                                   |
| <i>SetLbdCfg</i>                | 0x0130 | Trim(2:0)<br>Enable(4:0)                             | Enables/disables the low battery detection and setup the trigger voltage              |
| <i>SetXoscCpTrim</i>            | 0x0131 | xta(5:0)<br>xtb(5:0)<br>additional_start_time (7:0)  | Configures XOSC foot capacitor trim                                                   |
| <i>SetTempCompCfg</i>           | 0x0132 | ntc<br>comp_mode(1:0)                                | Configures the temperature compensation for Tx mode                                   |
| <i>SetNtcParams</i>             | 0x0133 | ntc_r_ratio(9:0)<br>ntc_beta(11:0)<br>delay(7:0)     | Configures the NTC parameters                                                         |

## 5.6.4 Radio Configuration / Status Operations (LR20xx except FLRC Commands)

### 5.6.4.1 Common Radio Commands (LR20xx)

Table 5-7: Common Radio Commands (Sheet 1 of 2)

| Command                      | Opcode | Parameters                                                                                       | Description                                                                |
|------------------------------|--------|--------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------|
| <i>SetRfFrequency</i>        | 0x0200 | rf_freq(31:0)                                                                                    | Sets RF frequency in Hz                                                    |
| <i>SetRxPath</i>             | 0x0201 | rx_path<br>rx_boost(2:0)                                                                         | Selects Rx path and boost level.                                           |
| <i>SetPaConfig</i>           | 0x0202 | pa_sel<br>pa_lf_mode(1:0)<br>pa_lf_duty_cycle(3:0)<br>pa_lf_slices(3:0)<br>pa_hf_duty_cycle(4:0) | Configures PA setting                                                      |
| <i>SetTxParams</i>           | 0x0203 | tx_power(7:0)<br>ramp_time(7:0)                                                                  | Sets Tx power and ramp time                                                |
| <i>SetRssiCalibration</i>    | 0x0205 | rx_path_hf<br>rx_path_lf<br>81 [gain(9:0),nf(7:0)]                                               | Sets RSSI calibration table                                                |
| <i>SetRxTxFallbackMode</i>   | 0x0206 | fallback_mode(3:0)                                                                               | Configures the fall-back mode after Rx or Tx operation                     |
| <i>SetPacketType</i>         | 0x0207 | PacketType(7:0)                                                                                  | Sets type of packet (LoRa/GFSK,...)                                        |
| <i>GetPacketType</i>         | 0x0208 | ---                                                                                              | Returns the current packet type of the radio                               |
| <i>StopTimeoutOnPreamble</i> | 0x0209 | stop_on_preamble(7:0)                                                                            | Stops Rx time-out on Syncword/Header (default) or preamble detection       |
| <i>ResetRxStats</i>          | 0x020A | ---                                                                                              | Resets the Rx statistics                                                   |
| <i>GetRssiInst</i>           | 0x020B | ---                                                                                              | Gets instantaneous RSSI                                                    |
| <i>SetRx</i>                 | 0x020C | rx_timeout(23:0)                                                                                 | Sets device into Rx mode                                                   |
| <i>SetTx</i>                 | 0x020D | tx_timeout(23:0)                                                                                 | Sets device into Tx mode                                                   |
| <i>SetTxTestMode</i>         | 0x020E | mode(7:0)                                                                                        | Sets device into Tx test mode with infinite CW, infinite preamble or PRBS9 |
| <i>SelPa</i>                 | 0x020F | pa_sel                                                                                           | Selects which PA to use (LF or HF)                                         |
| <i>SetRxDutyCycle</i>        | 0x0210 | rx_max_time(23:0)<br>cycle_time(23:0)<br>Mode(3:0)                                               | Starts Rx Duty Cycle mode                                                  |



Table 5-7: Common Radio Commands (Sheet 2 of 2)

| Command                      | Opcode | Parameters                                                                 | Description                                                                                   |
|------------------------------|--------|----------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|
| <i>SetAutoRxTx</i>           | 0x0211 | clear<br>Mode(7:0)<br>Timeout(23:0)<br>Delay(31:0)                         | Activates or deactivates the auto Tx/Auto Rx mode                                             |
| <i>GetRxPktLength</i>        | 0x0212 | ---                                                                        | Gets length of last received packet                                                           |
| <i>SetDefaultRxTxTimeout</i> | 0x0215 | rx_timeout(23:0)<br>tx_timeout(23:0)                                       | Sets default Rx and Tx timeout                                                                |
| <i>SetTimestampSource</i>    | 0x0216 | Index(1:0)<br>Source(3:0)                                                  | Configures the time-stamp event source                                                        |
| <i>GetTimestampValue</i>     | 0x0217 | Index(1:0)                                                                 | Gets the delay in HF clock tick between the event and the SPI NSS falling edge of the request |
| <i>SetCca</i>                | 0x0218 | Duration(23:0)<br>GainStep(7:0)                                            | Sets the radio into Rx mode for Clear Channel Assessment (CCA) measurement                    |
| <i>GetCcaResult</i>          | 0x0219 | ---                                                                        | Gets the SNR statistics for the CCA                                                           |
| <i>SetAgcGainManual</i>      | 0x021A | GainStep(3:0)                                                              | Sets the manual gain of the Rx chain (HF or LF)                                               |
| <i>SetCadParams</i>          | 0x021B | cad_timeout(23:0)<br>threshold(7:0)<br>exit_mode(7:0)<br>trx_timeout(23:0) | Sets the parameters for Channel Activity Detection for radio activity except LoRa             |
| <i>SetCad</i>                | 0x021C | --                                                                         | Sets device into Rx CAD mode for all packet types except LoRa                                 |

## 5.6.4.2 LoRa Packet Radio Commands (LR20xx)

Table 5-8: LoRa Packet Radio Commands

| Command                        | Opcode | Parameters                                                                                               | Description                                                                                                   |
|--------------------------------|--------|----------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
| <i>SetLoraModulationParams</i> | 0x0220 | sf(3:0)<br>bw(3:0)<br>cr(3:0)<br>ldro(1:0)                                                               | Sets LoRa modulation parameters                                                                               |
| <i>SetLoraPacketParams</i>     | 0x0221 | pbl_len_tx(15:0)<br>payload_len(7:0)<br>header_type<br>Crc<br>invert_iq                                  | Sets packet parameters for LoRa packets                                                                       |
| <i>SetLoRaSynchTimeout</i>     | 0x0222 | nb_symbols(7:0)<br>Format(7:0)                                                                           | Configures LoRa modem to search for a detect for N symbols. N can be given as number, or as mantissa/exponent |
| <i>SetLoraSyncword</i>         | 0x0223 | Syncword(7:0)                                                                                            | Sets the LoRa Syncword                                                                                        |
| <i>SetLoraSideDetConfig</i>    | 0x0224 | Sf1(3:0), ldro1(1:0), invert_iq1<br>Sf2(3:0), ldro2(1:0), invert_iq2<br>Sf3(3:0), ldro3(1:0), invert_iq3 | Sets the LoRa multi-SF Rx configuration                                                                       |
| <i>SetLoraSideDetSyncword</i>  | 0x0225 | Syncword1(7:0)<br>Syncword2(7:0)<br>Syncword3(7:0)                                                       | Sets the LoRa multi-SF Syncwords                                                                              |
| <i>SetLoraCadParams</i>        | 0x0227 | nb_symbols(7:0)<br>pbl_any<br>pnr_delta(3:0)<br>exit_mode(7:0)<br>timeout(23:0)<br>det_peak(7:0)         | Configures LoRa CAD mode                                                                                      |
| <i>SetLoraCAD</i>              | 0x0228 | ---                                                                                                      | Sets device into Rx CAD mode (LoRa)                                                                           |
| <i>GetLoraRxStats</i>          | 0x0229 | pkt_rx(15:0)<br>pkt_crc_error(15:0)<br>header_crc_error(15:0)<br>false_synch(15:0)                       | Gets the Rx statistics for LoRa packets                                                                       |
| <i>GetLoraPacketStatus</i>     | 0x022A | ---                                                                                                      | Gets Rx packet status informations for LoRa packets                                                           |
| <i>SetLoraAddress</i>          | 0x022B | addr_comp_len(3:0)<br>addr_comp_pos(3:0)<br>addr(7:0)                                                    | Sets Address for address filtering                                                                            |
| <i>SetLoraHopping</i>          | 0x022C | hop_ctrl(1:0)<br>hop_period(12:0)<br>freq_hopx[]                                                         | Configures the intra-packet hopping in LoRa mode                                                              |
| <i>SetLoraSideDetCad</i>       | 0x021E | pnr_delta1(3:0), det_peak1(7:0),<br>pnr_delta2(3:0), det_peak2(7:0),<br>pnr_delta3(3:0), det_peak3(7:0)  | Configures the CAD on the side detectors                                                                      |

## 5.6.4.3 FSK Packet Radio Commands (LR20xx)

Table 5-9: FSK Packet Radio Commands

| Command                       | Opcode | Parameters                                                                                                                                                 | Description                                                                             |
|-------------------------------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|
| <i>SetFskModulationParams</i> | 0x0240 | BitRate(31:0)<br>pulse_shape(7:0)<br>rx_bw(7:0)<br>fdev_hz(23:0)                                                                                           | Configures the modulation parameters for FSK packets                                    |
| <i>SetFskPacketParams</i>     | 0x0241 | pbl_len_tx(15:0)<br>pbl_detect(7:0)<br>long_preamble_mode<br>pld_lenUnit<br>addr_comp(1:0)<br>pkt_format(1:0)<br>pld_len(15:0)<br>Crc(3:0)<br>dc_free(3:0) | Configures the packets parameters for FSK packets                                       |
| <i>SetFskWhiteningParams</i>  | 0x0242 | whiten_type(3:0)<br>Init(11:0)                                                                                                                             | Configures the whitening parameters for FSK packets, SX126x/LR11xx or SX128x compatible |
| <i>SetFskCrcParams</i>        | 0x0243 | polynom(31:0)<br>Init(31:0)                                                                                                                                | Configures the CRC parameters for FSK packets                                           |
| <i>SetFskSyncword</i>         | 0x0244 | Syncword(63:0)<br>bit_order<br>nb_bits(6:0)                                                                                                                | Configures the Syncword for FSK packets                                                 |
| <i>SetFskAddress</i>          | 0x0245 | addr_node(7:0)<br>addr_bcast(7:0)                                                                                                                          | Configures the addresses for filtering for FSK packets                                  |
| <i>GetFskRxStats</i>          | 0x0246 | ---                                                                                                                                                        | Gets the Rx statistics for FSK packets                                                  |
| <i>GetFskPacketStatus</i>     | 0x0247 | ---                                                                                                                                                        | Gets Rx packet status informations for FSK packets                                      |

#### 5.6.4.4 FLRC Packet Radio Commands (LR2021)

Table 5-10: FLRC Packet Radio Commands

| Command                        | Opcode | Parameters                                                                                                 | Description                                           |
|--------------------------------|--------|------------------------------------------------------------------------------------------------------------|-------------------------------------------------------|
| <i>SetFlrcModulationParams</i> | 0x0248 | bitrate_bw(7:0)<br>cr(3:0)<br>pulse_shape(3:0)                                                             | Configures the modulation parameters for FLRC packets |
| <i>SetFlrcPacketParams</i>     | 0x0249 | agc_pbl_len(3:0)<br>sync_len<br>sync_tx(1:0)<br>sync_match(2:0)<br>pkt_format<br>Crc(1:0)<br>pld_len(15:0) | Configures the packets parameters for FLRC packets    |
| <i>GetFlrcRxStats</i>          | 0x024A | ---                                                                                                        | Gets the Rx statistics for FLRC packets               |
| <i>GetFlrcPacketStatus</i>     | 0x024B | ---                                                                                                        | Gets Rx packet status informations for FLRC packets   |
| <i>SetFlrcSyncword</i>         | 0x024C | sw_num(7:0)<br>Syncword(31:0)                                                                              | Sets the Syncword for FLRC                            |

#### 5.6.4.5 BPSK Packet Radio Commands (LR20xx)

Table 5-11: BPSK Packet Radio Commands

| Command                        | Opcode | Parameters                                                            | Description                                           |
|--------------------------------|--------|-----------------------------------------------------------------------|-------------------------------------------------------|
| <i>SetBpskModulationParams</i> | 0x0250 | BitRate(31:0)<br>pulse_shape(3:0)<br>diff_mode<br>diff_mode_init(1:0) | Configures the modulation parameters for BPSK packets |
| <i>SetBpskPacketParams</i>     | 0x0251 | pld_len(7:0)<br>Mode(1:0)<br>bpsk_s_msg<br>bpsk_s_rank(1:0)           | Configures the packets parameters for BPSK_S packets  |

## 5.6.4.6 LR-FHSS Packet Radio Commands (LR20xx)

Table 5-12: LR-FHSS Packet Radio Commands

| Command                        | Opcode | Parameters                                                                                                                                 | Description                                                         |
|--------------------------------|--------|--------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------|
| <i>LrFhssBuildFrame</i>        | 0x0256 | sync_header_cnt(3:0)<br>Cr(3:0)<br>mod_type(3:0)<br>Grid(3:0)<br>Hopping(3:0)<br>Bw(3:0)<br>Sequence(8:0)<br>Offset(7:0)<br>Payload[0:255] | Encodes the given payload and configures the internal hopping table |
| <i>LrFhssSetSyncword</i>       | 0x0257 | Syncword(31:0)                                                                                                                             | Sets the LR-FHSS Syncword. Reset value is {0x2C, 0x0F, 0x79, 0x95}  |
| <i>ReadLrFhssHoppingTable</i>  | 0x0258 | ---                                                                                                                                        | Reads the hopping table                                             |
| <i>WriteLrFhssHoppingTable</i> | 0x0259 | hopping_table[]                                                                                                                            | Writes the hopping table                                            |

### 5.6.4.7 Bluetooth Low Energy Packet Radio Commands (LR2021/LR2022)

Table 5-13: Bluetooth Low Energy Packet Radio Commands

| Command                       | Opcode | Parameters                                                                             | Description                                                   |
|-------------------------------|--------|----------------------------------------------------------------------------------------|---------------------------------------------------------------|
| <i>SetBleModulationParams</i> | 0x0260 | mode(7:0)<br>rx_bw(7:0)                                                                | Configures the modulation parameters for Bluetooth LE packets |
| <i>SetBleChannelParams</i>    | 0x0261 | crc_in_fifo<br>channel_type(3:0)<br>whit_init(7:0)<br>crc_init(23:0)<br>Syncword(31:0) | Sets the Bluetooth LE channel dependent parameters            |
| <i>SetBleTx</i>               | 0x0262 | pdu_len(7:0)                                                                           | Sets PDU length and sends a Bluetooth LE Low Energy packet    |
| <i>GetBleRxStats</i>          | 0x0264 | ---                                                                                    | Gets the Rx statistics for Bluetooth LE packets               |
| <i>GetBlePacketStatus</i>     | 0x0265 | ---                                                                                    | Gets Rx packet status informations for Bluetooth LE packets   |
| <i>SetBleTxPduLen</i>         | 0x0266 | pdu_len(7:0)                                                                           | Sets PDU length for Tx                                        |

### 5.6.4.8 WM-Bus Packet Radio Commands (LR20xx)

Table 5-14: WM-Bus Packet Radio Commands

| Command                         | Opcode | Parameters                                                                                                                | Description                                           |
|---------------------------------|--------|---------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------|
| <i>SetWmbusParams</i>           | 0x026A | Mode(7:0)<br>rx_bw(7:0)<br>pkt_formatTx(7:0)<br>addr_comp(7:0)<br>pld_len(7:0)<br>pbl_len_tx(15:0)<br>pbl_len_detect(7:0) | Configures the WM-Bus mode                            |
| <i>GetWmbusRxStats</i>          | 0x026C | ---                                                                                                                       | Gets the Rx statistics for WM-Bus packets             |
| <i>GetWmbusPacketStatus</i>     | 0x026D | ---                                                                                                                       | Gets Rx packet status informations for WM-Bus packets |
| <i>SetWmbusFilteringAddress</i> | 0x026E | Address(47:0)                                                                                                             | Sets the WM-Bus address for Rx filtering              |

## 5.6.4.9 Wi-SUN Packet Radio Commands (LR20xx)

Table 5-15: Wi-SUN Packet Radio Commands

| Command                     | Opcode | Parameters                                                                                                                 | Description                                                                              |
|-----------------------------|--------|----------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
| <i>SetWisunMode</i>         | 0x0270 | Mode(7:0)<br>rx_bw(7:0)                                                                                                    | Configures the Wi-SUN mode (1a, 1b, 2a, 2b, 3, 4a, 4b, 5)                                |
| <i>SetWisunPacketParams</i> | 0x0271 | fcs_tx<br>Whitening<br>crc_on<br>mode_switch_tx<br>fec_tx(1:0)<br>frame_len_tx(15:0)<br>pbl_len_tx(7:0)<br>pbl_detect(7:0) | Configures the Wi-SUN packet parameters                                                  |
| <i>GetWisunRxStats</i>      | 0x0272 | ---                                                                                                                        | Gets the Rx statistics for Wi-SUN packets                                                |
| <i>GetWisunPacketStatus</i> | 0x0273 | ---                                                                                                                        | Gets Rx packet status informations for Wi-SUN packets                                    |
| <i>SetWisunPacketLen</i>    | 0x0274 | frame_len_tx(15:0)                                                                                                         | Sets length of frame for Tx for normal packets, or header value for mode_switch packets. |

## 5.6.4.10 Z-Wave Packet Radio Commands (LR2021)

Table 5-16: Z-Wave Packet Radio Commands

| Command                         | Opcode | Parameters                                                                                                                                                                                                                                                                                                            | Description                                         |
|---------------------------------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|
| <i>SetZwaveParams</i>           | 0x0297 | Mode(7:0)<br>rx_bw(7:0)<br>addr_comp(7:0)<br>pId_len(7:0)<br>pbl_len_tx(15:0)<br>pbl_len_detect(7:0)<br>Options(7:0)                                                                                                                                                                                                  | Sets all parameters for Z-Wave packets              |
| <i>SetZwaveHomelIdFiltering</i> | 0x0298 | home_id(31:0)                                                                                                                                                                                                                                                                                                         | Sets the HomeID address to use as a filter in Rx    |
| <i>GetZwaveRxStats</i>          | 0x0299 | --                                                                                                                                                                                                                                                                                                                    | Gets the Rx statistics of Z-Wave packets            |
| <i>GetZwavePacketStatus</i>     | 0x029A | --                                                                                                                                                                                                                                                                                                                    | Gets Rx packet status information of Z-Wave packets |
| <i>SetZwaveBeamFiltering</i>    | 0x029B | beam_tag(7:0)<br>addr_len<br>node_id(11:0)<br>id_hash(7:0)                                                                                                                                                                                                                                                            | Filters incoming beam frames                        |
| <i>SetZwaveScanConfig</i>       | 0x029C | num_ch(3:0), det4, det3, det2, det1,<br>bitrate_ch4(1:0), bitrate_ch3(1:0),<br>bitrate_ch2(1:0), bitrate_ch1(1:0),<br>addr_comp(7:0), options(7:0),<br>rf_freq_ch1(31:0),<br>timeout_ch1(7:0), rf_freq_ch2(31:0),<br>timeout_ch2(7:0), rf_freq_ch3(31:0),<br>timeout_ch3(7:0), rf_freq_ch4(31:0),<br>timeout_ch4(7:0) | Configures the scan mode for the Z-Wave             |
| <i>SetZwaveScan</i>             | 0x029D | --                                                                                                                                                                                                                                                                                                                    | Enters Z-Wave Rx scan mode                          |



### 5.6.4.11 OQPSK\_15\_4 Packet Radio Commands (LR2021)

Table 5-17: OQPSK\_15\_4 Packet Radio Commands

| Command                     | Opcode | Parameters                                                                                                      | Description                                                  |
|-----------------------------|--------|-----------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------|
| <i>SetOqpskParams</i>       | 0x029F | Mode(7:0)<br>rx_bw(7:0)<br>pld_len(7:0)<br>pbl_len_tx(15:0)<br>address_on<br>bypass_rx_length_check<br>fcs_mode | Set parameters for OQPSK_15_4 packets                        |
| <i>GetOqpskRxStats</i>      | 0x02A0 | --                                                                                                              | Get the Rx statistics for OQPSK_15_4 packets                 |
| <i>GetOqpskPacketStatus</i> | 0x02A1 | --                                                                                                              | Get the Rx packet status informations for OQPSK_15_4 packets |
| <i>SetOqpskPacketLen</i>    | 0x02A2 | len(7:0)                                                                                                        | Sets new length for OQPSK_15_4 packets                       |
| <i>SetOqpskAddress</i>      | 0x02A3 | LongDestAddr(63:0)<br>ShortDestAddr(15:0)<br>PanId(15:0)<br>TransId(7:0)                                        | Set address parameters for OQPSK_15_4 packets                |

### 5.6.4.12 RTToF Packet Radio Commands (LR20xx)

Table 5-18: RTToF Packet Radio Commands

| Command                    | Opcode | Parameters                     | Description                                                    |
|----------------------------|--------|--------------------------------|----------------------------------------------------------------|
| <i>SetRangingAddr</i>      | 0x0278 | Addr(31:0)<br>CheckLength(7:0) | Sets the ranging Id for this device (used in the Subordinates) |
| <i>SetRangingReqAddr</i>   | 0x0279 | ReqAddr(31:0)                  | Sets the ranging Id for the requests (used in the Manager)     |
| <i>GetRangingResult</i>    | 0x027A | Type(7:0)                      | Gets the ranging result, return values based on type           |
| <i>SetRangingTxRxDelay</i> | 0x027B | Delay(31:0)                    | Sets the Tx->Rx delay for the ranging calibration              |
| <i>SetRangingParams</i>    | 0x027C | spy_mode<br>nb_symbols(5:0)    | Sets the ranging specific parameters                           |
| <i>GetRangingStats</i>     | 0x027D | ---                            | Returns ranging counters                                       |

RTToF is a functionality that can measure device to device distance and uses the ranging commands listed above.

## 5.6.4.13 OOK Packet Radio Commands (LR20xx)

Table 5-19: OOK Packet Radio Commands

| Command                       | Opcode | Parameters                                                                                                            | Description                                           |
|-------------------------------|--------|-----------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------|
| <i>SetOokModulationParams</i> | 0x0281 | BitRate(31:0)<br>pulse_shape(7:0)<br>rx_bw(7:0)<br>Depth(7:0)                                                         | Sets the OOK modulation parameters                    |
| <i>SetOokPacketParams</i>     | 0x0282 | pre_len_tx(15:0)<br>addr_comp(1:0)<br>pkt_format(1:0)<br>pld_len(15:0)<br>Crc(3:0)<br>Manchester(3:0)                 | Sets the OOK packet parameters                        |
| <i>SetOokCrcParams</i>        | 0x0283 | Polynom(31:0)<br>Init(31:0)                                                                                           | Sets the OOK CRC parameters                           |
| <i>SetOokSyncword</i>         | 0x0284 | Syncword(31:0)<br>bit_order<br>nb_bits(6:0)                                                                           | Sets the OOK Syncword                                 |
| <i>SetOokAddress</i>          | 0x0285 | addr_node(7:0)<br>addr_bcast(7:0)                                                                                     | Sets the OOK address                                  |
| <i>GetOokRxStats</i>          | 0x0286 | ---                                                                                                                   | Get the Rx statistics for OOK packets                 |
| <i>GetOokPacketStatus</i>     | 0x0287 | ---                                                                                                                   | Get the Rx packet status informations for OOK packets |
| <i>SetOokDetector</i>         | 0x0288 | preamble_pattern(15:0)<br>pattern_length(3:0)<br>pattern_num_repeats(4:0)<br>sw_is_raw<br>sfd_kind<br>sfd_length(3:0) | Configures the OOK detection                          |
| <i>SetOokwhiteningParams</i>  | 0x0289 | bit_idx(3:0)<br>Polynom(11:0)<br>Init(11:0)                                                                           | Configure the whitening for OOK packets               |

## 5.7 Host Interrupts (LR20xx)

The LR2021/LR2022/LR2012 uses IRQ lines configured on the DIOs to signal interrupts to the host. These IRQ lines serve as a means of communication between the LR2021/LR2022/LR2012 and the host controller, notifying the host of specific events or conditions that require attention.

When an interrupt occurs, it is stored in a 32-bit IRQ status register, providing detailed information about the specific events that triggered the interrupt. The host can read this IRQ status through the serial interface to determine the cause of the interrupt.

To acknowledge and clear the interrupt, the host can use the [ClearIrq](#) command with the respective bit set to 1. This action resets the interrupt flag, allowing the LR2021/LR2022/LR2012 to generate new interrupts as needed.

**Table 5-20: Host Interrupts (Sheet 1 of 2)**

| Bit | IRQ                              | Description                                                                                                                                                                               | Mode    | Packet Type    |
|-----|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|----------------|
| 0   | RxFifo                           | Rx FIFO threshold reached                                                                                                                                                                 | Rx      | All            |
| 1   | TxFifo                           | Tx FIFO threshold reached                                                                                                                                                                 | Tx      | All            |
| 2   | RngReqVld                        | Valid ranging request received (Subordinate)                                                                                                                                              | Rx      | RTToF          |
| 3   | TxTimestamp                      | IRQ for time-stamping end of packet Tx, without dependent delay of mode switching. Only to be used for time-stamping, not for changing mode or re-configuring the device.                 | TX      | All            |
| 4   | RxTimestamp                      | IRQ for time-stamping end of packet Rx, without dependent delay of demodulation or mode switching. Only to be used for time-stamping, not for changing mode or re-configuring the device. | Rx      | All            |
| 5   | PreambleDetected                 | Preamble detected                                                                                                                                                                         | Rx      | All            |
| 6   | LoRaHeaderValid<br>SyncwordValid | LoRa header detected<br>Valid Syncword                                                                                                                                                    | Rx      | LoRa<br>Others |
| 7   | CadDetected                      | Channel activity detected                                                                                                                                                                 | Rx, CAD | LoRa, FSK      |
| 8   | LoRaHdrTimestamp                 | LoRa header precise time-stamp (explicit mode).<br>In implicit mode asserts after 8 symbols of the payload.                                                                               | Rx      | LoRa           |
| 9   | LoRaHeaderErr                    | LoRa header CRC error                                                                                                                                                                     | Rx      | LoRa           |
| 10  | LowBattery                       | Low Battery detector                                                                                                                                                                      | All     | All            |
| 11  | PA OCP/OVP                       | PA OCP/OVP has triggered                                                                                                                                                                  | Tx      | All            |
| 12  | LoRaTxRxHop                      | IRQ for LoRa intra-packet hopping                                                                                                                                                         | Rx/Tx   | LoRa           |
| 13  | SyncFail                         | Syncword match failed after detection                                                                                                                                                     | Rx      | FSK            |
| 14  | LoRaSymbolEnd                    | End of LoRa symbol                                                                                                                                                                        | Rx      | LoRa           |
| 15  | LoRaTimestampStat                | New statistics available in time-stamp register                                                                                                                                           | Rx      | LoRa           |
| 16  | Error                            | An error other than a command error occurred (See <a href="#">GetErrors</a> command)                                                                                                      | All     | All            |
| 17  | CmdError                         | There was a host command fail/error                                                                                                                                                       | All     | -              |

Table 5-20: Host Interrupts (Sheet 2 of 2)

| Bit | IRQ          | Description                                                    | Mode    | Packet Type         |
|-----|--------------|----------------------------------------------------------------|---------|---------------------|
| 18  | RxDone       | Packet reception completed                                     | Rx      | All                 |
| 19  | TxDone       | Packet transmission completed                                  | Tx      | All                 |
| 20  | CadDone      | Channel activity detection finished                            | Rx, CAD | LoRa, FSK           |
| 21  | Timeout      | Rx or Tx timeout                                               | Rx, Tx  | All                 |
| 22  | CrcError     | Packet was received with a wrong Crc                           | Rx      | All                 |
| 23  | LenError     | Packet was received with a length error                        | Rx      | FSK<br>FLRC<br>LoRa |
| 24  | AddrError    | Packet was received with a wrong address match                 | Rx      | FSK<br>LoRa         |
| 25  | FHSS         | Ramp-up for intra-packet hopping occurred                      | Tx      | FHSS                |
| 26  | InterPacket1 | Host can load new frequencies table                            | Tx      | LoRa                |
| 27  | InterPacket2 | Host can load new payload                                      | Tx      | LoRa                |
| 28  | RngRespDone  | Subordinate sent its ranging response                          | Rx      | RTToF               |
| 29  | RngReqDis    | Ranging request discarded (no address match)                   | Rx      | RTToF               |
| 30  | RngExchVld   | Manager received a valid ranging response from the Subordinate | Tx      | RTToF               |
| 31  | RngTimeout   | Manager did not receive a response from the Subordinate        | Tx      | RTToF               |

The LR2021/LR2022/LR2012 offers the host controller the flexibility to assign IRQs to specific pins using the built-in command [SetDioIrqConfig](#). This allows the host to configure the DIO pins to trigger specific interrupts based on their requirements and application logic.

Additionally, the LR2021/LR2022/LR2012 provides a dedicated command [GetAndClearIrqStatus](#) to retrieve status information on pending IRQs and clear all pending interrupts at once.

Detailed information about these commands and their functionalities can be found in the next chapter.

## 6. Chip Control Commands (LR20xx)

### 6.1 Direct Radio FIFO Read/Write

#### 6.1.1 ReadRadioRxFifo

The *ReadRadioRxFifo* command allows direct data retrieval from the radio Rx FIFO. It operates when the Opcode is 0x0001, ensuring efficient and low-latency data transfer. The maximum data that can be read depends on the FIFO level (can be read with the [GetRxFifoLevel](#) command), the FIFO can contain 256 bytes maximum.

**Table 6-1: ReadRadioRxFifo Command**

| Byte           | 0          | 1         | 2         | 3         | 4         | 5         | 6         | 7         | ...       |
|----------------|------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| Data from host | 0x00       | 0x01      | 0x00      | 0x00      | 0x00      | 0x00      | 0x00      | 0x00      | 0x00      |
| Data to host   | Stat(15:8) | Stat(7:0) | Data(7:0) | Data(7:0) | Data(7:0) | Data(7:0) | Data(7:0) | Data(7:0) | Data(7:0) |

#### 6.1.2 WriteRadioTxFifo

The *WriteRadioTxFifo* command enables direct data writing to the radio Tx FIFO. This transfer occurs once the Opcode 0x0002 is identified. The maximum data that can be read depends on the FIFO level (can be read with the [GetTxFifoLevel](#) command), the FIFO can contain 256 bytes maximum.

**Table 6-2: WriteRadioTxFifo Command**

| Byte           | 0          | 1         | 2                | 3                | 4               | 5              | 6         | ...       |
|----------------|------------|-----------|------------------|------------------|-----------------|----------------|-----------|-----------|
| Data from host | 0x00       | 0x02      | Data(7:0)        | Data(7:0)        | Data(7:0)       | Data(7:0)      | Data(7:0) | Data(7:0) |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) | IrqStatus(23:16) | IrqStatus(15:8) | IrqStatus(7:0) | 0x00      | 0x00      |

## 6.2 Register / Memory Access Operations

### 6.2.1 WriteRegMem32

The *WriteRegMem32* command writes a block of 32-bit words to the register/memory space, starting from a specified address. The command uses auto-incrementing of the address after each data word is written, ensuring contiguous storage of data in the register/memory locations. The maximum value of N, representing the number of data words that can be written in a single block, is limited to 32.

**Table 6-3: WriteRegMem32 Command**

| Byte           | 0          | 1            | 2                 | 3                 | 4                | 5               | 6            | 7           |
|----------------|------------|--------------|-------------------|-------------------|------------------|-----------------|--------------|-------------|
| Data from host | 0x01       | 0x04         | Addr(23:16)       | Addr(15:8)        | Addr(7:0)        | Data1(31:24)    | Data1(23:16) | Data1(15:8) |
| Data to host   | Stat(15:8) | Stat(7:0)    | IrqStatus (31:24) | IrqStatus (23:16) | IrqStatus (15:8) | IrqStatus (7:0) | 0x00         | 0x00        |
| Byte           | 8          | 9            | ...               | N*4+4             |                  |                 |              |             |
| Data from host | Data1(7:0) | Data2(31:24) | ...               | DataN (7:0)       |                  |                 |              |             |
| Data to host   | 0x00       | 0x00         | ...               | 0x00              |                  |                 |              |             |

### 6.2.2 WriteRegMemMask32

The *WriteRegMemMask32* command reads, modifies and writes on a single 32-bit word in the register/memory space, specifically targeting the masked bits (Mask bits = 1). This command modifies only the selected bits in the word, while leaving the other bits unchanged. The operation takes place at the specified address in the register/memory space.

**Table 6-4: WriteRegMemMask32 Command**

| Byte           | 0          | 1           | 2                 | 3                 | 4                | 5               | 6            | 7           |
|----------------|------------|-------------|-------------------|-------------------|------------------|-----------------|--------------|-------------|
| Data from host | 0x01       | 0x05        | Addr (23:16)      | Addr (15:8)       | Addr (7:0)       | Mask (31:24)    | Mask (23:16) | Mask (15:8) |
| Data to host   | Stat(15:8) | Stat(7:0)   | IrqStatus (31:24) | IrqStatus (23:16) | IrqStatus (15:8) | IrqStatus (7:0) | 0x00         | 0x00        |
| Byte           | 8          | 9           | 10                | 11                | 12               |                 |              |             |
| Data from host | Mask(7:0)  | Data(31:24) | Data(23:16)       | Data(15:8)        | Data (7:0)       |                 |              |             |
| Data to host   | 0x00       | 0x00        | 0x00              | 0x00              | 0x00             |                 |              |             |

## 6.2.3 ReadRegMem32

The *ReadRegMem32* command reads a block of 32-bit words from the register/memory space, starting from a specified address. The command auto-increments the address after each data word is read, ensuring the sequential retrieval of data from contiguous register locations. The value of N represents the number of words to read, with a maximum limit of 32.

**Table 6-5: ReadRegMem32 Command**

| Byte           | 0          | 1         | 2                | 3                | 4               | 5               |
|----------------|------------|-----------|------------------|------------------|-----------------|-----------------|
| Data from host | 0x01       | 0x06      | Addr (23:16)     | Addr (15:8)      | Addr (7:0)      | Len (7:0)       |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) | IrqStatus(23:16) | IrqStatus(15:8) | IrqStatus (7:0) |

**Table 6-6: ReadRegMem32 Response**

| Byte           | 0              | 1             | 2                | 3                | 4               | 5              | 6                | ... | N*4+1          |
|----------------|----------------|---------------|------------------|------------------|-----------------|----------------|------------------|-----|----------------|
| Data from host | 0x00           | 0x00          | 0x00             | 0x00             | 0x00            | 0x00           | 0x00             | ... | 0x00           |
| Data to host   | Stat<br>(15:8) | Stat<br>(7:0) | Data1<br>(31:24) | Data1<br>(23:16) | Data1<br>(15:8) | Data1<br>(7:0) | Data2<br>(31:24) | ... | DataN<br>(7:0) |

## 6.3 Chip Mode Control

### 6.3.1 SetSleep

The *SetSleep* command puts the device into Sleep mode.

**Table 6-7: SetSleep Command**

| Byte           | 0          | 1          | 2                     | 3                     | 4                     | 5                    | 6                   |
|----------------|------------|------------|-----------------------|-----------------------|-----------------------|----------------------|---------------------|
| Data from host | 0x01       | 0x27       | sleep_config<br>(7:0) | sleep_time<br>(31:24) | sleep_time<br>(23:16) | sleep_time<br>(15:8) | sleep_time<br>(7:0) |
| Data to host   | Stat(15:8) | Stat((7:0) | IrqStatus(31:24)      | IrqStatus(23:16)      | IrqStatus (15:8)      | IrqStatus(7:0)       | 0x00                |

*sleep\_config* defines the specific parameters for the Sleep mode as follows:

- ◆ Bit 0: clk\_32k\_enable - If this bit is set to 1, the internal 32 kHz RC clock continues to run during sleep.
- ◆ Bit 1: Data retention enable
  - ◆ 1: When data is retained in its current state during sleep, allowing the device to preserve data across sleep cycles.
  - ◆ 0: Data is not retained. Device performs cold start upon waking from sleep, resulting in loss of data previously stored.
- ◆ Bit 2–7: RFU

*sleep\_time* is optional and only relevant when clk\_32k\_enable = 1. It sets the sleep duration in terms of the number of LF clock cycles. The device automatically wakes up after the specified *sleep\_time* has elapsed. If *sleep\_time* = 0 while clk\_32k\_enable = 1, the device enters Sleep mode with an infinite timeout. In this scenario, the device stays in Sleep mode indefinitely until an external event triggers it to wake up. This feature can be useful if the LF\_CLK is output on a digital I/O (DIO) pin for external monitoring or synchronization with the host system.

### 6.3.2 SetStandby

The *SetStandby* command switches the device into the requested Standby mode.

**Table 6-8: SetStandby Command**

| Byte           | 0          | 1         | 2                 |
|----------------|------------|-----------|-------------------|
| Data from host | 0x01       | 0x28      | standby_mode(7:0) |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus (31:24) |

**standby\_mode** defines the standby operating mode of the chip. It offers two options:

- 0: Device enters Standby RC mode. It uses the internal RC oscillator for timekeeping and low-power operations, suitable for applications where precise timing is not critical. An external high precision clock input on DIO11 is possible using [ConfigLfClock](#).
- 1: Device enters Standby XOSC mode. The device uses the crystal or XOSC for more accurate timekeeping.

### 6.3.3 SetFs

The *SetFs* command sets the device into Frequency synthesizer (FS) mode.

**Table 6-9: SetFs Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x01       | 0x29      |
| Data to host   | Stat(15:8) | Stat(7:0) |

### 6.3.4 SetAdditionalRegToRetain

*SetAdditionalRegToRetain* specifies the address of an additional register to retain in Sleep mode. See

**Table 6-10: SetAdditionalRegToRetain Command**

| Byte           | 0          | 1         | 2                | 3                | 4               | 5              |
|----------------|------------|-----------|------------------|------------------|-----------------|----------------|
| Data from host | 0x01       | 0x2A      | slot(7:0)        | addr(23:16)      | addr(15:8)      | addr(7:0)      |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) | IrqStatus(23:16) | IrqStatus(15:8) | IrqStatus(7:0) |

**slot:** Specifies the index in the table. Allowed values are [0 to 31].

**addr:** Address of the peripheral register to be saved and restored. Address must be word-aligned.



### 6.3.5 SetRx

The *SetRx* command configures the radio in Rx mode. If the device does not receive any packet within the specified ***rx\_timeout*** duration, it automatically returns to fallback mode as defined by *SetRxTxFallbackMode* command.

**Table 6-11: SetRx Command**

| Byte           | 0          | 1         | 2                  | 3                 | 4                |
|----------------|------------|-----------|--------------------|-------------------|------------------|
| Data from Host | 0x02       | 0x0C      | rx_timeout (23:16) | rx_timeout (15:8) | rx_timeout (7:0) |
| Data to Host   | Stat(15:8) | Stat(7:0) | IrqStatus (31:24)  | IrqStatus (23:16) | IrqStatus (15:8) |

***rx\_timeout*** is expressed in periods of the 32 kHz RTC (Real-Time Clock) and sets the maximum timeout value to 512 seconds.

- Setting ***rx\_timeout*** to 0x000000 keeps the device in Rx mode until a reception occurs. After packet reception, the device automatically returns to mode defined by command *SetRxTxFallbackMode* (Standby RC, Standby XOSC or FS mode). This is often called single Rx mode.
- Setting ***rx\_timeout*** to 0xFFFFF keeps the device in Rx mode until the host sends a command to change the mode. In this mode, the device can receive multiple packets, providing a packet-done indication to the host each time a packet is received. It automatically searches for new packets. This is often called continuous mode.
- Otherwise, ***rx\_timeout*** defines a timeout period after going to Rx. If the timeout function is active, the radio stops reception at the end of the timeout period unless a preamble or a header is detected, as specified by the *StopTimeoutOnPreamble* configuration.

**Note:**

***rx\_timeout*** is optional. If not set (i.e. bits 2/3/4 not sent) the value configured with the *SetDefaultRxTxTimeout* command is used.

This command fails if the packet type does not allow Rx operations.

While in Rx mode, the BUSY signal goes low after the device is set, indicating its status. Additionally, the RF path (HF or LF) is selected using *SetRxPath* command.

If image rejection was not calibrated for the current RF frequency, the command does not fail and still goes to Rx, but error RxFreqNoCalErr is generated.

If the device is already in Rx mode, the command fails as timeout cannot be updated.

### 6.3.6 SetTx

The *SetTx* command activates the radio's Tx mode, initiating RF packet transmission, and starting the RTC with the specified **tx\_timeout** value. While in Tx mode, the BUSY signal goes low after the device is set, indicating its status.

**Table 6-12: SetTx Command**

| Byte           | 0          | 1         | 2                  | 3                 | 4                |
|----------------|------------|-----------|--------------------|-------------------|------------------|
| Data from Host | 0x02       | 0x0D      | tx_timeout (23:16) | tx_timeout (15:8) | tx_timeout (7:0) |
| Data to Host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24)   | IrqStatus(23:16)  | IrqStatus(15:8)  |

**tx\_timeout** is expressed in periods of the 32 kHz RTC (Real-Time Clock) and can have a maximum value of 512 seconds.

- ◆ 0x000000: Disables the timeout function during Tx mode. Tx timeout can be used as a safeguard in case transmission fails and TxDone interrupt never occurs
- ◆ Otherwise, **tx\_timeout** defines a timeout (express in periods of 32 kHz RTC built-in clock) period after going to Tx

**tx\_timeout** is optional. If not explicitly set, the value configured with the *SetDefaultRxTxTimeout* command is used.

For successful execution of the command, a valid packet type must be configured, and the packet type must allow Tx operations; otherwise, the command fails.

If the device is already in Tx mode the command fails as tx\_timeout cannot be updated.

If the RTC event fires before the transmission completion, a Timeout IRQ is triggered, and the transmission is stopped prematurely. Otherwise, upon the successful completion of packet transmission, a *Txdoneinterrupt* is generated.

After a Timeout IRQ or TxDone IRQ, the device returns to one of the following modes defined by the command *SetRxTxFallbackMode* (Standby RC, Standby XOSC or FS mode).

### 6.3.7 SetRxTxFallbackMode

The *SetRxTxFallbackMode* command determines the mode the device enters after completion of a packet transmission or reception.

**Table 6-13: SetRxTxFallbackMode Command**

| Byte           | 0          | 1         | 2                              |
|----------------|------------|-----------|--------------------------------|
| Data from Host | 0x02       | 0x06      | rfu(3:0)<br>fallback_mode(3:0) |
| Data to Host   | Stat(15:8) | Stat(7:0) | IrqStatus (31:24)              |

**fallback\_mode** offers the following fallback mode values:

- ◆ 0x01: Standby RC mode (default value)
- ◆ 0x02: Standby XOSC mode
- ◆ 0x03: FS mode (Frequency Synthesizer)
- ◆ Other values are RFU and should not be used

In duty cycle mode, the chip goes to the defined fallback mode as soon as a packet is received (Rxdone).

**fallback\_mode** is also used during an AutoRxTx sequence, both after first Rx/Tx and second Tx/Rx. Fallback is also used after CAD.

### 6.3.8 SetRxDutyCycle

The *SetRxDutyCycle* command facilitates periodic opening of Rx windows. During intervals between Rx window starts, the device enters Sleep mode (with retention), conserving power efficiently.

**Table 6-14: SetRxDutyCycle Command**

| Byte           | 0           | 1          | 2                   | 3                  | 4                 | 5                  | 6                 | 7                | 8                   |
|----------------|-------------|------------|---------------------|--------------------|-------------------|--------------------|-------------------|------------------|---------------------|
| Data from Host | 0x02        | 0x10       | rx_max_time (23:16) | rx_max_time (15:8) | rx_max_time (7:0) | cycle_time (23:16) | cycle_time (15:8) | cycle_time (7:0) | Mode (3:0) rfu(3:0) |
| Data to Host   | Stat (15:8) | Stat (7:0) | IrqStatus (31:24)   | IrqStatus (23:16)  | IrqStatus (15:8)  | IrqStatus (7:0)    | 0x00              | 0x00             | 0x00                |

**rx\_max\_time** is the maximum duration of an Rx window, expressed in periods of the internal 32 kHz RC oscillator. The max. delay corresponds to 512s.

**cycle\_time** specifies the duration of the period between the start of consecutive Rx windows, expressed in periods of the 32 kHz RC. The maximum period value corresponds to 512 seconds. If **cycle\_time** is lower than **rx\_max\_time** a CMD\_ERR is returned in the status of the next command.

