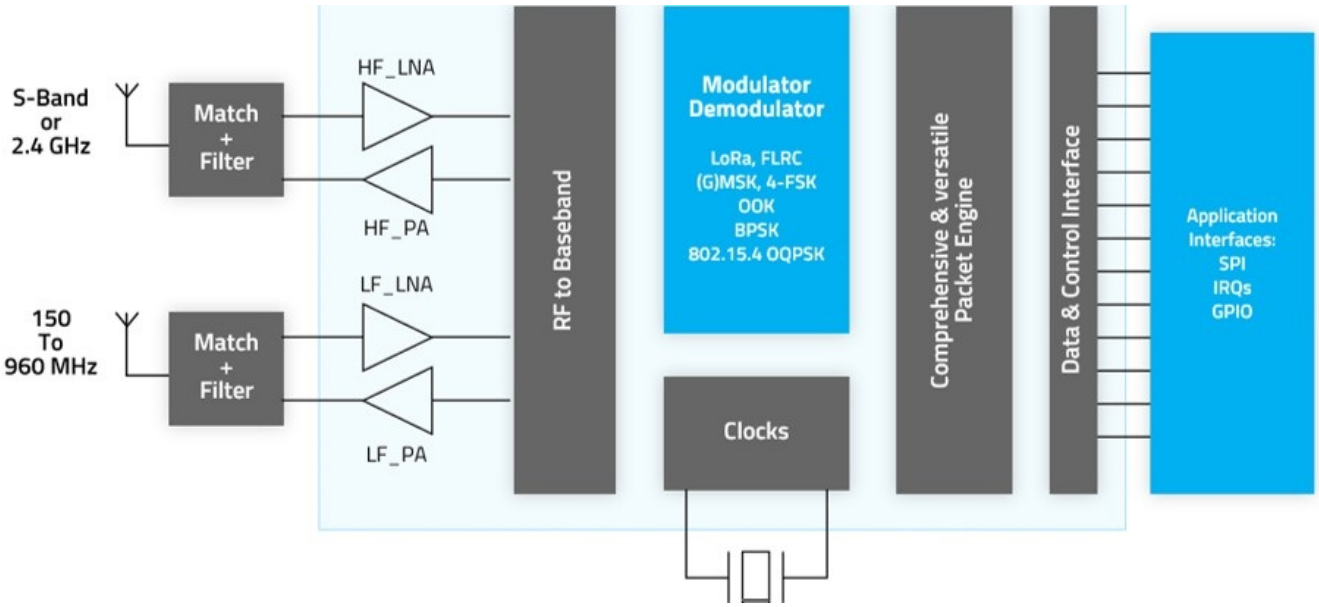
LR2021/LR2022/LR2012 Datasheet

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



Characteristics

- Receiver (Rx) Performance
  - LoRa® sub-GHz: -141.5dBm @LoRa® 125kHz
  - Bluetooth® LE PHY[1, 2]: -100dBm @1Mbps
    -97dBm @2Mbps
  - FLRC sub-GHz[1]: -111dBm @260kbps
    -104.5dBm @0.975Mbps
    -100.5dBm @1.95Mbps
  - O-QPSK 802.15.4 2.4GHz[1]: -103dBm @250kbps DSSS
  - O-QPSK sub-GHz[1]: -111dBm @100kbps DSSS
- Transmitter (Tx) Performance
  - Up to +22dBm @sub-GHz
  - Up to +12dBm @2.4GHz [1, 2]
  - 32dB dynamic in 0.5dB steps
- Low current consumption
  - Rx current 5.7mA @sub-GHz
  - Tx current 105mA @+22dBm
    27mA @+14dBm
    12mA @+10dBm sub-GHz
  - Low sleep current 470nA
- -40 to +85°C
- 1.8 to 3.7V 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 2.4GHz[1, 2] and Sub-GHz
- Amazon Sidewalk
- Sigfox BPSK
- Wi-SUN FSK
- Wireless M-BUS (device and collector)
- Bluetooth® LE 5.0 PHY compatible[1, 2]
- 2-FSK & 4-FSK based protocols
- Z-Wave®, Z-Wave LR[1]
- IEEE® 802.15.4™ Thread® and Zigbee™ O-QPSK[1]

Key Features

- 150 - 960MHz bands
- LoRa up to 125kbps
- 1.5 - 2.5GHz bands [1, 2]
- FLRC up to 2.6Mbps[1]
- LR-FHSS for SATCOM[1, 2] and sub-GHz

Downloaded from Arrow.com.

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, 2.4GHz ISM bands, and licensed L/S-bands. The transceiver is designed to be backward compatible with previous LoRa devices, ensuring seamless 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 advanced RF and analog architecture that supports a multi-region, single-SKU design, reducing external BOM costs, PCB footprint, and power consumption, while offering superior 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 (non smart-home) and LR2012 (sub-GHz industrial-focused bands) devices, defined as follows:

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.

Downloaded from Arrow.com.

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
2.1	April 2026	Expanded LR2021 datasheet to cover LR2022 and LR2012 specifications, also modified: • *Figure 1-9* description, to specify +12dBm is the power setting. • FLRC BW values in *Table 18-1*, *Table 18-3* and *Section 18.4.1* • *Section 9.3* • Added commands: *9.9.12 SetLoraTxSync* • Added *11.1.3 (G)FSK Limitations* • Modified *Figure 23-1: LR2021/LR2022 Reference Design Schematic* • Added expected degradation using SIMO with metallic RF shield in *Section 23.1* • Electrical spec conditions have been clarified to "no shield, SIMO" • Modified IDDSL1 value in *Table 3-4* • Modified Table 3-5 for 434/490MHz and Table 7-17 • Modified BRFLRC Max value in *Table 3-8* • Modified ZW_R2_S, ZW_R3_S, ZWLR1 Typ values in *Table 3-10* • Added 0.5dB to all sensitivity numbers in *Table 3-13* and *Table 3-18* • Added *Section 22.3 Firmware Patch RAM (PRAM)* • Modified footnote in *Table 18-2* • Added FLRC_260_CR05_915_S effective bit rate in *Table 3-12*

Downloaded from Arrow.com.

Contents

1. Chip Description 19
1.1 LR2021 19
1.2 LR2022 20
1.3 LR2012 21
1.4 Receiver RF Path 22
1.5 Transmitter RF Path 23
1.5.1 Low Frequency Power Amplifier (LF PA) (LR20xx) 24
1.5.2 High Frequency Power Amplifier (HF PA) (LR2021/22) 27
1.5.3 Direct-Tie Implementation (LR20xx) 28
1.6 Frequency Synthesizer 28
1.7 Modem 29
1.8 Power Distribution 30
1.8.1 Power-On Reset and Brown-Out Reset 30
1.8.2 Low Battery Detector 30
1.8.3 Over Current Protection 30
1.8.4 Power Supply Mode 31
1.9 Clock Sources 32
1.9.1 32kHz RC Oscillator 32
1.9.2 32MHz Crystal 32
1.9.3 32MHz TCXO 33
1.9.4 External Clock Reference Inputs 33
1.9.5 Output Clock 33
2. Pin Connection 34
2.1 I/O Description 34
3. Specifications (LR20xx) 37
3.1 Absolute Maximum Ratings 37
3.2 Operating Range 37
3.3 ESD Protection and Latch-up 38
3.4 Electrical Specifications 38
3.4.1 Power Consumption 39
3.4.2 General Specifications 42
3.4.3 Receive Mode Specifications 44
3.4.4 Transmit Mode Specifications 66
3.5 Reference Oscillator Crystal Specification 68
3.6 SIMO Inductor Specification 68
3.7 Digital I/O and Interface Specifications 69
3.7.1 Digital I/O Specifications 69
3.7.2 SPI Interface Specifications 70
4. Chip Modes (LR20xx) 71
4.1 Chip Mode State Diagram 71

Downloaded from Arrow.com.

4.2 Reset 73
4.3 Start-up and Boot Calibration Sequence 73
4.4 Chip Modes 74
4.4.1 Sleep 74
4.4.2 Standby RC / XOSC 74
4.4.3 Frequency Synthesis (FS) 74
4.4.4 Receive (Rx) 74
4.4.5 Transmit (Tx) 75
5. Host Interface 76
5.1 Hardware Interface (LR20xx) 76
5.1.1 DIO Pins 76
5.1.2 SPI Interface 76
5.2 Busy Line Control (LR20xx) 76
5.3 Data FIFO (LR20xx) 77
5.3.1 Rx Data FIFO 77
5.3.2 Tx Data FIFO 77
5.4 Host-Command Interface (LR20xx) 78
5.4.1 Read Write Command 78
5.5 DIO Rx / Tx Triggers (LR20xx) 80
5.6 Host Commands Summary 81
5.6.1 Direct Radio FIFO Read/Write Commands (LR20xx) 81
5.6.2 Register / Memory Access Operations (LR20xx) 81
5.6.3 System Configuration / Status Operations (LR20xx) 82
5.6.4 Radio Configuration / Status Operations (LR20xx except FLRC Commands) ... 84
5.7 Host Interrupts (LR20xx) 95
6. Chip Control Commands (LR20xx) 97
6.1 Direct Radio FIFO Read/Write 97
6.1.1 ReadRadioRxFifo 97
6.1.2 WriteRadioTxFifo 97
6.2 Register / Memory Access Operations 98
6.2.1 WriteRegMem32 98
6.2.2 WriteRegMemMask32 98
6.2.3 ReadRegMem32 99
6.3 Chip Mode Control 99
6.3.1 SetSleep 99
6.3.2 SetStandby 100
6.3.3 SetFs 100
6.3.4 SetAdditionalRegToRetain 100
6.3.5 SetRx 101
6.3.6 SetTx 102
6.3.7 SetRxTxFallbackMode 102
6.3.8 SetRxDutyCycle 103
6.3.9 SetAutoRxTx 104
6.3.10 StopTimeoutOnPreamble 104

Downloaded from Arrow.com.

6.3.11 SetLoraCadParams 105
6.3.12 SetLoraCAD 106
6.3.13 SetCad 107
6.3.14 SetCadParams 107
6.3.15 SetCca 108
6.3.16 ResetRxStats 108
6.3.17 SetDefaultRxTxTimeout 108
6.3.18 SetRegMode 109
6.4 Chip Auto Calibration 110
6.4.1 Calibrate 110
6.4.2 CalibFE 111
6.5 Measurements 112
6.5.1 GetVbat 112
6.5.2 GetTemp 113
6.6 Low Battery Detector Configuration 114
6.6.1 SetLbdCfg 114
6.6.2 GetRandomNumber 115
6.7 Chip Status 116
6.7.1 GetStatus 116
6.7.2 GetVersion 117
6.7.3 ClearErrors 117
6.7.4 GetErrors 118
6.8 DIO Control 119
6.8.1 SetDioFunction 119
6.8.2 SetDioRfSwitchConfig 120
6.8.3 SetDioIrqConfig 120
6.9 Interrupt Control 121
6.9.1 ClearIrq 121
6.9.2 GetAndClearIrqStatus 121
6.10 Data FIFO 122
6.10.1 ConfigFifoIrq 122
6.10.2 GetFifoIrqFlags 123
6.10.3 ClearFifoIrqFlags 123
6.10.4 GetAndClearFifoIrqFlags 124
6.10.5 GetRxFifoLevel 124
6.10.6 GetTxFifoLevel 125
6.10.7 ClearRxFifo 125
6.10.8 ClearTxFifo 125
6.11 Clock Control 126
6.11.1 ConfigLfClock 126
6.11.2 ConfigClkOutputs 126
6.11.3 SetTcxoMode 127
6.11.4 SetXoscCpTrim 128
6.12 Temperature Compensation 128

Downloaded from Arrow.com.

6.12.1 SetTempCompCfg 129
6.12.2 SetNtcParams 129
7. Radio Frequency Front End Control 130
7.1 Overview (LR2021/LR2022) 130
7.2 Overview (LR2012) 130
7.3 RF Front End Commands (LR20xx) 130
7.3.1 SetRfFrequency 130
7.3.2 SetRxPath 131
7.3.3 GetRssiInst 131
7.3.4 SetRssiCalibration 132
7.3.5 ResetRxStats 132
7.3.6 SetTimestampSource 133
7.3.7 GetTimestampValue 133
7.3.8 GetCcaResult 134
7.3.9 SetAgcGainManual 134
7.3.10 GetRxPktLength 135
7.4 Power Amplifier Control Commands (LR20xx) 136
7.4.1 SetPaConfig 136
7.4.2 SelPa 138
7.4.3 SetTxParams 140
8. Modem Configuration (LR20xx) 141
8.1 Packet Type Selection 142
8.1.1 SetPacketType 142
8.1.2 GetPacketType 143
9. LoRa Modulation (LR20xx) 144
9.1 LoRa Modulation Principle 144
9.1.1 Spreading Factor (SF) 144
9.1.2 LoRa Bandwidth (BWLORA) 144
9.1.3 Coding Rate (CR) 144
9.1.4 Low Data Rate Optimization (LDRO) 144
9.1.5 LoRa Symbol Rate (Rs) 145
9.2 LoRa Packet Handler 145
9.2.1 Preamble 145
9.2.2 Header (explicit packets only) 145
9.2.3 Payload 146
9.3 Multi-SF Detection 146
9.4 Channel Activity Detection (CAD) 147
9.5 Multi-SF CAD 147
9.6 Fast CAD 148
9.7 Extended LoRa Rx Bandwidth 148
9.8 Intra-packet Frequency Hopping Compatibility with SX127x 149
9.9 LoRa Commands 150
9.9.1 SetLoraModulationParams 150
9.9.2 SF6 Compatibility with SX127x 151

Downloaded from Arrow.com.

9.9.3 SetLoraPacketParams 152
9.9.4 SetLoRaSynchTimeout 153
9.9.5 SetLoraSyncword 153
9.9.6 SetLoraSideDetConfig 154
9.9.7 SetLoraSideDetSyncword 154
9.9.8 GetLoraRxStats 155
9.9.9 GetLoraPacketStatus 156
9.9.10 SetLoraAddress 157
9.9.11 SetLoraHopping 157
9.9.12 SetLoraTxSync 158
9.9.13 SetLoraSideDetCad 158
10. RTToF (Round-Trip Time of Flight) (LR20xx) 159
10.1 RTToF Engine 159
10.2 RTToF Commands 159
10.2.1 SetRangingAddr 159
10.2.2 SetRangingReqAddr 159
10.2.3 GetRangingResult 160
10.2.4 GetRangingStats 161
10.2.5 SetRangingTxRxDelay 162
10.2.6 SetRangingParams 162
11. (G)FSK Modulation (LR20xx) 163
11.1 (G)FSK Modulation Principle 163
11.1.1 Modem Bandwidth and Bit Rates 163
11.1.2 (G)FSK Modulation Index 164
11.1.3 (G)FSK Limitations 164
11.2 (G)FSK Packet Handler 165
11.2.1 Fixed-Length Packet 165
11.2.2 Variable-Length Packet 166
11.3 (G)FSK Commands 167
11.3.1 SetFskModulationParams 167
11.3.2 SetFskPacketParams 169
11.3.3 SetFskWhiteningParams 172
11.3.4 SetFskCrcParams 172
11.3.5 SetFskSyncword 173
11.3.6 SetFskAddress 173
11.3.7 GetFskRxStats 174
11.3.8 GetFskPacketStatus 175
12. Wireless M-Bus (WM-BUS) (LR20xx) 176
12.1 WM-BUS Modulation 176
12.2 WM-BUS Packet Handler 177
12.3 WM-BUS Radio Commands 177
12.3.1 SetWmbusParams 177
12.3.2 GetWmbusRxStats 178
12.3.3 GetWmbusPacketStatus 179

Downloaded from Arrow.com.

12.3.4 SetWmbusFilteringAddress 180
13. Wi-SUN FSK (LR20xx) 181
13.1 Wi-SUN Modulation 181
13.2 Wi-SUN Packet Handler 181
13.3 Wi-SUN Radio Commands 182
13.3.1 SetWisunMode 182
13.3.2 SetWisunPacketParams 183
13.3.3 GetWisunRxStats 184
13.3.4 GetWisunPacketStatus 185
13.3.5 SetWisunPacketLen 185
14. Bluetooth Low Energy Physical Layer (LR2021/LR2022) 186
14.1 Bluetooth LE Modulation 186
14.2 Bluetooth LE Packet Handler 186
14.3 Bluetooth LE Radio Commands 186
14.3.1 SetBleModulationParams 186
14.3.2 SetBleChannelParams 187
14.3.3 SetBleTxPduLen 187
14.3.4 SetBleTx 188
14.3.5 GetBleRxStats 188
14.3.6 GetBlePacketStatus 189
15. BPSK Modem (LR20xx) 190
15.1 (D)BPSK Modulation 190
15.2 BPSK Radio Commands 190
15.2.1 SetBpskModulationParams 190
15.2.2 SetBpskPacketParams 191
16. OOK Modem (LR20xx) 192
16.1 OOK Modulation 192
16.2 OOK Packet Handler 192
16.3 OOK Commands 192
16.3.1 SetOokModulationParams 192
16.3.2 SetOokPacketParams 193
16.3.3 SetOokCrcParams 194
16.3.4 SetOokSyncword 194
16.3.5 SetOokAddress 195
16.3.6 GetOokRxStats 195
16.3.7 GetOokPacketStatus 196
16.3.8 SetOokDetector 197
16.3.9 SetOokwhiteningParams 198
17. LR-FHSS (LR20xx) 199
17.1 LR-FHSS Modulation 199
17.2 LR-FHSS Commands 200
17.2.1 LrFhssBuildFrame 200
17.2.2 LrFhssSetSyncword 201
17.2.3 ReadLrFhssHoppingTable 202

Downloaded from Arrow.com.

17.2.4 WriteLrFhssHoppingTable 202
18. FLRC Modem (LR2021) 203
18.1 FLRC Modulation 203
18.1.1 Modem Bandwidth and Bit Rates 203
18.1.2 FEC Coding Rate 204
18.1.3 Gaussian Filtering 205
18.1.4 FLRC Frequency Tolerance 206
18.2 FLRC Packet Handler 207
18.2.1 Fixed-Length Packet Format 207
18.2.2 Variable-length Packet Format 207
18.3 FLRC Time-on-Air 208
18.4 FLRC Radio Commands 209
18.4.1 SetFlrcModulationParams 209
18.4.2 SetFlrcPacketParams 210
18.4.3 GetFlrcRxStats 211
18.4.4 GetFlrcPacketStatus 212
18.4.5 SetFlrcSyncword 212
19. OQPSK_15_4 (LR2021) 213
19.1 OQPSK_15_4 Modulation 213
19.2 OQPSK_15_4 Packet Handler 213
19.3 OQPSK_15_4 Radio Commands 213
19.3.1 SetOqpskParams 213
19.3.2 GetOqpskRxStats 214
19.3.3 GetOqpskPacketStatus 215
19.3.4 SetOqpskPacketLen 215
19.3.5 SetOqpskAddress 216
20. Z-Wave (LR2021) 217
20.1 Z-Wave Modulation 217
20.2 Z-Wave Packet Handler 217
20.3 Z-Wave Radio Commands 218
20.3.1 SetZwaveParams 218
20.3.2 SetZwaveHomeIdFiltering 219
20.3.3 GetZwaveRxStats 219
20.3.4 GetZwavePacketStatus 220
20.3.5 SetZwaveBeamFiltering 221
20.3.6 SetZwaveScanConfig 222
20.3.7 SetZwaveScan 223
21. Test Commands (LR20xx) 224
21.1 Regulatory Overview 224
21.1.1 ETSI 224
21.1.2 FCC 224
21.2 SetTxTestMode 225
22. Known Limitations and Workarounds 226
22.1 OOK Detection Threshold Adjustment (LR20xx) 226

Downloaded from Arrow.com.

22.1.1 Description 226
22.1.2 Workaround 226
22.2 RTTOF Accuracy for SX1280-compatible Bandwidths (LR2021/LR2022) 226
22.3 Firmware Patch RAM (PRAM) 227
22.3.1 Loading the Patch RAM 227
22.3.2 Checking the Patch RAM 227
23. Application Information 228
23.1 Power Management 228
23.2 RF Shield Impact 228
23.3 Oscillator Recommendations 229
23.4 Reference Design 230
23.5 Unused Pins 233
23.6 Configurations via Register Change 233
23.7 OCP registers 233
23.8 Compatibility of Generation 1-4 LoRa Radios 234
24. Package Information 235
24.1 Package Outline Drawing 235
24.2 Package Marking 236
24.3 Land Pattern 237
24.4 Reflow Profiles 237
24.5 Thermal Information 238
24.6 Tape & Reel Information 238

Downloaded from Arrow.com.

List of Figures

Figure 1-1: LR2021 Block Diagram 19
Figure 1-2: LR2022 Block Diagram 20
Figure 1-3: LR2012 Block Diagram 21
Figure 1-4: Power Amplifiers 23
Figure 1-5: LF PA Tx Power and IDD Tx vs Targeted Power (915MHz, 3.3V SIMO) 24
Figure 1-6: LF PA Tx Power and IDD Tx vs Targeted Power (490MHz, 3.3V SIMO) 25
Figure 1-7: LF PA Tx Power vs VBAT for +14, +17, +22dBm Targeted Output Power 26
Figure 1-8: PA IDD Tx vs VBAT for +14, +17, +22dBm Targeted Output Power 26
Figure 1-9: HF PA Tx Power and IDD Tx vs Targeted Power (2445MHz, 3.3V SIMO) 27
Figure 1-10: Switchless Dual Band Operation 28
Figure 1-11: Modem Top View 29
Figure 1-12: POR and BRN Functions 30
Figure 1-13: SIMO Configuration and Connection 31
Figure 1-14: LDO Configuration and Connection 31
Figure 1-15: NTC Connection 32
Figure 1-16: TCXO Circuit Diagram 33
Figure 2-1: LR2021/22 Pin-out in QFN32 Package 35
Figure 2-2: LR2012 Pin-out in QFN32 Package 36
Figure 3-1: SPI Timing Diagram 70
Figure 4-1: Chip Modes 71
Figure 5-1: Write Command 76
Figure 5-2: Write Command 78
Figure 5-3: Read Command 79
Figure 5-4: Direct Read Timing Diagram 79
Figure 5-5: Direct Write Timing Diagram 79
Figure 5-6: Tx Trigger Example 80
Figure 5-7: Tx Trigger Followed by FIFO Write Example 80
Figure 8-1: Modem Control Commands 141
Figure 9-1: LoRa Signal Bandwidth 144
Figure 9-2: LoRa Packet Format 145
Figure 9-3: LoRa Fast CAD Principle 148
Figure 11-1: FSK Modulation Parameters 163
Figure 11-2: Fixed-Length Packet Format 165
Figure 11-3: Variable-Length Packet Format 166
Figure 14-1: Bluetooth Low Energy Packet 186
Figure 17-1: LR-FHSS Spectral Plot Example 199
Figure 18-1: FLRC Fixed-length Packet Format 207
Figure 18-2: FLRC Variable-length Packet Format 207
Figure 23-1: LR2021/LR2022 Reference Design Schematic 230
Figure 23-2: LR2012 Reference Design Schematic 231
Figure 23-3: LR2021/LR2022 Reference Design PCB Layout 232
Figure 23-4: LR2012 Reference Design PCB Layout 232
Figure 24-1: Package Outline Drawing 235
Figure 24-2: Package Marking 236
Figure 24-3: Land Pattern 237
Figure 24-4: Tape & Reel Information 238

Downloaded from Arrow.com.

List of Tables

Table 2-1: Pin-out Description 34
Table 3-1: Absolute Maximum Ratings (LR20xx) 37
Table 3-2: Operating Range (LR20xx) 37
Table 3-3: ESD and Latch-up (LR20xx) 38
Table 3-4: Basic Modes Power Consumption (LR20xx) 39
Table 3-5: Transmit Mode Power Consumption (LR20xx unless specified otherwise) 39
Table 3-6: Receive Mode Power Consumption (LR20xx) 40
Table 3-7: 2.4GHz Specific Receive Mode Power Consumption (LR2021/LR2022) 41
Table 3-8: General Specifications (LR20xx unless specified otherwise) 42
Table 3-9: WM-Bus Receiver Specification (LR20xx) 44
Table 3-10: Z-Wave 1% PER (LR2021) 47
Table 3-11: Wi-SUN FSK 10% PER (LR20xx) 48
Table 3-12: FLRC Sub-GHz 1% PER (LR2021) 50
Table 3-13: FLRC 2.4GHz 1% PER (Packet Size = 31 bytes) (LR2021) 52
Table 3-14: FSK Sensitivity @0.1% BER (LR20xx) 54
Table 3-15: Bluetooth LE Physical Layer (LR2021/LR2022) 56
Table 3-16: OQPSK_15_4 (LR2021) 58
Table 3-17: LoRa Sub-GHz 64B Payload (LR20xx) 58
Table 3-18: LoRa 2.4GHz 64B Payload (LR2021/LR2022) 63
Table 3-19: OOK 1% PER (LR20xx) 64
Table 3-20: General Receiver Specifications (LR20xx unless specified otherwise) 65
Table 3-21: Brownout and Battery Specification (LR20xx) 66
Table 3-22: Transmit Mode Specification (LR20xx unless specified otherwise) 66
Table 3-23: Start-up/ Mode Switching Times Specifications (LR20xx) 67
Table 3-24: 32MHz Crystal Specifications. (LR20xx) For example NDK_NX2016SA 68
Table 3-25: TCXO Regulator Specifications (LR20xx) 68
Table 3-26: Digital I/O Specifications (LR20xx) 69
Table 3-27: SPI Timing Requirements (LR20xx) 70
Table 4-1: Chip Mode Conditions (LR20xx) 72
Table 5-1: Radio FIFO Memory Commands 81
Table 5-2: Register Memory Commands 81
Table 5-3: System Configuration Commands 82
Table 5-4: Common Radio Commands 84
Table 5-5: LoRa Packet Radio Commands 86
Table 5-6: FSK Packet Radio Commands 87
Table 5-7: FLRC Packet Radio Commands 88
Table 5-8: BPSK Packet Radio Commands 88
Table 5-9: LR-FHSS Packet Radio Commands 89
Table 5-10: Bluetooth Low Energy Packet Radio Commands 90
Table 5-11: WM-Bus Packet Radio Commands 90
Table 5-12: Wi-SUN Packet Radio Commands 91
Table 5-13: Z-Wave Packet Radio Commands 92
Table 5-14: OQPSK_15_4 Packet Radio Commands 93
Table 5-15: RTToF Packet Radio Commands 93
Table 5-16: OOK Packet Radio Commands 94
Table 5-17: Host Interrupts 95

Downloaded from Arrow.com.

