



LLCC68 Datasheet

Long Range, Low Power, sub-GHz RF Transceiver

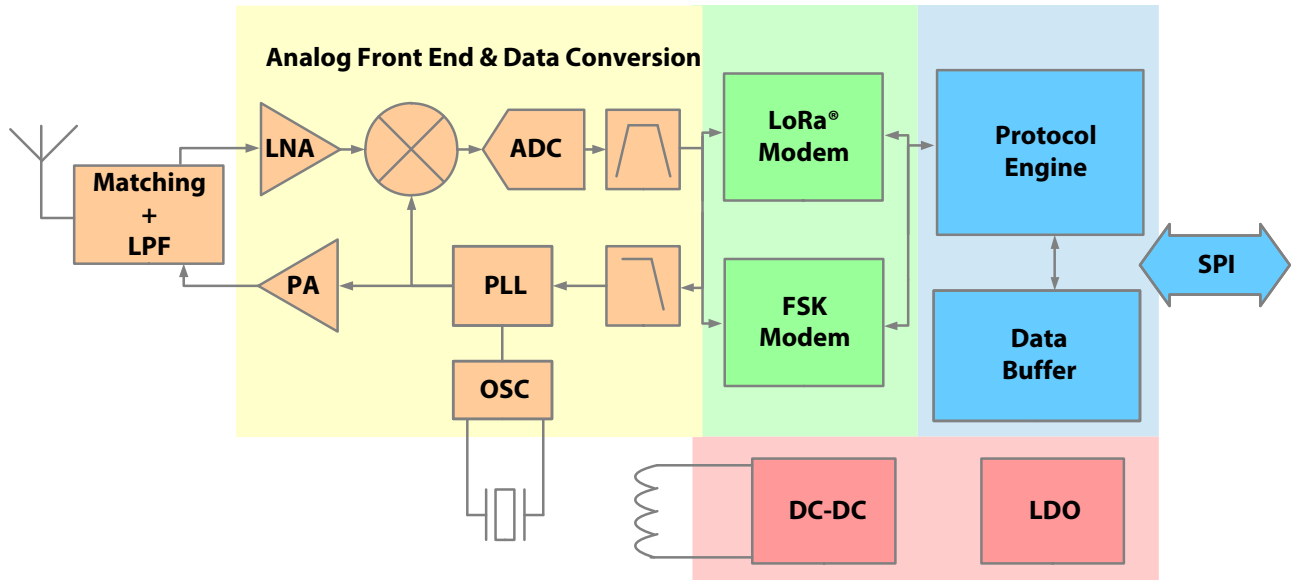


Figure A: LLCC68 Block Diagram

General Description

The LLCC68 sub-GHz radio transceiver is ideal for long range wireless applications. It is designed for long battery life with just 4.2mA of active receive current consumption. The LLCC68 can transmit up to +22dBm with highly efficient integrated power amplifiers.

The LLCC68 supports LoRa® modulation for LPWAN use cases and (G)FSK modulation for legacy use cases. The device is highly configurable to meet different application requirements utilizing the global LoRaWAN® standard or proprietary protocols.

The device is designed to comply with the physical layer requirements of the LoRaWAN specification released by the LoRa Alliance®.

The radio is suitable for systems targeting compliance with radio regulations including but not limited to ETSI EN 300 220, FCC CFR 47 Part 15, China regulatory requirements and the Japanese ARIB T-108. Continuous frequency coverage from 150MHz to 960MHz allows the support of all major sub-GHz ISM bands around the world.

Applications

The level of integration and the low consumption of the LLCC68 enable a new generation of Internet of Things applications.

- Smart meters
- Supply chain and logistics
- Building automation
- Agricultural sensors
- Smart cities
- Retail store sensors
- Asset tracking
- Street lights
- Parking sensors
- Environmental sensors
- Healthcare
- Safety and security sensors
- Remote control applications

Ordering Information

Part Number	Package Type	Delivery	Minimum Order Quantity
LLCC68IMLTRT	QFN	Tape & Reel	3,000 pieces

QFN 24 Package, Pb-free, Halogen free, RoHS/WEEE compliant product.

Revision History

Version	ECO	Date	Modifications
1.0	047881	July 2019	First Release
			<p>Modified Max val of BR_F to 500kb/s, FDA to 300kHz and updated Note Table 3-7</p> <p>Modified the Min value of ATCXO Table 3-7</p> <p>Reformatted Note in Max value of BW_F Table 3-8</p> <p>Added 500kb/s condition to RXS_2FB, BI_F and Note Table 3-8</p> <p>Trimming capacitors cannot be deactivated Section 4.1.3</p> <p>Added OCP max value Section 5.1</p> <p>GMSK is supported Section 6.</p> <p>Modified LoRa Time-on-air Section 6.1.4</p> <p>Added RSSI Calibration Section 6.1.6</p> <p>Added BW1000 to FSK packet bandwidth definitions and Note Table 6-5</p> <p>Added DIO3 as a Manual Switch Section 8.6</p> <p>Added pseudo-code for image calibration Section 9.2.1</p> <p>Added Rx gain details when entering sleep with retention Section 9.6</p> <p>Added registers to Table 12-1 between 0x0805 and 0x0891 for BW1000 usage, & LoRa Coding Rate RX and LoRa CRC Configuration Registers</p> <p>Removed Wake-up on RTC timeout Section 13.1.1</p> <p>Modified paDutyCycle description Section 13.1.14</p> <p>SetDio3AsTcxoCtrl min value Section 13.3.6</p> <p>GFSK Sync Word alignment information Section 13.4.6.1</p> <p>Added SetLoRaSymbNumTimeout valid values Section 13.4.9</p> <p>GetPacketStatus Section 13.5.3</p> <p>Added Settling time to GetRSSIInst Section 13.5.4</p> <p>Added LoRa coding rates with long interleave Table 13-49</p> <p>Added Tape and Reel Section 16.5</p>
1.1	067443	Dec 2024	

Table of Contents

1. Architecture.....	11
2. Pin Connection	12
2.1 I/O Description	12
2.2 Package View	13
3. Specifications.....	14
3.1 ESD Notice	14
3.2 Absolute Maximum Ratings	14
3.3 Operating Range	14
3.4 Crystal Specifications	15
3.5 Electrical Specifications	15
3.5.1 Power Consumption	16
3.5.2 General Specifications	18
3.5.3 Receive Mode Specifications	19
3.5.4 Transmit Mode Specifications	21
3.5.5 Digital I/O Specifications	21
4. Circuit Description	22
4.1 Clock References	22
4.1.1 RC Frequency References	22
4.1.2 High-Precision Frequency Reference.....	22
4.1.3 XTAL Control Block	23
4.1.4 TCXO Control Block.....	24
4.2 Phase-Locked Loop (PLL)	24
4.3 Receiver	25
4.3.1 Intermediate Frequencies	25
4.4 Transmitter	27
4.4.1 Power Amplifier Specifics	28
4.4.2 Power Amplifier Summary.....	30
5. Power Distribution	31
5.1 Selecting DC-DC Converter or LDO Regulation	31
5.1.1 Option A: DC-DC Regulator.....	32
5.1.2 Option B: LDO Regulator	32
5.1.3 Consideration on the DC-DC Inductor Selection	33
5.2 Flexible DIO Supply	33
6. Modems	34
6.1 LoRa® Modem	34
6.1.1 Modulation Parameter	34
6.1.2 LoRa Packet Engine.....	36
6.1.3 LoRa Frame	36
6.1.4 LoRa Time-on-Air	37
6.1.5 LoRa Channel Activity Detection (CAD)	37
6.1.6 RSSI Calibration.....	37

6.2 FSK Modem	39
6.2.1 Modulation Parameter	39
6.2.2 FSK Packet Engine	41
6.2.3 FSK Packet Format	42
6.2.4 Special Settings for BR= 0.6kbps and 1.2kbps	45
7. Data Buffer	46
7.1 Principle of Operation	46
7.2 Data Buffer in Receive Mode	47
7.3 Data Buffer in Transmit Mode	47
7.4 Using the Data Buffer	47
8. Digital Interface and Control	48
8.1 Reset	48
8.2 SPI Interface	48
8.2.1 SPI Timing When the Transceiver is in Active Mode	48
8.2.2 SPI Timing When the Transceiver Leaves Sleep Mode	49
8.3 Multi-Purpose Digital Input/Output (DIO)	50
8.3.1 BUSY Control Line	50
8.3.2 Digital Input/Output	52
8.4 Digital Interface Status versus Chip modes	52
8.5 IRQ Handling	53
8.6 DIO3 as a Manual Switch Control	54
9. Operational Modes	55
9.1 Startup	55
9.2 Calibration	55
9.2.1 Image Calibration for Specific Frequency Bands	56
9.3 Sleep Mode	56
9.4 Standby (STDBY) Mode	57
9.5 Frequency Synthesis (FS) Mode	57
9.6 Receive (RX) Mode	57
9.7 Transmit (TX) Mode	58
9.7.1 PA Ramping	58
9.8 Active Mode Switching Time	58
9.9 Transceiver Circuit Modes Graphic	59
10. Host Controller Interface	60
10.1 Command Structure	60
10.2 Transaction Termination	60
11. List of Commands	61
11.1 Operational Modes Commands	61
11.2 Register and Buffer Access Commands	62
11.3 DIO and IRQ Control	62
11.4 RF, Modulation and Packet Commands	62
11.5 Status Commands	63
12. Register Map	64
13. Commands Interface	66

13.1 Operational Modes Functions	66
13.1.1 SetSleep	66
13.1.2 SetStandby	67
13.1.3 SetFs	67
13.1.4 SetTx	68
13.1.5 SetRx	69
13.1.6 StopTimerOnPreamble	70
13.1.7 SetRxDutyCycle	71
13.1.8 SetCAD	73
13.1.9 SetTxContinuousWave	73
13.1.10 SetTxInfinitePreamble	73
13.1.11 SetRegulatorMode	74
13.1.12 Calibrate Function	74
13.1.13 CalibrateImage	75
13.1.14 SetPaConfig	75
13.1.15 SetRxTxFallbackMode	76
13.2 Registers and Buffer Access	77
13.2.1 WriteRegister Function	77
13.2.2 ReadRegister Function	77
13.2.3 WriteBuffer Function	77
13.2.4 ReadBuffer Function	78
13.3 DIO and IRQ Control Functions	78
13.3.1 SetDioIrqParams	78
13.3.2 IrqMask	78
13.3.3 GetIrqStatus	79
13.3.4 ClearIrqStatus	79
13.3.5 SetDIO2AsRfSwitchCtrl	80
13.3.6 SetDIO3AsTCXOCtrl	81
13.4 RF Modulation and Packet-Related Functions	82
13.4.1 SetRfFrequency	82
13.4.2 SetPacketType	82
13.4.3 GetPacketType	83
13.4.4 SetTxParams	83
13.4.5 SetModulationParams	84
13.4.6 SetPacketParams	87
13.4.7 SetCadParams	92
13.4.8 SetBufferBaseAddress	93
13.4.9 SetLoRaSymbNumTimeout	93
13.5 Communication Status Information	94
13.5.1 GetStatus	94
13.5.2 GetRxBufferStatus	95
13.5.3 GetPacketStatus	95
13.5.4 GetRssiInst	96
13.5.5 GetStats	97

13.5.6 ResetStats	97
13.6 Miscellaneous	98
13.6.1 GetDeviceErrors	98
13.6.2 ClearDeviceErrors	98
14. Application	99
14.1 HOST API Basic Read Write Function	99
14.2 Circuit Configuration for Basic Tx Operation	99
14.3 Circuit Configuration for Basic Rx Operation	100
14.4 Issuing Commands in the Right Order	100
14.5 Application Schematics	101
14.5.1 Application Design of the LLCC68 with RF Switch	101
15. Known Limitations	102
15.1 Modulation Quality with 500kHz LoRa Bandwidth	102
15.1.1 Description	102
15.1.2 Workaround	102
15.2 Better Resistance of the LLCC68 Tx to Antenna Mismatch	103
15.2.1 Description	103
15.2.2 Workaround	103
15.3 Implicit Header Mode Timeout Behavior	104
15.3.1 Description	104
15.3.2 Workaround	104
15.4 Optimizing the Inverted IQ Operation	104
15.4.1 Description	104
15.4.2 Workaround	104
16. Packaging Information	105
16.1 Package Outline Drawing	105
16.2 Package Marking	106
16.3 Land Pattern	106
16.4 Reflow Profiles	107
16.5 Tape and Reel Information	107

List of Figures

Figure 2-1: LLCC68 Top View Pin Location QFN 4x4 24L	13
Figure 4-1: LLCC68 Block Diagram	22
Figure 4-2: TCXO Control Block	24
Figure 4-3: PA Supply Scheme in DC-DC Mode	27
Figure 4-4: VR_PA versus Output Power on the LLCC68	28
Figure 4-5: Power Linearity on the LLCC68	29
Figure 4-6: Current versus Programmed Output Power on the LLCC68	29
Figure 5-1: LLCC68 Diagram with the DC-DC Regulator Power Option	32
Figure 5-2: LLCC68 Diagram with the LDO Regulator Power Option	32
Figure 5-3: Separate DIO Supply	33
Figure 6-1: LoRa® Signal Bandwidth	35
Figure 6-2: LoRa Packet Format	36
Figure 6-3: Fixed-Length Packet Format	42
Figure 6-4: Variable-Length Packet Format	42
Figure 6-5: Data Whitening LFSR	43
Figure 7-1: Data Buffer Diagram	46
Figure 8-1: SPI Timing Diagram	48
Figure 8-2: SPI Timing Transition	49
Figure 8-3: Switching Time Definition	50
Figure 8-4: Switching Time Definition in Active Mode	50
Figure 9-1: Transceiver Circuit Modes	59
Figure 13-1: Stopping Timer on Preamble or Header Detection	70
Figure 13-2: RX Duty Cycle Energy Profile	72
Figure 13-3: RX Duty Cycle when Receiving	72
Figure 14-1: Application Schematic of the LLCC68 with RF Switch	101
Figure 16-1: QFN 4x4mm Package Outline Drawing	105
Figure 16-2: LLCC68 Marking	106
Figure 16-3: QFN 4x4mm Land Pattern Drawing	106
Figure 16-4: Tape and Reel Drawing	107

List of Tables

Table 2-1: LLCC68 Pinout In QFN 4x4 24L.....	12
Table 3-1: ESD and Latch-up Notice	14
Table 3-2: Absolute Maximum Ratings	14
Table 3-3: Operating Range.....	14
Table 3-4: Crystal Specifications.....	15
Table 3-5: Power Consumption	16
Table 3-6: Power Consumption in Transmit Mode	17
Table 3-7: General Specifications	18
Table 3-8: Receive Mode Specifications.....	19
Table 3-9: Transmit Mode Specifications.....	21
Table 3-10: Digital I/O Specifications	21
Table 4-1: Internal Foot Capacitor Configuration.....	23
Table 4-2: Intermediate Frequencies in FSK Mode	25
Table 4-3: Intermediate Frequencies in LoRa Mode	26
Table 4-4: Power Amplifier Summary	30
Table 5-1: Regulation Type versus Circuit Mode.....	31
Table 5-2: OCP Configuration	31
Table 5-3: Recommended 15μH Inductors.....	33
Table 6-1: Range of Spreading Factors (SF)	35
Table 6-2: Signal Bandwidth Setting in LoRa® Mode.....	35
Table 6-3: Coding Rate Overhead.....	36
Table 6-4: Registers and Values.....	38
Table 6-5: Bandwidth Definition in FSK Packet Type.....	39
Table 6-6: Whitening Initial Value	43
Table 6-7: CRC Type Configuration.....	44
Table 6-8: CRC Initial Value	44
Table 6-9: CRC Polynomial.....	44
Table 8-1: SPI Timing Requirements.....	49
Table 8-2: Switching Time	51
Table 8-3: Digital Pads Configuration for each Chip Mode	52
Table 8-4: IRQ Status Registers	53
Table 9-1: LLCC68 Operating Modes	55
Table 9-2: Image Calibration Over the ISM Bands.....	56
Table 9-3: Rx Gain Configuration.....	57
Table 10-1: SPI Interface Command Sequence	60
Table 11-1: Commands Selecting the Operating Modes of the Radio	61
Table 11-2: Commands to Access the Radio Registers and FIFO Buffer.....	62
Table 11-3: Commands Controlling the Radio IRQs and DIOs.....	62
Table 11-4: Commands Controlling the RF and Packets Settings.....	62
Table 11-5: Commands Returning the Radio Status.....	63
Table 12-1: List of Registers	64
Table 13-1: SetSleep SPI Transaction	66
Table 13-2: Sleep Mode Definition.....	66
Table 13-3: SetConfig SPI Transaction.....	67
Table 13-4: STDBY Mode Configuration	67
Table 13-5: SetFs SPI Transaction.....	67
Table 13-6: SetTx SPI Transaction	68
Table 13-7: SetTx Timeout Duration	68

Table 13-8: SetRx SPI Transaction	69
Table 13-9: SetRx Timeout Duration	69
Table 13-10: StopTimerOnPreamble SPI Transaction	70
Table 13-11: StopOnPreambleParam Definition	70
Table 13-12: SetRxDutyCycle SPI Transaction	71
Table 13-13: SetCAD SPI Transaction	73
Table 13-14: SetTxContinuousWave SPI Transaction	73
Table 13-15: SetTxInfinitePreamble SPI Transaction	73
Table 13-16: SetRegulatorMode SPI Transaction	74
Table 13-17: Calibrate SPI Transaction	74
Table 13-18: Calibration Setting	74
Table 13-19: CalibrateImage SPI Transaction	75
Table 13-20: SetPaConfig SPI Transaction	75
Table 13-21: PA Operating Modes with Optimal Settings	76
Table 13-22: SetRxTxFallbackMode SPI Transaction	76
Table 13-23: FallbackMode Definition	76
Table 13-24: WriteRegister SPI Transaction	77
Table 13-25: ReadRegister SPI Transaction	77
Table 13-26: WriteBuffer SPI Transaction	77
Table 13-27: ReadBuffer SPI Transaction	78
Table 13-28: SetDiolrqParams SPI Transaction	78
Table 13-29: IRQ Registers	79
Table 13-30: GetIrqStatus SPI Transaction	79
Table 13-31: ClearIrqStatus SPI Transaction	79
Table 13-32: SetDIO2AsRfSwitchCtrl SPI Transaction	80
Table 13-33: Enable Configuration Definition	80
Table 13-34: SetDIO3asTCXOctrl SPI Transaction	81
Table 13-35: tcxoVoltage Configuration Definition	81
Table 13-36: SetRfFrequency SPI Transaction	82
Table 13-37: SetPacketType SPI Transaction	82
Table 13-38: PacketType Definition	82
Table 13-39: GetPacketType SPI Transaction	83
Table 13-40: SetTxParams SPI Transaction	83
Table 13-41: RampTime Definition	83
Table 13-42: SetModulationParams SPI Transaction	84
Table 13-43: GFSK ModParam1, ModParam2 & ModParam3 - br	84
Table 13-44: GFSK ModParam4 - PulseShape	84
Table 13-45: GFSK ModParam5 - Bandwidth	85
Table 13-46: GFSK ModParam6, ModParam7 & ModParam8 - Fdev	85
Table 13-47: LoRa ModParam1- SF	86
Table 13-48: LoRa ModParam2 - BW	86
Table 13-49: LoRa ModParam3 - CR	86
Table 13-50: LoRa ModParam4 - LowDataRateOptimize	86
Table 13-51: SetPacketParams SPI Transaction	87
Table 13-52: GFSK PacketParam1 & PacketParam2 - PreambleLength	87
Table 13-53: GFSK PacketParam3 - PreambleDetectorLength	87
Table 13-54: GFSK PacketParam4 - SyncWordLength	87
Table 13-55: Sync Word Programming	88
Table 13-56: GFSK PacketParam5 - AddrComp	88
Table 13-57: Node Address Programming	88
Table 13-58: Broadcast Address Programming	88

Table 13-59: GFSK PacketParam6 - PacketType	89
Table 13-60: GFSK PacketParam7 - PayloadLength	89
Table 13-61: GFSK PacketParam8 - CRCType	89
Table 13-62: CRC Initial Value Programming	89
Table 13-63: CRC Polynomial Programming	89
Table 13-64: GFSK PacketParam9 - Whitening	90
Table 13-65: Whitening Initial Value	90
Table 13-66: LoRa PacketParam1 & PacketParam2 - PreambleLength	91
Table 13-67: LoRa PacketParam3 - HeaderType	91
Table 13-68: LoRa PacketParam4 - PayloadLength	91
Table 13-69: LoRa PacketParam5 - CRCType	91
Table 13-70: LoRa PacketParam6 - InvertIQ	91
Table 13-71: SetCadParams SPI Transaction	92
Table 13-72: CAD Number of Symbol Definition	92
Table 13-73: CAD Exit Mode Definition	92
Table 13-74: SetBufferBaseAddress SPI Transaction	93
Table 13-75: SetLoRaSymbNumTimeout SPI Transaction	93
Table 13-76: Status Bytes Definition	94
Table 13-77: GetStatus SPI Transaction	94
Table 13-78: GetRxBuffersStatus SPI Transaction	95
Table 13-79: GetPacketStatus SPI Transaction	95
Table 13-80: Status Fields	95
Table 13-81: GetRssiInst SPI Transaction	96
Table 13-82: RSSI Settling Time per Bandwidth	96
Table 13-83: GetStats SPI Transaction	97
Table 13-84: ResetStats SPI Transaction	97
Table 13-85: GetDeviceErrors SPI Transaction	98
Table 13-86: OpError Bits	98
Table 13-87: ClearDeviceErrors SPI Transaction	98

1. Architecture

The LLCC68 is a half-duplex transceiver capable of low power operation in the 150-960MHz [ISM](#) frequency band. The radio comprises four main blocks:

1. **Analog Front End:** The transmit and receive chains, as well as the data converter interface to ensuing digital blocks. The LLCC68 transceiver is capable of delivering up to +22dBm with the battery supply.
2. **Digital Modem Bank:** a range of modulation options is available in the LLCC68:
 - ♦ LoRa Rx/Tx, [BW](#) = 125 -250 - 500kHz
 - ♦ LoRa [SF](#) = 5 - 6 - 7 - 8 - 9 for [BW](#) = 125kHz
 - ♦ LoRa [SF](#) = 5 - 6 - 7 - 8 - 9 - 10 for [BW](#) = 250kHz
 - ♦ LoRa [SF](#) = 5 - 6 - 7 - 8 - 9 - 10 - 11 for [BW](#) = 500kHz
 - ♦ (G)[FSK](#) Rx/Tx, with [BR](#) = 0.6 - 500kb/s
3. **Digital Interface and Control:** this comprises all payload data and protocol processing as well as access to configuration of the radio via the [SPI](#) interface.
4. **Power Distribution:** two forms of voltage regulation, DC-DC or linear regulator [LDO](#), are available depending upon the design priorities of the application.

2. Pin Connection

2.1 I/O Description

Table 2-1: LLCC68 Pinout In QFN 4x4 24L

Pin Number	Pin Name	Type (I = Input O = Output)	Description
0	GND	-	Exposed Ground pad
1	VDD_IN	I	Input voltage for power amplifier regulator, VR_PA: connected to pin 10
2	GND	-	Ground
3	XTA	-	Crystal oscillator connection, can be used to input external reference clock
4	XTB	-	Crystal oscillator connection
5	GND	-	Ground
6	DIO3	I/O	Multi-purpose digital I/O - external TCXO supply voltage
7	VREG	O	Regulated output voltage from the internal regulator LDO / DC-DC
8	GND	-	Ground
9	DCC_SW	O	DC-DC Switcher Output
10	VBAT	I	Supply for the RFIC
11	VBAT_IO	I	Supply for the Digital I/O interface pins (except DIO3)
12	DIO2	I/O	Multi-purpose digital I/O / RF Switch control
13	DIO1	I/O	Multi-purpose digital IO
14	BUSY	O	Busy indicator
15	NRESET	I	Reset signal, active low
16	MISO	O	SPI slave output
17	MOSI	I	SPI slave input
18	SCK	I	SPI clock
19	NSS	I	SPI Slave Select
20	GND	-	Ground
21	RFI_P	I	RF receiver input
22	RFI_N	I	RF receiver input
23	RFO	O	RF transmitter output
24	VR_PA	-	Regulated power amplifier supply

2.2 Package View

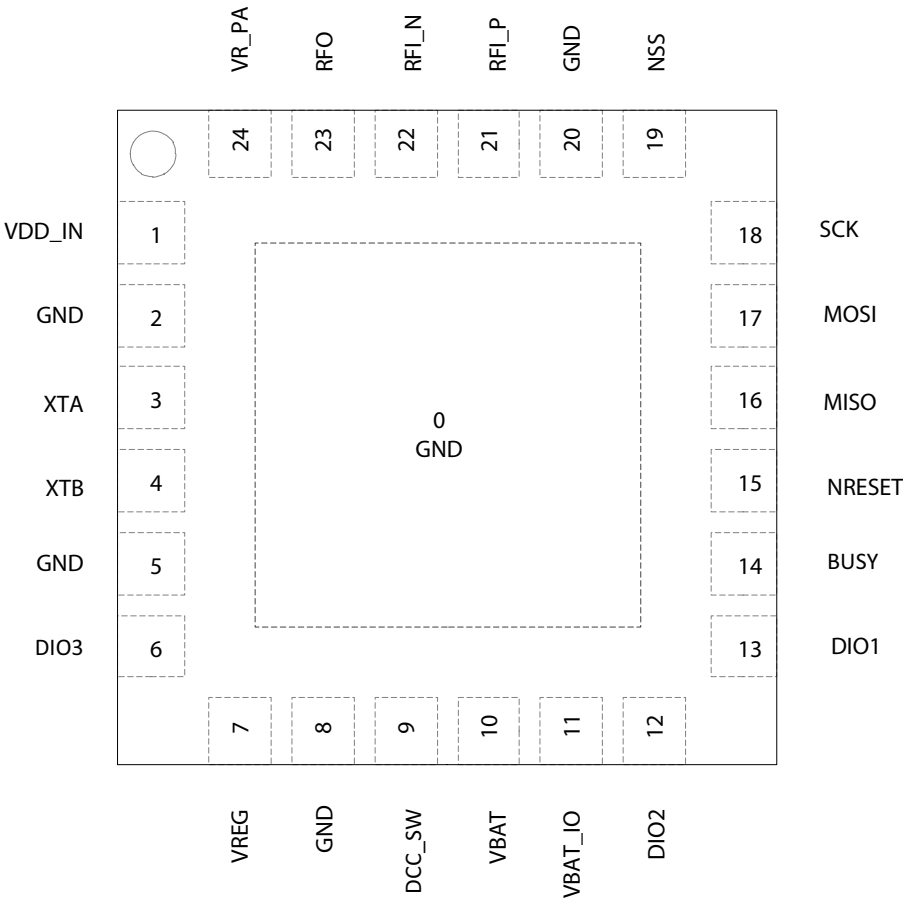


Figure 2-1: LLCC68 Top View Pin Location QFN 4x4 24L

3. Specifications



3.1 ESD Notice

The LLCC68 transceiver is a high-performance radio frequency device, with high ESD and latch-up resistance. The chip should be handled with all the necessary ESD precautions to avoid any permanent damage.