**Mode** is optional and selects the device mode during the Rx windows.

- **Mode = 0x0:** Available for (G)FSK, FLRC, OOK, and LoRa modems (returns CMD\_FAIL for other packet types).
  - ♦ The device enters Rx and listens for an incoming RF packet for a max. period of time defined by **rx\_max\_time**.
  - ♦ Upon preamble detection, the timeout restarts with the value **rx\_max\_time + cycle\_time**.
  - ♦ If no packet is received during the Rx window, the device goes into Sleep mode with context saved (retention) for a period defined by **cycle\_time-rx\_max\_time** to have a period of **cycle\_time** between each Rx start. No RxTimeout IRQ is asserted.
  - ♦ At the end of a Sleep window, the device automatically restarts the process, entering Rx mode repeatedly until stopped by the host.
- **Mode = 0x1:** Available only for LoRa packet types (returns CMD\_FAIL for other packet types).
  - ♦ Behaves like Mode = 0 but performs a Channel Activity Detection (CAD) on wake-up with parameters defined by a preceding [SetLoraCadParams](#) command.
  - ♦ If channel activity is detected, the exit CAD mode is used with the **rx\_max\_time** as time out for either Rx(CAD-RX) or Tx(CAD-LBT). In CAD only, the Rx is not started but the device exits Rx duty cycle mode.
  - ♦ If no channel activity is detected, the device goes back to Sleep mode for **cycle\_time - rx\_max\_time** to have a period of **cycle\_time** between each Rx start.

The loop is terminated either:

- ♦ On reception of a packet during an Rx window. In this case the chip goes back to the configured fallback mode. If an autoRxTx was setup, it is activated the same way as for a normal Rx.
- ♦ The host issues a [SetStandby](#) command during the Rx window.
- ♦ Wake up by the device with a falling edge of NSS. A [SetStandby](#) command should also be sent, to avoid the race conditions in case the NSS falling edge was issued during the boot phase of the device.
- ♦ In Mode = 0x1:
  - ♦ On NO CAD found with a CAD-LBT exit mode: A Tx is executed and then the chip goes back to the configured fall-back mode.
  - ♦ On CAD found with a CAD-ONLY exit mode: The chip directly goes back to the configured fall-back mode without Rx or Tx.

If [StopTimeoutOnPreamble](#) command is used to enable a stop timeout on header or sync detection, it has no impact on the Rx. Only a sync/header detection can stop the timeout.

The wakeup time is not taken into account, so the real Rx cycle period is **cycle\_time** + wakeup time.

In RxDutyCycleMode, the transceiver context is retained during sleep operations.

Returns CMD\_FAIL in the status of the next command, if the packet type is not supported.

### 6.3.9 SetAutoRxTx

The command *SetAutoRxTx* allows to automatically switch to Tx after an *RxDone*, or to Rx after a *TxDone* (depending on the *Mode* value), with a programmable **Delay** and a specific **Timeout** for the Tx/Rx. This mode can only be triggered once, and must be re-enabled: it is automatically disabled once triggered.

The configured fallback mode is used both as the mode between the Rx/Tx and the autoRxTx, and the mode after the autoRxTx.

If an RxDutyCycleMode is started, this mode behaves the same way as with a normal Rx. But if the **clear** bit is set, this mode is automatically disabled on the first Rx: if no packet is received, for the autoRxTx, it is considered an RxTimeout.

**Table 6-15: SetAutoRxTx Command**

| Byte           | 0              | 1             | 2                    | 3                    | 4                   | 5                  | 6                | 7                | 8               | 9              |
|----------------|----------------|---------------|----------------------|----------------------|---------------------|--------------------|------------------|------------------|-----------------|----------------|
| Data from host | 0x02           | 0x11          | clear<br>Mode(6:0)   | Timeout<br>(23:16)   | Timeout<br>(15:8)   | Timeout<br>(7:0)   | Delay<br>(31:24) | Delay<br>(23:16) | Delay<br>(15:8) | Delay<br>(7:0) |
| Data to host   | Stat<br>(15:8) | Stat<br>(7:0) | IrqStatus<br>(31:24) | IrqStatus<br>(23:16) | IrqStatus<br>(15:8) | IrqStatus<br>(7:0) | 0x00             | 0x00             | 0x00            | 0x00           |

**clear:** Setting this bit disables AutoRxTx mode on a TxTimeout or RxTimeout, as well as on an *RxDone* with an invalid packet when the mode is AUTO\_MODE\_OK.

**Mode:** Sets the auto Rx-Tx mode:

- ♦ 0x0: AUTO\_MODE\_NONE, this disables the AutoRxTx mode
- ♦ 0x1: AUTO\_MODE\_ALWAYS, this enables AutoRxTx on every *RxDone* or *TxDone*.
- ♦ 0x2: AUTO\_MODE\_OK, this enables AutoRxTx on a valid Rx packet only. No specific effect on Tx: works as AUTO\_MODE\_ALWAYS.

**Timeout:** (in 1/32.768 kHz steps): Timeout used for the auto Rx or auto Tx.

**Delay:** (in 1/32 MHz steps): Delay between the *RxDone*/*TxDone* and going into Tx/Rx, allowing a maximum of 134 seconds delay.

### 6.3.10 StopTimeoutOnPreamble

The *StopTimeoutOnPreamble* command determines whether the Rx timeout should halt upon Syncword/Header detection or Preamble detection.

**Table 6-16: StopTimeoutOnPreamble Command**

| Byte           | 0          | 1         | 2                     |
|----------------|------------|-----------|-----------------------|
| Data from Host | 0x02       | 0x09      | stop_on_preamble(7:0) |
| Data to Host   | Stat(15:8) | Stat(7:0) | IrqStatus (31:24)     |

**stop\_on\_preamble** offers the following values:

- ♦ 0x00: Stop on Syncword/Header detection (default value)
- ♦ 0x01: Stop on Preamble detection

### 6.3.11 SetLoraCadParams

The *SetLoraCadParams* command configures the Channel Activity Detection (CAD) features of the LoRa modem. CAD allows the device to detect the presence of LoRa signals on the channel before initiating transmission or reception.

**Table 6-17: SetLoraCadParams Command**

| Byte           | 0           | 1          | 2                 | 3                                     | 4                  | 5                   | 6                  | 7                 | 8              |
|----------------|-------------|------------|-------------------|---------------------------------------|--------------------|---------------------|--------------------|-------------------|----------------|
| Data from host | 0x02        | 0x27       | nb_symbols (7:0)  | rfu(2:0)<br>pbl_any<br>pnr_delta(3:0) | CadExit Mode (7:0) | Cad Timeout (23:16) | Cad Timeout (15:8) | Cad Timeout (7:0) | det_peak (7:0) |
| Data to host   | Stat (15:8) | Stat (7:0) | IrqStatus (31:24) | IrqStatus (23:16)                     | IrqStatus (15:8)   | IrqStatus (7:0)     | 0x00               | 0x00              | 0x00           |

**nb\_symbols** defines the number of contiguous symbols to search for channel activity.

**pbl\_any** parameter determines the type of symbols that trigger CAD detection.

- 0: Detect all (modulated or unmodulated symbols).
- 1: Detect only preamble.
- Note: even with pbl\_any = 1, it is still very likely to detect a payload anyway. Even more so for a small number of symbols, and even more so at low SF, and at high SNR.

**pnr\_delta** parameter controls the possible acceleration of CAD detection.

- 0: Exact number of symbols specified in the **nb\_symbols** parameter is used for the CadDone condition.
- 8: CadDone time is shortened in case of early no detection (fast CAD).

**CadExitMode** parameter specifies the next operation to be performed once the CAD operation is completed. For example, the device can proceed to the next action, such as starting transmission or reception, depending on the CAD result.

**Table 6-18: CadExitMode Parameter Definition**

| CadExitMode | Value | Operation                                                                                                                                                                       |
|-------------|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CAD_ONLY    | 0x00  | The chip performs the CAD operation in LoRa. Once done and whatever the activity on the channel, the chip goes back to the configured Fallback mode.                            |
| CAD_RX      | 0x01  | The chip performs a CAD operation and if an activity is detected, it stays in Rx until a packet is demodulated or the timer reaches the timeout defined by <b>cad_timeout</b> . |
| CAD_LBT     | 0x10  | The chip performs a CAD operation and if no activity is detected, it goes to Tx mode and takes <b>cad_timeout</b> as Tx timeout.                                                |

**cad\_timeout** defines the timeout duration after CAD for Rx (CAD\_RX) or Tx (CAD\_LBT), expressed in periods of 32 MHz crystal oscillator. This parameter is not used in CAD\_ONLY mode.

**det\_peak** value serves as a threshold to decide if there is LoRa activity. It represents the ratio between the peak value and the average, given in 3.4b format (meaning 3 integer bit and 4 fractional bits).

- 55: (Default) Activity is detected if a correlation peak is approximately 5.3 times stronger than the average energy per bin.

**Table 6-19: Lora CAD Recommended det\_peak values**

| Number of symbols | SF5 | SF6 | SF7 | SF8 | SF9 | SF10 | SF11 | SF12 |
|-------------------|-----|-----|-----|-----|-----|------|------|------|
| 1                 | 60  | 60  | 60  | 64  | 64  | 66   | 70   | 74   |
| 2                 | 56  | 56  | 56  | 58  | 58  | 60   | 64   | 68   |
| 3                 | 51  | 51  | 52  | 54  | 56  | 60   | 60   | 65   |
| 4                 | 51  | 51  | 51  | 54  | 56  | 60   | 60   | 64   |

### 6.3.12 SetLoraCAD

The *SetLoraCAD* command puts the chip into Channel Activity Detection (CAD) mode, where it actively searches for LoRa activity in the specified channel.

**Table 6-20: SetLoRaCAD Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x02       | 0x28      |
| Data to host   | Stat(15:8) | Stat(7:0) |

Channel Activity Detection (CAD) is a LoRa-specific mode of operation that enables the device to search for the presence of LoRa preamble symbols or any LoRa signal, depending on the configuration set through the *SetLoraCadParams* command.

The length of the CAD search period is determined by the parameters specified in the *SetLoraCadParams* command. At the end of the search period, the device generates the CadDone IRQ. If a valid signal is detected, it also triggers the CadDetected IRQ.

CAD supports multi-SF detections, allowing the device to perform CAD for multiple spreading factors in parallel. The same rules that apply to multi-SF detections also apply to multi-CAD.

To configure multi-CAD, the following steps should be followed:

- ◆ Set the packet type to LoRa using *SetPacketType*.
- ◆ Configure the LoRa modulation parameters using *SetLoraModulationParams*.
- ◆ Set the LoRa packet parameters using *SetLoraPacketParams*.
- ◆ Configure the CAD parameters using *SetLoraCadParams*.
- ◆ Set the side detectors configuration using *SetLoraSideDetConfig*.
- ◆ Enter Rx mode with *SetLoraCAD*.

Calling *SetLoraModulationParams* disables all side detectors, and they need to be reconfigured by the developer.

Depending on the configured CadExitMode (using *SetLoraCadParams* command), the device either goes back to the configured fallback mode, stays in Rx, or goes to Tx after CAD operation is completed.

In fast-multi-CAD mode (when pnr\_delta is not equal to 0), the main SF (the largest SF) determines the not-detect condition.

The CadDetected IRQ can occur multiple times during the n-symbols window, even for the same SF. The list of SFs that triggered the IRQ can be retrieved using the *GetLoraPacketStatus* command.

It is worth noting that multi-SF Rx detection (not multi-SF CAD) is generally preferred when the exit mode is Rx, as CAD-RX has worse detection performance compared to normal Rx. The choice between multi-SF and multi-CAD depends on the specific application requirements and trade-offs between detection performance and power consumption.

### 6.3.13 SetCad

Starts a CAD in non LoRa mode. Parameters must have been previously set using the [SetCadParams](#) command.

**Table 6-21: SetCad Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x02       | 0x1C      |
| Data to host   | Stat(15:8) | Stat(7:0) |

### 6.3.14 SetCadParams

The *SetCadParams* command defines the CAD parameters for channel activity detection of all radio activity (other than LoRa packets).

**Note:** LoRa specific CAD is defined in [Section 9.9.8 "GetLoraRxStats" on page 162](#).

This CAD is based on the measured RSSI.

**Table 6-22: SetCadParams Command**

| Byte           | 0          | 1         | 2                      | 3                     | 4                    | 5                  | 6                  |
|----------------|------------|-----------|------------------------|-----------------------|----------------------|--------------------|--------------------|
| Data from host | 0x02       | 0x1B      | cad_timeout<br>(23:16) | cad_timeout<br>(15:8) | cad_timeout<br>(7:0) | threshold<br>(7:0) | exit_mode<br>(7:0) |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24)       | IrqStatus(23:16)      | IrqStatus(15:8)      | IrqStatus(7:0)     | 0x00               |

| 7                   | 8                  | 9                 |
|---------------------|--------------------|-------------------|
| trx_timeout (23:16) | trx_timeout (15:8) | trx_timeout (7:0) |
| 0x00                | 0x00               | 0x00              |

**cad\_timeout** defines the maximum time spent measuring RSSI, in HF (32 MHz) clock ticks.

**threshold** defines the level of the RSSI to determine if a signal is present.

- ◆ 0x50 corresponds to a CAD threshold at -80 dBm.

**exit\_mode** defines action taken after the CAD:

- ◆ 0x00: Fallback mode, the chip returns to the setup fallback mode.
- ◆ 0x01: Tx  
If no activity detected: RSSI always below the threshold during **cad\_timeout**, the device goes into Tx.  
If activity is detected: Device returns to the setup fallback mode after the **cad\_timeout**.
- ◆ 0x02: Rx  
If activity is detected: As soon as RSSI gets above **Threshold**, the device goes into Rx.  
If no activity is detected: Device returns to the setup fallback mode after the **cad\_timeout**.
- ◆ Others are RFU

**trx\_timeout** sets a timeout used in case Rx or Tx is triggered, as specified in the [SetTx](#) and [SetRx](#) commands.

### 6.3.15 SetCca

The *SetCca* command sets the radio into Rx mode for Clear Channel Assessment (CCA) measurements for a specified duration.

**Table 6-23: SetCca Command**

| Byte           | 0          | 1         | 2                 | 3                 | 4                | 5               |
|----------------|------------|-----------|-------------------|-------------------|------------------|-----------------|
| Data from host | 0x02       | 0x18      | Duration(23:16)   | Duration(15:8)    | Duration(7:0)    | GainStep(7:0)   |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus (31:24) | IrqStatus (23:16) | IrqStatus (15:8) | IrqStatus (7:0) |

**Duration** is specified in clock ticks of the HF clock (32 MHz). After the CCA assessment is completed, a Timeout IRQ is raised to indicate the end of the assessment and the gain settings is reverted to the configuration which was present before *SetCca* command.

**GainStep** is optional and allows the developer to force a manual gain when performing CCA operations:

- ♦ 0x0: Sets automatic mode using AGC
- ♦ 0x1 to 0xD: Sets the gain to a fixed value defined by **GainStep** during CCA. See [Section 1.4 Receiver RF Path](#) for LNA gain information.
- ♦ Not provided, or 0xFF: Does not change the current configuration.

Results can be read with the [GetCcaResult](#) command.

### 6.3.16 ResetRxStats

The *ResetRxStats* command resets the internal statistics of the received packets. See *Get<modulation>ModulationParams* command for supported modulations.

**Table 6-24: ResetRxStats Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x02       | 0x2A      |
| Data to host   | Stat(15:8) | Stat(7:0) |

### 6.3.17 SetDefaultRxTxTimeout

The *SetDefaultRxTxTimeout* command configures the default Rx and Tx timeouts used for DIO (Digital Input/Output) Rx/Tx triggers.

It also serves as a fallback when timeout parameters are not explicitly provided in the [SetRx](#) and [SetTx](#) commands.

**Table 6-25: SetDefaultRxTxTimeout Command**

| Byte           | 0           | 1          | 2                  | 3                 | 4                | 5                  | 6                 | 7                |
|----------------|-------------|------------|--------------------|-------------------|------------------|--------------------|-------------------|------------------|
| Data from host | 0x02        | 0x15       | rx_timeout (23:16) | rx_timeout (15:8) | rx_timeout (7:0) | tx_timeout (23:16) | tx_timeout (15:8) | tx_timeout (7:0) |
| Data to host   | Stat (15:8) | Stat (7:0) | IrqStatus (31:24)  | IrqStatus (23:16) | IrqStatus (15:8) | IrqStatus (7:0)    | 0x00              | 0x00             |

The **rx\_timeout** and **tx\_timeout** parameters are expressed in periods of the 32.768 kHz RTC (Real-Time Clock).

These parameters determine the duration for which the device waits before triggering a timeout during Rx or Tx operations.



### 6.3.18 SetRegMode

The *SetRegMode* command sets how the SIMO can be used. The command can be issued only in Standby RC mode, otherwise it returns CMD\_FAIL on the next *GetStatus* command.

**Table 6-26: SetRegMode Command**

| Byte           | 0          | 1         | 2                 |
|----------------|------------|-----------|-------------------|
| Data from host | 0x01       | 0x21      | simo_usage(7:0)   |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus (31:24) |

*simo\_usage* defines if and how the SIMO can to be switched ON and used for VDCC and VPAX ramping:

- ♦ 0x00: SIMO\_OFF: no SIMO available, always use LDO regulators for VDCC and VPAX (Default)
- ♦ 0x01: RFU
- ♦ 0x02: SIMO\_NORMAL: use DC-DC converter with SIMO for VDCC and for VPAX during Tx mode, but not for VPAX ramping
- ♦ 0x03: RFU

## 6.4 Chip Auto Calibration

During the startup sequence, the device firmware calibrates the low and high frequency RC oscillators, the PLL, the ADC, and the image rejection of the mixer at 915 MHz. After the calibration procedure the device is set in Standby RC mode.

- If operating at another frequency, the image calibration procedure must be restarted using command [Calibrate](#).
- An image calibration is advised after large temperature variations and optimal image rejection using command [Calibrate](#).
- Image calibration is necessary if there is a frequency change > 10 MHz, or a temperature change > 10 °C.

### 6.4.1 Calibrate

The *Calibrate* command performs calibration for the requested blocks specified by the **blocks\_to\_calibrate** parameter. This calibration process remains effective in any mode of the chip. Upon completion of the calibration, the chip automatically enters Standby RC mode. This command can not be issued in Rx or Tx mode.

**Table 6-27: Calibrate Command**

| Byte           | 0          | 1         | 2                        |
|----------------|------------|-----------|--------------------------|
| Data from host | 0x01       | 0x22      | blocks_to_calibrate(7:0) |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus (31:24)        |

The definition of the parameter **blocks\_to\_calibrate** is in the next table:

**Table 6-28: Blocks\_to\_calibrate Parameter Definition**

| Bits | 7   | 6      | 5  | 4   | 3   | 2   | 1     | 0     |
|------|-----|--------|----|-----|-----|-----|-------|-------|
| Name | rfu | PA_OFF | MU | rfu | AAF | PLL | HF_RC | LF_RC |

With the following definition:

- ♦ LF\_RC: Low frequency clock RC calibration
- ♦ HF\_RC: High frequency clock RC calibration
- ♦ PLL: PLL calibrations
- ♦ AAF: Anti aliasing filter calibration
- ♦ MU: Measure unit calibration
- ♦ PA\_OFF: Power amplifier calibration

Potential calibration failures can be read out with the [ClearErrors](#) command.

It is advised to perform the PLL and AAF calibrations again for an RF frequency change greater than 50 MHz, or for a temperature change beyond +/-20C.

## 6.4.2 CalibFE

*CalibFE* launches all Receiver Front End (Rx + PLL) calibrations at the given frequencies. This command does not work if device is in Rx or Tx mode, and returns CMD\_FAIL on the next *GetStatus* command. On exit, the chip goes back to the mode where the command was initiated.

**Table 6-29: CalibFE Command**

| Byte           | 0              | 1             | 2                    | 3                    | 4                   | 5                  | 6           | 7          |
|----------------|----------------|---------------|----------------------|----------------------|---------------------|--------------------|-------------|------------|
| Data from host | 0x01           | 0x23          | Freq1(15:8)          | Freq1(7:0)           | Freq2(15:8)         | Freq2(7:0)         | Freq3(15:8) | Freq3(7:0) |
| Data to host   | Stat<br>(15:8) | Stat<br>(7:0) | IrqStatus<br>(31:24) | IrqStatus<br>(23:16) | IrqStatus<br>(15:8) | IrqStatus<br>(7:0) | 0x00        | 0x00       |

The Front End calibration can be performed on either Rx path (LF and HF). The Rx path is encoded in the most significant bit of each frequency: 0 = LF path, 1 = HF path.

**Freqn:** Frequencies are given in 4 MHz steps (Ex: 900 MHz -> 0x00E1 on LF path, 0x80E1 on HF path). All frequency parameters are optional:

- Calibration is only performed (and updated) on provided and non zero frequencies (Path can be any value). It is thus possible to update only 1 or 2 calibrations by not providing other calibration frequencies, or setting other calibration frequencies to 0.
- If same frequency and path values are provided, only the calibration on the first of the identical freq/paths is performed. If same frequencies but different paths are provided, both calibrations are executed.
- If no frequency is given as argument, the Front End is calibrated for the current RF frequency and Rx path by truncating the current RF frequency to a 4 MHz multiple (always lower). The first slot (freq1) is used to save the calibration values. This overwrites any previous calibration in that slot.

The calibration values for all frequencies are stored on chip.

**Notes:** There is no Front End calibration performed after a POR or cold start or wakeup after sleep without retention. The *CalibFE* command must thus be executed to provide optimum performance.

The Rx paths are called LF (low Frequency) and HF (High Frequency) as they are tuned to perform better for those range of frequencies. Thus it is better to use (and calibrate them) depending on the frequency used. But it is not forbidden to use the HF path for low frequencies and vice-versa.

## 6.5 Measurements

### 6.5.1 GetVbat

In standby mode, the *GetVBat* command retrieves the current value of the supply voltage (Vbat) in the specified **Format**:

**Table 6-30: GetVBat Request Command**

| Byte           | 0          | 1         | 2                                     |
|----------------|------------|-----------|---------------------------------------|
| Data from host | 0x01       | 0x24      | rfu(3:0)<br>Format<br>Resolution(2:0) |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus (31:24)                     |

**Table 6-31: GetVBat Response**

| Byte                     | 0          | 1         | 2            | 3           |
|--------------------------|------------|-----------|--------------|-------------|
| Data from host           | 0x00       | 0x00      | 0x00         | 0x00        |
| Data to host: Format = 0 | Stat(15:8) | Stat(7:0) | Raw(12:8)    | Raw(7:0)    |
| Data to host: Format = 1 | Stat(15:8) | Stat(7:0) | VBatMV(12:8) | VBatMV(7:0) |

**Format** defines the format of the returned value:

- ♦ **Format** = 0: the chip returns the raw supply Vbat value. The Vbat value is a function of the Vana voltage (typically 1.35 V) using the following formula:

$$Vbat(V) = \left( 5 \times \frac{Raw(12:0)}{8192} - 1 \right) \times vana(v)$$

- ♦ **Format** = 1: the chip provides the supply voltage value in millivolts [mV] via parameter **VBatMV**.

**Resolution** defines the number of significant LSB when **Format** is set to 1 (bit 0 is always the LSB).

- ♦ 0x0: 8 bits
- ♦ 0x1: 9 bits
- ♦ 0x2: 10 bits
- ♦ 0x3: 11 bits
- ♦ 0x4: 12 bits
- ♦ 0x5: 13 bits

The higher the resolution the longer the measurement duration is.

## 6.5.2 GetTemp

In standby mode, the *GetTemp* command retrieves the current value of the temperature in the specified **Format**.

**Table 6-32: GetTemp Command**

| Byte           | 0          | 1         | 2                                                |
|----------------|------------|-----------|--------------------------------------------------|
| Data from host | 0x01       | 0x25      | rfu(1:0), Source(1:0)<br>Format, Resolution(2:0) |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus (31:24)                                |

**Table 6-33: GetTemp Response**

| Byte                     | 0          | 1         | 2          | 3         |
|--------------------------|------------|-----------|------------|-----------|
| Data from host           | 0x00       | 0x00      | 0x00       | 0x00      |
| Data to host: Format = 0 | Stat(15:8) | Stat(7:0) | Raw(12:8)  | Raw(7:0)  |
| Data to host: Format = 1 | Stat(15:8) | Stat(7:0) | Temp(12:5) | Temp(4:0) |

**Source** sets the temperature sensor source for temperature measurement.

- ♦ 0x00: Built-in junction temperature Vbe
- ♦ 0x01: Built-in junction temperature close to XOSC
- ♦ 0x02: NTC
- ♦ 0x03: RFU

**Format** defines the meaning of the returned value for measured temperature:

- ♦ 0: The chip returns the raw value of the temperature. The actual temperature value is obtained using the following formula:

$$\text{Temperature} = \left( \text{Vana} \times \frac{\text{Raw}(12:0)}{8192} - \text{Vbe25} \right) \times \frac{1000}{\text{VbeSlope}} + 25$$

In this formula:

- ♦ Vana voltage is 1.35 V(typ)
- ♦ Vbe25 is Junction voltage@25 °C 0.7295 V(typ)
- ♦ VbeSlope is -1.7 mV/°C(typ)
- ♦ 1: The chip returns the temperature in °C in 13 bits. Actual temperature = **Temp**/32

**Resolution** defines the number of significant LSB when **Format** is set to 1 (bit 0 is always the LSB). The resolution is used in ADC to perform a temperature measurement (the higher the resolution the longer the measurement duration is), where resolution:

- ♦ 0x0: 8 bits
- ♦ 0x1: 9 bits
- ♦ 0x2: 10 bits
- ♦ 0x3: 11 bits
- ♦ 0x4: 12 bits
- ♦ 0x5: 13 bits

**Note:** If a TCXO is currently being used, this command fails if the NTC is used as source.

For the raw ADC format for NTC, the temperature can be calculated as follows:

- ♦  $\text{RRatio} = \text{RNTC} / \text{RNTC}_{25} = \text{RBiasRatio} * (\text{Raw} / 8192) / (1.0 - (\text{Raw} / 8192))$
- ♦ The temperature is then given by  $T = \text{Beta} / (\ln(\text{RRatio}) + \text{Beta} / 298.15) - 273.15$

Where RBiasRatio, Beta are values set by the *SetNtcParams* command. Refer to Application Note AN1200.106 for details.

## 6.6 Low Battery Detector Configuration

### 6.6.1 SetLbdCfg

The *SetLbdCfg* command sets the threshold, based on the measurement of VBAT, to trigger a low battery interrupt.

**Table 6-34: SetLbdCfg Command**

| Byte           | 0          | 1         | 2                               |
|----------------|------------|-----------|---------------------------------|
| Data from host | 0x01       | 0x30      | Trim(2:0)<br>rfu(3:0)<br>Enable |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24)                |

**Trim** defines the threshold value:

- ◆ 0x0: 1.6 V
- ◆ 0x1: 1.67 V
- ◆ 0x2: 1.74 V
- ◆ 0x3: 1.8 V
- ◆ 0x4: 1.88 V (default)
- ◆ 0x5: 1.95 V
- ◆ 0x6: 2 V
- ◆ 0x7: 2.1 V

**Note:** Settings 0, 1 and 2 are in principle not useful, since the specified voltage range is between 1.8 to 3.7 V.

**Enable** switches the low battery detection block ON/OFF:

- 0x0: Off
- 0x1: On (default)

## 6.6.2 GetRandomNumber

The *GetRandomNumber* command returns a 32-bit random number. This number should not be used for cryptographic purposes. *GetRandomNumber* works in any mode without perturbation on an ongoing Rx or Tx operation, it blocks any Tx/Rx trigger operation.

**Table 6-35: GetRandomNumber Request Command**

| Byte           | 0          | 1         | 2                |
|----------------|------------|-----------|------------------|
| Data from host | 0x01       | 0x26      | Source(1:0)      |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) |

**Source** (optional, default is 0x3) selects the entropy source(s).

- ♦ Bit 0: Entropy source PLL. If this bit is 1, the PLL is used as entropy source. The circuit automatically goes to FS mode if it is in a lower mode, and returns to current mode afterwards.
- ♦ Bit 1: Entropy source ADC. If this bit is 1 the ADC is used as entropy source. The circuit automatically goes to FS and enables the necessary blocks if it is in a lower mode, and returns to current mode and restores settings afterwards.

Semtech recommends to always use both entropy sources. If no source is selected, the randomness of the number is very low.

**Table 6-36: GetRandomNumber Response**

| Byte           | 0          | 1         | 2          | 3          | 4         | 5        |
|----------------|------------|-----------|------------|------------|-----------|----------|
| Data from host | 0x00       | 0x00      | 0x00       | 0x00       | 0x00      | 0x00     |
| Data to host   | Stat(15:8) | Stat(7:0) | Rnd(31:24) | Rnd(23:16) | Rnd(15:8) | Rnd(7:0) |

## 6.7 Chip Status

### 6.7.1 GetStatus

The *GetStatus* command serves two purposes that are based on the previous command executed:

- If the previous command was not a read command, issuing *GetStatus* returns the status register, similar to any other command. In this case, there is no need to send a second SPI frame with 0x00s after the *GetStatus* command to read the status register.
- If the previous transaction was a read command, *GetStatus* returns the result of the read command. It behaves similarly to sending 0x00s after the read command.

*GetStatus* also clears the reset status returned in **Stat**, which allows developers to identify any unwanted resets that may have occurred.

**Table 6-37: GetStatus Command**

| Byte           | 0          | 1         | 2                | 3                | 4               | 5              |
|----------------|------------|-----------|------------------|------------------|-----------------|----------------|
| Data from host | 0x01       | 0x00      | 0x00             | 0x00             | 0x00            | 0x00           |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) | IrqStatus(23:16) | IrqStatus(15:8) | IrqStatus(7:0) |

The **Stat** variable is a 16-bit value that is transmitted at the beginning of each SPI frame on the MISO (Manager In Subordinate Out) line. The **Stat** variable informs the host about the current state of the device and any relevant conditions.

**Table 6-38: Status (Stat) Definition**

| Bits | (15:12) | (11:9)        | (8)             | (7:4)       | (3) | (2:0)    |
|------|---------|---------------|-----------------|-------------|-----|----------|
| Name | 0x0     | CommandStatus | InterruptStatus | ResetSource | rfu | ChipMode |

**CommandStatus** definition:

- 0x0: CMD\_FAIL - The latest command could not be executed
- 0x1: CMD\_PERR - The latest command could not be processed (wrong Opcode, arguments)
- 0x2: CMD\_OK - The latest command was processed successfully
- 0x3: CMD\_DAT - The latest command was a successfully processed read, and data is currently transmitted instead of IrqStatus

An IRQ on DIO (Digital Input/Output) can be generated if a CMD\_FAIL or CMD\_PERR occurred using CmdError IRQ bit.

**InterruptStatus** definition:

- 0: No Interrupt active
- 1: At least 1 Interrupt is active

**ResetSource**: Reports the first reset source that triggered the chip reset:

- 0x0: Cleared
- 0x1: Analog (POR/BRN - Power-On Reset/Brown-Out Reset)
- 0x2: NRESET pin
- 0x3: RFU

**ChipMode**: Current chip mode definition:

- 0x0: SLEEP
- 0x1: STDBY RC (Internal RC oscillator)
- 0x2: STDBY XOSC (Crystal Oscillator)
- 0x3: FS (Frequency Synthesizer)
- 0x4: Rx (Receiver)
- 0x5: Tx (Transmitter)



## 6.7.2 GetVersion

The *GetVersion* command retrieves the version information of the device. LR2012 and LR2022 share the same Version

**Table 6-39: GetVersion Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x01       | 0x01      |
| Data to host   | Stat(15:8) | Stat(7:0) |

**Table 6-40: GetVersion Response**

| Byte           | 0          | 1         | 2            | 3            |
|----------------|------------|-----------|--------------|--------------|
| Data from host | 0x00       | 0x00      | 0x00         | 0x00         |
| Data to host   | Stat(15:8) | Stat(7:0) | FWMajor(7:0) | FWMinor(7:0) |

*FWMajor* + *FWMinor* is the current version of the LR20xx parts.

| GetVersion     | LR2021 | LR2022 | LR2012 |
|----------------|--------|--------|--------|
| <i>FWMajor</i> | 0x01   | 0x02   | 0x02   |
| <i>FWMinor</i> | 0x18   | 0x00   | 0x00   |

## 6.7.3 ClearErrors

The *ClearErrors* command clears all error flags that are pending in the device. This command clears all error conditions at once, and does not allow individual error flags to be cleared separately.

Calling *ClearErrors* does not clear the Error IRQ. The IRQ has to be cleared explicitly with the *ClearIrq* command.

**Table 6-41: ClearError Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x01       | 0x11      |
| Data to host   | Stat(15:8) | Stat(7:0) |

## 6.7.4 GetErrors

The *GetErrors* command provides the current pending errors that have occurred since the last *ClearError* call or since the startup of the circuit. This command is valuable for retrieving error information and diagnosing issues that might have arisen during device operation. It is also possible to generate an IRQ on DIO (Digital Input/Output) when an error occurs, allowing timely notification of any error condition.

Importantly, there is no masking of errors possible, meaning that all errors are reported and cannot be selectively suppressed.

**Table 6-42: GetErrors Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x01       | 0x10      |
| Data to host   | Stat(15:8) | Stat(7:0) |

**Table 6-43: GetErrors Response**

| Byte           | 0          | 1         | 2               | 3              |
|----------------|------------|-----------|-----------------|----------------|
| Data from host | 0x00       | 0x00      | 0x00            | 0x00           |
| Data to host   | Stat(15:8) | Stat(7:0) | ErrorStat(15:8) | ErrorStat(7:0) |

The **ErrorStat** variable contains all possible error flags that could occur during various chip operations. Each bit corresponds to a specific error type:

- ♦ bit 0: HF\_XOSC\_START\_ERR. High frequency Xosc did not start correctly.
- ♦ bit 1: LF\_XOSC\_START\_ERR. Low frequency Xosc did not start correctly. This is either a hardware problem, which might be resolved by doing a reset, or there is a TCXO instead which must be enabled through *SetTcxoMode* command.
- ♦ bit 2: PLL\_LOCK\_ERR. The PLL did not lock. This can come from too high or too low a frequency, or if the PLL was not calibrated. To fix it try redoing a PLL calibration, or using other frequencies.
- ♦ bit 3: LF\_RC\_CALIB\_ERR. Error during calibration of the low frequency RC, thus no calibration is available.
- ♦ bit 4: HF\_RC\_CALIB\_ERR. Error during calibration of the high frequency RC, thus no calibration is available.
- ♦ bit 5: PLL\_CALIB\_ERR. Error during calibration of the PLL, thus no calibration is available.
- ♦ bit 6: AAF\_CALIB\_ERR. Error during calibration of the aaf (anti-aliasing filter), thus no calibration is available.
- ♦ bit 7: IMG\_CALIB\_ERR. Error during calibration of the image rejection (IQ comp), thus no calibration is available.
- ♦ bit 8: CHIP\_BUSY. Error asserted when a DIO Tx or Rx trigger could not be executed because chip was busy changing mode.
- ♦ bit 9: RXFREQ\_NO\_FE\_CAL\_ERR. Front End calibration was not available for Rx operation with specified RF frequency.
- ♦ bit 10: MEAS\_UNIT\_ADC\_CALIB\_ERR. Error during calibration of the measure unit ADC, thus no calibration is available.
- ♦ bit 11: PA\_OFFSET\_CALIB\_ERR. Error during calibration of the PA offset, thus no calibration is available.
- ♦ bit 12–15: RFU

## 6.8 DIO Control

### 6.8.1 SetDioFunction

The *SetDioFunction* command configures the functionality of the freely configurable Digital Input/Output (DIO pins), and can set the pull-up/down configuration for Sleep modes.

**Table 6-44: SetDioFunction Command**

| Byte           | 0          | 1         | 2                | 3                          |
|----------------|------------|-----------|------------------|----------------------------|
| Data from host | 0x01       | 0x12      | Dio(7:0)         | Func(3:0), pull_drive(3:0) |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) | IrqStatus (23:16)          |

**Dio** defines the DIO being configured: Allowed values are [5:11].

**Func** defines the DIO functionality using the following options:

- ◆ 0x0: DIO\_FUNCTION\_NONE (Hi-Z)
- ◆ 0x1: DIO\_FUNCTION\_IRQ (Output)
- ◆ 0x2: DIO\_FUNCTION\_RF\_SWITCH (Output)
- ◆ 0x3: RFU
- ◆ 0x4: RFU
- ◆ 0x5: DIO\_FUNCTION\_GPIO\_OUTPUT\_LOW (Output)
- ◆ 0x6: DIO\_FUNCTION\_GPIO\_OUTPUT\_HIGH (Output)
- ◆ 0x7: DIO\_FUNCTION\_HF\_CLK\_OUT (Output)
- ◆ 0x8: DIO\_FUNCTION\_LF\_CLK\_OUT (for DIO7–11 only) (Output)
- ◆ 0x9: DIO\_FUNCTION\_TX\_TRIGGER (Input)
- ◆ 0xA: DIO\_FUNCTION\_RX\_TRIGGER (Input)

**pull\_drive** defines the pull-up/pull-down state of the selected DIO in Sleep mode with the following options:

- ◆ 0x0: DIO\_SLEEP\_PULL\_NONE
- ◆ 0x1: DIO\_SLEEP\_PULL\_DOWN
- ◆ 0x2: DIO\_SLEEP\_PULL\_UP
- ◆ 0x3: DIO\_SLEEP\_PULL\_AUTO

The **pull\_drive** setting is applied when entering Sleep mode if any functionality other than DIO\_FUNCTION\_NONE was previously configured.

DIO\_SLEEP\_PULL\_AUTO means that if the DIO[5 to 11] value in Standby mode is '1', it is pulled-up, and if it is '0', it is pulled-down. On sleep without retention, **pull\_drive** is reset to NONE on re-start (except for DIO5/6: pull-up by default).

If the external clock input is enabled (*ConfigLfClock*), and the command tries to set a function on DIO11, the command fails.

hf\_clk\_out disappears during sleep, but lf\_clk\_out still outputs the lf\_clock during sleep if it is enabled (clk\_32k\_enable in *SetSleep*). Any output with a lf\_clk\_out function is restored after sleep even without retention, and even if the lf\_clk is disabled during sleep.

Asserting a DIO to 1 configured as DIO\_FUNCTION\_TX\_TRIGGER, performs a *SetTx* (default Tx timeout).

Asserting a DIO to 1 configured as DIO\_FUNCTION\_RX\_TRIGGER, performs a *SetRx* (default Rx timeout).

DIO5 & DIO6 have a pull-up by default that can be disabled by forcing their function to DIO\_FUNCTION\_NONE. They should be forced to DIO\_FUNCTION\_NONE on cold starts / start without retention if they are connected to an external component that makes them toggle before their real function is set.

For DIO = 5, only DIO\_SLEEP\_PULL\_UP is accepted. Otherwise the command returns FAIL.

## 6.8.2 SetDioRfSwitchConfig

The *SetDioRfSwitchConfig* command configures the value of the specified DIO pin when it is set as an RF switch using the *SetDioFunction* command. Each bit in the configuration specifies the state of the DIO pin in the corresponding chip mode.

With this command, developers can set up the behaviour of the DIO pin to act as an RF switch in various chip modes.

**Table 6-45: SetDioRfSwitchConfig Command**

| Byte           | 0          | 1         | 2                | 3                                                  |
|----------------|------------|-----------|------------------|----------------------------------------------------|
| Data from host | 0x01       | 0x13      | Dio(7:0)         | rfu(2:0)   tx_hf   rx_hf   tx_lf   rx_lf   standby |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) | IrqStatus (23:16)                                  |

**Dio** is in the range 5 to 11.

## 6.8.3 SetDioIrqConfig

The *SetDioIrqConfig* command maps specific IRQs (Interrupt Requests) to the specified DIO pin when it is set as an IRQ using the *SetDioFunction* command. It is not possible to output the same IRQ onto 2 different DIOs (the last mapping for each IRQ is taken into account).

**Table 6-46: SetDioIrqConfig Command**

| Byte           | 0          | 1         | 2                | 3                 | 4                | 5               | 6        |
|----------------|------------|-----------|------------------|-------------------|------------------|-----------------|----------|
| Data from host | 0x01       | 0x15      | Dio(7:0)         | Irq(31:24)        | Irq(23:16)       | Irq(15:8)       | Irq(7:0) |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) | IrqStatus (23:16) | IrqStatus (15:8) | IrqStatus (7:0) | 0x00     |

**Dio**: In the range 5 to 11.

**Irq**: See [Section 5.7](#) for a description of the **Irq** bits.

## 6.9 Interrupt Control

### 6.9.1 ClearIrq

The *ClearIrq* command clears specific bits in the internal pending IRQ (Interrupt Request) register managed by the chip.

By issuing the *ClearIrq* command, developers can selectively clear IRQ flags, allowing for precise management of interrupt events and ensuring that only relevant IRQs are reset.

**Table 6-47: ClearIrq Command**

| Byte           | 0          | 1         | 2                     | 3                     | 4                    | 5                   |
|----------------|------------|-----------|-----------------------|-----------------------|----------------------|---------------------|
| Data from host | 0x01       | 0x16      | irqs_to_clear (31:24) | irqs_to_clear (23:16) | irqs_to_clear (15:8) | irqs_to_clear (7:0) |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus (31:24)     | IrqStatus (23:16)     | IrqStatus (15:8)     | IrqStatus (7:0)     |

### 6.9.2 GetAndClearIrqStatus

The *GetAndClearIrqStatus* command serves two purposes:

- It reads out the pending IRQs, allowing retrieval of information about the active interrupt events that are pending in the device.
- It clears all pending IRQs, effectively resetting the internal IRQ register and preparing it for new interrupt events.

**Table 6-48: GetAndClearIrqStatus Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x01       | 0x17      |
| Data to host   | Stat(15:8) | Stat(7:0) |

**Table 6-49: GetAndClearIrqStatus Response**

| Byte           | 0          | 1         | 2            | 3            | 4           | 5         |
|----------------|------------|-----------|--------------|--------------|-------------|-----------|
| Data from host | 0x00       | 0x00      | 0x00         | 0x00         | 0x00        | 0x00      |
| Data to host   | Stat(15:8) | Stat(7:0) | irqs (31:24) | irqs (23:16) | irqs (15:8) | irqs(7:0) |

**Note:** Any IRQs that trigger after the clear is done (before the read is executed) are not listed in the read: only cleared IRQs are listed.

## 6.10 Data FIFO

### 6.10.1 ConfigFifoIrq

The *ConfigFifoIrq* command configures which FIFO level status flags trigger an RxFifo or TxFifo IRQ (Interrupt Request) as well as the threshold levels triggering the flags.

**Table 6-50: ConfigFifoIrq Command**

| Byte                       | 0                         | 1                          | 2                           | 3                           | 4                           | 5                          |
|----------------------------|---------------------------|----------------------------|-----------------------------|-----------------------------|-----------------------------|----------------------------|
| Data from host             | 0x01                      | 0x1A                       | rx_fifo_irq_enable<br>(7:0) | tx_fifo_irq_enable<br>(7:0) | rx_high_threshold<br>(15:8) | rx_high_threshold<br>(7:0) |
| Data to host               | Stat(15:8)                | Stat(7:0)                  | IrqStatus (31:24)           | IrqStatus (23:16)           | IrqStatus (23:16)           | IrqStatus (23:16)          |
| 6                          | 7                         | 8                          | 9                           | 10                          | 11                          |                            |
| tx_low_threshold<br>(15:8) | tx_low_threshold<br>(7:0) | rx_low_threshold<br>(15:8) | rx_low_threshold<br>(7:0)   | tx_high_threshold<br>(15:8) | tx_high_threshold<br>(7:0)  |                            |
| 0x00                       | 0x00                      | 0x00                       | 0x00                        | 0x00                        | 0x00                        |                            |

**rx\_fifo\_irq\_enable** defines for all bits set in this parameter, the corresponding Rx FIFO flag that triggers the RxFifo IRQ.

**tx\_fifo\_irq\_enable** defines for all bits set in this parameter, the corresponding Tx FIFO flag that triggers the TxFifo IRQ.

Optional parameters for threshold levels:

**rx\_high\_threshold** sets the threshold level to use for the Rx FIFO high flag.

**tx\_low\_threshold** sets the threshold level to use for the Tx FIFO low flag.

**rx\_low\_threshold** sets the threshold level to use for the Rx FIFO low flag.

**tx\_high\_threshold** sets the threshold level to use for the Tx FIFO high flag.

Available FIFO flags:

- ♦ 0x01: FifoEmpty
- ♦ 0x02: FifoLow
- ♦ 0x04: FifoHigh
- ♦ 0x08: FifoFull
- ♦ 0x10: FifoOverflow
- ♦ 0x20: FifoUnderflow

## 6.10.2 GetFifoIrqFlags

The *GetFifoIrqFlags* command retrieves all FIFO flags that have been triggered since the last clear operation. This command is independent of whether the corresponding IRQs are enabled or not.