Table 6-1: ReadRadioRxFifo Command 97
Table 6-2: WriteRadioTxFifo Command 97
Table 6-3: WriteRegMem32 Command 98
Table 6-4: WriteRegMemMask32 Command 98
Table 6-5: ReadRegMem32 Command 99
Table 6-6: ReadRegMem32 Response 99
Table 6-7: SetSleep Command 99
Table 6-8: SetStandby Command 100
Table 6-9: SetFs Command 100
Table 6-10: SetAdditionalRegToRetain Command 100
Table 6-11: SetRx Command 101
Table 6-12: SetTx Command 102
Table 6-13: SetRxTxFallbackMode Command 102
Table 6-14: SetRxDutyCycle Command 103
Table 6-15: SetAutoRxTx Command 104
Table 6-16: StopTimeoutOnPreamble Command 104
Table 6-17: SetLoraCadParams Command 105
Table 6-18: CadExitMode Parameter Definition 105
Table 6-19: Lora CAD Recommended det_peak values 105
Table 6-20: SetLoRaCAD Command 106
Table 6-21: SetCad Command 107
Table 6-22: SetCadParams Command 107
Table 6-23: SetCca Command 108
Table 6-24: ResetRxStats Command 108
Table 6-25: SetDefaultRxTxTimeout Command 108
Table 6-26: SetRegMode Command 109
Table 6-27: Calibrate Command 110
Table 6-28: Blocks_to_calibrate Parameter Definition 110
Table 6-29: CalibFE Command 111
Table 6-30: GetVBat Request Command 112
Table 6-31: GetVBat Response 112
Table 6-32: GetTemp Command 113
Table 6-33: GetTemp Response 113
Table 6-34: SetLbdCfg Command 114
Table 6-35: GetRandomNumber Request Command 115
Table 6-36: GetRandomNumber Response 115
Table 6-37: GetStatus Command 116
Table 6-38: Status (Stat) Definition 116
Table 6-39: GetVersion Command 117
Table 6-40: GetVersion Response 117
Table 6-41: ClearError Command 117
Table 6-42: GetErrors Command 118
Table 6-43: GetErrors Response 118
Table 6-44: SetDioFunction Command 119
Table 6-45: SetDioRfSwitchConfig Command 120
Table 6-46: SetDioIrqConfig Command 120
Table 6-47: ClearIrq Command 121
Table 6-48: GetAndClearIrqStatus Command 121
Table 6-49: GetAndClearIrqStatus Response 121
Table 6-50: ConfigFifoIrq Command 122
Table 6-51: GetFifoIrqFlags Request Command 123

Downloaded from Arrow.com.

Table 6-52: GetFifoIrqFlags Read Command 123
Table 6-53: ClearFifoIrqFlags Command 123
Table 6-54: GetAndClearFifoIrqFlags Command 124
Table 6-55: GetAndClearFifoIrqFlags Response 124
Table 6-56: GetRxFifoLevel Command 124
Table 6-57: GetRxFifoLevel Response 124
Table 6-58: GetTxFifoLevel Command 125
Table 6-59: GetTxFifoLevel Response 125
Table 6-60: ClearRxFifo Command 125
Table 6-61: ClearTxFifo Command 125
Table 6-62: ConfigLfClock Command 126
Table 6-63: ConfigClkOutputs Command 126
Table 6-64: SetTcxoMode Command 127
Table 6-65: TCXO Supply Voltage Programming Values 127
Table 6-66: SetXoscCpTrim Command 128
Table 6-67: XTA Capacitor Value 128
Table 6-68: XTB Capacitor Value 128
Table 6-69: SetTempCompCfg Command 129
Table 6-70: SetNtcParams Command 129
Table 7-1: SetRfFrequency Command 130
Table 7-2: SetRxPath Command 131
Table 7-3: GetRssiInst Command 131
Table 7-4: GetRssiInst Response 131
Table 7-5: SetRssiCalibration Command 132
Table 7-6: ResetRxStats Command 132
Table 7-7: SetTimestampSource Command 133
Table 7-8: GetTimestampValue Command 133
Table 7-9: GetTimestampValue Response 133
Table 7-10: GetCcaResult Command 134
Table 7-11: GetCcaResult Response 134
Table 7-12: SetAgcGainManual Command 134
Table 7-13: GetRxPktLength Command 135
Table 7-14: GetRxPktLength Response 135
Table 7-15: SetPaConfig Command 136
Table 7-16: Optimal Values for 915MHz Semtech Reference Design 137
Table 7-18: Optimal Values for 2445MHz Semtech Reference Design (LR2021/LR2022) 138
Table 7-19: SelPa Command 138
Table 7-17: Optimal Values for 490MHz Semtech Reference Design 138
Table 7-20: SetTxParams Command 140
Table 7-21: Tx Ramp_time Parameter Definition 140
Table 8-1: SetPacketType Command 142
Table 8-2: GetPacketType Command 143
Table 8-3: GetPacketType Response 143
Table 9-1: SetLoraModulationParams Command 150
Table 9-2: LoRa Modulation sf Parameter Definition 150
Table 9-3: LoRa Modulation bw Parameter Definition 150
Table 9-4: LoRa Modulation cr Parameter Definition 151
Table 9-5: SetLoRaPacketParams Command 152
Table 9-6: SetLoraSynchTimeout Command 153
Table 9-7: SetLoraSyncword Command 153
Table 9-8: SetLoraSideDetConfig Command 154

Downloaded from Arrow.com.

Table 9-9: SetLoraSideDetSyncword Command 154
Table 9-10: GetLoraRxStats Command 155
Table 9-11: GetLoraRxStats Response 155
Table 9-12: GetLoraPacketStatus Command 156
Table 9-13: GetLoraPacketStatus Response 156
Table 9-14: SetLoraAddress Command 157
Table 9-15: SetLoRaHopping Command 157
Table 9-16: SetLoraTxSync Command 158
Table 9-17: SetLoraSideDetCad Command 158
Table 10-1: SetRangingAddr Command 159
Table 10-2: SetRangingReqAddr Command 159
Table 10-3: GetRangingResult Command 160
Table 10-4: GetRangingResult Response 160
Table 10-5: GetRangingStats Request Command 161
Table 10-6: GetRangingStats Response 161
Table 10-7: SetRangingTxRxDelay Command 162
Table 10-8: SetRangingParams Command 162
Table 11-1: Oversampling Rate Limitations 164
Table 11-2: (G)FSK Limitations Summary 164
Table 11-3: SetFskModulationParams Command 167
Table 11-4: Correlation of Bandwidth, rx_bw and Intermediate Frequency 168
Table 11-5: SetFskPacketParams Command 169
Table 11-6: (G)FSK Packet pbl_detect Parameter Definition 169
Table 11-7: (G)FSK Packet AddrComp Parameter Definition 170
Table 11-8: (G)FSK Packet pkt_format Parameter Definition 170
Table 11-9: (G)FSK packet CRC Parameter Definition 171
Table 11-10: (G)FSK packet dc-free Parameter Definition 171
Table 11-11: SetFskWhiteningParams Command 172
Table 11-12: SetFskCrcParams Command 172
Table 11-13: SetFskSyncword Command 173
Table 11-14: SetFskAddress Command 173
Table 11-15: GetFskRxStats Command Request 174
Table 11-16: GetFskRxStats Response 174
Table 11-17: GetFskPacketStatus Request Command 175
Table 11-18: GetFskPacketStatus Response 175
Table 12-1: Supported WM-BUS Modulation 176
Table 12-2: SetWmbusParams Command 177
Table 12-3: GetWmbusRxStats Request 178
Table 12-4: GetWmbusRxStats Response 178
Table 12-5: GetWmbusPacketStatus Request 179
Table 12-6: GetWmbusPacketStatus Response 179
Table 12-7: SetWmbusFilteringAddress Command 180
Table 13-1: Supported Wi-SUN Modulation 181
Table 13-2: SetWisunMode Command 182
Table 13-3: SetWisunPacketParams Command 183
Table 13-4: GetWisunRxStats Request Command 184
Table 13-5: GetWisunRxStats Response 184
Table 13-6: GetWisunPacketStatus Request Command 185
Table 13-7: GetWisunPacketStatus Response 185
Table 13-8: SetWisunPacketLen Command 185
Table 14-1: SetBleModulationParams Command 186

Downloaded from Arrow.com.

Table 14-2: SetBleChannelParams Command 187
Table 14-3: SetBleTxPduLen Command 187
Table 14-4: SetBleTx Command 188
Table 14-5: GetBleRxStats Command 188
Table 14-6: GetBleRxStats Response 188
Table 14-7: GetBlePacketStatus Command 189
Table 14-8: GetBlePacketStatus Response 189
Table 15-1: SetBpskModulationParams Command 190
Table 15-2: SetBpskPacketParams Command 191
Table 16-1: SetOokModulationParams Command 192
Table 16-2: SetOokPacketParams Command 193
Table 16-3: SetOokCrcParams Command 194
Table 16-4: SetOokSyncword Command 194
Table 16-5: SetOokAddress Command 195
Table 16-6: GetOokRxStats Command 195
Table 16-7: GetOokRxStats Response 195
Table 16-8: GetOokPacketStatus Command 196
Table 16-9: GetOokPacketStatus Response 196
Table 16-10: SetOokDetector Command 197
Table 16-11: SetOokwhiteningParams Command 198
Table 17-1: LrFhssBuildFrame Command 200
Table 17-2: LrFhssSetSyncword Command 201
Table 17-3: ReadLrFhssHoppingTable Command 202
Table 17-4: ReadLrFhssHoppingTable Command 202
Table 18-1: Valid FLRC Bit Rate and Bandwidth Combinations 203
Table 18-2: Effective FLRC Bit Rates 204
Table 18-3: Receiver Performance of the FLRC Modem 206
Table 18-4: SetFlrcModulationParams Command 209
Table 18-5: SetFlrcPacketParams Command 210
Table 18-6: GetFlrcRxStats Request Command 211
Table 18-7: GetFlrcRxStats Response 211
Table 18-8: GetFlrcPacketStatus Command 212
Table 18-9: GetFlrcPacketStatus Response 212
Table 18-10: SetFlrcSyncword Command 212
Table 19-1: SetOqpskParams Command 213
Table 19-2: GetOqpskRxStats Command 214
Table 19-3: GetOqpskRxStats Response 214
Table 19-4: GetOqpskPacketStatus Command 215
Table 19-5: GetOqpskPacketStatus Response 215
Table 19-6: SetOqpskPacketLen Command 215
Table 19-7: SetOqpskAddress Command 216
Table 20-1: Supported Z-Wave Modulation 217
Table 20-2: SetZwaveParams Command 218
Table 20-3: SetZwaveHomeIdFiltering Command 219
Table 20-4: GetZwaveRxStats Request Command 219
Table 20-5: GetZwaveRxStats Response 219
Table 20-6: GetZwavePacketStatus Request Command 220
Table 20-7: GetZwavePacketStatus Response 220
Table 20-8: SetZwaveBeamFiltering Command 221
Table 20-9: SetZwaveScanConfig Command 222
Table 20-10: SetZwaveScan Command 223

Downloaded from Arrow.com.

Table 21-1: ETSI Test Signals 224
Table 21-2: SetTxTestMode Command 225
Table 23-1: Sensitivity for SIMO with Metallic Shielding 228
Table 23-2: Register Usage 233
Table 24-1: Package Thermal Information 238

Downloaded from Arrow.com.

1. Chip Description

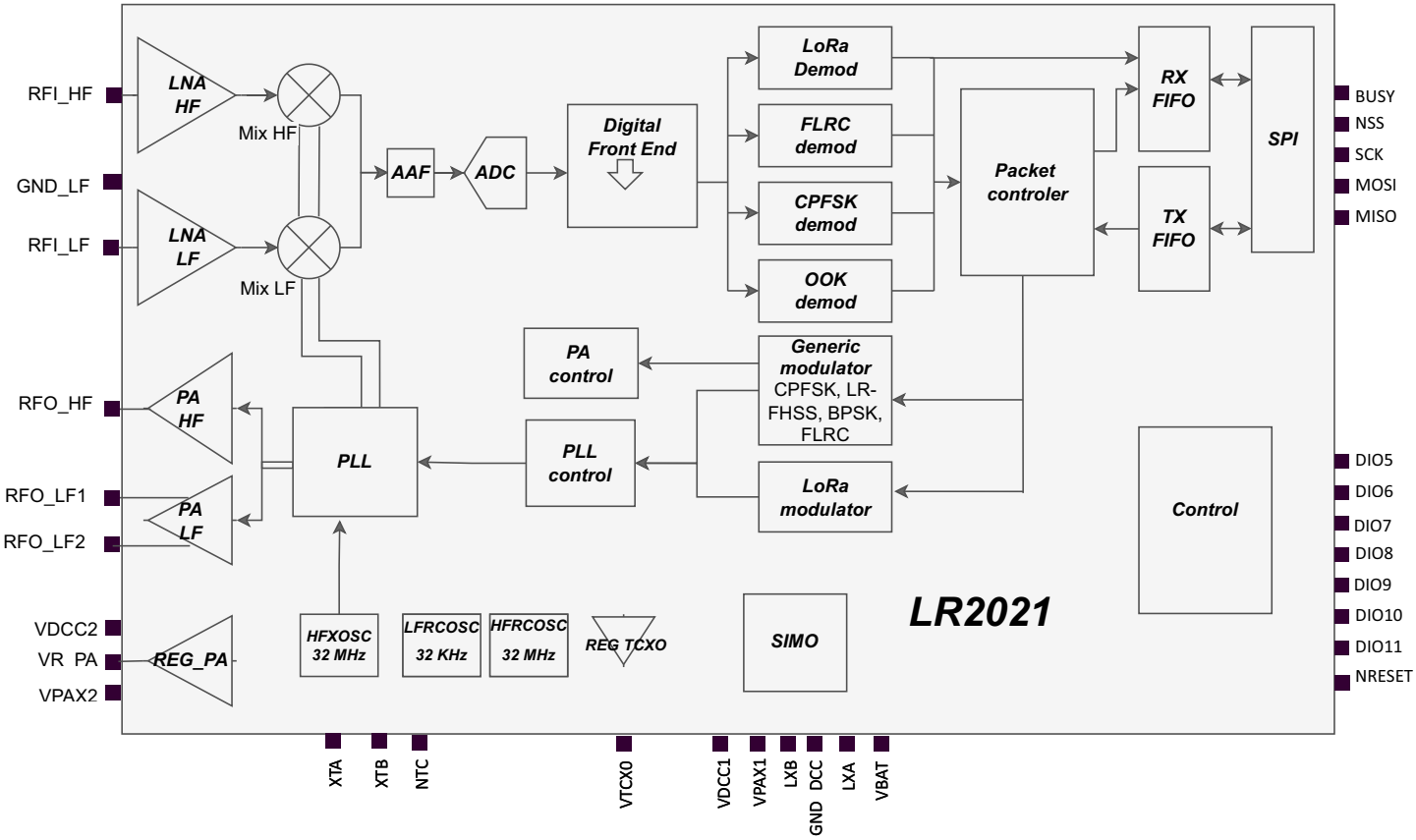
1.1 LR2021

The LR2021 is a highly capable half-duplex radio frequency (RF) transceiver, purpose-built to handle 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 a comprehensive set of modulation options to suit diverse wireless communication needs.

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

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

Figure 1-1: LR2021 Block Diagram

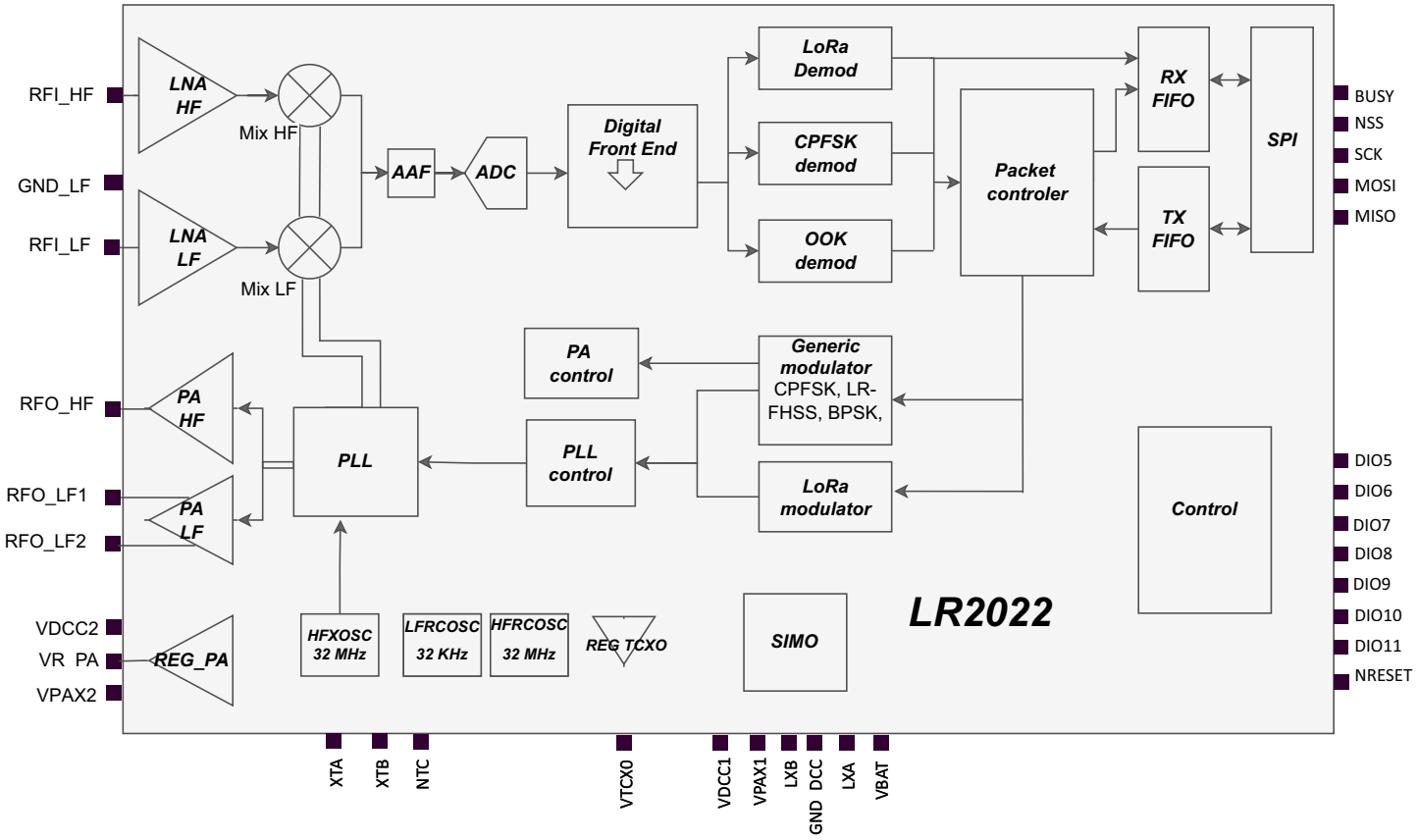


Downloaded from Arrow.com.

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

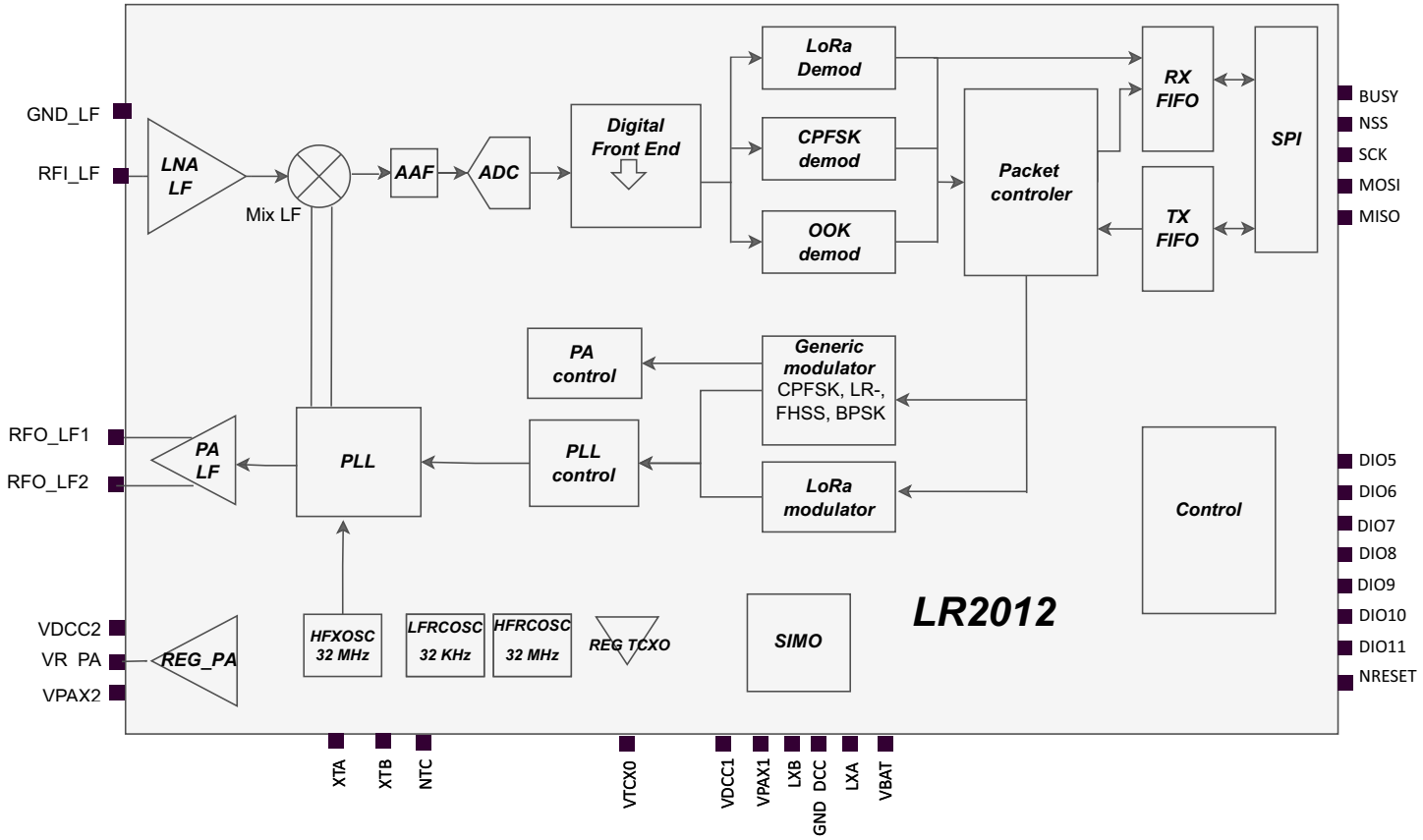


Downloaded from Arrow.com.

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



Downloaded from Arrow.com.

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 2.4GHz 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 - 2.5GHz frequency range through the RFI_HF input. This input is particularly suitable for signals in Band S, 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.

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 228* 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.

Downloaded from Arrow.com.

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.

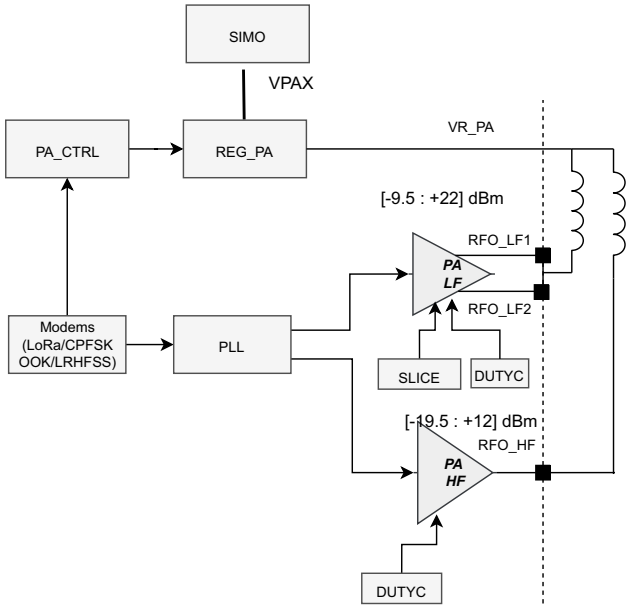
After modulation, the signal is further amplified using one of the two main Tx channels available in the circuit:

- High-Frequency Transmitter (PA_HF): This channel is dedicated to high-frequency transmission needs up to +12dBm, such as in the 1.5 - 2.5GHz 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 +22dBm, 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.

Downloaded from Arrow.com.