Table 3-1: ESD and Latch-up Notice

Symbol	Description	Min	Typ	Max	Unit
ESD_HBM	Class 2 of ANSI/ESDA/JEDEC Standard JS-001-2014 (Human Body Model)	-	-	2.0	kV
ESD_CDM	ESD Charged Device Model, JEDEC standard JESD22-C101D, class III	-	-	1000	V
LU	Latch-up, JEDEC standard JESD78 B, class I level A	-	-	100	mA

3.2 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, reducing product life time.

Table 3-2: Absolute Maximum Ratings

Symbol	Description	Min	Typ	Max	Unit
VDDmr	Supply voltage, applies to VBAT and VBAT_IO	-0.5	-	3.9	V
Tmr	Temperature	-55	-	125	°C
Pmr	RF Input level	-	-	10	dBm

3.3 Operating Range

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

Table 3-3: Operating Range

Symbol	Description	Min	Typ	Max	Unit
VDDop	Supply voltage, applies to VBAT and VBAT_IO	1.8	-	3.7	V
Top	Temperature under bias	-40	-	85	°C
Clop	Load capacitance on digital ports	-	-	20	pF
ML	RF Input power	-	-	0	dBm
VSWR	Voltage Standing Wave Ratio at antenna port	-	-	10:1	-

3.4 Crystal Specifications

Table 3-4: Crystal Specifications

Symbol	Description	Min	Typ	Max	Unit
FXOSC	Crystal oscillator frequency	-	32	-	MHz
CLOAD	Crystal load capacitance	-	10	-	pF
COXTAL	Crystal shunt capacitance	0.3	0.6	2	pF
RSXTAL	Crystal series resistance	-	30	60	Ω
CMXTAL	Crystal motional capacitance	1.3	1.89	2.5	fF
DRIVE	Drive level	-	-	100	μ W

The reference frequency accuracy is defined by the complete system, and should take into account precision of the transmitter and the receiver, as well as environmental parameters such as extreme temperature limits. In a LoRaWAN system, the expected reference frequency accuracy on the end-device should be about +/- 30ppm under all operating conditions. This includes initial error, temperature drift and ageing over the lifetime of the product.

3.5 Electrical Specifications

The electrical specifications are given with the following conditions unless otherwise specified:

- VBAT_IO = VBAT = 3.3V, all current consumptions are given for VBAT connected to VBAT_IO
- Temperature = 25°C
- FXOSC = 32MHz, with specified crystal
- F_{RF} = 434/490/868/915 MHz
- All RF impedances matched
- Transmit mode output power defined into a 50 Ω load impedance
- FSK BER = 0.1%, 2-level FSK modulation without pre-filtering, BR = 4.8kb/s, FDA = \pm 5kHz, BW_F = 20kHz double-sided
- LoRa PER = 1%, packet 64 bytes, preamble 8 symbols, CR = 4/5, CRC on payload enabled, explicit header mode
- RX/TX specifications given using default RX gain step and direct tie connection between Rx and Tx
- Blocking immunity, ACR and co-channel rejection are given for a single tone (CW) interferer and referenced to sensitivity +3dB
- Optional TCXO and RF Switch power consumption always excluded

Caution!

Throughout this document, all receiver bandwidths are expressed as “double-sideband”. This is valid for LoRa and FSK modulations.

3.5.1 Power Consumption

Table 3-5: Power Consumption

Symbol	Mode	Conditions	Min	Typ	Max	Unit
IDDOFF	OFF mode (SLEEP mode with cold start ¹)	All blocks off	-	160	-	nA
IDDSL	SLEEP mode (SLEEP mode with warm start ²)	Configuration retained	-	600	-	nA
		Configuration retained + RC64k	-	1.2	-	μA
IDDSBR	STDBY_RC mode	RC13M, XOSC OFF	-	0.6	-	mA
IDDSBX	STDBY_XOSC mode	XOSC ON	-	0.8	-	mA
IDDFS	Synthesizer mode	DC-DC mode used	-	2.1	-	mA
		LDO mode used	-	3.55	-	mA
IDDRX	Receive mode DC-DC mode used	FSK 4.8kb/s	-	4.2	-	mA
		LoRa 125kHz	-	4.6	-	mA
		Rx Boosted ³ , FSK 4.8kb/s	-	4.8	-	mA
		Rx Boosted, LoRa 125kHz	-	5.3	-	mA
		LoRa 125kHz, VBAT = 1.8V	-	8.2	-	mA
		FSK 4.8kb/s	-	8	-	mA
		LoRa 125kHz	-	8.8	-	mA
		Rx Boosted, FSK 4.8kb/s	-	9.3	-	mA
		Rx Boosted, LoRa 125kHz	-	10.1	-	mA

1. Cold start is equivalent to the device at POR or when it wakes up from Sleep mode with all blocks OFF, see [Section 13.1.1 "SetSleep" on page 66](#)

2. Warm start only happens when the device is woken from Sleep mode with its configuration retained, see [Section 13.1.1 "SetSleep" on page 66](#)

3. For more details on how to set the device in Rx Boosted gain mode, see [Section 9.6 "Receive \(RX\) Mode" on page 57](#)

Table 3-6: Power Consumption in Transmit Mode

Symbol	Frequency Band	PA Match / Condition	Power Output	Typical	Unit	
IDDTX ¹	868/915MHz	+22dBm	+22dBm	118	mA	
			+20dBm	102	mA	
			+17dBm	95	mA	
			+14dBm	90	mA	
		+20dBm / optimal settings ²		+20dBm	84	mA
		+17dBm / optimal settings ²		+17dBm	58	mA
		+14dBm / optimal settings ²		+14dBm	45	mA
		434/490MHz	+22dBm	+22dBm	107	mA
	+20dBm			90	mA	
	+17dBm			75	mA	
	+14dBm			63	mA	
	+20dBm / optimal settings ²		+20dBm	65	mA	
	+17dBm / optimal settings ²		+17dBm	42	mA	
	+14dBm / optimal settings ²		+14dBm	32	mA	

1. DC-DC mode is used for the IC core but the PA is supplied from VBAT. For more details, see [Section 5.1 "Selecting DC-DC Converter or LDO Regulation" on page 31](#)

2. Optimal settings adapted to the specified output power. For more details, see [Section 13.1.14.1 "PA Optimal Settings" on page 76](#)

3.5.2 General Specifications

Table 3-7: General Specifications

Symbol	Description	Conditions	Min	Typ	Max	Unit
FR	Synthesizer frequency range		150	-	960	MHz
FSTEP	Synthesizer frequency step	Bit 2 of <i>TxModulation</i> = 1 (main use) Bit 2 of <i>TxModulation</i> = 0 (see Section 15.)	- -	0.95 122	- -	Hz Hz
PHN ^{1 2}	Synthesizer phase noise (for 868 / 915MHz)	1kHz offset	-	-75	-	dBc/Hz
		10kHz offset	-	-95	-	dBc/Hz
		100kHz offset	-	-100	-	dBc/Hz
		1MHz offset	-	-120	-	dBc/Hz
		10MHz offset	-	-135	-	dBc/Hz
TS_FS	Synthesizer wake-up time	From STDBY_XOSC mode	-	40	-	μs
TS_OSC	Crystal oscillator wake-up time	from STDBY_RC ³	-	150	-	μs
OSC_TRM	Crystal oscillator trimming range for crystal frequency error compensation ⁴		+/-15	+/-30	-	ppm
BR_F	Bit rate, FSK	Programmable Minimum modulation index is 0.5	0.6	-	500 ⁵	kb/s
FDA	Frequency deviation, FSK	Programmable	0.6	-	300	kHz
BR_L	Bit rate LoRa	Min. for SF9, BW_L = 125kHz Max. for SF5, BW_L = 500kHz	1.76	-	62.5 ⁶	kb/s
BW_L	Signal BW , LoRa	Programmable	125	-	500 ⁶	kHz
SF	Spreading factor for LoRa	Programmable, chips/symbol = 2^SF	5	-	9 ⁷ 10 ⁸ 11 ⁹	-
VTCXO	Regulated voltage range for TCXO voltage supply	Min/Max values in typical conditions, Typ value for default setting VDDop > VTCXO + 200mV	1.6	1.7	3.3	V
ILTCXO	Load current for TCXO regulator		-	1.5	4	mA
TSVTCXO	Start-up time for TCXO regulator	From enable to regulated voltage within 25mV from target	-	-	100	μs
IDDTCXO	Current consumption of the TCXO regulator	Quiescent current Relative to load current	- -	- 1	70 2	μA %
ATCXO	Amplitude voltage for external TCXO applied to XTA pin	Provided through a 220Ω resistor in series with a 10pF capacitance. See Section 4.1.4 "TCXO Control Block" on page 24	0.2	0.6	1.2	Vpk-pk

1. Phase Noise specifications are given for the recommended PLL BW to be used for the specific modulation/BR, optimized settings may be used for specific applications.
2. Phase Noise is not constant over frequency, due to the topology of the PLL, for two frequencies close to each other, the phase noise could change significantly.
3. Wake-up time till crystal oscillator frequency is within +/- 10ppm.
4. OSC_TRIM is the available trimming range to compensate for crystal initial frequency error and to allow crystal temperature compensation implementation; the total available trimming range is higher and allows the compensation for all IC process variations.

5. Maximum bit rate is assumed to scale with RF frequency; for example 500kb/s in the 869/915MHz frequency bands and only 50kb/s at 150MHz. .
Please contact Semtech representative for more details.
6. For RF frequencies below 400MHz, there is a scaling between the frequency and supported BW, some BW may not be available below 400MHz
7. LoRa BW = 125kHz.
8. LoRa BW = 250kHz.
9. LoRa BW = 500kHz.

3.5.3 Receive Mode Specifications

Table 3-8: Receive Mode Specifications (Sheet 1 of 2)

Symbol	Description	Conditions	Min	Typ	Max	Unit
RXS_2FB	Sensitivity 2-FSK, RX Boosted gain, see Section 9.6 "Receive (RX) Mode" on page 57, split RF paths for Rx and Tx, RF switch insertion loss excluded	BR_F = 0.6kb/s, FDA = 0.8kHz, BW_F = 4kHz	-	-125	-	dBm
		BR_F = 1.2kb/s, FDA = 5kHz, BW_F = 20kHz	-	-123	-	dBm
		BR_F = 4.8kb/s, FDA = 5kHz, BW_F = 20kHz	-	-118	-	dBm
		BR_F = 38.4kb/s, FDA = 40kHz, BW_F = 160kHz	-	-109	-	dBm
		BR_F = 250kb/s, FDA = 125kHz, BW_F = 500kHz	-	-104	-	dBm
		BR_F = 500kb/s, FDA = 300kHz, BW_F = 1000kHz ¹	-	-94	-	dBm
RXS_LB	Sensitivity LoRa, Rx Boosted gain, see Section 9.6 "Receive (RX) Mode" on page 57, split RF paths for Rx and Tx, RF switch insertion loss excluded	BW_L = 125kHz, SF = 7	-	-124	-	dBm
		BW_L = 125kHz, SF = 9	-	-129	-	dBm
		BW_L = 250kHz, SF = 7	-	-121	-	dBm
		BW_L = 250kHz, SF = 10	-	-129	-	dBm
		BW_L = 500kHz, SF = 7	-	-117	-	dBm
		BW_L = 500kHz, SF = 11	-	-127	-	dBm
RXS_2F	Sensitivity 2-FSK Rx Power Saving gain with direct tie connection between Rx and Tx	BR_F = 4.8kb/s, FDA = 5kHz, BW_F = 20kHz	-	-115	-	dBm
RXS_L	Sensitivity LoRa Rx Power Saving gain with direct tie connection between Rx and Tx	BW_L = 125kHz, SF = 9	-	-125	-	dBm
CCR_F	Co-channel rejection, FSK		-	-9	-	dB
CCR_L	Co-channel rejection, LoRa	SF = 7	-	5	-	dB
		SF = 11	-	16	-	dB
ACR_F	Adjacent channel rejection, FSK	Offset = +/- 50kHz	-	45	-	dB
ACR_L	Adjacent channel rejection, LoRa	Offset = +/- 1.5 x BW_L				
		BW_L = 125kHz, SF = 7	-	60	-	dB
		BW_L = 125kHz, SF = 9	-	65	-	dB
BI_F	Blocking immunity, FSK	BR_F = 4.8kb/s, FDA = 5kHz, BW_F = 20kHz				
		Offset = +/- 1MHz	-	68	-	dB
		Offset = +/- 2MHz	-	70	-	dB
		Offset = +/- 10MHz	-	80	-	dB
		BR_F = 500kb/s, FDA = 300kHz, BW_F = 1000kHz				
		Offset = +/- 1MHz	-	45	-	dB
		Offset = +/- 2MHz	-	50	-	dB
		Offset = +/- 10MHz	-	60	-	dB
BI_L	Blocking immunity, LoRa	BW_L = 125kHz, SF = 9				
		Offset = +/- 1MHz	-	80	-	dB
		Offset = +/- 2MHz	-	82	-	dB
		Offset = +/- 10MHz	-	91	-	dB

Table 3-8: Receive Mode Specifications (Sheet 2 of 2)

Symbol	Description	Conditions	Min	Typ	Max	Unit
IIP3	3rd order input intercept point	Unwanted tones are 1MHz and 1.96MHz above LO	-	-5	-	dBm
IMA	Image attenuation	Without IQ calibration	-	35	-	dB
		With IQ calibration	-	54	-	dB
BW_F	DSB channel filter BW, FSK	Programmable, typical values	4.8	-	467 ⁽¹⁾	kHz
TS_RX	Receiver wake-up time	FS to RX	-	41	-	μs
FERR_L	Maximum tolerated frequency offset between transmitter and receiver, no sensitivity degradation, SF5 to SF11	All bandwidths, ±25% of BW The tighter limit applies (see below)		±25%		BW
	Maximum tolerated frequency offset between transmitter and receiver, no sensitivity degradation, SF10 to SF11	SF11	-100	-	100	ppm
		SF10	-200	-	200	ppm

1. Refer to Section 6.2.1 for BW-F = 1000kHz configuration

3.5.4 Transmit Mode Specifications

Table 3-9: Transmit Mode Specifications

Symbol	Description	Conditions	Min	Typ	Max	Unit
TXOP	Maximum RF output power	Highest power step setting	-	+22	-	dBm
TXDRP	RF output power drop versus supply voltage	+22dBm, VBAT = 2.7V	-	2	-	dB
		+22dBm, VBAT = 2.4V	-	3	-	dB
		+22dBm, VBAT = 1.8V	-	6	-	dB
TXPRNG	RF output power range	Programmable in 31 steps, typical value	TXOP-31	-	TXOP	dBm
TXACC	RF output power step accuracy		-	± 2	-	dB
TXRMP	Power amplifier ramping time	Programmable	10	-	3400	μs
TS_TX	Tx wake-up time	Frequency Synthesizer enabled	-	36 + PA ramping	-	μs

3.5.5 Digital I/O Specifications

Table 3-10: Digital I/O Specifications

Symbol	Description	Conditions	Min	Typ	Max	Unit
VIH	Input High Voltage	-	$0.7 \cdot V_{BAT_IO}^1$	-	$V_{BAT_IO}^1 + 0.3$	V
VIL	Input Low Voltage	-	-0.3	-	$0.3 \cdot V_{BAT_IO}^1$	V
VIL_N	Input Low Voltage for pin NRESET	-	-0.3	-	$0.2 \cdot V_{BAT}$	V
VOH	Output High Voltage	$I_{max} = -2.5mA$	$0.9 \cdot V_{BAT_IO}^1$	-	$V_{BAT_IO}^1$	V
VOL	Output Low Voltage	$I_{max} = 2.5mA$	0	-	$0.1 \cdot V_{BAT_IO}^1$	V
Ileak	Digital input leakage current (NSS, MOSI, SCK)	-	-1	-	1	μA

1. Excluding pin DIO3, which is referenced to VBAT.

4. Circuit Description

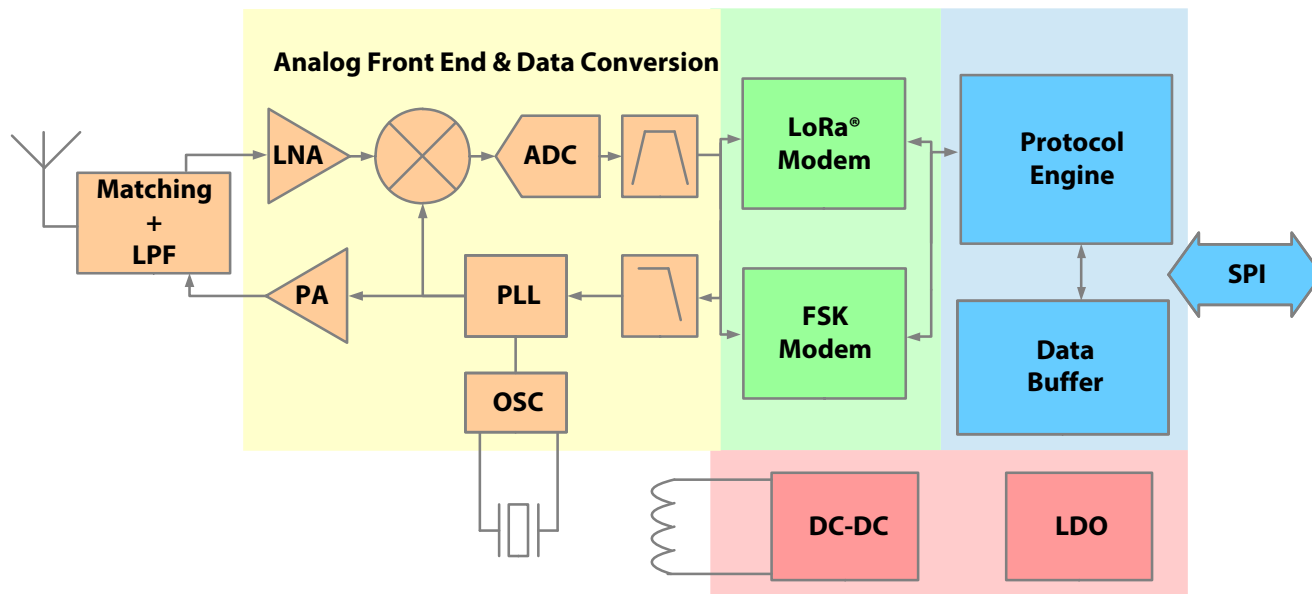


Figure 4-1: LLCC68 Block Diagram

The LLCC68 half-duplex RF transceiver operates in the sub-GHz frequency bands which support constant envelope modulation schemes such as LoRa and FSK constant envelope modulation schemes.

4.1 Clock References

4.1.1 RC Frequency References

Two RC oscillators are available: 64kHz and 13MHz RC oscillators. The 64kHz RC oscillator ([RC64k](#)) is optionally used by the circuit in SLEEP mode to wake-up the transceiver when performing periodic or duty cycled operations. Several commands make use of this 64kHz RC oscillator (called [RTC](#) across this document) to generate time-based events. The 13MHz RC oscillator ([RC13M](#)) is enabled for all SPI communications to permit configuration of the device without the need to start the crystal oscillator. Both RC oscillators are supplied directly from the battery.

4.1.2 High-Precision Frequency Reference

In the LLCC68, the high-precision frequency reference can come either from an on-chip crystal oscillator (OSC) using an external crystal resonator or from an external [TCXO](#) (Temperature Compensated Crystal Oscillator), supplied by an internal regulator.

The LLCC68 comes in a small form factor 4 x 4 mm QFN package and is able to transmit up to +22dBm. When in transmit mode the circuit may heat up depending on the output power and current consumption. Careful PCB design using thermal isolation techniques must be applied between the circuit and the crystal resonator to avoid transferring the heat to the external crystal resonator.

When using the LoRa modulation with LowDataRateOptimize set to 0x00 (see [Section Table 13-50: "LoRa ModParam4 - LowDataRateOptimize" on page 86](#)), the total frequency drift over the packet transmission time should be minimized and kept lower than $Freq_drift_max$:

$$Freq_drift_max = \frac{BW_L}{3 * 2^{SF}}$$

When possible, using LowDataRateOptimize set to 0x01 significantly relaxes the total frequency drift over the packet transmission requirement to $16 \times Freq_drift_max$.

Note: Recommendations for heat dissipation techniques to be applied to the PCB designs are given in detail in the application note AN1200.37 "Recommendations for Best Performance" on www.semtech.com.

In miniaturized design implementations where heat dissipations techniques cannot be implemented or the use of the LowDataRateOptimize is not supported, the use of a TCXO provides a more stable clock reference.

4.1.3 XTAL Control Block

The LLCC68 does not require the user to set external foot capacitors on the XTAL supplying the 32MHz clock. Indeed, the device is fitted with internal programmable capacitors connected independently to the pins XTA and XTB of the device. Each capacitor can be set independently, balanced or unbalanced to each other, by 0.47pF typical steps, but they cannot be deactivated.

Table 4-1: Internal Foot Capacitor Configuration

Pin	Register Address	Typical Values
XTA	0x0911	Each capacitor can be controlled independently in steps of 0.47pF added to the minimal value: <ul style="list-style-type: none"> •0x00 sets the trimming cap to 11.3pF (minimum) •0x2F sets the trimming cap to 33.4pF (maximum)
XTB	0x0912	

Note: When using an XTAL:

At POR or when waking-up from Sleep in cold start mode, the trimming cap registers are initialized at the value 0x05 (13.6pF). Once the device is set in STDBY_XOSC mode, the internal state machine overwrites both registers to the value 0x12 (19.7pF). Therefore, the user must ensure the device is already in STDBY_XOSC mode before changing the trimming cap values so that they are not overwritten by the state machine.

Note: When using a TCXO:

Once the command SetDIO3AsTCXOCtrl(. . .) is sent to the device, the register controlling the internal cap on XTA is automatically changed to 0x2F (33.4pF) to filter any spurious transitions which could occur and be propagated to the PLL.

4.1.4 TCXO Control Block

Under certain circumstances, typically small form factor designs with reduced heat dissipation or environments with extreme temperature variation, it may be required to use a **TCXO** (Temperature Compensated Crystal Oscillator) to achieve better frequency accuracy. This depends on the complete system, transmitter and receiver. The specification FERR_L in [Section Table 3-8: "Receive Mode Specifications"](#) on [page 19](#) provides information on the maximum tolerated frequency offset for optimal receiver performance.

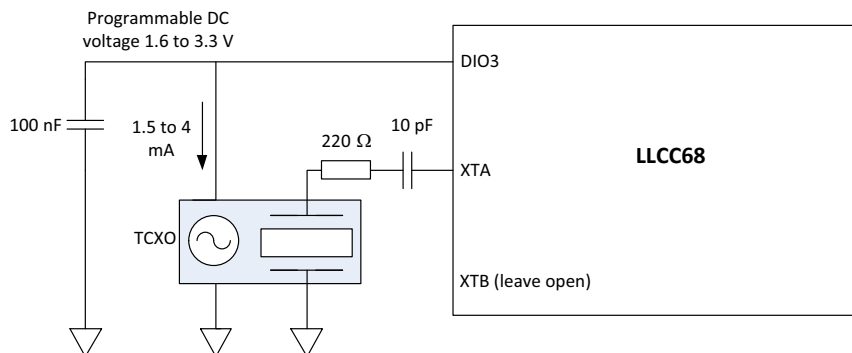


Figure 4-2: TCXO Control Block

When a TCXO is used, it should be connected to pin 3 XTA, through a 220Ω resistor and a 10pF DC-cut capacitor. Pin 4 XTB should be left not connected. Pin 6 DIO3 can be used to provide a regulated DC voltage to power the TCXO, programmable from 1.6 to 3.3V. VBAT should always be 200mV higher than the programmed voltage to ensure proper operation. AN1200.59 provides guidelines for the selection of 32MHz TCXOs and associated components.

The nominal current drain is 1.5mA, but the regulator can support up to 4mA of load. A clipped-sine-wave output TCXO is required, with the output amplitude not exceeding 1.2V peak-to-peak. The commands to enable TCXO mode are described in [Section 13.3.6 "SetDIO3AsTCXOctrl"](#) on [page 81](#), and that includes DC voltage and timing information.

Note: A complete Reset of the chip as described in [Section 8.1 "Reset"](#) on [page 48](#) is required to get back to normal XOSC operation, after the chip has been set to TCXO mode with the command SetDIO3AsTCXOctrl.

4.2 Phase-Locked Loop (PLL)

A fractional-N third order sigma-delta **PLL** acts as the frequency synthesizer for the **LO** of both receiver and transmitter chains. The LLCC68 can cover continuously all the sub-GHz frequency range 150-960MHz. The PLL is capable of auto-calibration and has low switching-on or hopping times. Frequency modulation is performed inside the PLL bandwidth. The PLL frequency is derived from the crystal oscillator circuit which uses an external 32MHz crystal reference.

4.3 Receiver

The received RF signal is first amplified by a differential Low Noise Amplifier (LNA), then down-converted to low-IF intermediate frequency by mixers operating in quadrature configuration. The I and Q signals are low-pass filtered and then digitized by a continuous time feedback architecture $\Sigma\Delta$ converter (ADC) allowing more than 80dB dynamic range. Once in the digital domain the signal is then decimated, down-converted again, decimated again, channel filtered and finally demodulated by the selected modem depending on modulation scheme: FSK modem or LoRa modem.

4.3.1 Intermediate Frequencies

The LLCC68 receiver mostly operates in low-IF configuration, expect for specific high-bandwidth settings.

Table 4-2: Intermediate Frequencies in FSK Mode

Setting Name	Bandwidth [kHz DSB]	Intermediate Frequency [kHz]
RX_BW_467	467.0	250
RX_BW_234	234.3	250
RX_BW_117	117.3	250
RX_BW_58	58.6	250
RX_BW_29	29.3	250
RX_BW_14	14.6	250
RX_BW_7	7.3	250
RX_BW_373	373.6	200
RX_BW_187	187.2	200
RX_BW_93	93.8	200
RX_BW_46	46.9	200
RX_BW_23	23.4	200
RX_BW_11	11.7	200
RX_BW_5	5.8	200
RX_BW_312	312.0	167
RX_BW_156	156.2	167
RX_BW_78	78.2	167
RX_BW_39	39.0	167
RX_BW_19	19.5	167
RX_BW_9	9.7	167
RX_BW_4	4.8	167

Table 4-3: Intermediate Frequencies in LoRa Mode

BW Setting	Bandwidth [kHz DSB]	Intermediate Frequency [kHz]
LoRa_BW_500	500	0
LoRa_BW_250	250	250
LoRa_BW_125	125	250

4.4 Transmitter

The transmit chain uses the modulated output from the modem bank which directly modulates the fractional-N PLL. An optional pre-filtering of the bit stream can be enabled to reduce the power in the adjacent channels, also dependent on the selected modulation type.

The default maximum RF output power of the transmitter is +22dBm. The RF output power is programmable with 32dB of dynamic range, in 1dB steps. The power amplifier ramping time is also programmable to meet regulatory requirements.

The power amplifier is supplied by the regulator VR_PA and the connection between VR_PA and RFO is done externally to the chip. VR_PA, supplied through VDD_IN, is taken directly from the battery and in this case maximum output power is limited by supply voltage at VDD_IN.

•

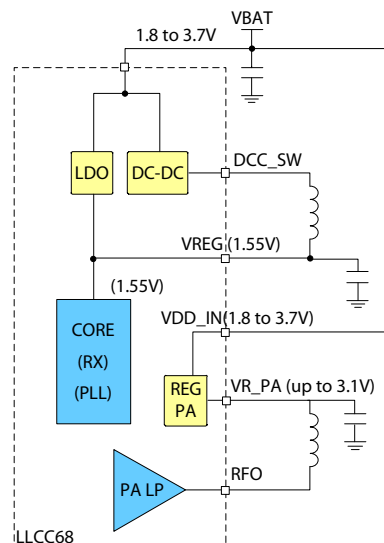


Figure 4-3: PA Supply Scheme in DC-DC Mode

4.4.1 Power Amplifier Specifics

Caution!