Important points to note: clearing the RxFifo/TxFifo IRQ does not clear the flags. As a result, the flags are not affected, but the IRQ can still be triggered if a flag is activated again.

**Table 6-51: GetFifoIrqFlags Request Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x01       | 0x1B      |
| Data to host   | Stat(15:8) | Stat(7:0) |

**Table 6-52: GetFifoIrqFlags Read Command**

| Byte           | 0          | 1         | 2                 | 3                 |
|----------------|------------|-----------|-------------------|-------------------|
| Data from host | 0x00       | 0x00      | 0x00              | 0x00              |
| Data to host   | Stat(15:8) | Stat(7:0) | RxFifoFlags (7:0) | TxFifoFlags (7:0) |

For *RxFifoFlag* and *TxFifoFlag* values, refer to [ConfigFifoIrq](#) command.

## 6.10.3 ClearFifoIrqFlags

The *ClearFifoIrqFlags* command clears specific FIFO IRQ flags. Clearing the flag does not clear RxFifo/TxFifo IRQ.

**Table 6-53: ClearFifoIrqFlags Command**

| Byte           | 0          | 1         | 2                       | 3                       |
|----------------|------------|-----------|-------------------------|-------------------------|
| Data from host | 0x01       | 0x14      | RxFifoFlagsToClear(7:0) | TxFifoFlagsToClear(7:0) |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus (31:24)       | IrqStatus (23:16)       |

*RxFifoFlagsToClear* is defined as a byte. For all bits set in the parameter, the corresponding Rx FIFO flag is cleared. By setting specific bits in this byte, developers can selectively clear the desired Rx FIFO flags.

*TxFifoFlagsToClear* is also defined as a byte. For all bits set in the parameter, the corresponding Tx FIFO flag is cleared. Setting specific bits in this byte selectively clears the desired Tx FIFO flags.

## 6.10.4 GetAndClearFifoIrqFlags

The *GetAndClearFifoIrqFlags* command serves two purposes:

- ♦ It clears specific FIFO IRQ flags that have triggered since the last clear operation.
- ♦ It returns those flags.

Importantly, it does not depend on enabled flags for the IRQs. Clearing the RxFifo/TxFifo IRQ does not clear the flags, but the IRQ can still be triggered if a flag is activated again.

**Table 6-54: GetAndClearFifoIrqFlags Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x01       | 0x2E      |
| Data to host   | Stat(15:8) | Stat(7:0) |

**Table 6-55: GetAndClearFifoIrqFlags Response**

| Byte           | 0          | 1         | 2           | 3           |
|----------------|------------|-----------|-------------|-------------|
| Data from host | 0x00       | 0x00      | 0x00        | 0x00        |
| Data to host   | Stat(15:8) | Stat(7:0) | RxFifoFlags | TxFifoFlags |

For *RxFifoFlags* and *TxFifoFlags* value, refer to [ConfigFifoIrq](#) command.

## 6.10.5 GetRxFifoLevel

The *GetRxFifoLevel* command retrieves the fill level of the Rx radio FIFO in bytes.

**Table 6-56: GetRxFifoLevel Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x01       | 0x1C      |
| Data to host   | Stat(15:8) | Stat(7:0) |

**Table 6-57: GetRxFifoLevel Response**

| Byte           | 0          | 1         | 2           | 3          |
|----------------|------------|-----------|-------------|------------|
| Data from host | 0x00       | 0x00      | 0x00        | 0x00       |
| Data to host   | Stat(15:8) | Stat(7:0) | Level(15:8) | Level(7:0) |



## 6.10.6 GetTxFifoLevel

The *GetTxFifoLevel* command retrieves the fill level of the Tx radio FIFO in bytes.

**Table 6-58: GetTxFifoLevel Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x01       | 0x1D      |
| Data to host   | Stat(15:8) | Stat(7:0) |

**Table 6-59: GetTxFifoLevel Response**

| Byte           | 0          | 1         | 2           | 3          |
|----------------|------------|-----------|-------------|------------|
| Data from host | 0x00       | 0x00      | 0x00        | 0x00       |
| Data to host   | Stat(15:8) | Stat(7:0) | Level(15:8) | Level(7:0) |

## 6.10.7 ClearRxFifo

The *ClearRxFifo* command clears the Rx FIFO.

**Table 6-60: ClearRxFifo Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x01       | 0x1E      |
| Data to host   | Stat(15:8) | Stat(7:0) |

## 6.10.8 ClearTxFifo

The *ClearTxFifo* command clears the Tx FIFO.

**Table 6-61: ClearTxFifo Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x01       | 0x1F      |
| Data to host   | Stat(15:8) | Stat(7:0) |

## 6.11 Clock Control

### 6.11.1 ConfigLfClock

The *ConfigLfClock* command configures the 32 kHz clock source.

**Table 6-62: ConfigLfClock Command**

| Byte           | 0          | 1         | 2                      |
|----------------|------------|-----------|------------------------|
| Data from Host | 0x01       | 0x18      | rfu(5:0) lf_clock(1:0) |
| Data to Host   | Stat(15:8) | Stat(7:0) | IrqStatus (31:24)      |

*lf\_clock* selects the desired LF clock source:

- ♦ 0x0: Use 32 kHz RC oscillator (default)
- ♦ 0x1: RFU
- ♦ 0x2: Use externally provided 32.768 kHz signal on DIO11 pin
- ♦ 0x3: RFU

When switching the LF clock source to an external source, the external source must already be running when calling this command and must run all the time afterwards. If it is not running, the command fails. If DIO11 is already used for a special function, the command fails.

When switching to LF\_RC, the LF\_RC is enabled and the FW waits for the LF\_RC startup. Then the LF clock source is switched.

### 6.11.2 ConfigClkOutputs

The *ConfigClkOutputs* command configures the HF (High-Frequency) clocks for output on a DIO (Digital Input/Output) pin.

**Table 6-63: ConfigClkOutputs Command**

| Byte           | 0          | 1         | 2                                   |
|----------------|------------|-----------|-------------------------------------|
| Data from host | 0x01       | 0x19      | rfu(3:0)<br>hf_clk_out_scaling(3:0) |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus (31:24)                   |

*hf\_clk\_out\_scaling* selects the scaling for the HF (High-Frequency) clock output:

- ♦ 0x0: Division by 1 (32 MHz)
- ♦ 0x1: Division by 2 (16 MHz)
- ♦ 0x2: Division by 4 (8 MHz)
- ♦ ...
- ♦ 0xF: Division by 32768 (976.5625 kHz)

By specifying the appropriate value for *hf\_clk\_out\_scaling*, developers can control the output frequency of the HF clock on the designated DIO pin defined by command [SetDioFunction](#).

### 6.11.3 SetTcxoMode

The *SetTcxoMode* command configures the chip for a connected TCXO (Temperature-Compensated Crystal Oscillator).

This command only functions correctly when the chip is in Standby RC mode. If attempted in any other mode, the command returns CMD\_FAIL upon the subsequent *GetStatus* command.

**Note:** A complete reset of the chip is required to return to normal XOSC (Crystal Oscillator) operation after using TCXO mode.

**Table 6-64: SetTcxoMode Command**

| Byte           | 0          | 1         | 2                | 3                  | 4                  | 5                 | 6                |
|----------------|------------|-----------|------------------|--------------------|--------------------|-------------------|------------------|
| Data from Host | 0x01       | 0x20      | tune(7:0)        | start_time (31:24) | start_time (23:16) | start_time (15:8) | start_time (7:0) |
| Data to Host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) | IrqStatus(23:16)   | IrqStatus (15:8)   | IrqStatus (7:0)   | 0x00             |

**tune** adjusts the output voltage on the TCXO supply pin, denoted as VTCXO. The tuning process follows values specified in [Table 6-65](#).

**Table 6-65: TCXO Supply Voltage Programming Values**

| Tune | TCXO Supply Voltage (typ) |
|------|---------------------------|
| 0x00 | 1.6 V                     |
| 0x01 | 1.7 V                     |
| 0x02 | 1.8 V                     |
| 0x03 | 2.2 V                     |
| 0x04 | 2.4 V                     |
| 0x05 | 2.7 V                     |
| 0x06 | 3.0 V                     |
| 0x07 | 3.3 V                     |

**start\_time** indicates the maximum duration for the 32 MHz oscillator to start and stabilize. The duration is measured in 32 MHz clock periods. If the 32 MHz oscillator from the TCXO is not detected internally at the end of the **start\_time** period, the device's internal firmware triggers an HF\_XOSC\_START\_ERR error.

- ♦ 0: (Default) disables TCXO mode

### 6.11.4 SetXoscCpTrim

The *SetXoscCpTrim* command allows the developer to trim the built-in foot capacitance of the 32 MHz Xtal oscillator. These trims are only used when going to a mode where the XOSC is enabled. If a TCXO is configured, this command has no effect, as the trims are not applied.

**Table 6-66: SetXoscCpTrim Command**

| Byte           | 0          | 1         | 2                    | 3                    | 4                          |
|----------------|------------|-----------|----------------------|----------------------|----------------------------|
| Data from Host | 0x01       | 0x31      | rfu(1:0)<br>xta(5:0) | rfu(1:0)<br>xtb(5:0) | additional_start_time(7:0) |
| Data to Host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24)     | IrqStatus(23:16)     | IrqStatus(15:7)            |

- **xta**: Trims the foot capacitor on XTA pin. 0 = 11.3 pF, max = 47 = 33.4 pF, 1 LSB = 0.47 pF. See [Table 6-67](#).
- **xtb**: Trims the foot capacitor on XTB pin. 0 = 11.1 pF, max = 47 = 33.2 pF, 1 LSB = 0.47 pF. See [Table 6-68](#).
- **additional\_start\_time**: (optional) Adds an additional delay in us for XTAL starting (RC->XSOC mode) to allow better stabilization of XTAL. If not sent, the previous value is kept. Default = 0.

**Table 6-67: XTA Capacitor Value**

| xta  | XTA Foot Capacitor (pF)            |
|------|------------------------------------|
| 0x00 | 11.3                               |
| ---  | ...                                |
| 0x05 | 13.6                               |
| 0x2F | 33.4 maximum implemented capacitor |

**Table 6-68: XTB Capacitor Value**

| xtb  | XTB Foot Capacitor (pF)            |
|------|------------------------------------|
| 0x00 | 11.1                               |
| ---  | ...                                |
| 0x05 | 13.45                              |
| 0x2F | 33.2 maximum implemented capacitor |

## 6.12 Temperature Compensation

The temperature compensation is useful to limit frequency drift during high power transmissions.

## 6.12.1 SetTempCompCfg

The *SetTempCompCfg* command configures the heating compensation block in Tx if an XTAL 32 MHz is used.

**Table 6-69: SetTempCompCfg Command**

| Byte           | 0          | 1         | 2                                 |
|----------------|------------|-----------|-----------------------------------|
| Data from Host | 0x01       | 0x32      | rfu(4:0)<br>ntc<br>comp_mode(1:0) |
| Data to Host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24)                  |

**ntc** controls the NTC source

- ♦ 1: Enables NTC
- ♦ 0: Disables NTC

**comp\_mode** defines the compensation mode:

- ♦ 0x0: Disabled
- ♦ 0x1: Relative
- ♦ 0x2: Absolute
- ♦ 0x3: RFU

If a TCXO was configured this command fails.

If an NTC source is available, it is used to compensate the variation in temperature of the crystal, while the internal temperature measurement can always be used to compensate the frequency deviation due to chip self heating.

## 6.12.2 SetNtcParams

The *SetNtcParams* command enters the NTC parameter to be used during heating compensation.

**Table 6-70: SetNtcParams Command**

| Byte           | 0          | 1         | 2                | 3                | 4               | 5              | 6           |
|----------------|------------|-----------|------------------|------------------|-----------------|----------------|-------------|
| Data from Host | 0x01       | 0x33      | ntc_r_ratio(9:8) | ntc_r_ratio(7:0) | ntc_beta(11:8)  | ntc_beta(7:0)  | delay (7:0) |
| Data to Host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) | IrqStatus(23:16) | IrqStatus(15:8) | IrqStatus(7:0) | 0           |

**ntc\_r\_ratio**: Defines the ratio between the resistor bias value and the NTC resistor value at 25 °C, on 10 bits with 9 fractional bits (10,9).

**ntc\_beta**: NTC temperature in 2 Kelvin/lb.

**delay**: First order time delay coefficient.

## 7. Radio Frequency Front End Control

### 7.1 Overview (LR2021/LR2022)

The LR2021/LR2022 is a half-duplex RF transceiver capable of handling constant envelope modulation schemes such as LoRa, (G)FSK, FLRC, OOK, O-QPSK, and LR-FHSS. It is fully compatible with the LR1110/20/21, SX1261/62/68/72/76 families for sub-GHz, and can be configured to be compatible with the SX1280/81 family for 2.4 GHz. FLRC modulation is compatible with the SX128x family.

The sub-GHz radio system comprises a frequency synthesizer (PLL), two Tx paths (high frequency and low frequency), and two Rx paths (LF and HF) followed by a high-bandwidth ADC. Both the ADC and the PLL are integrated with the digital subsystem.

The LR2021/LR2022 frequency synthesizer is clocked by a 32 MHz reference, provided either by a crystal oscillator or a TCXO.

Detailed descriptions of the LoRa, (G)FSK, FLRC, BPSK, OOK, O-QPSK, and LR-FHSS modulation and the PAs can be found in dedicated sections of the documentation.

### 7.2 Overview (LR2012)

The LR2012 is a half-duplex RF transceiver capable of handling constant envelope modulation schemes such as LoRa, (G)FSK, LR-FHSS, and BPSK. It is fully compatible with the LR1110/20/21, SX1261/62/68/72/76 families for sub-GHz.

The sub-GHz radio system comprises a frequency synthesizer (PLL), two Tx paths (high frequency and low frequency), and two Rx paths (LF and HF) followed by a high-bandwidth ADC. Both the ADC and the PLL are integrated with the digital subsystem.

The LR2012 frequency synthesizer is clocked by a 32 MHz reference, provided either by a crystal oscillator or a TCXO.

Detailed descriptions of the LoRa, (G)FSK, FLRC, and BPSK modulation and the PAs can be found in dedicated sections of the documentation.

### 7.3 RF Front End Commands (LR20xx)

#### 7.3.1 SetRfFrequency

The command *SetRfFrequency* sets the carrier frequency for transmit mode. When the radio operates in receive mode, the local oscillator is automatically set to the desired RF frequency minus the intermediated frequency (IF).

When this command is executed, the LR2021/LR2022/LR2012 automatically recomputes all frequency-dependent parameters, ensuring correct processing and configuration of the radio for the desired frequency.

**Table 7-1: SetRfFrequency Command**

| Byte           | 0          | 1         | 2                | 3                | 4                | 5               |
|----------------|------------|-----------|------------------|------------------|------------------|-----------------|
| Data from Host | 0x02       | 0x00      | rf_freq (31:24)  | rf_freq (23:16)  | rf_freq (15:8)   | rf_freq (7:0)   |
| Data to Host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) | IrqStatus(23:16) | IrqStatus (15:8) | IrqStatus (7:0) |

*rf\_freq* defines the RF Frequency of the radio, specified in Hertz (Hz).

### 7.3.2 SetRxPath

The *SetRxPath* command sets the Rx path to be used (LF or HF). In addition it defines if the receive path operates in boosted mode or not.

Note: LR2012 only supports LF, LR2021/LR2022 support LF and HF.

**Table 7-2: SetRxPath Command**

| Byte           | 0          | 1         | 2                    | 3                         |
|----------------|------------|-----------|----------------------|---------------------------|
| Data from host | 0x02       | 0x01      | rfu(6:0)<br>rx_path  | rfu(4:0)<br>rx_boost(2:0) |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus<br>(31:24) | IrqStatus<br>(23:16)      |

*rx\_path* selects Rx path:

- ♦ 0: Selects the low frequency path for Rx operation (LR20xx)
- ♦ 1: Selects the high frequency path for Rx operation (LR2021/LR2022)

*rx\_boost* (optional) enables Rx boost mode for Rx chain gain. Several boost levels can be defined by setting *rx\_boost* in a range from 0 to 7. If not defined, the previous or default value is used. The FW sets to 0 by default. The recommended default values are:

- ♦ 0: In LF
- ♦ 4: In HF

### 7.3.3 GetRssiInst

The *GetRssiInst* command returns the instantaneous RSSI (Received Signal Strength Indicator) value when NSS toggles up. In the absence of an RF packet, the RSSI value returned by the *GetRssiInst* command corresponds to the level of RF noise present in the environment. This allows developers to monitor the RF noise and assess the signal quality at any given moment.

**Table 7-3: GetRssiInst Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from Host | 0x02       | 0x0B      |
| Data to Host   | Stat(15:8) | Stat(7:0) |

**Table 7-4: GetRssiInst Response**

| Byte           | 0          | 1         | 2         | 3                   |
|----------------|------------|-----------|-----------|---------------------|
| Data from Host | 0x00       | 0x00      | 0x00      | 0x00                |
| Data to Host   | Stat(15:8) | Stat(7:0) | Rssi(8:1) | Rssi(0)<br>rfu(6:0) |

The RSSI (Received Signal Strength Indicator) is delivered on 9 bits, and is calculated using the following formula:  $\text{RSSI (dBm)} = -\text{Rssi} / 2$ .

If developers require only 1 dB resolution, reading of byte 3 is optional, and the RSSI value can be obtained directly from byte 2 of the response.

### 7.3.4 SetRssiCalibration

The *SetRssiCalibration* command sets the gain offset for the on-chip power estimation. The LR20xx has two internal LNAs, each with a set of predefined gains used to amplify the signal at the correct level, depending on the RF input power. The power seen by the LR20xx analog front-end is affected by external components such as the matching network, or RF switches.

- An incorrect RSSI results in a sensitivity degradation in (G)FSK mode and an incorrect gain selection in LoRa and GFSK mode.
- An incorrect gain can result in a missed detection (packet loss) or decreased resistance to interference.

By default, the chip is calibrated for the 868–915 MHz band on the LF path and 2.4 GHz band on the HF path. Both of these are computed for the LR20xx EVK reference designs seen in [23.4 Reference Design](#).

Therefore, to have a correct reading of the RSSI value of the chip integrated in the application PCB, the RSSI must be calibrated.

The RSSI must be calibrated for each hardware type, but not for each device individually: the RSSI offset must be determined once during the hardware debug or evaluation phase and then applied to all the devices sharing exactly the same hardware (identical schematic, layout and BOM).

**Table 7-5: SetRssiCalibration Command**

| Byte           | 0          | 1         | 2                                    | 3                         | 4               | 5              | ... |
|----------------|------------|-----------|--------------------------------------|---------------------------|-----------------|----------------|-----|
| Data from host | 0x02       | 0x05      | rfu(5:0)<br>rx_path_hf<br>rx_path_lf | rfu(5:0)<br>g1_gain (9:8) | g1_gain(7:0)    | g1_nf (7:0)    | ... |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24)                     | IrqStatus(23:16)          | IrqStatus(15:8) | IrqStatus(7:0) | 0x0 |

| Byte           | ... | 81                            | 82               | 83              |
|----------------|-----|-------------------------------|------------------|-----------------|
| Data from host | ... | rfu(5:0)<br>g13_gain_b7 (9:8) | g13_gain_b7(7:0) | g13_nf_b7 (7:0) |
| Data to host   | 0x0 | 0x0                           | 0x0              | 0x0             |

This command sets the RSSI calibration table for the selected Rx path.

If only either *rx\_path\_lf*, or *rx\_path\_hf* is set, only the table for the set path is updated.

If both bits are set, the values for both tables have to be sent as arguments. The LF path table must be sent first (bytes 3 to 83) followed by the HF path (bytes 84 to 164).

For each path, possible gains are: G1, G2, G3, G4, G5, G6, G7, G8, G9, G10, G11, G12\_boost0, G12\_boost1,..., G12\_boost7, G13\_boost0,..., G13\_boost7.

A table consists of 27 entries of Rx gains, each gain containing one GainValue and one NoiseFigure. Each of the 27 entries is encoded as:

- GainValue: front-end gain. Signed fixed-point, format 8.2 (1 LSB = 0.25 dB)
- NoiseFigure: input-referred noise figure. Unsigned fixed-point, format 6.2 (1 LSB = 0.25 dB)

Both tables (for LF and HF path) are automatically stored in the retention memory.

It is therefore typically expected to only call this function once, at boot up. If the application use case requires large frequency hops that would put the currently loaded table out of scope, the application code must call this function with the right table during runtime.

**Note:** Unlike values for 868-915 MHz and 2.4 GHz, which are in ROM and loaded by default, baseline values for the 490 MHz EVK **MUST** be loaded by the application.



**Table 7-6: Recommended Values for Reference EVK HF path 2.4 GHz (loaded by default)**

|             | G1  | G2  | G3  | G4  | G5  | G6  | G7  | G8  | G9  | G10 | G11 |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| GainValue   | 58  | 34  | 10  | 7   | 25  | 40  | 53  | 77  | 102 | 126 | 146 |
| NoiseFigure | 240 | 221 | 208 | 190 | 171 | 153 | 137 | 113 | 89  | 65  | 47  |

**G12:**

| G12:        | B0  | B1  | B2  | B3  | B4  | B5  | B6  | B7  |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|
| GainValue   | 163 | 166 | 168 | 170 | 171 | 172 | 176 | 175 |
| NoiseFigure | 34  | 32  | 31  | 30  | 30  | 29  | 26  | 27  |

**G13:**

| G13:        | B0  | B1  | B2  | B3  | B4  | B5  | B6  | B7  |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|
| GainValue   | 187 | 190 | 193 | 194 | 195 | 196 | 198 | 199 |
| NoiseFigure | 29  | 29  | 27  | 26  | 26  | 25  | 24  | 24  |

**Table 7-7: Recommended Values for Reference EVK LF path 868–915 MHz (loaded by default)**

|             | G1  | G2  | G3  | G4  | G5  | G6  | G7  | G8  | G9  | G10 | G11 |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| GainValue   | 32  | 8   | 16  | 36  | 56  | 73  | 90  | 112 | 130 | 152 | 173 |
| NoiseFigure | 219 | 204 | 194 | 175 | 156 | 137 | 122 | 95  | 62  | 41  | 26  |

**G12:**

| G12:        | B0  | B1  | B2  | B3  | B4  | B5  | B6  | B7  |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|
| GainValue   | 190 | 196 | 200 | 202 | 203 | 205 | 207 | 208 |
| NoiseFigure | 16  | 13  | 13  | 13  | 12  | 12  | 12  | 12  |

**G13:**

| G13:        | B0  | B1  | B2  | B3  | B4  | B5  | B6  | B7  |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|
| GainValue   | 214 | 220 | 223 | 226 | 227 | 228 | 230 | 232 |
| NoiseFigure | 12  | 11  | 11  | 10  | 10  | 11  | 10  | 9   |

**Table 7-8: Recommended Values for Reference EVK LF path 490 MHz (loaded by the application)**

|             | G1  | G2  | G3  | G4  | G5  | G6  | G7  | G8  | G9  | G10 | G11 |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| GainValue   | 46  | 22  | 2   | 24  | 44  | 63  | 84  | 104 | 138 | 158 | 181 |
| NoiseFigure | 221 | 205 | 190 | 169 | 147 | 130 | 113 | 87  | 46  | 30  | 18  |

**G12:**

| G12:        | B0  | B1  | B2  | B3  | B4  | B5  | B6  | B7  |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|
| GainValue   | 200 | 196 | 208 | 210 | 211 | 215 | 217 | 218 |
| NoiseFigure | 13  | 13  | 13  | 13  | 13  | 12  | 12  | 12  |

**G13:**

| G13:        | B0  | B1  | B2  | B3  | B4  | B5  | B6  | B7  |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|
| GainValue   | 224 | 228 | 231 | 234 | 237 | 238 | 240 | 242 |
| NoiseFigure | 12  | 12  | 12  | 12  | 11  | 11  | 10  | 10  |

### 7.3.4.1 How to Compute Values for New Hardware

Before following the gain and noise-figure calibration steps below, ensure that you:

- Use an RF signal generator connected directly to the receiver input.
- Account for all cable losses.
- Calibrate one table per RF band and always perform gain calibration before noise figure calibration.

**Note:**

***GetRssiInst* (opcode 0x020B) reads the instantaneous RSSI.**

***SetAgcGainManual* (opcode 0x021A) forces a manual gain step, setting a value 0 re-enables the automatic gain.**

#### 7.3.4.1.1 Gain calibration

Calibrate for each step G1 ... G13, and each boost level for G12/G13:

1. Setup the device in GFSK mode and set the desired RF frequency.
2. Force the step with *SetAgcGainManual*(step). Loop through all the steps. Boost levels are selected via the Rx path / RxBoost configuration. Ensure no unwanted signal is received.
3. Apply a CW tone of known power P [dBm] at the RF center frequency. P must be below saturation and well above the noise floor. A good rule of thumb is  $P = -10 - (\text{gain\_step} * 6)$
4. Read the channel RSSI with *GetRssiInst*() multiple times and get the median value.
5. The error for that step is  $\text{error} = P - \text{RSSI}$ . Offset the gain by that amount:  $\text{gain\_dB} = \text{gain\_dB} + \text{error}$ . In the raw domain (actual value that needs to be stored in the table),  $\text{GainValue} = \text{GainValue} + \text{round}(\text{error} * 4)$ .

Re-upload the corrected table and loop again to verify the residual error is within  $\pm 0.5$  dB on every step.

#### 7.3.4.1.2 2. Noise-figure calibration

This step assumes the gains are already correct. With no signal applied, for each step G1 ... G13 and each boost level for G12/G13:

1. Setup the device in GFSK mode and set the desired RF frequency.
2. Use the *SetRssiCalibration* API to set all gain values to the correct ones computed in the step above. Set all noise figures to 0.
3. Force the step with *SetAgcGainManual*(step). Ensure the channel is quiet, a screened environment is required.
4. Measure the channel RSSI with *GetRssiInst*() many times (100) and take the median.
5. Convert the measured noise power to a dB/Hz noise figure:  

$$\text{NF\_dB} = \text{RSSI\_dBm} + 174 + 10 * \log_{10}(\text{BW\_Hz}) + B$$
  - ♦ where: RSSI\_dBm: the returned median RSSI from step 4.
  - BW\_Hz: the channel (Rx) bandwidth in Hz.
  - 174: thermal-noise term (kT in dBm/Hz at room temperature).
  - B: a white-noise correction factor, B = 1 dB.
6. Write the entry:  $\text{NoiseFigure} = \text{round}(\text{NF\_dB} * 4)$ .

Re-upload the table and perform a sanity check: every NF should be  $> 1$  dB.

### 7.3.5 ResetRxStats

The *ResetRxStats* command resets the internal statistics related to the received RF packets. Refer to Get<modulation>RxStats function for each modulation.

**Table 7-9: ResetRxStats Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from Host | 0x02       | 0x0A      |
| Data to Host   | Stat(15:8) | Stat(7:0) |

### 7.3.6 SetTimestampSource

The *SetTimestampSource* command sets the source event for time-stamping various radio events, this feature can be useful for protocols such as LoRaWAN where a degree of synchronization is required.

**Table 7-10: SetTimestampSource Command**

| Byte           | 0          | 1         | 2                                     |
|----------------|------------|-----------|---------------------------------------|
| Data from host | 0x02       | 0x16      | rfu(1:0)<br>Index(1:0)<br>Source(3:0) |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus (31:24)                     |

The *SetTimestampSource* command can configure three sources in parallel for time-stamping.

Each **Source** is identified by an **Index** (0 to 2) and can be assigned one of the following values:

- ♦ 0x0: NONE
- ♦ 0x1: LAST\_BIT\_SENT
- ♦ 0x2: LAST\_BIT\_RECEIVED
- ♦ 0x3: FSK\_SYNC\_DETECTED (valid for any FSK modulation including W-MBUS and Bluetooth LE)
- ♦ 0x4: HEADER\_DETECTED (LoRa only)

### 7.3.7 GetTimestampValue

The *GetTimestampValue* command provides the delay in HF clock ticks (32 MHz) between the event and the SPI NSS falling edge of the request for the source specified in the **Index** parameter. Each source defined in *SetTimestampSource* can be mapped to an IRQ as described in [Table 5-20](#).

**Table 7-11: GetTimestampValue Command**

| Byte           | 0          | 1         | 2                      |
|----------------|------------|-----------|------------------------|
| Data from host | 0x02       | 0x17      | rfu(5:0)<br>Index(1:0) |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus (31:24)      |

**Table 7-12: GetTimestampValue Response**

| Byte           | 0          | 1         | 2                | 3                 | 4                | 5               |
|----------------|------------|-----------|------------------|-------------------|------------------|-----------------|
| Data from host | 0x00       | 0x00      | 0x00             | 0x00              | 0x00             | 0x00            |
| Data to host   | Stat(15:8) | Stat(7:0) | Timestamp(31:24) | Timestamp (23:16) | Timestamp (15:8) | Timestamp (7:0) |

**Timestamp:** Time in periods of 32 MHz clock.

### 7.3.8 GetCcaResult

The *GetCcaResult* command returns the results from the previous clear channel assessment (CCA).

**Table 7-13: GetCcaResult Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x02       | 0x19      |
| Data to host   | Stat(15:8) | Stat(7:0) |

**Table 7-14: GetCcaResult Response**

| Byte           | 0          | 1         | 2            | 3            | 4            | 5                                                  |
|----------------|------------|-----------|--------------|--------------|--------------|----------------------------------------------------|
| Data from host | 0x00       | 0x00      | 0x00         | 0x00         | 0x00         | 0x00                                               |
| Data to host   | Stat(15:8) | Stat(7:0) | RssiMin(8:1) | RssiMax(8:1) | Rssiavg(8:1) | rfu(4:0)<br>RssiMin(0)<br>RssiMax(0)<br>Rssiavg(0) |

The *GetCcaResult* command retrieves the following information from the previous Clear Channel Assessment (CCA):

- ♦ **RssiMin:** Minimum RSSI observed during CCA (in -0.5 dB steps)
- ♦ **RssiMax:** Maximum RSSI observed during CCA (in -0.5 dB steps)
- ♦ **Rssiavg:** Average RSSI observed during CCA (in -0.5 dB steps)

**Note:** If a CCA is stopped before the end using a *SetFs* or *SetStandby* command, the CCA values are not saved, and the chip returns the values of the previous CCA. Values are lost during Sleep mode.

1 dB resolution, which is usually sufficient, is obtained when the 5th parameter byte is not read, speeding up the command.

### 7.3.9 SetAgcGainManual

The *SetAgcGainManual* command sets the manual gain step of the Rx chain in HF or LF path.

**Table 7-15: SetAgcGainManual Command**

| Byte           | 0          | 1         | 2                         |
|----------------|------------|-----------|---------------------------|
| Data from host | 0x02       | 0x1A      | rfu(3:0)<br>GainStep(3:0) |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus (31:24)         |

**GainStep** sets the value of the Rx gain in a range from 0 to 13. A value of 0 enables the AGC.

### 7.3.10 GetRxPktLength

The command *GetRxPktLength* gets the payload length of the last received payload.

**Table 7-16: GetRxPktLength Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x02       | 0x12      |
| Data to host   | Stat(15:8) | Stat(7:0) |

**Table 7-17: GetRxPktLength Response**

| Byte           | 0          | 1         | 2             | 3            |
|----------------|------------|-----------|---------------|--------------|
| Data from host | 0x00       | 0x00      | 0x00          | 0x00         |
| Data to host   | Stat(15:8) | Stat(7:0) | pkt_len(15:8) | pkt_len(7:0) |

**PktLen**: Length of the last payload (in bytes).

**Note:** Z-Wave, Wi-SUN and 15.4 Packet handlers return the length of the payload plus CRC, in this case, the CRC is also available in the FIFO.

## 7.4 Power Amplifier Control Commands (LR20xx)

The LR2021/LR2022/LR2012 transceiver is equipped with two power amplifiers (PAs) optimized for different frequency bands:

- ♦ High Frequency PA (PA\_HF) (LR2021/LR2022): This PA is optimized for operation in the 2.4 GHz band and can achieve an output power of up to +12 dBm.
- ♦ Low Frequency PA (PA\_LF) (LR20xx): This PA is designed for operation in the sub-GHz frequency range and is capable of reaching up to +22 dBm output power.

### 7.4.1 SetPaConfig

The *SetPaConfig* command selects which power amplifier (PA) to use and configures its parameters.

**Table 7-18: SetPaConfig Command**

| Byte           | 0          | 1         | 2                                     | 3                                          | 4                                 |
|----------------|------------|-----------|---------------------------------------|--------------------------------------------|-----------------------------------|
| Data from host | 0x02       | 0x02      | pa_sel<br>rfu(4:0)<br>pa_lf_mode(1:0) | pa_lf_duty_cycle(3:0)<br>pa_lf_slices(3:0) | rfu(2:0)<br>pa_hf_duty_cycle(4:0) |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24)                      | IrqStatus(23:16)                           | IrqStatus(15:8)                   |

**pa\_sel** selects the desired PA:

- ♦ 0: LF PA is used
- ♦ 1: HF PA is used

**pa\_lf\_mode** sets the PA\_LF mode of operation.

- ♦ 0x0 (default): LF PA Full single-ended mode (FSM). This option selects the power amplifier optimized for low-frequency operating where 2 PAs branch running in phase.
- ♦ 0x1: RFU
- ♦ 0x2: RFU
- ♦ 0x3: RFU

**pa\_lf\_duty\_cycle** controls the duty cycle and maximum output power of the PA. If LF PA is not used, set the parameter to 6. See [Table 7-19](#) and [Table 7-20](#).

**pa\_lf\_slices**. If LF PA is not used, then set this parameter to 7. See [Table 7-19](#) and [Table 7-20](#).

**pa\_hf\_duty\_cycle** controls the duty cycle and the maximum output power of the PA in HF mode. (LR2021/22)

- ♦ Only values from 16–31 are authorized to avoid risk of aging the power amplifier.
- ♦ If the HF PA is not used, set the parameter to 16 (default).
- ♦ To be used for adapting the Tx multi-band operation using a single-matching network. See [Table 7-21](#).

Using non-optimized PA parameters may result in:

- Incorrect output power
- Excessive current consumption
- PA damage
- Regulatory non-compliance

For designs differing from reference design, contact Semtech applications support.

Table 7-19: Optimal Values for 915 MHz Semtech Reference Design

| Targeted Power [dBm] | TX_POWER | PA_LF_DUTY_CYCLE | PA_LF_SLICES |
|----------------------|----------|------------------|--------------|
| 22                   | 22       | 7                | 7            |
| 21.5                 | 22       | 6                | 7            |
| 21                   | 22       | 5                | 6            |
| 20.5                 | 21       | 6                | 7            |
| 20                   | 21       | 5                | 6            |
| 19.5                 | 21       | 4                | 7            |
| 19                   | 20       | 5                | 7            |
| 18.5                 | 20       | 4                | 7            |
| 18                   | 19       | 5                | 7            |
| 17.5                 | 19       | 5                | 4            |
| 17                   | 18       | 7                | 3            |
| 16.5                 | 18       | 5                | 4            |
| 16                   | 17       | 7                | 3            |
| 15.5                 | 18       | 4                | 3            |
| 15                   | 17       | 4                | 5            |
| 14.5                 | 17       | 4                | 3            |
| 14                   | 17       | 4                | 2            |
| 13.5                 | 16       | 4                | 3            |
| 13                   | 16       | 4                | 2            |
| 12.5                 | 15       | 5                | 2            |
| 12                   | 15       | 4                | 2            |
| 11.5                 | 14       | 5                | 2            |
| 11                   | 15       | 2                | 4            |
| 10.5                 | 17       | 4                | 0            |
| 10                   | 16       | 1                | 2            |

Table 7-20: Optimal Values for 490 MHz Semtech Reference Design

| Targeted Power [dBm] | TX_POWER | PA_LF_DUTY_CYCLE | PA_LF_SLICES |
|----------------------|----------|------------------|--------------|
| 21                   | 22       | 7                | 7            |
| 20.5                 | 22       | 7                | 4            |
| 20                   | 21       | 7                | 7            |
| 19.5                 | 21       | 7                | 4            |
| 19                   | 20       | 7                | 6            |
| 18.5                 | 22       | 6                | 2            |
| 18                   | 19       | 7                | 6            |
| 17.5                 | 19       | 7                | 3            |
| 17                   | 19       | 6                | 5            |
| 16.5                 | 18       | 6                | 7            |
| 16                   | 18       | 6                | 5            |
| 15.5                 | 17       | 6                | 7            |
| 15                   | 16       | 7                | 5            |
| 14.5                 | 16       | 6                | 7            |
| 14                   | 15       | 7                | 5            |
| 13.5                 | 15       | 7                | 3            |
| 13                   | 15       | 7                | 2            |
| 12.5                 | 15       | 6                | 3            |
| 12                   | 15       | 6                | 2            |
| 11.5                 | 15       | 5                | 3            |
| 11                   | 15       | 4                | 5            |
| 10.5                 | 17       | 6                | 0            |
| 10                   | 15       | 5                | 1            |



Table 7-21: Optimal Values for 2445 MHz Semtech Reference Design (LR2021/LR2022)

| Targeted Power [dBm] | TX_POWER | PA_HF_DUTY_CYCLE |
|----------------------|----------|------------------|
| 12                   | 12       | 16               |
| 11.5                 | 12       | 21               |
| 11                   | 12       | 27               |
| 10.5                 | 11       | 21               |
| 10                   | 12       | 30               |
| 9.5                  | 11       | 29               |
| 9                    | 11       | 30               |
| 8.5                  | 10       | 29               |
| 8                    | 10       | 30               |
| 7.5                  | 9        | 29               |
| 7                    | 8        | 27               |
| 6.5                  | 8        | 29               |
| 6                    | 7        | 27               |
| 5.5                  | 8        | 31               |
| 5                    | 6        | 27               |
| 4.5                  | 7        | 31               |
| 4                    | 5        | 27               |
| 3.5                  | 6        | 31               |
| 3                    | 4        | 27               |
| 2.5                  | 5        | 31               |
| 2                    | 3        | 27               |
| 1.5                  | 4        | 31               |
| 1                    | 2        | 28               |
| 0.5                  | 1        | 24               |
| 0                    | 1        | 28               |

## 7.4.2 SelPa

The *SelPa* command selects which PA to use. This command allows to switch between PAs that have already been configured using the *SetPaConfig* command (*SetPaConfig* must be called before any call to *SelPa*). Using *SelPa* is quicker than using *SetPaConfig* once the PA has been configured.

**Table 7-22: SelPa Command**

| Byte           | 0          | 1         | 2                 |
|----------------|------------|-----------|-------------------|
| Data from host | 0x02       | 0x0F      | rfu(6:0) pa_sel   |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus (31:24) |

*pa\_sel* defines which PA to use:

- ♦ 0: Low power (LF) PA is used
- ♦ 1: High power (HF) PA is used (LR2021/LR2022)

Selection cannot be changed during Tx mode: in this case a CMD\_FAIL is returned in the status of the next command.

### 7.4.3 SetTxParams

The *SetTxParams* command sets the Transmit (Tx) power and ramp time of the Power Amplifier (PA). Before using this command, the *SetPaConfig* command must be called, as it configures the relevant registers and ensures that the power is correctly set up for the specific PA configuration. Consult the *SetPaConfig* command since limitations apply on power depending on PA supply option.

**Table 7-23: SetTxParams Command**

| Byte           | 0          | 1         | 2                | 3                |
|----------------|------------|-----------|------------------|------------------|
| Data from host | 0x02       | 0x03      | tx_power(7:0)    | ramp_time(7:0)   |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) | IrqStatus(23:16) |

**tx\_power** sets the Tx power at PA output (HF or LF) in +0.5 dB steps:

- ♦ PA\_LF: [-19: 44] corresponding to output power range of [-9.5:22] dBm
- ♦ PA\_HF: [-39: 24] corresponding to output power range of [-19.5:12] dBm (LR2021/22)

**ramp\_time** sets the ramp time for both PAs as defined in the next table. It is subject to local regulations and Tx parameters:

**Table 7-24: Tx Ramp\_time Parameter Definition**

| ramp_time | PA Ramp Time (μs) |
|-----------|-------------------|
| 0x00      | 2                 |
| 0x01      | 4                 |
| 0x02      | 8                 |
| 0x03      | 16                |
| 0x04      | 32                |
| 0x05      | 48                |
| 0x06      | 64                |
| 0x07      | 80                |
| 0x08      | 96                |
| 0x09      | 112               |
| 0x0A      | 128               |
| 0x0B      | 144               |
| 0x0C      | 160               |
| 0x0D      | 176               |
| 0x0E      | 192               |
| 0x0F      | 208               |
| 0x10      | 240               |
| 0x11      | 272               |
| 0x12      | 304               |

In addition to the power ranges defined above, LF PA can output extra power steps as low as -22 dBm to support standards like Z-Wave. Contact Semtech for details about this feature.

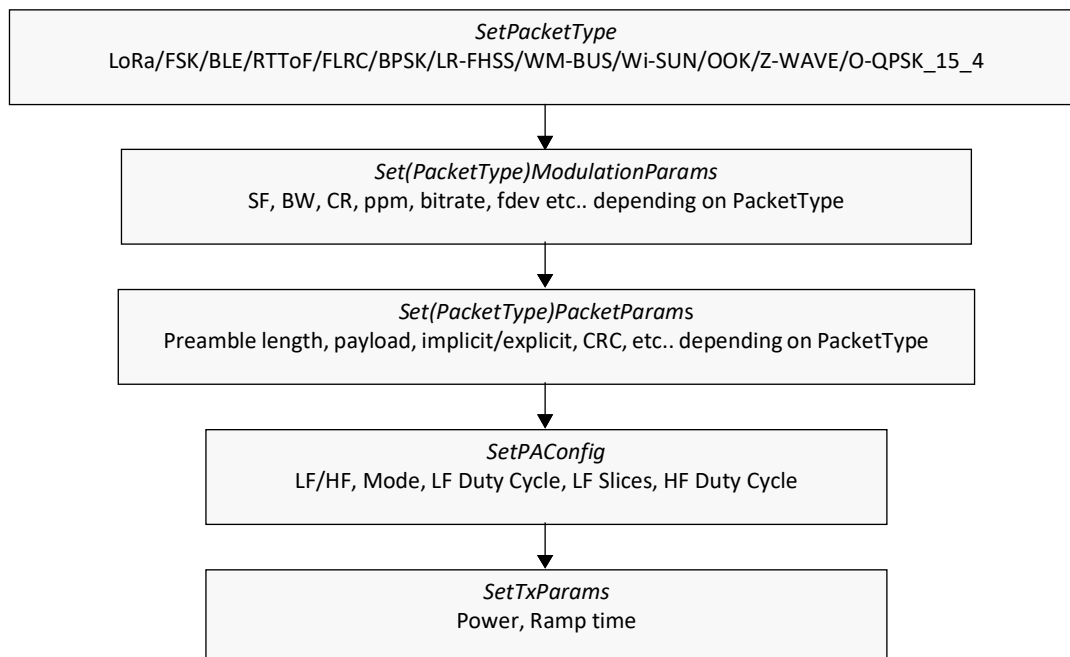
## 8. Modem Configuration (LR20xx)

The LR2021/LR2022/LR2012 incorporates various modems that support different constant envelope modulations. To be configured appropriately, the LR2021/LR2022/LR2012 must receive the following commands:

- ♦ *SetPacketType*: The developer needs to specify the desired packet type using this command. The packet type selection determines the modulation type, modulation parameters, and packet handler settings.
- ♦ *Set(PacketType)ModulationParams*: This command configures the specific modem parameters for the chosen packet type. Each packet type has its own corresponding command for setting the modulation parameters.
- ♦ *Set(PacketType)PacketParams*: This command defines the RF packet parameters, such as payload length, implicit/explicit mode, etc. It should be noted that this command is not necessary for LR-FHSS modulation. Similar to the modulation parameters, there is a specific command for each packet type to configure the packet parameters.
- ♦ *SetPaConfig*: This command configures the Power Amplifier (PA) settings used during RF packet transmission. It includes options for selecting the appropriate PA, supply mode, and other related configurations.
- ♦ *SetTxParams*: The final step involves defining the PA parameters specific to the transmit operation. This includes setting the output power and ramp time for the PA.

**Note:** The command order is not important provided that at least the *SetPacketType* is sent first.

**Figure 8-1: Modem Control Commands**



Apart from the general commands mentioned earlier for setting up modems and packet parameters, the LR2021/LR2022/LR2012 offers more specific commands to configure different modes that use specific modulations, such as CAD (Channel Activity Detection), ranging, etc.

## 8.1 Packet Type Selection

### 8.1.1 SetPacketType

The *SetPacketType* command in the LR2021/LR2022/LR2012 defines the modulation scheme and packet handler parameters to be used for data transmission and reception. The *SetPacketType* command must be the first command sent when setting up the transceiver operation.