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

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

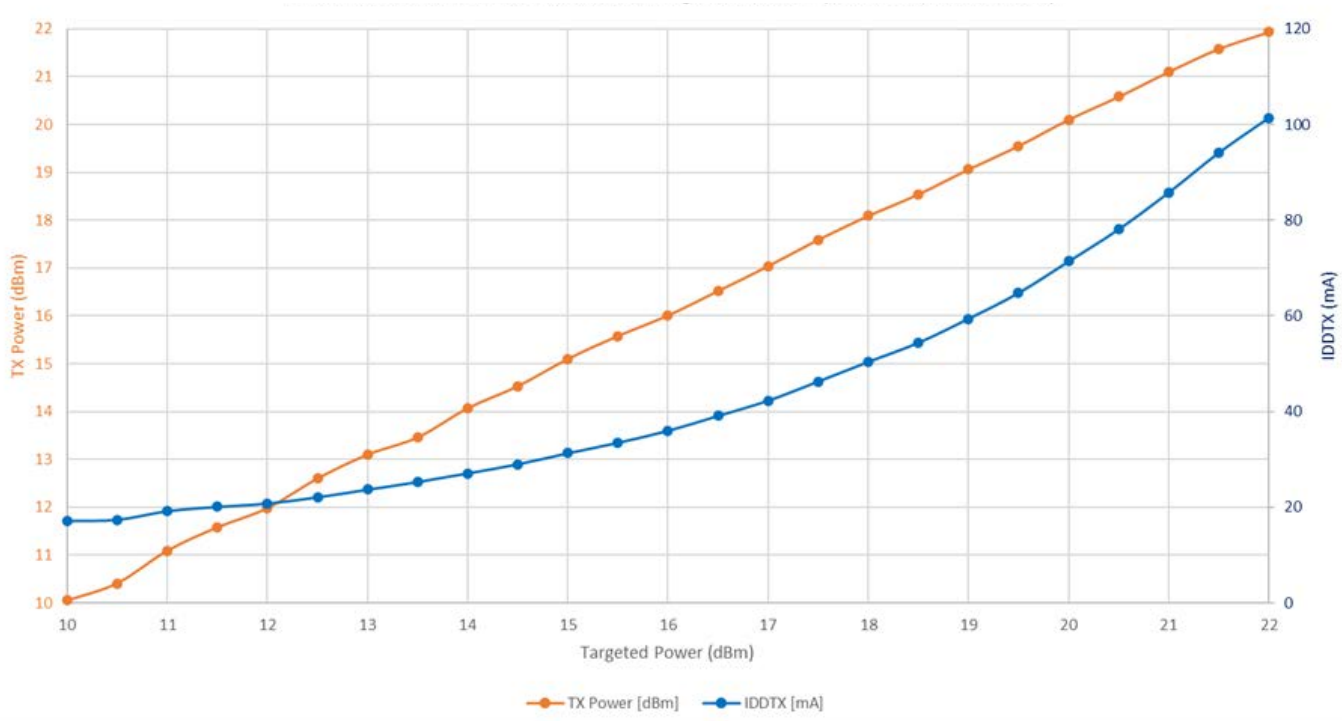
The PA output power is controlled by three main parameters, configurable through SPI commands:

- ***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. Please refer to *Table 7-16* & *Table 7-17* for optimal values for 490MHz and 915MHz 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 3.3V in SIMO configuration with these optimized settings for both Semtech 915MHz reference design, and 490MHz reference design respectively.

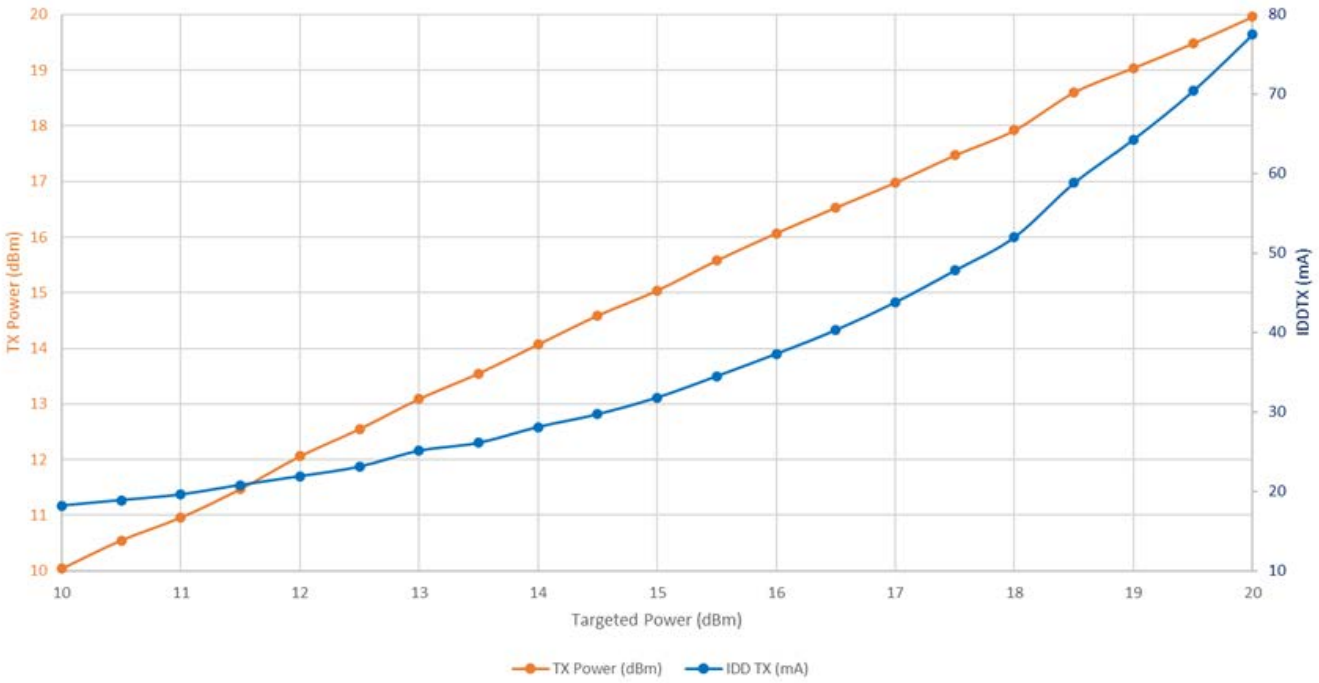
Figure 1-5: LF PA Tx Power and IDD Tx vs Targeted Power (915MHz, 3.3V SIMO)



Downloaded from Arrow.com.

The Tx Power (orange curve) demonstrates good linearity from 10dBm to 22dBm 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 18mA at +10dBm to 120mA at +22dBm.

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



Downloaded from Arrow.com.

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

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

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

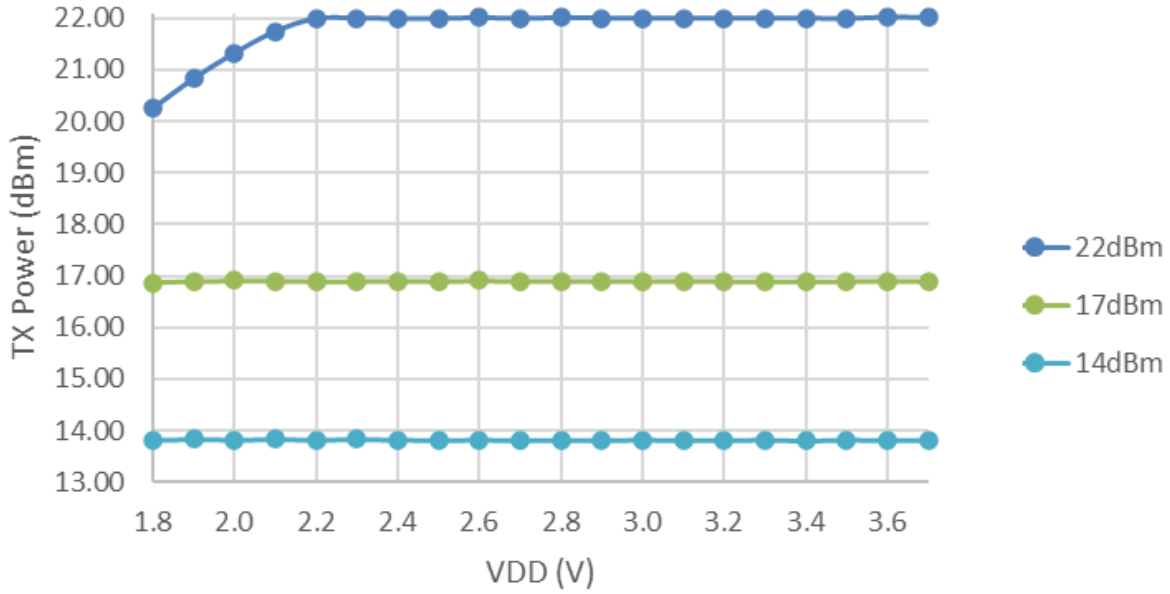
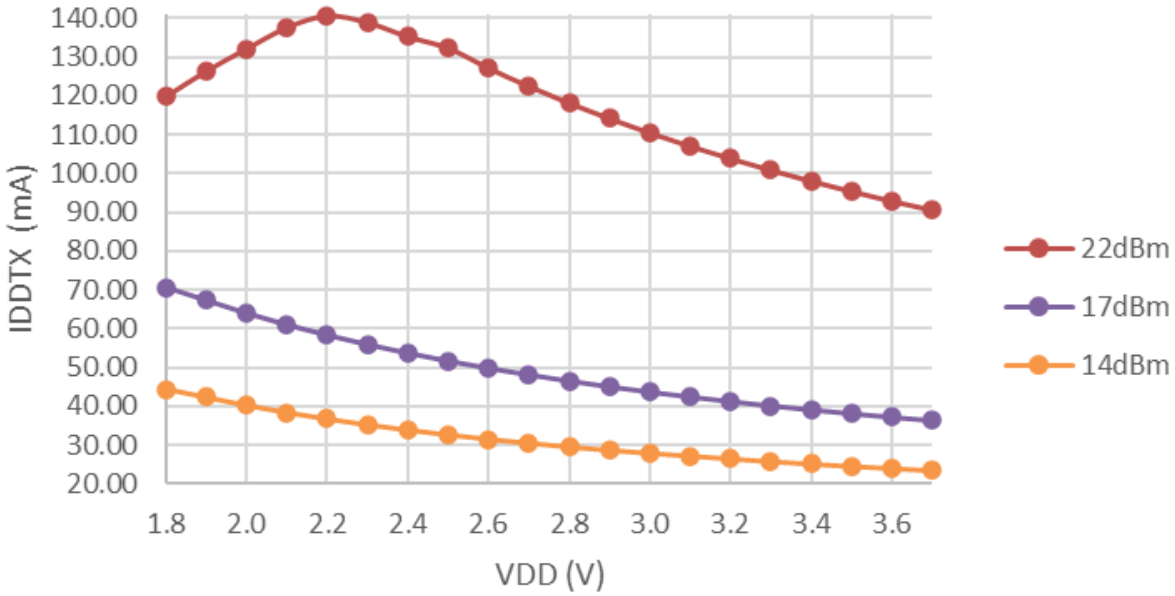


Figure 1-8 shows LF PA current consumption decreasing for +14dBm and +17dBm, with increasing VBAT due to improved regulator efficiency at higher supply voltages. The +22dBm setting exhibits a peak at approximately 2.2V (~140mA) where the regulator operates with reduced efficiency while output is clamped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

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



Downloaded from Arrow.com.

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

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

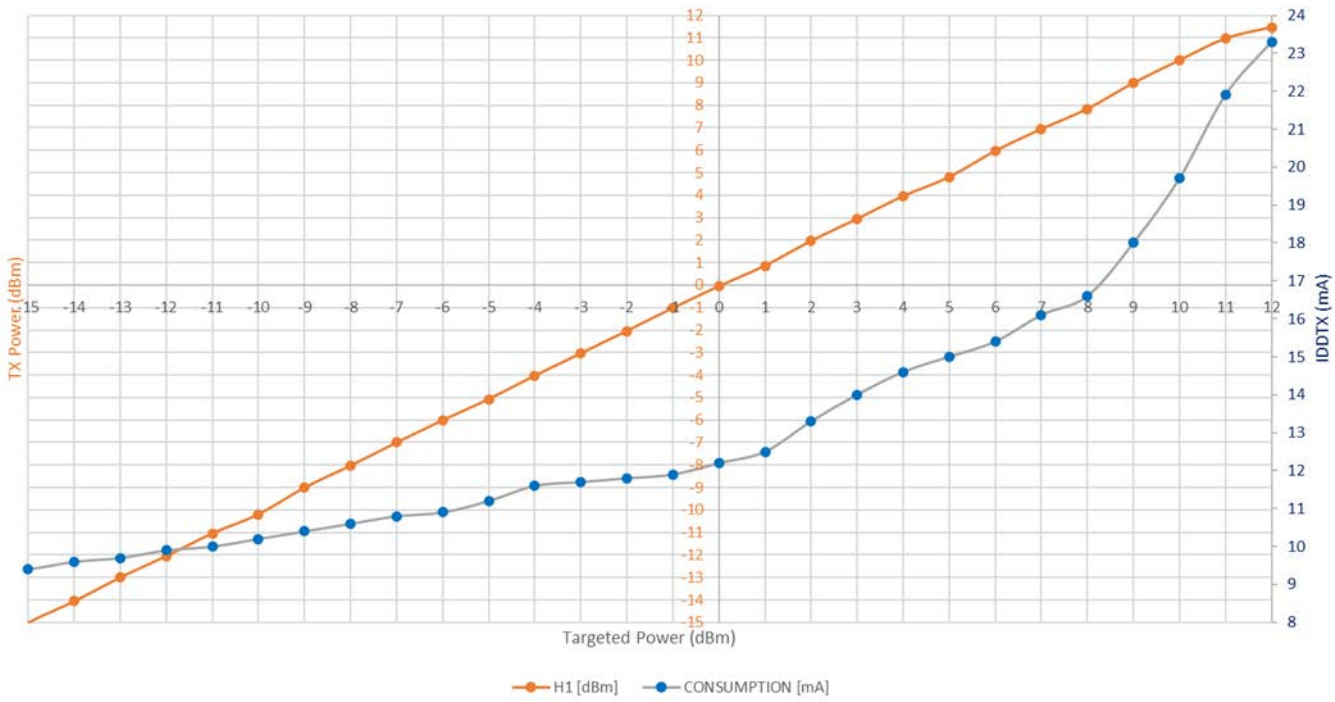
As for LF PA, the various output power steps can be obtained through combination of these parameters. Therefore a defined set of optimized PA parameters can be defined to reach the targeted Tx Power for each application PCB. Please refer to *Table 7-18* for *Optimal Values for 2445MHz Semtech Reference Design (LR2021/LR2022)*.

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

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

It is possible to obtain higher efficiency at 10dBm or less, contact Semtech for assistance.

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



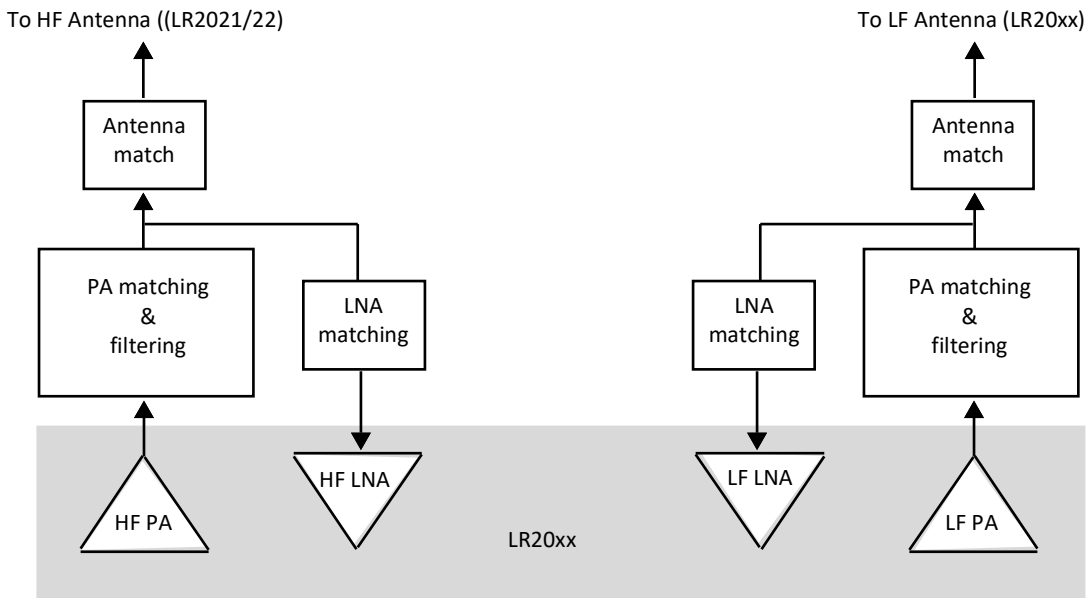
Downloaded from Arrow.com.

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



Note that 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 - 2500MHz 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).

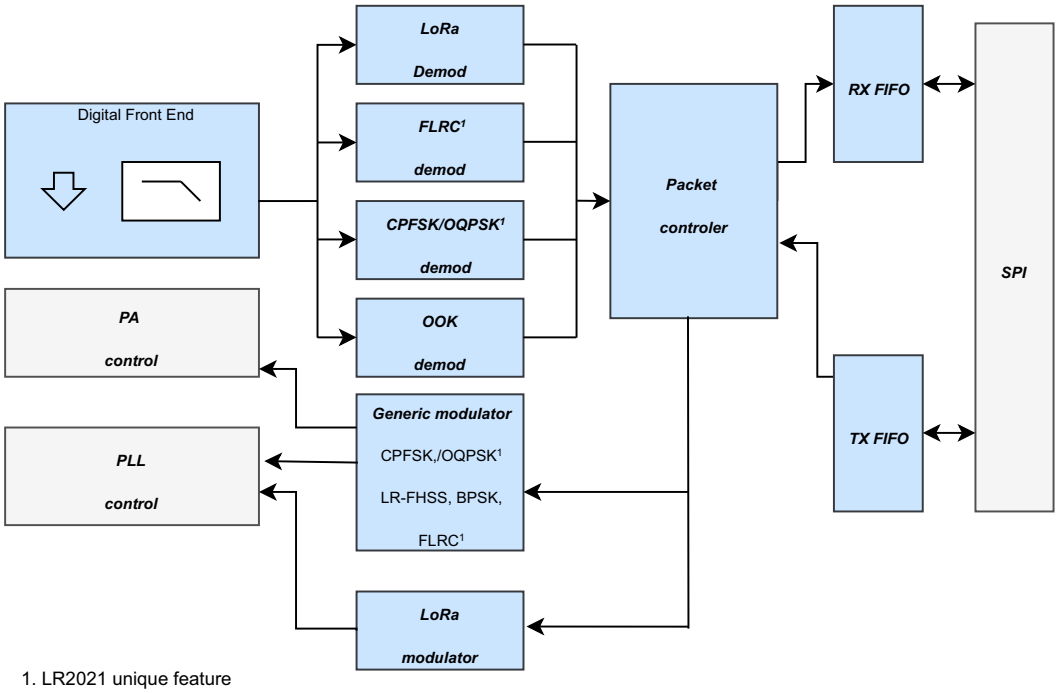
Downloaded from Arrow.com.

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 versatile modulator is utilized 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 case and more robust link thank s to a new FEC mechanism. Additionally, this modem extends the frequency capture range, providing enhanced capabilities for LoRa-based 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. LR2021 unique feature

Downloaded from Arrow.com.

1.8 Power Distribution

1.8.1 Power-On Reset and Brown-Out Reset

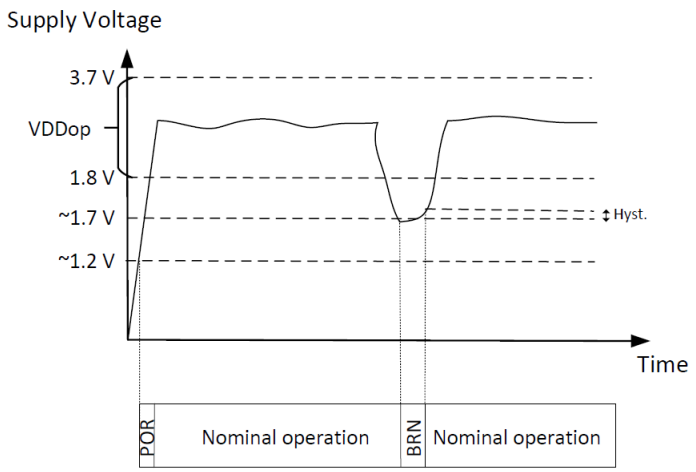
The LR2021/LR2022/LR2012 is equipped with a Power-On Reset (POR) and a Brown-Out Reset (BRN).

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 *SetDioIrqConfig*, offering a seamless 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. In order 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.

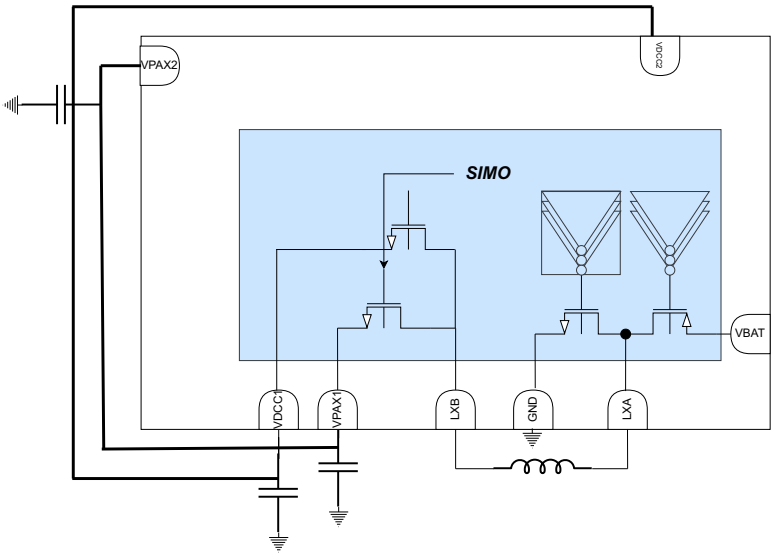
Downloaded from Arrow.com.

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:

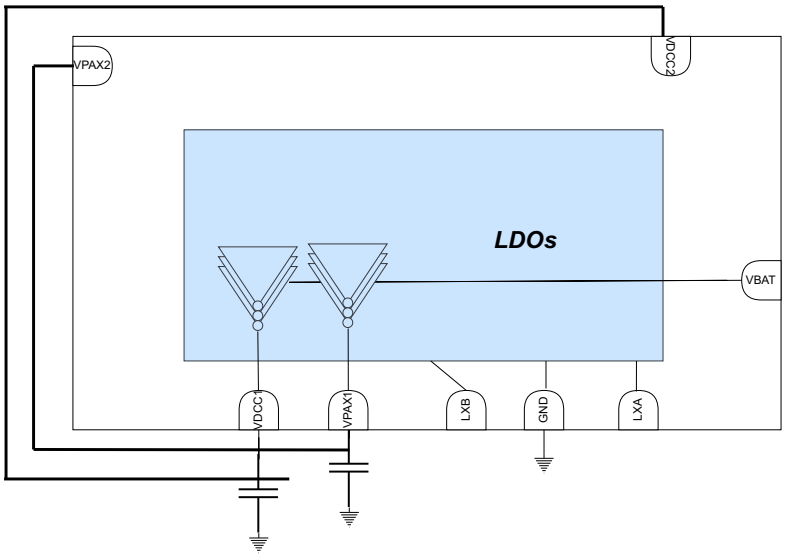
- VDCC: To supply the LDOs for the analogue and digital block of the circuit. Typical value for VDCC = 1.55V, max IDCC = 20mA. The regulated supply is available on pin VDCC1 and is connected to the rest of the chip externally via PCB and pin VDCC2.
- VPAX: To specifically supply the PA (through the REG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

Downloaded from Arrow.com.

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 32MHz 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 32kHz RC Oscillator

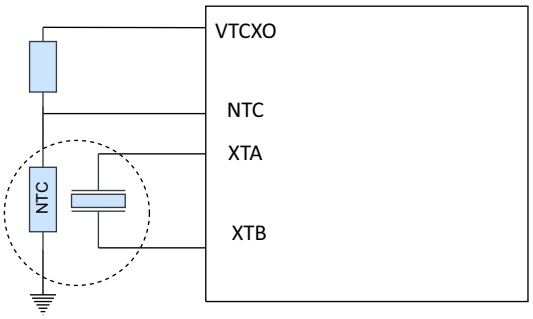
The 32kHz RC oscillator is optionally used by the circuit in Sleep mode to wake-up the device when performing periodic or duty cycled operations. Several commands make use of this 32kHz RC oscillator (RTC) to generate time-based events like time-outs.

1.9.2 32MHz Crystal

A 32MHz crystal oscillator provides the 32MHz 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 particularity useful when high power PA is desired on a small PCB footprint. Refer to AN1200.102 for further details.

Figure 1-15: NTC Connection



Downloaded from Arrow.com.

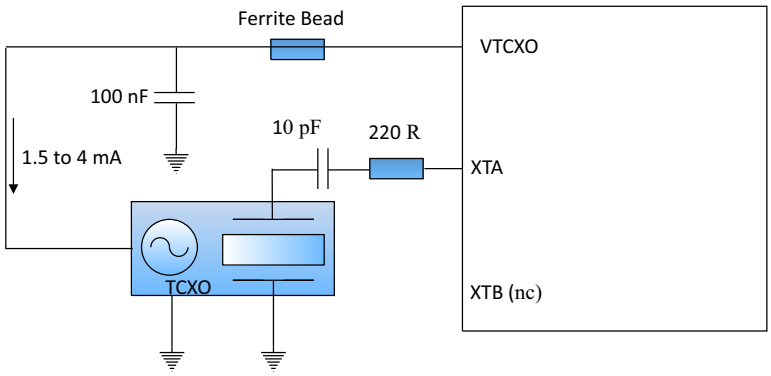
1.9.3 32MHz TCXO

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 (32kHz) 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 32MHz down to 976.562kHz through *ConfigClkOutputs*. Note that 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 32kHz. This internal 32kHz 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 RC32kHz oscillator is kept on during Sleep mode.

Downloaded from Arrow.com.