All figures below are indicative and typical, and are not a specification. These figures only highlight behavior of the PA over voltage and current.

Figures are given with DC-DC regulation enabled, which applies only to the circuit core.

The PA is optimized for maximum output power whilst maximizing the efficiency, which makes it mandatory to supply the power amplifier with fairly high voltages to maintain an high output power. To summarize:

- The current efficiency of the PA is optimal at the highest output power step
- Output power is limited by the voltage supplied to VBAT

This is illustrated in the following figure:

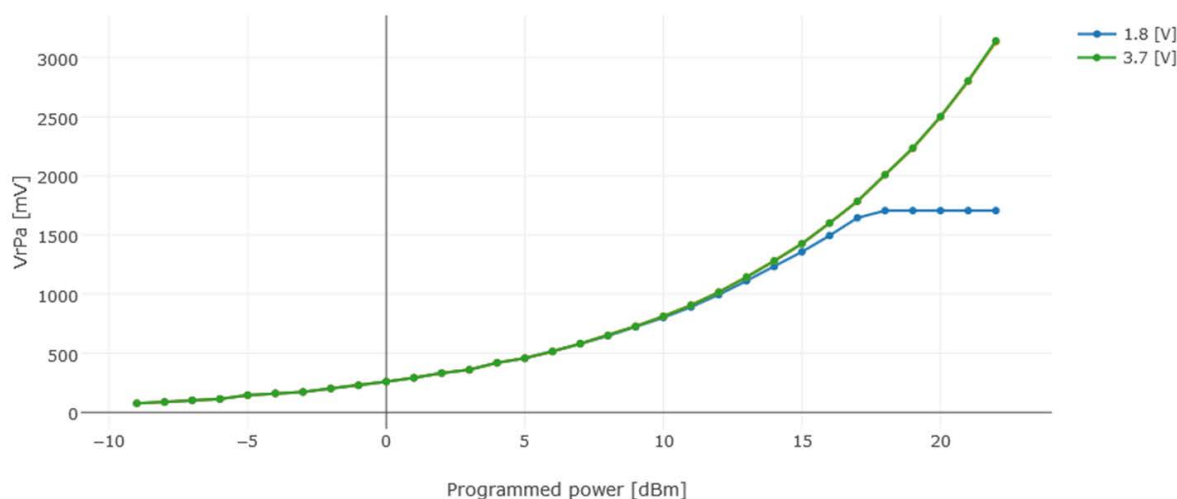


Figure 4-4: VR_PA versus Output Power on the LLCC68

The internal regulator for VR_PA has a little less than 200mV of drop-out, which means VBAT must be 200mV higher than the published VR_PA voltages in order to attain the corresponding output power. For example, for $P_{out} = +20\text{dBm}$, $VR_{PA} = 2.5\text{V}$ is required, which means that the LLCC68 is able to maintain $P_{out} = +20\text{dBm}$ on the $2.7\text{V} < V_{BAT} < 3.7\text{V}$ voltage range. Below 2.7V, the output power degrades as VBAT reduces.

As can be seen from the blue curve on [Figure 4-4: VR_PA versus Output Power on the LLCC68](#), the LLCC68 is capable of supplying almost 1.7V when $V_{BAT} = 1.8\text{V}$, which, in turn, makes the output power plateau at +17dBm for all power settings above +17dBm.

The following plot confirms the linearity of the output power, as long as the VBAT voltage is high enough to supply the required VR_PA voltage:

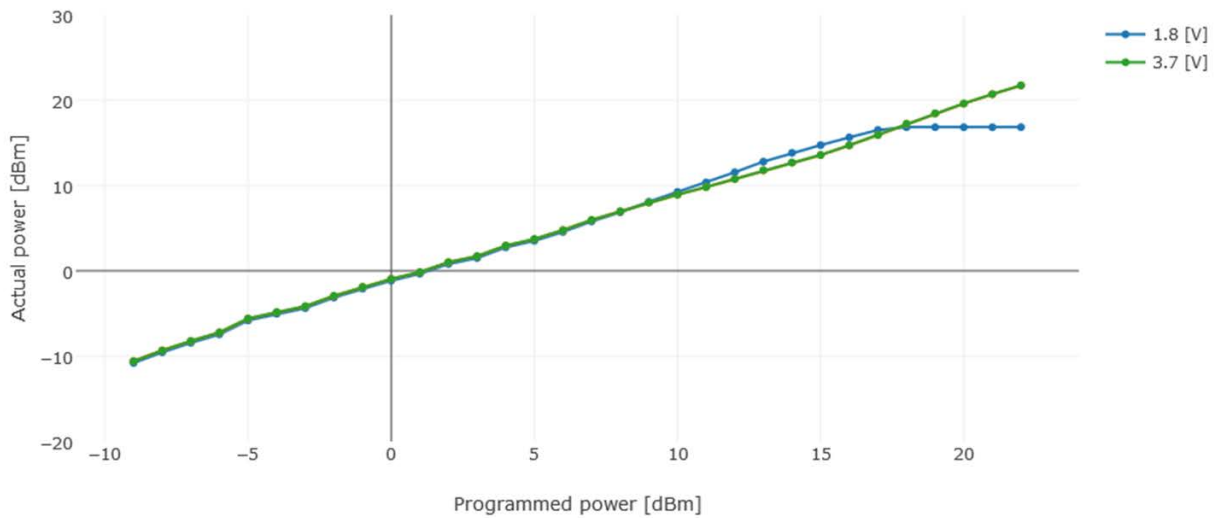


Figure 4-5: Power Linearity on the LLCC68

The power consumption evolves with the programmed output power, as follows (DC-DC regulation):

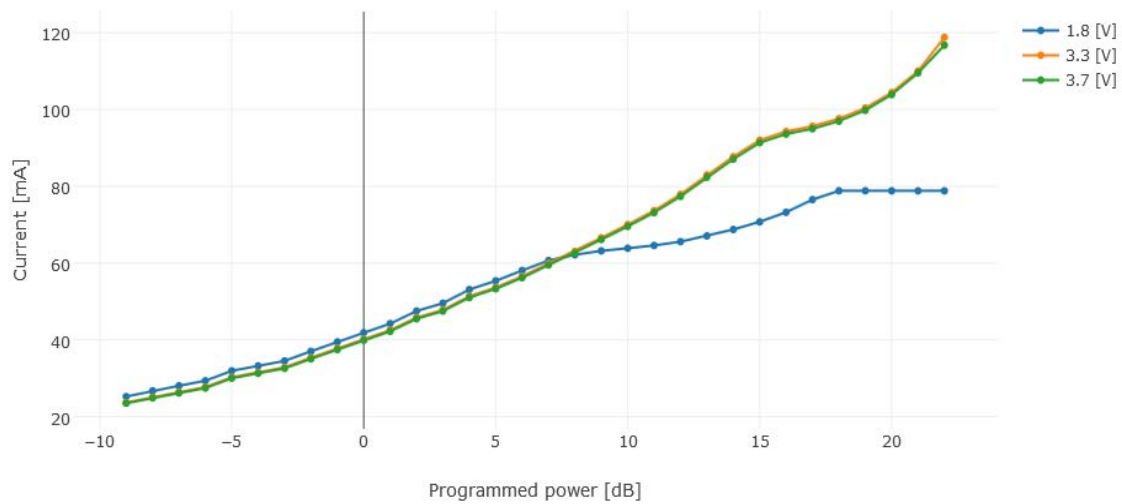


Figure 4-6: Current versus Programmed Output Power on the LLCC68

4.4.2 Power Amplifier Summary

The following table summarizes the power amplifier optimization keys between both PA supply modes:

Table 4-4: Power Amplifier Summary

PA Summary	Conditions	LLCC68
Max Power	with relevant matching and settings	+ 22dBm
IDDTX	at + 22dBm, indicative	118mA
	at + 14dBm, indicative	90/45 ¹ mA
Output Power vs. VBAT		Flat from 3.3V to 3.7V
		VBAT = 3.3V for +22dBm
		VBAT = 2.7V for +20dBm
		VBAT = 1.8V for +16dBm

1. See [Section 13.1.14.1 "PA Optimal Settings"](#) on page 76 .

5. Power Distribution

5.1 Selecting DC-DC Converter or LDO Regulation

Two forms of voltage regulation (DC-DC buck converter or linear LDO regulator) are available depending upon the design priorities of the application. The linear LDO regulator is always present in all modes but the transceiver uses DC-DC when selected. Alternatively a high efficiency DC to DC buck converter can be enabled in FS, Rx and Tx modes.

The DC-DC can be driven by two clock sources:

- in STDBY_XOSC: RC13M supplies the clock and the frequency is $RC13M / 4$ so the switching frequency of the DC-DC converter is 3.25MHz
- in FS, RX, TX: the PLL supplies the clock and the frequency is ~5MHz; every time command `SetRFFrequency(...)` is called the divider ratio is recalculated so that the switching frequency is as close as possible to the 5MHz target.

Unless specified, all specifications of the transceiver are given with the DC-DC regulator enabled. For applications where cost and size are constrained, LDO-only operation is possible which negates the need for the 47nH inductor before pin 1 and the 15μH inductor between pins 7 and 9, conferring the benefits of a reduced bill of materials and reduced board space. The following table illustrates the power regulation options for different modes and user settings.

Table 5-1: Regulation Type versus Circuit Mode

Circuit Mode	Sleep	STDBY_RC	STDBY_XOSC	FS	Rx	Tx
Regulator Type = 0	-	LDO	LDO	LDO	LDO	LDO
Regulator Type = 1	-	LDO	DC-DC + LDO	DC-DC + LDO	DC-DC + LDO	DC-DC + LDO

The user can specify the use of DC-DC by using the command `SetRegulatorMode(...)`. This operation must be carried out in STDBY_RC mode only.

When the DC-DC is enabled, the LDO remains ON and its target voltage is set 50mV below the DC-DC voltage to ensure voltage stability for high current peaks. If the DC-DC voltage drops to this level due to high current peak, the LDO covers for the current need at the expense of the energy consumption of the radio which is increased.

However, to avoid consuming too much energy, the user is free to configure the Over Current Protection (OCP) register manually. At Reset, the OCP is configured to limit the current at 140mA.

Table 5-2: OCP Configuration

Register Address	OCP default	Maximum Current
0x08E7	0x38	157.5mA

The OCP value is given on 6 bits, therefore the max OCP is $63 * 2.5mA = 157.5mA$. It is configurable in steps of 2.5mA and the default value is re-configured automatically each time the function `SetPaConfig(...)` is called. To adjust the OCP value, it is necessary to change the register as a second step after calling the function `SetPaConfig(...)`.

5.1.1 Option A: DC-DC Regulator

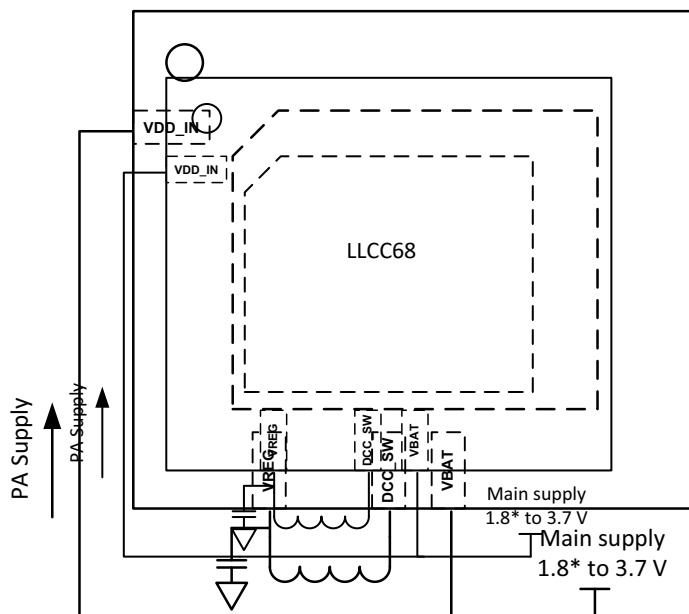
The DC-DC Regulator is used with about 90% of efficiency, for the chip core only. The PA regulator is supplied with VBAT. .

Advantage of this option:

The power consumption is reduced.

*VBAT=3.3V min. to reach +22dBm

Figure 5-1: LLCC68 Diagram with the DC-DC Regulator Power Option



5.1.2 Option B: LDO Regulator

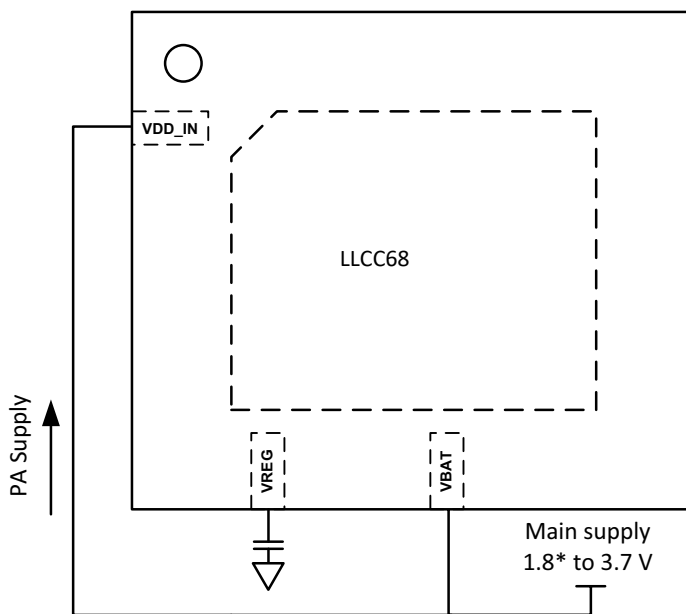
The LDO Regulator is used. Power consumption of the core is slightly higher than in Option A.

Advantage of this option:

The cost and space for the external 15μH inductor are spared.

*VBAT=3.3V min. to reach +22dBm

Figure 5-2: LLCC68 Diagram with the LDO Regulator Power Option



5.1.3 Consideration on the DC-DC Inductor Selection

The inductor is essential to ensure optimal performance of the DC-DC internal block. An incorrect inductor could cause unwanted effects ranging from ripple currents to early aging of the device, and a degradation of DC-DC regulator efficiency.

The preferred inductor is shielded, presenting a low internal series resistance and a resonance frequency much higher than the DC-DC switching frequency. When selecting the 15μH inductor, the user should therefore select a part with the following considerations:

- DCR (max) = 2ohms
- Idc (min) = 100mA
- Freq (min) = 20MHz

Table 5-3: Recommended 15μH Inductors

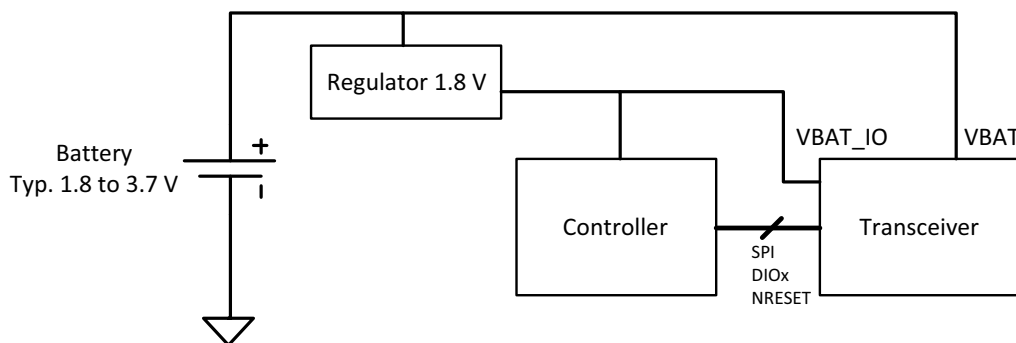
Reference	Manufacturer	Value (μH)	Idc max (mA)	Freq (MHz)	DCR (ohm)	Package (L x W x H in mm)
LPS3010-153	Coilcraft	15	370	43	0.95	2.95 x 2.95 x 0.9
MLZ2012N150L	TDK	15	90	40	0.47	2 x 1.25 x 1.25
MLZ2012M150W	TDK	15	120	40	0.95	2 x 1.25 x 1.25
VLS2010ET-150M	TDK	15	440	40	1.476	2 x 2 x 1
VLS2012ET-150M	TDK	15	440	40	1.062	2 x 2 x 1.2

5.2 Flexible DIO Supply

The transceiver has two power supply pins, one for the core of the transceiver called VBAT and one for the host controller interface (SPI, DIOs, BUSY) called VBAT_IO. Both power supplies can be connected together in the application. In case a low voltage micro-controller (typically with IO pads at 1.8V) is used to control the transceiver, the user can:

- Use VBAT at 3.3V for optimal RF performance
- Directly connect VBAT_IO to the same supply used for the micro-controller
- Connect the digital IOs directly to the micro-controller DIOs

At any time, VBAT_IO must be lower than or equal to VBAT



Requirement: $VBAT \geq VBAT_IO$

Figure 5-3: Separate DIO Supply

6. Modems

The LLCC68 contains different modems capable of handling LoRa and FSK modulations. LoRa and FSK are associated with their own frame and modem.

- LoRa modem ⇔ LoRa Frame
- FSK modem ⇔ FSK Frame

The user specifies the modem and frame type by using the command `SetPacketType(. .)`. This command specifies the frame used and consequently the modem implemented.

This function is the first one to be called before going to Rx or Tx and before defining frequency, modulation and packet parameters. The command `GetPacketType()` returns the current protocol of the radio.

The LLCC68 does not have a packet handler dedicated to LR-FHSS modulation. It provides the relevant modulation and frequency hopping capability.

GMSK is supported by the modems. and is treated as a special case of (G)FSK where $f_{dev}=br/4$.

6.1 LoRa® Modem

The LoRa modem uses spread spectrum modulation and forward error correction techniques to increase the range and robustness of radio communication links compared to traditional FSK based modulation.

An important facet of the LoRa modem is its increased immunity to interference. The LoRa modem is capable of co-channel GMSK rejection of up to 16dB. This immunity to interference permits the simple coexistence of LoRa modulated systems either in bands of heavy spectral usage or in hybrid communication networks that use LoRa to extend range when legacy modulation schemes fail.

6.1.1 Modulation Parameter

It is possible to optimize the LoRa modulation for a given application, access is given to the designer to four critical design parameters, each one permitting a trade-off between the link budget, immunity to interference, spectral occupancy and nominal data rate. These parameters are set using command `SetModulationParams(. . .)` which must be called after `SetPacketType(. . .)`:

- Modulation BandWidth (BW_L)
- Spreading Factor (SF)
- Coding Rate (CR)
- Low Data Rate Optimization (LDRO)

6.1.1.1 Spreading Factor

The spread spectrum LoRa modulation is performed by representing each bit of payload information by multiple chips of information. The rate at which the spread information is sent is referred to as the symbol rate (Rs). The ratio between the nominal symbol rate and chip rate is the spreading factor and it represents the number of symbols sent per bit of information.

SF5 and SF6 Considerations

Two spreading factors, SF5 and SF6, have been modified slightly in the LLCC68 and can now operate in both implicit and explicit mode. However, these new spreading factors are incompatible with previous generation devices, in particular SF6 on the LLCC68 is **not** backward compatible with SF6 on the SX1276. Furthermore, due to the higher symbol rate, the minimum recommended preamble length needed to ensure correct detection and demodulation from the receiver is increased compared to other Spreading Factors. For SF5 and SF6, you should use 12 symbols of preamble to have optimal performances over the dynamic range or the receiver.

Note: The spreading factor must be known in advance on both transmit and receive sides of the link as different spreading factors are orthogonal to each other. Note also the resulting Signal to Noise Ratio (SNR) required at the receiver input.

The capability to receive signals with negative SNR increases the sensitivity, link budget and range of the LoRa receiver.

Table 6-1: Range of Spreading Factors (SF)

Spreading Factor (SF) ¹	5	6	7	8	9	10	11
2 ^{SF} (Chips / Symbol)	32	64	128	256	512	1024	2048
Typical LoRa Demodulator SNR [dB]	-2.5	-5	-7.5	-10	-12.5	-15	-17.5

1. Not all SF are available for any bandwidth with the LLCC68

A higher spreading factor provides better receiver sensitivity at the expense of longer transmission times (time-on-air).

With a knowledge of the key parameters that can be selected by the user, the LoRa symbol rate is defined as:

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

where BW is the programmed bandwidth and SF is the spreading factor. The transmitted signal is a constant envelope signal. Equivalently, one chip is sent per second per Hz of bandwidth.

6.1.1.2 Bandwidth

An increase in signal bandwidth permits the use of a higher effective data rate, thus reducing transmission time at the expense of reduced sensitivity improvement.

The LoRa modem operates at a programmable bandwidth (BW_L) around a programmable central frequency f_{RF}

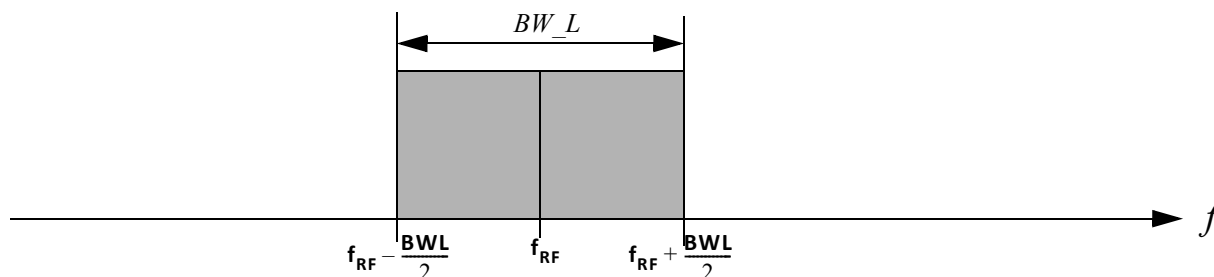


Figure 6-1: LoRa® Signal Bandwidth

An increase in LoRa signal bandwidth (BW_L) permits the use of a higher effective data rate, thus reducing transmission time at the expense of reduced sensitivity improvement. There are regulatory constraints in most countries on the permissible occupied bandwidth. The LoRa modem bandwidth always refers to the double side band (DSB). The range of LoRa signal bandwidths available is given in the table below:

Table 6-2: Signal Bandwidth Setting in LoRa® Mode

Signal Bandwidth	7	8	9
BW_L [kHz]	125	250 ¹	500 ¹

1. For RF frequencies below 400MHz, there is a scaling between the frequency and supported BW, some BW may not be available below 400MHz

For BW_L up to 250kHz, the receiver performs a double conversion. A first down conversion to low-IF is performed inside the RF chain, a second conversion to baseband is performed digitally inside the baseband modem. When the 500kHz bandwidth is used, a single down-conversion to zero-IF is performed in the RF part.

6.1.1.3 FEC Coding Rate

To further improve the robustness of the link the LoRa modem employs cyclic error coding to perform forward error detection and correction.

Forward Error Correction (FEC) is particularly efficient in improving the reliability of the link in the presence of interference. So that the coding rate and robustness to interference can be changed in response to channel conditions, the coding rate selected on the transmitter side is communicated to the receiver through the header (when present).

Table 6-3: Coding Rate Overhead

Coding Rate	Cyclic Coding Rate CR [in raw bits / total bits]	Overhead Ratio
1	4/5	1.25
2	4/6	1.5
3	4/7	1.75
4	4/8	2

A higher coding rate provides better noise immunity at the expense of longer transmission time. In normal conditions a factor of 4/5 provides the best trade-off; in the presence of strong interferers a higher coding rate may be used. Error correction code does not have to be known in advance by the receiver since it is encoded in the header part of the packet.

6.1.1.4 Low Data Rate Optimization

For low data rates (typically for high SF or low BW) and very long payloads which may last several seconds in the air, the low data rate optimization (LDRO) can be enabled. This reduces the number of bits per symbol to the given SF minus two (see [Section 6.1.4 "LoRa Time-on-Air" on page 37](#)) in order to allow the receiver to have a better tracking of the LoRa signal. Depending on the payload size, the low data rate optimization is usually recommended when a LoRa symbol time is equal or above 16.38ms.

6.1.2 LoRa Packet Engine

LoRa has its own packet engine that supports the LoRa PHY as described in the following section.

6.1.3 LoRa Frame

The LoRa modem employs two types of packet formats: explicit and implicit. The explicit packet includes a short header that contains information about the number of bytes, coding rate and whether a CRC is used in the packet. The packet format is shown in the following figure.

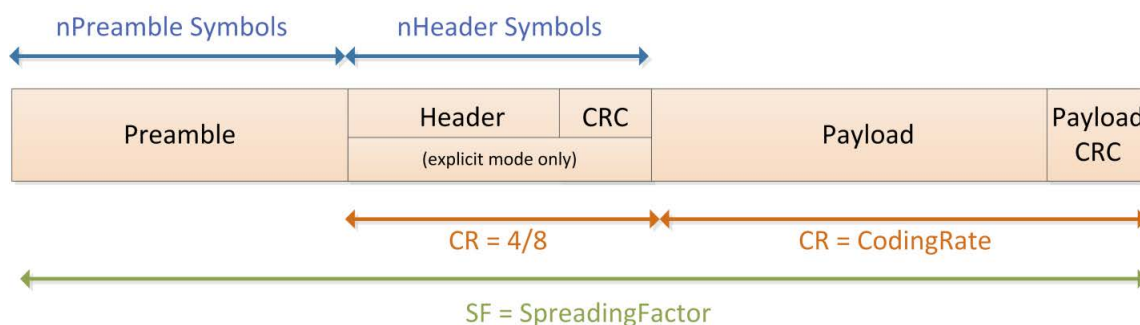


Figure 6-2: LoRa Packet Format

The LoRa packet starts with a preamble sequence which is used to synchronize the receiver with the incoming signal. By default the packet is configured with a 12-symbol long sequence. This is a programmable variable so the preamble length may be extended; for example, in

the interest of reducing the receiver duty cycle in receive intensive applications. The transmitted preamble length may vary from 10 to 65535 symbols, once the fixed overhead of the preamble data is considered. This permits the transmission of near arbitrarily long preamble sequences.

The receiver undertakes a preamble detection process that periodically restarts. For this reason the preamble length should be configured as identical to the transmitter preamble length. Where the preamble length is not known, or can vary, the maximum preamble length should be programmed on the receiver side.

The preamble is followed by a header which contains information about the following payload. The packet payload is a variable-length field that contains the actual data coded at the error rate either as specified in the header in explicit mode or as selected by the user in implicit mode. An optional CRC may be appended.

Depending upon the chosen mode of operation two types of header are available.

6.1.3.1 Explicit Header Mode

This is the default mode of operation. The header is transmitted with maximum error correction code (4/8). It also has its own CRC to allow the receiver to discard invalid headers. The header provides information on the payload, namely:

- The payload length in bytes
- The forward error correction coding rate
- The presence of an optional 16-bit CRC for the payload

6.1.3.2 Implicit Header Mode

In certain scenarios, where the payload, coding rate and CRC presence are fixed or known in advance, it may be advantageous to reduce transmission time by invoking implicit header mode. In this mode the header is removed from the packet. In this case the payload length, error coding rate and presence of the payload CRC must be manually configured identically on both sides of the radio link.