This command only works when the chip is in Standby RC, Standby Xosc, or Fs (Frequency Synthesizer) mode. If the chip is in any other mode, the command returns CMD\_FAIL in the status of the next command.

**Table 8-1: SetPacketType Command**

| Byte           | 0          | 1         | 2                 |
|----------------|------------|-----------|-------------------|
| Data from Host | 0x02       | 0x07      | PacketType        |
| Data to Host   | Stat(15:8) | Stat(7:0) | IrqStatus (31:24) |

**PacketType** specifies the current packet type supported by a dedicated modem in the LR2021/LR2022/LR2012 transceiver. The allowed values for this parameter are as follows (when applicable on the chosen product):

- ♦ 0x0: LoRa
- ♦ 0x1: RFU
- ♦ 0x2: FSK
- ♦ 0x3: BLE (Bluetooth Low Energy)
- ♦ 0x4: RTToF (Round-Trip Time of Flight)
- ♦ 0x5: FLRC
- ♦ 0x6: BPSK
- ♦ 0x7: LR-FHSS
- ♦ 0x8: WM-BUS
- ♦ 0x9: Wi-SUN
- ♦ 0xA: OOK
- ♦ 0xB: RFU
- ♦ 0xC: Z-WAVE
- ♦ 0xD: O-QPSK\_15\_4

## 8.1.2 GetPacketType

The *GetPacketType* command retrieves the current packet type that is being used by the LR2021/LR2022/LR2012 radio.

**Table 8-2: GetPacketType Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from Host | 0x02       | 0x08      |
| Data to Host   | Stat(15:8) | Stat(7:0) |

**Table 8-3: GetPacketType Response**

| Byte           | 0          | 1         | 2          |
|----------------|------------|-----------|------------|
| Data from Host | 0x00       | 0x00      | 0x00       |
| Data to Host   | Stat(15:8) | Stat(7:0) | PacketType |

## 9. LoRa Modulation (LR20xx)

### 9.1 LoRa Modulation Principle

The LoRa modem employs a proprietary spread spectrum modulation. This proprietary modulation allows for an augmented link budget and enhanced immunity to in-band interference. Additionally, the modem exhibits the capability to receive signals even with negative Signal-to-Noise Ratio (SNR), leading to improved sensitivity, extended link budget, and increased range for the LoRa receiver.

#### 9.1.1 Spreading Factor (SF)

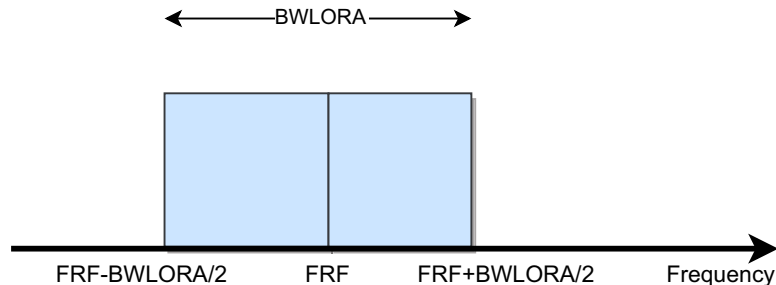
The spread spectrum LoRa modulation operates by encoding each bit of payload information using multiple chips of information. The symbol rate ( $R_s$ ) denotes the rate at which a spread symbol, comprising  $2^{SF}$  chips, is transmitted. The Spreading Factor (SF) is the ratio between the nominal symbol rate and chip rate and determines the number of bits sent per symbol.

**Note:** The spreading factor must be predetermined and known on both the transmit and receive sides of the link. This is crucial because different spreading factors are orthogonal to each other.

#### 9.1.2 LoRa Bandwidth (BWLORA)

The LoRa modem functions with a customizable bandwidth (BWLORA) centered around a programmable frequency, denoted as fRF. The LoRa modem bandwidth always refers to the double sideband (DSB), as illustrated in [Figure 9-1](#).

**Figure 9-1: LoRa Signal Bandwidth**



An increase in signal bandwidth enables the utilization of a higher effective bit rate, resulting in reduced transmission time. However, this comes at the cost of reduced sensitivity, as the signal becomes more susceptible to noise.

In most countries, there are regulatory constraints governing the permissible occupied bandwidth. Consequently, only a subset of the programmable bandwidth (BWLORA) can be used to comply with these regulations.

#### 9.1.3 Coding Rate (CR)

To enhance the reliability of the link, the LoRa modem incorporates cyclic error coding, which enables forward error detection and correction. However, this error coding introduces a transmission overhead.

#### 9.1.4 Low Data Rate Optimization (LDRO)

Low Data Rate Optimization (LDRO) is employed to enhance the reliability of the LoRa link, particularly at low effective data rates. By utilizing LDRO, the sensitivity level is improved, and the link becomes more resilient to frequency drift and Doppler events.

### 9.1.5 LoRa Symbol Rate (Rs)

By considering the key parameters that developers can control, the LoRa symbol rate is defined as follows:

$$\text{Symbol rate} = \text{BWLORA} / (2^{\text{SF}})$$

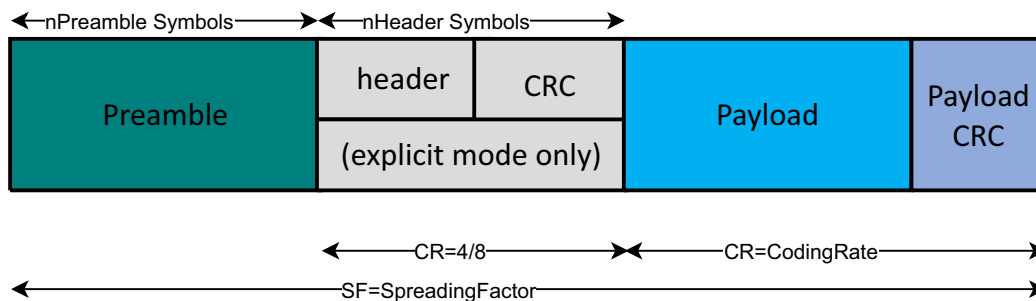
where BWLORA represents the programmed bandwidth and SF is the spreading factor. The transmitted signal is a constant envelope signal, and equivalently, one chip is sent per second per Hertz (Hz) of bandwidth. Understanding and manipulating these parameters allows developers to tailor LoRa communication to their specific requirements and optimize performance.

## 9.2 LoRa Packet Handler

The LoRa modem supports two packet formats: explicit and implicit. The explicit packet format includes a short header containing the number of payload bytes, coding rate configuration, and the presence of a CRC field in the packet.

The implicit packet format does not contain the header. This reduces the packet's time-on-air, but requires transmitter and receiver(s) to be configured identically.

**Figure 9-2: LoRa Packet Format**



### 9.2.1 Preamble

The LoRa packet begins with a preamble sequence, which serves to synchronize the receiver with the incoming signal. The length of the transmitted preamble can vary between 1 and 65535 symbols.

To optimize the reception of packets, use a minimum preamble length of 12 symbols for Spreading Factors (SF) 5 and 6, and a length of 8 symbols for other SF settings.

The receiver performs a periodic preamble detection process, which necessitates the configuration of an identical preamble length on both the transmitter and receiver sides. In cases where the preamble length is not known or can vary, it is advisable to program the maximum preamble length on the receiver side to ensure successful synchronization with incoming packets.

### 9.2.2 Header (explicit packets only)

The LoRa header serves as a carrier of payload information:

- ♦ The payload length expressed in bytes.
- ♦ The forward error correction coding rate.
- ♦ An optional 16-bit CRC for the payload.

The header is transmitted with maximum error correction code (4/8).

Additionally, the header itself may include its own CRC, enabling the receiver to identify and discard any invalid headers.



### 9.2.3 Payload

The LoRa packet payload consists of a variable-length field that contains the actual data. The data is coded at a rate specified in the header for explicit mode or in the register settings for implicit mode.

Additionally, an optional CRC may be appended to the payload to provide an extra layer of error detection and ensure data integrity during transmission.

## 9.3 Multi-SF Detection

The chips offer the capability of parallel detection for multiple spreading factors simultaneously. Among these spreading factors, the one set in [SetLoraModulationParams](#) is considered the main spreading factor. This main spreading factor is used for functions such as timeout processing during detection and general settings.

- For LoRa, the main detector must be programmed to receive the smallest spreading factor that the receiver is expected to handle.
- For CAD multi-SF the main SF must be the largest spreading factor that the receiver is expected to handle.

The first SF to do a detection is the one that takes over and does the reception. The other spreading factors can be detected in parallel to the main SF by the Side-detectors. They must be on the same channel and bandwidth. Each Side-detector has its own set of adjustable parameters, which include:

- ♦ Spreading Factor
- ♦ Symbol Direction
- ♦ Low Data Rate Optimization (LDRO)

In summary, the following restrictions apply:

- ♦ The smallest SF must be set on the main detector by using the base API for LoRa (such as [SetLoraModulationParams](#)). This rule does not apply to CAD Multi-SF, where the opposite should be done!
- ♦ All the SF must be different.
- ♦ The SF\_max - SF\_min must be less than or equal to 4.
- ♦ Each active SF detector consumes an internal processing resource proportional to a Detection Factor that increases with SF. For SF5 through SF12, the Detection Factors are 10, 10, 12, 12, 14, 14, 16, and 16, respectively. The total resource consumption — computed as the sum of the Detection Factors for all active SFs, multiplied by the channel bandwidth in Hz — must remain strictly below  $32 \times 10^6$ . For example, enabling SF11 and SF12 simultaneously at BW = 1000 kHz is not permitted, because  $(16 + 16) \times 1 \times 10^6 = 32 \times 10^6$ , which equals the limit.

## 9.4 Channel Activity Detection (CAD)

The use of a spread spectrum modulation technique presents challenges in determining whether the channel is already in use by a signal that may be below the noise floor of the receiver. The use of the RSSI in this situation would clearly be impracticable. To this end the channel activity detector is used to detect the presence of other LoRa signals.

The chip offers flexible configuration options for LoRa channel activity detection (CAD). Developers can configure the chip to listen for a fixed duration, specified in the number of symbols, to detect the presence of LoRa symbols on the channel. There are settings available to choose between detecting only the preamble or any LoRa symbol on the channel.

The CAD detection can be performed in two modes:

- ♦ A fixed user-defined amount of symbols.
- ♦ A best-effort mode, where CAD stops as soon as a clear result of non-detection is obtained otherwise it continues until the user-defined CAD length. This option results in a minimal CAD window when there is no activity in the channel.

Once the CAD operation is completed, the interrupt `CadDone` is asserted. If LoRa symbols were detected during the CAD process, the interrupt `CadDetected` is also asserted.

Several possibilities exist to exit from CAD mode, as defined in `exit_mode` of [SetCadParams](#):

- Transition to fallback mode defined by [SetRxTxFallbackMode](#) mode if only CAD is required
- Continue reception
- Start transmission (for example to perform LBT)

If the CAD ends up with `CadDetected` IRQ high, indicating the presence of LoRa symbols, the chip continues reception.

If the CAD was initially looking for the preamble, the chip starts searching for the LoRa Syncword (see [Section 9.9.5 SetLoraSyncword](#)).

If the CAD was searching for any LoRa symbol, it transitions to Rx mode and start seeking the LoRa preamble.

## 9.5 Multi-SF CAD

In the case of multi-SF CAD, the result of each detector is available using command [GetLoraPacketStatus](#).

The CAD duration is determined by the main Spreading Factor (SF) setting, which means that lower SF is executed multiple times. There are two modes to capture the CAD status:

- ♦ The CAD status of a detector is 1 if at least one run detected activity. This mode is typically used for general network activity monitoring, where any detection during any run is sufficient to indicate channel activity.
- ♦ The CAD status corresponds to the last run of CAD. This mode is useful for the listen-before-talk use-case, where the CAD status reflects the outcome of the final run, providing a more precise indication of channel activity.

**Note: In case of CAD to Rx transition:**

- If CAD multi-SF is configured to detect preamble only, the LoRa demodulator will start synchronization using the SF found during the CAD multi SF detection process.
- If CAD multi-SF is configured to detect any LoRa symbol, then the LoRa demodulator will start a multi-SF detection.

## 9.6 Fast CAD

Traditional CAD on previous LoRa generations (SX126x, LR11xx) uses a fixed peak-to-noise ratio threshold across all symbols in the detection window. This requires the full configured symbol count (e.g., 5 symbols) to complete before declaring no activity detected, consuming unnecessary power when the channel is idle.

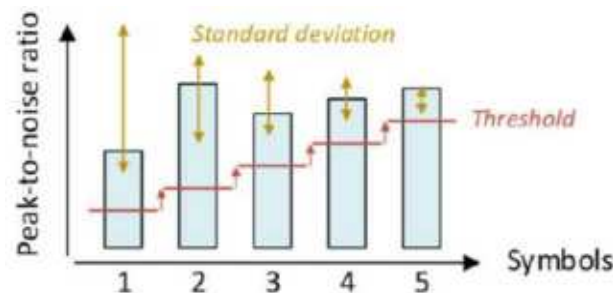
The LR2021/LR2022/LR2012 Fast CAD feature implements an adaptive threshold mechanism that progressively increases the peak-to-noise detection threshold for each successive symbol, as shown in [Figure 9-3](#) below, where each blue bar represents the measured peak-to-noise ratio of the received signal for that symbol, with the orange arrows indicating the standard deviation of the noise floor used to establish the adaptive threshold at each step.

When the measured peak-to-noise ratio falls below the progressively rising threshold for any symbol (indicating clear absence of LoRa activity), the CAD process terminates early without processing the remaining symbols. The starting detection peak is set to ``cad_det_peak` - `pnr_delta` * (n-1)`, where "n" starts at ``cad_symbol_num`` and increases with each detection period.

This optimization reduces average CAD duration from ~5 symbols to approximately as low as 1.5 symbols when no LoRa signal is present, which can lead to as much as ~70% power consumption reduction for Listen-Before-Talk (LBT) and Carrier Sense Multiple Access (CSMA) applications that perform frequent channel sensing. A more typical power saving is 30%-50%.

The feature is activated via the `pnr_delta` parameter in the [SetLoraCadParams](#) command, with detection sensitivity maintained within 3 dB of the receiver's maximum sensitivity threshold.

**Figure 9-3: LoRa Fast CAD Principle**



## 9.7 Extended LoRa Rx Bandwidth

One of the key benefits of LoRa modulation is its ability to reach best-in-class sensitivity even when there is significant frequency offset between two ends of the link. As tabulated in specification LORA\_FERR\_L1 (see [Table 3-18](#)), LoRa links can cope with up to 31 kHz of frequency misalignment (for the nominal 125 kHz bandwidth). This unique feature makes it possible to use a precise frequency reference source (typically a TCXO) only on one end of the link. In LoRaWAN scenarios, more expensive gateways have a precise TCXO reference, and end-devices can use a cheap 30 ppm XTAL, preserving the low Total Cost of Ownership (TCO) for the system.

The LR2021/LR2022/LR2012 expands further this capability by offering up to +/-33% of LoRa BW frequency tolerance, pushing the limit to +/-41.25 kHz of frequency lenience for 125 kHz BW. The extended LoRa frequency error tracking is enabled as follows:

```
// Configure mode at address 0xF30A2C, bits [17:16]
WriteRegMemMask32(addr = 0xf30A2C, Mask = (3 << 16), Value = (mode << 16));
// where mode = 0 (legacy) or 1 (extended)
```

## 9.8 Intra-packet Frequency Hopping Compatibility with SX127x

LoRa intra-packet frequency hopping is an advanced feature where a single LoRa packet transmission switches between different frequencies during the packet transmission.

The newer LR20xx transceivers implement frequency hopping differently from the older SX127x chips. This creates incompatibility when trying to communicate between SX127x-based and LR20xx-based devices using frequency hopping. The fundamental issue is that the internal timing, frequency switching mechanisms, and control logic evolved between the chip generations, making them unable to properly synchronize their hopping sequences.

The LR2021/LR2022/LR2012 includes a specific compatibility mode for frequency hopping that makes it behave exactly like an SX1276 when performing intra-packet hopping.

Implementation Steps:

1. Configure Basic LoRa Hopping: Call [SetLoraHopping](#) to set up your hopping parameters:
  - ♦ *hop\_ctrl*: Enable hopping (1)
  - ♦ *hop\_period*: Number of LoRa symbols between hops
  - ♦ *freq\_hop[]*: List of frequencies to hop between (up to 40)
2. Enable SX1276 Compatibility: Call *lr20xx\_workarounds\_lora\_freq\_hop\_enable\_sx1276\_compatibility\_mode()*, this modifies internal register *lora\_modem\_main\_tx\_cfg1* at address 0xF30A24.
3. Handle Sleep Mode (if needed): Use *lr20xx\_workarounds\_lora\_freq\_hop\_sx1276\_compatibility\_mode\_store\_retention\_mem()* to preserve the compatibility mode setting during sleep.
4. Disable When Not Needed: Call *lr20xx\_workarounds\_lora\_freq\_hop\_disable\_sx1276\_compatibility\_mode()* to return to native operation.

## 9.9 LoRa Commands

### 9.9.1 SetLoraModulationParams

The *SetLoraModulationParams* command configures the LoRa modulation parameters. If the packet type is not LoRa, the status of the next command returns CMD\_FAIL, indicating that the command execution is not compatible with the current packet type.

**Table 9-1: SetLoraModulationParams Command**

| Byte           | 0          | 1         | 2                  | 3                                |
|----------------|------------|-----------|--------------------|----------------------------------|
| Data from host | 0x02       | 0x20      | sf(3:0)<br>bw(3:0) | cr(3:0)<br>rfu(1:0)<br>ldro(1:0) |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24)   | IrqStatus(23:16)                 |

*sf* defines the spreading factor to be used.

**Table 9-2: LoRa Modulation sf Parameter Definition**

| sf  | Description      | sf  | Description |
|-----|------------------|-----|-------------|
| 0x5 | SF5              | 0x9 | SF9         |
| 0x6 | SF6 <sup>1</sup> | 0xA | SF10        |
| 0x7 | SF7              | 0xB | SF11        |
| 0x8 | SF8              | 0xC | SF12        |

1. For SF6 compatibility with SX127x products, refer to [Section 9.9.2](#)

*bw* defines the LoRa signal bandwidth (DSB) used in Tx and Rx.

**Table 9-3: LoRa Modulation bw Parameter Definition**

| bw  | Description                   | bw  | Description                       |
|-----|-------------------------------|-----|-----------------------------------|
| 0x0 | RFU                           | 0x8 | RFU                               |
| 0x1 | RFU                           | 0x9 | RFU                               |
| 0x2 | LORA_BW_31 (31.25 kHz actual) | 0xA | LORA_BW_41 (41.67 kHz actual)     |
| 0x3 | LORA_BW_62 (62.50 kHz actual) | 0xB | LORA_BW_83 (83.34 kHz actual)     |
| 0x4 | LORA_BW_125                   | 0xC | LORA_BW_101 (101.5625 kHz actual) |
| 0x5 | LORA_BW_250                   | 0xD | LORA_BW_203 (203.125 kHz actual)  |
| 0x6 | LORA_BW_500                   | 0xE | LORA_BW_406 (406.25 kHz actual)   |
| 0x7 | LORA_BW_1000                  | 0xF | LORA_BW_812 (812.5 kHz actual)    |

**cr** defines the FEC scheme used. Each CR brings an overhead on the number of actual bits transported, expressing the redundancy of the forward error correction code, as tabulated below:

**Table 9-4: LoRa Modulation cr Parameter Definition**

| cr   | Description                                                             |
|------|-------------------------------------------------------------------------|
| 0x00 | RFU                                                                     |
| 0x01 | Coding Rate 4/5 Overhead Ratio 1.25                                     |
| 0x02 | Coding Rate 4/6 Overhead Ratio 1.5                                      |
| 0x03 | Coding Rate 4/7 Overhead Ratio 1.75                                     |
| 0x04 | Coding Rate 4/8 Overhead Ratio 2                                        |
| 0x05 | Long Interleaver Coding Rate 4/5 Overhead Ratio 1.25                    |
| 0x06 | Long Interleaver Coding Rate 4/6 Overhead Ratio 1.5                     |
| 0x07 | Long Interleaver Coding Rate 4/8 Overhead Ratio 2                       |
| 0x08 | Long Interleaver Convolutional code 4/6 Overhead Ratio 1.5 <sup>1</sup> |
| 0x09 | Long Interleaver Convolutional code 4/8 Overhead Ratio 2 <sup>1</sup>   |

1. Incompatible with SX127x, 8x, 6x and LR11xx devices.

**ldro** defines the symbols used for modulation:

- ♦ 0: LDRO off: Use the full range of modulation symbols
- ♦ 1: LDRO on: Use 1/4 of the available modulation symbols
- ♦ 2–3: RFU

Recommended settings for **ldro** are:

- ♦ 1 for SF11 if BW ≤ 125kHz, and for SF12 if BW ≤ 250kHz
- ♦ 0 for all other SF and BW combinations

## 9.9.2 SF6 Compatibility with SX127x

The SX127x supports SF6 only in implicit mode (fixed packet length, no header).

The LR20xx chips handle SF6 differently by default, making them incompatible with SX127x devices at SF6.

Find detailed information about enabling the legacy mode in the LR20xx driver in *lr20xx\_workarounds.c*.

### 9.9.3 SetLoraPacketParams

The *SetLoraPacketParams* command configures the LoRa packet parameters. If the packet type is not LoRa, the status of the next command returns CMD\_FAIL, indicating that the command execution is not compatible with the current packet type.

**Table 9-5: SetLoRaPacketParams Command**

| Byte           | 0          | 1         | 2                | 3                | 4                | 5                                           |
|----------------|------------|-----------|------------------|------------------|------------------|---------------------------------------------|
| Data from host | 0x02       | 0x21      | pbl_len(15:8)    | pbl_len(7:0)     | payload_len(7:0) | rfu(4:0)<br>header_type<br>Crc<br>invert_iq |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) | IrqStatus(23:16) | IrqStatus(15:8)  | IrqStatus(7:0)                              |

**pbl\_len** defines the number of preamble symbols to be sent before the payload.

**payload\_len** specifies the maximum length of expected payload, allowing to filter packets not expected by this device. For explicit header mode:

- ♦ 0: Device accepts any payload length
- ♦ xx: Device only accepts payload lengths from 1 to “xx”. Payloads with a length of 0 or greater than “xx” are rejected, resulting in a header error interrupt

**header\_type** determines the header mode:

- ♦ 0: Explicit mode, where the packet length is unknown (variable packet length)
- ♦ 1: Implicit mode, where the packet length is known and pre-defined

**Crc** controls the CRC (Cyclic Redundancy Check) setting:

- ♦ 0: CRC off, disabling the CRC for the transmitted packets
- ♦ 1: CRC on, enabling the CRC for enhanced error detection

**invert\_iq** indicates the polarity or sense of the symbols transmitted during packet transmission:

- ♦ 0: Standard IQ setup, representing the default symbol polarity
- ♦ 1: Inverted IQ setup, inverting the symbol polarity

**Note:** For SX1280-compatible bandwidths (812, 406, 203 kHz), the *invert\_iq* parameter behaves differently, leading to inverted symbols. Proper consideration of these parameters is crucial to ensure correct LoRa packet transmission and reception, and to achieve optimal performance in various communication scenarios.

In summary, the chip automatically selects the symboling polarity for de-facto compatibility with previous generation chips.

## 9.9.4 SetLoRaSynchTimeout

The *SetLoRaSynchTimeout* command configures the LoRa modem to search for and detect for N symbols. By setting the value of **nb\_symbols**, developers define the duration for which the modem performs the search process to synchronize with incoming LoRa signals.

**Table 9-6: SetLoraSynchTimeout Command**

| Byte           | 0          | 1         | 2                | 3                |
|----------------|------------|-----------|------------------|------------------|
| Data from host | 0x02       | 0x22      | nb_symbols (7:0) | Format (7:0)     |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) | IrqStatus(23:16) |

**nb\_symbols** can be specified either as a number or in mantissa/exponent format, depending on the chosen format:

- ♦ If **Format** = 0, **nb\_symbols** represents the number of LoRa symbols
- ♦ If **Format** = 1, **nb\_symbols** defines the number of LoRa symbols given as mantissa/exponent  
(number of symbols =  $\text{nb\_symbols}(7:3) * 2^{\text{nb\_symbols}(2:0)+1}$ )

When **nb\_symbols** = 0x00, no symbol based timeout is configured, and the default value (defined in *SetRx*) is used by the modem.

Upon executing the command, the associated LoRa modem IRQs for RX\_SYNCH\_TIMEOUT, RX\_FalseSynch, and LoRaHeaderErr are enabled.

If any of these IRQs are asserted by the modem, and the **nb\_symbols** value is not set to 0x00, an RxTimeout sequence is initiated, causing the modem to exit Rx mode.

## 9.9.5 SetLoraSyncword

The *SetLoraSyncword* command configures the LoRa Syncword.

**Table 9-7: SetLoraSyncword Command**

| Byte           | 0          | 1         | 2                |
|----------------|------------|-----------|------------------|
| Data from host | 0x02       | 0x23      | Syncword(7:0)    |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) |

**Syncword** customizes a synchronisation pattern (Syncword) used for synchronization during data transmission and reception. Acceptable Syncwords are:

- ♦ Public Network: 0x34
- ♦ Private Network: 0x12 (default)

Other values are not valid, but other Syncword formats are supported, reach out to your Semtech contact for details.



## 9.9.6 SetLoraSideDetConfig

The *SetLoraSideDetConfig* command configures the LoRa Rx multi-SF detection. The multi-SF (or side-detection) can listen to multiple SF in parallel and automatically switch to Rx mode to demodulate the correct SF.

**Note:** The SF configured using the command *SetLoraModulationParams* is called the **main SF**, and the additional SFs are called **side SFs**. These side SF are configured using this command *SetLoraSideDetConfig*.

The LoRa modem supports multi-SF detection but not multi-SF demodulation. All detectors work with the same bandwidth (BW) as the main detector but with different spreading factors (*sf*), *ldro*, or *invert\_iq* settings.

When a LoRa signal is detected with any of the targeted SF, that detector occupies the demodulator for the rest of the reception process.

In addition to the main detector (configured with *SetLoraModulationParams*), up to three side detectors can be configured. Each side detector is enabled using the *config\_x* argument. To disable all side detectors, this command should be sent without any argument.

**Table 9-8: SetLoraSideDetConfig Command**

| Byte           | 0          | 1         | 2                | 3                | 4               |
|----------------|------------|-----------|------------------|------------------|-----------------|
| Data from host | 0x02       | 0x24      | config_1(7:0)    | config_2(7:0)    | config_3(7:0)   |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) | IrqStatus(23:16) | IrqStatus(15:8) |

*config\_x*, defines the settings for the xth side detector. The format of *config\_x* (x = 1, 2, or 3) is as follows:

- Bits 7:4: *sfx* (spreading factor) to be used for detector x according to the rules below
- Bits 3:2: *ldrox* used for detector x
- Bit 1: RFU
- Bit 0: *invert\_iqx* for detector x

The following rules apply (also apply to Multi-CAD):

- Up to 4 Spreading Factors (SF) can be detected in parallel (main SF + 3 side detectors).
- The main SF (configured with *SetLoraModulationParams*) must be the smallest SF in normal Rx operation and the largest SF for CAD operation.
- The header type is fixed by the main SF.
- If the BW is greater than 500 kHz, only a maximum of two side detectors are allowed.
- If the main SF is 10, 11, or 12, only one side detector is allowed (maximum two SFs can be detected simultaneously).
- All SFs must be different; it is not possible to have two of the same SF in parallel.
- The difference between the Max SF and Min SF must be less than or equal to 4.
- After a successful reception, *GetLoraPacketStatus* (see detector(2:0)) indicates which SF was demodulated.
- Calling *SetLoraModulationParams* disables all side detectors, and the developer must re-configure them.

## 9.9.7 SetLoraSideDetSyncword

The *SetLoraSideDetSyncword* command configures the LoRa Rx multi-SF (side detectors) Syncwords.

This command sets the synchronization words for the side detectors in LoRa reception.

**Table 9-9: SetLoraSideDetSyncword Command**

| Byte           | 0          | 1         | 2                | 3                | 4               |
|----------------|------------|-----------|------------------|------------------|-----------------|
| Data from host | 0x02       | 0x25      | Syncword1(7:0)   | Syncword2(7:0)   | Syncword3(7:0)  |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) | IrqStatus(23:16) | IrqStatus(15:8) |

### 9.9.8 GetLoraRxStats

The *GetLoraRxStats* command provides access to the internal statistics of received packets in LoRa mode.

**Table 9-10: GetLoraRxStats Command**

| Byte           | 0          | 1          |
|----------------|------------|------------|
| Data from host | 0x02       | 0x29       |
| Data to host   | Stat(15:8) | Stat(15:8) |

**Table 9-11: GetLoraRxStats Response**

| Byte           | 0              | 1             | 2                | 3               | 4                           | 5                          | 6                              | 7                             | 8                     | 9                    |
|----------------|----------------|---------------|------------------|-----------------|-----------------------------|----------------------------|--------------------------------|-------------------------------|-----------------------|----------------------|
| Data from host | 0x00           | 0x00          | 0x00             | 0x00            | 0x00                        | 0x00                       | 0x00                           | 0x00                          | 0x00                  | 0x00                 |
| Data to host   | Stat<br>(15:8) | Stat<br>(7:0) | pkt_rx<br>(15:8) | pkt_rx<br>(7:0) | pkt_crc_<br>error<br>(15:8) | pkt_crc_<br>error<br>(7:0) | header_<br>crc_error<br>(15:8) | header_<br>crc_error<br>(7:0) | false_synch<br>(15:8) | false_synch<br>(7:0) |

Returned values include:

- ♦ ***pkt\_rx***: Total number of received packets
- ♦ ***pkt\_crc\_error***: Number of received packets with a CRC error
- ♦ ***header\_crc\_error***: Number of received packets with a header error
- ♦ ***false\_synch***: Number of false syncs (false synchronizations)

These statistics are reset under specific conditions, such as a Power-on Reset (POR), sleep without memory retention, or the execution of the *ResetRxStats* command.

### 9.9.9 GetLoraPacketStatus

The *GetLoraPacketStatus* command retrieves the status of the last received packet in LoRa mode. This status is updated at the end of a reception (RxDone or CadDone irq).

**Table 9-12: GetLoraPacketStatus Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x02       | 0x2A      |
| Data to host   | Stat(15:8) | Stat(7:0) |

**Table 9-13: GetLoraPacketStatus Response**

| Byte           | 0              | 1             | 2                                   | 3                   | 4                | 5                  | 6                         | 7                                                                 |
|----------------|----------------|---------------|-------------------------------------|---------------------|------------------|--------------------|---------------------------|-------------------------------------------------------------------|
| Data from host | 0x00           | 0x00          | 0x00                                | 0x00                | 0x00             | 0x00               | 0x00                      | 0x00                                                              |
| Data to host   | Stat<br>(15:8) | Stat<br>(7:0) | rfu(2:0)<br>crc<br>coding_rate(3:0) | pkt_length<br>(7:0) | snr_pkt<br>(7:0) | rss_i_pkt<br>(8:1) | rss_i_signal_pkt<br>(8:1) | rfu (1:0)<br>detector(3:0)<br>rss_i_pkt(0)<br>rss_i_signal_pkt(0) |

| Byte           | 8                      | 9                     | 10                   |
|----------------|------------------------|-----------------------|----------------------|
| Data from host | 0x00                   | 0x00                  | 0x00                 |
| Data to host   | freq_offset<br>(23:16) | freq_offset<br>(15:8) | freq_offset<br>(7:0) |

The *GetLoraPacketStatus* command provides the following returned values:

- ♦ **crc:**
  - ♦ 1: Crc\_On
  - ♦ 0: Crc\_OFF

The value is determined either by the information coded in the last received packet header (explicit header mode) or the configured settings for **Cr** and **Crc\_type** (implicit header mode).
- ♦ **coding\_rate:** The coding rate used in the packet. Refer to the [SetLoraModulationParams](#) command for specific coding rate values.
- ♦ **pkt\_length:** Length of the last received payload in bytes.
- ♦ **snr\_pkt:** Provides an estimation of the Signal-to-Noise Ratio (SNR) on the last received packet. The value is in two's complement format and multiplied by 4. The actual SNR in dB can be calculated as  $\text{snr\_pkt}/4$ .
- ♦ **rss\_i\_pkt:** Average Received Signal Strength Indication (RSSI) over the last received packet. The actual signal power is calculated as  $-\text{rss\_i\_pkt}/2$  [dBm].
- ♦ **rss\_i\_signal\_pkt:** Estimation of the RSSI of the LoRa signal (after despreading) on the last received packet. The actual value is  $-\text{rss\_i\_signal\_pkt}/2$  [dBm].
- ♦ **detector:** Flags which detector(s) received the packet. In normal Rx, only one flag is set: the one used to receive the packet. In CAD, all detector paths triggered are set.
  - ♦ 0001: Main
  - ♦ 0010: Side1
  - ♦ 0100: Side2
  - ♦ 1000: Side3
- ♦ **freq\_offset:** Frequency error in Hz as a signed 24-bit integer.

## 9.9.10 SetLoraAddress

The *SetLoraAddress* command defines the LoRa Rx address filtering.

**Table 9-14: SetLoraAddress Command**

| Byte           | 0          | 1         | 2                                        | 3                | ... | 2+ addr_comp_len |
|----------------|------------|-----------|------------------------------------------|------------------|-----|------------------|
| Data from host | 0x02       | 0x2B      | addr_comp_len(3:0)<br>addr_comp_pos(3:0) | addr             | ... | addr             |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24)                         | IrqStatus(23:16) | ... | 0x00             |

**addr\_comp\_len** defines the number of bytes (0 to 8) used in addr filtering. 0 = no addr filtering.

**addr\_comp\_pos** defines the address position in bytes in the payload (header not counted).

**addr** defines the address to filter (keep). It can be 1 to 8 bytes depending on **addr\_comp\_len** configured.

This feature can be used to separate traffic in systems where different network IDs co-exist, the receiver aborts prematurely if the address doesn't match, conserving energy in the device.

## 9.9.11 SetLoraHopping

The *SetLoraHopping* command configures the LoRa intra packet hopping feature.

**Table 9-15: SetLoraHopping Command**

| Byte           | 0              | 1             | 2                                            | 3                    | 4                    | 5                    | 6                   | 7                  | 8                    | ... |
|----------------|----------------|---------------|----------------------------------------------|----------------------|----------------------|----------------------|---------------------|--------------------|----------------------|-----|
| Data from host | 0x02           | 0x2C          | hop_ctrl(1:0)<br>rfu<br>hop_period<br>(12:8) | hop_period<br>(7:0)  | freq_hop1<br>(31:24) | freq_hop1<br>(23:16) | freq_hop1<br>(15:8) | freq_hop1<br>(7:0) | freq_hop2<br>(31:24) | ... |
| Data to host   | Stat<br>(15:8) | Stat<br>(7:0) | IrqStatus<br>(31:24)                         | IrqStatus<br>(23:16) | IrqStatus<br>(15:8)  | IrqStatus<br>(7:0)   | 0x00                | 0x00               | 0x00                 | ... |

**hop\_ctrl** enables the hopping feature:

- 0: Hopping disabled and other parameters are ignored
- 1: Intra packet hopping enabled

**hop\_period** defines the number of LoRa symbols for one hop. (The number of LoRa symbols to wait before changing the RF frequency).

**freq\_hop<sub>x</sub>** (with **x** from 1 to max 40) defines the hopping frequencies value. A maximum of 40 hopping frequencies are accepted.

After the header is received or transmitted, the transceiver waits for **hop\_period** symbols before switching to the next frequency in the list. If no more frequencies are available, the hopping sequence stops but the transmission / reception is not interrupted.

### 9.9.11.1 SX1276 Compatibility

To configure for SX1276 compatible frequency hopping, write into the **lora\_modem\_main\_tx\_cfg1** register at address 0xF30A24.

- Pseudo code to enable/disable the legacy hopping mode:  
*WriteRegMemMask32(addr = 0xF30A24, Mask = (1 << 18), Value = (enable << 18))*
- To correctly save the legacy mode in Sleep modes, register **lora\_modem\_main\_tx\_cfg1** must be added to the additional register:  
*SetAdditionalRegToRetain(slot = 1, addr = 0xF30A24)*

## 9.9.12 SetLoraTxSync

The *SetLoraTxSync* command configures the LoRa Tx synchronization using a DIO signal for RTToF mode.

**Table 9-16: SetLoraTxSync Command**

| Byte           | 0          | 1         | 2                            |
|----------------|------------|-----------|------------------------------|
| Data from host | 0x02       | 0x1D      | Function(1:0)<br>DioNum(5:0) |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus (31:24)            |

**Function** defines Tx sync function:

- ◆ 0x0: Disabled
- ◆ 0x1: Tx Sync Manager. In this mode, the radio waits for an external signal to transmit the sync frame
- ◆ 0x2: Tx Sync Subordinate. In this mode, the radio listens for a sync frame and outputs a sync signal

**DioNum** defines which DIO to use to start the Tx sync or output the sync signal.

## 9.9.13 SetLoraSideDetCad

The *SetLoraSideDetCad* command configures the LoRa channel activity detection (CAD) mode when the receiver is expected to detect frames of varying Spreading Factors.

**Table 9-17: SetLoraSideDetCad Command**

| Byte           | 0              | 1             | 2                               | 3                    | 4                               | 5                  | 6                               | 7                  |
|----------------|----------------|---------------|---------------------------------|----------------------|---------------------------------|--------------------|---------------------------------|--------------------|
| Data from host | 0x02           | 0x1E          | rfu(2:0)<br>pnr_delta1<br>(3:0) | det_peak1<br>(7:0)   | rfu(2:0)<br>pnr_delta2<br>(3:0) | det_peak2<br>(7:0) | rfu(2:0)<br>pnr_delta3<br>(3:0) | det_peak3<br>(7:0) |
| Data to host   | Stat<br>(15:8) | Stat<br>(7:0) | IrqStatus<br>(31:24)            | IrqStatus<br>(23:16) | IrqStatus<br>(15:8)             | IrqStatus<br>(7:0) | 0x00                            | 0x00               |

**pnr\_deltaX**: pnr\_delta for side detector X used during CAD. The same definition as for the main detector in the [GetLoraRxStats](#).

**det\_peakX**: det\_peak for side detector X used during CAD. The same definition as for the main detector in the [GetLoraRxStats](#).

## 10. RTToF (Round-Trip Time of Flight) (LR20xx)

### 10.1 RTToF Engine

RTToF is a functionality that uses LoRa modulation to measure device to device distance and uses the commands listed here below. (It is also referred to as ranging).

A lot of application related information regarding RTToF can be found in application notes AN1200.114, AN1200.89 and AN1200.29 on the Semtech web site.

### 10.2 RTToF Commands

#### 10.2.1 SetRangingAddr

The *SetRangingAddr* command sets the ranging ID for the device, which is specifically used in Subordinate devices for ranging operations.

**Table 10-1: SetRangingAddr Command**

| Byte           | 0          | 1         | 2                | 3                | 4               | 5              | 6                 |
|----------------|------------|-----------|------------------|------------------|-----------------|----------------|-------------------|
| Data from host | 0x02       | 0x78      | Addr(31:24)      | Addr(23:16)      | Addr(15:8)      | Addr(7:0)      | CheckLength (7:0) |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) | IrqStatus(23:16) | IrqStatus(15:8) | IrqStatus(7:0) | 0x00              |

**Addr** defines the ranging device's address, using a 32-bit value. Default value is 0x00000019.

**CheckLength** defines how many bytes of the address are to be compared against the request address sent by the Manager.

- ◆ 4: (Default), all four bytes of the address are checked
- ◆ Less than 4: The least significant bytes of the address are checked against the request address

#### 10.2.2 SetRangingReqAddr

The *SetRangingReqAddr* command configures the ranging ID for the requests in the Manager device during the ranging process.

**Table 10-2: SetRangingReqAddr Command**

| Byte           | 0          | 1         | 2                | 3                | 4               | 5              |
|----------------|------------|-----------|------------------|------------------|-----------------|----------------|
| Data from host | 0x02       | 0x79      | ReqAddr (31:24)  | ReqAddr (23:16)  | ReqAddr (15:8)  | ReqAddr (7:0)  |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) | IrqStatus(23:16) | IrqStatus(15:8) | IrqStatus(7:0) |

**ReqAddr** represents a 32-bit ranging ID used by the Manager. Default value is 0x00000019.

### 10.2.3 GetRangingResult

The *GetRangingResult* command gives the ranging result obtained during the ranging process (for Manager only).

The ranging result contains valuable information related to the distance measurement between the Manager and Subordinate devices involved in the ranging procedure.

**Table 10-3: GetRangingResult Command**

| Byte           | 0          | 1         | 2                |
|----------------|------------|-----------|------------------|
| Data from host | 0x02       | 0x7A      | Type(7:0)        |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) |

**Table 10-4: GetRangingResult Response**

| Byte           | 0          | 1         | 2               | 3              | 4             | 5     | 6               | 7              | 8             |
|----------------|------------|-----------|-----------------|----------------|---------------|-------|-----------------|----------------|---------------|
| Data from host | 0x00       | 0x00      | 0x00            | 0x00           | 0x00          | 0x00  | 0x00            | 0x00           | 0x00          |
| Data to host   | Stat(15:8) | Stat(7:0) | Rng1<br>(23:16) | Rng1<br>(15:8) | Rng1<br>(7:0) | Rssi1 | Rng2<br>(23:16) | Rng2<br>(15:8) | Rng2<br>(7:0) |

**Type** defines the type of value returned:

- 0: Latest raw results. This option returns the most recent raw ranging result (**Rng1**) and received signal strength indicator (**Rssi1**) in dB.
- 1: Returns the latest raw ranging results (**Rng1** and **Rssi1**) as well as additional ranging data (**Rng2**) required for the extended ranging mode.  
**Rng1** and **Rng2** values should be averaged to get a distance estimation unaffected by Doppler effect. The difference between the two values provides an estimation of the Doppler effect, i.e. the relative speed between Manager and Subordinate.
- 2: 2 bytes are returned by the chip i.e., **GainStep1**, **GainStep2** results from AGC operation obtained during ranging exchanges for single exchange or extended exchange respectively.  
The AGC gain step(s) on reception (in the Manager) of the Subordinate response(s) are latched and can be read out. This may be useful for compensation a potential delay depending on the AGC gain.

Developers can choose the appropriate **Type** parameter based on their ranging requirements and obtain the corresponding ranging results.

$$\text{The Distance [m]} = \text{Rng\_i} * 150 / (2^{12} * \text{BWLoRa}), \text{ with BWLoRa in MHz.}$$

## 10.2.4 GetRangingStats

The *GetRangingStats* command provides information about the number of ranging exchanges collected by the Manager, the ones that are valid or discarded, etc.

**Table 10-5: GetRangingStats Request Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x02       | 0x7D      |
| Data to host   | Stat(15:8) | Stat(7:0) |

**Table 10-6: GetRangingStats Response**

| Byte           | 0          | 1         | 2                       | 3                      | 4                      | 5                     |
|----------------|------------|-----------|-------------------------|------------------------|------------------------|-----------------------|
| Data from host | 0x00       | 0x00      | 0x00                    | 0x00                   | 0x00                   | 0x00                  |
| Data to host   | Stat(15:8) | Stat(7:0) | ExchangeValid<br>(15:8) | ExchangeValid<br>(7:0) | RequestValid<br>(15:8) | RequestValid<br>(7:0) |

| Byte           | 6                      | 7                     | 8                 | 9                | 10                         | 11                        |
|----------------|------------------------|-----------------------|-------------------|------------------|----------------------------|---------------------------|
| Data from host | 0x00                   | 0x00                  | 0x00              | 0x00             | 0x00                       | 0x00                      |
| Data to host   | ResponseDone<br>(15:8) | ResponseDone<br>(7:0) | Timeout<br>(15:8) | Timeout<br>(7:0) | RequestDiscarded<br>(15:8) | RequestDiscarded<br>(7:0) |

The *GetRangingStats* command provides the following values:

- ♦ **ExchangeValid** indicates the number of valid ranging exchanges.
- ♦ **RequestValid** indicates the number of valid ranging requests done.
- ♦ **ResponseDone** indicates the number of ranging responses done.
- ♦ **Timeout** indicates the number of timeout events that have occurred. The timeout value is incremented:
  - ♦ For Manager role if no response received from a Subordinate
  - ♦ For Subordinate role if no extended requested received (extended mode only)
- ♦ **RequestDiscarded** indicates the number of discarded ranging requests.

Statistics are reset on a POR, sleep without memory retention, or by the command [ResetRxStats](#).

**Note:** For extended ranging mode, the counters are incremented twice, once for each request/response.



## 10.2.5 SetRangingTxRxDelay

The *SetRangingTxRxDelay* command configures the Tx->Rx delay used for calibrating the ranging value.