2. Pin Connection

2.1 I/O Description

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

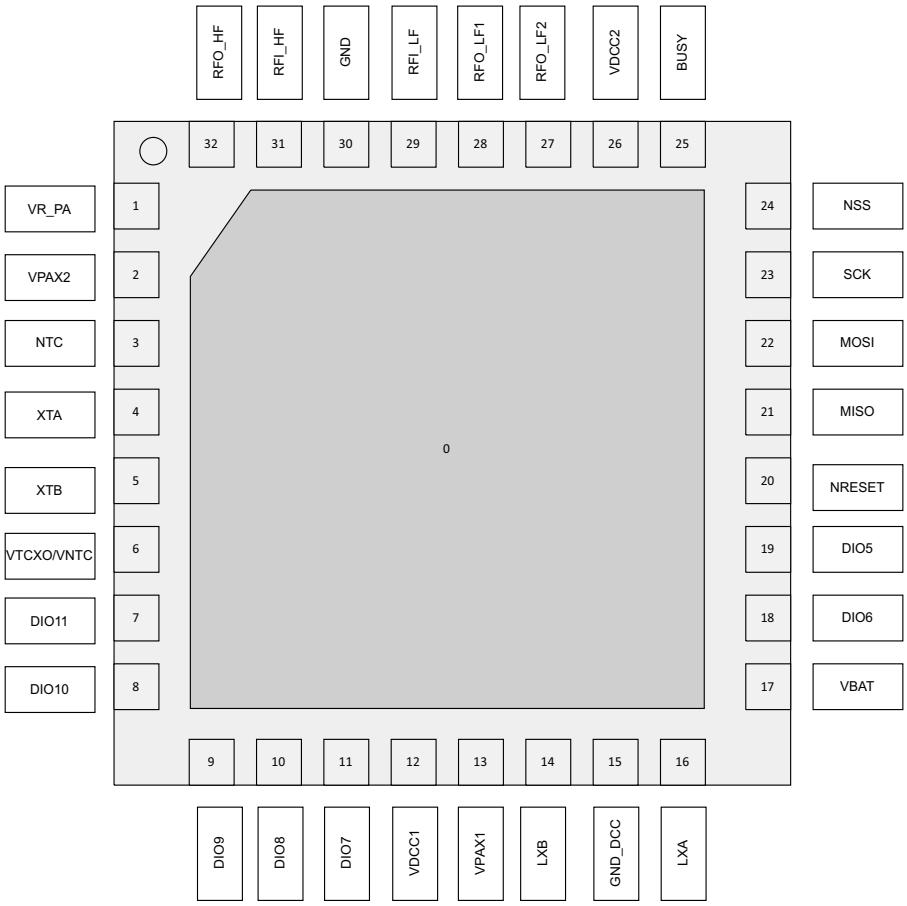
Pin	Device LR2021/22	Device LR2012	Description
0	GND	GND	Exposed Ground pad
1	VR_PA	VR_PA	Regulated power supply for all Power Amplifiers
2	VPAX2	VPAX2	Regulated supply for Power Amplifier regulator
3	NTC	NTC	Negative Temperature Coefficient (NTC) resistor connection
4	XTA	XTA	32MHz crystal oscillator connection; can be used to input external reference clock
5	XTB	XTB	32MHz 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	DI010	DI010	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

Downloaded from Arrow.com.

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

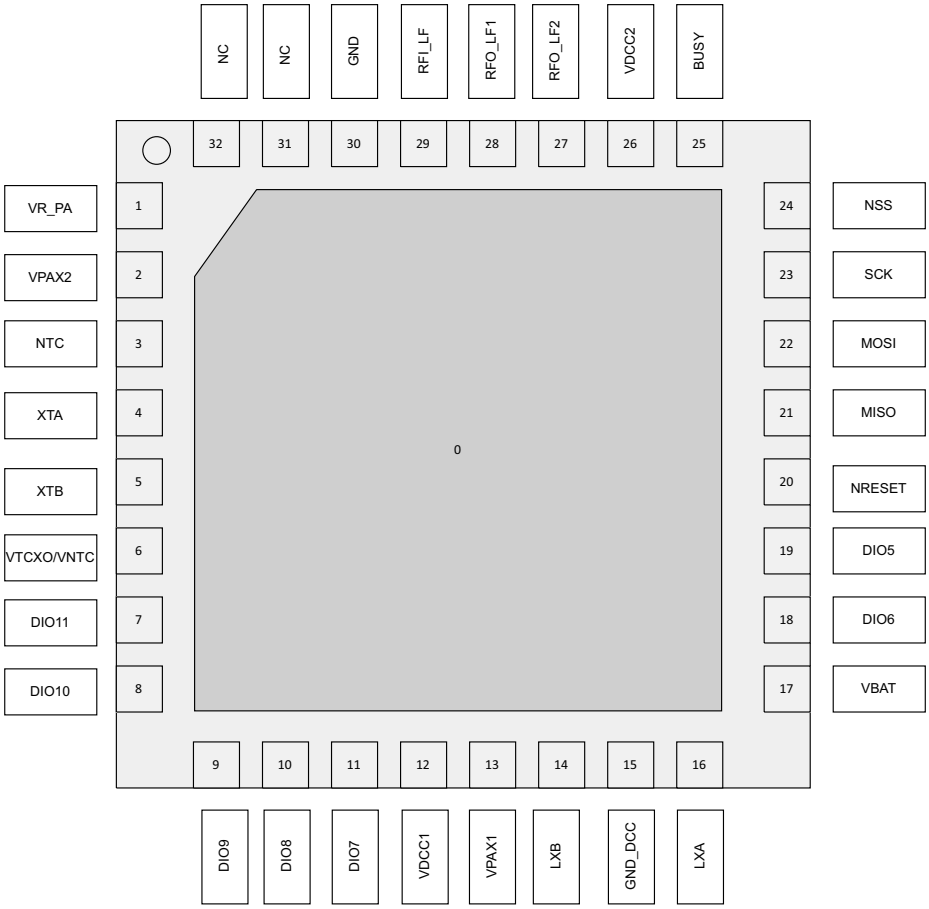
Pin	Device LR2021/22	Device LR2012	Description
25	BUSY	BUSY	Busy indicator signal
26	VDCC2	VDCC2	Regulated supply input
27	RFO_LF2	RFO_LF2	RF transmitter output for LF PA
28	RFO_LF1	RFO_LF1	RF transmitter output for LF PA
29	RFI_LF	RFI_LF	RF LF receiver input
30	GND	GND	Ground
31	RFI_HF	NC	RF HF receiver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



Downloaded from Arrow.com.

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



Downloaded from Arrow.com.

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[1]	dBm
VSWR	Voltage Standing Wave Ratio to guarantee PA stability. Note that any mismatch causes power loss	-	-	10:1	-

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

Downloaded from Arrow.com.

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 JEDEC JS-001, Class 2 HBM	-	-	2.0	kV
ESD_CDM	ESD Charged Device Model JEDEC ESDA JS-002, Class 2a	-	-	500	V
LU	Latch-up 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 2.4GHz ISM band and licensed L/S bands are applicable only to LR2021/LR2022

- VDD = 3.3V as supplied on VBAT, Temperature = 25 °C, FXOSC = 32MHz.
- 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 = 915/869MHz for sub-GHz path.
- FRF = 2.45GHz for HF path. (LR2021/LR2022)
- All RF impedances matched using multi-band 868-915MHz reference design for LF and dedicated 2.4GHz reference design.
- Transmit mode output power defined in 50Ω 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 23.1* 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 = 4.8kbps, FDA = 5kHz, BWF = 20kHz.
- 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 3dB, blocking tests are performed with unmodulated interferer.
- All power consumption numbers were taken with an XTAL, and indifferently at FRF = 915MHz or 2.4GHz, unless specified otherwise.

Downloaded from Arrow.com.

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(32MHz) ON	-	1.21	-	mA
IDDSBX	Supply current in STDBY_XOSC	HF XOSC ON	-	1.79	-	mA
IDDFSDC	Supply current in Synthesizer mode	Frequency at 915MHz	-	3	-	mA

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

Symbol	Description	Conditions	Min	Typ	Max	Unit
IDDTXLF1	868/915MHz LF PA	+22dBm		105		mA
IDDTXLF2		+20dBm	-	72	-	mA
IDDTXLF3		+17dBm	-	43	-	mA
IDDTXLF4		+16dBm	-	36	-	mA
IDDTXLF5		+14dBm	-	27	-	mA
IDDTXLF6		+13dBm	-	24	-	mA
IDDTXLF7		+10dBm	-	17	-	mA
IDDTXLF8		+10dBm[1]	-	12	-	mA
IDDTX8	434/490MHz LF PA	+22dBm		100	-	mA
IDDTX9		+20dBm	-	72	-	mA
IDDTX10		+17dBm	-	42	-	mA
IDDTX11		+16dBm	-	35	-	mA
IDDTX12		+14dBm	-	26	-	mA
IDDTX13		+13dBm	-	24	-	mA
IDDTXHF	2.4GHz, HF PA (LR2021/LR2022)	+12dBm		23		mA

1. For 10dBm, only PA_HF is used with dedicated matching.

Downloaded from Arrow.com.

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

Symbol	Description	Conditions	Min	Typ	Max	Unit
IDDRXLFFD1	Supply current in Receive mode	BRF = 0.6kbps, FDA = 0.8kHz, BWF = 4kHz	-	5.5	-	mA
IDDRXLFFD2		BRF = 1.2kbps, FDA = 5kHz, BWF = 20kHz	-	5.5	-	mA
IDDRXLFFD3		BRF = 4.8kbps, FDA = 5kHz, BWF = 20kHz	-	5.6	-	mA
IDDRXLFFD4		BRF = 38.4kbps, FDA = 40kHz, BWF = 160kHz	-	5.7	-	mA
IDDRXLFFD5		BRF = 50kbps, FDA = 12.5kHz, BWF = 100kHz	-	5.6	-	mA
IDDRXLFFD6		BRF = 100kbps, FDA = 50kHz, BWF = 200kHz	-	5.7	-	mA
IDDRXLFFD7		BRF = 250kbps, FDA = 125kHz, BWF = 500kHz	-	5.9	-	mA
IDDRXLFD2		MSK 2Mbps	-	7.9	-	mA
IDDRXLFFLD1		FLRC 260kbps	-	6.6	-	mA
IDDRXLFFLD2		FLRC 2600kbps	-	10.4	-	mA
IDDRXLFL4		LoRa SF7 125kHz	-	5.7	-	mA
IDDRXLFL5		LoRa SF7 250kHz	-	5.7	-	mA
IDDRXLFL8		LoRa SF7 406kHz	-	6	-	mA
IDDRXLFL6		LoRa SF7 500kHz	-	5.9	-	mA
IDDRXLFL9		LoRa SF7 812kHz	-	6.9	-	mA
IDDRXLFL7		LoRa SF7 1000kHz	-	6.5	-	mA
IDDDELTA1	Additional consumption of Rx boosted mode LF path	Boosted LF vs default LF	-	1.2	-	mA

Downloaded from Arrow.com.

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

Symbol	Description	Conditions	Min	Typ	Max	Unit
IDDRXHFD1	Supply current in Receive mode FRF=2.4GHz	MSK 2Mbps	-	9.2	-	mA
IDDRXHFD2		MSK 1Mbps	-	7.6	-	mA
IDDRXHFFLD1		FLRC 260kbps	-	7.3	-	mA
IDDRXHFFLD2		FLRC 2600kbps	-	10.9	-	mA
IDDRXHFL4		LoRa SF7 125kHz	-	6.4	-	mA
IDDRXHFL5		LoRa SF7 250kHz	-	6.4	-	mA
IDDRXHFL8		LoRa SF7 406kHz	-	6.8	-	mA
IDDRXHFL6		LoRa SF7 500kHz	-	6.6	-	mA
IDDRXHFL9		LoRa SF7 812kHz	-	7.5	-	mA
IDDRXHFL7		LoRa SF7 1000kHz	-	7.1	-	mA
IDDRXHFL10		O-QPSK 250kbps		10.5		mA
IDDDELTA2	Boosted mode Additional current in HF	Boosted HF vs default HF		0.4		mA

Downloaded from Arrow.com.

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^{25}	-	0.95	-	Hz
PHNLF1	Synthesizer phase noise[1] sub-GHz band 915MHz	100kHz offset		-104	-	dBc/Hz
PHNLF2		300kHz offset	-	-112.5	-	dBc/Hz
PHNLF3		1MHz offset	-	-121	-	dBc/Hz
PHNLF4		10MHz offset	-	-133.5	-	dBc/Hz
PHNLF5	Synthesizer phase noise[1] sub-GHz band 490MHz	100kHz offset	-	-108	-	dBc/Hz
PHNLF6		300kHz offset	-	-117	-	dBc/Hz
PHNLF7		1MHz offset	-	-126	-	dBc/Hz
PHNLF8		10MHz offset	-	-135	-	dBc/Hz
PHNHF1	Synthesizer phase noise[1] 2.45GHz (LR2021/LR2022)	100kHz offset	-	-93	-	dBc/Hz
PHNHF2		300kHz offset		-102.5	-	dBc/Hz
PHNHF3		1MHz offset	-	-113.5	-	dBc/Hz
PHNHF4		10MHz 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 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 OSCTRM2	HF Crystal oscillator trimming step	Normal	-	1	-	ppm
LFCLKRF	Rise/fall time for external LF clock		-	-	5	ns
BRFSK1	Bit rate, FSK	For frequency range 433MHz - 2.5GHz	0.5	-	2000	kbps
BRFSK2	Programmable	For frequency range 150MHz - 433MHz	0.5	-	1000	kbps
FDAFSK1	Frequency deviation, FSK	For frequency range 433MHz - 2.5GHz	0.6	-	500	kHz
FDAFSK2	Programmable	For frequency range 150MHz - 433MHz	0.6	-	330	kHz
BRFLRC	Effective bit rate	For frequency range 150MHz - 470MHz	0.13	-	1.3	Mbps
		For frequency range 470MHz - 2.5GHz	0.13	-	2.6	Mbps
MODIND1	Modulation index	GFSK	0.3	-	6.2	-
MODIND2	Programmable	FSK	0.3		8.3	

Downloaded from Arrow.com.

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 BRLORA2 BRLORA3 BRLORA4	Bit rate, LoRa	SF12, BWLORA = 31.25kHz, CR = 4/8 SF5, BWLORA = 500kHz, CR = 4/5 SF5, BWLORA = 1000kHz, CR = 4/5 SF12, BWLORA = 125kHz, CR = 4/5 SF5, BWLORA = 812kHz, CR = 4/5	- - - - -	45.8 62.5 125 292 101.5	- - - - -	bps kbps kbps bps kbps
BWLORA1 BWLORA2	Signal BW, LoRa	For frequency range 434MHz - 2500MHz For frequency range 150MHz - 433MHz	7.825 7.825	- -	1000 500	kHz 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 LF		0.5 1		dB dB

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

Downloaded from Arrow.com.

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 = 2.4kbps, FDA = 2.4kHz		-124		dBm
WM_N2P4_FREQERR	Max frequency error tolerated	FRF = 169MHz		+/- 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 + 3dB		-30		dBm
WM_N2P4_B_2M	Blocker +/- 2MHz	Wanted at sensi + 3dB		-42		dBm
WM_N2P4_B_10M	Blocker +/- 10MHz	Wanted at sensi + 3dB		-32		dBm
WM_N4P8_S	Sensitivity	BR = 4.8kbps, FDA = 2.4kHz		-124		dBm
WM_N4P8_FREQERR	Max frequency error tolerated	FRF = 169MHz		+/- 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 + 3dB		-29		dBm
WM_N4P8_B_2M	Blocker +/- 2MHz	Wanted at sensi + 3dB		-42		dBm
WM_N4P8_B_10M	Blocker +/- 10MHz	Wanted at sensi + 3dB		-32		dBm
WM_F2P4_S	Sensitivity	BR = 2.4kbps, FDA = 5.5kHz		-123		dBm
WM_F2P4_FREQERR	Max frequency error tolerated	FRF = 434MHz		+/-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 + 3dB		-21		dBm
WM_F2P4_B_2M	Blocker +/- 2MHz	Wanted at sensi + 3dB		-42		dBm
WM_F2P4_B_10M	Blocker +/- 10MHz	Wanted at sensi + 3dB		-27		dBm
WM_C2O2M50_S	Sensitivity	BR = 50kbps, FDA = 25kHz		-113		dBm
WM_C2O2M50_FREQERR	Max frequency error tolerated	FRF = 868MHz		+/-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 + 3dB		-21		dBm

Downloaded from Arrow.com.

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

Symbol	Description	Conditions	Min	Typ	Max	Unit
WM_C2O2M50_B_2M	Blocker +/- 2MHz	Wanted at sensi + 3dB		-40		dBm
WM_C2O2M50_B_10M	Blocker +/- 10MHz	Wanted at sensi + 3dB		-24		dBm
WM_C1C2M2O100_S	Sensitivity	BR = 100kbps, FDA = 45kHz		-111		dBm
WM_C1C2M2O100_FREQERR	Max frequency error tolerated	FRF = 868MHz		+/-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 + 3dB		-19		dBm
WM_C1C2M2O100_B_2M	Blocker +/- 2MHz	Wanted at sensi + 3dB		-40		dBm
WM_C1C2M2O100_B_10M	Blocker +/- 10MHz	Wanted at sensi + 3dB		-27		dBm
WM_R2_4P8_S	Sensitivity	CR = 4.8kcps, FDA = 6kHz		-122		dBm
WM_R2_4P8_FREQERR	Max frequency error tolerated	FRF = 868MHz		+/-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 + 3dB		-15		dBm
WM_R2_4P8_B_2M	Blocker +/- 2MHz	Wanted at sensi + 3dB		-38		dBm
WM_R2_4P8_B_10M	Blocker +/- 10MHz	Wanted at sensi + 3dB		-21		dBm
WM_T2O2M32_S	Sensitivity	CR = 32.768kcps, FDA = 50kHz		-113		dBm
WM_T2O2M32_FREQERR	Max frequency error tolerated	FRF = 868MHz		+/-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 + 3dB		-19		dBm
WM_T2O2M32_B_2M	Blocker +/- 2MHz	Wanted at sensi + 3dB		-42		dBm
WM_T2O2M32_B_10M	Blocker +/- 10MHz	Wanted at sensi + 3dB		-24		dBm
WM_T1T2M2O100_S	Sensitivity	CR = 100kcps, FDA = 50kHz		-110		dBm
WM_T1T2M2O100_FREQERR	Max frequency error tolerated	FRF = 868MHz		+/-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 + 3dB		-19		dBm
WM_T1T2M2O100_B_2M	Blocker +/- 2MHz	Wanted at sensi + 3dB		-41		dBm

Downloaded from Arrow.com.

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

Symbol	Description	Conditions	Min	Typ	Max	Unit
WM_T1T2M2O100_B_10M	Blocker +/- 10MHz	Wanted at sensi + 3dB		-28		dBm
WM_S1S232_S	Sensitivity	CR = 32.768kcps, FDA = 50kHz		-113		dBm
WM_S1S232_FREQERR	Max frequency error tolerated	FRF = 868MHz		+/-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 + 3dB		-21		dBm
WM_S1S232_B_2M	Blocker +/- 2MHz	Wanted at sensi + 3dB		-37		dBm
WM_S1S232_B_10M	Blocker +/- 10MHz	Wanted at sensi + 3dB		-23		dBm

Downloaded from Arrow.com.

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

Downloaded from Arrow.com.

The following table provides the receiver specification for Wi-SUN FSK @10% PER 250 bytes, FRF = 915MHz. 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	SR = 50kbps, FDA = 12.5kHz, no FEC	-	-113.5	-	dBm
WS_1a_ACRM	Lower adjacent channel rejection			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	SR = 50kbps, FDA = 25kHz, no FEC	-	-110.5	-	dBm
WS_1b_ACRM	Lower adjacent channel rejection			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	SR = 100kbps, FDA = 25kHz, no FEC	-	-110	-	dBm
WS_2a_ACRM	Lower adjacent channel rejection			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	SR = 100kbps, FDA = 50kHz, no FEC	-	-108	-	dBm
WS_2b_ACRM	Lower adjacent channel rejection			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	SR = 150kbps, FDA = 37.5kHz, no FEC	-	-108	-	dBm
WS_3_ACRM	Lower adjacent channel rejection			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

Downloaded from Arrow.com.

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	SR = 200kbps, FDA = 50kHz, no FEC		-107.5		dBm
WS_4a_ACRM	Lower adjacent channel rejection			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	SR = 200kbps, FDA = 100kHz, no FEC		-104.5		dBm
WS_4b_ACRM	Lower adjacent channel rejection			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	SR = 300kbps, FDA = 75kHz, no FEC		-106		dBm
WS_5_ACRM	Lower adjacent channel rejection			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, 2b,3, 4, 4b,5 FRF = 915MHz		+/-30		%
WS_BRERR	Bit rate error			+/-100		ppm
WS_FERR	Max frequency error tolerated			+-20		ppm

Downloaded from Arrow.com.

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

Table 3-12: FLRC Sub-GHz 1% PER (LR2021) (Sheet 1 of 2)

Symbol	Description	Conditions	Min	Typ	Max	Unit
FLRC_2600_CR05_915_S	Sensitivity 1% PER 915MHz	BRF = 2600kbps, CR = 3/4, BWF = 2666kHz Effective bit rate= 1.95Mbps, G13 rx_boost=7		-100.5		dBm
FLRC_2600_CR05_FERR	Max frequency error tolerated	FRF = 915MHz		+/-150		kHz
FLRC_2600_CR05_PRE	Min preamble length			24		bits
FLRC_2080_CR05_915_S	Sensitivity 1% PER 915MHz	BRF = 2080kbps, CR = 3/4, BWF = 2666kHz Effective bit rate= 1.56Mbps, G13 rx_boost=7		-101.5		dBm
FLRC_2080_CR05_FERR	Max frequency error tolerated	FRF = 915MHz		+/-150		kHz
FLRC_2080_CR05_PRE	Min preamble length			20		bits
FLRC_1300_CR05_915_S	Sensitivity 1% PER 915MHz	BRF = 1300kbps, CR = 3/4, BWF = 1200kHz Effective bit rate= 975kbps, G13 rx_boost=7		-104.5		dBm
FLRC_1300_CR05_FERR	Max frequency error tolerated	FRF = 915MHz		+/-100		kHz
FLRC_1300_CR05_PRE	Min preamble length			16		bits
FLRC_1040_CR05_915_S	Sensitivity 1% PER 915MHz	BRF = 1040kbps, CR = 3/4, BWF = 1200kHz Effective bit rate= 780kbps, G13 rx_boost=7		-105		dBm
FLRC_1040_CR05_FERR	Max frequency error tolerated	FRF = 915MHz		+/-100		kHz
FLRC_1040_CR05_PRE	Min preamble length			16		bits
FLRC_650_CR05_915_S	Sensitivity 1% PER 915MHz	BRF = 650kbps, CR = 3/4, BWF = 600kHz Effective bit rate= 487kbps, G13 rx_boost=7		-107		dBm
FLRC_650_CR05_FERR	Max frequency error tolerated	FRF = 915MHz		+/-70		kHz
FLRC_650_CR05_PRE	Min preamble length			16		bits
FLRC_520_CR05_915_S	Sensitivity 1% PER 915MHz	BRF = 520kbps, CR = 3/4, BWF = 600kHz Effective bit rate= 390kbps, G13 rx_boost=7		-108.5		dBm
FLRC_520_CR05_FERR	Max frequency error tolerated	FRF = 915MHz		+/-50		kHz
FLRC_520_CR05_PRE	Min preamble length			16		bits

Downloaded from Arrow.com.

Table 3-12: FLRC Sub-GHz 1% PER (LR2021) (Sheet 2 of 2)

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

Downloaded from Arrow.com.

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

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

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

Downloaded from Arrow.com.

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

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

Downloaded from Arrow.com.

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	250kbps 100kHz Bw555, G13 rx_boost=7		-106		dBm
FSK_250_FERR	Max frequency error tolerated	FRF = 915MHz		+/- 60		ppm
FSK_100_S	Sensitivity	100kbps 50kHz Bw333, G13 rx_boost=7		-110.5		dBm
FSK_100_FERR	Max frequency error tolerated	FRF = 915MHz		+/- 60		ppm
FSK_50_S	Sensitivity	50kbps 12.5kHz Bw138, G13 rx_boost=7		-116		dBm
FSK_50_LW	Sensitivity	50kbps 25kHz Bw138, in LoRaWAN conditions, G13 rx_boost=7		-113		dBm
FSK_50_FERR	Max frequency error tolerated	FRF = 915MHz		+/- 60		ppm
FSK_38P4_S	Sensitivity	38.4kbps 40kHz Bw185, G13 rx_boost=7		-113		dBm
FSK_38P4_FERR	Max frequency error tolerated	FRF = 915MHz		+/- 60		ppm
FSK_4P8_S	Sensitivity	4.8kbps 5kHz Bw27, G13 rx_boost=7		-121.5		dBm
FSK_4P8_FERR	Max frequency error tolerated	FRF = 915MHz		+/- 2.5		kHz
FSK_1P2_S	Sensitivity	1.2kbps 5kHz Bw17, G13 rx_boost=7		-124.5		dBm
RXSIG	Sigfox sensitivity	BRF= 0.6kbs, FDA = 0.8kHz, BWF = 4kHz G13 rx_boost=7		-128.5		dBm
FSK_1P2_FERR	Max frequency error tolerated	FRF = 915MHz		+/- 2.5		kHz
FSK_0P600_S	Sensitivity	FRF = 868MHz		-126		dBm
FSK_ACR	Adjacent channel rejection, FSK	Offset = max of +/- 1.5 BW or +/- 50kHz. 4.8kbps 5kHz Bw27	-	50	-	dB
FSK_BI1_3	Blocking immunity	Offset = +/- 1MHz, 4.8kbps 5kHz Bw27, sensitivity + 3dB	-	65	-	dB
FSK_BI2_3	Blocking immunity	Offset = +/- 2MHz, 4.8kbps 5kHz Bw27, sensitivity + 3dB	-	70	-	dB
FSK_BI10_3	Blocking immunity	Offset = +/- 10MHz, 4.8kbps 5kHz Bw27, sensitivity + 3dB	-	79	-	dB
FSK_B1_10	Blocking immunity	Offset = +/- 1MHz, 4.8kbps 5kHz Bw27, sensitivity + 10dB	-	63	-	dB
FSK_BI2_10	Blocking immunity	Offset = +/- 2MHz, 4.8kbps 5kHz Bw27, sensitivity + 10dB	-	65	-	dB
FSK_BI10_10	Blocking immunity	Offset = +/- 10MHz, 4.8kbps 5kHz Bw27, sensitivity + 10dB	-	80	-	dB
FSK_B1_30	Blocking immunity	Offset = +/- 1MHz, 4.8kbps 5kHz Bw27, sensitivity + 30dB	-	62	-	dB

Downloaded from Arrow.com.