6.1.4 LoRa Time-on-Air

The Time-On-Air of the LoRa packet is shown in the SX126x drivers.

6.1.5 LoRa 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.

On the LLCC68, the channel activity detection mode is designed to detect the presence of a LoRa preamble or data symbols while the previous generations of products were only able to detect LoRa preamble symbols.

Once in CAD mode, the LLCC68 performs a scan of the band for a user-selectable duration (defined in number of symbols) and then returns with the Channel Activity Detected IRQ if LoRa symbols have been detected during the CAD.

The time taken for the channel activity detection is dependent upon the LoRa modulation settings used. For a given configuration (SF/BW) the typical CAD detection time can be selected to be either 1, 2, 4, 8 or 16 symbols. Once the duration of the selected number of symbols has passed, the radio remains for around half a symbol in Rx to post-process the measurement.

6.1.6 RSSI Calibration

The power seen by the LLCC68 analog front-end is affected by external components such as the matching network and RF switches. An incorrect RSSI results in a sensitivity degradation in (G)FSK mode and an incorrect gain selection in LoRa and GFSK mode. An incorrect gain can result in a missed detection (packet loss) or decreased resistance to interference.

The RSSI gain can be calibrated using the registers described in [Table 6-4](#). By default, the chip is calibrated for the 868-915MHz band on the LLCC68 EVK.

Table 6-4: Registers and Values

Name	Address	Bits	EVK 470-490MHz Band	EVK 868-915MHz Band
			Data	Data
AgcRssiMeasCalH	0x89c	4:0	0x01	0x01
AgcRssiMeasCalL	0x89d	7:0	0x27	0x53
AgcGainTune_1_2	0x8f5	7:4	0xD	0x0
AgcGainTune_1_2	0x8f5	3:0	0xE	0x0
AgcGainTune_3_4	0x8f6	7:4	0xE	0x0
AgcGainTune_3_4	0x8f6	3:0	0x2	0x0
AgcGainTune_5_6	0x8f7	7:4	0x3	0x0
AgcGainTune_5_6	0x8f7	3:0	0x2	0x0
AgcGainTune_7_8	0x8f8	7:4	0x4	0x0
AgcGainTune_7_8	0x8f8	3:0	0x4	0x0
AgcGainTune_9_10	0x8f9	7:4	0x3	0x0
AgcGainTune_9_10	0x8f9	3:0	0x3	0x0
AgcGainTune_11_12	0x8fa	7:4	0x3	0x0
AgcGainTune_11_12	0x8fa	3:0	0x4	0x0
AgcGainTune_13	0x8fb	3:0	0x4	0x0
AgcGforstPowthr	0x8b9	7:0	0x4	0xA
AgcSensiAdjust	0x8ac	7:2	0x22	0x25

Gain Tune and Gain Offset values can be determined using an RF generator with the LLCC68 hardware implementation as follows:

1. Setup the device in GFSK mode and set the desired RF frequency.
2. Setup the device in manual gain mode (disable the AGC (Auto Gain Control)).
3. Use SetRssiCalibration API to set all tunes and offsets to 0.
4. Loop through all the different gains, including LNA boost mode (high power mode):
 - a. Use the generator to emit a continuous tone at a specific output power gen_pwr (gain dependent), and account for all cable losses. A direct connection is recommended.
 - b. Read the instantaneous RSSI ($rssi_inst$).
 - c. Calculate the $rssi_error = rssi_inst - gen_pwr$.
 - d. Log the $rssi_error$ in a file.
5. Find the common offset between all errors, and use it as the new global offset.
6. Find the individual errors (relative to the new global offset) and use them as the tunes.
7. Write the RSSI Calibration register values, and go through the main loop again to check that the instantaneous RSSI is now calibrated.

6.2 FSK Modem

6.2.1 Modulation Parameter

The FSK modem is able to perform transmission and reception of 2-FSK modulated packets over a range of data rates from 0.6kbps to 500kbps. All parameters are set by using the command `SetModulationParams(...)`. This function should be called only after defining the protocol.

The bitrate setting is referenced to the crystal oscillator and provides a precise means of setting the bit rate (or equivalently chip) rate of the radio. In the command `SetModulationParams(...)`, the bitrate is expressed as 32 times the XTAL frequency divided the real bit rate used by the device. The generic formula is:

$$BR = \frac{F_{XOSC}}{BitRate} * 32$$

FSK modulation is performed inside the PLL bandwidth, by changing the fractional divider ratio in the feedback loop of the PLL. The high resolution of the sigma-delta modulator, allows for very narrow frequency deviation. The frequency deviation F_{dev} is one of the parameters of the function `SetModulationParams(...)` and is expressed as:

$$F_{dev} = \frac{F_{devHz}}{FreqStep}$$

where:

$$FreqStep = \frac{XtalFreq}{2^{25}}$$

Additionally, in transmission mode, several shaping filters can be applied to the signal. In reception mode, the user needs to select the best reception bandwidth depending on its conditions. To ensure correct demodulation, the following limit must be respected for the selection of the bandwidth:

$$(2 * F_{dev} + BR) < BW$$

The bandwidth is defined by parameter BW as described in the following table.

Table 6-5: Bandwidth Definition in FSK Packet Type (Sheet 1 of 2)

BW	Value	Bandwidth [kHz DS8]
BW4	0x1F	4.8
BW5	0x17	5.8
BW7	0x0F	7.3
BW9	0x1E	9.7
BW11	0x16	11.7
BW14	0x0E	14.6
BW19	0x1D	19.5
BW23	0x15	23.4
BW29	0x0D	29.3
BW39	0x1C	39.0

Table 6-5: Bandwidth Definition in FSK Packet Type (Sheet 2 of 2)

BW	Value	Bandwidth [kHz DSB]
BW46	0x14	46.9
BW58	0x0C	58.6
BW78	0x1B	78.2
BW93	0x13	93.8
BW117	0x0B	117.3
BW156	0x1A	156.2
BW187	0x12	187.2
BW234	0x0A	234.3
BW312	0x19	312.0
BW373	0x11	373.6
BW467	0x09	467.0
BW1000	0x08	1000.0 ¹

1. BW1000 requires additional settings, as follows:

- read register value at address 0x0805, and or the value with 0x7 before re-writing it at address 0x0805
- read register value at address 0x0806 and or the value with 0x32 before re-writing it at address 0x0806
- write value 0x01 at register address 0x0818
- write value 0xFA at register address 0x088F
- write value 0xAA at register address 0x0890
- write value 0xAC at register address 0x0891

The bandwidth must be chosen so that

$$\text{Bandwidth}[DSB] \geq BR + 2 * \text{frequency deviation} + \text{frequency error}$$

where the frequency error is two times the crystal frequency error used.

The LLCC68 offers several pulse shaping options defined by the parameter *PulseShape*. If other unspecified values are given as parameters, then no filtering is used.

6.2.2 FSK Packet Engine

The LLCC68 is designed for packet-based transmission. The packet controller block is responsible for assembly of the received data bit-stream into packets and their storage into the data buffer. It also performs the bit-stream decoding operations such as de-whitening & CRC-checks on the received bit-stream.

On the transmit side, the packet handler can construct a packet and send it bit by bit to the modulator for transmission. It can whiten the payload and append the CRC-checksum to the end of the packet. The packet controller only works in half-duplex mode i.e. either in transmit or receive at a time.

The packet controller is configured using the command `SetPacketParams(. . .)` as in [Section 13.4.6 "SetPacketParams" on page 87](#). This function can be called only after defining the protocol. The next chapters describe in detail the different frames available in the LLCC68.

6.2.2.1 Preamble Detection in Receiver Mode

The LLCC68 is able to gate the reception of a packet if an insufficient number of alternating preamble symbols (usually referred to 0x55 or 0xAA in hexadecimal form) has been detected. This can be selected by the user by using the parameter *PreambleDetectorLength* used in the command `SetPacketParams(. . .)`. The user can select a value ranging from “Preamble detector length off” - where the radio does not perform any gating and tries to lock directly on the following Sync Word - to “Preamble detector length 32 bits” where the radio is expecting to receive 32 bits of preamble before the following Sync Word. In this case, if the 32 bits of preamble are not detected, the radio either drops the reception in RxSingle mode, or restarts its tracking loop in RxContinuous mode.

To achieve best performance of the device, it is recommended to set *PreambleDetectorLength* to “Preamble detector length 8 bits” or “Preamble detector length 16 bits” depending of the complete size of preamble which is sent by the transmitter.

Note: In all cases, *PreambleDetectorLength* must be smaller than the size of the following Sync Word to achieve proper detection of the packets. If the preamble length is greater than the following Sync Word length (typically when no Sync Word is used) the user should fill some of the Sync Word bytes with 0x55.

6.2.3 FSK Packet Format

The FSK packet format provides a conventional packet format for application in proprietary NRZ coded, low energy communication links. The packet format has built in facilities for CRC checking of the payload, dynamic payload size and packet acknowledgement. Optionally whitening based upon pseudo random number generation can be enabled. Two principle packet formats are available in the FSK protocol: fixed length and variable length packets.

6.2.3.1 Fixed-Length Packet

If the packet length is fixed and known on both sides of the link then knowledge of the packet length does not need to be transmitted over the air. Instead the packet length can be written to the parameter *packetLength* which determines the packet length in bytes (0 to 255, but limited to 254 when the address filtering is activated, see Table 13-56).

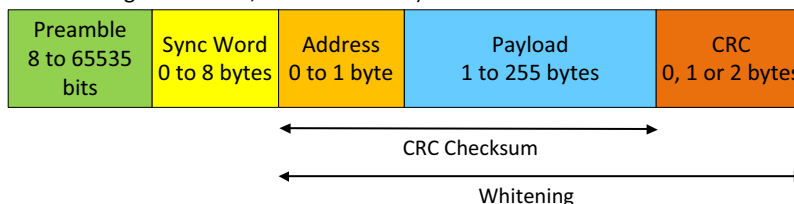


Figure 6-3: Fixed-Length Packet Format

The preamble length is set from 8 to 65535 bits using the parameter *PreambleLen*. **It is usually recommended to use a minimum of 16 bits for the preamble to guarantee a valid reception of the packet on the receiver side.** The CRC operation, packet length and preamble length are defined using the command `SetPacketParams(...)` as defined in Section 11. "List of Commands" on page 61.

6.2.3.2 Variable-Length Packet

Where the packet is of uncertain or variable size, then information about the packet length (0 to 255 bytes, but limited to 254 bytes when the address filtering is activated, see Table 13-56) must be transmitted within the packet. The format of the variable-length packet is shown below.

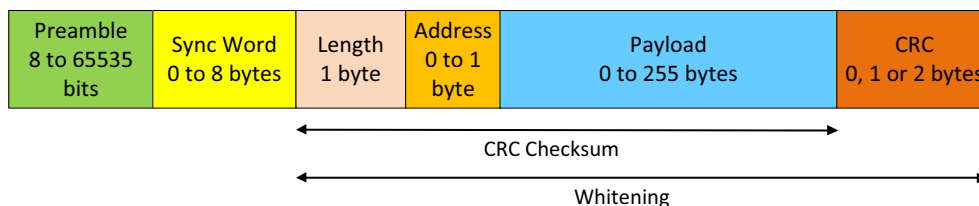


Figure 6-4: Variable-Length Packet Format

6.2.3.3 Setting the Packet Length or Node Address

The packet length and Node or Broadcast address are not considered part of the payload and they are added automatically in hardware.

The packet length is added automatically in the packet when the *packetType* field is set to variable size in the command `SetPacketParam(...)`.

The node or broadcast address can be enabled by using the *AddrComp* field in the command `SetPacketParam(...)`. This field allow the user to enable and select an additional packet filtering at the payload level.

6.2.3.4 Whitening

The whitening process is built around a 9-bit LFSR which is used to generate a random sequence and the payload (including the payload length, the Node or Broadcast address and CRC checksum when needed) is then XORed with this random sequence to generate the whitened payload. The data is de-whitened on the receiver side by XORing with the same random sequence. This setup limits the number of consecutive 1's or 0's to 9.

Note: Data whitening is only required when the user data has high correlation with long strings of 0's and 1's. If the data is already random then the whitening is not required. For example a random source generating the Transmit data, when whitened, could produce longer strings of 1's and 0's, thus it's not required to randomize an already random sequence.

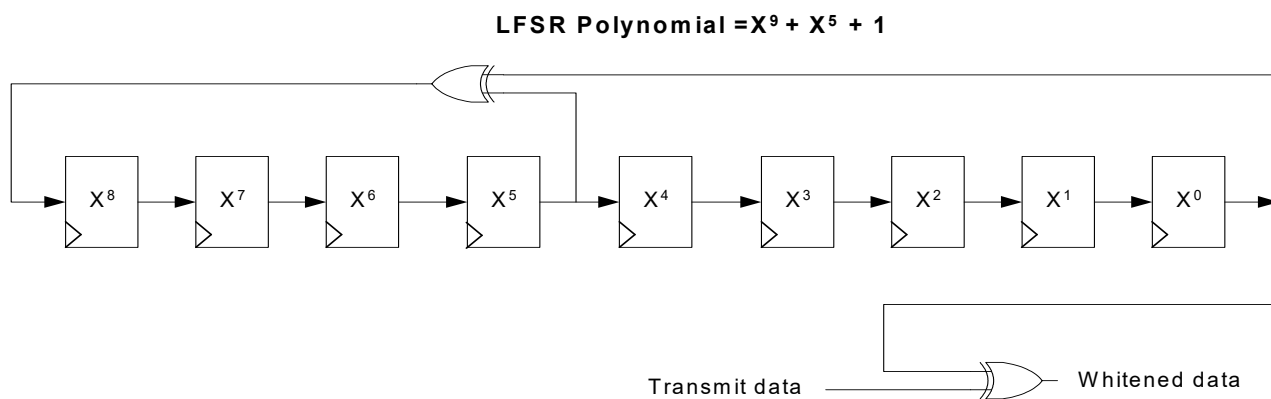


Figure 6-5: Data Whitening LFSR

The whitening is based around the 9-bit LFSR polynomial $x^9 + x^5 + 1$. With this structure, the LSB at the output of the LFSR is XORed with the MSB of the data.

At the initial stage, each flip-flop of the LFSR can be initialized through the registers at addresses 0x06B8 and 0x6B9.

Table 6-6: Whitening Initial Value

Whitening Initial Value	Register Address	Default Value
Whitening initial value MSB	0x06B8	0x01
Whitening initial value LSB	0x06B9	0x00

6.2.3.5 CRC

The LLCC68 offers full flexibility to select the polynomial and initial value of the selected polynomial. In additions, the user can also select a complete inversion of the computed CRC to comply with some international standards.

The CRC can be enabled and configured by using the *CRCType* field in the command `SetPacketParam(. . .)`. This field allows the user to enable and select the length and configuration of the CRC.

Table 6-7: CRC Type Configuration

CRCType	Description
0x01	CRC_OFF (No CRC)
0x00	CRC_1_BYTE (CRC computed on 1 byte)
0x02	CRC_2_BYTE (CRC computed on 2 bytes)
0x04	CRC_1_BYTE_INV (CRC computed on 1 byte and inverted)
0x06	CRC_2_BYTE_INV (CRC computed on 2 bytes and inverted)

The CRC selected must be modified together with the CRC initial value and CRC polynomial.

Table 6-8: CRC Initial Value

	Register Address	Default Value
CRC MSB Initial Value [15:8]	0x06BC	0x1D
CRC LSB Initial Value [7:0]	0x06BD	0x0F

Table 6-9: CRC Polynomial

	Register Address	Default Value
CRC MSB Polynomial Value [15:8]	0x06BE	0x10
CRC LSB Polynomial Value [7:0]	0x06BF	0x21

This flexibility permits the user to select any standard CRC or to use his own CRC allowing a specific detection of a given packet. Examples:

To use the IBM CRC configuration, the user must select:

- 0x8005 for the CRC polynomial
- 0xFFFF for the initial value
- CRC_2_BYTE for the field *CRCType* in the command `SetPacketParam(. . .)`.

For the CCIT CRC configuration the user must select:

- 0x1021 for the CRC polynomial
- 0x1D0F for the initial value
- CRC_2_BYTE_INV for the field *CRCType* in the command `SetPacketParam(. . .)`

6.2.4 Special Settings for BRF= 0.6kbps and 1.2kbps

Additional radio settings are necessary for bit rates 0.6kbps and 1.2kbps specifically:

- BRF= 1.2kbps: Write 0 inside bit 4 at address 0x08B8 (read-modify-write instruction).
- BRF= 0.6kbps: Write the following bits (read-modify-write):
 - ◆ bits 4:3 @ 0x06D1 to 0x03
 - ◆ bits 4:2 @ 0x089B to 1
 - ◆ bit 4 @ 0x08B8 to 0
 - ◆ bit 6 @ 0x06AC to 1
 - ◆ bit 5 @ 0x06AC to 0
 - ◆ bit 4 @ 0x06AC to 1

The user can read the default values of registers 0x6D1, 0x89B, 0x6AC, and 0x8B8 at any time, and use those values to restore these registers in case of need.

Note: These registers are not restored to their default value by commands `SetPacketParam()` or `SetModulationParams()`.

7. Data Buffer

The transceiver is equipped with a 256-byte RAM data buffer which is accessible in all modes except sleep mode. This RAM area is fully customizable by the user and allows access to either data for transmission or from the last packet reception.

7.1 Principle of Operation

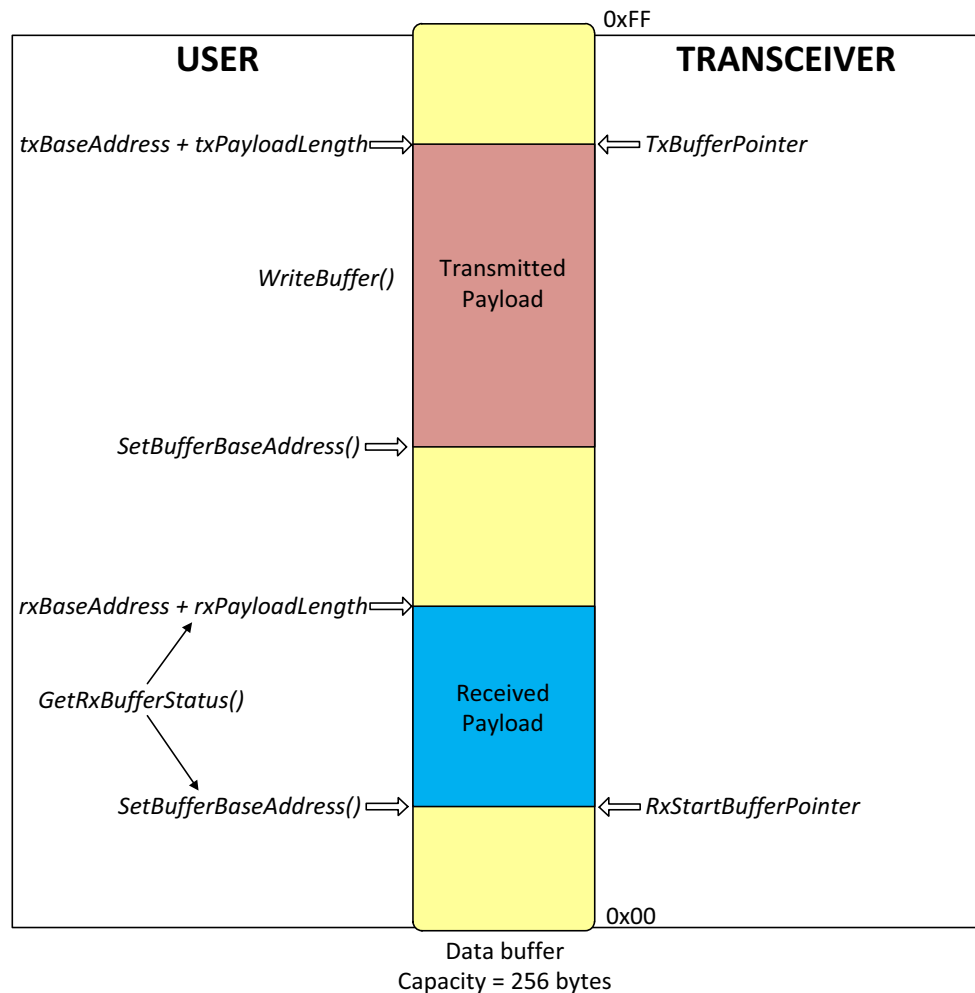


Figure 7-1: Data Buffer Diagram

The data buffer can be configured to store both transmit and receive payloads.

7.2 Data Buffer in Receive Mode

In receive mode *RxBaseAddr* specifies the buffer offset in memory at which the received packet payload data is written. The buffer offset of the last byte written in receive mode is then stored in *RxDataPointer* which is initialized to the value of *RxBaseAddr* at the beginning of the reception.

The pointer to the first byte of the last packet received and the packet length can be read with the command `GetRxbufferStatus()`.

In single mode, *RxDataPointer* is automatically initialized to *RxBaseAddr* each time the transceiver enters Rx mode. In continuous mode the pointer is incremented starting from the previous position.

7.3 Data Buffer in Transmit Mode

Upon each transition to transmit mode *TxDataPointer* is initialized to *TxBaseAddr* and is incremented each time a byte is sent over the air. This operation stops once the number of bytes sent equals the *payloadlength* parameter as defined in the function `SetPacketParams(...)`.

7.4 Using the Data Buffer

RxBaseAddr and *TxBaseAddr* are both set using the command `SetBufferBaseAddresses(...)`.

By default *RxBaseAddr* and *TxBaseAddr* are initialized at address 0x00.

Due to the contiguous nature of the data buffer, the base addresses for Tx and Rx are fully configurable across the 256-byte memory area. Each pointer can be set independently anywhere within the buffer. To exploit the maximum data buffer size in transmit or receive mode, the whole data buffer can be used in each mode by setting the base addresses *TxBaseAddr* and *RxBaseAddr* at the bottom of the memory (0x00).

The data buffer is cleared when the device is put into Sleep mode (implying no access). The data is retained in all other modes of operation.

The data buffer is accessed via the command `WriteBuffer(...)` and `ReadBuffer(...)`. In this function the parameter offset defines the address pointer of the first data to be written or read. Offset zero defines the first position of the data buffer.

Before any read or write operation it is hence necessary to initialize this offset to the corresponding beginning of the buffer. Upon reading or writing to the data buffer the address pointer then increments automatically.

Two possibilities exist to obtain the offset value:

- First is to use the *RxBaseAddr* value since the user defines it before receiving a payload.
- Second, offset can be initialized with the value of *RxStartBufferPointer* returned by `GetRxbufferStatus(...)` command.

Note: All the received data is written to the data buffer even if the CRC is invalid, permitting user-defined post processing of corrupted data. When receiving, if the packet size exceeds the buffer memory allocated for the Rx, it overwrites the transmit portion of the data buffer.

8. Digital Interface and Control

The LLCC68 is controlled via a serial **SPI** interface and a set of general purpose input/output (DIOs). At least one **DIO** must be used for **IRQ** and the **BUSY** line is mandatory to ensure the host controller is ready to accept the commands. The LLCC68 uses an internal controller (CPU) to handle communication and chip control (mode switching, **API** etc...). **BUSY** is used as a busy signal indicating that the chip is ready for new command only if this signal is low. When **BUSY** is high, the host controller must wait until it goes down again before sending another command. Through **SPI** the application sends commands to the internal chip or access directly the data memory space.

8.1 Reset

A complete “factory reset” of the chip can be issued on request by toggling pin 15 **NRESET** of the LLCC68. It is automatically followed by the standard calibration procedure and any previous context is lost. The pin should be held low for typically 100µs for the Reset to happen.

8.2 SPI Interface

The **SPI** interface gives access to the configuration register via a synchronous full-duplex protocol corresponding to **CPOL** = 0 and **CPHA** = 0 in Motorola™/Freescale™ nomenclature. Only the slave side is implemented.

An address byte followed by a data byte is sent for a write access whereas an address byte is sent and a read byte is received for the read access. The **NSS** pin goes low at the beginning of the frame and goes high after the data byte.

MOSI is generated by the master on the falling edge of **SCK** and is sampled by the slave (i.e. this **SPI** interface) on the rising edge of **SCK**. **MISO** is generated by the slave on the falling edge of **SCK**.

A transfer is always started by the **NSS** pin going low. **MISO** is high impedance when **NSS** is high.

The **SPI** runs on the external **SCK** clock to allow high speed up to 16MHz.

8.2.1 SPI Timing When the Transceiver is in Active Mode

In this mode the chip is able to handle **SPI** command in a standard way i.e. no extra delay needed at the first **SPI** transaction.

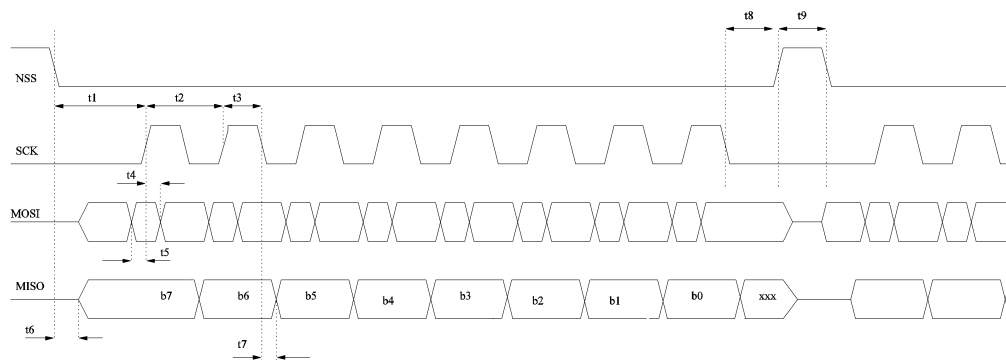


Figure 8-1: SPI Timing Diagram

All timings in the following table are given for a maximum load cap of 10pF.

Table 8-1: SPI Timing Requirements

Symbol	Description	Minimum	Typical	Maximum	Unit
t1	NSS falling edge to SCK setup time	32	-	-	ns
t2	SCK period	62.5	-	-	ns
t3	SCK high time	31.25	-	-	ns
t4	MOSI to SCK hold time	5	-	-	ns
t5	MOSI to SCK setup time	5	-	-	ns
t6	NSS falling to MISO delay	0	-	15	ns
t7	SCK falling to MISO delay,	0	-	15	ns
t8	SCK to NSS rising edge hold time	31.25	-	-	ns
t9	NSS high time	125	-	-	ns
t10	NSS falling edge to SCK setup time when switching from SLEEP to STDBY_RC mode	100	-	-	μs
t11	NSS falling to MISO delay when switching from SLEEP to STDBY_RC mode	0	-	150	μs

8.2.2 SPI Timing When the Transceiver Leaves Sleep Mode

One way for the chip to leave Sleep mode is to wait for a falling edge of NSS. At falling edge, all necessary internal regulators are switched On; the chip starts chip initialization before being able to accept first SPI command. This means that the delay between the falling edge of NSS and the first rising edge of SCK must take into account the wake-up sequence and the chip initialization. In Sleep mode and during the initialization phase, the busy signal mapped on BUSY pin, is set high indicating to the host that the chip is not able to accept a new command. Once the chip is in STDBY_RC mode, the busy signal goes low and the host can start sending a command. This is also true for startup at battery insertion or after a hard reset.