**Table 10-7: SetRangingTxRxDelay Command**

| Byte           | 0          | 1         | 2                | 3                | 4               | 8              |
|----------------|------------|-----------|------------------|------------------|-----------------|----------------|
| Data from host | 0x02       | 0x7B      | Delay (31:24)    | Delay (23:16)    | Delay (15:8)    | Delay (7:0)    |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) | IrqStatus(23:16) | IrqStatus(15:8) | IrqStatus(7:0) |

**Delay** defines the Tx-> Rx delay expressed in  $1/(BWL*1024)$  where BWL is the LoRa bandwidth in MHz.

## 10.2.6 SetRangingParams

The *SetRangingParams* command configures specific parameters related to the RTTOF functionality.

**Table 10-8: SetRangingParams Command**

| Byte           | 0          | 1         | 2                                          |
|----------------|------------|-----------|--------------------------------------------|
| Data from host | 0x02       | 0x7C      | extended_op<br>spy_mode<br>nb_symbols(5:0) |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24)                           |

**extended\_op** enables or disables the extended mode:

- ♦ 0: Normal RTTOF operation
- ♦ 1: Extended RTTOF operation

**spy\_mode** enables or disables the advanced ranging:

- ♦ 0: Spy mode disabled
- ♦ 1: Spy mode enabled on the next *SetRx* (instead of Subordinate mode)

In spy mode, the chip acts like a normal device in a Subordinate modem and sends a response to any request for its address. It also tells the device to listen to request and response pairs that are not addressed to it and measure the ranging distance from them. See application notes AN1200.114 for details.

**nb\_symbols** sets the number of symbols sent in the responses.

## 11. (G)FSK Modulation (LR20xx)

### 11.1 (G)FSK Modulation Principle

The (G)FSK modem integrated into the LR2021/LR2022/LR2012 offers an efficient and adaptable CPFSK (Continuous Phase Frequency Shift Keying) modulator/demodulator. This modem provides support for FSK (Frequency Shift Keying), GFSK (Gaussian Frequency Shift Keying), MSK (Minimum Shift Keying), and GMSK (Gaussian Minimum Shift Keying) modulation schemes.

One of the key advantages of the (G)FSK modem is its flexibility, as it allows optional Gaussian filtering, enabling developers to fine-tune and optimize their modulation characteristics according to their specific requirements.

Additionally, the (G)FSK modem is engineered to ensure physical layer compatibility with various wireless communication standards. It is fully compatible with Bluetooth Low Energy, Wi-SUN FSK, WM-Bus, as well as Semtech's previous chip families, including SX127xx, SX126xx, and LR11xx. This compatibility ensures direct integration and easy migration for developers who have previously used these chip-sets in their projects. See Application Note AN1200.103 for details of FSK usage.

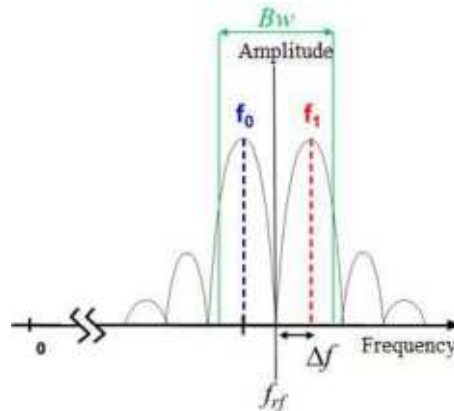
#### 11.1.1 Modem Bandwidth and Bit Rates

The FSK modem integrated into the LR2021/LR2022/LR2012 offers 2-FSK modulation capabilities, supporting a wide range of bit rates, see [BRFSK1](#). The specific bit rate is configured using the [SetFskModulationParams](#) command, allowing developers to tailor the modulation scheme to their desired communication requirements. The settings for FSK are tabulated in the electrical specifications [Table 3-8](#), and its usage is detailed in the Application Note AN1200.103.

The FSK modulation is based on the principle of shifting the carrier frequency between two distinct values, representing binary data states. This straightforward technique enables reliable data transmission over different distances and through varying channel conditions.

The FSK modem provides a double side band (DSB) occupied bandwidth, as illustrated in the image below, along with other essential modulation parameters.

**Figure 11-1: FSK Modulation Parameters**



Where  $\Delta f$  is the frequency deviation, and  $f_{RF}$  is the RF center frequency.

In receive mode, the bandwidth of the receiver is configured to the lowest value that can effectively accommodate the signal bandwidth of the FSK (Frequency Shift Keying) signal. The receiver bandwidth is typically determined based on the following formula:

$$B_{-20dB} = 2\Delta f + R_b + \text{freqoffset}$$

Where  $R_b$  is the bit rate and "freqoffset" represents the difference in RF carrier frequency between the transmitter and the receiver, primarily caused by XTAL (crystal) impairments. This difference can lead to deviations in the received signal, affecting the overall performance of the communication.

### 11.1.2 (G)FSK Modulation Index

In addition to the raw bit rate and bandwidth, the modulation index can be over the range 0.35 to 5. The modulation index,  $\beta$ , is a figure of merit that describes the proximity of '1' and '0' frequencies for a given bit rate. This influences the ease with which each logical level can be discriminated by the demodulator and is given by:

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

where  $\Delta f$  is the frequency deviation and  $R_b$  is the programmed bit rate.

### 11.1.3 (G)FSK Limitations

The demodulator has important limitations:

- Maximum bitrate is 2 Mbps
- Maximum BW to use is:  $BW\_max \leq \text{bitrate} * \text{osr}$ , where osr (oversampling rate) is:

**Table 11-1: Oversampling Rate Limitations**

| Modulation Index    | osr |
|---------------------|-----|
| $h < 0.85$          | 4   |
| $0.85 \leq h < 1.6$ | 8   |
| $1.6 \leq h < 2.6$  | 8   |
| $h \geq 2.6$        | 8   |

- In GFSK (filtered FSK) the maximum modulation index is  $h=6.2$
- In FSK the maximum modulation index is  $h=8.33$
- The  $\text{bitrate} * \text{osr}$  product must not exceed 8 MHz
- The BW must be large enough to contain the signal and any required frequency error:  
 $BW\_min \geq 2 * F_{dev} + \text{bitrate} + \text{Freq\_err\_max}$  (Carlson approximation)
- If 0 bits of detection are set, the Syncword is used for the detection. This configuration is only recommended for  $\text{osr}=4$  and can improve situations where the AGC has to react quickly.
- At least 20b of detection (or 20b of Syncword, for the previous case) are recommended at  $\text{osr}=4$ . Higher osr require fewer bits.
- The minimum modulation index is  $h=0.3$  for both FSK and GFSK (tested at 500 kbps). 0 bits of detection are recommended in this case.

**Table 11-2: (G)FSK Limitations Summary**

| Name                   | Value                                               |
|------------------------|-----------------------------------------------------|
| bitrate max            | 2 Mbps                                              |
| bitrate min            | 500 bps                                             |
| $h\_min$               | 0.3                                                 |
| $h\_max$ FSK           | 8.33                                                |
| $h\_max$ GFSK (bt=0.5) | 6.2                                                 |
| BW max                 | 3076 kHz or $\text{bitrate} * \text{osr}$           |
| BW min                 | 3.5 kHz or $\text{signal\_bw} + \text{Freq\_error}$ |
| Sample rate max        | $\text{bitrate} * \text{osr} \leq 8 \text{ MHz}$    |

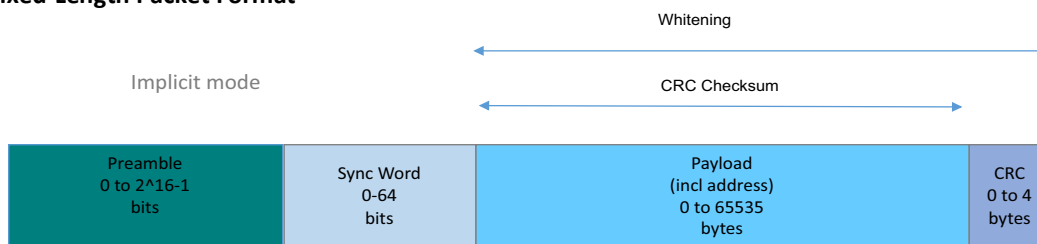
## 11.2 (G)FSK Packet Handler

The FSK packet format offers a well-established structure suitable for implementation in proprietary communication links that use NRZ (Non-Return-to-Zero) coding and are designed for low-energy applications. This packet format incorporates essential features such as CRC (Cyclic Redundancy Check) for payload validation, dynamic payload size support, and packet acknowledgement. Additionally, a developer can enable whitening based on pseudo-random number generation. Within the FSK packet handler, two main packet formats are available: fixed length and variable length packets.

### 11.2.1 Fixed-Length Packet

In scenarios where the packet length is fixed and known by both ends of the communication link, there is no need to transmit this information over the air. Instead, the packet length can be directly specified in bytes, via a host command, without the need for additional overhead or transmission of packet length information during communication. In this fixed-length packet mode, the address information is considered part of the payload. Therefore, the developer must ensure that the specified packet length includes the address field and any other necessary data fields within the packet.

**Figure 11-2: Fixed-Length Packet Format**

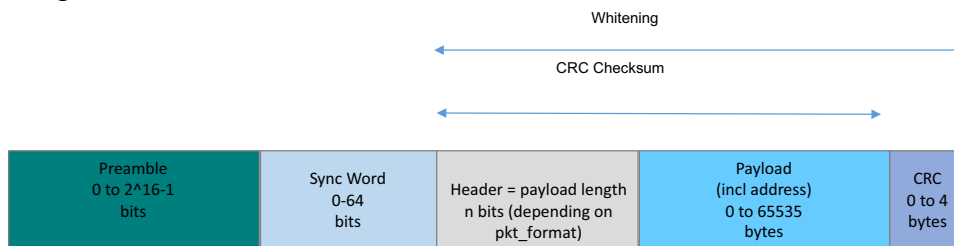


### 11.2.2 Variable-Length Packet

In scenarios where the packet size is uncertain or variable, it becomes necessary to include information about the packet length within the transmitted packet. This is particularly important when dealing with data packets that can vary in size and need to be accurately processed by the receiver. The variable-length packet format includes a field dedicated to specifying the length of the packet. This length field allows for values ranging from 0 to 65535. This format ensures that the receiver can determine the size of incoming packets, enabling proper data handling and processing.

**Note:** When address filtering is activated, the maximum packet length is limited to 254 bytes to accommodate the additional addressing information.

**Figure 11-3: Variable-Length Packet Format**



In the LR2021/LR2022/LR2012, the handling of packet length and address information is streamlined and automated by the hardware, making data transmission and reception more efficient and user-friendly.

When the **pkt\_format** field is set to indicate variable size in the **SetFskPacketParams** command (see [Table 11-8](#)), the packet length is automatically added to the transmitted packet. This ensures that the receiver can precisely determine the size of incoming variable-length packets, as mentioned earlier.

The LR2021/LR2022/LR2012 can include node or broadcast addresses in the transmitted packet using the **addr\_comp** field in the **SetFskPacketParams** command. By enabling this feature, developers can implement an additional level of packet filtering at the payload level. The LR2021/LR2022/LR2012 FSK hardware takes care of inserting the appropriate address information into the packet automatically.

## 11.3 (G)FSK Commands

### 11.3.1 SetFskModulationParams

The *SetFskModulationParams* command in the LR2021/LR2022/LR2012 configures the modulation parameters specifically for Frequency-Shift Keying (FSK) packets. FW configures respective modem registers.

When executed, the LR2021/LR2022/LR2012 takes care of configuring the relevant modem registers with the specified FSK modulation parameters. This command returns CMD\_FAIL in the status of the subsequent command if the current packet type is not set to FSK. Therefore, developers must ensure that the packet type is correctly set to FSK using the *SetPacketType* command before invoking *SetFskModulationParams*.

**Table 11-3: SetFskModulationParams Command**

| Byte           | 0           | 1          | 2                 | 3                 | 4                | 5               | 6           | 7     | 8               | 9              | 10            |
|----------------|-------------|------------|-------------------|-------------------|------------------|-----------------|-------------|-------|-----------------|----------------|---------------|
| Data from host | 0x02        | 0x40       | bitrate (31:24)   | bitrate (23:16)   | bitrate (15:8)   | bitrate (7:0)   | pulse_shape | rx_bw | fdev_hz (23:16) | fdev_hz (15:8) | fdev_hz (7:0) |
| Data to host   | Stat (15:8) | Stat (7:0) | IrqStatus (31:24) | IrqStatus (23:16) | IrqStatus (15:8) | IrqStatus (7:0) | 0x00        | 0x00  | 0x00            | 0x00           | 0x00          |

**bitrate:** defines the symbol rate:

- When **BitRate(31)** = 0, the BitRate is taken directly as **BitRate(30:0)** and expressed in bits per second.
- When **BitRate(31)** = 1, the BitRate is calculated as **BitRate(30:0)/256**, representing a fractional part of the bits per second.

**pulse\_shape** defines pulse shaping filter applied to the FSK symbol.

- 0x0: No pulse shaping
- 0x1: RFU
- 0x2: Gaussian with Bandwidth-Time bit period (BT) = 2.0
- 0x3: RFU
- 0x4: Gaussian with BT 0.3
- 0x5: Gaussian with BT 0.5
- 0x6: Gaussian with BT 0.7
- 0x7: Gaussian with BT 1.0
- All other values RFU

**rx\_bw** defines the double-sided bandwidth (RxBW) of the channel filter seen by each demodulator. See Table 11-4, “Correlation of Bandwidth, rx\_bw and Intermediate Frequency,” on page 174.

**fdev\_hz** defines the frequency deviation of the FSK modulation in Hertz (Hz). It determines half the difference in frequency between the two FSK states.

Table 11-4: Correlation of Bandwidth, rx\_bw and Intermediate Frequency

| BW Name | BW (kHz) | rx_bw (d) | IF (kHz) | BW Name | BW (kHz) | rx_bw (d) | IF (kHz) | BW Name | BW (kHz) | rx_bw (d) | IF (kHz) |
|---------|----------|-----------|----------|---------|----------|-----------|----------|---------|----------|-----------|----------|
| BW_3076 | 3076.9   | 0         | 2666.7   | BW_185  | 185.2    | 210       | 444.4    | BW_29   | 29.8     | 85        | 444.4    |
| BW_2857 | 2857.1   | 64        | 2666.7   | BW_178  | 178.6    | 90        | 333.3    | BW_27   | 27.8     | 228       | 266.7    |
| BW_2666 | 2666.7   | 128       | 2666.7   | BW_166  | 166.7    | 154       | 333.3    | BW_24   | 24.0     | 29        | 333.3    |
| BW_2222 | 2222.2   | 192       | 2666.7   | BW_153  | 153.8    | 34        | 266.7    | BW_23   | 23.1     | 213       | 444.4    |
| BW_1333 | 1333.3   | 136       | 1333.3   | BW_142  | 142.9    | 98        | 266.7    | BW_22   | 22.3     | 93        | 333.3    |
| BW_1111 | 1111.1   | 200       | 1333.3   | BW_138  | 138.9    | 218       | 333.3    | BW_20   | 20.8     | 157       | 333.3    |
| BW_888  | 888.9    | 144       | 888.9    | BW_133  | 133.3    | 162       | 266.7    | BW_19   | 19.2     | 37        | 266.7    |
| BW_769  | 769.2    | 24        | 666.7    | BW_128  | 128.2    | 19        | 444.4    | BW_17   | 17.9     | 101       | 266.7    |
| BW_740  | 740.7    | 208       | 888.9    | BW_119  | 119.0    | 83        | 444.4    | BW_16   | 16.7     | 165       | 266.7    |
| BW_714  | 714.3    | 88        | 666.7    | BW_111  | 111.1    | 226       | 266.7    | BW_14   | 14.9     | 86        | 444.4    |
| BW_666  | 666.7    | 152       | 666.7    | BW_96   | 96.2     | 27        | 333.3    | BW_13   | 13.9     | 229       | 266.7    |
| BW_615  | 615.4    | 32        | 533.3    | BW_92   | 92.6     | 211       | 444.4    | BW_12   | 12.0     | 30        | 333.3    |
| BW_571  | 571.4    | 96        | 533.3    | BW_89   | 89.3     | 91        | 333.3    | BW_11   | 11.2     | 94        | 333.3    |
| BW_555  | 555.6    | 216       | 666.7    | BW_83   | 83.3     | 155       | 333.3    | BW_10   | 10.4     | 158       | 333.3    |
| BW_533  | 533.3    | 160       | 533.3    | BW_76   | 76.9     | 35        | 266.7    | BW_9_6  | 9.6      | 38        | 266.7    |
| BW_512  | 512.8    | 17        | 444.4    | BW_71   | 71.4     | 99        | 266.7    | BW_8_9  | 8.9      | 102       | 266.7    |
| BW_476  | 476.2    | 81        | 444.4    | BW_69   | 69.4     | 219       | 333.3    | BW_8_7  | 8.7      | 222       | 333.3    |
| BW_444  | 444.4    | 224       | 533.3    | BW_66   | 66.7     | 163       | 266.7    | BW_8_3  | 8.3      | 166       | 266.7    |
| BW_384  | 384.6    | 25        | 333.3    | BW_64   | 64.1     | 20        | 444.4    | BW_8    | 8.0      | 23        | 444.4    |
| BW_370  | 370.4    | 209       | 444.4    | BW_59   | 59.5     | 84        | 444.4    | BW_7_4  | 7.4      | 87        | 444.4    |
| BW_357  | 357.1    | 89        | 333.3    | BW_55   | 55.6     | 227       | 266.7    | BW_6_9  | 6.9      | 230       | 266.7    |
| BW_333  | 333.3    | 153       | 333.3    | BW_48   | 48.1     | 28        | 333.3    | BW_6    | 6.0      | 31        | 333.3    |
| BW_307  | 307.7    | 33        | 266.7    | BW_46   | 46.3     | 212       | 444.4    | BW_5_8  | 5.8      | 215       | 444.4    |
| BW_285  | 285.7    | 97        | 266.7    | BW_44   | 44.6     | 92        | 333.3    | BW_5_6  | 5.6      | 95        | 333.3    |
| BW_277  | 277.8    | 217       | 333.3    | BW_41   | 41.7     | 156       | 333.3    | BW_5_2  | 5.2      | 159       | 333.3    |
| BW_266  | 266.7    | 161       | 266.7    | BW_38   | 38.5     | 36        | 266.7    | BW_4_8  | 4.8      | 39        | 266.7    |
| BW_256  | 256.4    | 18        | 444.4    | BW_35   | 35.7     | 100       | 266.7    | BW_4_5  | 4.5      | 103       | 266.7    |
| BW_238  | 238.1    | 82        | 444.4    | BW_34   | 34.7     | 220       | 333.3    | BW_4_3  | 4.3      | 223       | 333.3    |
| BW_222  | 222.2    | 225       | 266.7    | BW_33   | 33.3     | 164       | 266.7    | BW_4_2  | 4.2      | 167       | 266.7    |
| BW_192  | 192.3    | 26        | 333.3    | BW_32   | 32.1     | 21        | 444.4    | BW_3_5  | 3.5      | 231       | 266.7    |

### 11.3.2 SetFskPacketParams

The *SetFskPacketParams* command in the LR2021/LR2022/LR2012 sets the packet parameters specifically for Legacy Frequency-Shift Keying (FSK) packets. This command fails if the packet type is not set to FSK.

**Table 11-5: SetFskPacketParams Command**

| Byte           | 0           | 1          | 2                 | 3                 | 4                | 5                                                                                  | 6              | 7             | 8                        |
|----------------|-------------|------------|-------------------|-------------------|------------------|------------------------------------------------------------------------------------|----------------|---------------|--------------------------|
| Data from host | 0x02        | 0x41       | pbl_len_tx (15:8) | pbl_len_tx (7:0)  | pbl_detect (7:0) | rfu(1:0)<br>long_preamble_mode<br>pld_lenUnit<br>addr_comp(1:0)<br>pkt_format(1:0) | pld_len (15:8) | pld_len (7:0) | Crc(3:0)<br>dc_free(3:0) |
| Data to host   | Stat (15:8) | Stat (7:0) | IrqStatus (31:24) | IrqStatus (23:16) | IrqStatus (15:8) | IrqStatus (7:0)                                                                    | 0x00           | 0x00          | 0x00                     |

**pbl\_len\_tx** specifies the number of preamble bits to be transmitted during packet transmission.

**pbl\_detect** determines how the preamble behaves during packet detection:

- 0x00: Preamble detection is off. Packet detection is based solely on the Syncword.
- Other than 0x00: the receiver looks for a certain number of preamble bits to be detected before initiating packet detection. Detection on preamble is useful if transmitting device sends a long preamble and the receiver is in duty cycle mode, for example.
- **pbl\_detect** also controls the triggering of the PreambleDetected IRQ. When preamble detection is enabled, the PreambleDetected IRQ is activated upon successful preamble detection.

**Table 11-6: (G)FSK Packet pbl\_detect Parameter Definition<sup>1</sup>**

| pbl_detect | Description                      |
|------------|----------------------------------|
| 0x00       | Preamble detector length off     |
| 0x08       | Preamble detector length 8 bits  |
| 0x10       | Preamble detector length 16 bits |
| 0x18       | Preamble detector length 24 bits |
| 0x20       | Preamble detector length 32 bits |

1. Usual lengths, other lengths are possible

**long\_preamble\_mode**: Long preamble mode for Rx.

- 1: Enabled. Device detects the preamble, and then switches to Syncword detection. Detect IRQ is issued twice, once on preamble detection, once on Syncword detection.
  - If preamble length of more than 2000 bits is expected, this mode must be enabled.
  - Packets with preambles shorter than 8 bits (minimum detection) + 10us (time to switch the detection) cannot be received.
- 0: Disabled (default).

**pld\_lenUnit** defines the format for payload length definition (**pld\_len**):

- 0x0: **pld\_len** defines payload length in bytes
- 0x1: **pld\_len** defines payload length in bits

**addr\_comp** configures the behaviour of the address comparison function in the LR2021/LR2022/LR2012.

**Table 11-7: (G)FSK Packet AddrComp Parameter Definition**

| AddrComp | Description                                                 |
|----------|-------------------------------------------------------------|
| 0x00     | Address Filtering Disabled                                  |
| 0x01     | Address Filtering activated on Node address                 |
| 0x02     | Address Filtering activated on Node and broadcast addresses |

The LR2021/LR2022/LR2012 wireless transceiver's node address and broadcast address are configured by the [SetFskAddress](#) command. The node address is the specific address assigned to a particular device, while the broadcast address is used to send packets to all devices in the network.

When a packet is received, the LR2021/LR2022/LR2012 performs an address comparison to determine whether the packet is intended for the receiving device. If the address comparison fails, it means that the received packet's destination address does not match the node address or broadcast address configured in the receiver. In this case, the packet reception is aborted, and AddrError IRQ is activated to indicate that the address comparison failed.

**pkt\_format** specifies how the packet length is handled during transmission and reception. It offers four options:

- ♦ **pkt\_format** = 0x0: Fixed length packet mode.  
In this mode, the length of the packet is assumed to be known by both the transmitter and the receiver. Therefore, the packet length does not need to be included in the transmitted packet. The receiver knows the packet length based on its prior knowledge of the packet format.
- ♦ **pkt\_format** = 0x1: Variable length packet mode.  
In this mode, the packet length is not assumed to be known by the receiver. Instead, the transmitter includes the packet length located in an 8-bit header as part of the transmitted packet, and the receiver retrieves this length during packet reception. The explicit mode is used when the packet length is uncertain or variable and needs to be transmitted within the packet.
- ♦ **pkt\_format** = 0x2: Variable length packet mode with backward compatibility with Sx128x chips.  
In this mode, the packet length is not assumed to be known by the receiver. Instead, the transmitter includes the packet length located in a 9-bit header (SX128x compatible) as part of the transmitted packet, and the receiver retrieves this length during packet reception. The explicit mode is used when the packet length is uncertain or variable and needs to be transmitted within the packet.
- ♦ **pkt\_format** = 0x3: Variable length packet with a 16-bit header: The 15 LSBs of the header are the size of the payload (limited to 32767 bytes). The MSB is RFU.

**Table 11-8: (G)FSK Packet pkt\_format Parameter Definition**

| pkt_format | Description                                                                                                                                                   |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x00       | The packet length is known on both sides, the size of the payload is not added to the packet                                                                  |
| 0x01       | The packet has a variable size with an 8-bit header: the 8 bits of the header are the size of the payload. This is SX126x /SX127x compatible                  |
| 0x02       | The packet has a variable size with a 9-bit header: the first 8 bits of the header are the size of the payload, the 9th bit is RFU. This is SX128x compatible |
| 0x03       | The packet has a variable size with a 16-bit header: The 15 LSB of the header are the size of the payload (limited to 32767 bytes). The MSB is RFU            |



**pld\_len** usage depends on whether the transceiver is operating in fixed length packet mode or variable length packet mode:

- In fixed length packet mode, the **pld\_len** parameter specifies the length of the payload in bytes or bits (depending on **pld\_lenUnit**) to be received after the header. The transceiver expects all incoming packets to have a fixed payload length defined by this parameter. When a packet is received, the transceiver processes exactly the number of bytes specified by **pld\_len** as the payload.
- Variable Length Packet Mode: the **pld\_len** parameter defines the maximum length of the packet that can be received. Unlike fixed length packets, variable length packets can have different payload sizes within the defined maximum length.

If a received packet has a payload length greater than the value set by **pld\_len**, the LenError IRQ (Interrupt Request) is raised, together with an RxDone IRQ. The LenError IRQ alerts the system to potential issues with the packet.

**Table 11-9: (G)FSK packet CRC Parameter Definition**

| CRC  | Description                                       |
|------|---------------------------------------------------|
| 0x00 | CrcOff (No CRC)                                   |
| 0x01 | Crc1Byte (CRC computed on 1 byte)                 |
| 0x02 | Crc2Byte(CRC computed on 2 bytes)                 |
| 0x03 | Crc3Byte(CRC computed on 3 bytes)                 |
| 0x04 | Crc4Byte(CRC computed on 4 bytes)                 |
| 0x09 | Crc1ByteInv(CRC computed on 1 byte and inverted)  |
| 0x0A | Crc2ByteInv(CRC computed on 2 bytes and inverted) |
| 0x0B | Crc3ByteInv(CRC computed on 3 bytes and inverted) |
| 0x0C | Crc4ByteInv(CRC computed on 4 bytes and inverted) |

The **Crc** (Cyclic Redundancy Check) parameter defines the CRC types available. Depending on the packet type, i.e., fixed packet or variable packet length, the **Crc** is applied differently:

- Fixed Packet Length Mode: The **Crc** parameter defines the length and type of the CRC to be used. The CRC is applied solely to the payload of the packet. The payload refers to the actual data being transmitted or received, excluding any additional headers or control information.
- Variable Packet Length Mode: The **Crc** parameter is still responsible for defining the length and type of the CRC, but, in this mode, the CRC is applied to both the payload and the header of the packet. The header includes information such as packet length, addressing, and other control details. The CRC calculation covers both header and payload to ensure overall data integrity.

Regardless of the mode, the calculated CRC bytes are appended to the end of the packet frame before transmission. This approach allows the receiving end to verify the integrity of the received packet by performing the same CRC calculation and comparing the result with the received CRC bytes. If the CRC verification fails, it indicates that the data might have been corrupted during transmission.

**dc\_free** controls the whitening mechanism applied to the data portion of the frame following the Syncword. When the **dc\_free** parameter is activated (set to 0x01), the whitening mechanism is enabled for the data portion of the frame. This means that before transmission, the data is subjected to a whitening process specified in the [SetFskWhiteningParams](#) command, where a pseudo-random sequence is XORed with the original data. This process helps to spread the energy of the data over a wider frequency range, reducing the likelihood of long sequences of zeros or ones that might cause synchronization issues.

**Table 11-10: (G)FSK packet dc-free Parameter Definition**

| DcFree | Description                                                  |
|--------|--------------------------------------------------------------|
| 0x00   | No whitening                                                 |
| 0x01   | Whitening according to <a href="#">SetFskWhiteningParams</a> |

### 11.3.3 SetFskWhiteningParams

The *SetFskWhiteningParams* command configures the whitening mechanism used in FSK packets.

This whitening scheme is compatible with sx127x, sx126x/lr11xx and sx128x transceivers.

Whitening helps to improve communication performance by spreading data energy over a wider frequency range, reducing synchronization issues.

**Table 11-11: SetFskWhiteningParams Command**

| Byte           | 0          | 1         | 2                              | 3                 |
|----------------|------------|-----------|--------------------------------|-------------------|
| Data from host | 0x02       | 0x42      | whiten_type(3:0)<br>Init(11:8) | Init(7:0)         |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus (31:24)              | IrqStatus (23:16) |

**whiten\_type** has two options:

- ♦ 0x0: Compatible with SX126x and LR11xx devices
- ♦ 0x1: Compatible with SX128x devices

The **Init** parameter sets the whitening seed value for the chosen whitening scheme.

### 11.3.4 SetFskCrcParams

The *SetFskCrcParams* command configures the CRC (Cyclic Redundancy Check) parameters for FSK packets.

**Table 11-12: SetFskCrcParams Command**

| Byte           | 0           | 1          | 2                 | 3                 | 4                | 5               | 6            | 7            | 8           | 9          |
|----------------|-------------|------------|-------------------|-------------------|------------------|-----------------|--------------|--------------|-------------|------------|
| Data from host | 0x02        | 0x43       | polynom (31:24)   | polynom (23:16)   | polynom (15:8)   | polynom (7:0)   | Init (31:24) | Init (23:16) | Init (15:8) | Init (7:0) |
| Data to host   | Stat (15:8) | Stat (7:0) | IrqStatus (31:24) | IrqStatus (23:16) | IrqStatus (15:8) | IrqStatus (7:0) | 0x00         | 0x00         | 0x00        | 0x00       |

**polynom** CRC polynomial (example 0x1021 ->  $x^{12} + x^5 + 1$ ). This flexibility permits the user to select any standard CRC or to use their own CRC, allowing a specific detection of a given packet. Examples:

- To use the IBM CRC configuration, the user must select:
  - ♦ • 0xFFFF for the initial value
  - ♦ • 0x8005 for the CRC polynomial
  - ♦ • 0x02 (CRC\_2\_BYTE) for the field **Crc** in command [SetFskPacketParams](#)
- For the CCITT CRC configuration the user must select:
  - ♦ • 0x1D0F for the initial value
  - ♦ • 0x1021 for the CRC polynomial
  - ♦ • 0x0A (CRC\_2\_BYTE\_INV) for the field **Crc** in command [SetFskPacketParams](#)

**Init** sets the initial LFSR (Linear Feedback Shift Register) value used to compute the CRC.

### 11.3.5 SetFskSyncword

The *SetFskSyncword* command configures the Syncword for FSK packets.

**Table 11-13: SetFskSyncword Command**

| Byte           | 0          | 1         | 2                | 3                | ... | 9              | 10                        |
|----------------|------------|-----------|------------------|------------------|-----|----------------|---------------------------|
| Data from host | 0x02       | 0x44      | Syncword (63:56) | Syncword (55:48) | ... | Syncword (7:0) | bit_order<br>nb_bits(6:0) |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) | IrqStatus(23:16) | ... | 0x00           | 0x00                      |

**Syncword** defines the synchronization word used during detection if the preamble is not used.

**bit\_order** determines the order in which the synchronization word is transmitted over the air. It can take two possible values:

- ♦ 0 = LSB first: The synchronization word is transmitted with the Least Significant Bit (LSB) first.
- ♦ 1 = MSB first: The synchronization word is transmitted with the Most Significant Bit (MSB) first. To ensure compatibility with SX126x, LR11xx, and SX128x, the **bit\_order** must be set to 1 = MSB first.

**nb\_bits** specifies the length of the Syncword used for detection. If **nb\_bits** < 64, the Least Significant Bits (LSBs) of the 64-bit Syncwords are sent, regardless of the **bit\_order** value.

### 11.3.6 SetFskAddress

The *SetFskAddress* command configures the addresses for filtering FSK packets.

**Table 11-14: SetFskAddress Command**

| Byte           | 0          | 1         | 2                 | 3                 |
|----------------|------------|-----------|-------------------|-------------------|
| Data from host | 0x02       | 0x45      | addr_node (7:0)   | addr_bcast (7:0)  |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus (31:24) | IrqStatus (23:16) |

**addr\_node** defines the node address (of one particular device).

**addr\_bcast** defines the broadcast address (of the network).

### 11.3.7 GetFskRxStats

The *GetFskRxStats* command accesses the internal statistics of received packets.

**Table 11-15: GetFskRxStats Command Request**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x02       | 0x46      |
| Data to host   | Stat(15:8) | Stat(7:0) |

**Table 11-16: GetFskRxStats Response**

| Byte           | 0              | 1             | 2                | 3               | 4                       | 5                      | 6                  | 7                 |
|----------------|----------------|---------------|------------------|-----------------|-------------------------|------------------------|--------------------|-------------------|
| Data from host | 0x00           | 0x00          | 0x00             | 0x00            | 0x00                    | 0x00                   | 0x00               | 0x00              |
| Data to host   | Stat<br>(15:8) | Stat<br>(7:0) | pkt_rx<br>(15:8) | pkt_rx<br>(7:0) | pkt_crc_error<br>(15:8) | pkt_crc_error<br>(7:0) | LenError<br>(15:8) | LenError<br>(7:0) |

| Byte           | 8                 | 9                | 10                | 11               | 12                  | 13                 | 14                | 15               |
|----------------|-------------------|------------------|-------------------|------------------|---------------------|--------------------|-------------------|------------------|
| Data from host | 0x00              | 0x00             | 0x00              | 0x00             | 0x00                | 0x00               | 0x00              | 0x00             |
| Data to host   | pbl_det<br>(15:8) | pbl_det<br>(7:0) | sync_ok<br>(15:8) | sync_ok<br>(7:0) | sync_fail<br>(15:8) | sync_fail<br>(7:0) | timeout<br>(15:8) | timeout<br>(7:0) |

Gets the internal statistics of the received packets. Returned values are:

- **pkt\_rx**: Total number of received packets
- **crc\_error**: Number of received packets with a CRC error
- **len\_error**: Number of packets with a length error
- **pbl\_det**: Number of detections
- **sync\_ok**: Number of correct found Syncwords
- **sync\_fail**: Number of failed Syncwords
- **timeout**: Number of rtc timeouts

Statistics are reset on a POR, sleep without memory retention and the command [ResetRxStats](#).

### 11.3.8 GetFskPacketStatus

The *GetFskPacketStatus* command retrieves the status of the last received packet.

**Table 11-17: GetFskPacketStatus Request Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x02       | 0x47      |
| Data to host   | Stat(15:8) | Stat(7:0) |

Status is updated at the end of a reception (RxDone IRQ), but *rss\_i\_sync* is already updated on SyncwordValid IRQ.

**Table 11-18: GetFskPacketStatus Response**

| Byte           | 0          | 1         | 2                 | 3                | 4                  | 5                   | 6                                                                                          | 7    |
|----------------|------------|-----------|-------------------|------------------|--------------------|---------------------|--------------------------------------------------------------------------------------------|------|
| Data from host | 0x00       | 0x00      | 0x00              | 0x00             | 0x00               | 0x00                | 0x00                                                                                       | 0x00 |
| Data to host   | Stat(15:8) | Stat(7:0) | pkt_len<br>(15:8) | pkt_len<br>(7:0) | rss_i_avg<br>(8:1) | rss_i_sync<br>(8:1) | rfu(1:0)<br>AddrMatchBcast<br>AddrMatchNode<br>rfu<br>rss_i_avg(0)<br>rfu<br>rss_i_sync(0) | Lqi  |

The following values are returned by the command:

- ♦ **pkt\_len**: Length of the last received packet in bytes (including optional data added in the FIFO, CRC,...)
- ♦ **rss\_i\_avg**: Average Received Signal Strength Indication (RSSI) value over the last received packet. The actual signal power can be calculated as  $-rss\_i\_avg/2$  (dBm)
- ♦ **rss\_i\_sync**: Parameter gives the RSSI value latched at Syncword detection. Actual signal power can be calculated as  $-rss\_i\_sync/2$  (dBm)
- ♦ **AddrMatchBcast**: Tells if the last packet received matched the broadcast address
- ♦ **AddrMatchNode**: Tells if the last packet received matched the node address
- ♦ **Lqi**: Link quality indicator (in 0.25 dB steps). LQI is the ratio in dB between detection correlation peak and average power. LQI of >10 dB indicates good quality (in absence of FEC).

## 12. Wireless M-Bus (WM-BUS) (LR20xx)

The LR2021/LR2022/LR2012 wireless chip provides comprehensive support for various WM-BUS modes, making it a good solution for a wide range of applications.

### 12.1 WM-BUS Modulation

The LR2021/LR2022/LR2012 supports the following modulation types.

**Table 12-1: Supported WM-BUS Modulation**

| Mode                     | Modulation Type            | Chip Rate               | Frequency Deviation                             | Notes    |
|--------------------------|----------------------------|-------------------------|-------------------------------------------------|----------|
| S                        | 2FSK<br>Manchester encoded | 32.768 kcps<br>+/- 1.5% | +/- 50 kHz<br>-20% +60%                         | Rx/Tx    |
| T1                       | 2FSK<br>3 out of 6         | 100 kcps<br>+/- 12%     | +/-50 kHz<br>-20% +60%                          | Tx only  |
| T2                       | 2FSK<br>Manchester encoded | 32.768 kcps<br>+/- 1.5% | +/-50 kHz<br>-20% +60%                          | Meter Rx |
| T2                       | 2FSK<br>3 out of 6         | 100 kcps<br>+/- 10%     | +/-50 kHz<br>-20% +60%                          | Meter Tx |
| R2                       | 2FSK<br>Manchester encoded | 4.8 kcps<br>+/- 1.5%    | +/-6 kHz<br>+/-20%                              | Rx/Tx    |
| C1                       | 2FSK                       | 100 kcps<br>+/-100 ppm  | +/-45 kHz<br>+/-25%                             | Tx only  |
| C2                       | 2GFSK<br>BT=0.5            | 50 kcps<br>+/- 100 ppm  | +/-25 kHz<br>+/-25%                             | Meter Rx |
| C2                       | 2FSK                       | 100 kcps<br>+/- 100 ppm | +/-45 kHz<br>+/-25%                             | Meter Tx |
| F2                       | 2FSK                       | 2.4 kcps<br>+/- 100 ppm | +/-5.5 kHz<br>-11%, +28%                        | Rx/Tx    |
| N<br>2.4 kbps            | 2GFSK<br>BT=0.5            | 2.4 kcps<br>+/- 100 ppm | +/-2.4 kHz<br>+/-30%                            | Rx/Tx    |
| N<br>4.8 kbps            | 2GFSK<br>BT=0.5            | 4.8 kcps<br>+/-100 ppm  | +/-2.4 kHz<br>+/-30%                            | Rx/Tx    |
| N 19.2 kbps <sup>1</sup> | 4GFSK<br>BT=0.5            | 19.2 kcps<br>+/-100 ppm | -7.2 kHz, -2.4 kHz<br>2.4 kHz, 7.2 kHz<br>+/-5% | Tx       |

1. Requires a Firmware patch, as described in [22.3 Firmware Patch RAM \(PRAM\)](#). The maximum variation of the frequency deviation is 5%, which differs from the 30% specified in the WM-BUS standard, yet, it is compatible with all model transmitters with frequency synthesis.

## 12.2 WM-BUS Packet Handler

The generic packet handler supports the two WM-BUS packet formats, STD\_WM-BUS A and B. Both header and payload bit ordering follow MSB first (Big Endian) convention. On the receive side, the choice between the two formats is automatically determined by the detected Syncword, as long as the standard is set to either STD\_WM-BUS A or B. The result of multiple CRC checks is available in the status register, along with the number of CRC checks performed.

The settings of the packet handler are defined through packet type definitions and commands described below.

## 12.3 WM-BUS Radio Commands

### 12.3.1 SetWmbusParams

The *SetWmbusParams* command configures the WM-BUS mode in accordance with EN13757-4 2019 standards

**Table 12-2: SetWmbusParams Command**

| Byte           | 0            | 1                | 2                | 3                   | 4                 | 5               |
|----------------|--------------|------------------|------------------|---------------------|-------------------|-----------------|
| Data from host | 0x02         | 0x6A             | Mode(7:0)        | rx_bw(7:0)          | pkt_formatTx(7:0) | addr_comp(7:0)  |
| Data to host   | Stat(15:8)   | Stat(7:0)        | IrqStatus(31:24) | IrqStatus(23:16)    | IrqStatus(15:8)   | IrqStatus(7:10) |
|                |              |                  |                  |                     |                   |                 |
| Byte           | 6            | 7                | 8                | 9                   |                   |                 |
| Data from host | pId_len(7:0) | pbl_len_tx(15:8) | pbl_len_tx(7:0)  | pbl_len_detect(7:0) |                   |                 |
| Data to host   | 0x00         | 0x00             | 0x00             | 0x00                |                   |                 |

**Mode** defines the WM-BUS mode of operation according to EN13757-4 2019, and can take the following values:

- ♦ 0x0 = S (S1 or S2 mode)
- ♦ 0x1 = T1
- ♦ 0x2 = T2 meterRx
- ♦ 0x3 = T2 meterTx
- ♦ 0x4 = R2
- ♦ 0x5 = C1
- ♦ 0x6 = C2 meterRx
- ♦ 0x7 = C2 meterTx
- ♦ 0x8 = N 4.8 kbps
- ♦ 0x9 = N 2.4 kbps
- ♦ 0xA = Reserved
- ♦ 0xB = N 19.2 kbps\*
- ♦ 0xC = F2

\*Requires a Firmware patch, as described in [22.3 Firmware Patch RAM \(PRAM\)](#). The maximum variation of the frequency deviation is 5%, which differs from the 30% specified in the WM-BUS standard.

**rx\_bw** defines the receiver bandwidth used during reception. If **rx\_bw** is set to 0xFF, the receiver bandwidth is automatically set to the optimal value based on frequency error tolerance allowed by the standard, bit rate and frequency deviation.

***pkt\_formatTx*** sets the packet format:

- ♦ 0: STD\_WM-BUS A
- ♦ 1: STD\_WM-BUS B

***addr\_comp*** defines the address comparison feature:

- ♦ 0x00: Address Filtering Disabled
- ♦ 0x01: Address Filtering Activated on Node address

***pbl\_len*** defines the payload size to be sent, including C, M, A, and CI fields, and excluding CRCs during transmission. In reception, it represents the maximum L-field size accepted by the receiver.

***pbl\_len\_tx*** defines the length (in bits) of the preamble during transmission. Note: For 4-FSK, the number of symbols sent is  $pbl\_len\_tx/2$ , corresponding to the asked *pbl\_len\_tx* bits.

***pbl\_len\_detect*** defines the length (in bits) of the preamble to be used in reception. For manual settings, Semtech recommends to use 32 bits in all cases except C1 and C2 mode, where 24 bits are sufficient.

- ♦ 0xFF: The preamble length is automatically computed based on ***pbl\_len\_tx***

### 12.3.2 GetWmbusRxStats

The *GetWmbusRxStats* command retrieves the internal statistics of received packets in WM-BUS mode.

**Table 12-3: GetWmbusRxStats Request**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x02       | 0x6C      |
| Data to host   | Stat(15:8) | Stat(7:0) |

**Table 12-4: GetWmbusRxStats Response**

| Byte           | 0          | 1         | 2            | 3           | 4                       | 5                      | 6                  | 7                 |
|----------------|------------|-----------|--------------|-------------|-------------------------|------------------------|--------------------|-------------------|
| Data from host | 0x00       | 0x00      | 0x00         | 0x00        | 0x00                    | 0x00                   | 0x00               | 0x00              |
| Data to host   | Stat(15:8) | Stat(7:0) | pkt_rx(15:8) | pkt_rx(7:0) | pkt_crc_error<br>(15:8) | pkt_crc_error<br>(7:0) | LenError<br>(15:8) | LenError<br>(7:0) |

The command returns the following values:

- ♦ ***pkt\_rx***: Total number of received packets
- ♦ ***pkt\_crc\_error***: number of received packets with a CRC error
- ♦ ***LenError***: number of packets with a length error

The statistics are reset under the following conditions: Power-on reset (POR), sleep without memory retention, or the execution of the *ResetRxStats* command.



### 12.3.3 GetWmbusPacketStatus

The *GetWmbusPacketStatus* command obtains the status of the last received packet in WM-BUS mode.