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 = +/- 2MHz, 4.8kbps 5kHz Bw27, sensitivity + 30dB	-	62	-	dB
FSK_BI10_30	Blocking immunity	Offset = +/- 10MHz, 4.8kbps 5kHz Bw27, sensitivity + 30dB	-	72	-	dB
FSK_B1_50	Blocking immunity	Offset = +/- 1MHz, 4.8kbps 5kHz Bw27, sensitivity + 50dB	-	65	-	dB
FSK_BI2_50	Blocking immunity	Offset = +/- 2MHz, 4.8kbps 5kHz Bw27, sensitivity + 50dB	-	65	-	dB
FSK_BI10_50	Blocking immunity	Offset = +/- 10MHz, 4.8kbps 5kHz Bw27, sensitivity + 50dB	-	75	-	dB

Downloaded from Arrow.com.

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

Downloaded from Arrow.com.

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

Downloaded from Arrow.com.

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@2.4GHz, 250kbps			-103		dBm
OQPSK_250_ACR	Adjacent interference	Signal @sensitivity + 3dB, RF@2.4GHz		27		dB
OQPSK_250_ALT	Alternate Adjacent interference	Signal @sensitivity + 3dB, RF@2.4GHz		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 = 1000kHz, SF = 5, CR 4/5	-	-112.5	-	dBm
LORA_SUB_1000_SF6		BWLORA = 1000kHz, SF = 6, CR 4/5	-	-115.5	-	dBm
LORA_SUB_1000_SF7		BWLORA = 1000kHz, SF = 7, CR 4/5	-	-118	-	dBm
LORA_SUB_1000_SF8		BWLORA = 1000kHz, SF = 8, CR 4/5	-	-120	-	dBm
LORA_SUB_1000_SF9		BWLORA = 1000kHz, SF = 9, CR 4/5	-	-123	-	dBm
LORA_SUB_1000_SF10		BWLORA = 1000kHz, SF = 10, CR 4/5	-	-126	-	dBm
LORA_SUB_1000_SF11		BWLORA = 1000kHz, SF = 11, CR 4/5	-	-129	-	dBm
LORA_SUB_1000_SF12		BWLORA = 1000kHz, SF = 12, CR 4/5	-	-131	-	dBm
LORA_SUB_800_SF5	Sensitivity LoRa	BWLORA = 800kHz, SF = 5, CR 4/5	-	-113	-	dBm
LORA_SUB_800_SF6		BWLORA = 800kHz, SF = 6, CR 4/5	-	-115.5	-	dBm
LORA_SUB_800_SF7		BWLORA = 800kHz, SF = 7, CR 4/5	-	-118	-	dBm
LORA_SUB_800_SF8		BWLORA = 800kHz, SF = 8, CR 4/5	-	-121	-	dBm
LORA_SUB_800_SF9		BWLORA = 800kHz, SF = 9, CR 4/5	-	-124	-	dBm
LORA_SUB_800_SF10		BWLORA = 800kHz, SF = 10, CR 4/5	-	-126.5	-	dBm
LORA_SUB_800_SF11		BWLORA = 800kHz, SF = 11, CR 4/5	-	-130	-	dBm
LORA_SUB_800_SF12		BWLORA = 800kHz, SF = 12, CR 4/5	-	-132	-	dBm

Downloaded from Arrow.com.

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

Downloaded from Arrow.com.

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

Downloaded from Arrow.com.

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 = 31kHz, SF = 5, CR 4/5	-	-128	-	dBm
LORA_SUB_31_SF6		BWLORA = 31kHz, SF = 6, CR 4/5	-	-130.5	-	dBm
LORA_SUB_31_SF7		BWLORA = 31kHz, SF = 7, CR 4/5	-	-133	-	dBm
LORA_SUB_31_SF8		BWLORA = 31kHz, SF = 8, CR 4/5	-	-135.5	-	dBm
LORA_SUB_31_SF9		BWLORA = 31kHz, SF = 9, CR 4/5	-	-138	-	dBm
LORA_SUB_31_SF10		BWLORA = 31kHz, SF = 10, CR 4/5	-	-142	-	dBm
LORA_SUB_31_SF11		BWLORA = 31kHz, SF = 11, CR 4/5	-	-144.5	-	dBm
LORA_SUB_31_SF12		BWLORA = 31kHz, 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 Offset = +/- 1.5 x BWLORA	BWLORA = 125kHz, SF = 7	-	61	-	dB
LORA_ACR12		BWLORA = 125kHz, SF = 12	-	75	-	dB
LORA_BI_SF12_1M	Blocking immunity, LoRa, BWLORA = 125kHz	SF =12, offset = +/- 1MHz	-	79	-	dB
LORA_BI_SF12_2M		SF =12, offset = +/- 2MHz	-	89	-	dB
LORA_BI_SF12_10M		SF =12, offset = +/- 10MHz	-	100	-	dB
LORA_BI_SF7_1M		SF =7, offset = +/- 1MHz	-	64	-	dB
LORA_BI_SF7_2M		SF =7, offset = +/- 2MHz	-	76	-	dB
LORA_BI_SF7_10M		SF =7, offset = +/- 10MHz	-	86	-	dB
LORA_FDRIFT1	LoRa frequency drift tolerance For sensitivity degradation below 2dB	BWLORA = 125kHz, SF10, LowDataRateOptimize = 0	-	-	200	Hz/s
LORA_FDRIFT2	LoRa frequency drift tolerance	BWLORA = 125kHz, 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 3dB degradation	All bandwidths	-33		33	%BW

Downloaded from Arrow.com.

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 1.5dB degradation. 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 1.5dB 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

Downloaded from Arrow.com.

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

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

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

Downloaded from Arrow.com.

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

Symbol	Description	Conditions	Min	Typ	Max	Unit
LORA_2G4_200_SF5	Sensitivity LoRa	BWLORA = 200kHz, SF = 5, CR 4/5	-	-117	-	dBm
LORA_2G4_200_SF6		BWLORA = 200kHz, SF = 6, CR 4/5	-	-120	-	dBm
LORA_2G4_200_SF7		BWLORA = 200kHz, SF = 7, CR 4/5	-	-123	-	dBm
LORA_2G4_200_SF8		BWLORA = 200kHz, SF = 8, CR 4/5	-	-125.5	-	dBm
LORA_2G4_200_SF9		BWLORA = 200kHz, SF = 9, CR 4/5	-	-128.5	-	dBm
LORA_2G4_200_SF10		BWLORA = 200kHz, SF = 10, CR 4/5	-	-131	-	dBm
LORA_2G4_200_SF11		BWLORA = 200kHz, SF = 11, CR 4/5	-	-134	-	dBm
LORA_2G4_200_SF12		BWLORA = 200kHz, 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 Offset = +/- 1.5 x BWLORA	BWLORA = 800kHz, SF = 7	-	53	-	dB
LORA_2G4_ACR12		BWLORA = 800kHz, SF = 12	-	68	-	dB
LORA_2G4_BI12_2M	Blocking immunity LoRa, Offset = +/- 2 MHz	BWL = 802 kHz, SF12	-	68	-	dB
LORA_2G4_BI12_10M	Blocking immunity LoRa, Offset = +/- 10MHz	BWL = 802 kHz, SF12	-	82	-	dB
LORA_2G4_BI7_2M	Blocking immunity LoRa, Offset = +/- 2 MHz	BWL = 802 kHz, SF7	-	54	-	dB
LORA_2G4_BI7_10M	Blocking immunity 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 = 4.8kbps, BW = 24kHz	-	-114	-	dBm
OOK_S3		BR = 32.768kbps, BW = 153kHz	-	-105	-	dBm
OOK_S1		BR = 1.562kbps, BW = 7.4kHz	-	-120	-	dBm
OOK_FREQ_RG_1	Max tolerated frequency offset between transmitter and receiver for 2dB sensitivity degradation[1]	BR = 1.562kbps, BW = 7.4kHz	-26		26	kHz

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

Downloaded from Arrow.com.

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 @1MHz and 1.96MHz	-	-12	-	dBm
HF_N_IIP3	3rd order input intercept point HF path	Unwanted tones @1MHz and 1.96MHz	-	-12	-	dBm
LF_W_IIP3	3rd order input intercept point LF path	Unwanted tones @20MHz and 39.4MHz	-	-10	-	dBm
HF_W_IIP3	3rd order input intercept point HF path	Unwanted tones @20MHz and 39.4MHz	-	-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 < 500kHz	-	45	-	dB
IMR_HF_UC	Image attenuation, 2.4GHz path	(LR2021/LR2022) Without image calibration on high frequency path	-	25	-	dB
IMR_HF_C	Image attenuation, 2.4GHz path after calibration	(LR2021/LR2022) With image calibration BW < 500kHz 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

Downloaded from Arrow.com.

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)

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 3.7V)		-	2	-	dB
OCPLF	Over current protection trigger value	Active when PA_LF is used	-	164	-	mA
TXPRNGLF	Tx power range	Programmable in steps of 0.5dB 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	@22dBm	-	40	-	dB
OOKOOLF2		@12dBm	-	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
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 0.5dB from maximum Tx power	-	63	-	steps
TXRMPHF1	Power amplifier ramping time	Programmable, lowest setting	-	2	-	µs
TXRMPHF2		Programmable, highest setting	-	340	-	µs
OOKOOHF1	OOK ON OFF ratio	@12dBm	-	30	-	dB

Downloaded from Arrow.com.

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)

Symbol	Description	Timing Precisions	Min	Typ	Max	Unit
TSHFOSC	HF Crystal oscillator wake-up time	From STDBY_RC[1]	-	70[2]	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 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
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

Downloaded from Arrow.com.

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:

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

Table 3-24: 32MHz 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
C0XTALHF	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, 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 10pF DC-cut series with 220Ω	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: 2.2µH

DCR (max): 0.5Ω

Isat (min): 200mA

Resonant Frequency (min): 20MHz

Downloaded from Arrow.com.

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

3.7.1 Digital I/O Specifications

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

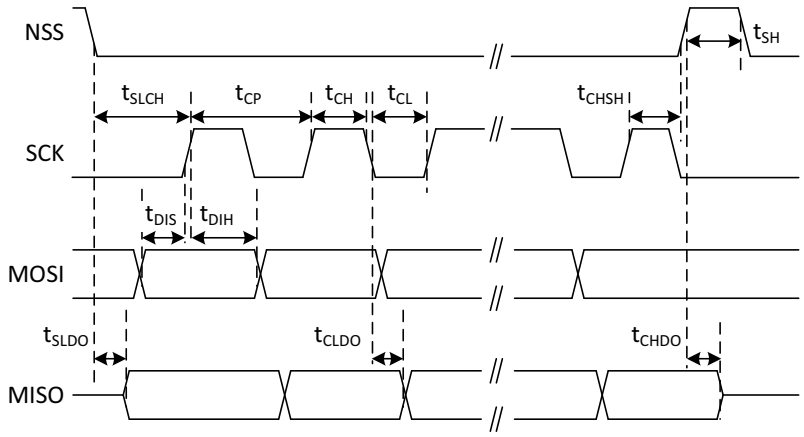
Symbol	Description	Condition	Min	Typ	Max	Unit
VIH	Input High Voltage		0.7*VBAT	-	VBAT+0.3	V
VIL	Input Low Voltage		-0.3		0.3*VBAT	V
VOH	Output High Voltage	I_{max} = 2.5mA	0.9*VBAT		VBAT	V
VOL	Output Low Voltage	I_{max}=-2.5mA	0		0.1*VBAT	V
ZPUD	Pull-up and Pull-down Resistor	Any DIO pin when pulled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

Downloaded from Arrow.com.

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 16MHz.

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	Maximum load capacitance of 10pF	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		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

Downloaded from Arrow.com.

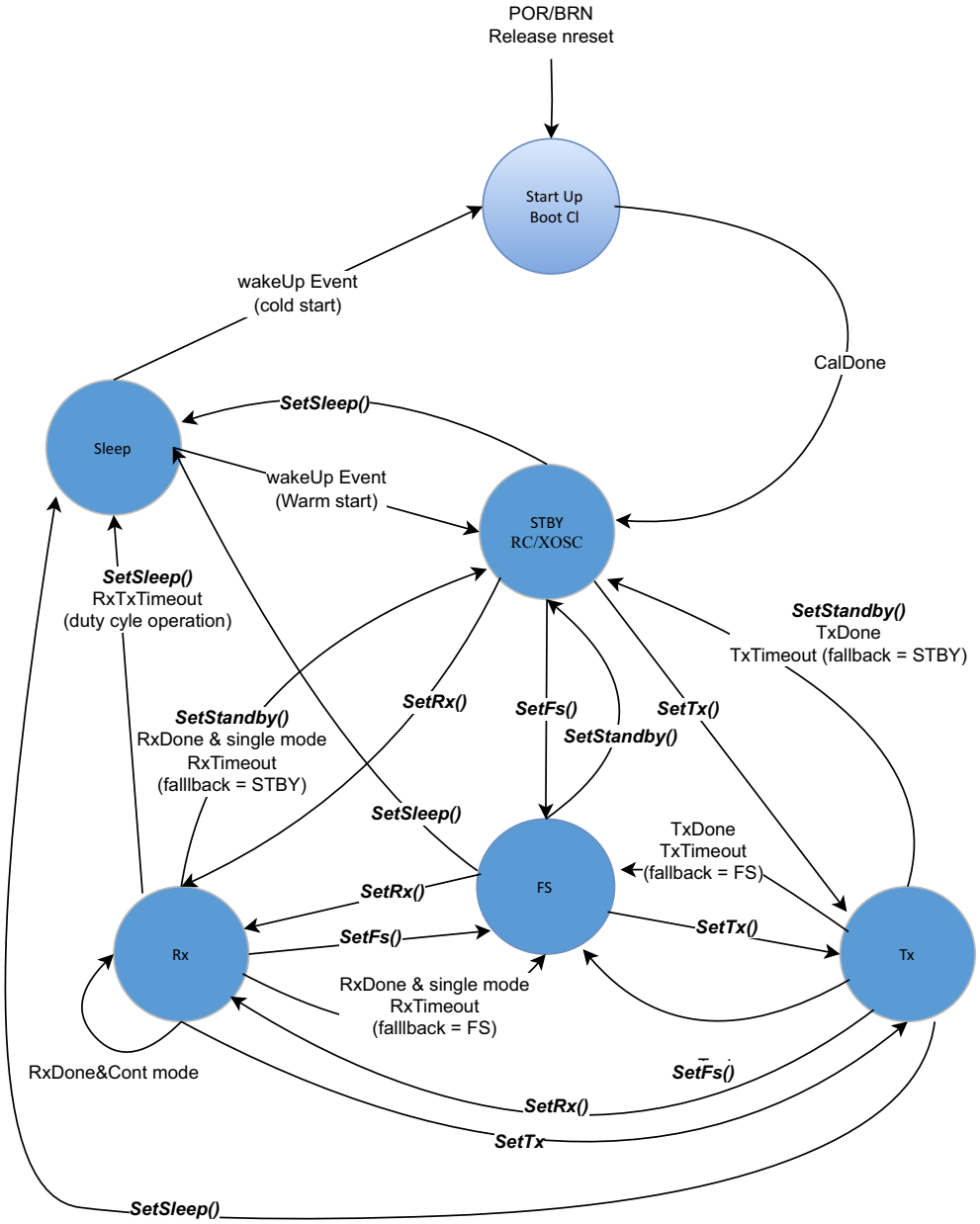
4. Chip Modes (LR20xx)

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 32kHz oscillator and XOSC refers to the 32MHz precision clock from XTAL oscillator or TCXO.

Figure 4-1: Chip Modes



Downloaded from Arrow.com.

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[1] 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*1*	Lowest power mode can be with or without data RAM in retention Can be with or without RTC clock and timer running	After Reset: Standby RC via cold start NSS falling: Standby RC via warm start if data RAM is in retention else Standby RC via cold start RTC count reached if RC clock active: Standby RC via warm start if data RAM is in retention else Standby RC via cold start RTC count reached during an Rx duty cycle operation. Final state is Rx after a warm start.
Standby RC[2]	StandbyRC	Mode for chip configuration	After Reset: Standby RC via cold start Command: all modes
Standby XOSC	StandbyXOSC	Mode for chip configuration	After Reset: Standby RC via cold start Command: all modes
FS	FS	Frequency synthesiser mode	After Reset: Standby RC via cold start Command: all modes
Rx	Rx	Rx mode for packet reception	After Reset: Standby RC via cold start Command: all modes
Tx	Tx	Tx mode for packet transmission	After Reset: Standby RC via cold start 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.

Downloaded from Arrow.com.

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 100µs.

  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.

Downloaded from Arrow.com.

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, it is essential for the host to 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. It is essential to note that 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 32kHz RC or external 32kHz input) should be configured using *SetRxDutyCycle* command.

Downloaded from Arrow.com.

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 utilized 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.

Downloaded from Arrow.com.

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.

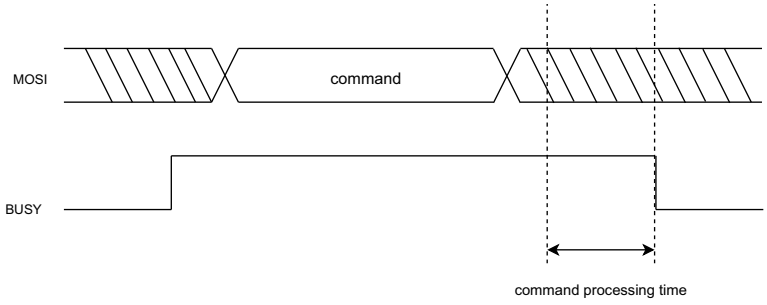
5.1.2 SPI Interface

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.

It is essential to wait for the BUSY line to 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.

Downloaded from Arrow.com.

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, it's important to note that 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 facilitates seamless 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.

It is essential to note that 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.

Downloaded from Arrow.com.

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.

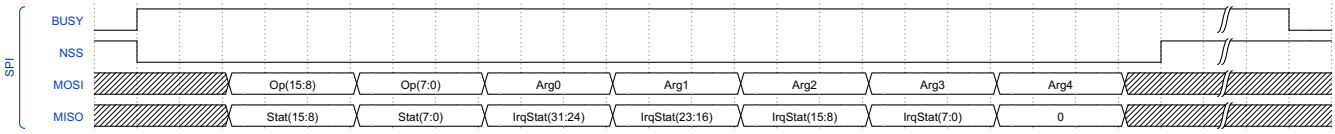
The BUSY signal serves as a handshake mechanism. It indicates whether the LR2021/LR2022/LR2012 is ready to accept a new command from the Host controller. When the BUSY signal is in a high state, it signifies that the LR2021/LR2022/LR2012 is occupied and cannot receive a new command. Conversely, when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

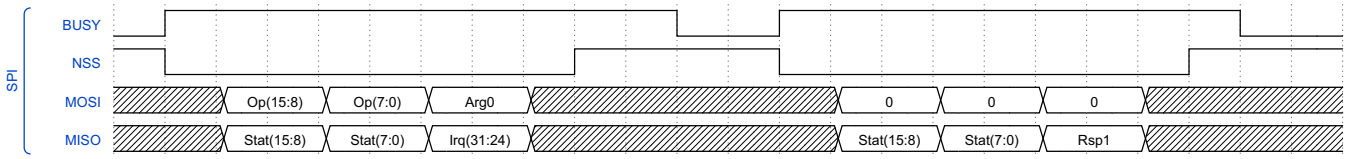
Downloaded from Arrow.com.

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.

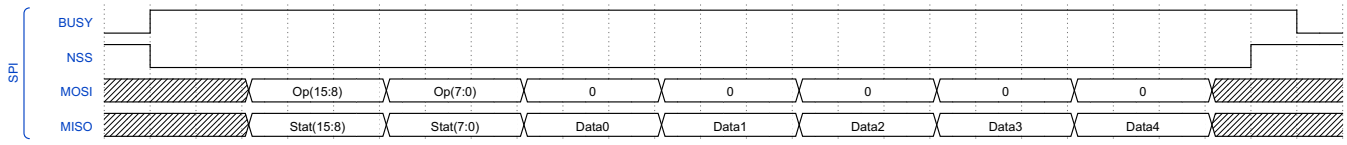
Figure 5-3: Read Command



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.

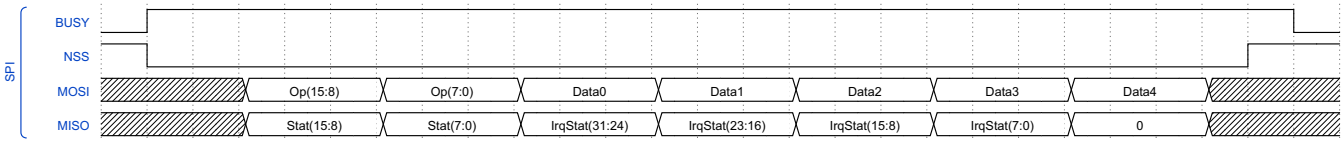
Figure 5-4: Direct Read Timing Diagram



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



Downloaded from Arrow.com.

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

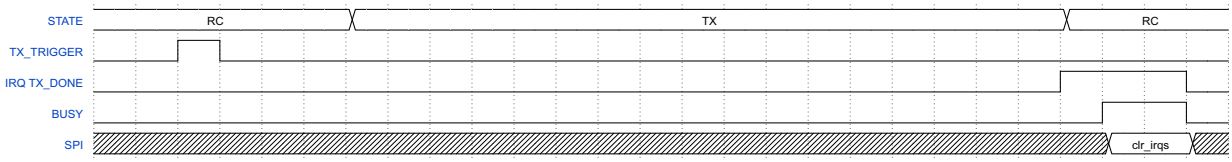
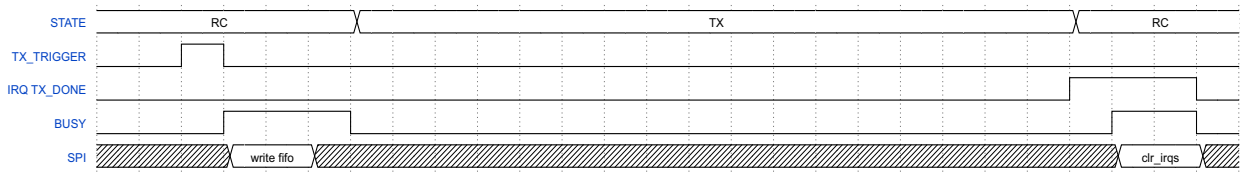


Figure 5-7: Tx Trigger Followed by FIFO Write Example



Downloaded from Arrow.com.

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 5-1: Radio FIFO Memory Commands

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

5.6.2 Register / Memory Access Operations (LR20xx)

Table 5-2: Register Memory Commands

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

Downloaded from Arrow.com.

5.6.3 System Configuration / Status Operations (LR20xx)

Table 5-3: System Configuration Commands (Sheet 1 of 2)

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

Downloaded from Arrow.com.

Table 5-3: System Configuration Commands (Sheet 2 of 2)

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

Downloaded from Arrow.com.

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

5.6.4.1 Common Radio Commands (LR20xx)

Table 5-4: Common Radio Commands (Sheet 1 of 2)

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

Downloaded from Arrow.com.

Table 5-4: Common Radio Commands (Sheet 2 of 2)

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

Downloaded from Arrow.com.

5.6.4.2 LoRa Packet Radio Commands (LR20xx)

Table 5-5: LoRa Packet Radio Commands

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

Downloaded from Arrow.com.

5.6.4.3 FSK Packet Radio Commands (LR20xx)

Table 5-6: FSK Packet Radio Commands

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

Downloaded from Arrow.com.

5.6.4.4 FLRC Packet Radio Commands (LR2021)

Table 5-7: FLRC Packet Radio Commands

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

5.6.4.5 BPSK Packet Radio Commands (LR20xx)

Table 5-8: BPSK Packet Radio Commands

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

Downloaded from Arrow.com.

5.6.4.6 LR-FHSS Packet Radio Commands (LR20xx)

Table 5-9: LR-FHSS Packet Radio Commands

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

Downloaded from Arrow.com.

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

Table 5-10: Bluetooth Low Energy Packet Radio Commands

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

5.6.4.8 WM-Bus Packet Radio Commands (LR20xx)

Table 5-11: WM-Bus Packet Radio Commands

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

Downloaded from Arrow.com.

5.6.4.9 Wi-SUN Packet Radio Commands (LR20xx)

Table 5-12: Wi-SUN Packet Radio Commands

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

Downloaded from Arrow.com.