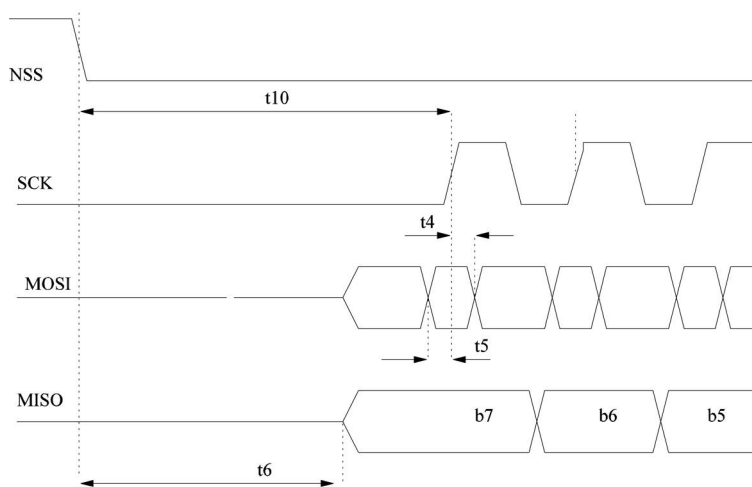


Figure 8-2: SPI Timing Transition

8.3 Multi-Purpose Digital Input/Output (DIO)

The chip is interfaced through the 4 control lines which are composed of the BUSY pin and 3 DIOs pins that can be configured as interrupt, debug or to control the radio immediate peripherals (TCXO or RF Switch).

8.3.1 BUSY Control Line

The BUSY control line is used to indicate the status of the internal state machine. When the BUSY line is held low, it indicates that the internal state machine is in idle mode and that the radio is ready to accept a command from the host controller.

The BUSY control line is set back to zero once the chip has reached a stable mode and it is ready for a new 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.

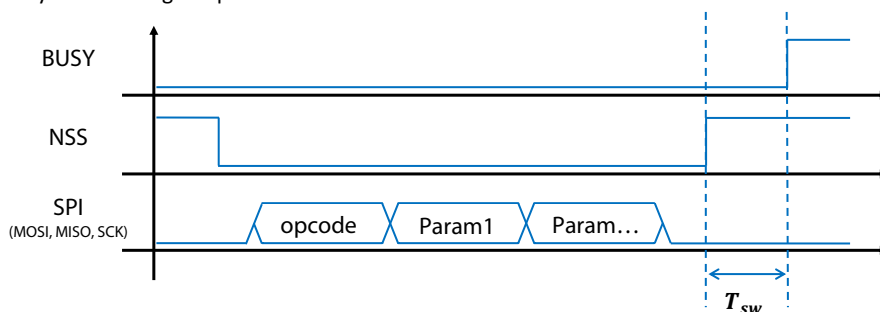


Figure 8-3: Switching Time Definition

For the internal state machine, all “write” commands cause the BUSY line to be asserted high after time T_{SW} , as per the graph above. T_{SW} represents the time required for the internal state machine to wake-up and start processing the command.

Conversely, the “read” command is handled directly without the help of the internal state machine and thus the BUSY line remains low after a “read” command.

The max value for T_{SW} from NSS rising edge to the BUSY rising edge is, in all cases, 600ns.

In Sleep mode, the BUSY pin is held high through a 20kΩ resistor. The BUSY line goes low as soon as the radio leaves Sleep mode.

In FS, BUSY goes low when the PLL is locked.

In RX, BUSY goes low as soon as the RX is up and ready to receive data.

In TX, BUSY goes low when the PA has ramped-up and transmission of preamble starts.

In addition to this, the BUSY also goes high to handle its internal IRQ. In this scenario, it is essential to wait for the BUSY line to go low before sending an SPI command (either a “read” or “write” command).

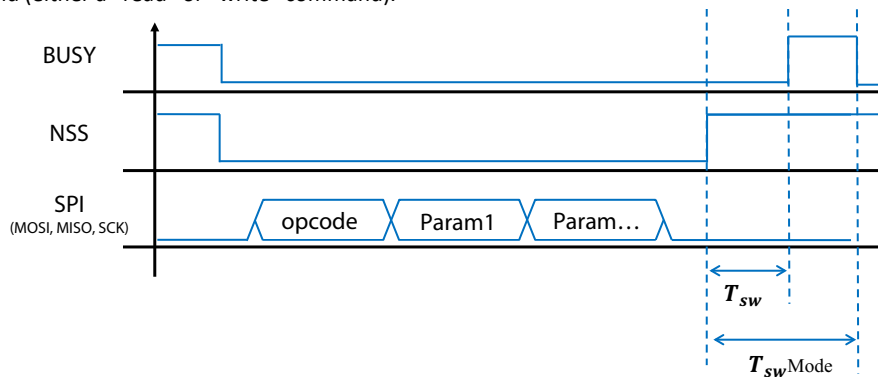


Figure 8-4: Switching Time Definition in Active Mode

The following table gives the value of $T_{SW}Mode$ for all possible transitions. The switching time is defined as the time between the rising edge of the NSS ending the SPI transaction and the falling edge of BUSY.

Table 8-2: Switching Time

Transition	$T_{SW}Mode$ Typical Value [μs]
SLEEP to STBY_RC cold start (no data retention)	3500
SLEEP to STBY_RC warm start (with data retention)	340
STBY_RC to STBY_XOSC	31
STBY_RC to FS	50
STBY_RC to RX	83
STBY_RC to TX	126
STBY_XOSC to FS	40
STBY_XOSC to TX	105
STBY_XOSC to RX	62
FS to RX	41
FS to TX	76
RX to FS	15
RX to TX	92

8.3.2 Digital Input/Output

Any of the 3 DIOs can be selected as an output interrupt source for the application. When the application receives an interrupt, it can determine the source by using the command `GetIrqStatus(...)`. The interrupt can then be cleared using the `ClearIrqStatus(...)` command. The Pin Description is as follows:

DIO1 is the generic IRQ line, any interrupt can be mapped to DIO1. The complete list of available IRQ can be found in [Section 8.4 "Digital Interface Status versus Chip modes" on page 52](#).

DIO2 has a double functionality. As DIO1, DIO2 can be used as a generic IRQ line and any IRQ can be routed through this pin. Also, DIO2 can be configured to drive an RF switch through the use of the command `SetDio2AsRfSwitchCtrl(...)`. In this mode, DIO2 is at a logical 1 during Tx and at a logical 0 in any other mode.

DIO3 also has a double functionality and as DIO1 or DIO2, it can be used as a generic IRQ line. Also, DIO3 can be used to automatically control a TCXO through the command `SetDio3AsTCXOctrl(...)`. In this case, the device automatically power cycles the TCXO when needed.

8.4 Digital Interface Status versus Chip modes

Table 8-3: Digital Pads Configuration for each Chip Mode

Mode	DIO3	DIO2	DIO1	BUSY	MISO	MOSI	SCK	NSS	NRESET
Reset	PD	PD	PD	PU	HIZ	HIZ	HIZ	IN	-
Start-up	PD	PD	PD	PU	HIZ	HIZ	HIZ	IN	IN PU
Sleep	PD	PD	PD	PU	HIZ	HIZ	HIZ	IN	IN PU
STBY_RC	OUT	OUT	OUT	OUT	OUT	IN	IN	IN	IN PU
STBY_XOSC	OUT	OUT	OUT	OUT	OUT	IN	IN	IN	IN PU
FS	OUT	OUT	OUT	OUT	OUT	IN	IN	IN	IN PU
RX	OUT	OUT	OUT	OUT	OUT	IN	IN	IN	IN PU
TX	OUT	OUT	OUT	OUT	OUT	IN	IN	IN	IN PU

Note:

- ◆ PU = pull up with 50kΩ at typical conditions
- ◆ PD = pull down with 50kΩ at typical conditions (the resistor value varies with the supply voltage)

8.5 IRQ Handling

In total there are 10 possible interrupt sources depending on the selected frame and chip mode. Each one can be enabled or masked. In addition, each one can be mapped to DIO1, DIO2 or DIO3.

Table 8-4: IRQ Status Registers

Bit	IRQ	Description	Modulation
0	TxDone	Packet transmission completed	All
1	RxDone	Packet received	All
2	PreambleDetected	Preamble detected	All
3	SyncWordValid	Valid Sync Word detected	FSK
4	HeaderValid	Valid LoRa Header received	LoRa
5	HeaderErr	LoRa Header CRC error	LoRa
6	CrcErr	Wrong CRC received	All
7	CadDone	Channel activity detection finished	LoRa
8	CadDetected	Channel activity detected	LoRa
9	Timeout	Rx or Tx Timeout	All

For more information on how to setup IRQ and DIOs, refer to the function `SetDioIrqParams()` in [Section 13.3.1 "SetDioIrqParams"](#) on [page 78](#).

8.6 DIO3 as a Manual Switch Control

DIO3 can be controlled via several register accesses. Simply perform the following steps to configure DIO3:

1. Set bit 3 of register@0x0580 to 1 (output enable on DIO3).
2. Set bit 3 of register@0x0583 to 0 (input disable on DIO3).
3. Set bit 3 of register@0x0584 to 0 (pull-up disable on DIO3).
4. Set bit 3 of register@0x0585 to 0 (pull-down disable on DIO3).
5. Set bits [0 to 2] of register@0x0920 to the output voltage you need on DIO3 (see Table 13-35, "txoVoltage Configuration Definition," on page 81).

To set the level of DIO3:

- Set bit 3 of register@0x0920 to 1 to have DIO3 high
- Set bit 3 of register@0x0920 to 0 to have DIO3 low

The configuration step is performed at PoR and each time the chip comes from SLEEP mode (because the state machine overwrites this custom configuration). For instance, if you want to configure DIO3 as a 3V output:

1. `Write(0x0580, Read(0x0580) | 0x08);`
2. `Write(0x0583, Read(0x0583) & ~0x08);`
3. `Write(0x0584, Read(0x0584) & ~0x08);`
4. `Write(0x0585, Read(0x0585) & ~0x08);`
5. `Write(0x0920, 0x06); // 0x06 for "DIO3 outputs 3.0 V"`

To set DIO3 high: `write(0x0920, read(0x0920) | 0x08);`

To set DIO3 low: `write(0x0920, read(0x0920) & ~0x08);`

9. Operational Modes

The LLCC68 features six operating modes. The analog front-end and digital blocks that are enabled in each operating mode are explained in the following table.

Table 9-1: LLCC68 Operating Modes

Mode	Enabled Blocks
SLEEP	Optional registers, backup regulator, RC64k oscillator, data RAM
STDBY_RC	Top regulator (LDO), RC13M oscillator
STDBY_XOSC	Top regulator (DC-DC or LDO), XOSC
FS	All of the above + Frequency synthesizer at Tx frequency
Tx	Frequency synthesizer and transmitter, Modem
Rx	Frequency synthesizer and receiver, Modem

9.1 Startup

At power-up or after a reset, the chip goes into STARTUP state, the control of the chip being done by the sleep state machine operating at the battery voltage. The BUSY pin is set to high indicating that the chip is busy and cannot accept a command. When the digital voltage and RC clock become available, the chip can boot up and the CPU takes control. At this stage the BUSY line goes down and the device is ready to accept commands.

9.2 Calibration

The calibration procedure is automatically called in case of POR or via the calibration command. Parameters can be added to the calibrate command to identify which section of calibration should be repeated. The following blocks can be calibrated:

- RC64k using the 32MHz crystal oscillator as reference
- RC13M using the 32MHz crystal oscillator as reference
- PLL to select the proper VCO frequency and division ratio for any RF frequency
- RX ADC
- Image (RX mode with defined tone)

Once the calibration is finished, the chip enters STDBY_RC mode.

9.2.1 Image Calibration for Specific Frequency Bands

Image calibration is done through the command `CalibrateImage(. . .)` for a given range of frequencies defined by the parameters *freq1* and *freq2*. Once performed, the calibration is valid for all frequencies between the two extremes used as parameters. Typically, the user can select the parameters *freq1* and *freq2* to cover any specific ISM band.

Table 9-2: Image Calibration Over the ISM Bands

Frequency Band [MHz]	Freq1	Freq2
430 - 440	0x6B	0x6F
470 - 510	0x75	0x81
779 - 787	0xC1	0xC5
863 - 870	0xD7	0xDB
902 - 928	0xE1 (default)	0xE9 (default)

In case of POR or when the device is recovering from Sleep mode in cold start mode, the image calibration is performed as part of the initial calibration process and for optimal image rejection in the band 902 - 928MHz. However at this stage the internal state machine has no information whether an XTAL or a TCXO is fitted. When the 32MHz clock is coming from a TCXO, the calibration fails and the user should request a complete calibration after calling the function `SetDI03AsTcxoCtrl(. . .)`.

By default, the image calibration is made in the band 902 - 928MHz. Nevertheless, it is possible to request the device to perform a new image calibration at other frequencies. For other frequency bands, a new function is available to setup the frequency using the parameters floor and ceil frequency in MHz. This is located in the SX126x drivers *sx126x.c* file:

```
sx126x_status_t sx126x_cal_img_in_mhz( const void* context, const uint16_t freq1_in_mhz, const uint16_t
freq2_in_mhz )
```

The pseudo-code is written hereafter:

```
#define SX126X_IMAGE_CALIBRATION_STEP_IN_MHZ 4
sx126x_status_t sx126x_cal_img_in_mhz( const void* context, const uint16_t freq1_in_mhz, const uint16_t
freq2_in_mhz )
{
    // Perform a floor() to get a value for freq1 corresponding to a frequency < or = freq1_in_mhz
    const uint8_t freq1 = freq1_in_mhz / SX126X_IMAGE_CALIBRATION_STEP_IN_MHZ;

    // Perform a ceil() to get a value for freq2 corresponding to a frequency > or = freq2_in_mhz
    const uint8_t freq2 = ( freq2_in_mhz + SX126X_IMAGE_CALIBRATION_STEP_IN_MHZ - 1 ) /
        SX126X_IMAGE_CALIBRATION_STEP_IN_MHZ;

    return sx126x_cal_img( context, freq1, freq2 );
}
```

9.3 Sleep Mode

In this mode, most of the radio internal blocks are powered down or in low power mode and optionally the **RC64k** clock and the timer are running. The chip may enter this mode from `STDBY_RC` and can leave the SLEEP mode if one of the following events occurs:

- NSS pin goes low in any case
- **RTC** timer generates an End-Of-Count (corresponding to Listen mode)

When the radio is in Sleep mode, the BUSY pin is held high.

9.4 Standby (STDBY) Mode

In standby mode the host should configure the chip before going to RX or TX modes. By default in this state, the system is clocked by the 13MHz RC oscillator to reduce power consumption (in all other modes except SLEEP the XTAL is turned ON). However if the application is time critical, the XOSC block can be turned or left ON.

XOSC or RC13M selection in standby mode is determined by mode parameter in the command `SetStandby(...)`.

The mode where only RC13M is used is called STDBY_RC and the one with XOSC ON is called STDBY_XOSC.

If DC-DC is to be used, the selection should be made while the circuit is in STDBY_RC mode by using the `SetRegulatorMode(...)` command, then the DC-DC automatically switches ON when entering STDBY_XOSC mode. The DC-DC is clocked by the RC13M. The LDO remains active with a target voltage 50mV lower than the DC-DC one.

9.5 Frequency Synthesis (FS) Mode

In FS mode, PLL and related regulators are switched ON. The BUSY goes low as soon as the PLL is locked or timed out.

For debugging purposes the chip may be requested to remain in this mode by using the `SetFs()` command.

Since the LLCC68 uses low IF architecture, the RX and TX frequencies are different. The RX frequency is equal to TX one minus the intermediate frequency (IF). In FS or TX modes, the RF frequency is directly programmed by the user.

9.6 Receive (RX) Mode

In RX mode, the RF front-end, RX ADC and the selected modem (LoRa or FSK) are turned ON.

In RX mode the circuit can operate in different sub-modes:

- **Continuous mode:** the device remains in RX mode and waits for incoming packet reception until the host requests a different mode
- **Single mode:** the device returns automatically to STDBY_RC mode after packet reception
- **Single mode with timeout:** the device returns automatically to STDBY_RC mode after packet reception or after the selected timeout
- **Listen mode:** the device alternate between Sleep and Rx mode until an IRQ is triggered

In RX mode, BUSY goes low as soon as the RX is enabled and ready to receive data.

The LLCC68 can operate with two levels of sensitivity:

- In Rx power saving gain, the radio consumes less power at a small cost in sensitivity
- In Rx Boosted gain, the radio consumes more power to improve the sensitivity

Table 9-3: Rx Gain Configuration

Rx Gain	Register Address	Value
Rx Gain	0x08AC	Rx power saving gain: 0x94 (default) Rx Boosted gain: 0x96

The *Rx Gain* register is not part of the retention memory when waking-up from a warm-start mode. To include this register in the retention memory, or to continue using Rx Boosted gain when using the `SetRxDutyCycle` method, the following steps are mandatory:

- Set register 0x029F to 0x01
- Set register 0x02A0 to 0x08
- Set register 0x02A1 to 0xAC

Note: The above register addresses are not detailed in Table 12-1: List of Registers.

When using Rx Boosted gain, the Rx gain configuration is not saved when entering sleep mode with retention activated. The Rx gain configuration, and up to 4 additional registers, can be saved when configuring the chip:

- Write 1 (8 bits) to address 0x29F
- Write 0x08AC (16-bit address of the Rx Gain register) to address 0x2A0-0x2A1 msb first

In LoRa mode, command `SetLoRaSymbNumTimeout(...)` can perform a quick and immediate assessment of the presence (or not) of LoRa preamble symbols. If the user defined parameter *SymbNum* is not 0, the modem waits for a total of *SymbNum* LoRa symbols to validate, or not, the correct detection of a LoRa packet. If the various states of the demodulator are not locked at this moment, the radio generates an *RxTimeout* IRQ. Otherwise, the radio stays in Rx for the full duration of the packet. For more information, see [Section 13.4.9 "SetLoRaSymbNumTimeout" on page 93](#).

9.7 Transmit (TX) Mode

In TX mode, after enabling and ramping-up the Power Amplifier (PA), the contents of the data buffer are transmitted. The circuit can operate in different sub-modes: single mode or single with timeout mode.

The timeout in Tx mode can be used as a security to ensure that if for any reason the Tx is aborted or does not succeed (i.e. the *TxDone* IRQ never is never triggered), the *TxTimeout* prevents the system from waiting for an undefined amount of time. Using the timeout while in Tx mode removes the need to use resources from the host MCU to perform the same task.

In TX mode, BUSY goes low as soon as the PA has ramped-up and transmission of preamble starts.

9.7.1 PA Ramping

The ramping of the PA can be selected while setting the output power by using the command `SetTxParams(...)`.

The PA ramp time can be selected to go from 10µs up to 3.4ms.

9.8 Active Mode Switching Time

For active mode switching time details, see [Section 8.3.1 "BUSY Control Line" on page 50](#).

9.9 Transceiver Circuit Modes Graphic

The device operating modes and the states through which each mode transitions are illustrated here:

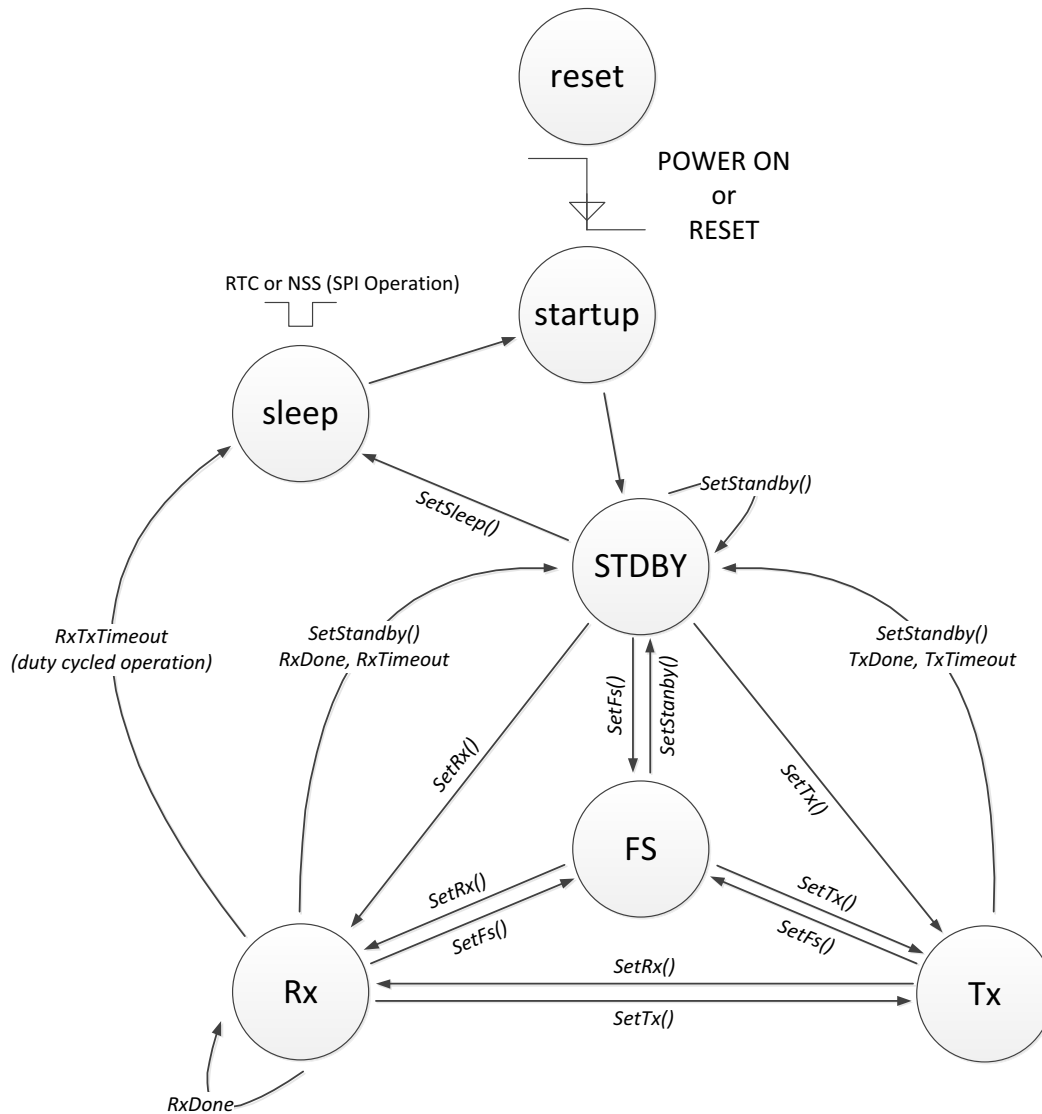


Figure 9-1: Transceiver Circuit Modes

10. Host Controller Interface

Through the SPI interface, the host can issue commands to the chip or access the data memory space to directly retrieve or write data. In normal operation, a reduced number of direct data write operations is required except when accessing the data buffer.

The user interacts with the circuit through an API (instruction set).

The LLCC68 uses the pin BUSY to indicate the status of the chip and its ability (or not) to receive another command while internal processing occurs. Prior to executing one of the generic functions, it is thus necessary to check the status of BUSY to make sure the chip is in a state where it can process another function.

10.1 Command Structure

In case of a command that does not require any parameter, the host sends only the opcode (1 byte).

In case of a command which requires one or several parameters, the opcode byte is followed immediately by parameter bytes with the NSS rising edge terminating the command.

Table 10-1: SPI Interface Command Sequence

Byte	0	[1:n]
Data from host	Opcode	Parameters
Data to host	RFU	Status

10.2 Transaction Termination

The host terminates an SPI transaction with the rising NSS signal; the host does not explicitly send the command length as a parameter. The host must not raise NSS within the bytes of a transaction.

If a transaction sends a command requiring parameters, all the parameters must be sent before rising NSS. If not the chip takes some unknown value for the missing parameters.

11. List of Commands

The following tables list all commands and their corresponding opcode. Unless specified, all parameters are 8-bit values.

11.1 Operational Modes Commands

These functions have a direct impact on the behaviour of the device. They control the internal state machine to transmit or receive packets, and all the modes in-between.