**Table 12-5: GetWmbusPacketStatus Request**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x02       | 0x6D      |
| Data to host   | Stat(15:8) | Stat(7:0) |

**Table 12-6: GetWmbusPacketStatus Response**

| Byte           | 0              | 1             | 2               | 3                 | 4                | 5                  | 6                   | 7                  | 8                 | 9                                                | 10           |
|----------------|----------------|---------------|-----------------|-------------------|------------------|--------------------|---------------------|--------------------|-------------------|--------------------------------------------------|--------------|
| Data from host | 0x00           | 0x00          | 0x00            | 0x00              | 0x00             | 0x00               | 0x00                | 0x00               | 0x00              | 0x00                                             | 0x00         |
|                |                |               |                 |                   |                  |                    |                     |                    |                   | SwIndex<br>CrcError(0)                           |              |
| Data to host   | Stat<br>(15:8) | Stat<br>(7:0) | LField<br>(7:0) | pkt_len<br>(15:8) | pkt_len<br>(7:0) | rss_i_avg<br>(8:1) | rss_i_sync<br>(8:1) | CrcError<br>(16:9) | CrcError<br>(8:1) | rfu<br>rss_i_avg(0)<br>rfu(2:0)<br>rss_i_sync(0) | Lqi<br>(7:0) |

Status is updated at the end of a reception (RxDone IRQ), but *SwIndex* and *rss\_i\_sync* are already updated on SyncwordValid IRQ.

The command returns the following values:

- ♦ **LField**: The demodulated length field. This can be different from `get_rx_packet_length` (in mode A).
- ♦ **pkt\_len**: Length of the last received packet in bytes (including optional data added in the FIFO, CRC, ...).
- ♦ **rss\_i\_avg**: Average over the last received packet of RSSI (Received Signal Strength Indication). The actual signal power can be calculated as  $-rss\_i\_avg/2$  (dBm).
- ♦ **rss\_i\_sync**: Latched RSSI value after Syncword detection. The actual signal power can be calculated as  $-rss\_i\_sync/2$  (dBm).
- ♦ **CrcError** (17 bits): Indicates which CRC has failed. In STD\_WM-BUS A, bit 0 represents the CRC of the header, and each subsequent bit corresponds to the CRC of the subsequent chunks.
- ♦ **SwIndex**: Indicates the received mode:
  - ♦ 0: Mode A received
  - ♦ 1: Mode B received
- ♦ **Lqi**: Link quality indicator (in 0.25 dB steps)

### 12.3.4 SetWmbusFilteringAddress

The *SetWmbusFilteringAddress* command configures the address filtering mechanism (A-field) for WM-BUS communication.

**Table 12-7: SetWmbusFilteringAddress Command**

| Byte           | 0          | 1         | 2                | 3                | 4               | 5               | 6              | 7             |
|----------------|------------|-----------|------------------|------------------|-----------------|-----------------|----------------|---------------|
| Data from host | 0x02       | 0x6E      | Address (47:40)  | Address (39:32)  | Address (31:24) | Address (23:16) | Address (15:8) | Address (7:0) |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) | IrqStatus(23:16) | IrqStatus(15:8) | IrqStatus(7:0)  | 0x00           | 0x00          |

**Address:** Sets the WM-Bus address for Rx filtering i.e the A-Field.

## 13. Wi-SUN FSK (LR20xx)

### 13.1 Wi-SUN Modulation

The LR2021/LR2022/LR2012 implements a 2FSK modem, enabling support for both optional and mandatory Wi-SUN profiles, such as ECHONET, FAN, RLMM, and JUTA. The table below illustrates the supported Wi-SUN operating modes and their corresponding symbol rates.

**Table 13-1: Supported Wi-SUN Modulation**

| Modes and Sub modes | Modulation type | Symbol rate             | Modulation index |
|---------------------|-----------------|-------------------------|------------------|
| 1a                  | 2FSK            | 50 ksps<br>+/-100 ppm   | 0.5              |
| 1b                  | 2FSK            | 50 ksps<br>+/-100 ppm   | 1                |
| 2a                  | 2FSK            | 100 ksps<br>+/-100 ppm  | 0.5              |
| 2b                  | 2FSK            | 100 ksps<br>+/-100 ppm  | 1                |
| 3                   | 2FSK            | 150 ksps<br>+/-100 ppm  | 0.5              |
| 4a                  | 2FSK            | 200 ksps<br>+/-100 ppm  | 0.5              |
| 4b                  | 2FSK            | 200 ksps<br>+/- 100 ppm | 1                |
| 5                   | 2FSK            | 300 ksps<br>+/- 100 ppm | 0.5              |

### 13.2 Wi-SUN Packet Handler

The chip supports the Wi-SUN 2FSK physical layer standard and dual synch word.

The settings of the packet handler are defined via packet type definitions and commands as described below.

## 13.3 Wi-SUN Radio Commands

### 13.3.1 SetWisunMode

The *SetWisunMode* command configures the Wi-SUN mode of operation. This command sets and customizes the LR2021/LR2022/LR2012's behavior according to the desired Wi-SUN profile and specifications.

**Table 13-2: SetWisunMode Command**

| Byte           | 0          | 1         | 2                | 3                |
|----------------|------------|-----------|------------------|------------------|
| Data from host | 0x02       | 0x70      | Mode(7:0)        | rx_bw(7:0)       |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) | IrqStatus(23:16) |

**Mode** can take the following values:

- ♦ 0x0 = 1a
- ♦ 0x1 = 1b
- ♦ 0x2 = 2a
- ♦ 0x3 = 2b
- ♦ 0x4 = 3
- ♦ 0x5 = 4a
- ♦ 0x6 = 4b
- ♦ 0x7 = 5

**rx\_bw** defines the receiver bandwidth used during packet reception, as defined in the *SetFskModulationParams* function. It is optional, and if not explicitly defined, it is automatically determined by the chip based on the operating mode and other relevant factors.

- ♦ 0xFF: Bandwidth is automatically set by the firmware

### 13.3.2 SetWisunPacketParams

The *SetWisunPacketParams* command configures the Wi-SUN packet parameters. This command customizes and sets various parameters related to Wi-SUN packet transmission and reception, tailoring the LR2021/LR2022/LR2012 behavior according to specific Wi-SUN requirements and operational needs.

**Table 13-3: SetWisunPacketParams Command**

| Byte           | 0          | 1         | 2                   | 3                      | 4                     | 5                   | 6                   |
|----------------|------------|-----------|---------------------|------------------------|-----------------------|---------------------|---------------------|
|                |            |           | rfu(1:0)            |                        |                       |                     |                     |
|                |            |           | fcs_tx              |                        |                       |                     |                     |
| Data from host | 0x02       | 0x71      | Whitening<br>crc_on | frame_len_tx<br>(15:8) | frame_len_tx<br>(7:0) | pbl_len_tx<br>(7:0) | pbl_detect<br>(7:0) |
|                |            |           | mode_switch_tx      |                        |                       |                     |                     |
|                |            |           | fec_tx(1:0)         |                        |                       |                     |                     |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24)    | IrqStatus(23:16)       | IrqStatus(15:8)       | IrqStatus(7:0)      | 0                   |

**fcs\_tx** defines the CRC (Cyclic Redundancy Check) type used for packet transmission:

- ♦ 0 = 4 bytes CRC
- ♦ 1 = 2 bytes CRC

**Whitening** determines the whitening mode for data transmission:

- ♦ 0 = No whitening (test mode)
- ♦ 1 = Whitening

**crc\_on** defines the CRC computation mode for packet transmission:

- ♦ 0 = CRC is provided in the payload
- ♦ 1 = CRC is automatically calculated by the device

**mode\_switch\_tx** selects the payload type to be transmitted:

- ♦ 0 = Normal payload
- ♦ 1 = Mode switch payload

**fec\_tx** selects the FEC (Forward Error Correction) encoding to be used in Tx:

- ♦ 0x0 = None
- ♦ 0x1 = NRNSC (Non-Recursive Non-Systematic Convolutional) with interleaver
- ♦ 0x2 = RSC (Recursive Systematic Convolutional) without interleaver
- ♦ 0x3 = RSC with interleaver (RSC is only for Tx)

**frame\_len\_tx** defines the length of the frame for normal packets or the header for mode switch packets during transmission.

**pbl\_len\_tx** defines the length, in bits, of the transmitted preamble.

**pbl\_detect** defines the detection mode on the preamble (16, 32 bits).

- ♦ 0x00: Detection is done on the Syncword
- ♦ no intermediate value
- ♦ 0x10: detection of 16 bits
- ♦ ---
- ♦ 0x12: detection on 18 bits
- ♦ ---
- ♦ 0x20: detection of 32 bits
- ♦ no intermediate value
- ♦ 0xFF(default): Detection is done on 24 bits (**pbl\_detect** must be smaller or equal to **pbl\_len\_tx**)

### 13.3.3 GetWisunRxStats

The *GetWisunRxStats* command retrieves the internal statistics of received packets in Wi-SUN mode.

**Table 13-4: GetWisunRxStats Request Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x02       | 0x72      |
| Data to host   | Stat(15:8) | Stat(7:0) |

**Table 13-5: GetWisunRxStats Response**

| Byte           | 0           | 1          | 2             | 3            | 4               | 5              | 6               | 7              |
|----------------|-------------|------------|---------------|--------------|-----------------|----------------|-----------------|----------------|
| Data from host | 0x00        | 0x00       | 0x00          | 0x00         | 0x00            | 0x00           | 0x00            | 0x00           |
| Data to host   | Stat (15:8) | Stat (7:0) | pkt_rx (15:8) | pkt_rx (7:0) | CrcError (15:8) | CrcError (7:0) | LenError (15:8) | LenError (7:0) |

The command returns the following values:

- ♦ **pkt\_rx**: Total number of received packets
- ♦ **CrcError**: Number of received packets with a CRC error
- ♦ **LenError**: Number of packets with a length error

The statistics are reset under the following conditions: Power-on reset (POR), sleep without memory retention, or the execution of the *ResetRxStats* command.

### 13.3.4 GetWisunPacketStatus

The *GetWisunPacketStatus* command obtains the status of the last received packet in Wi-SUN mode.

**Table 13-6: GetWisunPacketStatus Request Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x02       | 0x73      |
| Data to host   | Stat(15:8) | Stat(7:0) |

**Table 13-7: GetWisunPacketStatus Response**

| Byte           | 0              | 1             | 2                | 3               | 4                 | 5                | 6                  | 7                   | 8                                                               | 9            |
|----------------|----------------|---------------|------------------|-----------------|-------------------|------------------|--------------------|---------------------|-----------------------------------------------------------------|--------------|
| Data from host | 0x00           | 0x00          | 0x00             | 0x00            | 0x00              | 0x00             | 0x00               | 0x00                | 0x00                                                            | 0x00         |
| Data to host   | Stat<br>(15:8) | Stat<br>(7:0) | Header<br>(15:8) | Header<br>(7:0) | pkt_len<br>(15:8) | pkt_len<br>(7:0) | rss_i_avg<br>(8:1) | rss_i_sync<br>(8:1) | SyncwordIdx<br>rfu(3:0)<br>rss_i_avg(0)<br>rfu<br>rss_i_sync(0) | Lqi<br>(7:0) |

The status is updated at the end of a reception (RxDone IRQ), but SyncwordIdx and rss\_i\_sync are already updated on SyncwordValid IRQ.

The values returned by the command are:

- ♦ **Header:** Raw 16 bits header
- ♦ **pkt\_len:** Length of the packet in bytes (including optional data added in the FIFO, CRC...)
- ♦ **SyncwordIdx** (0 or 1): Index of the detected Syncword
- ♦ **rss\_i\_avg:** Average over the last received packet of RSSI (Received Signal Strength Indication). The actual signal power can be calculated as  $-rss\_i\_avg/2$  (dBm)
- ♦ **rss\_i\_sync:** RSSI value latched at Syncword detection. The actual signal power can be calculated as  $-rss\_i\_sync/2$  (dBm).
- ♦ **Lqi:** Link quality indicator (in 0.25 dB steps)

### 13.3.5 SetWisunPacketLen

The *SetWisunPacketLen* command configures the Wi-SUN packet length for Tx normal mode or the header value for mode switch packets.

**Table 13-8: SetWisunPacketLen Command**

| Byte           | 0          | 1         | 3                   | 4                  |
|----------------|------------|-----------|---------------------|--------------------|
| Data from host | 0x02       | 0x74      | frame_len_tx (15:8) | frame_len_tx (7:0) |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(21:24)    | IrqStatus(13:16)   |

*frame\_len\_tx* is either packet length for normal payload, or Header value for mode switch payload.

## 14. Bluetooth Low Energy Physical Layer (LR2021/LR2022)

Semtech's products are designed to be used in connection with qualified **Bluetooth®** products and applications but are not certified or qualified Bluetooth products.

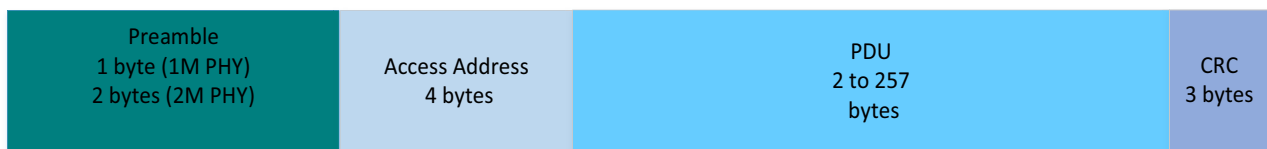
### 14.1 Bluetooth LE Modulation

The CPFSK demodulator and generic packet modulator support the physical layer of Bluetooth Low Energy up to version 5.1, both coded and uncoded. The modulation used is GMSK (Gaussian Minimum Shift Keying) with a symbol rate of 1 or 2 Msps.

### 14.2 Bluetooth LE Packet Handler

The Bluetooth Low Energy packet format is illustrated in the diagram below. It consists of one or two bytes of preamble, followed by 4 bytes of access address, a Protocol Data Unit (PDU), and 3 CRC bytes. Headers are not generated by the transceiver and should be externally calculated and included as part of the payload to the data buffer.

**Figure 14-1: Bluetooth Low Energy Packet**



### 14.3 Bluetooth LE Radio Commands

#### 14.3.1 SetBleModulationParams

The *SetBleModulationParams* command configures the modulation parameters for Bluetooth LE packets.

**Table 14-1: SetBleModulationParams Command**

| Byte           | 0          | 1         | 2                | 3                |
|----------------|------------|-----------|------------------|------------------|
| Data from host | 0x02       | 0x60      | mode(7:0)        | rx_bw(7:0)       |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) | IrqStatus(23:16) |

**mode** defines the actual modulation scheme to be used, and it can take the following values:

- ♦ 0x0: LE 1M PHY (1 Mbps)
- ♦ 0x1: LE 2M PHY (2 Mbps)
- ♦ 0x2: LE Coded S = 2 (500 kbps)
- ♦ 0x3: LE Coded S = 8 (125 kbps)

For all LE coded modes, contact Semtech.

**rx\_bw** (optional) manually forces the Rx bandwidth. Calculated as defined in *SetFskModulationParams*. If not provided, or set to 0xFF, the optimal bandwidth is selected automatically based on the mode.



### 14.3.2 SetBleChannelParams

The *SetBleChannelParams* command sets the Bluetooth LE channel-dependent parameters and configure the packet parameters for Bluetooth LE packets.

**Table 14-2: SetBleChannelParams Command**

| Byte           | 0          | 1         | 2                                            | 3                | 4               | 5              |
|----------------|------------|-----------|----------------------------------------------|------------------|-----------------|----------------|
| Data from host | 0x02       | 0x61      | rfu(2:0)<br>crc_in_fifo<br>channel_type(3:0) | whit_init(7:0)   | crc_init(23:16) | crc_init(15:8) |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24)                             | IrqStatus(23:16) | IrqStatus(15:8) | IrqStatus(7:0) |

| Byte           | 6             | 7               | 8               | 9              | 10            |
|----------------|---------------|-----------------|-----------------|----------------|---------------|
| Data from host | crc_init(7:0) | Syncword(31:24) | Syncword(23:16) | Syncword(15:8) | Syncword(7:0) |
| Data to host   | 0x00          | 0x00            | 0x00            | 0x00           | 0x00          |

**crc\_in\_fifo** appends, or not, the CRC in the Rx FIFO:

- ♦ 0x0: CRC is not appended to the FIFO in Rx mode
- ♦ 0x1: CRC is appended to the FIFO in Rx mode

**channel\_type** selects the packet type, either advertiser or data packet:

- ♦ 0x0: Advertiser packet
- ♦ 0x1: Data packet with 16-bit header
- ♦ 0x2: Data packet with 24-bit header

**whit\_init** provides the initial value for the whitening process, which is channel dependent in Bluetooth LE. Example: 0x53 for channel 37. Can be set to 0 to deactivate whitening (test mode).

**crc\_init** indicates the seed to be used during CRC computation. For example, **crc\_init** = 0x555555 is used for advertising.

**Syncword** defines the access address. Example: 0x8E89BED6 for advertising.

### 14.3.3 SetBleTxPduLen

The *SetBleTxPduLen* command configures the PDU (Protocol Data Unit) length of the Bluetooth LE packet to be sent.

**Table 14-3: SetBleTxPduLen Command**

| Byte           | 0          | 1         | 2                |
|----------------|------------|-----------|------------------|
| Data from host | 0x02       | 0x66      | pdu_len(7:0)     |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) |

**pdu\_len** represents the length of the PDU including the 16-bit header but excluding the 3-byte CRC.

The data to be written into the FIFO corresponds to this PDU, including the 16-bit header.

### 14.3.4 SetBleTx

The *SetBleTx* command configures the PDU length for transmission and send a Bluetooth Low Energy packet.

This command combines the functionalities of *SetBleTxPduLen(pld\_len)* and *SetTx(0)*.

**Table 14-4: SetBleTx Command**

| Byte           | 0          | 1         | 2                 |
|----------------|------------|-----------|-------------------|
| Data from host | 0x02       | 0x62      | pld_len           |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus (31:24) |

*pld\_len* represents the length of the PDU (including the 16-bit header) in the Bluetooth LE packet, excluding the 3-byte CRC. The data to be written into the FIFO corresponds to this PDU.

### 14.3.5 GetBleRxStats

The *GetBleRxStats* command retrieves the internal statistics of received packets.

**Table 14-5: GetBleRxStats Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x02       | 0x64      |
| Data to host   | Stat(15:8) | Stat(7:0) |

**Table 14-6: GetBleRxStats Response**

| Byte           | 0              | 1             | 2                | 3               | 4                  | 5                 | 6                  | 7                 |
|----------------|----------------|---------------|------------------|-----------------|--------------------|-------------------|--------------------|-------------------|
| Data from host | 0x00           | 0x00          | 0x00             | 0x00            | 0x00               | 0x00              | 0x00               | 0x00              |
| Data to host   | stat<br>(15:8) | stat<br>(7:0) | pkt_rx<br>(15:8) | pkt_rx<br>(7:0) | CrcError<br>(15:8) | CrcError<br>(7:0) | LenError<br>(15:8) | LenError<br>(7:0) |

The command returns the following values:

- ♦ **pkt\_rx**: Total number of received packets
- ♦ **CrcError**: Number of received packets with a CRC error
- ♦ **LenError**: Number of packets with a length error

The statistics are reset under the following conditions: Power-on reset (POR), sleep without memory retention, or execution of the *ResetRxStats* command.

### 14.3.6 GetBlePacketStatus

The *GetBlePacketStatus* command obtains the status of the last received packet in Bluetooth LE mode.

**Table 14-7: GetBlePacketStatus Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x02       | 0x65      |
| Data to host   | Stat(15:8) | Stat(7:0) |

**Table 14-8: GetBlePacketStatus Response**

| Byte           | 0              | 1             | 2                 | 3                | 4                  | 5                   | 6                                                   | 7            |
|----------------|----------------|---------------|-------------------|------------------|--------------------|---------------------|-----------------------------------------------------|--------------|
| Data from host | 0x00           | 0x00          | 0x00              | 0x00             | 0x00               | 0x00                | 0x00                                                | 0x00         |
| Data to host   | stat<br>(15:8) | stat<br>(7:0) | pkt_len<br>(15:8) | pkt_len<br>(7:0) | rss_i_avg<br>(8:1) | rss_i_sync<br>(8:1) | rfu(4:0)<br>rss_i_avg(0)<br>rfu(0)<br>rss_i_sync(0) | Lqi<br>(7:0) |

The status is updated at the end of a reception (RxDone IRQ), but *rss\_i\_sync* is already updated on SyncwordValid IRQ.

The command returns the following values:

- ♦ **pkt\_len**: The length of the last received packet in bytes (including optional data added in the FIFO, CRC, ...).
- ♦ **rss\_i\_avg**: The average Received Signal Strength Indication (RSSI) value over the last received packet. The actual signal power is calculated as  $-rss\_i\_avg/2$  (dBm).
- ♦ **rss\_i\_sync**: The RSSI value latched after Syncword detection. The actual signal power is calculated as  $-rss\_i\_sync/2$  (dBm).
- ♦ **Lqi**: The link quality indicator in 0.25 dB steps.

## 15. BPSK Modem (LR20xx)

The LR2021/LR2022/LR2012 supports a Tx-only BPSK modulation, primarily designed for low bit rates of up to 600 bps (bits per second). This modulation technique enables efficient transmission of data in applications that require slower communication and longer range capabilities.

### 15.1 (D)BPSK Modulation

The BPSK (Binary Phase-Shift Keying) and (D)BPSK (Differential Binary Phase-Shift Keying) are straightforward digital modulation techniques that use phase positions (0 or  $\pi$  radian) to encode binary values (0 or 1). In BPSK modulation, the phase directly represents the binary value, while in (D)BPSK modulation, the phase shift occurs only when there is a change in the binary value (from 0 to 1 or from 1 to 0).

### 15.2 BPSK Radio Commands

#### 15.2.1 SetBpskModulationParams

The *SetBpskModulationParams* command configures the modulation parameters for BPSK packets. It ensures that the correct modulation settings are applied for BPSK modulation. If the packet type is not BPSK, the command returns CMD\_FAIL in the status of the next command, indicating that the modulation parameters cannot be set as intended for BPSK packets.

**Table 15-1: SetBpskModulationParams Command**

| Byte           | 0              | 1             | 2                    | 3                    | 4                   | 5                  | 6                                                           |
|----------------|----------------|---------------|----------------------|----------------------|---------------------|--------------------|-------------------------------------------------------------|
| Data from host | 0x02           | 0x50          | BitRate(31:24)       | BitRate(23:16)       | BitRate(15:8)       | BitRate(7:0)       | pulse_shape(3:0)<br>rfu<br>diff_mode<br>diff_mode_init(1:0) |
| Data to host   | Stat<br>(15:8) | Stat<br>(7:0) | IrqStatus<br>(31:24) | IrqStatus<br>(23:16) | IrqStatus<br>(15:8) | IrqStatus<br>(7:0) | 0x00                                                        |

**BitRate** has the same meaning as in FSK modulation parameters.

**pulse\_shape** defines the symbol pulse shaping:

- ♦ 0x0 No pulse shaping
- ♦ 0x1 RFU
- ♦ 0x2 Gaussian with BT 2.0
- ♦ 0x3 Root-Raised-Cosine (RRC) with Roll-Off 0.4
- ♦ 0x4 Gaussian with BT 0.3
- ♦ 0x5 Gaussian with BT 0.5
- ♦ 0x6 Gaussian with BT 0.7
- ♦ 0x7 Gaussian with BT 1.0
- ♦ 0x8 Raised-Cosine with Roll-Off 0.3
- ♦ 0x9 Raised-Cosine with Roll-Off 0.5

- ♦ 0xA Raised-Cosine with Roll-Off 0.7
- ♦ 0xB Raised-Cosine with Roll-Off 1.0
- ♦ 0xC Root-Raised-Cosine with Roll-Off 0.3
- ♦ 0xD Root-Raised-Cosine with Roll-Off 0.5
- ♦ 0xE Root-Raised-Cosine with Roll-Off 0.7
- ♦ 0xF Root-Raised-Cosine with Roll-Off 1.0

**diff\_mode** enables the differential encoder:

- ♦ 1 to enable differential encoding (DBPSK)
- ♦ 0 to disable differential encoding (BPSK)

**diff\_mode\_init** sets the initial value for the differential encoder:

- ♦ Bit 0 is the previous value, and bit 1 is the parity
- ♦ This parameter is discarded if **diff\_mode** = 0

## 15.2.2 SetBpskPacketParams

The *SetBPSKPacketParams* command configures the packet parameters for BPSK packets. It ensures that the correct settings related to BPSK packet structure are applied. If the packet type is not BPSK, the command returns CMD\_FAIL in the status of the next command, indicating that the packet parameters cannot be set as intended for BPSK packets.

**Table 15-2: SetBpskPacketParams Command**

| Byte           | 0          | 1         | 2                 | 3                |
|----------------|------------|-----------|-------------------|------------------|
|                |            |           |                   | rfu(1:0)         |
|                |            |           |                   | Mode(1:0)        |
| Data from host | 0x02       | 0x51      | pld_len<br>(7:0)  | rfu              |
|                |            |           |                   | bpsk_s_msg       |
|                |            |           |                   | bpsk_s_rank(1:0) |
| Data to host   | Stat(15:8) | Stat(7:0) | Irq status(31:24) | IrqStatus(23:16) |

**pld\_len** defines the payload length in bits when **Mode** is 0 (raw encoded), or bytes when **Mode** is 1.

**Mode** defines how the content of the data stored in the FIFO is handled:

- ♦ 0x0: Raw encoded: Nothing is added to the data put in FIFO
- ♦ 0x1: Bpsk\_s PHY. The data in FIFO corresponds to the UL-CONTAINER. CRC, convolution, frame type and preamble are handled by the chip.

**bpsk\_s\_msg** defines the Sigfox message type:

- ♦ 0: Application message
- ♦ 1: Control message

The **bpsk\_s\_rank** defines the Sigfox frame emission rank:

- ♦ 0x0: Bpsk\_s multiple frame -first frame / single frame
- ♦ 0x1: Bpsk\_s multiple frame -second frame
- ♦ 0x2: Bpsk\_s multiple frame -third frame

## 16. OOK Modem (LR20xx)

### 16.1 OOK Modulation

The LR2021 supports OOK modulation, which is applied by switching on and off the power amplifier. It uses digital control and ramping features to enhance the transient power response of the OOK transmitter. Moreover, modulation shaping can be applied to further improve the narrow band response of the LR2021 transmitter. RSSI based discriminator is also integrated in the chip as well as a complete packet engine.

### 16.2 OOK Packet Handler

This packet handler works in exactly the same way as the (G)FSK handler. Refer to [Section 11.2](#) for details.

### 16.3 OOK Commands

#### 16.3.1 SetOokModulationParams

The *SetOokModulationParams* command configures the parameters used during OOK modulation.

**Table 16-1: SetOokModulationParams Command**

| Byte           | 0          | 1         | 2                    | 3                    | 4                   | 5                  | 6                    | 7              | 8              |
|----------------|------------|-----------|----------------------|----------------------|---------------------|--------------------|----------------------|----------------|----------------|
| Data from host | 0x02       | 0x81      | BitRate<br>(31:24)   | BitRate<br>(23:16)   | BitRate<br>(15:8)   | BitRate<br>(7:0)   | pulse_shape<br>(7:0) | rx_bw<br>(7:0) | Depth<br>(7:0) |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus<br>(31:24) | IrqStatus<br>(23:16) | IrqStatus<br>(15:8) | IrqStatus<br>(7:0) | 0                    | 0              | 0              |

**BitRate** determines the bit rate and its interpretation depends on the value of BitRate(31).

- ♦ If **BitRate**(31) is 0, the **BitRate** is in bps and is represented by BitRate(30:0).
- ♦ If **BitRate**(31) is 1, the **BitRate** in bps is calculated as BitRate(30:0)/256.

**pulse\_shape** defines the symbol pulse shaping employed during modulation and follows the same definition as used for FSK.

**rx\_bw** defines the receiver bandwidth used during OOK reception and shares the same definition as used for FSK. See [Table 11-4](#).

**Depth** sets the magnitude depth, and it is an optional argument. There are two options:

- ♦ 0x00: OOK\_DEPTH\_FULL (default): The depth is limited by the Power Amplifier (PA) and may vary based on the output power, external components, and BitRate.
- ♦ 0x01: OOK\_DEPTH\_UP\_TO\_20DB: The depth is limited to a maximum of 20 dB. The exact value needs to be measured on the device and depends on the external components and BitRate.

### 16.3.2 SetOokPacketParams

The *SetOokPacketParams* command sets the packet parameters for OOK packets. It returns a failure (CMD\_FAIL) if the packet type is not OOK.

**Table 16-2: SetOokPacketParams Command**

| Byte           | 0          | 1         | 2                    | 3                    | 4                                             | 5                  | 6                | 7                               |
|----------------|------------|-----------|----------------------|----------------------|-----------------------------------------------|--------------------|------------------|---------------------------------|
| Data from host | 0x02       | 0x82      | pre_len_tx<br>(15:8) | pre_len_tx<br>(7:0)  | rfu(3:0)<br>addr_comp(1:0)<br>pkt_format(1:0) | pld_len<br>(15:8)  | pld_len<br>(7:0) | Crc(3:0)<br>Manchester<br>(3:0) |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus<br>(31:24) | IrqStatus<br>(23:16) | IrqStatus<br>(15:8)                           | IrqStatus<br>(7:0) | 0                | 0                               |

**pre\_len\_tx** defines the number of symbols in the preamble and follows the same definition as FSK.

**pkt\_format** select fixed or variable packet length and uses the same definitions and values as for FSK (*SetFskPacketParams* command), except that value 0x2 (SX128x compatibility) is not available.

- ♦ 0x0: Fixed length
- ♦ 0x1: Variable length
- ♦ 0x2: Not valid value for OOK
- ♦ 0x3: Variable length packet with 16-bit header

**addr\_comp** sets the behavior of the address comparison function in a similar fashion to FSK.

**pld\_len** defines the length of the payload in bytes to be received after the header for fixed-length packets, or it defines the max length of the packet to be received for variable length packets. In variable length mode, if a packet is received indicating a length superior to **pld\_len**, the LenError IRQ is raised.

**Crc** defines the length and type of the CRC and uses the same definition as for FSK (*SetFskPacketParams* command). CRC is applied to the payload in case of fixed packet length and applied to the payload and header in case of variable packet length. The CRC byte(s) are appended to the end of the frame.

**Manchester** defines if Manchester coding is applied to the frame:

- ♦ 0: OFF
- ♦ 1: ON
- ♦ 3: ON\_INV

Semtech recommends to have either whitening or Manchester encoding enabled for OOK.

### 16.3.3 SetOokCrcParams

The *SetOokCrcParams* command configures the CRC parameters for OOK packets.

**Table 16-3: SetOokCrcParams Command**

| Byte           | 0          | 1         | 2                | 3                | 4               | 5              | 6            |
|----------------|------------|-----------|------------------|------------------|-----------------|----------------|--------------|
| Data from host | 0x02       | 0x83      | Polynom (31:24)  | Polynom (23:16)  | Polynom (15:8)  | Polynom (7:0)  | Init (31:24) |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) | IrqStatus(23:16) | IrqStatus(15:8) | IrqStatus(7:0) | 0            |

| Byte           | 7            | 8           | 9          |
|----------------|--------------|-------------|------------|
| Data from host | Init (23:16) | Init (15:8) | Init (7:0) |
| Data to host   | 0            | 0           | 0          |

**Polynom** defines the polynomial used to generate the CRC information.

**Init** sets the initial LFSR (Linear Feedback Shift Register) value used for CRC computation.

### 16.3.4 SetOokSyncword

The *SetOokSyncword* command configures the synchronization word (Syncword) for OOK packets.

**Table 16-4: SetOokSyncword Command**

| Byte           | 0          | 1         | 2                | 3                | 3                | 9              | 10                        |
|----------------|------------|-----------|------------------|------------------|------------------|----------------|---------------------------|
| Data from host | 0x02       | 0x84      | Syncword(31:24)  | Syncword(23:16)  | Syncword(15:8)   | Syncword (7:0) | bit_order<br>nb_bits(6:0) |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) | IrqStatus(23:16) | IrqStatus(23:16) | IrqStatus(7:0) | 0                         |

**Syncword:** Defines:

- The synchronization word to be checked, if preamble detected is used
- The synchronization word used during detection, if preamble is not used

**bit\_order** determines the order in which the synchronization word is sent over the air:

- 0: LSB first
- 1: MSB first (MSB first must be selected to be compatible with SX126x, LR11xx, and SX128x)

**nb\_bits** specifies the length of the Syncword used for detection:

- If **nb\_bits** is less than 64, only the LSBs of the 64-bit **Syncwords** are sent, irrespective of the **bit\_order** value.



### 16.3.5 SetOokAddress

The *SetOokAddress* command configures address filtering for OOK packets. This command allows you to set specific addresses that are used for filtering incoming OOK packets.

**Table 16-5: SetOokAddress Command**

| Byte           | 0          | 1         | 2                | 3                |
|----------------|------------|-----------|------------------|------------------|
| Data from host | 0x02       | 0x85      | addr_node (7:0)  | addr_bcast (7:0) |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) | IrqStatus(23:16) |

**addr\_node** defines the node address for OOK packets.

**addr\_bcast** defines the broadcast address for OOK packets, same as used for FSK.

### 16.3.6 GetOokRxStats

The *GetOokRxStats* command retrieves the internal statistics of the received packets for OOK modulation.

**Table 16-6: GetOokRxStats Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x02       | 0x86      |
| Data to host   | Stat(15:8) | Stat(7:0) |

**Table 16-7: GetOokRxStats Response**

| Byte           | 0              | 1             | 2                | 3               | 4                  | 5                 | 6                  | 7                 |
|----------------|----------------|---------------|------------------|-----------------|--------------------|-------------------|--------------------|-------------------|
| Data from host | 0x00           | 0x00          | 0x00             | 0x00            | 0x00               | 0x00              | 0x00               | 0x00              |
| Data to host   | Stat<br>(15:8) | Stat<br>(7:0) | pkt_rx<br>(15:8) | pkt_rx<br>(7:0) | CrcError<br>(15:8) | CrcError<br>(7:0) | LenError<br>(15:8) | LenError<br>(7:0) |

The *GetOokRxStats* command returns the following values:

- ♦ **pkt\_rx**: total number of received packets
- ♦ **CrcError**: number of received packets with a CRC error
- ♦ **LenError**: number of packets with a length error

These statistics are reset in the following situations: power-on reset (POR), sleep without memory retention, or executing the [ResetRxStats](#) command.

### 16.3.7 GetOokPacketStatus

The *GetOokPacketStatus* command obtains the status of the last received packet.

**Table 16-8: GetOokPacketStatus Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x02       | 0x87      |
| Data to host   | Stat(15:8) | Stat(7:0) |

The indicators are updated at the end of a reception or transmission

**Table 16-9: GetOokPacketStatus Response**

| Byte           | 0              | 1             | 2                 | 3                | 4                  | 5                 | 6                                                                                        | 7            |
|----------------|----------------|---------------|-------------------|------------------|--------------------|-------------------|------------------------------------------------------------------------------------------|--------------|
| Data from host | 0x00           | 0x00          | 0x00              | 0x00             | 0x00               | 0x00              | 0x00                                                                                     | 0x00         |
| Data to host   | Stat<br>(15:8) | Stat<br>(7:0) | pkt_len<br>(15:8) | pkt_len<br>(7:0) | rss_i_avg<br>(8:1) | RssiHigh<br>(8:1) | rfu(1:0)<br>AddrMatchBcast<br>AddrMatchNode<br>rfu<br>rss_i_avg(0)<br>rfu<br>RssiHigh(0) | Lqi<br>(7:0) |

The *GetOokPacketStatus* command returns the following values:

- ♦ **pkt\_len**: Length of the last received packet in bytes (including optional data added in the FIFO, CRC, ...).
- ♦ **rss\_i\_avg**: Average over the last received packet of Received Signal Strength Indicator (RSSI). The actual signal power is calculated as  $-rss\_i\_avg/2$  (in dBm).
- ♦ **RssiHigh**: RSSI value of the high bits ('1'). The actual signal power is calculated as  $-RssiHigh/2$  (in dBm).
- ♦ **AddrMatchBcast**: Indicates whether the received packet matched the broadcast address, when address comparison (**addr\_comp**) is enabled.
- ♦ **AddrMatchNode**: Indicates whether the received packet matched the node address, when address comparison (**addr\_comp**) is enabled.
- ♦ **Lqi**: Link quality indicator in 0.25 dB steps.

### 16.3.8 SetOokDetector

The *SetOokDetector* command configures for receiver mode, the OOK detection process.

**Table 16-10: SetOokDetector Command**

| Byte           | 0          | 1         | 2                          | 3                         | 4                                   | 5                                        | 6                                                    |
|----------------|------------|-----------|----------------------------|---------------------------|-------------------------------------|------------------------------------------|------------------------------------------------------|
| Data from host | 0x02       | 0x88      | preamble_<br>pattern(15:8) | preamble_<br>pattern(7:0) | rfu(3:0)<br>pattern_length<br>(3:0) | rfu(2:0)<br>pattern_num_<br>repeats(4:0) | rfu(1:0)<br>sw_is_raw<br>sfd_kind<br>sfd_length(3:0) |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24)           | IrqStatus(23:16)          | IrqStatus(15:8)                     | IrqStatus(7:0)                           | 0x00                                                 |

**preamble\_pattern** defines the detection pattern with the following constraint/recommendation:

- ♦ The LSB is the first bit received
- ♦ The pattern's 2 LSB must be either 01 or 10
- ♦ For ADS-B detection, the pattern is 0x0285
- ♦ In Tx the detection pattern must be part of the data Tx FIFO

**pattern\_length** defines the length of the pattern -1 bit to be used for detection (for a pattern length of 2, set this parameter to 1).

**pattern\_num\_repeats** defines the number of repetitions of the preamble pattern, to control the timeout for Syncword search:

- ♦ 31 (max value) disables the timeout of Syncword search
- ♦ For ADS-B this parameter must be set to 0

**sw\_is\_raw** defines if the Syncword is encoded:

- ♦ 0: Syncword is encoded same way as the payload
- ♦ 1: Syncword is not encoded

**sfd\_kind** defines the type for start of frame delimiter:

- ♦ 0: Falling edge. Must be set to 0 for ADS-B
- ♦ 1: Rising edge

**sfd\_length** defines length of the start of frame delimiter (sfd) in bits:

- ♦ Must be set to 0 for ADS-B

### 16.3.9 SetOokwhiteningParams

The *SetOokWhiteningParams* command configures the chip whitening scheme to be used in OOK.

**Table 16-11: SetOokwhiteningParams Command**

| Byte           | 0          | 1         | 2                             | 3                | 4                      | 5              |
|----------------|------------|-----------|-------------------------------|------------------|------------------------|----------------|
| Data from host | 0x02       | 0x89      | bit_idx(3:0)<br>Polynom(11:8) | Polynom(7:0)     | rfu(3:0)<br>Init(11:8) | Init(7:0)      |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24)              | IrqStatus(23:16) | IrqStatus(15:8)        | IrqStatus(7:0) |

**bit\_idx** defines the index of the LFSR bit used to whiten the data.

**Polynom** defines the 12-bit whitening polynomial:

- 0: Disables the whitening

**Init** defines the initial value for the LFSR used for whitening the data.

Semtech recommends to have either whitening or Manchester encoding enabled when using OOK modulation.

## 17. LR-FHSS (LR20xx)

### 17.1 LR-FHSS Modulation

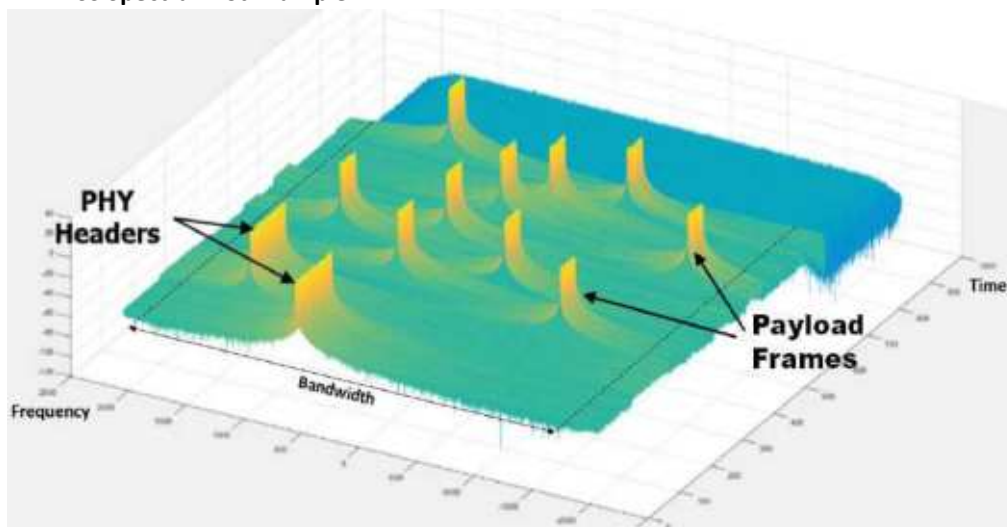
LR-FHSS (Frequency Hopping Spread Spectrum) is a modulation technique specifically designed for long-range applications in dense areas, making it ideal for scenarios where numerous devices need to communicate over an extended area. LR-FHSS can achieve a link-budget performance similar to LoRa SF12 BW125 kHz, enabling multiple devices to coexist within the same spectral allocation. In the LR20xx, LR-FHSS is implemented as a transmit-only mode. The LR20xx takes full control over the following functionalities:

- ♦ Frame construction: This includes the generation of preambles, sync headers, payload, integrity check, whitening, interleaving, and forward error correction calculations.
- ♦ Calculation of hopping sequences: The LR20xx calculates and manages the hopping sequences used in frequency hopping.
- ♦ Handling the entire frequency hopping mechanism: The LR20xx efficiently manages the process of frequency hopping, ensuring reliable communication.
- ♦ With LR-FHSS in the LR20xx, developers can enjoy the benefits of long-range communication while accommodating a large number of devices operating simultaneously in the same frequency band.

LR-FHSS (Long Range Frequency Hopping Spread Spectrum) is a specialized modulation technique derived from GMSK (Gaussian Minimum Shift Keying). It operates at a low bit rate, with the ability to configure intra-packet frequency hopping sequences for individual devices. Presently, the LR-FHSS supports a bit rate of 488.28215 Hz. The payload uses convolution encoding for forward error detection and correction, which introduces a transmission overhead.

From a spectral perspective, the LR-FHSS packet is divided into multiple frames, each transmitted over pseudo-randomly distributed frequencies, with configurable spacing known as the "Grid" or step size. The total spectral occupancy of the LR-FHSS packet is determined by the Bandwidth parameter. For a visual representation of LR-FHSS modulation, refer to [Figure 17-1: LR-FHSS Spectral Plot Example](#).

**Figure 17-1: LR-FHSS Spectral Plot Example**



In LR-FHSS modulation, each payload frame (except the last frame) consists of precisely 50 encoded bits, ensuring equal time duration for each frame. The number of frames in a transmission varies based on the payload size and Coding Rate. The last frame may contain fewer than 50 bits, accommodating the remaining payload data.

Prior to the payload frames, one to four synchronization sequences (Headers) are included, ensuring stable frequency and timing synchronization. Although a single Header is sufficient for decoding the payload, using a minimum of 2 Headers enhances immunity against radio interference. When multiple headers are used, they are transmitted on different frequencies. Each header comprises precisely 114 encoded bits, ensuring uniform time duration for each header. For more comprehensive details and information on LR-FHSS, refer to AN1200.58 Long Range FHSS Demo and AN1200.64 Long Range FHSS System Performance documentation.

## 17.2 LR-FHSS Commands

### 17.2.1 LrFhssBuildFrame

The *LrFhssBuildFrame* command performs two key functions:

- encodes the provided payload
- sets up the internal hopping table

When the parameters are valid, and the payload can be encoded within the length limitation, the command returns CMD\_OK. However, if there are errors with the parameters or the payload length exceeds the limitation, the command returns CMD\_PERR.

This command does not initiate the packet transmission. Instead, it prepares the encoded payload and hopping table for future transmission.

To send the packet, this command must be followed by the normal *SetTx* command, which triggers the actual transmission process.