5.6.4.10 Z-Wave Packet Radio Commands (LR2021)

Table 5-13: Z-Wave Packet Radio Commands

Command	Opcode	Parameters	Description
SetZwaveParams	0x0297	Mode(7:0) rx_bw(7:0) addr_comp(7:0) pld_len(7:0) pbl_len_tx(15:0) pbl_len_detect(7:0) Options(7:0)	Sets all parameters for Z-Wave packets
SetZwaveHomeIdFiltering	0x0298	home_id(31:0)	Sets the HomeID address to use as a filter in Rx
GetZwaveRxStats	0x0299	--	Gets the Rx statistics of Z-Wave packets
GetZwavePacketStatus	0x029A	--	Gets Rx packet status information of Z-Wave packets
SetZwaveBeamFiltering	0x029B	beam_tag(7:0) addr_len node_id(11:0) id_hash(7:0)	Filters incoming beam frames
SetZwaveScanConfig	0x029C	num_ch(3:0), det4, det3, det2, det1, bitrate_ch4(1:0), bitrate_ch3(1:0), bitrate_ch2(1:0), bitrate_ch1(1:0), addr_comp(7:0), options(7:0), rf_freq_ch1(31:0), timeout_ch1(7.0), rf_freq_ch2(31:0), timeout_ch2(7.0), rf_freq_ch3(31:0), timeout_ch3(7.0), rf_freq_ch4(31:0), timeout_ch4(7.0)	Configures the scan mode for the Z-Wave
SetZwaveScan	0x029D	--	Enters Z-Wave Rx scan mode

Downloaded from Arrow.com.

5.6.4.11 OQPSK_15_4 Packet Radio Commands (LR2021)

Table 5-14: OQPSK_15_4 Packet Radio Commands

Command	Opcode	Parameters	Description
SetOqpskParams	0x029F	Mode(7:0) rx_bw(7:0) pld_len(7:0), pbl_len_tx(15:0) AddressOn, FcsMode	Set parameters for OQPSK_15_4 packets
GetOqpskRxStats	0x02A0	--	Get the Rx statistics for OQPSK_15_4 packets
GetOqpskPacketStatus	0x02A1	--	Get the Rx packet status informations for OQPSK_15_4 packets
SetOqpskPacketLen	0x02A2	len(7:0)	Sets new length for OQPSK_15_4 packets
SetOqpskAddress	0x02A3	LongDestAddr(63:0) ShortDestAddr(15:0) PanId(15:0) TransId(7:0)	Set address parameters for OQPSK_15_4 packets

5.6.4.12 RTToF Packet Radio Commands (LR20xx)

Table 5-15: RTToF Packet Radio Commands

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

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

Downloaded from Arrow.com.

5.6.4.13 OOK Packet Radio Commands (LR20xx)

Table 5-16: OOK Packet Radio Commands

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

Downloaded from Arrow.com.

5.7 Host Interrupts (LR20xx)

The LR2021/LR2022/LR2012 utilizes 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 5-17: 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 SyncwordValid	LoRa header detected Valid Syncword	Rx	LoRa Others
7	CadDetected	Channel activity detected	Rx, CAD	LoRa, FSK
8	LoRaHdrTimestamp	LoRa header precise time-stamp (explicit mode). 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 *GetErrors* command)	All	All
17	CmdError	There was a host command fail/error	All	-

Downloaded from Arrow.com.

Table 5-17: 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 FLRC LoRa
24	AddrError	Packet was received with a wrong address match	Rx	FSK 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.

Downloaded from Arrow.com.

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

Downloaded from Arrow.com.

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 utilizes 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			

Downloaded from Arrow.com.

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 (15:8)	Stat (7:0)	Data1 (31:24)	Data1 (23:16)	Data1 (15:8)	Data1 (7:0)	Data2 (31:24)	...	DataN (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 (7:0)	sleep_time (31:24)	sleep_time (23:16)	sleep_time (15:8)	sleep_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

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 32kHz 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.

Downloaded from Arrow.com.

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 utilizes 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.

Downloaded from Arrow.com.

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 32kHz 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 0xFFFFFF 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.

Downloaded from Arrow.com.

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 32kHz 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 32kHz 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) 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 a s 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.

Downloaded from Arrow.com.

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 32kHz 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 32kHz 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*** in order 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*** in order 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 enabled 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.

Downloaded from Arrow.com.

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 Mode(6:0)	Timeout (23:16)	Timeout (15:8)	Timeout (7:0)	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)	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 1/32.768kHz steps): Timeout used for the auto Rx or auto Tx.

Delay: (in 1/32MHz 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

Downloaded from Arrow.com.

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) pbl_any 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 ***cad_timeout*** .
CAD_LBT	0x10	The chip performs a CAD operation and if no activity is detected, it goes to Tx mode and takes ***cad_timeout*** as Tx timeout.

cad_timeout defines the timeout duration after CAD for Rx (CAD_RX) or Tx (CAD_LBT), expressed in periods of 32MHz 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

Downloaded from Arrow.com.

Table 6-19: Lora CAD Recommended det_peak values

Number of symbols	SF5	SF6	SF7	SF8	SF9	SF10	SF11	SF12
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.

Downloaded from Arrow.com.

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 155* **.**

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 (23:16)	cad_timeout (15:8)	cad_timeout (7:0)	threshold (7:0)	exit_mode (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 (32MHz) clock ticks.

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

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

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.

Downloaded from Arrow.com.

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 (32MHz). 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 32.768kHz RTC (Real-Time Clock).

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

Downloaded from Arrow.com.

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

Downloaded from Arrow.com.

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 915MHz. 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 > 10MHz, 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 50MHz, or for a temperature change beyond +/-20C.

Downloaded from Arrow.com.

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 (15:8)	Stat (7:0)	IrqStatus (31:24)	IrqStatus (23:16)	IrqStatus (15:8)	IrqStatus (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 4MHz steps (Ex: 900MHz -> 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 4MHz 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.

Downloaded from Arrow.com.

6.5 Measurements

6.5.1 GetVbat

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) Format 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 1.35V) using the following formula:

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

- ***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.

Downloaded from Arrow.com.

6.5.2 GetTemp

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) 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:

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

  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:

- RRatio = RNTC/RNTC_25 = RBiasRatio * (Raw/8192)/(1.0 - (Raw/8192))
- The temperature is then given by T = Beta/(ln(RRatio) + Beta/298.15) - 273.15
  Where RBiasRatio, Beta are values set by the *SetNtcParams* command. Refer to Application Note AN1200.106 for details.

Downloaded from Arrow.com.

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) rfu(3:0) Enable
Data to host	Stat(15:8)	Stat(7:0)	IrqStatus(31:24)

Trim defines the threshold value:

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

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

Enable switches the low battery detection block ON/OFF:

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

Downloaded from Arrow.com.

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.

It is highly recommended 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)

Downloaded from Arrow.com.

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

It is possible to generate an IRQ on DIO (Digital Input/Output) 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)

Downloaded from Arrow.com.

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
FWMajor	0x01	0x02	0x02
FWMinor	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)

Downloaded from Arrow.com.

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

Downloaded from Arrow.com.

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.

Downloaded from Arrow.com.

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.

Downloaded from Arrow.com.

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.

Downloaded from Arrow.com.

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 (7:0)	tx_fifo_irq_enable (7:0)	rx_high_threshold (15:8)	rx_high_threshold (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 (15:8)	tx_low_threshold (7:0)	rx_low_threshold (15:8)	rx_low_threshold (7:0)	tx_high_threshold (15:8)	tx_high_threshold (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

Downloaded from Arrow.com.

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, please 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.

Downloaded from Arrow.com.

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)

Downloaded from Arrow.com.

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)

Downloaded from Arrow.com.

6.11 Clock Control

6.11.1 ConfigLfClock

The *ConfigLfClock* command configures the 32kHz 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 32kHz RC oscillator (default)
- 0x1: RFU
- 0x2: Use externally provided 32.768kHz 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) 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 (32MHz)
- 0x1: Division by 2 (16MHz)
- 0x2: Division by 4 (8MHz)
- ...
- 0xF: Division by 32768 (976.5625kHz)

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*.

Downloaded from Arrow.com.

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. It's important to note that 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	1.6V
0x01	1.7V
0x02	1.8V
0x03	2.2V
0x04	2.4V
0x05	2.7V
0x06	3.0V
0x07	3.3V

start_time indicates the maximum duration for the 32MHz oscillator to start and stabilize. The duration is measured in 32MHz clock periods. If the 32MHz 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

Downloaded from Arrow.com.

6.11.4 SetXoscCpTrim

The *SetXoscCpTrim* command allows the developer to trim the built-in foot capacitance of the 32MHz 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) xta(5:0)	rfu(1:0) 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 = 11.3pF, max = 47 = 33.4pF, 1 LSB = 0.47pF. See *Table 6-67*.
- **xtb**: Trims the foot capacitor on XTB pin. 0 = 11.1pF, max = 47 = 33.2pF, 1 LSB = 0.47pF. 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.

Downloaded from Arrow.com.

6.12.1 SetTempCompCfg

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

Table 6-69: SetTempCompCfg Command

Byte	0	1	2
Data from Host	0x01	0x32	rfu(4:0) ntc 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/lsb.

delay: First order time delay coefficient.

Downloaded from Arrow.com.

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 2.4GHz. 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 32MHz 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 32MHz 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 seamless processing and proper 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).

Downloaded from Arrow.com.

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) rx_path	rfu(4:0) rx_boost(2:0)
Data to host	Stat(15:8)	Stat(7:0)	IrqStatus (31:24)	IrqStatus (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) rfu(6:0)

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

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

Downloaded from Arrow.com.

7.3.4 SetRssiCalibration

The *SetRssiCalibration* command adjusts the RSSI (Received Signal Strength Indicator) value to compensate for variations in the Rx chain gain, which may occur due to differences in the Rx matching network.

Table 7-5: SetRssiCalibration Command

Byte	0	1	2	3	4	5	...
Data from host	0x02	0x05	rfu(5:0) rx_path_hf rx_path_lf	rfu(5:0) 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) 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 gain 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 the following entries for each Rx gain:

- Gain value 10 bits (signed fixed 8.2)
- Noise figure value 8 bits (unsigned fixed 6.2)

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

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 7-6: ResetRxStats Command

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

Downloaded from Arrow.com.

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 7-7: SetTimestampSource Command

Byte	0	1	2
Data from host	0x02	0x16	rfu(1:0) Index(1:0) 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 (32MHz) between the event and the SPI NSS falling edge of the request for the source specified in the ***Index*** parameter.

Table 7-8: GetTimestampValue Command

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

Table 7-9: 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)

The function returns ***Timestamp*** in periods of 32MHz clock.

Note that each source defined in *SetTimestampSource* can be mapped to an IRQ as described in *Table 5-17*.

Downloaded from Arrow.com.

7.3.8 GetCcaResult

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

Table 7-10: GetCcaResult Command

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

Table 7-11: 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) RssiMin(0) RssiMax(0) Rssiavg(0)

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

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

1dB 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 7-12: SetAgcGainManual Command

Byte	0	1	2
Data from host	0x02	0x1A	rfu(3:0) 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 form 0 to 13. A value of 0 enables the AGC.

Downloaded from Arrow.com.

7.3.10 GetRxPktLength

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

Table 7-13: GetRxPktLength Command

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

Table 7-14: 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.

Downloaded from Arrow.com.

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 2.4GHz band and can achieve an output power of up to +12dBm.
- 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 +22dBm output power.

7.4.1 SetPaConfig

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

Table 7-15: SetPaConfig Command

Byte	0	1	2	3	4
Data from host	0x02	0x02	pa_sel rfu(4:0) pa_lf_mode(1:0)	pa_lf_duty_cycle(3:0) pa_lf_slices(3:0)	rfu(2:0) 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 7-16* and *Table 7-17*.

pa_lf_slices. If LF PA is not used, then set this parameter to 7. See *Table 7-16* and *Table 7-17*.

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 7-18*.

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.

Downloaded from Arrow.com.

Table 7-16: Optimal Values for 915MHz Semtech Reference Design

Targeted Power [dBm]	TX_POWER	PA_LF_ DUTY_CYCLE	PA_LF_ SLICES
22	22	7	6
21.5	21.5	7	7
21	21	7	7
20.5	20.5	7	7
20	20.5	6	6
19.5	20	6	6
19	19.5	6	6
18.5	19	6	6
18	19	5	6
17.5	18.5	5	6
17	18	5	6
16.5	17.5	5	6
16	18	4	4
15.5	17	5	4
15	16.5	5	4
14.5	16	5	4
14	17	4	2
13.5	16	6	1
13	15.5	4	3
12.5	15.5	4	2
12	15	5	1
11.5	14.5	4	2
11	16	2	2
10.5	15	3	1
10	16	2	1

Downloaded from Arrow.com.

Table 7-17: Optimal Values for 490MHz Semtech Reference Design

Targeted Power [dBm]	TX_POWER	PA_LF_ DUTY_CYCLE	PA_LF_SLICES
21	22	7	7
20	21	7	7
17	17.5	7	7
16	16.5	7	7
14	15.5	7	3
13	15	7	2
10	15	5	1

Table 7-18: Optimal Values for 2445MHz Semtech Reference Design (LR2021/LR2022)

Targeted Power [dBm]	TX_POWER	PA_HF_DUTY_CYCLE
12	12	16
11	12	26
10	12	30
9	11	30
8	10.5	31
7	9	30
6	8	30
5	7.5	31
4	5	25
3	4	25
2	3.5	28
1	3	30
0	2	30

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 7-19: 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

Downloaded from Arrow.com.

- 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.

Downloaded from Arrow.com.

7.4.3 SetTxParams

The *SetTxParams* command sets the Transmit (Tx) power and ramp time of the Power Amplifier (PA). Before using this command, it is essential to call the *SetPaConfig* command, 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 7-20: 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 +0.5dB 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 7-21: 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 -22dBm to support standards like Z-Wave. Contact Semtech for details about this feature.

Downloaded from Arrow.com.

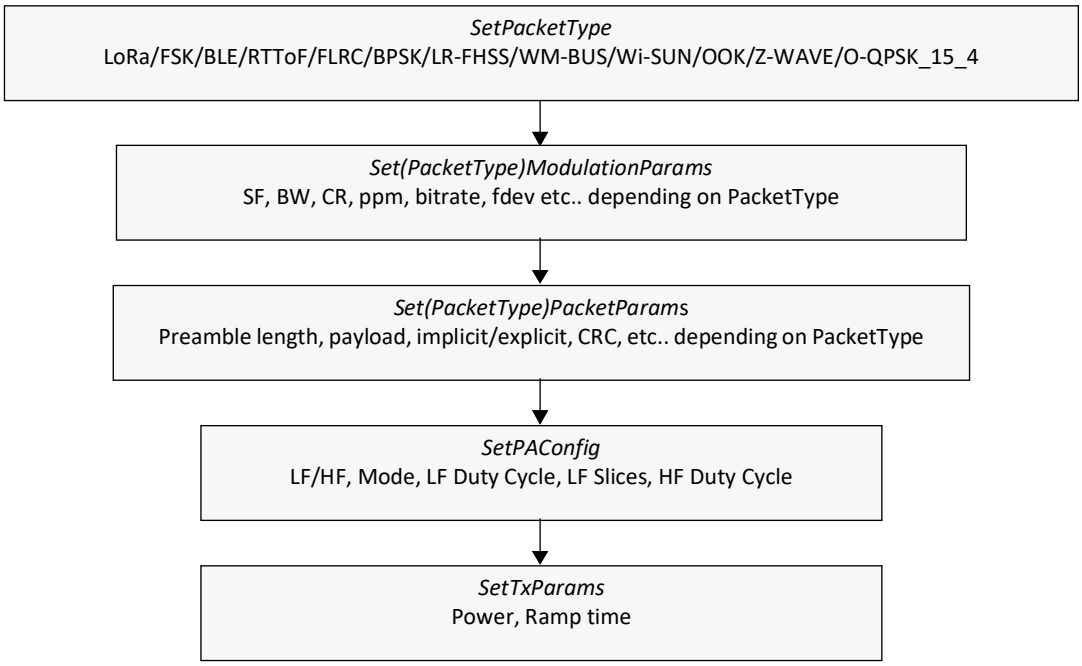
8. Modem Configuration (LR20xx)

The LR2021/LR2022/LR2012 incorporates various modems that support different constant envelope modulations. In order 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 that 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 utilize specific modulations, such as CAD (Channel Activity Detection), ranging, etc.

Downloaded from Arrow.com.

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

Downloaded from Arrow.com.

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

Downloaded from Arrow.com.

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)

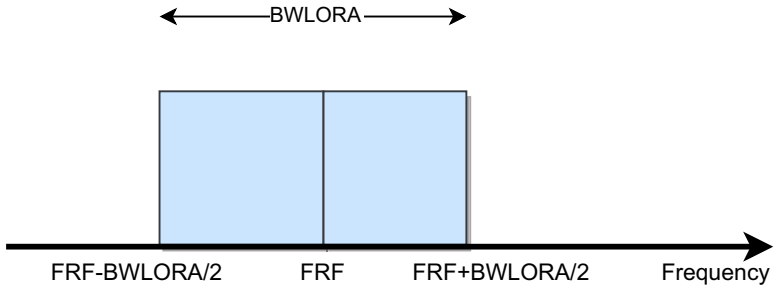
The spread spectrum LoRa modulation operates by encoding each bit of payload information using multiple chips of information. The symbol rate (Rs) denotes the rate at which a spread symbol, comprising 2^{SF} chips, is transmitted. The Spreading Factor (SF) is the ratio between the nominal symbol rate and chip rate and determines the number of bits sent per symbol.

It's essential to note that 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. It's important to note that 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.

Please be aware that 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 robustness of the link, the LoRa modem incorporates cyclic error coding, which enables forward error detection and correction. However, it's important to note that 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 robustness 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.

Downloaded from Arrow.com.

9.1.5 LoRa Symbol Rate (Rs)

By considering the key parameters that developers can control, the LoRa symbol rate is defined as follows:

Symbol rate = BWLORA / (2^SF)

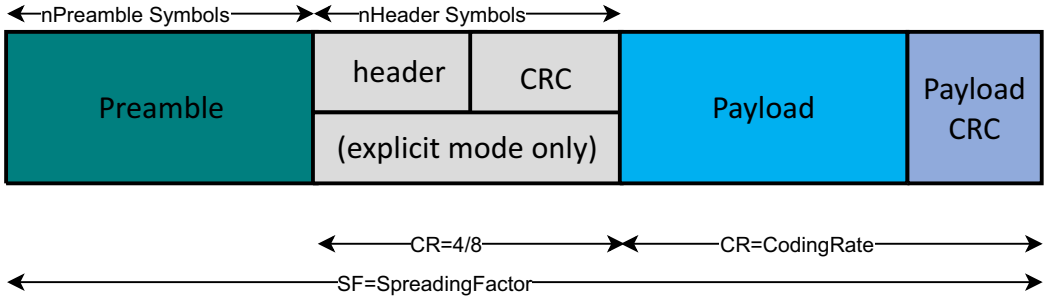
where BWLORA represents the programmed bandwidth and SF is the spreading factor. It's essential to note that the transmitted signal is a constant envelope signal, and equivalently, one chip is sent per second per Hertz (Hz) of bandwidth. Understanding and manipulating these parameters allow developers 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, it is recommended to 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.

Downloaded from Arrow.com.

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, it is essential to ensure that the main detector is 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.
- The sum of the Detection Factors, multiplied by the BW, must be smaller than 32e6.
  For SF5-12, the Detection Factors are: 10, 10, 12, 12, 14, 14, 16, 16. For example, the configuration BW1000 SF11+SF12 is not allowed because the sum of the detection factors is (16+16)*1e6 = 32e6.

Downloaded from Arrow.com.

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, in order 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)

In case the CAD ends up with CadDetected IRQ high, indicating the presence of LoRa symbols, the chip can seamlessly continue 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.

Downloaded from Arrow.com.

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.

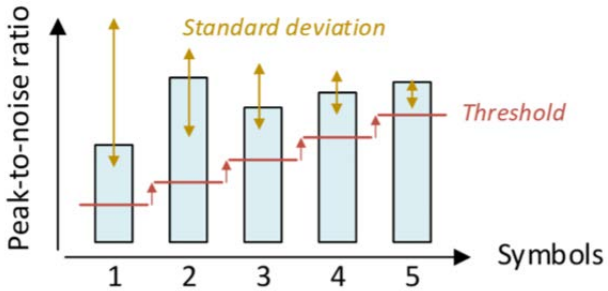
The LR2021/LR2022/LR2012 Fast CAD feature implements an adaptive threshold mechanism that progressively increases the peak-to-noise detection threshold for each successive symbol, as shown in *Figure 9-3* below, where each blue bar represents the measured peak-to-noise ratio of the received signal for 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.

This optimization reduces average CAD duration from ~5 symbols to approximately 1.5 symbols when no LoRa signal is present, directly translating to ~70% power consumption reduction for Listen-Before-Talk (LBT) and Carrier Sense Multiple Access (CSMA) applications that perform frequent channel sensing.