Table 11-1: Commands Selecting the Operating Modes of the Radio

Command	Opcode	Parameters	Description
SetSleep	0x84	sleepConfig	Set Chip in SLEEP mode
SetStandby	0x80	standbyConfig	Set Chip in STDBY_RC or STDBY_XOSC mode
SetFs	0xC1	-	Set Chip in Frequency Synthesis mode
SetTx	0x83	timeout[23:0]	Set Chip in Tx mode
SetRx	0x82	timeout[23:0]	Set Chip in Rx mode
StopTimerOnPreamble	0x9F	StopOnPreambleParam	Stop Rx timeout on Sync Word/Header or preamble detection
SetRxDutyCycle	0x94	rxPeriod[23:0], sleepPeriod[23:0]	Store values of RTC setup for listen mode and if period parameter is not 0, set chip into RX mode
SetCad	0xC5	-	Set chip in RX mode with passed CAD parameters
SetTxContinuousWave	0xD1	-	Set chip in TX mode with infinite carrier wave settings
SetTxInfinitePreamble	0xD2	-	Set chip into TX mode with infinite preamble settings
SetRegulatorMode	0x96	regModeParam	Select LDO or DC_DC+LDO for CFG_XOSC, FS, RX or TX mode
Calibrate	0x89	calibParam	Calibrate the RC13, RC64, ADC, PLL, Image according to parameter
CalibrateImage	0x98	freq1, freq2	Launches image calibration at the given frequencies
SetPaConfig	0x95	paDutyCycle, HpMax, deviceSel, paLUT	Configure Duty Cycle, Max output power, device for the PA
SetRxTxFallbackMode	0x93	fallbackMode	Defines into which mode the chip goes after a TX / RX done

11.2 Register and Buffer Access Commands

Table 11-2: Commands to Access the Radio Registers and FIFO Buffer

Command	Opcode	Parameters	Description
WriteRegister	0x0D	address[15:0], data[0:n]	Write into one or several registers
ReadRegister	0x1D	address[15:0]	Read one or several registers
WriteBuffer	0x0E	offset, data[0:n]	Write data into the FIFO
ReadBuffer	0x1E	offset	Read data from the FIFO

11.3 DIO and IRQ Control

Table 11-3: Commands Controlling the Radio IRQs and DIOs

Command	Opcode	Parameters	Description
SetDioIrqParams	0x08	IrqMask[15:0], Dio1Mask[15:0], Dio2Mask[15:0], Dio3Mask[15:0],	Configure the IRQ and the DIOs attached to each IRQ
GetIrqStatus	0x12	-	Get the values of the triggered IRQs
ClearIrqStatus	0x02	-	Clear one or several of the IRQs
SetDIO2AsRfSwitchCtrl	0x9D	enable	Configure radio to control an RF switch from DIO2
SetDIO3AsTcxoCtrl	0x97	tcxoVoltage, timeout[23:0]	Configure the radio to use a TCXO controlled by DIO3

11.4 RF, Modulation and Packet Commands

Table 11-4: Commands Controlling the RF and Packets Settings

Command	Opcode	Parameters	Description
SetRfFrequency	0x86	rfFreq[31:0]	Set the RF frequency of the radio
SetPacketType	0x8A	protocol	Select the packet type corresponding to the modem
GetPacketType	0x11	-	Get the current packet configuration for the device
SetTxParams	0x8E	power, rampTime	Set output power and ramp time for the PA

Table 11-4: Commands Controlling the RF and Packets Settings (Continued)

Command	Opcode	Parameters	Description
SetModulationParams	0x8B	modParam1, modParam2, modParam3	Compute and set values in selected protocol modem for given modulation parameters
SetPacketParams	0x8C	packetParam1, packetParam2, packetParam3, packetParam4, packetParam5, packetParam6, packetParam7, packetParam8, packetParam9	Set values on selected protocol modem for given packet parameters
SetCadParams	0x88	cadSymbolNum, cadDetPeak, cadDetMin, cadExitMode, cadTimeout	Set the parameters which are used for performing a CAD (LoRa only)
SetBufferBaseAddress	0x8F	TxbaseAddr, RxbaseAddr	Store TX and RX base address in register of selected protocol modem
SetLoRaSymbNumTimeout	0xA0	SymbNum	Set the number of symbol the modem has to wait to validate a lock

11.5 Status Commands

Table 11-5: Commands Returning the Radio Status

Command	Opcode	Parameters	Description
GetStatus	0xC0	-	Returns the current status of the device
GetRssiInst	0x15	-	Returns the instantaneous measured RSSI while in Rx mode
GetRxBufferStatus	0x13	-	Returns PayloadLengthRx(7:0), RxBufferPointer(7:0)
GetPacketStatus	0x14	-	Returns RssiAvg, RssiSync, PStatus2, PStatus3, PStatus4 in FSK protocol, returns RssiPkt, SnrPkt in LoRa protocol
GetDeviceErrors	0x17	-	Returns the error which has occurred in the device
ClearDeviceErrors	0x07	0x00	Clear all the error(s). The error(s) cannot be cleared independently
GetStats	0x10	-	Returns statistics on the last few received packets
ResetStats	0x00	-	Resets the value read by the command GetStats

12. Register Map

The register map is as shown below:

Table 12-1: List of Registers (Sheet 1 of 2)

Register Name	Address	Reset Value	Function
DIOx output enable	0x0580	0x00	Non-standard DIOx control ¹
DIOx input enable	0x0583	0x00	
DIOx pull-up control	0x0584	0x00	
DIOx pull-down control	0x0585	0x00	
Whitening initial value MSB	0x06B8	0xX1	Initial value used for the whitening LFSR in FSK mode. The user should not change the value of the 7 MSB of this register Note: X is here an undefined value
Whitening initial value LSB	0x06B9	0x00	
CRC MSB Initial Value [0]	0x06BC	0x1D	Initial value used for the polynomial used to compute the CRC in FSK mode
CRC LSB Initial Value [1]	0x06BD	0x0F	
CRC MSB polynomial Value [0]	0x06BE	0x10	Polynomial used to compute the CRC in FSK mode
CRC LSB polynomial Value [1]	0x06BF	0x21	
SyncWord[0]	0x06C0	-	1st byte of the Sync Word in FSK mode
SyncWord[1]	0x06C1	-	2nd byte of the Sync Word in FSK mode
SyncWord[2]	0x06C2	-	3rd byte of the Sync Word in FSK mode
SyncWord[3]	0x06C3	-	4th byte of the Sync Word in FSK mode
SyncWord[4]	0x06C4	-	5th byte of the Sync Word in FSK mode
SyncWord[5]	0x06C5	-	6th byte of the Sync Word in FSK mode
SyncWord[6]	0x06C6	-	7th byte of the Sync Word in FSK mode
SyncWord[7]	0x06C7	-	8th byte of the Sync Word in FSK mode
Node Address	0x06CD	0x00	Node Address used in FSK mode
Broadcast Address	0x06CE	0x00	Broadcast Address used in FSK mode
IQ Polarity Setup	0x0736	0x0D	Optimize the inverted IQ operation
LoRa Sync Word MSB	0x0740	0x14	Differentiate the LoRa signal for Public or Private Network Set to 0x3444 for Public Network Set to 0x1424 for Private Network
LoRa Sync Word LSB	0x0741	0x24	
LoRa Coding Rate Rx	0x0749	0x00	Bits 6 to 4 (included): LoRa coding rate extracted from the header - valid only in explicit header mode

Table 12-1: List of Registers (Sheet 2 of 2)

Register Name	Address	Reset Value	Function
LoRa CRC configuration Rx	0x076B	0x00	Bit 4: LoRa CRC configuration extracted from the header - valid only in explicit header mode
DccCtrl	0x0805	0x00	For BW1000 usage ²
MixCtrl	0x0806	0x20	For BW1000 usage ²
MixMode	0x0818	0x00	For BW1000 usage ²
IF Freq[0]	0x088F	0x0C	For BW1000 usage ²
IF Freq[1]	0x0890	0x00	For BW1000 usage ²
IF Freq[2]	0x0891	0x00	For BW1000 usage ²
RandomNumberGen[0]	0x0819	-	Can be used to get a 32-bit random number
RandomNumberGen[1]	0x081A	-	
RandomNumberGen[2]	0x081B	-	
RandomNumberGen[3]	0x081C	-	
TxModulation	0x0889	0x01	Refer to Section 15 .
Rx Gain	0x08AC	0x94	Set the gain used in Rx mode: Rx Power Saving gain: 0x94 Rx Boosted gain: 0x96
TxClampConfig	0x08D8	0xC8	Refer to Section 15 .
OCP Configuration	0x08E7	0x18	Set the Over Current Protection level The value is changed internally depending on the device selected. Default value: 0x38 (140mA)
RTC Control ³	0x0902	0x00	Enable or disable RTC Timer
XTA trim	0x0911	0x05	Value of the trimming cap on XTA pin This register should only be changed while the radio is in STDBY_XOSC mode
XTB trim	0x0912	0x05	Value of the trimming cap on XTB pin This register should only be changed while the radio is in STDBY_XOSC mode
DIO3 output voltage control	0x920	0x01	Non-standard DIO3 control ¹
Event Mask	0x944	0x00	Used to clear events ¹

1. Use only with Semtech-provided code samples

2. Use only with Semtech-provided configuration of [Section 6.2.1](#)

3. Use only with Semtech-provided workaround of [Section 15.3](#)

13. Commands Interface

13.1 Operational Modes Functions

13.1.1 SetSleep

The command `SetSleep(. . .)` sets the device in SLEEP mode with the lowest current consumption possible. This command can be sent only while in STDBY mode (STDBY_RC or STDBY_XOSC). After the rising edge of NSS, all blocks are switched OFF except the backup regulator if needed and the blocks specified in the parameter *sleepConfig*.

Table 13-1: SetSleep SPI Transaction

Byte	0	1
Data from host	Opcode = 0x84	sleepConfig

The *sleepConfig* argument is defined in [Table 13-2](#).

Table 13-2: Sleep Mode Definition

SleepConfig[7:3]	SleepConfig [2]	SleepConfig [1]	SleepConfig [0]
RESERVED	0: cold start	0: RFU	0: RTC timeout disable
RESERVED	1: warm start (device configuration in retention) ¹	0: RFU	RFU

1. Only configuration for the activated modem before going to sleep is retained. Configuration of other modems is lost and must be re-configured.

When entering SLEEP mode, the BUSY line is asserted high and stays high for the complete duration of the SLEEP period.

When entering SLEEP mode, by default the chip configuration is lost. However, if the `SetSleep(. . .)` command is sent with *sleepConfig[2]* set to 1, the chip configuration can be stored, to lower host interaction or during RxDutyCycle mode, using the register in retention mode during SLEEP state. Once the chip leaves SLEEP mode (by NSS or RTC event), the chip first restores the registers with the values stored in the retention register.

Caution: Once the command `SetSleep(. . .)` has been sent, the device becomes unresponsive for around 500μs, time needed for the configuration saving process and proper switch off of the various blocks. The user must thus make sure the device does not receive any SPI command during these 500μs to ensure proper operations of the device.

Once in SLEEP mode, it is possible to wake the device up from the host processor with a falling edge on the NSS line.

13.1.2 SetStandby

The command `SetStandby(...)` sets the device in a configuration mode which is at an intermediate level of consumption. In this mode, the chip is placed in halt mode waiting for instructions via SPI. This mode is dedicated to chip configuration using high level commands such as `SetPacketType(...)`.

By default, after battery insertion or reset operation (pin NRESET goes low), the chip enters in STDBY_RC mode running with a 13MHz RC clock.

Table 13-3: SetConfig SPI Transaction

Byte	0	1
Data from host	Opcode = 0x80	StdbyConfig

The *StdbyConfig* byte definition is as follows:

Table 13-4: STDBY Mode Configuration

StdbyConfig	Value	Description
STDBY_RC	0	Device running on RC13M , set STDBY_RC mode
STDBY_XOSC	1	Device running on XTAL 32MHz, set STDBY_XOSC mode

13.1.3 SetFs

The command `SetFs()` sets the device in the frequency synthesis mode where the PLL is locked to the carrier frequency. This mode is used for test purposes of the PLL and can be considered as an intermediate mode. It is automatically reached when going from STDBY_RC mode to TX mode or RX mode.

Table 13-5: SetFs SPI Transaction

Byte	0
Data from host	Opcode = 0xC1

In FS mode, the PLL is set to the frequency programmed by the function `SetRfFrequency(...)` which is the same used for TX or RX operations.

13.1.4 SetTx

The command SetTx() sets the device in transmit mode.

Table 13-6: SetTx SPI Transaction

Byte	0	1-3
Data from host	Opcode = 0x83	timeout(23:0)

- Starting from STDBY_RC mode, the oscillator is switched ON followed by the PLL, then the PA is switched ON and the PA regulator starts ramping according to the ramping time defined by the command SetTxParams(. . .)
- When the ramping is completed the packet handler starts the packet transmission
- When the last bit of the packet has been sent, an IRQ TX_DONE is generated, the PA regulator is ramped down, the PA is switched OFF and the chip goes back to STDBY_RC mode
- A TIMEOUT IRQ is triggered if the TX_DONE IRQ is not generated within the given timeout period
- The chip goes back to STBY_RC mode after a TIMEOUT IRQ or a TX_DONE IRQ.

The timeout duration can be computed with the formula:

$$\text{Timeout duration} = \text{Timeout} * 15.625\mu\text{s}$$

Timeout is a 23-bit parameter defining the number of step used during timeout as defined in the following table.

Table 13-7: SetTx Timeout Duration

Timeout(23:0)	Timeout Duration
0x000000	Timeout disable, Tx Single mode, the device stays in TX Mode until the packet is transmitted and returns in STBY_RC mode upon completion.
Others	Timeout active, the device remains in TX mode, it returns automatically to STBY_RC mode on timer end-of-count or when a packet has been transmitted. The maximum timeout is then 262s.

The value given for the timeout should be calculated for a given packet size, given modulation and packet parameters. The timeout behaves as a security in case of conflicting commands from the host controller.

The timeout in Tx mode can be used as a security to ensure that if for any reason the Tx is aborted or does not succeed (i.e. the TxDone IRQ never is never triggered), the TxTimeout prevents the system from waiting for an undefined amount of time. Using the timeout while in Tx mode removes the need to use resources from the host MCU to perform the same task.

13.1.5 SetRx

The command `SetRx()` sets the device in receiver mode.

Table 13-8: SetRx SPI Transaction

Byte	0	1-3
Data from host	Opcode = 0x82	timeout(23:0)

This command sets the chip in RX mode, waiting for the reception of one or several packets. The receiver mode operates with a timeout to provide maximum flexibility to end users.

Table 13-9: SetRx Timeout Duration

Timeout(23:0)	Timeout Duration
0x000000	No timeout. Rx Single mode. The device stays in RX Mode until a reception occurs and the devices return in STBY_RC mode upon completion
0xFFFFFF	Rx Continuous mode. The device remains in RX mode until the host sends a command to change the operation mode. The device can receive several packets. Each time a packet is received, a packet done indication is given to the host and the device automatically searches for a new packet.
Others	Timeout active. The device remains in RX mode, it returns automatically to STBY_RC mode on timer end-of-count or when a packet has been received. As soon as a packet is detected, the timer is automatically disabled to allow complete reception of the packet. The maximum timeout is then 262s.

When the timeout is active ($0x000000 < \text{timeout} < 0xFFFFFF$), the radio stops the reception at the end of the timeout period unless a preamble and Sync Word (in GFSK) or Header (in LoRa) has been detected. This is to ensure that a valid packet is not dropped in the middle of the reception due to the pre-defined timeout. By default, the timer is stopped only if the Sync Word or header has been detected. However, it is also possible to stop the timer upon preamble detection by using the command `StopTimerOnPreamble(...)`.

13.1.6 StopTimerOnPreamble

The command `StopTimerOnPreamble(...)` allows the user to select if the timer is stopped upon preamble detection of Sync Word / header detection.

Table 13-10: StopTimerOnPreamble SPI Transaction

Byte	0	1
Data from host	Opcode = 0x9F	StopOnPreambleParam

The enable byte definition is given in the following table.

Table 13-11: StopOnPreambleParam Definition

StopOnPreambleParam	Value	Description
disable	0x00	Timer is stopped upon Sync Word or Header detection
enable	0x01	Timer is stopped upon preamble detection

By default, the timer is stopped only when the Sync Word (in GFSK) or Header (in LoRa) has been detected. When the function `StopTimerOnPreamble(...)` is used with the value *enable* at 0x01, then the timer stops upon preamble detection and the device stays in RX mode until a packet is received. It is important to notice that stopping the timer upon preamble may cause the device to stay in Rx for an unexpected long period of time in case of false detection.

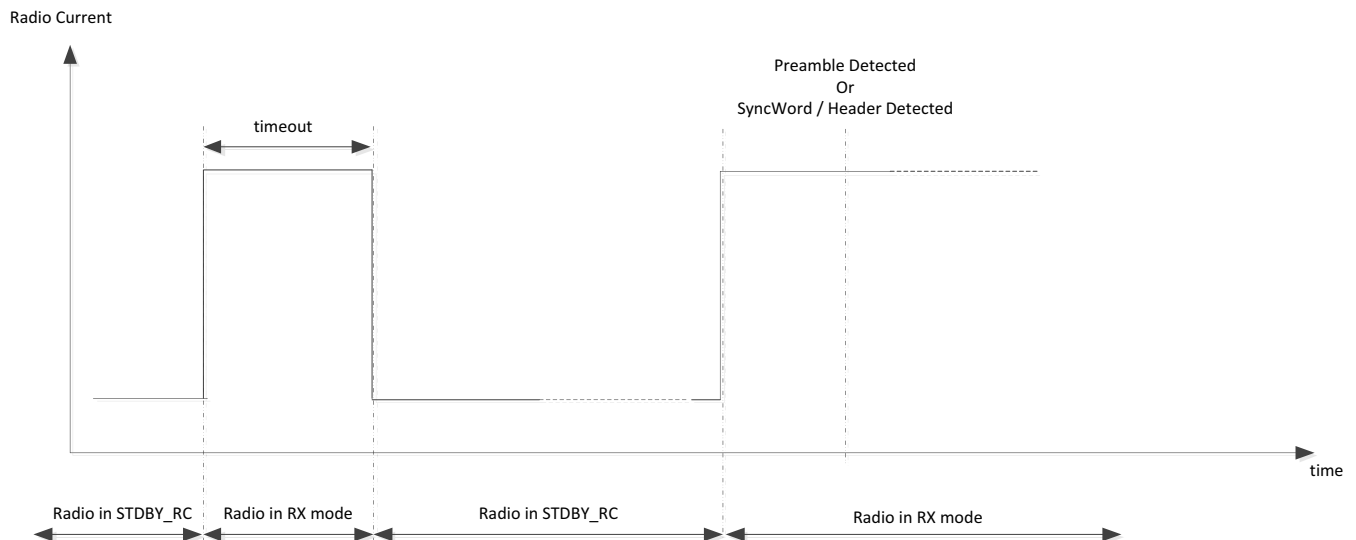


Figure 13-1: Stopping Timer on Preamble or Header Detection

13.1.7 SetRxDutyCycle

This command sets the chip in sniff mode so that it regularly looks for new packets. This is the listen mode.

Table 13-12: SetRxDutyCycle SPI Transaction

Byte	0	1-3	4-6
Data from host	Opcode = 0x94	rxPeriod(23:0)	sleepPeriod(23:0)

When this command is sent in STDBY_RC mode, the context (device configuration) is saved and the chip enters in a loop defined by the following steps:

- The chip enters RX and listens for a packet for a period of time defined by *rxPeriod*
- The chip looks for a preamble in either LoRa or FSK
- Upon preamble detection, the timeout is stopped and restarted with the value **2 * rxPeriod + sleepPeriod**
- If no packet is received during the RX window (defined by rxPeriod), the chip goes into SLEEP mode with context saved for a period of time defined by *sleepPeriod*
- At the end of the SLEEP window, the chip automatically restarts the process of restoring context and enters the RX mode, and so on. At any time, the host can stop the procedure.

The loop is terminated if either:

- A packet is detected during the RX window, at which moment the chip interrupts the host via the RX_DONE flag and returns to STBY_RC mode
- The host issues a SetStandby(. . .) command during the RX window (during SLEEP mode, the device is unable to receive commands straight away and must first be waken up by a falling edge of NSS).

The SLEEP mode duration is defined by:

$$\text{Sleep Duration} = \text{sleepPeriod} * 15.625\mu\text{s}$$

The RX mode duration is defined by

$$\text{Rx Duration} = \text{rxPeriod} * 15.625\mu\text{s}$$

The following figure highlights operations being performed while in RxDutyCycle mode. It can be observed that the radio takes around 1ms to save the context, go into SLEEP mode, re-initialize the radio, lock the PLL and go into RX. The delay is not accurate and may vary depending on the time needed for the XTAL to start, the PLL to lock, etc.

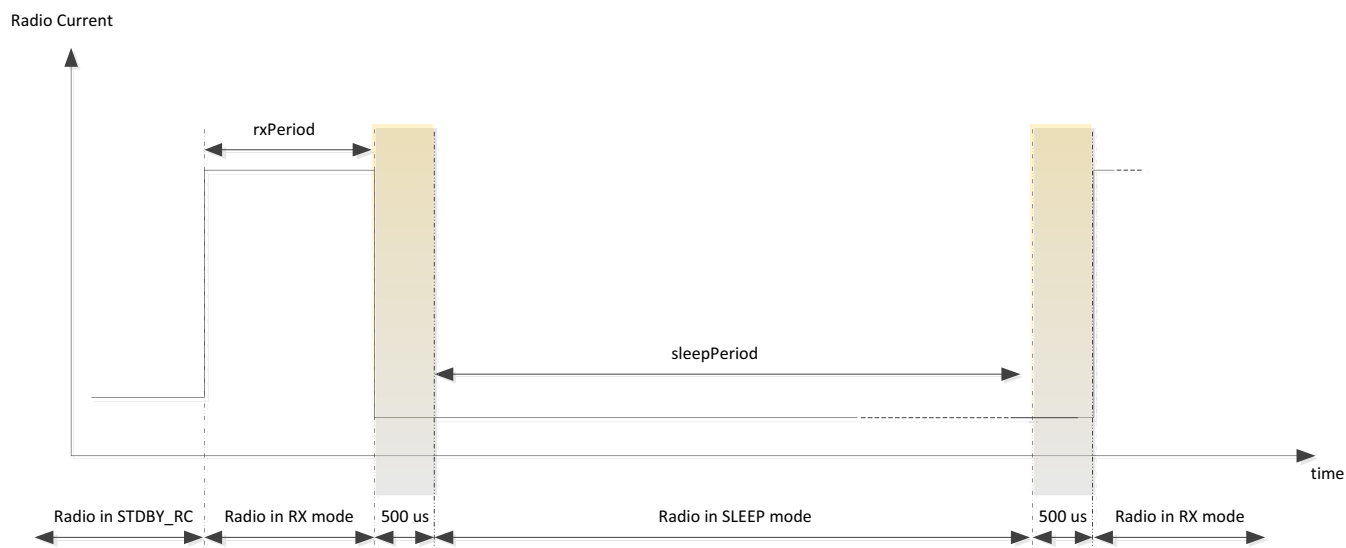


Figure 13-2: RX Duty Cycle Energy Profile

Upon preamble detection, the radio is set to look for a Sync Word (in GFSK) or a header (in LoRa) and the timer is restarted with a new value which is computed as $2 * rxPeriod + sleepPeriod$. This is to ensure that the radio does not spend an indefinite amount of time waiting in Rx for a packet which may never arrive (false preamble detection).

This implies a strong relationship between the time-on-air of the packet to be received, and the amount of time the radio spends in RX and in SLEEP mode. If a long preamble is used on the TX side, care must be taken that the formula below is respected:

$$T_{preamble} + T_{header} \leq 2 * rxPeriod + sleepPeriod$$

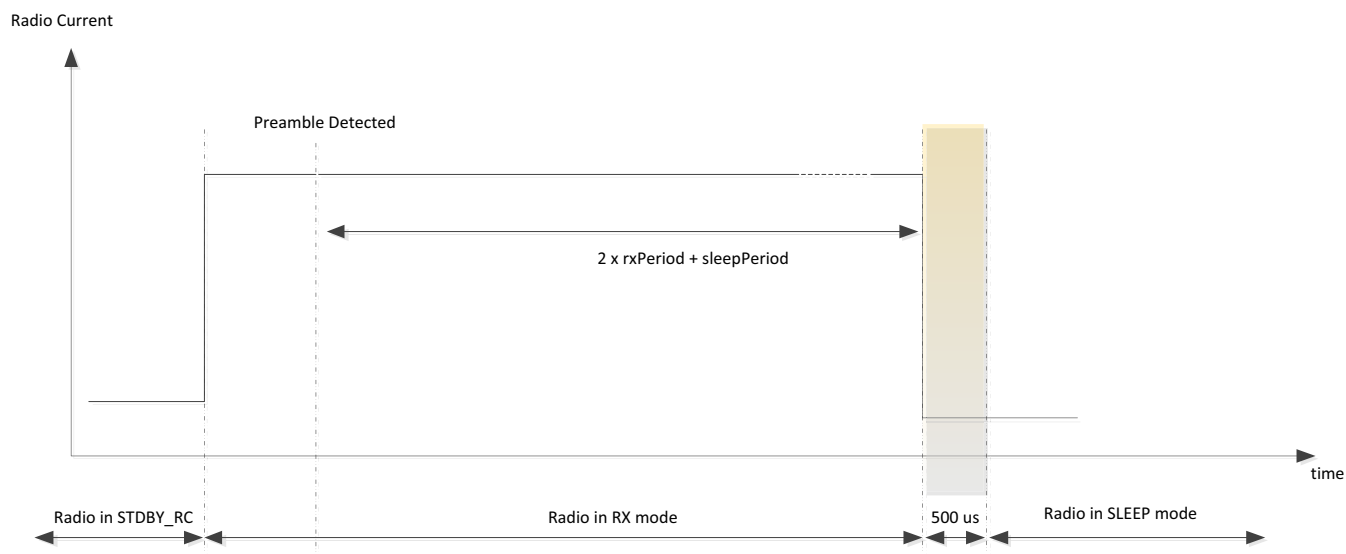


Figure 13-3: RX Duty Cycle when Receiving

Note: When using a TCXO controlled by the LLCC68 itself, the startup delay defined in *delay(23:0)* is added between the *Sleep* and *Rx* periods.

13.1.8 SetCAD

The command `SetCAD()` can only be used in LoRa packet types. The Channel Activity Detection is a LoRa specific mode of operation where the device searches for the presence of a LoRa preamble signal. After the search has completed, the device returns in STDBY_RC mode. The length of the search is configured via the command `SetCadParams(...)`. At the end of the search period, the device triggers the IRQ *CADdone* if it has been enabled. If a valid signal has been detected it also generates the IRQ *CadDetected*.

Table 13-13: SetCAD SPI Transaction

Byte	0
Data from host	Opcode = 0xC5

13.1.9 SetTxContinuousWave

`SetTxContinuousWave()` is a test command for all packet types to generate a continuous wave (RF tone) at selected frequency and output power. The device stays in TX continuous wave until the host sends a mode configuration command.

Table 13-14: SetTxContinuousWave SPI Transaction

Byte	0
Data from host	Opcode = 0xD1

While this command has no real use case in real life, it can provide valuable help to the developer to check and monitor the performances of the radio while in Tx mode.

13.1.10 SetTxInfinitePreamble

`SetTxInfinitePreamble()` is a test command that generates an infinite sequence of alternating zeros and ones in FSK modulation. In LoRa, the radio constantly modulates LoRa preamble symbols. The device remains in TX infinite preamble until the host sends a mode configuration command.

While this command has no real use case in real life, it can provide valuable help to the developer to check and monitor the performances of the radio while modulating in Tx mode.

Table 13-15: SetTxInfinitePreamble SPI Transaction

Byte	0
Data from host	Opcode = 0xD2

However, when using this function, it is impossible to define any data sent by the device. In LoRa mode, the radio is only able to constantly modulate LoRa preamble symbols and, in FSK mode, the radio is only able to generate FSK preamble (0x55). Nevertheless, the end user can easily monitor the spectral impact of its modulation parameters.

13.1.11 SetRegulatorMode

By default only the LDO is used. This is useful in low cost applications where the cost of the extra components needed for a DC-DC converter is prohibitive. Using only a linear regulator implies that the RX or TX current is almost doubled. This function specifies if DC-DC or LDO is used for power regulation. The regulation mode is defined by parameter *regModeParam*.

Note: This function is clearly related to the hardware implementation of the device. The user should always use this command while knowing what has been implemented at the hardware level.

Table 13-16: SetRegulatorMode SPI Transaction

Byte	0	1
Data from host	Opcode = 0x96	regModeParam 0: Only LDO used for all modes 1: DC_DC+LDO used for STBY_XOSC,FS, RX and TX modes

13.1.12 Calibrate Function

At power up the radio performs calibration of RC64k, RC13M, PLL and ADC. It is however possible to launch a calibration of one or several blocks at any time starting in STDBY_RC mode. The calibrate function starts the calibration of a block defined by *calibParam*.

Table 13-17: Calibrate SPI Transaction

Byte	0	1
Data from host	Opcode = 0x89	calibParam

The total calibration time if all blocks are calibrated is 3.5ms. The calibration must be launched in STDBY_RC mode and the BUSY pins are high during the calibration process. A falling edge of BUSY indicates the end of the procedure.

Table 13-18: Calibration Setting

CalibParam	Calibration Setting
Bit 0	0: RC64k calibration disabled 1: RC64k calibration enabled
Bit 1	0: RC13M calibration disabled 1: RC13M calibration enabled
Bit 2	0: PLL calibration disabled 1: PLL calibration enabled
Bit 3	0: ADC pulse calibration disabled 1: ADC pulse calibration enabled
Bit 4	0: ADC bulk N calibration disabled 1: ADC bulk N calibration enabled
Bit 5	0: ADC bulk P calibration disabled 1: ADC bulk P calibration enabled
Bit 6	0: Image calibration disabled 1: Image calibration enabled
Bit 7	0: RFU

13.1.13 CalibrateImage

The function `CalibrateImage(...)` allows the user to calibrate the image rejection of the device for the device operating frequency band.

Table 13-19: CalibrateImage SPI Transaction

Byte	0	1	2
Data from host	Opcode = 0x98	freq1	freq2

For more details on the specific frequency bands, see [Section 9.2.1 "Image Calibration for Specific Frequency Bands" on page 56](#).

13.1.14 SetPaConfig

The `SetPaConfig` is controlled as mentioned below.