**Table 17-1: LrFhssBuildFrame Command**

| Byte           | 0              | 1             | 2                               | 3                          | 4                       | 5                       | 6                 | 7               |
|----------------|----------------|---------------|---------------------------------|----------------------------|-------------------------|-------------------------|-------------------|-----------------|
| Data from host | 0x02           | 0x56          | sync_header_cnt(3:0)<br>Cr(3:0) | mod_type(3:0)<br>Grid(3:0) | Hopping(3:0)<br>Bw(3:0) | rfu(6:0)<br>Sequence(8) | Sequence<br>(7:0) | Offset<br>(7:0) |
| Data to host   | Stat<br>(15:8) | Stat<br>(7:0) | IrqStatus<br>(31:24)            | IrqStatus<br>(23:16)       | Irq status<br>(15:8)    | IrqStatus<br>(7:0)      | 0x00              | 0x00            |

| Byte           | 8          | 9          | ..  | N+8        |
|----------------|------------|------------|-----|------------|
| Data from host | Payload[0] | Payload[1] | ... | Payload[N] |
| Data to host   | 0x00       | 0x00       | ... | 0x00       |

**sync\_header\_cnt** defines the number of sent sync headers.

- Value can be 1,2,3 or 4 sync headers

**Cr** defines the FEC used in LR-FHSS.

- 0x00: CR = 5/6
- 0x01: CR = 2/3
- 0x02: CR = 1/2
- 0x03: CR = 1/3

**mod\_type** defines the physical layer modulation to be used. Only one defined as of today.

- Mode = 0: GMSK, 488.28125 Hz

**Grid** set the frequency grid space used for hopping sequences

- 0: 25.390625 kHz
- 1: 3.90625 kHz

**Hopping** defines the hopping mode and switch OFF hopping if desired:

- 0x0: No hopping
- 0x1: Hopping
- 0x2: Test mode (payload encoded as hopping, but no actual hopping)
- 0x3: Test mode (no hopping, but PA ramps up/down as if we would do the hopping)

**Bw** defines the frequency range span of the hopping sequence: [0:9] according to LR-FHSS specifications:

- ♦ 0x00: 39.06 kHz
- ♦ 0x01: 85.94 kHz
- ♦ 0x02: 136.72 kHz
- ♦ 0x03: 183.59 kHz
- ♦ 0x04: 335.94 kHz
- ♦ 0x05: 386.72 kHz
- ♦ 0x06: 722.66 kHz
- ♦ 0x07: 773.44 kHz
- ♦ 0x08: 1523.4 kHz
- ♦ 0x09: 1574.2 kHz
- ♦ Other values are RFU

**Sequence** defines the seed of the hopping sequence (9 bits, where **Sequence** bit-8 is in byte 5 of the command).

**Offset** defines the allowed signed values per device frequency offset to reduce the risk of packet collision.

Frequency offset (Hz) = DeviceOffset \* 488.28125Hz:

- ♦ Signed value in [-26; 25] for Grid = 25.390625 kHz
- ♦ Signed values in [-4; 3] for Grid = 3.90625 kHz

For an FCC use case, DeviceOffset shall not be changed once configured.

**Payload** is the actual payload to be encoded. Returns CMD\_OK if the parameters are OK.

## 17.2.2 LrFhssSetSyncword

The *LrFhssSetSyncword* command configures the synchronization word used for LR-FHSS detection on the receiver side, and for building the Tx frame on the transmitter side.

**Table 17-2: LrFhssSetSyncword Command**

| Byte           | 0          | 1         | 2                | 3                | 4               | 5              |
|----------------|------------|-----------|------------------|------------------|-----------------|----------------|
| Data from host | 0x02       | 0x57      | Syncword (31:24) | Syncword (23:16) | Syncword (15:8) | Syncword (7:0) |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) | IrqStatus(23:16) | IrqStatus(15:8) | IrqStatus(7:0) |

**Note:** The Reset value for Syncword is 0x2C0F7995.

### 17.2.3 ReadLrFhssHoppingTable

Returns the internal LR-FHSS hopping table.

**Table 17-3: ReadLrFhssHoppingTable Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x02       | 0x58      |
| Data to host   | Stat(15:8) | Stat(7:0) |

If hopping is not enabled, only freq0 is returned, otherwise  $N$  frequency/nb\_symbols couples are returned, where  $N = \min(\text{nb\_used\_freqs}, \text{nb\_hopping\_blocks})$ , and has a maximum value of 40.

- byte0: ctrl
  - ♦ bit0: hopping\_en
  - ♦ bit7: convert\_freq (if frequencies are provided in Hz, not PLL steps)
- byte1: pkt\_length(15:8)
- byte2: pkt\_length(7:0)
- byte3: nb\_used\_freqs (how many frequencies are used)
- byte4: nb\_hopping\_blocks
- byte5: freq0(31:24)
- byte6: freq0(23:16)
- byte7: freq0(15:8)
- byte8: freq0(7:0)
- byte9: nb\_symbols0(15:8)
- byte10: nb\_symbols0(7:0)

### 17.2.4 WriteLrFhssHoppingTable

Writes the internal LR-FHSS hopping table. See [ReadLrFhssHoppingTable](#) for data to send.

**Table 17-4: ReadLrFhssHoppingTable Command**

| Byte           | 0          | 1         | 2                     | ... | 41  |
|----------------|------------|-----------|-----------------------|-----|-----|
| Data from host | 0x02       | 0x59      | data                  | ... | ... |
| Data to host   | Stat(15:8) | Stat(7:0) | Irq Status<br>(31:24) | ... | ... |



## 18. FLRC Modem (LR2021)

### 18.1 FLRC Modulation

The Fast Long Range Communication (FLRC) modem is designed for high-speed communication, utilizing a combination of GMSK (Gaussian Minimum Shift Keying) with forward error correction and interleaving techniques. These features significantly enhance receiver sensitivity, resulting in a 5 dB improvement in link budget compared to FSK (Frequency Shift Keying) modulation at the same bit rate.

Developers have access to the FLRC modem's parameters, enabling them to achieve efficient and reliable communication. The available packet type for use with the FLRC modem is the FLRC packet, which is described in detail in section [18.2 FLRC Packet Handler](#).

For applications where maximum sensitivity and low average current consumption are required, Semtech provides the FLRC Protocol (FLRP). FLRP prepends a LoRa® Wake-on-Radio exchange to synchronize the two devices in time and frequency before the FLRC bursts, enabling CR = 1/2 operation with 1–2 dB additional sensitivity gain and substantially reduced average current consumption<sup>1</sup>. More details on FLRP configuration and performance are available in the Semtech application notes on the LoRa Plus product page, and on <https://github.com/Lora-net/usp>.

More details on the use and performance of FLRC can be found in AN1200.101 on Semtech website.

#### 18.1.1 Modem Bandwidth and Bit Rates

The FLRC modem supports high bit rates, ranging from 130 kbps to 2.6 Mbps, with available modulation bandwidths spanning from 300 kHz to 2.6 MHz. Not all combinations of bandwidth and bit rate are supported. To configure the desired raw bit rate, the [SetFlrcModulationParams](#) command is used, where the first parameter selects a valid combination of raw bit rate and double sideband modulation bandwidth.

**Table 18-1: Valid FLRC Bit Rate and Bandwidth Combinations**

| Symbol               | Raw Bit Rate<br>Rb<br>[Mbps] | Bandwidth<br>BW<br>[MHz <i>DSB</i> ] |
|----------------------|------------------------------|--------------------------------------|
| FLRC_BR_2_600_BW_2_6 | 2.6                          | 2.666                                |
| FLRC_BR_2_080_BW_2_2 | 2.08                         | 2.222                                |
| FLRC_BR_1_300_BW_1_3 | 1.3                          | 1.333                                |
| FLRC_BR_1_040_BW_1_3 | 1.04                         | 1.333                                |
| FLRC_BR_0_650_BW_0_7 | 0.65                         | 0.740                                |
| FLRC_BR_0_520_BW_0_6 | 0.52                         | 0.571                                |
| FLRC_BR_0_325_BW_0_3 | 0.325                        | 0.357                                |
| FLRC_BR_0_260_BW_0_3 | 0.26                         | 0.307                                |

1. Requires also the number of correlators to be reduced when the two devices are synchronized in time and frequency, writing 0x00 at register address 0x00F30D18: WriteRegMemMask32(addr = 0x00F30D18, Mask = 0x00000FFF, Value = 0x00000000).

## 18.1.2 FEC Coding Rate

The FLRC modem can use forward error correction (FEC) which uses convolutional coding to add redundant information to the packet, enabling the modem to perform error correction, ensuring the packet payload information remains robust, even when exposed to bursts of interference from other radio services on the same band. FEC introduces some overhead, affecting the raw bit rate and effective bit rate.

**Table 18-2: Effective FLRC Bit Rates Based on FEC Usage**

| Symbol               | Raw Programmed Bit Rate<br>Rb [Mbps] | Programmed Coding Rate<br>CR | Effective Bit Rate<br>Rbeff [Mbps] |
|----------------------|--------------------------------------|------------------------------|------------------------------------|
| FLRC_BR_2_600_BW_2_6 | 2.6                                  | 1 <sup>(1)</sup>             | 2.6                                |
|                      | 2.6                                  | 3/4                          | 1.95                               |
|                      | 2.6                                  | 2/3                          | 1.73                               |
|                      | 2.6                                  | 1/2                          | 1.3                                |
| FLRC_BR_2_080_BW_2_2 | 2.08                                 | 1 <sup>(1)</sup>             | 2.08                               |
|                      | 2.08                                 | 3/4                          | 1.56                               |
|                      | 2.08                                 | 2/3                          | 1.38                               |
|                      | 2.08                                 | 1/2                          | 1.04                               |
| FLRC_BR_1_300_BW_1_3 | 1.3                                  | 1 <sup>(1)</sup>             | 1.3                                |
|                      | 1.3                                  | 3/4                          | 0.975                              |
|                      | 1.3                                  | 2/3                          | 0.86                               |
|                      | 1.3                                  | 1/2                          | 0.65                               |
| FLRC_BR_1_040_BW_1_3 | 1.04                                 | 1 <sup>(1)</sup>             | 1.04                               |
|                      | 1.04                                 | 3/4                          | 0.78                               |
|                      | 1.04                                 | 2/3                          | 0.69                               |
|                      | 1.04                                 | 1/2                          | 0.52                               |
| FLRC_BR_0_650_BW_0_7 | 0.65                                 | 1 <sup>(1)</sup>             | 0.65                               |
|                      | 0.65                                 | 3/4                          | 0.488                              |
|                      | 0.65                                 | 2/3                          | 0.433                              |
|                      | 0.65                                 | 1/2                          | 0.325                              |
| FLRC_BR_0_520_BW_0_6 | 0.52                                 | 1 <sup>(1)</sup>             | 0.52                               |
|                      | 0.52                                 | 3/4                          | 0.39                               |
|                      | 0.52                                 | 2/3                          | 0.34                               |
|                      | 0.52                                 | 1/2                          | 0.26                               |
| FLRC_BR_0_325_BW_0_3 | 0.325                                | 1 <sup>(1)</sup>             | 0.325                              |
|                      | 0.325                                | 3/4                          | 0.244                              |
|                      | 0.325                                | 2/3                          | 0.216                              |
|                      | 0.325                                | 1/2                          | 0.163                              |
| FLRC_BR_0_260_BW_0_3 | 0.26                                 | 1 <sup>(1)</sup>             | 0.26                               |
|                      | 0.26                                 | 3/4                          | 0.195                              |
|                      | 0.26                                 | 2/3                          | 0.173                              |
|                      | 0.26                                 | 1/2                          | 0.130                              |

1. CR=1 offers limited advantages over GMSK and is not recommended.

### 18.1.3 Gaussian Filtering

In transmit mode, the FLRC modem offers an optional Gaussian filter, which can be controlled using the parameter BT. This Gaussian filtering function serves to minimize the side-lobe emissions of the transmitted FLRC signal, thus improving spectral efficiency.

The parameter **pulse\_shape** determines the level of filtering applied, with the following valid values in order of decreasing filtering effort: 0.3, 0.5, 1, or OFF. The configuration of the Gaussian filter (BT) is achieved through the [SetFlrcModulationParams](#) command.

### 18.1.4 FLRC Frequency Tolerance

The modem's configuration relies on bit rate parameters set using the [SetFlrcPacketParams](#) and [SetFlrcModulationParams](#) commands. The reception process consists of three phases.

- Initially, a bank of correlators is employed to detect a valid incoming preamble. The number of correlators running depends on the bandwidth and bit rate, ensuring that a frequency offset, as tabulated in the electrical specification, is tolerated.
- Once a valid preamble is detected, the modem proceeds to verify the synchronization word to ensure that the received packet is intended for that specific radio. In the final phase of the demodulation process, the packet data itself is demodulated.
- The acceptable frequency tolerance for each modem setting is summarized in the following table.

**Table 18-3: Receiver Performance of the FLRC Modem**

| Bit Rate<br>[Mbps] | Bandwidth<br>[MHz] | Frequency Tolerance<br>[kHz] | Frequency Tolerance<br>@ 490 MHz<br>[ppm] | Frequency Tolerance<br>@ 915 MHz<br>[ppm] | Frequency Tolerance<br>@ 2450 MHz<br>[ppm] |
|--------------------|--------------------|------------------------------|-------------------------------------------|-------------------------------------------|--------------------------------------------|
| 2.6                | 2.666              | +/- 150                      | +/- 306                                   | +/- 164                                   | +/- 61                                     |
| 2.08               | 2.222              | +/- 150                      | +/- 306                                   | +/- 164                                   | +/- 61                                     |
| 1.3                | 1.333              | +/- 100                      | +/- 204                                   | +/- 109                                   | +/- 41                                     |
| 1.04               | 1.333              | +/- 100                      | +/- 204                                   | +/- 109                                   | +/- 41                                     |
| 0.65               | 0.740              | +/- 70                       | +/- 143                                   | +/- 77                                    | +/- 29                                     |
| 0.52               | 0.571              | +/- 50                       | +/- 102                                   | +/- 55                                    | +/- 20                                     |
| 0.325              | 0.357              | +/- 30                       | +/- 61                                    | +/- 33                                    | +/- 12                                     |
| 0.260              | 0.307              | +/- 25                       | +/- 51                                    | +/- 27                                    | +/- 10                                     |

## 18.2 FLRC Packet Handler

The FLRC packet, although proprietary, follows a conventional construction. It comprises a header, Syncword, and CRC structure. Just like the (G)FSK mode format, the FLRC mode also offers two packet formats for fixed and variable-length packets.

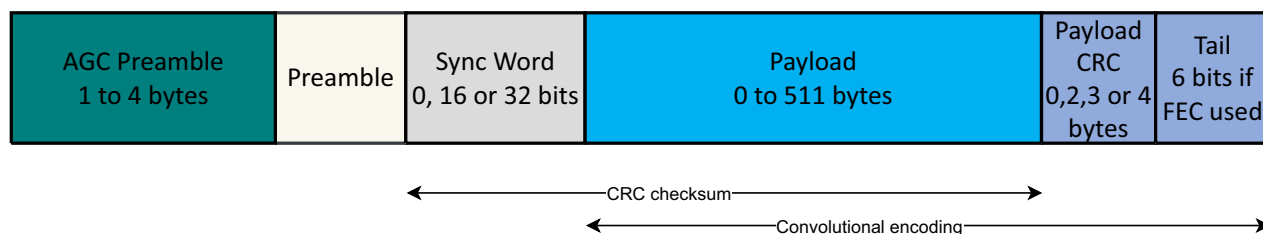
### 18.2.1 Fixed-Length Packet Format

The fixed packet length format is depicted in the diagram below. The packet consists of the following components:

- A variable-length preamble of at least 2 bytes which can be reduced to 1 byte for bit rates lower than 650 kbps.
- A 21-bit timing recovery preamble.
- A configurable Syncword which can take the value of 0, 2 or 4 bytes. The modem can search up to 3 Syncwords at the same time.
- The fixed length payload, which can range from 0 to 511 bytes.
- A CRC (Cyclic Redundancy Check) field, with a length of 0, 2, 3, or 4 bytes.
- Finally, the packet concludes with a short 6-bit sequence of trailing zeros if FEC is enabled.

The sizes of the Syncword, payload length, and CRC length can be configured using the command [SetFlrcPacketParams](#). The CRC calculation is performed on the entire preceding packet, excluding the preamble.

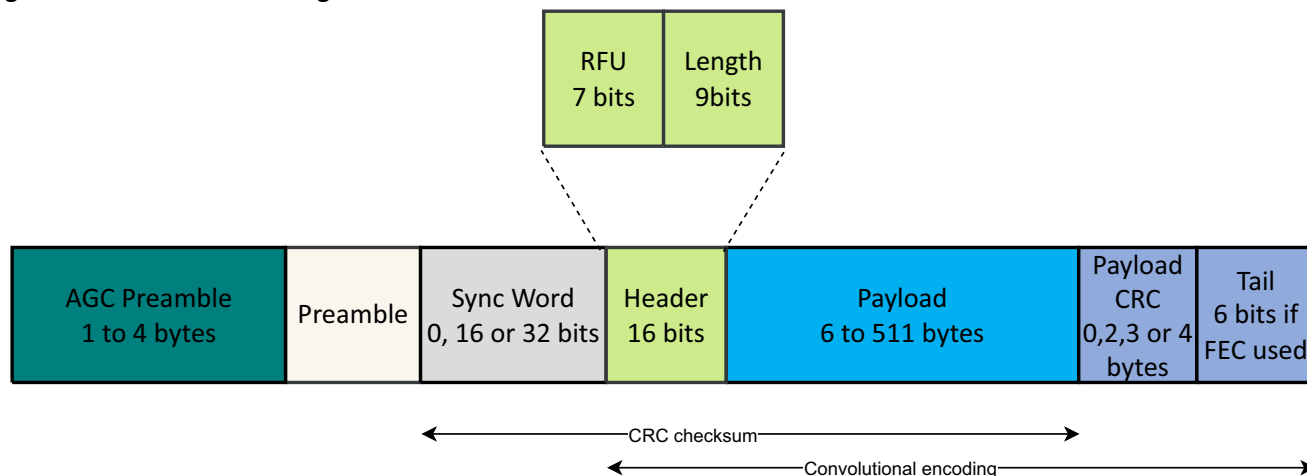
**Figure 18-1: FLRC Fixed-length Packet Format**



### 18.2.2 Variable-length Packet Format

The variable format packet shares the same structure and functionality as the fixed packet length format, with the inclusion of a header to which the CRC and convolutional coding are applied. The minimum payload is 6 bytes. The header structure is fixed and consists of a 2 RFU bits, as shown in the mapping in the figure below. Following the those RFU bits is the payload length information, over 9 bits.

**Figure 18-2: FLRC Variable-length Packet Format**



## 18.3 FLRC Time-on-Air

The total number of bits transmitted in an FLRC packet is as defined in [Figure 18-1: FLRC Fixed-length Packet Format](#). The calculation of the total time-on-air is therefore the combination of the number of payload bits (compensated for the influence of convolutional coding) and the number of header bits. Denoting the number of bits in each field of the packet,  $n$ , the number of header bits is:

$$n_{uncoded} = n_{AGCPreamble} + n_{Preamble} + n_{SyncWord} + n_{header}$$

where  $n_{header}$  is 16 bits if packet format is variable length, = 0 otherwise.

and the effective number of coded bits by:

$$n_{coded} = \text{ceil}\left[(n_{Payload} + n_{CRC} + n_{tail}) \times \left(\frac{1}{n_{CR}}\right)\right]$$

where  $n_{CR}$  is the value programmed as the coding rate, see [Section 18.1.2](#).

$n_{tail}$  depends on the CR:  $n_{tail} = 6$  bit if  $CR = 1/2$  or  $3/4$ ;  $n_{tail} = 0$  in other cases.

The bit period for a given FLRC bit rate is simply:

$$t_{bit} = \frac{1}{R_b}$$

Values of  $R_b$  in FLRC can be found at [Section 18.1.2](#).

and the total packet time-on-air is given by:

$$ToA_{FLRC} = t_{bit} * (n_{uncoded} + n_{coded})$$

## 18.4 FLRC Radio Commands

### 18.4.1 SetFlrcModulationParams

The *SetFlrcModulationParams* command configures the modulation parameters for FLRC packets. If the packet type is not FLRC, the command returns CMD\_FAIL in the status of the next command.

**Table 18-4: SetFlrcModulationParams Command**

| Byte           | 0          | 1         | 2                | 3                           |
|----------------|------------|-----------|------------------|-----------------------------|
| Data from host | 0x02       | 0x48      | bitrate_bw(7:0)  | cr(3:0)<br>pulse_shape(3:0) |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) | IrqStatus(23:16)            |

**bitrate\_bw** defines the raw bit rate (not taken the FEC gain) and bandwidth used for the FLRC modulation:

- ♦ 0x00: BR = 2600 kbps and BW = 2666 kHz
- ♦ 0x01: BR = 2080 kbps and BW = 2222 kHz
- ♦ 0x02: BR = 1300 kbps and BW = 1333 kHz
- ♦ 0x03: BR = 1040 kbps and BW = 1333 kHz
- ♦ 0x04: BR = 650 kbps and BW = 740 kHz
- ♦ 0x05: BR = 520 kbps and BW = 571 kHz
- ♦ 0x06: BR = 325 kbps and BW = 357 kHz
- ♦ 0x07: BR = 260 kbps and BW = 307 kHz

The **cr** parameter defines the coding rate for the FEC used:

- ♦ 0x0: Coding rate = 1/2
- ♦ 0x1: Coding rate = 3/4
- ♦ 0x2: Coding rate = 1
- ♦ 0x3: Coding rate = 2/3

The **pulse\_shape** parameter defines the pulse shaping used. Its values are the same as for FSK, but only the following are valid:

- ♦ 0x0: No pulse shaping
- ♦ 0x4: Gaussian with BT 0.3<sup>1</sup>
- ♦ 0x5: Gaussian with BT 0.5
- ♦ 0x7: Gaussian with BT 1.0

1. Sensitivity degradation to be expected.

## 18.4.2 SetFlrcPacketParams

The *SetFlrcPacketParams* command sets the packet parameters for FLRC packets. If the packet type is not FLRC, the command returns CMD\_FAIL in the status of the next command.

**Table 18-5: SetFlrcPacketParams Command**

| Byte           | 0          | 1         | 2                                             | 3                                                       | 4               | 5              |
|----------------|------------|-----------|-----------------------------------------------|---------------------------------------------------------|-----------------|----------------|
| Data from host | 0x02       | 0x49      | rfu(1:0)<br>agc_pbl_len(3:0)<br>sync_len(1:0) | sync_tx(1:0)<br>sync_match(2:0)<br>pkt_format, Crc(1:0) | pld_len(15:8)   | pld_len(7:0)   |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24)                              | IrqStatus(23:16)                                        | IrqStatus(15:8) | IrqStatus(7:0) |

**agc\_pbl\_len** defines the number of bytes preamble bits to be transmitted in Tx and considered in Rx to define AGC behavior:

- ◆ 0x00: 4 bits
- ◆ 0x01: 8 bits
- ◆ 0x02: 12 bits
- ◆ 0x03: 16 bits
- ◆ 0x04: 20 bits
- ◆ 0x05: 24 bits
- ◆ 0x06: 28 bits
- ◆ 0x07: 32 bits

**sync\_len** defines the length of the Syncword in unit of 2 bytes (0/16/32 bits). It must be set to 0 if **sync\_match** is OFF.

**sync\_tx** defines which Syncword to use for Tx operations:

- ◆ 0 = No Syncword
- ◆ 1..3 = Use Syncword 1, 2 or 3

**sync\_match** defines the synchronization word (Syncword) detection behavior for the FLRC modem. It determines which Syncword patterns the modem searches for during packet reception. The Syncword serves as a predefined bit pattern used to identify the start of a packet. The available options and their meanings are as follows:

- ◆ 0x00 (OFF): Synchronization word detection is disabled. Must be disabled if sync\_len = 0.
- ◆ 0x01 (MATCH\_1): The modem looks for the first specified Syncword pattern to detect packet synchronization.
- ◆ 0x02 (MATCH\_2): The modem looks for the second specified Syncword pattern to detect packet synchronization.
- ◆ 0x03 (MATCH\_1\_OR\_2): Modem looks for either the first or second specified Syncword pattern to detect packet synchronization.
- ◆ 0x04 (MATCH\_3): The modem looks for the third specified Syncword pattern to detect packet synchronization.
- ◆ 0x05 (MATCH\_1\_OR\_3): The modem looks for either the first or third specified Syncword pattern to detect packet synchronization.
- ◆ 0x06 (MATCH\_2\_OR\_3): Modem looks for either the second or third specified Syncword pattern to detect packet synchronization.
- ◆ 0x07 (MATCH\_1\_OR\_2\_OR\_3): The modem looks for any of the three specified Syncword patterns to detect packet synchronization.

**pkt\_format** defines the type of payload:

- ◆ 0: Dynamic, variable payload length, the payload length is part of the frame and recovered during packet reception
- ◆ 1: Fixed, the payload length is fixed and assumed to be known at receiver side

**Crc** configures the CRC length:

- ◆ 0x0: CRC OFF
- ◆ 0x1: CRC on 16 bits
- ◆ 0x2: CRC on 24 bits
- ◆ 0x3: CRC on 32 bits

**pld\_len** determines the length of the payload in the FLRC packets. It specifies the number of data bytes that can be transmitted or received within a single FLRC packet. The allowed range of **pld\_len** is from 6 to 511 bytes. For payloads over 256 bytes, usage of IRQs Rx Fifo and Tx Fifo is necessary.

**Note:** The valid range for **pld\_len** is from 6 to 127 bytes for an SX1280 modem, so ensure that **pld\_len** is within this range to ensure proper functioning of FLRC communication. In Dynamic **pkt\_format**, the device outputs a LenError IRQ, but remains in Rx mode.

### 18.4.3 GetFlrcRxStats

The *GetFlrcRxStats* command retrieves the internal statistics of the received packets in the FLRC modem.

**Table 18-6: GetFlrcRxStats Request Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x02       | 0x4A      |
| Data to host   | Stat(15:8) | Stat(7:0) |

**Table 18-7: GetFlrcRxStats Response**

| Byte           | 0              | 1             | 2                | 3               | 4                  | 5                 | 6                  | 7                 |
|----------------|----------------|---------------|------------------|-----------------|--------------------|-------------------|--------------------|-------------------|
| Data from host | 0x00           | 0x00          | 0x00             | 0x00            | 0x00               | 0x00              | 0x00               | 0x00              |
| Data to host   | Stat<br>(15:8) | Stat<br>(7:0) | pkt_rx<br>(15:0) | pkt_rx<br>(7:0) | CrcError<br>(15:8) | CrcError<br>(7:0) | LenError<br>(15:8) | LenError<br>(7:0) |

The command returns the following values:

- ♦ **pkt\_rx**: Total number of received packets
- ♦ **CrcError**: Number of received packets with a CRC error
- ♦ **LenError**: Number of packets with a length error

The statistics are automatically reset under specific conditions, including power-on reset (POR), sleep without memory retention, or executing the *ResetRxStats* command.



### 18.4.4 GetFlrcPacketStatus

The *GetFlrcPacketStatus* command retrieves the status of the last received packet in the FLRC modem.

**Table 18-8: GetFlrcPacketStatus Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x02       | 0x4B      |
| Data to host   | Stat(15:8) | Stat(7:0) |

**Table 18-9: GetFlrcPacketStatus Response**

| Byte           | 0          | 1         | 2             | 3            | 4              | 5               | 6                                                          |
|----------------|------------|-----------|---------------|--------------|----------------|-----------------|------------------------------------------------------------|
| Data from host | 0x00       | 0x00      | 0x00          | 0x00         | 0x00           | 0x00            | 0x00                                                       |
| Data to host   | Stat(15:8) | Stat(7:0) | pkt_len(15:8) | pkt_len(7:0) | rss_i_avg(8:1) | rss_i_sync(8:1) | sw_num(3:0)<br>rfu<br>rss_i_avg(0)<br>rfu<br>rss_i_sync(0) |

| Byte           | 7                      | 8                     | 9                    |
|----------------|------------------------|-----------------------|----------------------|
| Data from host | 0x00                   | 0x00                  | 0x00                 |
| Data to host   | freq_offset<br>(23:16) | freq_offset<br>(15:8) | freq_offset<br>(7:0) |

The *GetFlrcPacketStatus* command provides the following returned values:

- ♦ **pkt\_len**: Represents the length of the last received packet in bytes.
- ♦ **sw\_num**: Represents the detected Syncword of the latest received packet.
- ♦ **rss\_i\_avg**: Represents the average Received Signal Strength Indicator (RSSI) over the last received packet. The actual signal power can be calculated as  $-rss\_i\_avg/2$  [dBm]. This value gives an indication of the signal strength during the entire packet reception.
- ♦ **rss\_i\_sync**: Indicates the RSSI value latched after Syncword detection. The actual signal power is given as  $-rss\_i\_sync/2$  [dBm]. This value specifically reflects the signal strength at the moment when the Syncword is detected during the reception process.
- ♦ **freq\_offset**: Frequency error in Hz as a signed 24-bit integer.

The status information is updated at the completion of either a reception or a transmission operation.

### 18.4.5 SetFlrcSyncword

The *SetFlrcSyncword* configures the synchronisation word for an FLRC packet.

**Table 18-10: SetFlrcSyncword Command**

| Byte           | 0          | 1         | 2                | 3                   | 4                   | 5                  | 6                 |
|----------------|------------|-----------|------------------|---------------------|---------------------|--------------------|-------------------|
| Data from host | 0x02       | 0x4C      | sw_num<br>(7:0)  | Syncword<br>(31:24) | Syncword<br>(23:16) | Syncword<br>(15:8) | Syncword<br>(7:0) |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) | IrqStatus(23:16)    | IrqStatus(15:8)     | IrqStatus(7:0)     | 0x00              |

- ♦ **sw\_num** indicates the number of correlators to be used. Valid values are 1,2 and 3.
- ♦ **Syncword** defines the synchronization word used to detect the FLRC packet. Syncword can be 16 or 32 bits long. If only 16-bit Syncword is desired, then bytes 5 and 6 are optional.

## 19. OQPSK\_15\_4 (LR2021)

### 19.1 OQPSK\_15\_4 Modulation

The LR2021 implements the O-QPSK half-pulse shaped modem, which is compatible with 802.15.4 physical layer at up to 2 Mcps and a bit rate of 250 kbps. This modem is named OQPSK\_15\_4 in the rest of the document.

### 19.2 OQPSK\_15\_4 Packet Handler

The settings of the packet handler are defined via packet type definitions and commands described below.

### 19.3 OQPSK\_15\_4 Radio Commands

#### 19.3.1 SetOqpskParams

This command configures the LR2021 radio for transmission and reception of OQPSK\_15\_4 radio frames.

**Table 19-1: SetOqpskParams Command**

| Byte           | 0              | 1             | 2                    | 3                    | 4                   | 5                    | 6                   | 7                                                            |
|----------------|----------------|---------------|----------------------|----------------------|---------------------|----------------------|---------------------|--------------------------------------------------------------|
| Data from host | 0x02           | 0x9F          | Mode<br>(7:0)        | rx_bw<br>(7:0)       | pld_len<br>(7:0)    | pbl_len_tx<br>(15:8) | pbl_len_tx<br>(7:0) | rfu(4:0)<br>address_on<br>bypass_rx_length_check<br>fcs_mode |
| Data to host   | stat<br>(15:8) | stat<br>(7:0) | IrqStatus<br>(31:24) | IrqStatus<br>(23:16) | IrqStatus<br>(15:8) | IrqStatus<br>(7:0)   | 0x00                | 0x00                                                         |

**Mode** defines the modulation and bit rate to be used for the Rx and the Tx:

- ♦ 0x0: OQPSK pn32, chip rate 2 Mcps, bit rate 250 kbps
- ♦ others are rfu

**rx\_bw** defines the bandwidth in Rx mode in same way as FSK. 0xFF automatically selects the bandwidth depending on the selected mode.

**pld\_len** defines the payload length:

- ♦ In Rx, this is the maximum accepted MPDU size, i.e. what is decoded from the header
- ♦ In Tx, this is the total bytes to transmit (i.e. what is put in the FIFO)

Maximum value: 127. Returns CMD\_PERR in the status of the next command if higher.

**pbl\_len\_tx** sets the size, in bits, of the preamble to send. Typically, set to 32.

**address\_on** controls the address filtering feature (Address is set using the [SetOqpskAddress](#) command):

- ♦ 0: Disable address filtering
- ♦ 1: Enable address filtering

**bypass\_rx\_length\_check** requires a firmware patch, as described in [22.3 Firmware Patch RAM \(PRAM\)](#).

**fcs\_mode** sets the Rx/Tx mode for FCS. FCS is 16 bits.

- ♦ 0 (FCS\_ON): FCS is automatically generated in Tx. The FCS is checked in Rx but it is not sent to the FIFO. In Tx, the **pld\_len** is the MPDU length to transmit including the FCS.
- ♦ 1 (FCS\_IN\_FIFO): FCS is not automatically generated in Tx, the developer must generate it. The FCS is not checked in Rx and is sent to the Rx FIFO. In Tx, the **pld\_len** is the MPDU length to transmit.

### 19.3.2 GetOqpskRxStats

The *GetOqpskRxStats* command gets the internal statistics of the received packets.

**Table 19-2: GetOqpskRxStats Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x02       | 0xA0      |
| Data to host   | Stat(15:8) | Stat(7:0) |

**Table 19-3: GetOqpskRxStats Response**

| Byte           | 0          | 1         | 2            | 3           | 4              | 5             | 6              | 7             |
|----------------|------------|-----------|--------------|-------------|----------------|---------------|----------------|---------------|
| Data from host | 0x00       | 0x00      | 0x00         | 0x00        | 0x00           | 0x00          | 0x00           | 0x00          |
| Data to host   | Stat(15:8) | Stat(7:0) | pkt_rx(15:8) | pkt_rx(7:0) | CrcError(15:8) | CrcError(7:0) | LenError(15:8) | LenError(7:0) |

The returned values are:

- **pkt\_rx**: Total number of received packets
- **CrcError**: Number of received packets with a CRC error
- **LenError**: Number of packet with a length error

Statistics are reset on a POR, sleep without memory retention and the command [ResetRxStats](#).

### 19.3.3 GetOqpskPacketStatus

The *GetOqpskPacketStatus* command gets the status of the last received packet. Status is updated at the end of a reception (RxDone IRQ), but *rss\_i\_sync* is already updated on SyncwordValid IRQ.

**Table 19-4: GetOqpskPacketStatus Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x02       | 0xA1      |
| Data to host   | Stat(15:8) | Stat(7:0) |

**Table 19-5: GetOqpskPacketStatus Response**

| Byte           | 0              | 1             | 2                 | 3                 | 4                | 5                  | 6                   | 7                                                | 8            |
|----------------|----------------|---------------|-------------------|-------------------|------------------|--------------------|---------------------|--------------------------------------------------|--------------|
| Data from host | 0x00           | 0x00          | 0x00              | 0x00              | 0x00             | 0x00               | 0x00                | 0x00                                             | 0x00         |
| Data to host   | Stat<br>(15:8) | Stat<br>(7:0) | RxHeader<br>(7:0) | pkt_len<br>(15:8) | pkt_len<br>(7:0) | rss_i_avg<br>(8:1) | rss_i_sync<br>(8:1) | rfu(4:0)<br>rss_i_avg(0)<br>rfu<br>rss_i_sync(0) | lqi<br>(7:0) |

The returned values are:

- ♦ **RxHeader**: Content of the received PHY header (the frame length)
- ♦ **pkt\_len**: Length of the last received packet in bytes (including optional data added in the FIFO, crc, ...)
- ♦ **rss\_i\_avg**: RSSI average over the last packet received. Actual signal power is  $-rss\_i\_avg/2$  (dBm)
- ♦ **rss\_i\_sync**: Latched RSSI value after Syncword detection. Actual signal power is  $-rss\_i\_sync/2$  (dBm)
- ♦ **lqi**: Link quality indicator (0.25 dB)

### 19.3.4 SetOqpskPacketLen

This command sets the OQPSK\_15\_4 packet length without the need to call the full *SetOqpskParams* command (faster).

**Table 19-6: SetOqpskPacketLen Command**

| Byte           | 0          | 1         | 2                |
|----------------|------------|-----------|------------------|
| Data from host | 0x02       | 0xA2      | len(7:0)         |
| Data to host   | Stat(15:8) | Stat(7:0) | lqiStatus(31:24) |

*len* defines:

- ♦ In Rx, the maximum accepted MPDU size, i.e. what is decoded from the header.
- ♦ In Tx, the total bytes to transmit (i.e. what is put in the FIFO). Maximum value allowed for *len* is 127.

The chip returns CMD\_PERR in the status of the next command if *len* is higher than 127.

### 19.3.5 SetOqpskAddress

The *SetOqpskAddress* sets the OQPSK\_15\_4 addresses for filtering in Rx. If a received frames does not match the addresses, an AddrError IRQ is raised.

**Table 19-7: SetOqpskAddress Command**

| Byte           | 0          | 1         | 2                       | 3                       | 4                       | 5                       |
|----------------|------------|-----------|-------------------------|-------------------------|-------------------------|-------------------------|
| Data from host | 0x02       | 0xA3      | LongDestAddr<br>(63:56) | LongDestAddr<br>(55:48) | LongDestAddr<br>(47:40) | LongDestAddr<br>(39:32) |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24)        | IrqStatus(23:16)        | IrqStatus(15:8)         | IrqStatus(7:0)          |

| Byte           | 6                       | 7                       | 8                      | 9                     | 10                      | 11                     |
|----------------|-------------------------|-------------------------|------------------------|-----------------------|-------------------------|------------------------|
| Data from host | LongDestAddr<br>(31:24) | LongDestAddr<br>(23:16) | LongDestAddr<br>(15:8) | LongDestAddr<br>(7:0) | ShortDestAddr<br>(15:8) | ShortDestAddr<br>(7:0) |
| Data to host   | 0x00                    | 0x00                    | 0x00                   | 0x00                  | 0x00                    | 0x00                   |

| Byte           | 12              | 13             | 14               |
|----------------|-----------------|----------------|------------------|
| Data from host | PanId<br>(15:8) | PanId<br>(7:0) | TransId<br>(7:0) |
| Data to host   | 0x00            | 0x00           | 0x00             |

**LongDestAddr** sets the 64-bit physical device address.

**ShortDestAddr** sets the 16-bit network device address.

**PanId** sets the personal area network ID.

**TransId** sets the transaction ID.

When a packet is received, the destination address (**LongDestAddr** or **ShortDestAddr**) and PAN ID are both checked. If they don't match, an AddrError IRQ is raised and the reception of the packet is interrupted. The broadcast addresses are checked for a match as well.

**Note: Multi-cast is not supported or filtered.**

## 20. Z-Wave (LR2021)

### 20.1 Z-Wave Modulation

The LR2021 supports all modulations defined in the Z-Wave protocol standard, covering R1, R2, R3 and LR1 modes.

**WARNING: IN LR1 MODE, THE DEVELOPER MUST WAIT FOR THE RXDONE IRQ AND CALCULATE THE CRC MANUALLY.**

The table below illustrates the supported Z-Wave operating modes and their corresponding symbol rates.

**Table 20-1: Supported Z-Wave Modulation**

| Mode | Modulation Type     | Symbol Rate            | Modulation Index |
|------|---------------------|------------------------|------------------|
| R1   | 2FSK<br>Manchester  | 19.2 ksps<br>+/-27 ppm | 2.08             |
| R2   | 2FSK                | 40 ksps<br>+/-27 ppm   | 1                |
| R3   | 2GFSK<br>NRZ BT 0.6 | 100 ksps<br>+/-100 ppm | 0.58             |
| LR1  | O-QPSK              | 25 ksps<br>+/-27 ppm   | 0.5              |

### 20.2 Z-Wave Packet Handler

The chip supports the Z-Wave physical layer standard. The settings of the packet handler are defined via packet type definitions and the commands are described next.

## 20.3 Z-Wave Radio Commands

### 20.3.1 SetZwaveParams

The *SetZwaveParams* command defines the parameters for Z-Wave packets.

**Table 20-2: SetZwaveParams Command**

| Byte           | 0                 | 1                | 2                    | 3                | 4               | 5              |
|----------------|-------------------|------------------|----------------------|------------------|-----------------|----------------|
| Data from host | 0x02              | 0x97             | Mode (7:0)           | rx_bw (7:0)      | addr_comp (7:0) | pld_len (7:0)  |
| Data to host   | stat(15:8)        | stat(7:0)        | IrqStatus(31:24)     | IrqStatus(23:16) | IrqStatus(15:8) | IrqStatus(7:0) |
| Byte           | 6                 | 7                | 8                    | 9                |                 |                |
| Data from host | pbl_len_tx (15:8) | pbl_len_tx (7:0) | pbl_len_detect (7:0) | Options (7:0)    |                 |                |
| Data to host   | 0x00              | 0x00             | 0x00                 | 0x00             |                 |                |

**Mode** defines the Z-Wave mode of operation and as consequence, the bit rate used. See [Table 20-1](#).

- ♦ 0x0: ZWAVE\_MODE\_LR1
- ♦ 0x1: ZWAVE\_MODE\_R1
- ♦ 0x2: ZWAVE\_MODE\_R2
- ♦ 0x3: ZWAVE\_MODE\_R3

**rx\_bw** defines the bandwidth in Rx mode, as defined in [SetFskModulationParams](#)

- ♦ 0xFF automatically selects the bandwidth depending on the selected mode.

**addr\_comp** enables or disables the filtering of the HomeID

- ♦ 0x0: Received packets are not filtered
- ♦ 0x1: Received packets are filtered by HomeID
- ♦ 0x2: Received packets are filtered by HomeID, **and** beam frames are accepted as well. For beam frames, see [SetZwaveBeamFiltering](#).

**pld\_len** defines:

- ♦ In Rx, maximum accepted MPDU size, i.e. what is decoded from the header.
- ♦ In Tx, total bytes to transmit.

**pbl\_len\_tx** indicates the size in bits of the preamble to send, or 0 to automatically set the preamble size to the minimum possible for the selected mode.

**pbl\_len\_detect** indicates the preamble size, in bits, that is used for the detection. The recommended value is 32 in all modes, or 0xFF to automatically set the size to the optimal value.

**Options:** Determines how FCS is handled.

- ♦ bits [7:1]: RFU
- ♦ bit 0: fcs\_mode
  - ♦ 0: FCS is automatically generated in Tx. The FCS is checked in Rx but it is not sent to the FIFO. In Tx, the **pld\_len** is the MPDU length to transmit excluding the FCS.
  - ♦ 1: FCS is not automatically generated in Tx, the developer must generate it. FCS is not checked in Rx and is sent to the Rx FIFO. In Tx, the **pld\_len** is the MPDU length to transmit plus the FCS.



### 20.3.2 SetZwaveHomeldFiltering

The *SetZwaveHomeldFiltering* command sets the HomeID address to use as a filter in Rx. Frames that don't match the **home\_id** raise an AddrError IRQ.

**Table 20-3: SetZwaveHomeldFiltering Command**

| Byte           | 0          | 1         | 2                 | 3                | 4               | 5              |
|----------------|------------|-----------|-------------------|------------------|-----------------|----------------|
| Data from host | 0x02       | 0x98      | home_id (31:24)   | home_id (23:16)  | home_id (15:8)  | home_id (7:0)  |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus (31:24) | IrqStatus(23:16) | IrqStatus(15:8) | IrqStatus(7:0) |

### 20.3.3 GetZwaveRxStats

The *GetZwaveRxStats* command gets the internal statistics of the received packets.

**Table 20-4: GetZwaveRxStats Request Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x02       | 0x99      |
| Data to host   | Stat(15:8) | Stat(7:0) |

**Table 20-5: GetZwaveRxStats Response**

| Byte           | 0          | 1         | 2            | 3           | 4              | 5             | 6              | 7             |
|----------------|------------|-----------|--------------|-------------|----------------|---------------|----------------|---------------|
| Data from host | 0x00       | 0x00      | 0x00         | 0x00        | 0x00           | 0x00          | 0x00           | 0x00          |
| Data to host   | Stat(15:8) | Stat(7:0) | pkt_rx(15:8) | pkt_rx(7:0) | CrcError(15:8) | CrcError(7:0) | LenError(15:8) | LenError(7:0) |

The command returns the following parameters:

- ♦ **pkt\_rx**: Total number of received packets
- ♦ **CrcError**: Number of received packets with a CRC error
- ♦ **LenError**: Number of packet with a length error

Statistics are reset on a POR, sleep without memory retention, or the command *ResetRxStats*.

### 20.3.4 GetZwavePacketStatus

*GetZwavePacketStatus* command gets the status of the last received packet. Status is updated at the end of a reception (RxDone IRQ), but *rss\_i\_sync* is already updated on SyncwordValid IRQ.