The feature is activated via the ***pnr_delta*** parameter in the *SetLoraCadParams* command, with detection sensitivity maintained within 3dB 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 state of the art 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 31kHz of frequency misalignment (for the nominal 125kHz 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 30ppm 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 +/-41.25kHz of frequency lenience for 125kHz 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)
```


Downloaded from Arrow.com.

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_compatiblity_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_compatiblity_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_compatiblity_mode*() to return to native operation.

Downloaded from Arrow.com.

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) bw(3:0)	cr(3:0) rfu(1:0) 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[1]	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 (31.25kHz actual)	0xA	LORA_BW_41 (41.67kHz actual)
0x3	LORA_BW_62 (62.50kHz actual)	0xB	LORA_BW_83 (83.34kHz actual)
0x4	LORA_BW_125	0xC	LORA_BW_101 (101.5625kHz actual)
0x5	LORA_BW_250	0xD	LORA_BW_203(203.125kHz actual)
0x6	LORA_BW_500	0xE	LORA_BW_406 (406.25kHz actual)
0x7	LORA_BW_1000	0xF	LORA_BW_812 (812.5kHz actual)

Downloaded from Arrow.com.

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 [1]
0x09	Long Interleaver Convolutional code 4/8 Overhead Ratio 2 *1*

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:

- 0 for SF5-10, BW250 / SF11, BW500, BW1000
- 1 for all other BW and SF11/12 (including SX1280 compatible bandwidths BW800/400/200)

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.

Please find detailed information about enabling the legacy mode in the LR20xx driver in *lr20xx_workarounds.c*.

Downloaded from Arrow.com.

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) header_type Crc 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, 203kHz), 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.

Downloaded from Arrow.com.

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 = ***nb_symbols***(7:3)*2^(***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.

Downloaded from Arrow.com.

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 500kHz, 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, it is possible to know which SF was demodulated with *GetLoraPacketStatus* (see detector(2:0)).
- 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)

Downloaded from Arrow.com.

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 (15:8)	Stat (7:0)	pkt_rx (15:8)	pkt_rx (7:0)	pkt_crc_ error (15:8)	pkt_crc_ error (7:0)	header_ crc_error (15:8)	header_ crc_error (7:0)	false_synch (15:8)	false_synch (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.

Downloaded from Arrow.com.

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 irqs).

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 (15:8)	Stat (7:0)	rfu(2:0) crc coding_rate (3:0)	pkt_length (7:0)	snr_pkt (7:0)	rssi_pkt(8:1)	rssi_signal_pkt (8:1)	rfu (1:0) detector(3:0) rssi_pkt_bit(0) ssi_signal_pkt(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.
- ***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 ***snr_pkt***/4.
- ***pkt_length***: Length of the last received payload in bytes.
- ***rssi_pkt***: Average Received Signal Strength Indication (RSSI) over the last received packet. The actual signal power is calculated as -***rssi_pkt***/2 [dBm].
- ***rssi_signal_pkt***: Estimation of the RSSI of the LoRa signal (after despreading) on the last received packet. The actual value is -***rssi_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

Downloaded from Arrow.com.

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) 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) rfu hop_period (12:8)	hop_period (7:0)	freq_hop1 (31:24)	freq_hop1 (23:16)	freq_hop1 (15:8)	freq_hop1 (7:0)	freq_hop2 (31:24)	...
Data to host	Stat (15:8)	Stat (7:0)	IrqStatus (31:24)	IrqStatus (23:16)	IrqStatus (15:8)	IrqStatus (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_hopx (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)

Downloaded from Arrow.com.

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) 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) pnr_delta1 (3:0)	det_peak1 (7:0)	rfu(2:0) pnr_delta2 (3:0)	det_peak2 (7:0)	rfu(2:0) pnr_delta3 (3:0)	det_peak3 (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

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*.

Downloaded from Arrow.com.

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.

Downloaded from Arrow.com.

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 (23:16)	Rng1 (15:8)	Rng1 (7:0)	Rssi1	Rng2 (23:16)	Rng2 (15:8)	Rng2 (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.

The Distance [m] = ***Rng_i****150/(2^12*BWLoRa), with BWLoRa in MHz.

Downloaded from Arrow.com.

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 (15:8)	ExchangeValid (7:0)	RequestValid (15:8)	RequestValid (7:0)

Byte	6	7	8	9	10	11
Data from host	0x00	0x00	0x00	0x00	0x00	0x00
Data to host	ResponseDone (15:8)	ResponseDone (7:0)	Timeout (15:8)	Timeout (7:0)	RequestDiscarded (15:8)	RequestDiscarded (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.

Downloaded from Arrow.com.

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 spy_mode 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.

Downloaded from Arrow.com.

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 seamless integration and easy migration for developers who have previously utilized these chip-sets in their projects. See Application Note AN1200.103 for details of FSK usage.

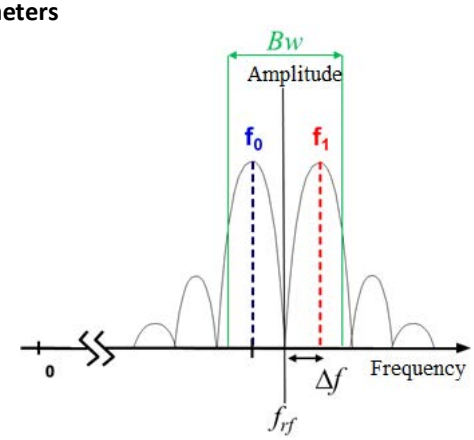
11.1.1 Modem Bandwidth and Bit Rates

The FSK modem integrated into the LR2021/LR2022/LR2012 offers versatile 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 Δf 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:

$$B_{-20\text{dB}} = 2\Delta f + R_b + \textbf{freqoffset}$$

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.

Downloaded from Arrow.com.

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, β, 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 Δf 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 2Mbps
- Maximum BW to use is: BW_max <= bitrate * osr, where osr (oversampling rate) is:

Table 11-1: Oversampling Rate Limitations

Modulation Index	osr
h < 0.85	4
0.85 <= h < 1.6	8
1.6 <= h < 2.6	8
h >= 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 bitrate * osr product must not exceed 8MHz
- The BW must be large enough to contain the signal and any required frequency error:
  BW_min >= 2*Fdev + bitrate + Freq_err_max (Carlson approximation)
- It is possible to set 0 bits of detection, in which case the Syncword is used for the detection. This configuration is only recommended for 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 osr=4. Higher osr require fewer bits.
- The minimum modulation index is h=0.3 for both FSK and GFSK (tested at 500kbps). 0 bits of detection are recommended in this case.

Table 11-2: (G)FSK Limitations Summary (Sheet 1 of 2)

Name	Value
bitrate max	2Mbps
bitrate min	500bps
h_min	0.3
h_max FSK	8.33
h_max GFSK (bt=0.5)	6.2

Downloaded from Arrow.com.

Table 11-2: (G)FSK Limitations Summary (Sheet 2 of 2)

Name	Value
BW max	3076kHz or bitrate * osr
BW min	3.5kHz or signal_bw + Freq_error
Sample rate max	bitrate*osr <= 8MHz

11.2 (G)FSK Packet Handler

The FSK packet format offers a well-established structure suitable for implementation in proprietary communication links that utilize 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.

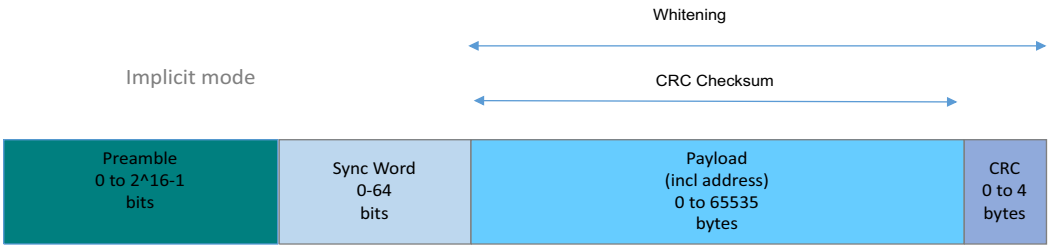
Within the FSK packet handler, two main packet formats are available: fixed length and variable length packets.

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.

It is important to note that 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



Downloaded from Arrow.com.

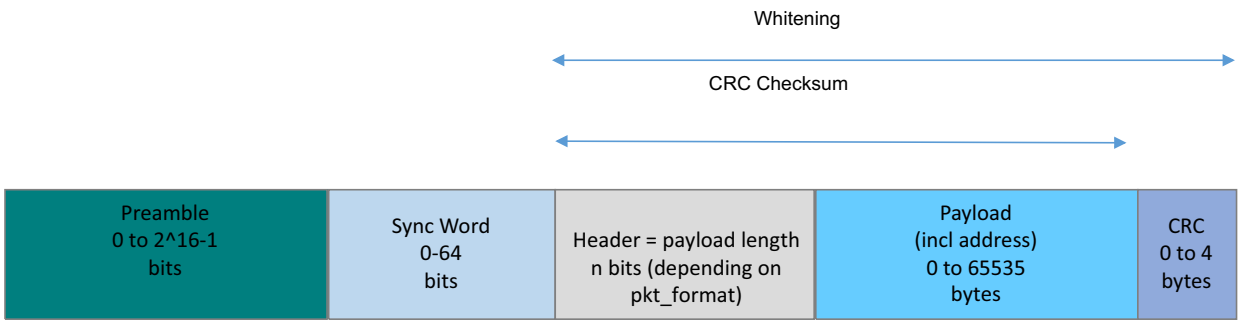
11.2.2 Variable-Length Packet

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. However, it is essential to note that when address filtering is activated, the maximum packet length is limited to 254 bytes to accommodate the additional addressing information.

The format of the variable-length packet ensures that the receiver can determine the size of incoming packets, enabling proper data handling and processing.

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.

Downloaded from Arrow.com.

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. However, it's essential to note that 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 168.

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.

Downloaded from Arrow.com.

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

Downloaded from Arrow.com.

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) long_preamble_mode pld_lenUnit addr_comp(1:0) pkt_format(1:0)	pld_len (15:8)	pld_len (7:0)	Crc(3:0) 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[1]

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

Downloaded from Arrow.com.

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

Downloaded from Arrow.com.

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 on4 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 *SetFskWhiteningParams*

Downloaded from Arrow.com.

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) 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.

Downloaded from Arrow.com.

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 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 utilized.

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).

Downloaded from Arrow.com.

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 (15:8)	Stat (7:0)	pkt_rx (15:8)	pkt_rx (7:0)	pkt_crc_error (15:8)	pkt_crc_error (7:0)	LenError (15:8)	LenError (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 (15:8)	pbl_det (7:0)	sync_ok (15:8)	sync_ok (7:0)	sync_fail (15:8)	sync_fail (7:0)	timeout (15:8)	timeout (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*.

Downloaded from Arrow.com.

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 ***rssi_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 (15:8)	pkt_len (7:0)	rssi_avg (8:1)	rssi_sync (8:1)	rfu(1:0) AddrMatchBcast AddrMatchNode rfu rssi_avg(0) rfu rssi_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,...)
- ***rssi_avg***: Average Received Signal Strength Indication (RSSI) value over the last received packet. The actual signal power can be calculated as -***rssi_avg***/2 (dBm)
- ***rssi_sync***: Parameter gives the RSSI value latched at Syncword detection. Actual signal power can be calculated as -***rssi_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 0.25dB steps). LQI is the ratio in dB between detection correlation peak and average power. LQI of >10dB indicates good quality (in absence of FEC).

Downloaded from Arrow.com.

12. Wireless M-Bus (WM-BUS) (LR20xx)

The LR2021/LR2022/LR2012 wireless chip provides comprehensive support for various WM-BUS modes, making it a versatile 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 Manchester encoded	32.768kcps +/- 1.5%	+/- 50kHz -20% +60%	Rx/Tx
T1	2FSK 3 out of 6	100kcps +/- 12%	+/-50kHz -20% +60%	Tx only
T2	2FSK Manchester encoded	32.768kcps +/- 1.5%	+/-50kHz -20% +60%	Meter Rx
T2	2FSK 3 out of 6	100kcps +/- 10%	+/-50kHz -20% +60%	Meter Tx
R2	2FSK Manchester encoded	4.8kcps +/- 1.5%	+/-6kHz +/-20%	Rx/Tx
C1	2FSK	100kcps +/-100ppm	+/-45kHz +/-25%	Tx only
C2	2GFSK BT=0.5	50kcps +/- 100ppm	+/-25kHz +/-25%	Meter Rx
C2	2FSK	100kcps +/- 100ppm	+/-45kHz +/-25%	Meter Tx
F2	2FSK	2.4kcps +/- 100ppm	+/-5.5kHz -11%, +28%	Rx/Tx
N 2.4kbps	2GFSK BT=0.5	2.4kcps +/- 100ppm	+/-2.4kHz +/-30%	Rx/Tx
N 4.8kbps	2GFSK BT=0.5	4.8kcps +/-100ppm	+/-2.4kHz +/-30%	Rx/Tx
N 6.4kbps[1]	4GFSK BT=0.5	6.4kcps +/- 100ppm	-3.2kHz, -10.6kHz 1.06kHz, 3.2kHz +/-5%	Rx/Tx
N 19.2kbps[1]	4GFSK BT=0.5	19.2kcps +/-100ppm	-7.2kHz, -2.4kHz 2.4kHz, 7.2kHz +/-5%	Rx/Tx

1. Note that the 4GFSK modes are supported; however, 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.

Downloaded from Arrow.com.

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	pld_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 4.8kbps
- 0x9 = N 2.4kbps
- 0xA = N 6.4kbps*
- 0xB = N 19.2kbps*
- 0xC = F2

*Note that the 4GFSK modes are supported but with a maximum variation of the frequency deviation of 5% instead of 30% as 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.

Downloaded from Arrow.com.

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

pld_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, it is recommended 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 (15:8)	pkt_crc_error (7:0)	LenError (15:8)	LenError (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.

Downloaded from Arrow.com.

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
Data to host	Stat (15:8)	Stat (7:0)	LField (7:0)	pkt_len (15:8)	pkt_len (7:0)	rssi_avg (8:1)	rssi_sync (8:1)	CrcError (16:9)	CrcError (8:1)	SwIndex CrcError(0) rfu rssi_avg(0) rfu(2:0) rssi_sync(0)	Lqi (7:0)

Status is updated at the end of a reception (RxDone IRQ), but ***SwIndex*** and ***rssi_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, ...).
- ***rssi_avg***: Average over the last received packet of RSSI (Received Signal Strength Indication). The actual signal power can be calculated as -***rssi_avg***/2 (dBm).
- ***rssi_sync***: Latched RSSI value after Syncword detection. The actual signal power can be calculated as -***rssi_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 0.25dB steps)

Downloaded from Arrow.com.

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.

Downloaded from Arrow.com.

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	50ksps +/-100ppm	0.5
1b	2FSK	50ksps +/-100ppm	1
2a	2FSK	100ksps +/-100ppm	0.5
2b	2FSK	100ksps +/-100ppm	1
3	2FSK	150ksps +/-100ppm	0.5
4a	2FSK	200ksps +/-100ppm	0.5
4b	2FSK	200ksps +/- 100ppm	1
5	2FSK	300ksps +/- 100ppm	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.

Downloaded from Arrow.com.

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

Downloaded from Arrow.com.

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
Data from host	0x02	0x71	rfu(1:0) fcs_tx Whitening crc_on mode_switch_tx fec_tx(1:0)	frame_len_tx (15:8)	frame_len_tx (7:0)	pbl_len_tx (7:0)	pbl_detect (7:0)
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***)

Downloaded from Arrow.com.

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.

Downloaded from Arrow.com.

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 (15:8)	Stat (7:0)	Header (15:8)	Header (7:0)	pkt_len (15:8)	pkt_len (7:0)	rssi_avg (8:1)	rssi_sync (8:1)	SyncwordIdx rfu(3:0) rssi_avg(0) rfu rssi_sync(0)	Lqi (7:0)

The status is updated at the end of a reception (RxDone IRQ), but SyncwordIdx and rssi_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
- ***rssi_avg***: Average over the last received packet of RSSI (Received Signal Strength Indication). The actual signal power can be calculated as -***rssi_avg***/2 (dBm)
- ***rssi_sync***: RSSI value latched at Syncword detection. The actual signal power can be calculated as -***rssi_sync***/2 (dBm).
- ***Lqi***: Link quality indicator (in 0.25dB 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.

Downloaded from Arrow.com.

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. It's important to note that 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 (1Mbps)
- 0x1: LE 2M PHY (2Mbps)
- 0x2: LE Coded S = 2 (500kbps)
- 0x3: LE Coded S = 8 (125kbps)

For all LE coded modes, please 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.

Downloaded from Arrow.com.

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) crc_in_fifo 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.

Downloaded from Arrow.com.

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 (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 execution of the *ResetRxStats* command.

Downloaded from Arrow.com.

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 (15:8)	stat (7:0)	pkt_len (15:8)	pkt_len (7:0)	rssi_avg (8:1)	rssi_sync (8:1)	rfu(4:0) rssi_avg(0) rfu(0) rssi_sync(0)	Lqi (7:0)

The status is updated at the end of a reception (RxDone IRQ), but ***rssi_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, ...).
- ***rssi_avg***: The average Received Signal Strength Indication (RSSI) value over the last received packet. The actual signal power is calculated as -***rssi_avg***/2 (dBm).
- ***rssi_sync***: The RSSI value latched after Syncword detection. The actual signal power is calculated as -***rssi_sync***/2 (dBm).
- ***Lqi***: The link quality indicator in 0.25dB steps.

Downloaded from Arrow.com.

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 π 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) rfu diff_mode diff_mode_init(1:0)
Data to host	Stat (15:8)	Stat (7:0)	IrqStatus (31:24)	IrqStatus (23:16)	IrqStatus (15:8)	IrqStatus (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

Downloaded from Arrow.com.

- 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
Data from host	0x02	0x51	pld_len (7:0)	rfu(1:0) Mode(1:0) 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

Downloaded from Arrow.com.

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 utilizes 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 (31:24)	BitRate (23:16)	BitRate (15:8)	BitRate (7:0)	pulse_shape (7:0)	rx_bw (7:0)	Depth (7:0)
Data to host	Stat(15:8)	Stat(7:0)	IrqStatus (31:24)	IrqStatus (23:16)	IrqStatus (15:8)	IrqStatus (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 20dB. The exact value needs to be measured on the device and depends on the external components and BitRate.

Downloaded from Arrow.com.

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 (15:8)	pre_len_tx (7:0)	rfu(3:0) addr_comp(1:0) pkt_format(1:0)	pld_len (15:8)	pld_len (7:0)	Crc(3:0) Manchester (3:0)
Data to host	Stat(15:8)	Stat(7:0)	IrqStatus (31:24)	IrqStatus (23:16)	IrqStatus (15:8)	IrqStatus (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 6-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). Note that 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

Note that is recommended to have either whitening or Manchester encoding enabled for OOK.

Downloaded from Arrow.com.

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 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.

Downloaded from Arrow.com.

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 (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 *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.