Table 13-20: SetPaConfig SPI Transaction

Byte	0	1	2	3	4
Data from host	Opcode = 0x95	paDutyCycle	hpMax	deviceSel reserved and always 0x00	paLut reserved and always 0x01

- *paDutyCycle* controls the duty cycle (conduction angle). The maximum output power, the power consumption, and the harmonics drastically change with *paDutyCycle*. The values given across this datasheet are the recommended settings to achieve the best efficiency of the PA, obtained on Semtech's reference design PCB. Changing the *paDutyCycle* affects the distribution of the power in the harmonics and should be selected to work in conjunction with a given matching network.
- *hpMax* selects the size of the PA. The maximum output power can be reduced by reducing the value of *hpMax*. The valid range is between 0x00 and 0x07 and 0x07 is the maximum supported value to achieve +22dBm output power. Increasing *hpMax* above 0x07 could cause early aging of the device could damage the device when used in extreme temperatures.
- *deviceSel* is reserved and has always the value 0x00.
- *paLut* is reserved and has always the value 0x01.

13.1.14.1 PA Optimal Settings

PA optimal settings are used to maximize the PA efficiency when the requested output power is lower than the nominal +22dBm. For example, the maximum output power in Japan is +10dBm, and in China it is +17dBm in some bands. Those optimal settings require:

- a dedicated matching / PA load impedance
- a specific tweaking of the PA settings, described in [Table 13-21: PA Operating Modes with Optimal Settings](#)

Table 13-21: PA Operating Modes with Optimal Settings

Output Power	paDutyCycle	hpMax	deviceSel	paLut	Value in SetTxParams ¹
+22dBm	0x04	0x07	0x00	0x01	+22dBm
+20dBm	0x03	0x05	0x00	0x01	+22dBm
+17dBm	0x02	0x03	0x00	0x01	+22dBm
+14dBm	0x02	0x02	0x00	0x01	+22dBm

1. See [Section 13.4.4 "SetTxParams" on page 83](#).

Note: These changes make the use of nominal power either sub-optimal or unachievable.

Caution! The following restrictions must be observed to avoid voltage overstress on the PA, exceeding the maximum ratings may cause irreversible damage to the device: *paDutyCycle* should not be higher than 0x04.

13.1.15 SetRxTxFallbackMode

The command `SetRxTxFallbackMode()` defines into which mode the chip goes after a successful transmission or after a packet reception.

Table 13-22: SetRxTxFallbackMode SPI Transaction

Byte	0	1
Data from host	Opcode = 0x93	fallbackMode

The *fallbackMode* byte definition is given as follows:

Table 13-23: FallbackMode Definition

Fallback Mode	Value	Description
FS	0x40	The radio goes into FS mode after Tx or Rx
STDBY_XOSC	0x30	The radio goes into STDBY_XOSC mode after Tx or Rx
STDBY_RC	0x20	The radio goes into STDBY_RC mode after Tx or Rx

By default, the radio always returns in STDBY_RC unless the configuration is changed by using this command. Changing the default mode from STDBY_RC to STDBY_XOSC or FS impacts the switching time of the radio.

13.2 Registers and Buffer Access

13.2.1 WriteRegister Function

The command `WriteRegister(...)` writes a block of bytes in a data memory space starting at a specific address. The address is auto-incremented after each data byte so that data is stored in contiguous memory locations. The SPI data transfer is described in the following table.

Table 13-24: WriteRegister SPI Transaction

Byte	0	1	2	3	4	...	n
Data from host	Opcode = 0x0D	address[15:8]	address[7:0]	data@address	data@address+1	...	data@address+ (n-3)
Data to host	RFU	Status	Status	Status	Status	...	Status

13.2.2 ReadRegister Function

The command `ReadRegister(...)` reads a block of data starting at a given address. The address is auto-incremented after each byte. The SPI data transfer is described in [Table 13-25](#).

Note: The host must send an **NOP** after sending the 2 bytes of address to start receiving data bytes on the next NOP sent.

Table 13-25: ReadRegister SPI Transaction

Byte	0	1	2	3	4	5	...	n
Data from host	Opcode = 0x1D	address[15:8]	address[7:0]	NOP	NOP	NOP	...	NOP
Data to host	RFU	Status	Status	Status	data@address s	data@address+1	...	data@address+(n-4)

13.2.3 WriteBuffer Function

This function stores data payload to be transmitted. The address is auto-incremented; when it exceeds the value of 255 it is wrapped back to 0 due to the circular nature of the data buffer. The address starts with an offset set as a parameter of the function. [Table 13-26](#) describes the SPI data transfer.

Table 13-26: WriteBuffer SPI Transaction

Byte	0	1	2	3	...	n
Data from host	Opcode = 0x0E	offset	data@offset	data@offset+1	...	data@offset+(n-2)
Data to host	RFU	Status	Status	Status	...	Status

13.2.4 ReadBuffer Function

This function reads (n-3) bytes of payload received starting at offset.

Note: An NOP must be sent after sending the offset.

Table 13-27: ReadBuffer SPI Transaction

Byte	0	1	2	3	4	...	n
Data from host	Opcode = 0x1E	offset	NOP	NOP	NOP	...	NOP
Data to host	RFU	Status	Status	data@offset	data@offset+1	...	data@offset+(n-3)

13.3 DIO and IRQ Control Functions

13.3.1 SetDioIrqParams

This command sets the IRQ flag.

Table 13-28: SetDioIrqParams SPI Transaction

Byte	0	1-2	3-4	5-6	7-8
Data from host	SetDioIrqParams (0x08)	IrqMask(15:0)	DIO1Mask(15:0)	DIO2Mask(15:0)	DIO3Mask(15:0)

13.3.2 IrqMask

The *IrqMask* masks or unmasks the IRQ which can be triggered by the device. By default, all IRQ are masked (all '0') and the user can enable them one by one (or several at a time) by setting the corresponding mask to '1'.

13.3.2.1 DioxMask

The interrupt causes a DIO to be set if the corresponding bit in *DioxMask* and the *IrqMask* are set. As an example, if bit 0 of *IrqMask* is set to 1 and bit 0 of *Dio1Mask* is set to 1 then, a rising edge of IRQ source TxDone is logged in the IRQ register and appears at the same time on DIO1.

One IRQ can be mapped to all DIOs, one DIO can be mapped to all IRQs (an OR operation is done) but some IRQ sources are available only on certain modes of operation and frames.

In total there are 10 possible interrupt sources depending on the chosen frame and chip mode. Each one of them can be enabled or masked. In addition, every one of them can be mapped to DIO1, DIO2 or DIO3.

Note: If DIO2 or DIO3 are used to control the RF Switch or the TCXO, the IRQ is not generated even if it is mapped to the pins.

A dedicated 10-bit register called *IRQ_reg* is used to log IRQ sources. Each position corresponds to one IRQ source as described in the table above. A set of user commands is used to configure IRQ mask, DIOs mapping and IRQ clearing as explained in the following chapters.

Table 13-29: IRQ Registers

Bit	IRQ	Description	Modulation
0	TxDone	Packet transmission completed	All
1	RxDone	Packet received	All
2	PreambleDetected	Preamble detected	All
3	SyncWordValid	Valid sync word detected	FSK
4	HeaderValid	Valid LoRa header received	LoRa
5	HeaderErr	LoRa header CRC error	LoRa
6	CrcErr	Wrong CRC received	All
7	CadDone	Channel activity detection finished	LoRa
8	CadDetected	Channel activity detected	LoRa
9	Timeout	Rx or Tx timeout	All
10-13	-	RFU	-
15	-	RFU	-

13.3.3 GetIrqStatus

This command returns the value of the IRQ register.

Table 13-30: GetIrqStatus SPI Transaction

Byte	0	1	2-3
Data from host	Opcode = 0x12	NOP	NOP
Data to host	RFU	Status	IrqStatus(15:0)

13.3.4 ClearIrqStatus

This command clears an IRQ flag in the IRQ register.

Table 13-31: ClearIrqStatus SPI Transaction

Byte	0	1-2
Data from host	Opcode = 0x02	ClearIrqParam(15:0)

This function clears an IRQ flag in the IRQ register by setting to 1 the bit of *ClearIrqParam* corresponding to the same position as the IRQ flag to be cleared. As an example, if bit 0 of *ClearIrqParam* is set to 1 then IRQ flag at bit 0 of IRQ register is cleared.

If a DIO is mapped to one single IRQ source, the DIO is cleared if the corresponding bit in the IRQ register is cleared. If DIO is set to 0 with several IRQ sources, then the DIO remains set to one until all bits mapped to the DIO in the IRQ register are cleared.

13.3.5 SetDIO2AsRfSwitchCtrl

This command is used to configure DIO2 so that it can be used to control an external RF switch.

Table 13-32: SetDIO2AsRfSwitchCtrl SPI Transaction

Byte	0	1
Data from host	Opcode = 0x9D	enable

When controlling the external RX switch, the pin DIO2 toggles accordingly to the internal state machine. DIO2 is asserted high a few microseconds before the ramp-up of the PA and is set to zero after the ramp-down of the PA.

The enable byte definition is given as follows:

Table 13-33: Enable Configuration Definition

Enable	Description
0	DIO2 is free to be used as an IRQ
1	DIO2 is selected to be used to control an RF switch. In this case: DIO2 = 0 in SLEEP, STDBY_RX, STDBY_XOSC, FS and RX modes, DIO2 = 1 in TX mode

13.3.6 SetDIO3asTCXOCtrl

This command configures the chip for an external TCXO reference voltage controlled by DIO3.

Table 13-34: SetDIO3asTCXOCtrl SPI Transaction

Byte	0	1	2-4
Data from host	Opcode = 0x97	tcxoVoltage	delay(23:0)

When this command is used, the device now controls the TCXO itself through DIO3. When needed (in mode STDBY_XOSC, FS, TX and RX), the internal state machine is set DIO3 to a predefined output voltage (control through the parameter *tcxoVoltage*). Internally, the clock controller waits for the 32MHz to appear before releasing the internal state machine.

The time needed for the 32MHz to appear and stabilize can be controlled through the parameter *delay(23:0)*. If the 32MHz from the TCXO is not detected internally at the end the delay period, the error XOSC_START_ERR is flagged in the error controller.

The XOSC_START_ERR flag is raised at POR or at wake-up from Sleep mode in a cold-start condition, when a TCXO is used. It is an expected behaviour since the chip is not yet aware of being clocked by a TCXO. The user should simply clear this flag with the `ClearDeviceErrors` command.

The *tcxoVoltage* byte definition is given in as follows:

Table 13-35: tcxoVoltage Configuration Definition

tcxoVoltage	Description
0x00	DIO3 outputs 1.6V to supply the TCXO
0x01	DIO3 outputs 1.7V to supply the TCXO
0x02	DIO3 outputs 1.8V to supply the TCXO
0x03	DIO3 outputs 2.2V to supply the TCXO
0x04	DIO3 outputs 2.4V to supply the TCXO
0x05	DIO3 outputs 2.7V to supply the TCXO
0x06	DIO3 outputs 3.0V to supply the TCXO
0x07	DIO3 outputs 3.3V to supply the TCXO

tcxoVoltage precision varies with the *tcxoVoltage* setting value. It is typically +/-50mV for 1.8V setting.

The power regulation for *tcxoVoltage* is configured to be 200mV below the supply voltage. This means that even if *tcxoVoltage* is configured above the supply voltage, the supply voltage is limited by: **VDDop > VTCXO + 200mV**

The timeout duration is defined by: **Delay duration = Delay(23:0) * 15.625μs**

Most TCXO is not be immediately ready at the desired frequency and suffers from an initial setup time where the frequency is gently drifting towards the wanted frequency. This setup time varies from one TCXO to another and is also dependent on the TCXO manufacturer. To ensure this setup time does not have any effect on the modulation or packets, the delay value internally gates the 32MHz coming from the TCXO to give enough time for this initial drift to stabilize. At the end of the delay period, the internal block stops gating the clock and the radio carries on to the next step.

Note: The user should take the delay period into account when going into Tx or Rx mode from STDBY_RC mode, since the time needed to switch modes increases with the duration of delay. To avoid increasing the switching mode time, the user can first set the device in STDBY_XOSC which switches on the TCXO and wait for the delay period. Then, the user can set the device into Tx or Rx mode without suffering from any delay additional to the internal processing.

13.4 RF Modulation and Packet-Related Functions

13.4.1 SetRfFrequency

The command `SetRfFrequency(...)` sets the frequency of the RF frequency mode.

Table 13-36: SetRfFrequency SPI Transaction

Byte	0	1-4
Data from host	Opcode = 0x86	RfFreq(31:0)

The **LSB** of Freq is equal to the **PLL** step:

$$RF_{frequency} = \frac{RF_{Freq} * F_{XTAL}}{2^{25}}$$

`SetRfFrequency(...)` defines the chip frequency in FS, TX and RX modes. In RX, the required IF frequency offset is automatically configured.

13.4.2 SetPacketType

The command `SetPacketType(...)` sets the LLCC68 radio in LoRa, or **FSK**, mode. The command `SetPacketType(...)` must be the first of the radio configuration sequence. The parameter for this command is *PacketType*.

Table 13-37: SetPacketType SPI Transaction

Byte	0	1
Data from host	Opcode = 0x8A	PacketType

Table 13-38: PacketType Definition

PacketType	Value	Modem Mode of Operation
PACKET_TYPE_GFSK	0x00	GFSK packet type
PACKET_TYPE_LoRa	0x01	LoRa mode

Changing from one mode of operation to another is done using the command `SetPacketType(...)`. The parameters from the previous mode are not kept internally. The switch from one packet type to another must be done in STDBY_RC mode.

13.4.3 GetPacketType

The command `GetPacketType()` returns the current operating packet type of the radio.

Table 13-39: GetPacketType SPI Transaction

Byte	0	1	2
Data from host	Opcode = 0x11	NOP	NOP
Data to host	RFU	Status	packetType

13.4.4 SetTxParams

This command sets the TX output power by using the parameter *power* and the TX ramping time by using the parameter *RampTime*. This command is available for all protocols selected.

Table 13-40: SetTxParams SPI Transaction

Byte	0	1	2
Data from host	Opcode = 0x8E	power	RampTime

The output power is defined as *power* in dBm in a range of -9 (0xF7) to +22 (0x16)dBm by steps of 1dB.

The power ramp time is defined by the parameter *RampTime* as defined in the following table:

Table 13-41: RampTime Definition

RampTime	Value	RampTime (μs)
SET_RAMP_10U	0x00	10
SET_RAMP_20U	0x01	20
SET_RAMP_40U	0x02	40
SET_RAMP_80U	0x03	80
SET_RAMP_200U	0x04	200
SET_RAMP_800U	0x05	800
SET_RAMP_1700U	0x06	1700
SET_RAMP_3400U	0x07	3400

13.4.5 SetModulationParams

The command `SetModulationParams(...)` configures the modulation parameters of the radio. Depending on the packet type selected prior to calling this function, the parameters is interpreted differently by the chip.

Table 13-42: SetModulationParams SPI Transaction

Byte	0	1	2	3	4	5	6	7	8
Data from host for Modulation Params	Opcode 0x8B	Mod Param1	Mod Param2	Mod Param3	Mod Param4	Mod Param5	Mod Param6	Mod Param7	Mod Param8

The meaning of the parameter depends on the selected protocol.

In [FSK](#), bitrate (BR) and Frequency Deviation (Fdev) are used for the transmission or reception. Bandwidth is used for reception purposes. The pulse represents the Gaussian filter used to filter the modulation stream on the transmitter side.

ModParam[1] defines SF in LoRa packet types, where SF corresponds to the Spreading Factor used for the LoRa modulation.

ModParam[2] defines BW in LoRa, where BW corresponds to the bandwidth onto which the LoRa signal is spread.

ModParam[3] defines Coding Rate (CR) in LoRa. The LoRa payload includes a forward error correcting mechanism which has several levels of encoding.

ModParam[4] defines the parameter *LdOpt* which corresponds to the Low Data Rate Optimization (LDRO). This parameter is usually set when the LoRa symbol time is equal or above 16.38ms but can be used if necessary in other situations. See [Section 6.1.1.4 "Low Data Rate Optimization" on page 36](#).

13.4.5.1 GFSK Modulation Parameters

The tables below provide more details on the GFSK modulation parameters:

Table 13-43: GFSK ModParam1, ModParam2 & ModParam3 - br

BR(23:0)	Description
0x000001 to 0xFFFFF	$br = 32 * F_{xtal} / \text{bit rate}$

The bit rate is entered with the parameter *br* which is related to the frequency of the main oscillator (32MHz). The bit rate range is from 600b/s up to 500kb/s with a default value at 4.8kb/s.

Table 13-44: GFSK ModParam4 - PulseShape

PulseShape	Description
0x00	No Filter applied
0x08	Gaussian BT 0.3
0x09	Gaussian BT 0.5
0x0A	Gaussian BT 0.7
0x0B	Gaussian BT 1

Table 13-45: GFSK ModParam5 - Bandwidth

Bandwidth	Description
0x1F	RX_BW_4800 (4.8kHz DSB)
0x17	RX_BW_5800 (5.8kHz DSB)
0x0F	RX_BW_7300 (7.3kHz DSB)
0x1E	RX_BW_9700 (9.7kHz DSB)
0x16	RX_BW_11700 (11.7kHz DSB)
0x0E	RX_BW_14600 (14.6kHz DSB)
0x1D	RX_BW_19500 (19.5kHz DSB)
0x15	RX_BW_23400 (23.4kHz DSB)
0x0D	RX_BW_29300 (29.3kHz DSB)
0x1C	RX_BW_39000 (39kHz DSB)
0x14	RX_BW_46900 (46.9kHz DSB)
0x0C	RX_BW_58600 (58.6kHz DSB)
0x1B	RX_BW_78200 (78.2kHz DSB)
0x13	RX_BW_93800 (93.8kHz DSB)
0x0B	RX_BW_117300 (117.3kHz DSB)
0x1A	RX_BW_156200 (156.2kHz DSB)
0x12	RX_BW_187200 (187.2kHz DSB)
0x0A	RX_BW_234300 (232.3kHz DSB)
0x19	RX_BW_312000 (312kHz DSB)
0x11	RX_BW_373600 (373.6kHz DSB)
0x09	RX_BW_467000 (467kHz DSB)

Table 13-46: GFSK ModParam6, ModParam7 & ModParam8 - Fdev

Fdev(23:0)	Description
0x000000 to 0xFFFFFFFF	Fdev = (Frequency Deviation * 2 ²⁵) / Fxtal

$$Frequencydeviation = \frac{F_{dev} * F_{XTAL}}{2^{25}}$$

13.4.5.2 LoRa Modulation Parameters

The tables below provide more details on the LoRa modulation parameters:

Table 13-47: LoRa ModParam1- SF

SF	Description
0x05	SF5
0x06	SF6
0x07	SF7
0x08	SF8
0x09	SF9
0x0A	SF10
0x0B	SF11

Table 13-48: LoRa ModParam2 - BW

BW	Description
0x04	LoRa_BW_125 (125kHz real)
0x05	LoRa_BW_250 (250kHz real)
0x06	LoRa_BW_500 (500kHz real)

Table 13-49: LoRa ModParam3 - CR¹

CR	Description
0x01	LoRa_CR_4_5
0x02	LoRa_CR_4_6
0x03	LoRa_CR_4_7
0x04	LoRa_CR_4_8
0x05	LoRa_CR_4_5_LI
0x06	LoRa_CR_4_6_LI
0x07	LoRa_CR_4_8_LI

1. Long interleaver increases robustness against interferers

Table 13-50: LoRa ModParam4 - LowDataRateOptimize

LowDataRateOptimize	Description
0x00	LowDataRateOptimize OFF
0x01	LowDataRateOptimize ON

13.4.6 SetPacketParams

This command sets the parameters of the packet handling block.

Table 13-51: SetPacketParams SPI Transaction

Byte	0	1	2	3	4	5	6	7	8	9
Data from host for packet type	Opcode = 0x8C	packet Param1	packet Param2	packet Param3	packet Param4	packet Param5	packet Param6	packet Param7	packet Param8	packet Param9

13.4.6.1 GFSK Packet Parameters

The tables below provide more details on the GFSK packet parameters:

Table 13-52: GFSK PacketParam1 & PacketParam2 - PreambleLength

PreambleLength (15:0)	Description
0x0001 to 0xFFFF	Transmitted preamble length: number of bits sent as preamble

The preamble length is a 16-bit value which represents the number of bytes which are sent by the radio. Each preamble byte represents an alternating 0 and 1, and each byte is coded as 0x55.

Table 13-53: GFSK PacketParam3 - PreambleDetectorLength

PreambleDetector	Description
0x00	Preamble detector length off
0x04	Preamble detector length 8 bits
0x05	Preamble detector length 16 bits
0x06	Preamble detector length 24 bits
0x07	Preamble detector length 32 bits

The preamble detector acts as a gate to the packet controller, when different from 0x00 (preamble detector length off), the packet controller only becomes active if a certain number of preamble bits have been successfully received by the radio.

Table 13-54: GFSK PacketParam4 - SyncWordLength

SyncWordLength	Description
0x00 to 0x40	Sync Word length in bits (going from 0 to 8 bytes)

The Sync Word is directly programmed into the device through simple register access. The tables below provide the addresses to program the Sync Word value. If the Sync Word is not aligned on 8 bits, it must be padded with a preamble pattern.

Table 13-55: Sync Word Programming

Sync Word	Register Address
Byte 0	0x06C0
Byte 1	0x06C1
Byte 2	0x06C2
Byte 3	0x06C3
Byte 4	0x06C4
Byte 5	0x06C5
Byte 6	0x06C6
Byte 7	0x06C7

Table 13-56: GFSK PacketParam5 - AddrComp

AddrComp	Description
0x00	Address Filtering Disable
0x01	Address Filtering activated on Node address
0x02	Address Filtering activated on Node and broadcast addresses

The node address and the broadcast address are directly programmed into the device through simple register access. The tables below provide the addresses to program the values.

Table 13-57: Node Address Programming

	Register Address	Default value
NodeAddrReg	0x06CD	0x00

Table 13-58: Broadcast Address Programming

	Register Address	Default value
BroadcastReg	0x06CE	0x00

Table 13-59: GFSK PacketParam6 - PacketType

PacketType	Description
0x00	The packet length is known on both sides, the size of the payload is not added to the packet
0x01	The packet is on variable size, the first byte of the payload is the size of the packet

Table 13-60: GFSK PacketParam7 - PayloadLength

AddrComp	Description
0x00 to 0xFF	Size of the payload (in bytes) to transmit or maximum size of the payload that the receiver can accept.

Table 13-61: GFSK PacketParam8 - CRCType

CRCType	Description
0x01	CRC_OFF (No CRC)
0x00	CRC_1_BYTE (CRC computed on 1 byte)
0x02	CRC_2_BYTE (CRC computed on 2 byte)
0x04	CRC_1_BYTE_INV (CRC computed on 1 byte and inverted)
0x06	CRC_2_BYTE_INV (CRC computed on 2 byte and inverted)

In the LLCC68, the CRC can be fully configured and the polynomial used, and the initial values can be entered directly via register access.

Table 13-62: CRC Initial Value Programming

	Register Address	Default Value
CRC MSB Initial Value [15:8]	0x06BC	0x1D
CRC LSB Initial Value [7:0]	0x06BD	0x0F

Table 13-63: CRC Polynomial Programming

	Register Address	Default Value
CRC MSB polynomial value [15:8]	0x06BE	0x10
CRC LSB polynomial value [7:0]	0x06BF	0x21

Table 13-64: GFSK PacketParam9 - Whitening

AddrComp	Description
0x00	No encoding
0x01	Whitening enable

Table 13-65: Whitening Initial Value

Whitening initial value	Register Address	Default Value
Whitening initial value MSB	0x06B8	0x01
Whitening initial value LSB	0x06B9	0x00

13.4.6.2 LoRa Packet Parameters

The tables below provide more details on the LoRa packet parameters:

Table 13-66: LoRa PacketParam1 & PacketParam2 - PreambleLength

PreambleLength (15:0)	Description
0x0001 to 0xFFFF	Preamble length: number of symbols sent as preamble

The preamble length is a 16-bit value which represents the number of LoRa symbols which are sent by the radio.

Table 13-67: LoRa PacketParam3 - HeaderType

HeaderType	Description
0x00	Variable length packet (explicit header)
0x01	Fixed length packet (implicit header)

When the byte HeaderType is at 0x00, the payload length, coding rate and the header CRC are added to the LoRa header and transported to the receiver.

Table 13-68: LoRa PacketParam4 - PayloadLength

Payloadlength	Description
0x00 to 0xFF	Size of the payload (in bytes) to transmit or maximum size of the payload that the receiver can accept.

Table 13-69: LoRa PacketParam5 - CRCType

CRCType	Description
0x00	CRC OFF
0x01	CRC ON

Table 13-70: LoRa PacketParam6 - InvertIQ

AddrComp	Description
0x00	Standard IQ setup
0x01	Inverted IQ setup

13.4.7 SetCadParams

The command `SetCadConfig(...)` defines the number of symbols on which CAD operates.

Table 13-71: SetCadParams SPI Transaction

Byte	0	1	2	3	4	5-7
Data from host	Opcode = 0x88	cadSymbolNum	cadDetPeak	cadDetMin	cadExitMode	cadTimeout(23:0)

The number of symbols used is defined in the following table.

Table 13-72: CAD Number of Symbol Definition

cadSymbolNum	Value	Number of Symbols used for CAD
CAD_ON_1_SYMB	0x00	1
CAD_ON_2_SYMB	0x01	2
CAD_ON_4_SYMB	0x02	4
CAD_ON_8_SYMB	0x03	8
CAD_ON_16_SYMB	0x04	16

Parameters *cadDetPeak* and *cadDetMin* define the sensitivity of the LoRa modem when trying to correlate to actual LoRa preamble symbols. These two settings depend on the LoRa spreading factor and Bandwidth, but also depend on the number of symbols used to validate or not the detection. Choosing the right value is not easy and the values selected must be carefully tested to ensure a good detection at sensitivity level, and also to limit the number of false detections. Application note AN1200.48 provides guidance for the selection of these parameters.

The parameter *cadExitMode* defines the action to be done after a CAD operation. This is optional.

Table 13-73: CAD Exit Mode 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 STBY_RC mode.
CAD_RX	0x01	The chip performs a CAD operation and if an activity is detected, it stays in RX until a packet is detected or the timer reaches the timeout defined by <i>cadTimeout</i> * 15.625us

The parameter *cadTimeout* is only used when the CAD is performed with *cadExitMode* = CAD_RX. Here, the *cadTimeout* indicates the time the device stays in Rx following a successful CAD.

$$Rx\ Timeout = cadTimeout * 15.625$$

13.4.8 SetBufferBaseAddress

This command sets the base addresses in the data buffer, in all modes of operation, for the packet handing operation in TX and RX modes. The usage and definition of these parameters are described in the relevant packet type sections.

Table 13-74: SetBufferBaseAddress SPI Transaction

Byte	0	1	2
Data from host	Opcode = 0x8F	TX base address	RX base address

13.4.9 SetLoRaSymbNumTimeout

This command sets the number of symbols used by the modem to validate a successful reception.

Table 13-75: SetLoRaSymbNumTimeout SPI Transaction

Byte	0	1
Data from host	Opcode = 0xA0	SymbNum

In LoRa mode, when going into Rx, the modem locks as soon as a LoRa symbol has been detected which may lead to false detection. This phenomena is quite rare but nevertheless possible. To avoid this, the command SetLoRaSymbNumTimeout can be used to define the number of symbols which are used to validate the correct reception of a packet.

SymbNum valid values are even numbers up to 64, then intervals of 8 up to 248.

- When *SymbNum* is 0, the modem validates the reception as soon as a LoRa Symbol has been detected.
- When *SymbNum* is not 0, the modem waits for a total of *SymbNum* LoRa symbols to validate, or not, the correct detection of a LoRa packet. If the various states of the demodulator are not locked at this moment, the radio generates an RxTimeout IRQ.