**Table 20-6: GetZwavePacketStatus Request Command**

| Byte           | 0          | 1         |
|----------------|------------|-----------|
| Data from host | 0x02       | 0x9A      |
| Data to host   | Stat(15:8) | Stat(7:0) |

**Table 20-7: GetZwavePacketStatus Response**

| Byte           | 0              | 1             | 2                 | 3                | 4                  | 5                   | 6                     | 7                                                | 8            |
|----------------|----------------|---------------|-------------------|------------------|--------------------|---------------------|-----------------------|--------------------------------------------------|--------------|
| Data from host | 0x00           | 0x00          | 0x00              | 0x00             | 0x00               | 0x00                | 0x00                  | 0x00                                             | 0x00         |
| Data to host   | Stat<br>(15:8) | Stat<br>(7:0) | pkt_len<br>(15:8) | pkt_len<br>(7:0) | rss_i_avg<br>(8:1) | rss_i_sync<br>(8:1) | LastDetected<br>(7:0) | rfu(4:0)<br>rss_i_avg(0)<br>rfu<br>rss_i_sync(0) | Lqi<br>(7:0) |

The command returns the following parameters

- ♦ **pkt\_len**: Length of the last received packet in bytes (including optional data added in the FIFO, CRC,...)
- ♦ **rss\_i\_avg**: Average over last packet received of RSSI. Actual signal power is  $-rss\_i\_avg/2$  (dBm)
- ♦ **rss\_i\_sync**: Latch RSSI value after Syncword detection. Actual signal power is  $-rss\_i\_sync/2$  (dBm)
- ♦ **LastDetected**: Data rate of the latest Rx packet. This requires a firmware patch, as described in [22.3 Firmware Patch RAM \(PRAM\)](#).
  - ♦ 0x1: R1 mode
  - ♦ 0x2: R2 mode
  - ♦ 0x3: R3 mode
  - ♦ 0x4: LR1 mode
  - ♦ Other: RFU
- ♦ **Lqi**: Link quality indicator (in 0.25 dB steps)

## 20.3.5 SetZwaveBeamFiltering

The *SetZwaveBeamFiltering* command defines the filtering criteria for incoming beam frames in Rx.

**Table 20-8: SetZwaveBeamFiltering Command**

| Byte           | 0          | 1         | 2                | 3                                      | 4               | 5              |
|----------------|------------|-----------|------------------|----------------------------------------|-----------------|----------------|
| Data from host | 0x02       | 0x9B      | beam_tag(7:0)    | addr_len<br>rfu (2:0)<br>node_id(11:8) | node_id(7:0)    | id_hash(7:0)   |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) | IrqStatus(23:16)                       | IrqStatus(15:8) | IrqStatus(7:0) |

**beam\_tag** indicates the beam tag value to accept, typically 0x55

**addr\_len** defines the address length to be considered:

- ♦ 0x0: 8-bit address
- ♦ 0x1: 12-bit address

**node\_id** indicates the node id value to accept. This is an 8-bit value, or 12-bit for the LR1 bit rate. A value of 0xFF is always accepted.

**id\_hash** indicates the Home ID Hash to accept. A value of 0x55, 0x0A or 0x4A is always accepted.

## 20.3.6 SetZwaveScanConfig

The *SetZwaveScanConfig* command defines the bit rates and channels to be used during scan mode.

**Table 20-9: SetZwaveScanConfig Command**

| Byte               | 0                  | 1                  | 2                 | 3                | 4                  | 5                |
|--------------------|--------------------|--------------------|-------------------|------------------|--------------------|------------------|
| Data from host     | 0x02               | 0x9C               | num_ch(3:0)       | bitrate_ch4(1:0) | addr_comp<br>(7:0) | options<br>(7:0) |
|                    |                    |                    | det4              | bitrate_ch3(1:0) |                    |                  |
|                    |                    |                    | det3              | bitrate_ch2(1:0) |                    |                  |
|                    |                    |                    | det2              | bitrate_ch1(1:0) |                    |                  |
|                    |                    |                    | det1              |                  |                    |                  |
| Data to host       | Stat(15:8)         | Stat(7:0)          | IrqStatus(31:24)  | IrqStatus(23:16) | IrqStatus(15:8)    | IrqStatus(7:0)   |
| 6                  | 7                  | 8                  | 9                 | 10               | ...                |                  |
| rf_freq_ch1(31:24) | rf_freq_ch1(23:16) | rf_freq_ch1(15:8)  | rf_freq_ch1(7:0)  | timeout_ch1(7:0) | ...                |                  |
| 0x00               | 0x00               | 0x00               | 0x00              | 0x00             | ...                |                  |
| ...                | 21                 | 22                 | 23                | 24               | 25                 |                  |
| ...                | rf_freq_ch4(31:24) | rf_freq_ch4(23:16) | rf_freq_ch4(15:8) | rf_freq_ch4(7:0) | timeout_ch4(7:0)   |                  |
| ...                | 0x00               | 0x00               | 0x00              | 0x00             | 0x00               |                  |

Configures the Z-Wave scan mode. The developer can configure the 4 bit rates and channels to be scanned.

**num\_ch**: The number of channels to scan in a range of 2 to 4.

**detx**: CCA assessment:

- 0: No CCA assessment (recommended for all channels)
- 1: CCA assessment. If no CCA detected, goes directly to the next channel without trying to detect until the timeout

**bitrate\_chX**: The bit rate of the Xth channel to be scanned. 0 = LR1, 1 = R1, 2 = R2, 3 = R3, other values are reserved.

**addr\_comp**: The address comparator mode, same format as for the *SetZwaveParams* command.

**options**: The same format as for the *SetZwaveParams* command.

**rf\_freq\_chX**: The RF frequency of the Xth channel. Same format as for the *SetRfFrequency* command.

**timeout\_chX**: The scan timeout for the Xth channel, 1LSB = 30 microseconds

When two bit rates are scanned, the recommended values for the scan timeouts are: LR1-> 11(330us), LR1-> 11(330us)

When three bit rates are scanned, the recommended values for the scan timeouts are: R1->19 (570us), R2->34 (1020us), R3->8(240us)

When four bit rates are scanned, the recommended values for the scan timeouts are: R1->19 (570us), R2->30 (900us), R3->7(210us), LR1->12 (360us)

The **rf\_freq\_chX** and **timeoutChX** parameters only need to be sent for num\_ch channels.

### 20.3.7 SetZwaveScan

The *SetZwaveScan* command enters the Z-Wave in Rx scan mode. The scan mode alternates between the specified number of channels in *SetZwaveScanConfig* and attempts to detect an incoming packet.

**Table 20-10: SetZwaveScan Command**

| Byte           | 0          | 1         | 2                |
|----------------|------------|-----------|------------------|
| Data from host | 0x02       | 0x9D      | mode             |
| Data to host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) |

**mode:** optional 1 byte parameter:

- ♦ 0: Default, computes an internal scan configuration based on SetZwaveScanConfig and starts scanning.
- ♦ 1: Only compute the internal configuration.
- ♦ 2: Start scan without configuration computation. Configuration must have been computed before (mode=0 or 1).

Pre-computation allows faster TX->RX switching.

**Note:** Internal configuration must be re-computed on any RF-FE configuration change (RX path, AGC config, RF-FE calibration).

## 21. Test Commands (LR20xx)

The LR2021/LR2022/LR2012 test command allows an easy configuration of the device for regulatory ETSI or FCC compliance.

### 21.1 Regulatory Overview

This section only describes the RF modes necessary for ETSI and FCC regulatory testing. Refer to the ETSI and FCC documents for a detailed test description and for the test limits indication.

#### 21.1.1 ETSI

The EN 300 220 standards describe 4 test signals which the EUT (Equipment Under Test) must be able to transmit for CE certification. These test signals are listed in the table hereafter, with the operating mode correspondence for the LR2021/LR2022/LR2012.

**Table 21-1: ETSI Test Signals**

| Test Signal | Description                                                           | LR2021/LR2022/LR2012 Operation                                     |
|-------------|-----------------------------------------------------------------------|--------------------------------------------------------------------|
| D-M1        | Unmodulated carrier                                                   | Tx CW mode ( <i>SetTxCW(...)</i> command)                          |
| D-M2        | Continuously modulated signal with the greatest occupied RF bandwidth | Continuous modulation ( <i>SetTxInfinitePreamble(...)</i> command) |
| D-M2a       | Same as D-M2 signal, but not continuous                               | RF packet transmission: LoRa SF12, BW500, 50% duty cycle           |
| D-M3        | Normal operating mode of the EUT in the application                   | Operation as in the application                                    |

The developer should be able to modify the operating frequency, output power, and modulation parameters for the ETSI tests. The developer should also be able to receive the incoming RF packets for any configuration (frequency, modulation parameters), and to determine a PER (Packet Error Rate) indication of the receive quality.

All this can be done using the regular LR2021/LR2022/LR2012 radio commands.

#### 21.1.2 FCC

The FCC part 15.247 is applicable to frequency hopping and digitally modulated systems. For those tests, only an unmodulated carrier (Tx CW) and a regular packet transmission are required.

The developer should be able to modify the operating frequency, output power, and modulation parameters for the FCC tests. This can be done using the regular LR2021/LR2022/LR2012 radio commands.

## 21.2 SetTxTestMode

Command *SetTxTestMode* sets the device in Tx test mode.

**Table 21-2: SetTxTestMode Command**

| Byte           | 0          | 1         | 2                |
|----------------|------------|-----------|------------------|
| Data from Host | 0x02       | 0x0E      | mode(7:0)        |
| Data to Host   | Stat(15:8) | Stat(7:0) | IrqStatus(31:24) |

Sets the device into Tx test **mode**, according to:

- ♦ 0x00: Normal Tx (same as a SetTx(0))
- ♦ 0x01: Infinite preamble (not available in LR-FHSS)
- ♦ 0x02: Continuous wave (not available in LR-FHSS)
- ♦ 0x03: PRBS9, Pseudo Random Bit Sequence (not available in LoRa/LR-FHSS)

## 22. Known Limitations and Workarounds

At the time of authoring this document, some limitations are known.

Implementations of workarounds are described inside the *readme* of the LR2021/LR2022/LR2012 drivers hosted on the Semtech Github page available at <https://github.com/lora-net>.

Some workarounds are automatically applied by the drivers. Review the guidelines provided individually for each workaround.

### 22.1 OOK Detection Threshold Adjustment (LR20xx)

#### 22.1.1 Description

The LR20xx uses a fixed RSSI threshold, calculated automatically by the IC as a function of the Receiver Bandwidth. However, this fixed threshold calculation can be limiting:

- Threshold set too high: The receiver becomes overly cautious about noise and may miss weak valid signals, resulting in weak sensitivity
- Threshold set too low: In presence of noise and interference in the channel, the threshold may be set overly low, resulting in false detections and missed packets (PER)

#### 22.1.2 Workaround

The OOK Detection Threshold Override workaround allows the developer to manually set a more optimal threshold instead of relying on the chip's automatic calculation. The optimal threshold should typically be set to the current noise floor+ small margin. To measure this:

1. Configure the radio with your intended OOK modulation parameters.
2. Use `GetRssiInst` to measure the noise level with the same modulation settings.
3. Compare this measured noise level with the default computed threshold using `lr20xx_workarounds_ook_get_default_detection_threshold_level()`.
4. If the noise level is higher than the default threshold, use the measured level as your new threshold.

Implementation Steps:

1. Configure your OOK modulation using `SetOokModulationParams`.
2. Apply the threshold override using `lr20xx_workarounds_ook_set_detection_threshold_level()` with your optimized threshold value (in dB).

### 22.2 RTTOF Accuracy for SX1280-compatible Bandwidths (LR2021/LR2022)

For LoRa bandwidths 203, 406 and 812 kHz (at either 2.4 GHz or sub-GHz), the RTToF accuracy is optimized when calling specific workarounds described in `lr20xx_workarounds.c`. Refer to the LR20xx driver for reference.



## 22.3 Firmware Patch RAM (PRAM)

The LR202x transceiver family is provided with a program (PRAM) that can be loaded into the chip to optimize performance and provide a workaround for some limitations.

While not strictly required, using the chip without the PRAM can create performance issues and unexpected bugs. The use of the PRAM is therefore highly recommended.

Note that:

- The PRAM is lost after a reset or a cold start.
- The PRAM is preserved during a sleep with retention.
- The sleep current with retention will increase by 80 nA when the PRAM is loaded (as reflected in IDDSL1 parameter).
- The PRAM also modifies the following:
  - ♦ Z-Wave: addition of last received packet channel in GetZwavePacketStatus command.
  - ♦ OQPSK: addition of configuration flag to bypass the received packet length check.

### 22.3.1 Loading the Patch RAM

The PRAM must be loaded in two situations:

- After a reset, as part of the reset sequence.
- After waking up from a deep sleep ("cold sleep").

Follow the following procedure after a reset (or a wake-up from cold sleep):

1. Obtain the relevant PRAM. There is one dedicated to the LR2021 and another one for the LR2012/LR2022. The GetVersion(...) API indicates which PRAM is needed.
2. Load the PRAM into memory: The host must write the entire contents of the pram array to the chip, using the command WriteRegMem32(...), starting from address 0x801000.
3. Activate the PRAM: The host must send the command with the opcode 0x012D with a 1 bytes argument set to 0x00. The MOSI output bytes should look like this: [0x01, 0x2D, 0x00].

At this point, the chip is ready to be used.

### 22.3.2 Checking the Patch RAM

To check that the PRAM is loaded:

- Read register value at address 0x800FF8, using the ReadRegMem32(...) command. The returned value should be 0x600DB002.

To read the version of the PRAM:

- Read register value at address 0x800FFC, using the ReadRegMem32(...) command. The version is then ((returned\_value >> 8) & 0xFFFF).

## 22.4 Regulatory Compliance

### 22.4.1 CN470 SRRC Compliance for China

The following modifications are necessary to ensure LR20xx full compliance to the CN470 SRRC regulations while using FLRC modulation:

- Use Gaussian Filter BT=0.3
- Modify the PLL bandwidth writing 0x05 at register address 0xF40110: WriteRegMemMask32(addr =0xF40110, Mask = 0x00000007, Value = 0x05)
- Modify the modulator sample rate conversion at register address 0xF30904: WriteRegMemMask32(addr =0xF30904, Mask = 0x00000100, Value = (1<< 8)).

### 22.4.2 IN865 Compliance for India

The following modifications are necessary to ensure LR20xx full compliance to the IN865 Indian regulations while using FLRC modulation:

- Modify the modulator sample rate conversion at register address 0xF30904: WriteRegMemMask32(addr =0xF30904, Mask = 0x00000100, Value = (1<< 8)).

## 23. Application Information

The LR2021/LR2022/LR2012's multi-band, multi-protocol capabilities require careful consideration of thermal management, power optimization, and RF design to achieve optimal performance. This section provides design guidelines and application-specific recommendations for implementing the LR2021/LR2022/LR2012 transceiver in various wireless communication systems.

### 23.1 Power Management

The LR2021/LR2022/LR2012 offers flexible power supply configurations to optimize efficiency for different applications, providing the user with the possibility to use either the integrated SIMO DC-DC converter or the LDO.

The DC-DC configuration is strongly recommended for battery-operated applications as it reduces power consumption by up to 50% compared to LDO mode, as its efficiency exceeds 85%. This also maintains a constant heat dissipation across the 1.8 V to 3.7 V supply voltage range, limiting thermal drift. PA\_LF operation at +22 dBm requires a supply voltage greater than 2.2 V when running on the SIMO.

In contrast, the LDO configuration provides a simpler implementation with fewer external components, at the expense of a higher current consumption.

### 23.2 RF Shield Impact

FCC Part 15.247/249 compliance at +22 dBm requires a RF metallic shield. However, when using the SIMO with a shield, up to 2 dB of sensitivity degradation can be observed:

**Table 23-1: Sensitivity for SIMO with Metallic Shielding**

| Band        | Modulation/Protocol | Expected degradation using SIMO AND with metallic RF shield |
|-------------|---------------------|-------------------------------------------------------------|
| 2.4 GHz     | All                 | None                                                        |
| All Sub-GHz | FLRC > 1040 kbps    | Up to 1.5 dB                                                |
|             | FLRC <= 1040 kbps   | Up to 0.5 dB                                                |
|             | LoRa BW >= 800 kHz  | Up to 1.5 dB                                                |
|             | LoRa <= 500 kHz     | Up to 1 dB                                                  |
|             | (G)FSK <= 100 kbps  | Up to 0.5 dB                                                |
|             | (G)FSK > 100 kbps   | Up to 1 dB                                                  |
|             | O-QPSK 802.15.4     | None                                                        |
|             | Wi-SUN FSK          | None                                                        |

Layout recommendations to limit the impact of a metallic shielding in SIMO mode are provided in App Note AN1200.66 PCB Design Guidelines: <https://www.semtech.com/products/wireless-rf/lora-plus/lr2021>.

Semtech recommends that you closely follow the Semtech reference design on LoRa Plus™ product page (available for download at <https://www.semtech.com/products/wireless-rf/lora-plus/lr2021>), and submit your design to Semtech for review.

## 23.3 Oscillator Recommendations

A standard 32 MHz crystal oscillator is suitable for most non-narrowband LPWAN applications and offers a lower-cost solution. It however requires proper thermal management in PCB design.

The crystal oscillation frequency is influenced by the total load capacitance present on its pins, and therefore might be affected by parasitics on the PCB in case of poor PCB design. Built-in crystal loading capacitors allow compensation for the crystal frequency shift. Refer to AN1200.106 LR20xx Series Crystal Temperature Mitigation for additional information.

A TCXO eliminates frequency drift during transmission and becomes essential for extreme temperature operation ( $<-20\text{ }^{\circ}\text{C}$  or  $>+70\text{ }^{\circ}\text{C}$ ) or applications that require continuous transmission or very long packet durations. This however incurs 1–2 mA additional current consumption.

The selection between crystal and TCXO should be based on the application's targeted power consumption budget, thermal environment, packet duration requirements, and whether frequency stability must be guaranteed under all operating conditions. Semtech provides a list of recommended references in AN1200.59 Reference Clock Selection.

## 23.4 Reference Design

Semtech provides the following reference designs on a standard 4-layer FR4 PCB. Follow the reference design as precisely as possible to match the performance stated in the datasheet. Follow the recommendations in AN1200.66 PCB Design Guidelines for an optimal PCB design. The LR2012 is the same as LR2021/LR2022 except pins 31 and 32 (the HF path) are not connected.

**Figure 23-1: LR2021/LR2022 Reference Design Schematic**

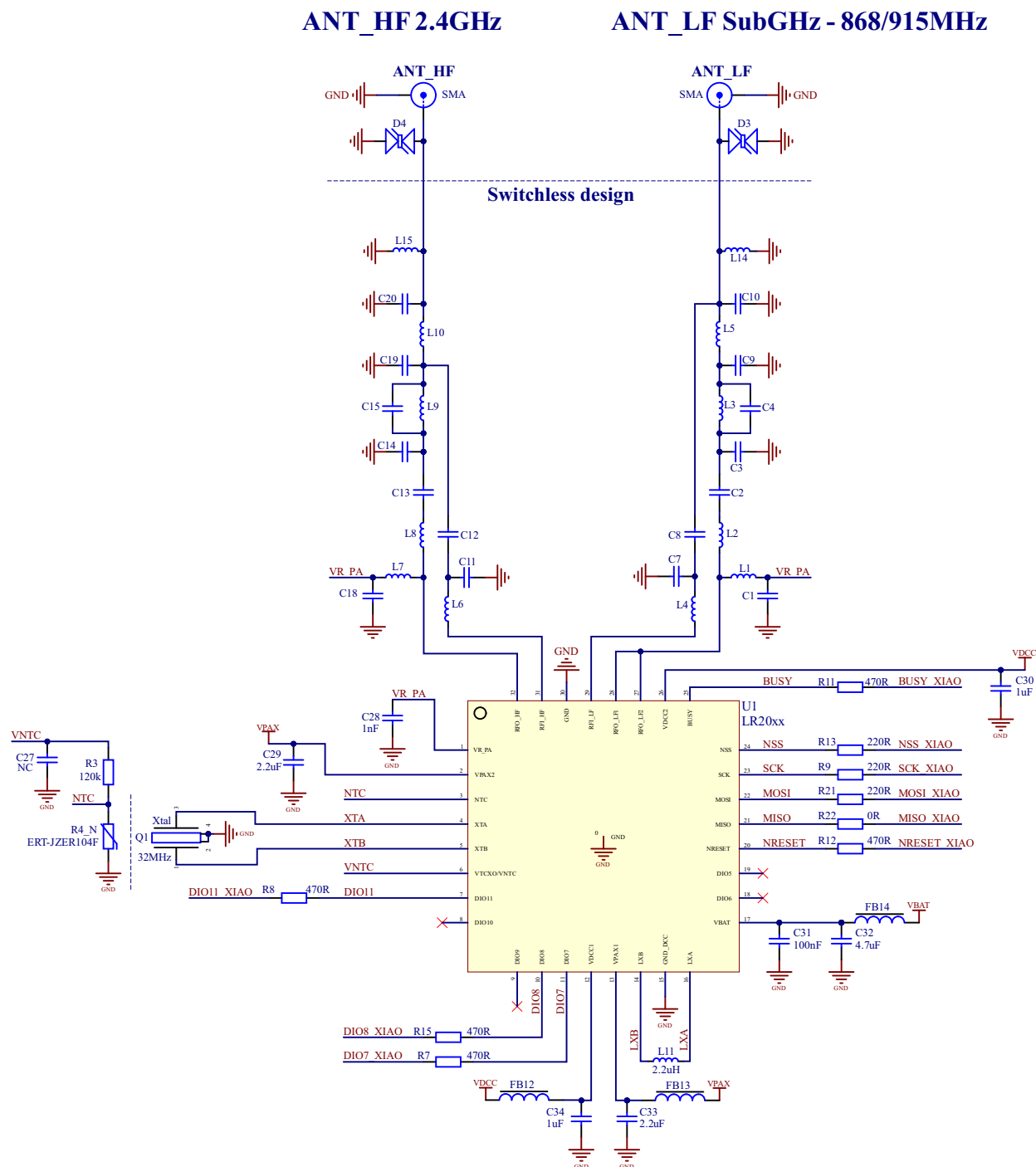




Figure 23-3: LR2021/LR2022 Reference Design PCB Layout

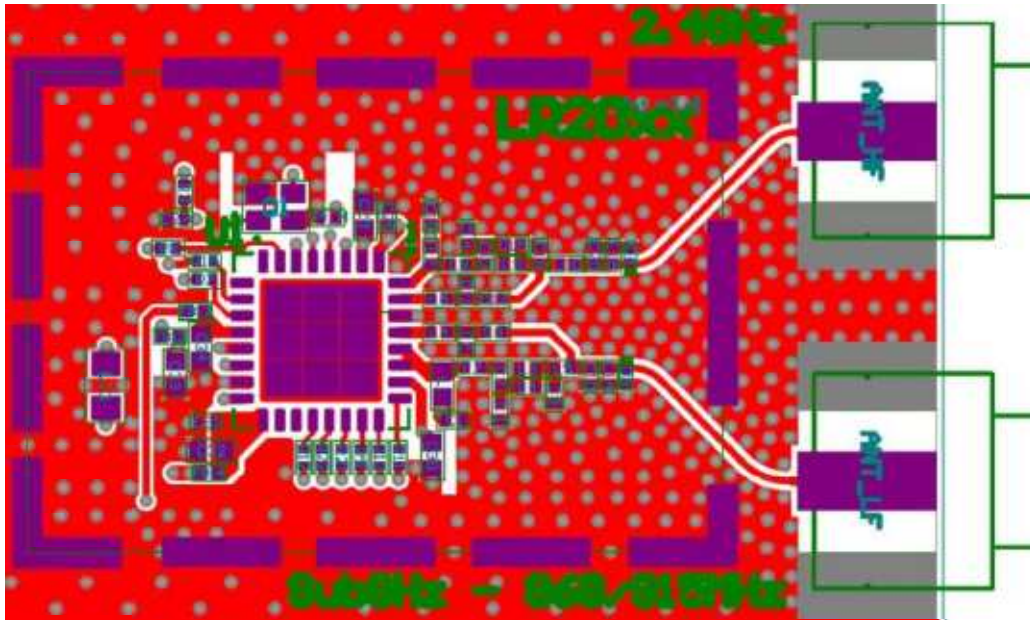
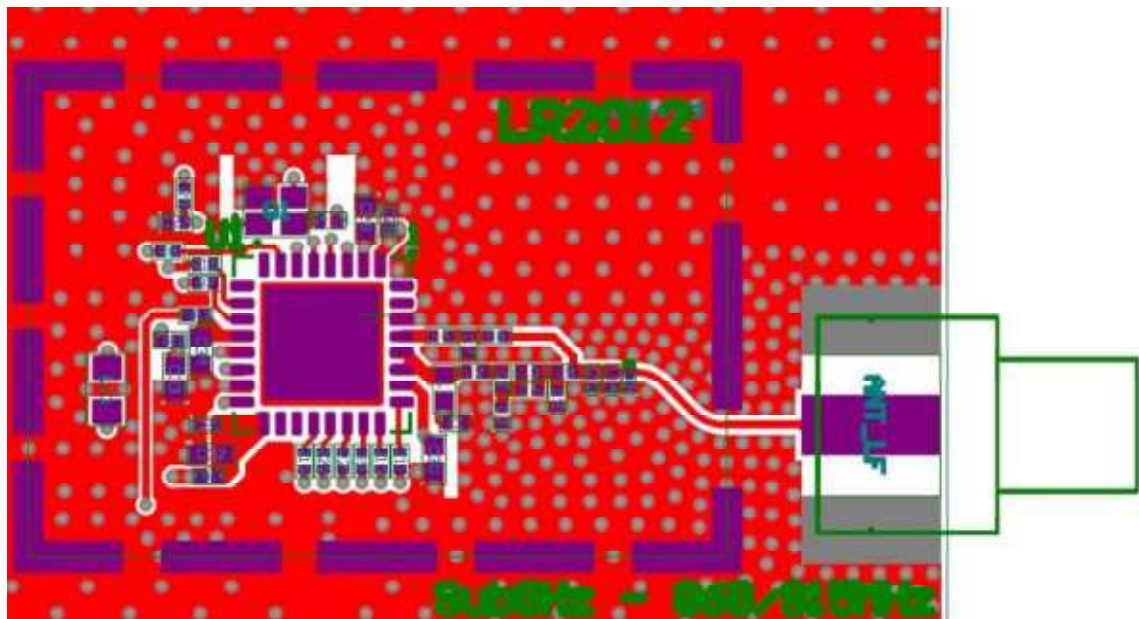


Figure 23-4: LR2012 Reference Design PCB Layout



## 23.5 Unused Pins

Several pins may not be required by some applications. This section indicates how to connect those pins:

- Unused RF pins: Leave unconnected
- Unused DIOs: Leave unconnected
- XTB pin (when a TCXO is used as 32 MHz oscillator): Leave unconnected
- LXA/LXB when LDO regulation is used: Leave unconnected. VPAX1/VPAX2 and VDCC1/VDCC2 are still to be connected together in LDO mode
- NTC pin if no NTC Resistor: Leave unconnected
- Pins 31 and 32 are not connected for the LR2012

## 23.6 Configurations via Register Change

Some configurations are allowed via direct register write. The table below summarizes the registers and their usage:

**Table 23-2: Register Usage**

| Address    | Description                                   |
|------------|-----------------------------------------------|
| 0xF30A14   | LoRa SX1276 SF6 Implicit Mode compatibility   |
| 0xF30A24   | SX1276 - compatible LoRa intra-packet hopping |
| 0xF30A2C   | LoRa extended frequency error control         |
| 0x00F40338 | OCP Control unlock                            |
| 0x00F40300 | OCP LF PA and HF PA thresholds                |
| 0x00F30034 | Tx FIFO size                                  |
| 0x00F30030 | Rx FIFO size                                  |
| 0x00F3002C | Tx FIFO base address                          |
| 0x00F30028 | Rx FIFO base address                          |
| 0x00F30904 | Modulator sample rate conversion              |
| 0x00F40110 | PLL bandwidth                                 |
| 0x00F30D18 | FLRC correlators                              |

Command [SetAdditionalRegToRetain](#) allows the user store up to 32 registers in the RAM, in particular, for the workarounds indicated in [Section 22](#).

When entering Sleep mode with retention, the register configuration is lost (only RAM is retained).

All RAM is lost in Sleep mode without retention.



## 23.7 OCP registers

The over-current protection (OCP) registers are protected and require unlocking before configuration by writing 0xCODE to address 0x00F40338 [15:0].

Once unlocked, the PA\_LF and PA\_HF OCP are accessible as follows, each providing 64 programmable levels:

- PA\_LF OCP threshold via register bits [24:19] at address 0x00F40300
- PA\_HF OCP threshold via bits [30:25] of the same register

The OCP thresholds must be set below the battery's current limit while maintaining sufficient margin ( $\geq 25\%$ ) above the nominal PA current to prevent false triggering during normal operation.

## 23.8 Compatibility of Generation 1–4 LoRa Radios

Semtech has developed four generations of LoRa radios:

- Generation 1 (Gen 1): SX1272, SX1276 - First LoRa chips (2013)
- Generation 2 (Gen 2): SX1261, SX1262, SX1280 - Enhanced performance and efficiency
- Generation 3 (Gen 3): LR1110, LR1120, LR1121 - LoRa Edge™ - Added geolocation
- Generation 4 (Gen 4): LR2021 LoRa Plus™ Multi-band, and multi-protocol

LoRa radios across the different generations can communicate with each other when properly configured with identical parameters (frequency, spreading factor, bandwidth, and coding rate).

However, certain features have evolved across generations, requiring specific compatibility modes or workarounds for interoperability:

- SF5 is available starting from generation 2
- SF6 requires additional configuration for backwards compatibility
- Long interleaver is available and compatible in generations 3 and 4
- Intra-packet hopping is available in generations 1 and 4
- Multi-SF CAD and Fast CAD are additions in generation 4
- Convolutional CR8/CR9 are new in generation 4

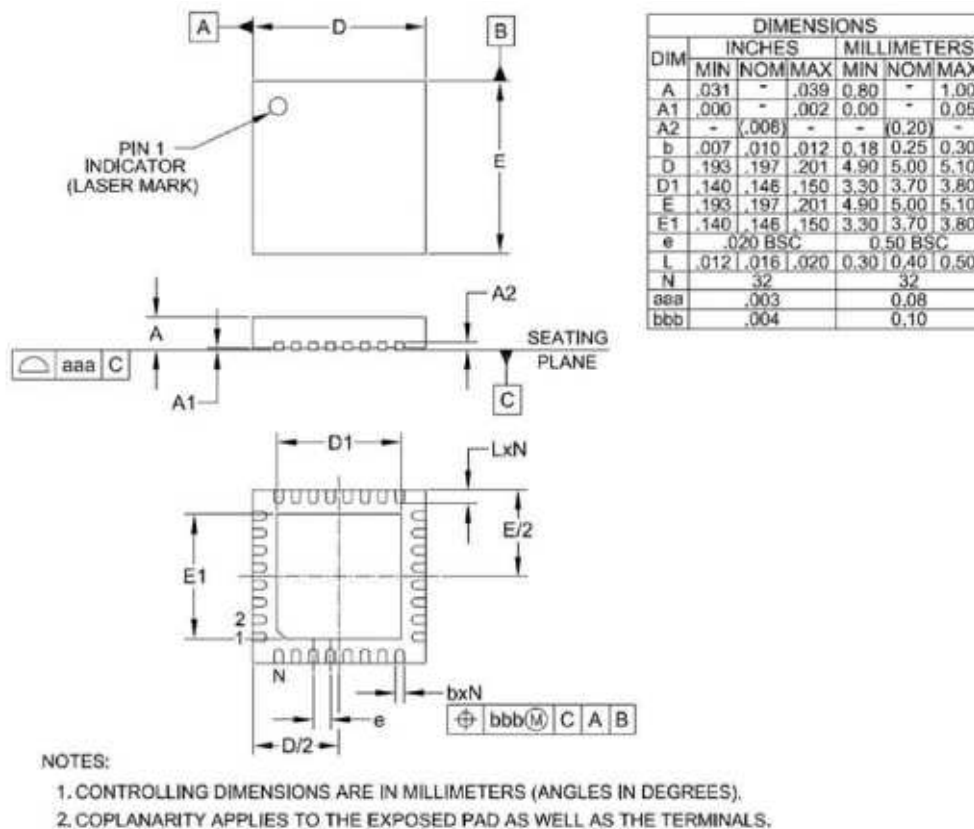
All the LoRaWAN configurations are fully compatible between all four generations of LoRa radios.

Refer to AN1200.102 LR2021 LoRa Performance for additional information.

## 24. Package Information

### 24.1 Package Outline Drawing

Figure 24-1: Package Outline Drawing



## 24.2 Package Marking

Figure 24-2: Package Marking

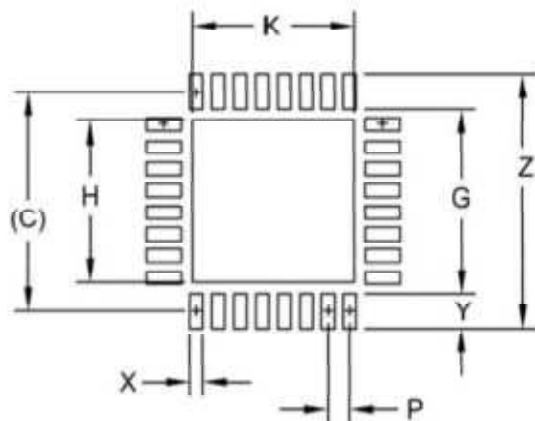


| TOP MARK |      |
|----------|------|
| CHAR     | ROWS |
| 6/6/6/6  | 4    |

Marking for the 5 x 5mm MLPQ 32 Lead package:  
 nnnnnn = Part Number (Example: LR2021)  
 yyww = Date Code (2052)  
 xxxxxx = Semtech Lot No. (Example: E90100)

## 24.3 Land Pattern

Figure 24-3: Land Pattern



| DIMENSIONS |        |             |
|------------|--------|-------------|
| DIM        | INCHES | MILLIMETERS |
| C          | (.197) | (5.00)      |
| G          | .165   | 4.20        |
| H          | .146   | 3.70        |
| K          | .146   | 3.70        |
| P          | .020   | 0.50        |
| X          | .012   | 0.30        |
| Y          | .031   | 0.80        |
| Z          | .228   | 5.80        |

**NOTES:**

1. THIS LAND PATTERN IS FOR REFERENCE PURPOSES ONLY. CONSULT YOUR MANUFACTURING GROUP TO ENSURE YOUR COMPANY'S MANUFACTURING GUIDELINES ARE MET.
2. THERMAL VIAS IN THE LAND PATTERN OF THE EXPOSED PAD SHALL BE CONNECTED TO A SYSTEM GROUND PLANE. FAILURE TO DO SO MAY COMPROMISE THE THERMAL AND/OR FUNCTIONAL PERFORMANCE OF THE DEVICE.
3. SQUARE PACKAGE - DIMENSIONS APPLY IN BOTH "X" AND "Y" DIRECTIONS.

## 24.4 Reflow Profiles

Reflow process instructions are available from the Semtech website, at the following address:  
[http://www.semtech.com/quality/ir\\_reflow\\_profiles.html](http://www.semtech.com/quality/ir_reflow_profiles.html)

The device uses a QFN32 5x5mm package, also named MLP package.

## 24.5 Thermal Information

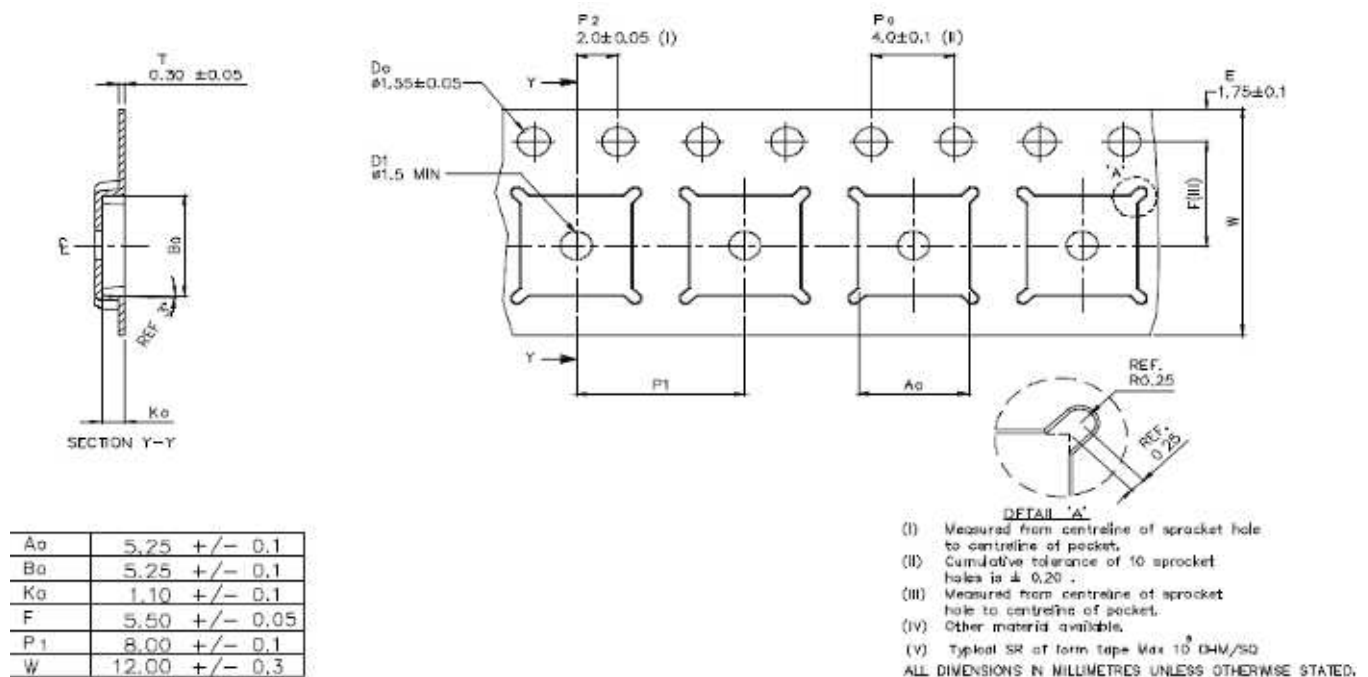
Table 24-1: Package Thermal Information

| Name                           | Value <sup>1</sup> | Unit |
|--------------------------------|--------------------|------|
| Theta j-a, Junction to Ambient | 26.7               | °C/W |

1. Measured on a 4-layer test board with 9 thermal vias, per Jedec standard JESD51-5.

## 24.6 Tape & Reel Information

Figure 24-4: Tape & Reel Information



## Glossary

### List of Acronyms and their Meaning (Sheet 1 of 4)

| Acronym   | Meaning                                                                                         |
|-----------|-------------------------------------------------------------------------------------------------|
| ADC       | Analog-to-Digital Converter                                                                     |
| ADS-B     | Automatic Dependent Surveillance–Broadcast                                                      |
| AGC       | Automatic Gain Control                                                                          |
| API       | Application Programming Interface                                                               |
| $\beta$   | Modulation Index                                                                                |
| BPSK      | Binary Phase-Shift Keying                                                                       |
| BR        | Bit Rate                                                                                        |
| BRN       | Brown-Out Reset                                                                                 |
| BT        | Bandwidth-Time bit period product                                                               |
| BW        | Bandwidth                                                                                       |
| BWF       | Bandwidth of the (G)FSK modem                                                                   |
| BWLORA    | Bandwidth of the LoRa® Modem                                                                    |
| CAD       | Channel Activity Detection                                                                      |
| CCA       | Clear Channel Assessment                                                                        |
| CPFSK     | Continuous Phase Frequency Shift Keying                                                         |
| CPHA      | Clock Phase                                                                                     |
| CR        | Coding Rate                                                                                     |
| CRC       | Cyclical Redundancy Check                                                                       |
| CSS       | Chirp Spread Spectrum PHY layer protocol                                                        |
| CW        | Continuous Wave                                                                                 |
| DCR       | Direct Current Resistance                                                                       |
| DIO       | Digital Input / Output                                                                          |
| DS        | Distribution System                                                                             |
| DSB       | Double Side Band                                                                                |
| DSP       | Digital Signal Processing                                                                       |
| DSSS      | Direct-Sequence Spread Spectrum                                                                 |
| ETSI - EN | European Telecommunications Standards Institute - European Standards (German: Europäische Norm) |
| FCC       | Federal Communications Commission                                                               |

## List of Acronyms and their Meaning (Sheet 2 of 4)

| Acronym | Meaning                                                                                               |
|---------|-------------------------------------------------------------------------------------------------------|
| FCS     | Frame Check Sequence                                                                                  |
| FREQDEV | Frequency Deviation                                                                                   |
| FIFO    | First In First Out                                                                                    |
| FLRC    | Fast Long Range Communication                                                                         |
| FS      | Frequency Synthesis                                                                                   |
| FSK     | Frequency Shift Keying                                                                                |
| GMSK    | Gaussian Minimum-Shift Keying                                                                         |
| IF      | Intermediate Frequencies                                                                              |
| IRQ     | Interrupt Request                                                                                     |
| ISI     | Inter-Symbol Interference                                                                             |
| ISM     | Industrial, Scientific, and Medical                                                                   |
| LDO     | Low-Dropout                                                                                           |
| LDRO    | Low Data Rate Optimization                                                                            |
| LE      | Low Energy                                                                                            |
| LF      | Low Frequency                                                                                         |
| LFSR    | Linear-Feedback Shift Register                                                                        |
| LNA     | Low-Noise Amplifier                                                                                   |
| LoRa®   | Long Range Communication<br>the LoRa® Mark is a registered trademark of the Semtech Corporation       |
| LQI     | Link Quality Indication. LQI is the ratio in dB between detection correlation peak and average power. |
| LR-FHSS | Long Range Frequency Hopping Spread Spectrum                                                          |
| LSB     | Least Significant Bit                                                                                 |
| HF      | High Frequency                                                                                        |
| M-BUS   | Meter Bus European standard (EN 13757-2 physical and link layer, EN 13757-3 application layer)        |
| MIC     | Message Integrity Code                                                                                |
| MISO    | Manager Input Subordinate Output                                                                      |
| MOSI    | Manager Output Subordinate Input                                                                      |
| MOQ     | Minimum Order Quantity                                                                                |
| MSB     | Most Significant Bit                                                                                  |
| MSK     | Minimum-Shift Keying                                                                                  |

## List of Acronyms and their Meaning (Sheet 3 of 4)

| Acronym | Meaning                                              |
|---------|------------------------------------------------------|
| NF      | Noise Figure                                         |
| NOP     | No Operation (0x00)                                  |
| NRZ     | Non-Return-to-Zero                                   |
| NSS     | Subordinate Select active low                        |
| OCP     | Over Current Protection                              |
| OOK     | On-Off Keying                                        |
| O-QPSK  | Offset Quadrature Phase Shift Keying                 |
| PA      | Power Amplifier                                      |
| PAN ID  | Personal Area Network ID                             |
| PER     | Packet Error Rate                                    |
| PHY     | Physical Layer                                       |
| PID     | Product Identification                               |
| PLL     | Phase-Locked Loop                                    |
| POR     | Power On Reset                                       |
| RFO     | Radio Frequency Output                               |
| RFU     | Reserved for Future Use                              |
| RoHS    | Restriction of Hazardous Substances                  |
| RTC     | Real-Time Clock                                      |
| SATCOM  | Satellite Communications                             |
| SCK     | Serial Clock                                         |
| SF      | Spreading Factor                                     |
| SN      | Sequence Number                                      |
| SNR     | Signal to Noise Ratio                                |
| SPI     | Serial Peripheral Interface                          |
| STDBY   | Standby                                              |
| TCXO    | Temperature-Compensated Crystal Oscillator           |
| VR      | Voltage Regulator                                    |
| VR_PA   | Voltage Regulator Power Amplifier                    |
| WEEE    | Waste Electrical and Electronic Equipment            |
| Wi-SUN  | Wireless Smart Utility Network from Wi-SUN Alliance® |



**List of Acronyms and their Meaning (Sheet 4 of 4)**

| Acronym            | Meaning                             |
|--------------------|-------------------------------------|
| XOSC               | Crystal Oscillator                  |
| Z-Wave / Z-Wave LR | Z-Wave / Z-Wave Long Range Protocol |



### Important Notice

Information relating to this product and the application or design described herein is believed to be reliable, however such information is provided as a guide only and Semtech assumes no liability for any errors in this document, or for the application or design described herein.

Semtech reserves the right to make changes to the product or this document at any time without notice. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. Semtech warrants performance of its products to the specifications applicable at the time of sale, and all sales are made in accordance with Semtech's standard terms and conditions of sale.

SEMTECH PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS, OR IN NUCLEAR APPLICATIONS IN WHICH THE FAILURE COULD BE REASONABLY EXPECTED TO RESULT IN PERSONAL INJURY, LOSS OF LIFE OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. INCLUSION OF SEMTECH PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE

UNDERTAKEN SOLELY AT THE CUSTOMER'S OWN RISK. Should a customer purchase or use Semtech products for any such unauthorized application, the customer shall indemnify and hold Semtech and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs damages and attorney fees which could arise.

The Semtech name and logo are registered trademarks of the Semtech Corporation. All other trademarks and trade names mentioned may be marks and names of Semtech or their respective companies. Semtech reserves the right to make changes to, or discontinue any products described in this document without further notice. Semtech makes no warranty, representation or guarantee, express or implied, regarding the suitability of its products for any particular purpose. All rights reserved.

© Semtech 2026