Downloaded from Arrow.com.

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 (15:8)	Stat (7:0)	pkt_len (15:8)	pkt_len (7:0)	rssi_avg (8:1)	RssiHigh (8:1)	rfu(1:0) AddrMatchBcast AddrMatchNode rfu rssi_avg(0) rfu RssiHigh(0)	Lqi (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, ...).
- ***rssi_avg***: Average over the last received packet of Received Signal Strength Indicator (RSSI). The actual signal power is calculated as -***rssi_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 0.25dB steps.

Downloaded from Arrow.com.

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_ pattern(15:8)	preamble_ pattern(7:0)	rfu(3:0) pattern_length (3:0)	rfu(2:0) pattern_num_ repeats(4:0)	rfu(1:0) sw_is_raw sfd_kind 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

Downloaded from Arrow.com.

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) Polynom(11:8)	Polynom(7:0)	rfu(3:0) Init(11:8)	Init(7:0)
Data to host	Stat(15:8)	Stat(7:0)	IrqStatus(31:24)	IrqStatus(23:16)	IrqStatus(15:8)	IrqStatus7:0)

bit_idx defines the index of the LFSR bit used to whitened 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.

It is recommended to have either whitening or Manchester encoding enabled when using OOK modulation.

Downloaded from Arrow.com.

17. LR-FHSS (LR20xx)

17.1 LR-FHSS Modulation

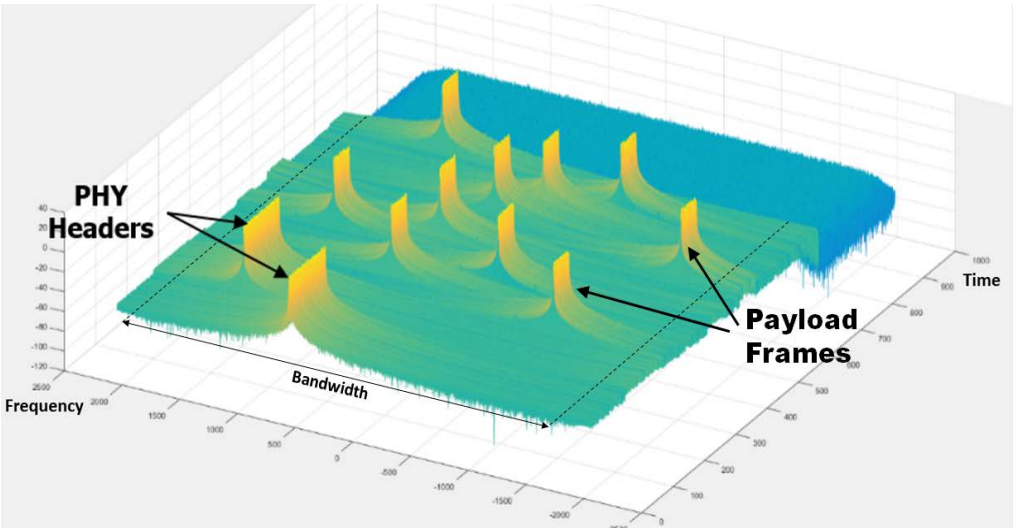
LR-FHSS (Frequency Hopping Spread Spectrum) is a modulation technique specifically designed for long-range applications in dense areas, making it ideal for scenarios where numerous devices need to communicate over an extended area. LR-FHSS can achieve a link-budget performance similar to LoRa SF12 BW125kHz, 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 robust and 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. However, it is important to note that the payload utilizes 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 robust 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.

Downloaded from Arrow.com.

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.

It's important to note that 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) Cr(3:0)	mod_type(3:0) Grid(3:0)	Hopping(3:0) Bw(3:0)	rfu(6:0) Sequence(8)	Sequence (7:0)	Offset (7:0)
Data to host	Stat (15:8)	Stat (7:0)	IrqStatus (31:24)	IrqStatus (23:16)	Irq status (15:8)	IrqStatus (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: 25.390625kHz
- 1: 3.90625kHz

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)

Downloaded from Arrow.com.

Bw defines the frequency range span of the hopping sequence: [0:9] according to LR-FHSS specifications:

- 0x00: 39.06kHz
- 0x01: 85.94kHz
- 0x02: 136.72kHz
- 0x03: 183.59kHz
- 0x04: 335.94kHz
- 0x05: 386.72kHz
- 0x06: 722.66kHz
- 0x07: 773.44kHz
- 0x08: 1523.4kHz
- 0x09: 1574.2kHz
- 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 = 25.390625kHz
- Signed values in [-4; 3] for Grid = 3.90625kHz

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 utilized 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.

Downloaded from Arrow.com.

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(nb_used_freqs, 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 (31:24)	...	...

Downloaded from Arrow.com.

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 an 8dB 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*.

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 130kbps to 2.6Mbps, with available modulation bandwidths spanning from 300kHz to 2.6MHz. However, it's important to note that 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 Rb [Mbps]	Bandwidth BW [MHz *DSB*]
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

Downloaded from Arrow.com.

18.1.2 FEC Coding Rate

The FLRC modem offers the option to utilize forward error correction (FEC). When FEC is applied, convolutional coding is used to add redundant information to the packet, enabling the modem to perform error correction. This redundancy ensures that the packet payload information remains robust, even when exposed to bursts of interference from other radio services sharing the same band or channel. However, it's important to note that FEC introduces some overhead, affecting the raw bit rate and effective bit rate. The next table shows the effective FLRC bit rates based on FEC usage.

Table 18-2: Effective FLRC Bit Rates (Sheet 1 of 2)

Symbol	Raw Programmed Bit Rate Rb [Mbps]	Programmed Coding Rate CR	Effective Bit Rate Rbeff [Mbps]
FLRC_BR_2_600_BW_2_6	2.6	1(1)	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	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	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	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	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	0.52
	0.52	3/4	0.39
	0.52	2/3	0.34
	0.52	1/2	0.26

Downloaded from Arrow.com.

Table 18-2: Effective FLRC Bit Rates (Sheet 2 of 2)

Symbol	Raw Programmed Bit Rate Rb [Mbps]	Programmed Coding Rate CR	Effective Bit Rate Rbeff [Mbps]
FLRC_BR_0_325_BW_0_3	0.325	1	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	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: 0.5, 1, or OFF. The configuration of the Gaussian filter (BT) is achieved through the *SetFlrcModulationParams* command.

Downloaded from Arrow.com.

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 [Mbps]	Bandwidth [MHz]	Frequency Tolerance [kHz]
2.6	2.666	+/- 150
2.08	2.222	+/- 150
1.3	1.333	+/- 100
1.04	1.333	+/- 100
0.65	0.740	+/- 70
0.52	0.571	+/- 50
0.325	0.357	+/- 30
0.260	0.307	+/- 25

Downloaded from Arrow.com.

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.

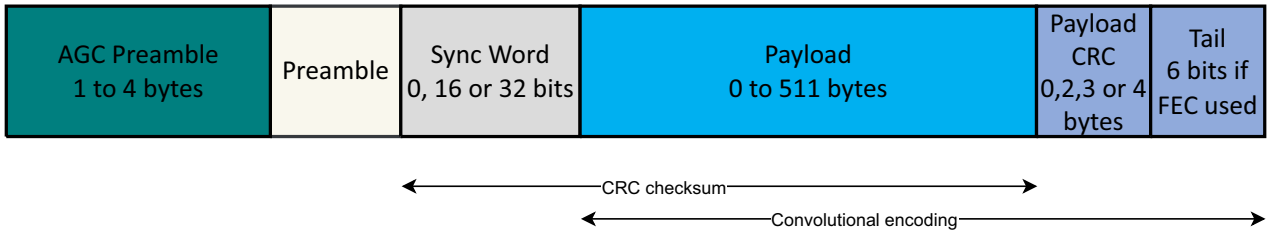
18.2.1 Fixed-Length Packet Format

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. Note that 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.

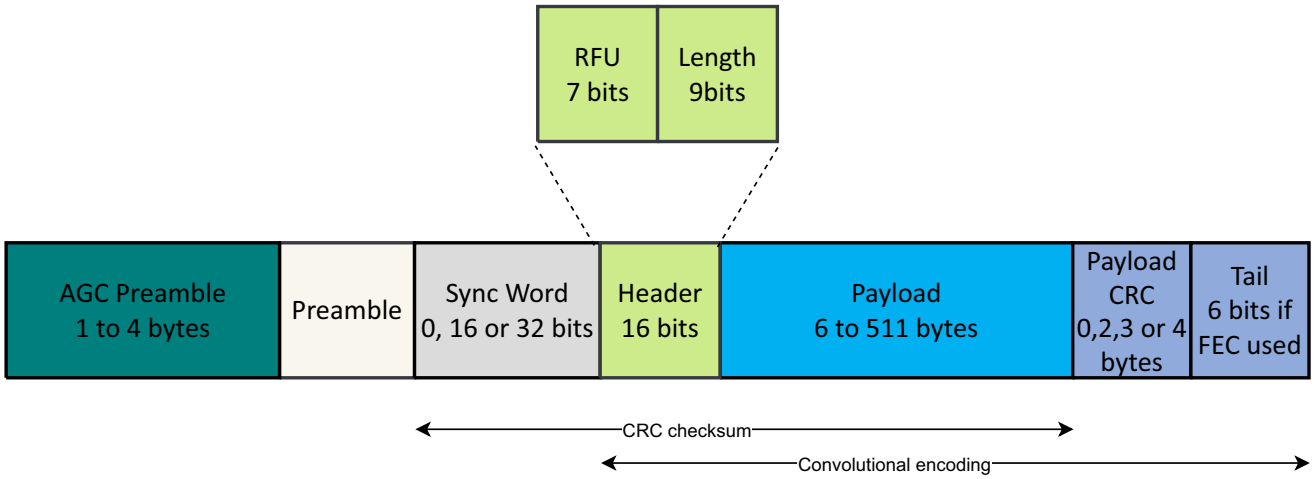
Figure 18-1: FLRC Fixed-length Packet Format



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



Downloaded from Arrow.com.

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:

$$n_{coded} = ceil\left[(n_{Payload} + n_{CRC} + n_{tail})\text{x}\left(\frac{1}{n_{CR}}\right)\right]$$

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:

$$ToA_{FLRC} = t_{bit}{}^{*}(n_{uncoded} + n_{coded})$$

Downloaded from Arrow.com.

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) 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 = 2600kbps and BW = 2666kHz
- 0x01: BR = 2080kbps and BW = 2222kHz
- 0x02: BR = 1300kbps and BW = 1333kHz
- 0x03: BR = 1040kbps and BW = 1333kHz
- 0x04: BR = 650kbps and BW = 740kHz
- 0x05: BR = 520kbps and BW = 571kHz
- 0x06: BR = 325kbps and BW = 357kHz
- 0x07: BR = 260kbps and BW = 307kHz

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
- 0x5: Gaussian with BT 0.5
- 0x7: Gaussian with BT 1.0

Downloaded from Arrow.com.

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) agc_pbl_len(3:0) sync_len(1:0)	sync_tx(1:0) sync_match(2:0) 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*** if 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 RxFifo and TxFifo 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.

Downloaded from Arrow.com.

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 (15:8)	Stat (7:0)	pkt_rx (15:0)	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 automatically reset under specific conditions, including power-on reset (POR), sleep without memory retention, or executing the *ResetRxStats* command.

Downloaded from Arrow.com.

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)	rssi_avg(8:1)	rssi_sync(8:1)	sw_num(3:0) rfu rssi_avg(0) rfu rssi_sync(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.
- ***rssi_avg***: Represents the average Received Signal Strength Indicator (RSSI) over the last received packet. The actual signal power can be calculated as -***rssi_avg***/2 [dBm]. This value gives an indication of the signal strength during the entire packet reception.
- ***rssi_sync***: Indicates the RSSI value latched after Syncword detection. The actual signal power is given as -***rssi_sync***/2 [dBm]. This value specifically reflects the signal strength at the moment when the Syncword is detected during the reception process.

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 (7:0)	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)	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.

Downloaded from Arrow.com.

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 2Mcps and a bit rate of 250kbps. 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 (7:0)	rx_bw (7:0)	pld_len (7:0)	pbl_len_tx (15:8)	pbl_len_tx (7:0)	rfu(5:0) AddressOn FcsMode
Data to host	stat (15:8)	stat (7:0)	IrqStatus (31:24)	IrqStatus (23:16)	IrqStatus (15:8)	IrqStatus (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 2Mcps, bit rate 250kbps
- others are rfu

rx_bw defines the bandwidth in Rx mode in the 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.

AddressOn controls the address filtering feature (Address is set using the *SetOqpskAddress* command):

- 0: Disable address filtering
- 1: Enable address filtering

FcsMode 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.

Downloaded from Arrow.com.

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*.

Downloaded from Arrow.com.

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 ***rssi_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 (15:8)	Stat (7:0)	RxHeader (7:0)	pkt_len (15:8)	pkt_len (7:0)	rssi_avg (8:1)	rssi_sync (8:1)	rfu(4:0) rssi_avg(0) rfu rssi_sync(0)	lqi (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, ...)
- ***rssi_avg***: RSSI average over the last packet received. Actual signal power is –***rssi_avg***/2 (dBm)
- ***rssi_sync***: Latched RSSI value after Syncword detection. Actual signal power is –***rssi_sync***/2 (dBm)
- ***lqi***: Link quality indicator (0.25dB)

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)	IrqStatus(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.

Downloaded from Arrow.com.

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 (63:56)	LongDestAddr (55:48)	LongDestAddr (47:40)	LongDestAddr (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 (31:24)	LongDestAddr (23:16)	LongDestAddr (15:8)	LongDestAddr (7:0)	ShortDestAddr (15:8)	ShortDestAddr (7:0)
Data to host	0x00	0x00	0x00	0x00	0x00	0x00

Byte	12	13	14
Data from host	PanId (15:8)	PanId (7:0)	TransId (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.

Downloaded from Arrow.com.

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 Manchester	19.2ksps +/-27ppm	2.08
R2	2FSK	40ksps +/-27ppm	1
R3	2GFSK NRZ BT 0.6	100ksps +/-100ppm	0.58
LR1	O-QPSK	25ksps +/27ppm	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.

Downloaded from Arrow.com.

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: FcsMode
  - 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.

Downloaded from Arrow.com.

20.3.2 SetZwaveHomeIdFiltering

The *SetZwaveHomeIdFiltering* 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: SetZwaveHomeIdFiltering 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*.

Downloaded from Arrow.com.

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 ***rssi_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 (15:8)	Stat (7:0)	pkt_len (15:8)	pkt_len (7:0)	rssi_avg (8:1)	rssi_sync (8:1)	LastDetected (7:0)	rfu(4:0) rssi_avg(0) rfu rssi_sync(0)	Lqi (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,...)
- ***rssi_avg***: Average over last packet received of RSSI. Actual signal power is –***rssi_avg***/2 (dBm)
- ***rssi_sync***: Latch RSSI value after Syncword detection. Actual signal power is –***rssi_sync***/2 (dBm)
- ***LastDetected***: Data rate of the latest Rx packet:
  - 0x1: R1 mode
  - 0x2: R2 mode
  - 0x3: R3 mode
  - 0x4: LR1 mode
  - Other: RFU
- ***Lqi***: Link quality indicator (in 0.25dB steps)

Downloaded from Arrow.com.

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 rfu (2:0) 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.

Downloaded from Arrow.com.

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) det4 det3 det2 det1	bitrate_ch4(1:0) bitrate_ch3(1:0) bitrate_ch2(1:0) bitrate_ch1(1:0)	addr_comp (7:0)	options (7:0)
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.

Downloaded from Arrow.com.

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
Data from host	0x02	0x9D
Data to host	Stat(15:8)	Stat(7:0)

Downloaded from Arrow.com.

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. Please 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 (*SetTxCw*(...) command)
D-M2	Continuously modulated signal with the greatest occupied RF bandwidth	Continuous modulation (*SetTxInfinitePreamble(...)* 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.

Downloaded from Arrow.com.

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)

Downloaded from Arrow.com.

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 Unified Software Platform (USP) Github page available at https://github.com/lora-net/usp.
Most of the workarounds are implemented in the Firmware Patch RAM, loaded directly by the drivers during initialization. The other workarounds are listed here below, and are automatically applied by the drivers.
Please review the guidelines provided individually in the drivers documentation 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 2.4GHz or sub-GHz), the RTToF accuracy is optimized when calling specific workarounds described in *lr20xx_workarounds.c*. Please refer to the LR20xx driver for reference.

Downloaded from Arrow.com.

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.

Please 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 80nA when the PRAM is loaded (as reflected in IDDSL1 parameter)

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").

Please 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).

Downloaded from Arrow.com.

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 1.8V to 3.7V supply voltage range, limiting thermal drift. Note that PA_LF operation at +22dBm requires a supply voltage greater than 2.2V 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 +22dBm requires a RF metallic shield. However, when using the SIMO with a shield, up to 1.5dB 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
2.4GHz	All	None
All Sub-GHz	FLRC > 1040kbps	Up to 1.5dB
	FLRC <= 1040kbps	Up to 0.5dB
	LoRa BW >= 800kHz	Up to 1.5dB
	LoRa <= 500kHz	Up to 1dB
	(G)FSK <= 100kbps	Up to 0.5dB
	(G)FSK > 100kbps	Up to 1dB
	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.

It is recommended that you follow closely 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.

Downloaded from Arrow.com.

23.3 Oscillator Recommendations

A standard 32MHz 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°C or >+70°C) or applications that require continuous transmission or very long packet durations. This comes however at the expense of 1-2mA additional current consumption.

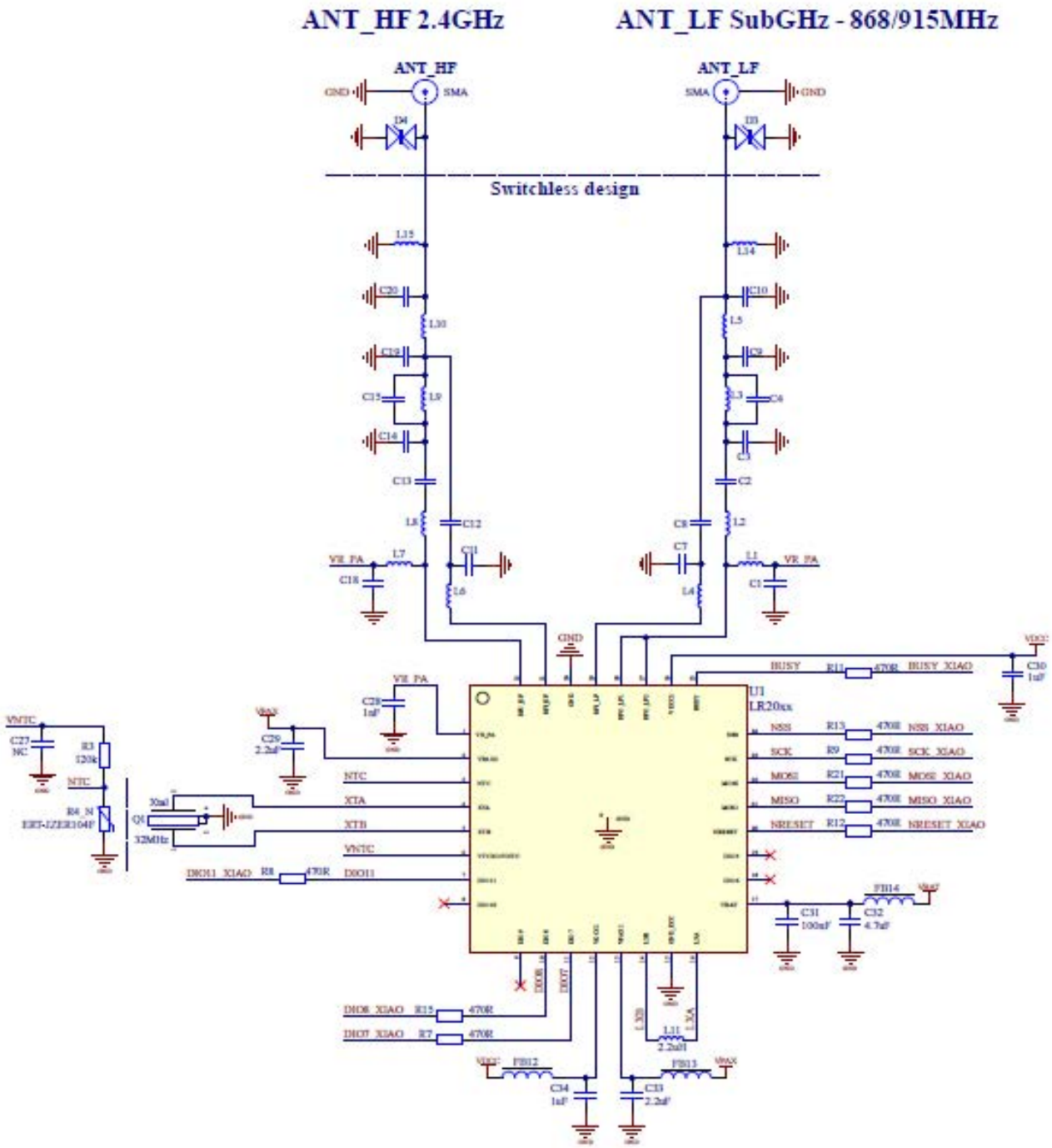
The selection between crystal and TCXO should be based on the application's targeted power consumption budget, thermal environment, packet duration requirements, and whether frequency stability must be guaranteed under all operating conditions. Semtech provides a list of recommended references in AN1200.59 Reference Clock Selection.

Downloaded from Arrow.com.

23.4 Reference Design

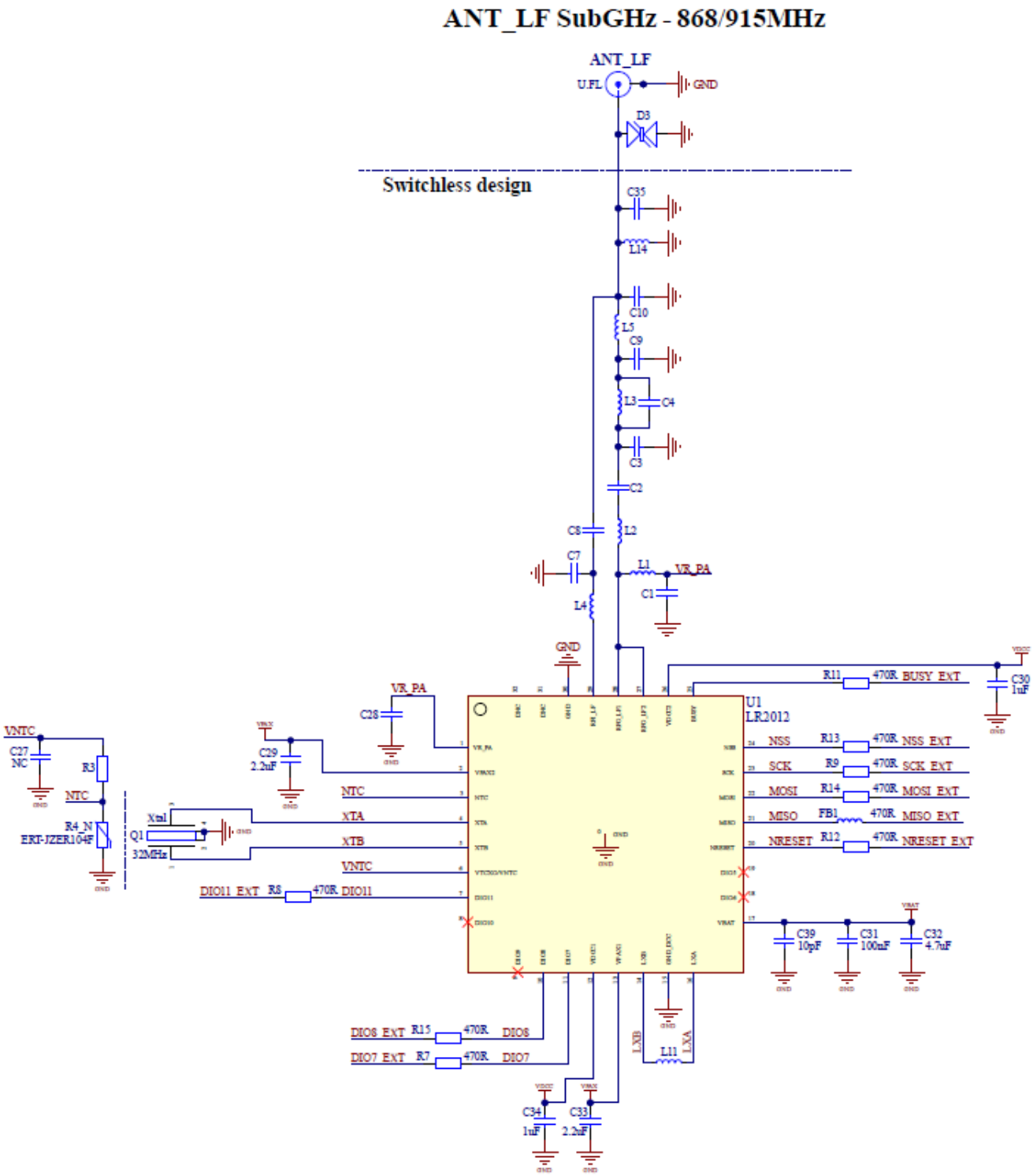
Semtech provides the following reference designs on a standard 4-layer FR4 PCB. It is recommended to follow the reference design as precisely as possible to match the performance stated in the datasheet. Follow the recommendations in AN1200.66 PCB Design Guidelines for an optimal PCB design. The LR2012 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



Downloaded from Arrow.com.

Figure 23-2: LR2012 Reference Design Schematic



Downloaded from Arrow.com.

Figure 23-3: LR2021/LR2022 Reference Design PCB Layout

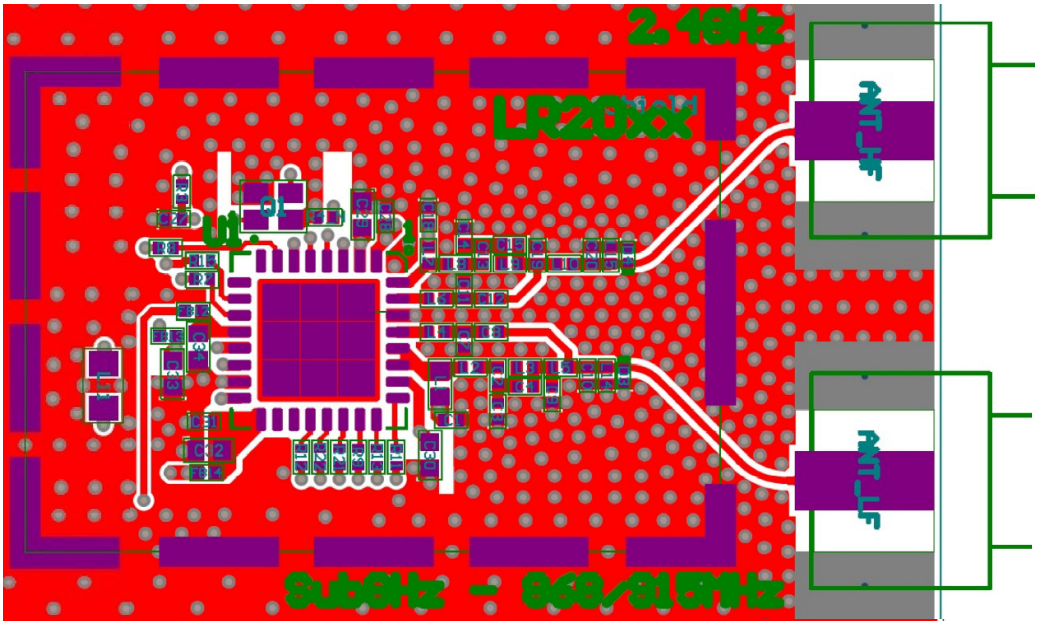
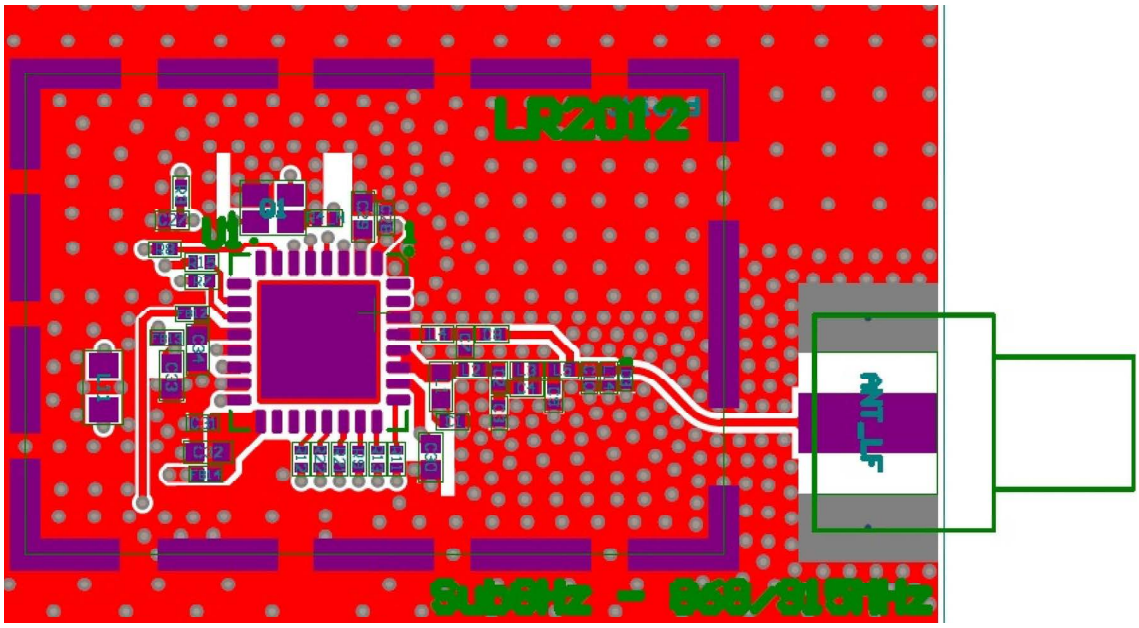


Figure 23-4: LR2012 Reference Design PCB Layout



Downloaded from Arrow.com.

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 32MHz oscillator): Leave unconnected
- LXA/LXB when LDO regulation is used: Leave unconnected. Note that 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

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 0xC0DE 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 (≥25%) above the nominal PA current to prevent false triggering during normal operation.

Downloaded from Arrow.com.

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

Note that all the LoRaWAN configurations are fully compatible between all four generations of LoRa radios.

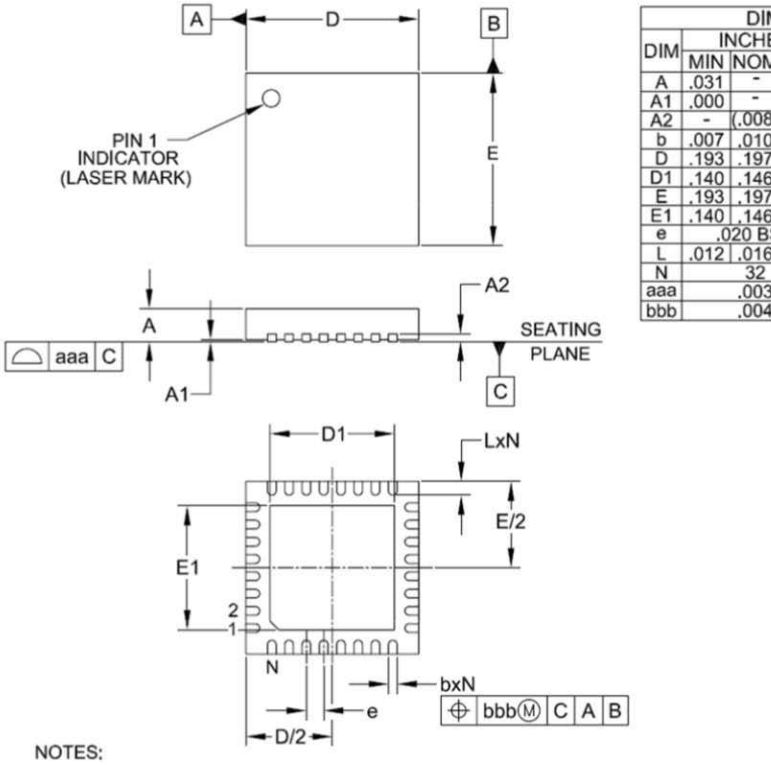
Refer to AN1200.102 LR2021 LoRa Performance for additional information.

Downloaded from Arrow.com.

24. Package Information

24.1 Package Outline Drawing

Figure 24-1: Package Outline Drawing



DIM	DIMENSIONS					
	INCHES			MILLIMETERS		
	MIN	NOM	MAX	MIN	NOM	MAX
A	.031	-	.039	0.80	-	1.00
A1	.000	-	.002	0.00	-	0.05
A2	-	(.008)	-	-	(0.20)	-
b	.007	.010	.012	0.18	0.25	0.30
D	.193	.197	.201	4.90	5.00	5.10
D1	.140	.146	.150	3.30	3.70	3.80
E	.193	.197	.201	4.90	5.00	5.10
E1	.140	.146	.150	3.30	3.70	3.80
e	.020 BSC			0.50 BSC		
L	.012	.016	.020	0.30	0.40	0.50
N	32			32		
aaa	.003			0.08		
bbb	.004			0.10		

NOTES:

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

Downloaded from Arrow.com.

24.2 Package Marking

Figure 24-2: Package Marking

TOP MARK	
CHAR	ROWS
6/6/6/6	4

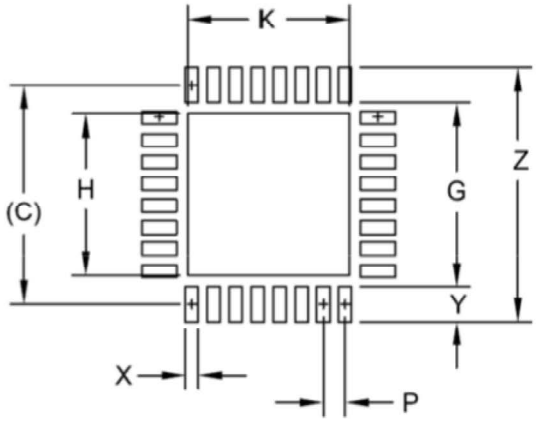
Marking for the 5 x 5mm MLPQ 32 Lead package:

nnnnnn = Part Number (Example: LR2021)
yyww = Date Code (2052)
xxxxxx = Semtech Lot No. (Example: E90100)

Downloaded from Arrow.com.

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*

The device uses a QFN32 5x5mm package, also named MLP package.

Downloaded from Arrow.com.

24.5 Thermal Information

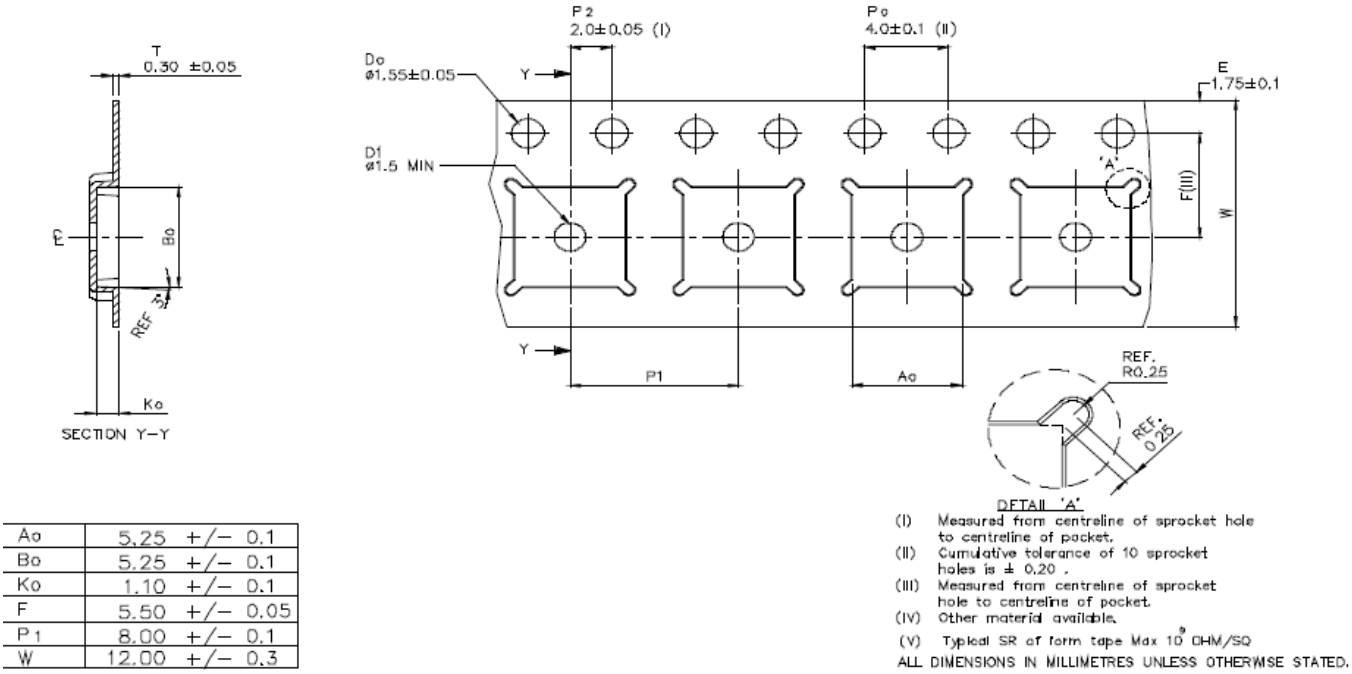
Table 24-1: Package Thermal Information

Name	Value[1]	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



Ao	5.25	+/−	0.1
Bo	5.25	+/−	0.1
Ko	1.10	+/−	0.1
F	5.50	+/−	0.05
P1	8.00	+/−	0.1
W	12.00	+/−	0.3

Downloaded from Arrow.com.

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
β	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

Downloaded from Arrow.com.

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 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

Downloaded from Arrow.com.

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®

Downloaded from Arrow.com.

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

Downloaded from Arrow.com.

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

Downloaded from Arrow.com.