13.5 Communication Status Information

These commands return the information about the chip status, and received packet such a packet length, received power during packet, several flags indicating if the packet as been correctly received. The returned parameters differ for the LoRa protocol.

13.5.1 GetStatus

The host can retrieve chip status directly through the command `GetStatus()`: this command can be issued at any time and the device returns the status of the device. The command `GetStatus()` is not strictly necessary since device returns status information also on command bytes. The status byte returned is described in [Table 13-76](#).

Table 13-76: Status Bytes Definition

7	6:4	3:1	0
Reserved	Chip mode	Command status	Reserved
	0x0: Unused	0x0: Reserved	
	RFU	RFU	
	0x2: STBY_RC	0x2: Data is available to host ¹	
-	0x3: STBY_XOSC	0x3: Command timeout ²	-
	0x4: FS	0x4: Command processing error ³	
	0x5: RX	0x5: Failure to execute command ⁴	
	0x6: TX	0x6: Command TX done ⁵	

1. A packet has been successfully received and data can be retrieved

2. A transaction from host took too long to complete and triggered an internal watchdog. The watchdog mechanism can be disabled by host; it is meant to ensure all outcomes are flagged to the host MCU

3. Processor was unable to process command either because of an invalid opcode or because an incorrect number of parameters has been provided

4. The command was successfully processed, however the chip could not execute the command; for instance it was unable to enter the specified device mode or send the requested data

5. The transmission of the current packet has terminated

The SPI transaction for the command `GetStatus()` is given in the following table.

Table 13-77: GetStatus SPI Transaction

Byte	0	1
Data from host	Opcode = 0xC0	NOP
Data to host	RFU	Status

13.5.2 GetRxBufferStatus

This command returns the length of the last received packet (*PayloadLengthRx*) and the address of the first byte received (*RxStartBufferPointer*). It is applicable to all modems. The address is an offset relative to the first byte of the data buffer.

Table 13-78: GetRxBufferStatus SPI Transaction

Byte	0	1	2	3
Data from host	Opcode = 0x13	NOP	NOP	NOP
Data to host	RFU	Status	PayloadLengthRx	RxStartBufferPointer

13.5.3 GetPacketStatus

Table 13-79: GetPacketStatus SPI Transaction

Byte	0	1	2	3	4
Data from host	Opcode = 0x14	NOP	NOP	NOP	NOP
Data to host for FSK packet type	RFU	Status	RxStatus	RssiSync	RssiAvg
Data to host for LoRa packet type	RFU	Status	RssiPkt	SnrPkt	SignalRssiPkt

The next table gives the description of the different RSSI and SNR available on the chip depending on the packet type.

Table 13-80: Status Fields (Sheet 1 of 2)

RSSI	Description
RxStatus	bit 7: preamble err
	bit 6: sync err
	bit 5: adrs err
	bit 4: crc err
FSK	bit 3: length err
	bit 2: abort err
	bit 1: pkt received
	bit 0: pkt sent
RssiSync	RSSI value latched upon the detection of the sync address.
FSK	[negated,dBm, fixdt(0,8,1)] Actual signal power is $-RssiSync/2$ (dBm)
RssiAvg	RSSI average value over the payload of the received packet. Latched upon the pkt_done IRQ.
FSK	[negated,dBm, fixdt(0,8,1)] Actual signal power is $-RssiAvg/2$ (dBm)
RssiPkt	Average over last packet received of RSSI
LoRa	Actual signal power is $-RssiPkt/2$ (dBm)

Table 13-80: Status Fields (Sheet 2 of 2)

RSSI	Description
SnrPkt LoRa	Estimation of SNR on last packet received in two's complement format multiplied by 4. Actual SNR in dB = SnrPkt/4
SignalRssiPkt LoRa	Estimation of RSSI of the LoRa signal (after despreading) on last packet received Actual Rssi in dB = -SignalRssiPkt/2

13.5.4 GetRssiInst

This command returns the instantaneous RSSI value during reception of the packet. The command is valid for all protocols.

Table 13-81: GetRssiInst SPI Transaction

Byte	0	1	2
Data from host	Opcode = 0x15	NOP	NOP
Data to host	RFU	Status	RssiInst Signal power indBm = -RssiInst/2 (dBm)

With GFSK, the RSSI settling time varies with the BW, as described in the following table. The RSSI delay is the delay between the moment the chip is put in Rx mode i.e. busy falling edge, and the moment the first RSSI value is available. The RSSI averaging window is the delay between two RSSI samples.

Table 13-82: RSSI Settling Time per Bandwidth

Bandwidth	RSSI delay (μs)	RSSI averaging window (μs)
LLCC6X_GFSK_BW_4800	2295	192
LLCC6X_GFSK_BW_5800	1913	160
LLCC6X_GFSK_BW_7300	1531	128
LLCC6X_GFSK_BW_9700	1152	96
LLCC6X_GFSK_BW_11700	960	80
LLCC6X_GFSK_BW_14600	769	64
LLCC6X_GFSK_BW_19500	581	48
LLCC6X_GFSK_BW_23400	484	40
LLCC6X_GFSK_BW_29300	388	32
LLCC6X_GFSK_BW_39000	295	24
LLCC6X_GFSK_BW_46900	246	20
LLCC6X_GFSK_BW_58600	197	16
LLCC6X_GFSK_BW_78200	152	12
LLCC6X_GFSK_BW_93800	127	10

Table 13-82: RSSI Settling Time per Bandwidth

Bandwidth	RSSI delay (μ s)	RSSI averaging window (μ s)
LLCC6X_GFSK_BW_117300	102	8
LLCC6X_GFSK_BW_156200	80	6
LLCC6X_GFSK_BW_187200	67	5
LLCC6X_GFSK_BW_234300	54	4
LLCC6X_GFSK_BW_312000	44	3
LLCC6X_GFSK_BW_373600	37	2.5
LLCC6X_GFSK_BW_467000	30	2

13.5.5 GetStats

This command returns the number of informations received on a few last packets. The command is valid for all protocols.

Table 13-83: GetStats SPI Transaction

Byte	0	1	2-3	4-5	6-7
Data from host	Opcode = 0x10	NOP	NOP	NOP	NOP
Data to host in GFSK packet type	RFU	Status	NbPktReceived(15:0)	NbPktCrcError(15:0)	NbPktLengthError(15:0)
Data to host in LoRa packet type	RFU	Status	NbPktReceived(15:0)	NbPktCrcError(15:0)	NbPktHeaderErr(15:0)

13.5.6 ResetStats

This command resets the value read by the command GetStats. To execute this command, the OpCode is 0x00 followed by 6 zeros (so 7 zeros in total).

Table 13-84: ResetStats SPI Transaction

Byte	0	1-6
Data from host	OpCode = 0x00	0x00

13.6 Miscellaneous

13.6.1 GetDeviceErrors

This command returns possible errors flag that could occur during different chip operation as described below.

Table 13-85: GetDeviceErrors SPI Transaction

Byte	0	1	2-3
Data from host	Opcode = 0x17	NOP	NOP
Data to host	RFU	Status	OpError(15:0)

The following table gives the meaning of each OpError.

Table 13-86: OpError Bits

OpError	0	1
bit 0	RC64K_CALIB_ERR	RC64k calibration failed
bit 1	RC13M_CALIB_ERR	RC13M calibration failed
bit 2	PLL_CALIB_ERR	PLL calibration failed
bit 3	ADC_CALIB_ERR	ADC calibration failed
bit 4	IMG_CALIB_ERR	IMG calibration failed
bit 5	XOSC_START_ERR	XOSC failed to start
bit 6	PLL_LOCK_ERR	PLL failed to lock
bit 7	RFU	RFU
bit 8	PA_RAMP_ERR	PA ramping failed
bit 15:9	RFU	RFU

13.6.2 ClearDeviceErrors

This command clears all the errors recorded in the device. The errors can not be cleared independently.

Table 13-87: ClearDeviceErrors SPI Transaction

Byte	0	1	2
Data from host	Opcode = 0x07	0x00	0x00
Data to host	RFU	Status	Status

14. Application

14.1 HOST API Basic Read Write Function

Communication with the LLCC68 is organized around generic functions which allow users to control the device behavior. Each function is based on an Operational Command (referred to throughout this document as “Opcode”) followed by a set of parameters. The LLCC68 uses the BUSY pin to indicate the status of the chip. In the following chapters, it is assumed that host microcontroller has an SPI and access to it via `spi.write(data)`. Data is an 8-bit word. The SPI chip select is defined by NSS, active low.

14.2 Circuit Configuration for Basic Tx Operation

This chapter describes the sequence of operations needed to send or receive a frame starting from a power up.

After power up (battery insertion or hard reset) the chip runs automatically a calibration procedure and goes to STDBY_RC mode. This is indicated by a low state on BUSY pin. From this state the steps are:

1. If not in STDBY_RC mode, then go to this mode with command `SetStandby(...)`
2. Define the protocol (LoRa or FSK) with command `SetPacketType(...)`
3. Define the RF frequency with command `SetRfFrequency(...)`
4. Define the Power Amplifier configuration with command `SetPaConfig(...)`
5. Define output power and ramping time with command `SetTxParams(...)`
6. Define where the data payload is stored with command `SetBufferBaseAddress(...)`
7. Send the payload to the data buffer with command `WriteBuffer(...)`
8. Define the modulation parameter according to the chosen protocol with command `SetModulationParams(...)`¹
9. Define the frame format to be used with command `SetPacketParams(...)`²
10. Configure DIO and IRQ: use command `SetDioIrqParams(...)` to select *TxDone IRQ* and map this IRQ to a DIO (DIO1, DIO2 or DIO3)
11. Define Sync Word value: use command `WriteReg(...)` to write the value of the register via direct register access
12. Set the circuit in transmitter mode to start transmission with command `SetTx()`. Use the parameter to enable *Timeout*
13. Wait for the IRQ *TxDone* or *Timeout*: once the packet has been sent the chip goes automatically to STDBY_RC mode
14. Clear the IRQ *TxDone* flag

1. Please refer to [Section 15.1](#)

2. Please refer to [Section 15.2](#)

14.3 Circuit Configuration for Basic Rx Operation

This chapter describes the sequence of operations needed to receive a frame starting from a power up. This sequence is valid for all protocols.

After power up (battery insertion or hard reset) the chip runs automatically a calibration procedure and goes to STDBY_RC mode. This is indicated by a low state on BUSY pin. From this state the steps are:

1. If not in STDBY_RC mode, then set the circuit in this mode with command `SetStandby()`.
2. Define the protocol (LoRa or **FSK**) with command `SetPacketType(...)`.
3. Define the RF frequency with command `SetRfFrequency(...)`.
4. Define where the data is stored inside the data buffer in Rx with command `SetBufferBaseAddress(...)`.
5. Define the modulation parameter according to the chosen protocol with command `SetModulationParams(...)`¹.
6. Define the frame format to be used with command `SetPacketParams(...)`.
7. Configure DIO and IRQ: use command `SetDioIrqParams(...)` to select the IRQ *RxDone* and map this IRQ to a DIO (DIO1 or DIO2 or DIO3), set IRQ *Timeout* as well.
8. Define Sync Word value: use command `WriteReg(...)` to write the value of the register via direct register access.
9. Set the circuit in reception mode: use command `SetRx()`. Set the parameter to enable timeout or continuous mode.
10. Wait for IRQ *RxDone*² or *Timeout*: the chip stays in Rx and looks for a new packet if the continuous mode is selected otherwise it goes to STDBY_RC mode.
11. In case of the IRQ *RxDone*, check the status to ensure CRC is correct: use the command `GetIrqStatus()`.
Note: The IRQ *RxDone* means that a packet has been received but the CRC could be wrong: the user must check the CRC before validating the packet.
12. Clear IRQ flag *RxDone* or *Timeout*: use command `ClearIrqStatus()`. In case of a valid packet (CRC OK), get the packet length and address of the first byte of the received payload by using the command `GetRxBufferStatus(...)`.
13. In case of a valid packet (CRC OK), start reading the packet.

14.4 Issuing Commands in the Right Order

Most of the commands can be sent in any order except for the radio configuration commands which sets the radio in the proper operating mode. Indeed, it is mandatory to set the radio protocol using the command `SetPacketType(...)` as a first step before issuing any other radio configuration commands. In a second step, the user should define the modulation parameter according to the chosen protocol with the command `SetModulationParams(...)`. Finally, the user should then select the packet format with the command `SetPacketParams(...)`.

Note: If this order is not respected, the behavior of the device could be unexpected.

1. Please refer to [Section 15.4](#)
 2. Please refer to [Section 15.3](#)

14.5 Application Schematics

14.5.1 Application Design of the LLCC68 with RF Switch

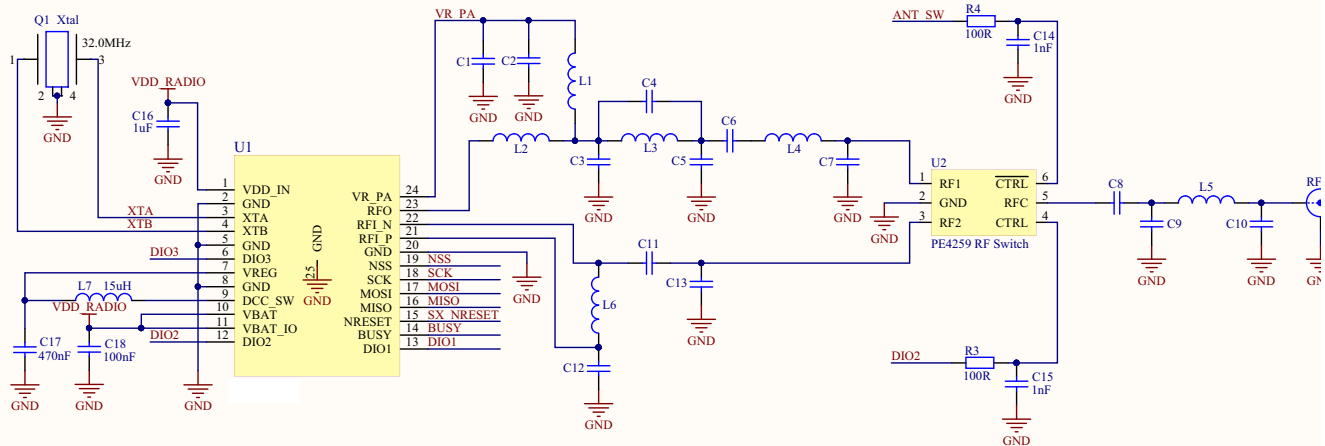


Figure 14-1: Application Schematic of the LLCC68 with RF Switch

Note: The application schematics presented here are for information only. Always refer to the latest reference designs posted on www.semtech.com.

Recommendations for heat dissipation techniques to be applied to the PCB designs are given in detail in the application note AN1200.37 "Recommendations for Best Performance" on www.semtech.com.

In miniaturized design implementations where heat dissipation techniques cannot be implemented or the use of the *LowDataRateOptimize* is not supported, the use of a TCXO provides a more stable clock reference.

15. Known Limitations

This section summarizes the known limitations of the LLCC68 chips, and the related workarounds.

15.1 Modulation Quality with 500kHz LoRa Bandwidth

15.1.1 Description

Some sensitivity degradation may be observed on any LoRa device, when receiving signals transmitted by the LLCC68 with a LoRa BW of 500kHz.

15.1.2 Workaround

Before any packet transmission, bit #2 at address 0x0889 shall be set to:

- 0 if the LoRa BW = 500kHz
- 1 for any other LoRa BW
- 1 for any (G)FSK configuration

The following pseudo-code can be used **before each packet transmission**, to properly configure the chip:

```

1  if (packetType == LORA) {
2      if(loraBandwidth == BW500){
3          value = ReadRegister@0x0889;
4          value = value & 0xFB;
5          WriteRegister(value)@0x0889;
6      }
7  }
8  else {
9      value = ReadRegister@0x0889;
10     value = value | 0x04;
11     WriteRegister(value)@0x0889;
12 }

```

15.2 Better Resistance of the LLCC68 Tx to Antenna Mismatch

15.2.1 Description

The LLCC68 platform embeds a Power Amplifier (PA) clamping mechanism, backing-off the power when over-voltage conditions are detected internally. This method is put in place to protect the internal devices and ensure long-term reliability of the chip. Considering a high-power operation of the LLCC68 (supporting +22dBm on-chip), these “clamping” devices are overly protective, causing the chip to back-down its output power when even a reasonable mismatch is detected at the PA output. The observation is typically 5 to 6dB less output power than the expected.

Note: Using the described workaround improves chip functionality, but is not required to ensure long-term reliability, which is guaranteed with or without workaround.

15.2.2 Workaround

During the chip initialization, the register *TxClampConfig* should be modified to optimize the PA clamping threshold. Bits 4-1 must be set to “1111” (default value “0100”).

This register modification must be done after a Power On Reset, or a wake-up from cold Start.

The following pseudo-code can be used as a reference to implement the fix:

```
1 value = ReadRegister@0x08D8;  
2 value = value | 0x1E;  
3 WriteRegister(value)@0x08D8;
```

15.3 Implicit Header Mode Timeout Behavior

15.3.1 Description

When receiving LoRa packets in Rx mode with Timeout active, and no header (Implicit Mode), the timer responsible for generating the Timeout (based on the RTC timer) is not stopped on *RxDone* event. Therefore, it may trigger an unexpected timeout in any subsequent mode where the RTC isn't re-invoked, and therefore reset and re-programmed.

15.3.2 Workaround

It is advised to add the following commands after ANY Rx with Timeout active sequence, which stop the RTC and clear the timeout event, if any. The register at address 0x0902 is used to stop the counter, while the register at address 0x0944 clears the potential event.

The following pseudo-code can be used as a reference to implement the fix:

```
1 WriteRegister(0x00)@0x0920;
2 value = ReadRegister@0x0944;
3 value = value | 0x02;
4 WriteRegister(value)@0x0944;
```

15.4 Optimizing the Inverted IQ Operation

15.4.1 Description

When exchanging LoRa packets with inverted IQ polarity, some packet losses may be observed for longer packets.

15.4.2 Workaround

Bit 2 at address 0x0736 must be set to:

- "0" when using inverted IQ polarity (see the `SetPacketParam(...)` command)
- "1" when using standard IQ polarity

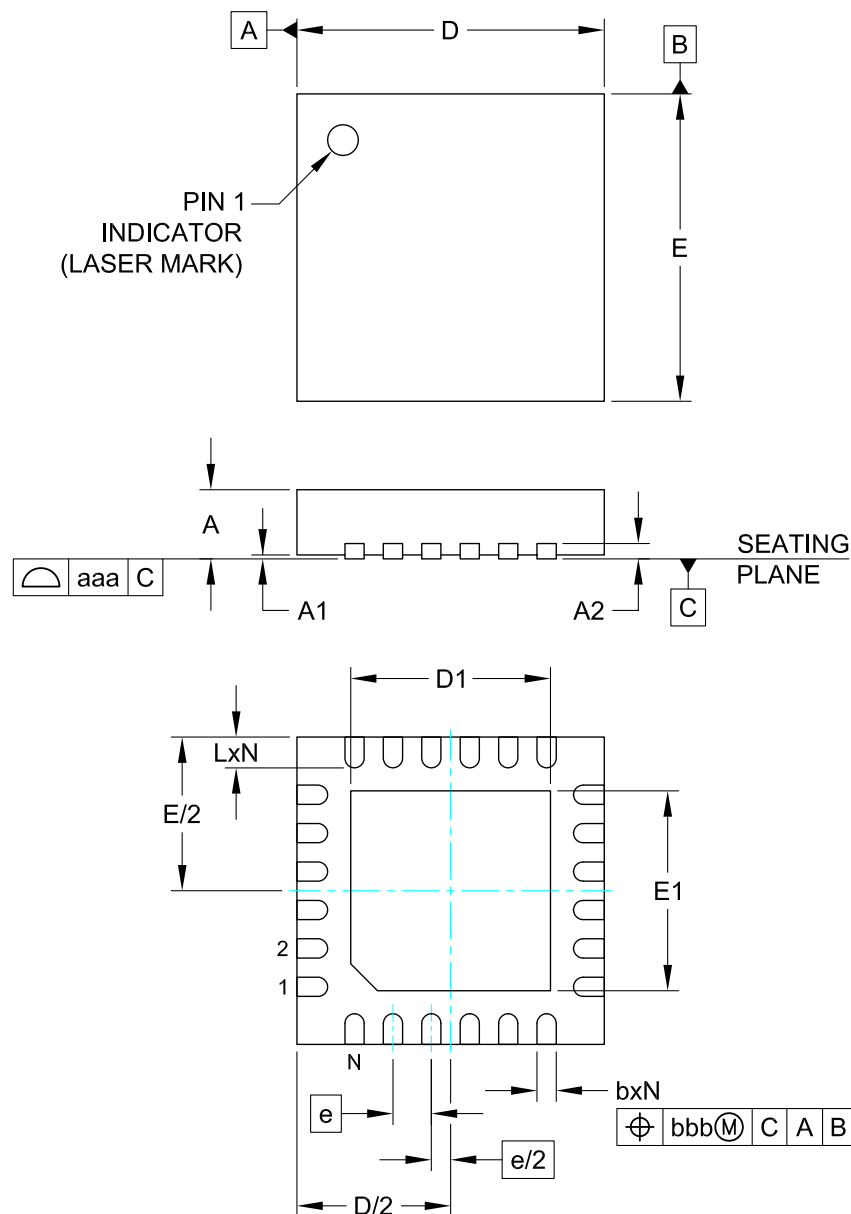
The following pseudo-code can be used as a reference to implement the fix:

```
1 value = ReadRegister@0x0736
2 value = value | 0x04;
3 WriteRegister(value)@0x0736
```


16. Packaging Information

16.1 Package Outline Drawing

The transceiver is delivered in a 4x4mm QFN package with 0.5mm pitch:



DIMENSIONS			
DIM	MILLIMETERS		
	MIN	NOM	MAX
A	0.80	0.90	1.00
A1	0.00	0.02	0.05
A2	(0.20)		
b	0.18	0.23	0.30
D	3.90	4.00	4.10
D1	2.55	2.60	2.65
E	3.90	4.00	4.10
E1	2.55	2.60	2.65
e	0.50 BSC		
L	0.35	0.40	0.45
N	24		
aaa	0.08		
bbb	0.10		

NOTES:

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

Figure 16-1: QFN 4x4mm Package Outline Drawing

16.2 Package Marking

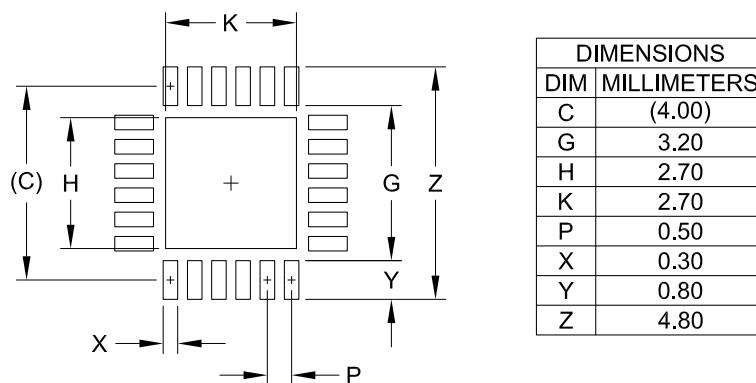


Marking for the 4 x 4 mm MLPQ 24 Package
 nnnnnn = Part Number
 yyww = Date Code
 xxxxx = Semtech Lot No.

Figure 16-2: LLCC68 Marking

16.3 Land Pattern

The recommended land pattern is as follows:



NOTES:

1. CONTROLLING DIMENSIONS ARE IN MILLIMETERS (ANGLES IN DEGREES).
2. THIS LAND PATTERN IS FOR REFERENCE PURPOSE ONLY. CONSULT YOUR MANUFACTURING GROUP TO ENSURE YOUR COMPANY'S MANUFACTURING GUIDELINES ARE MET.
3. 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.
4. SQUARE PACKAGE - DIMENSIONS APPLY IN BOTH " X " AND " Y " DIRECTIONS.

Figure 16-3: QFN 4x4mm Land Pattern Drawing

16.4 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 transceiver uses a QFN24 4x4mm package, also named MLP package.

16.5 Tape and Reel Information

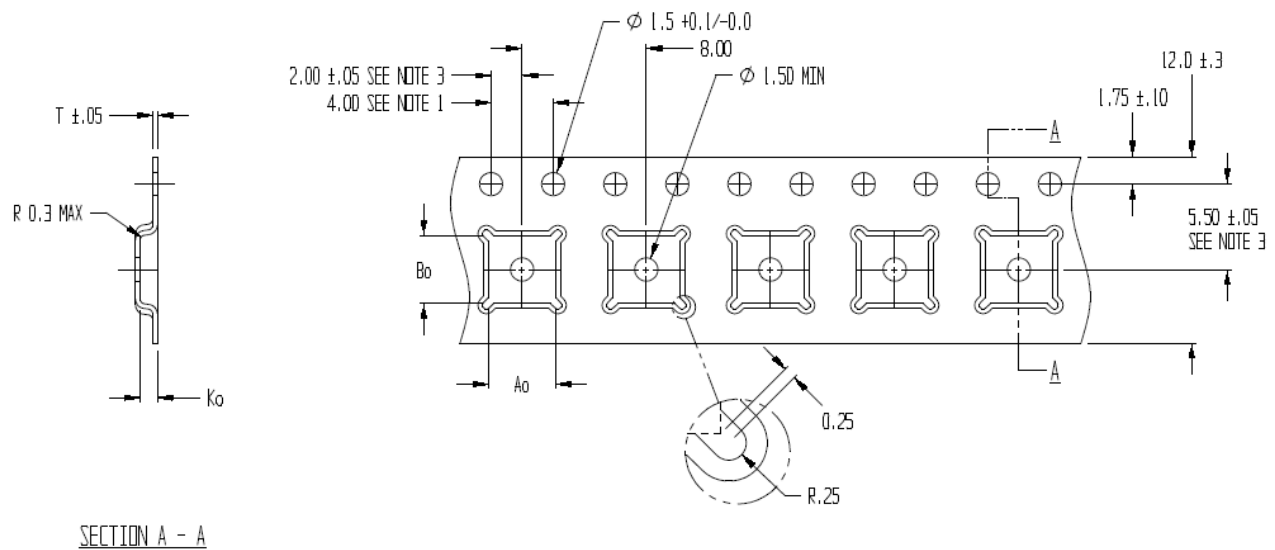


Figure 16-4: Tape and Reel Drawing

Glossary

List of Acronyms and their Meaning (Sheet 1 of 3)

Acronym	Meaning
ACR	Adjacent Channel Rejection
ADC	Analog-to-Digital Converter
API	Application Programming Interface
β	Modulation Index
BER	Bit Error Rate
BR	Bit Rate
BT	Bandwidth-Time bit period product
BW	BandWidth
CAD	Channel Activity Detection
CPOL	Clock Polarity
CPHA	Clock Phase
CR	Coding Rate
CRC	Cyclical Redundancy Check
CW	Continuous Wave
DC-DC	Direct Current to Direct Current Converter
DIO	Digital Input / Output
DSB	Double Side Band
ECO	Engineering Change Order
FDA	Frequency Deviation
FEC	Forward Error Correction
FIFO	First In First Out
FSK	Frequency Shift Keying
GFSK	Gaussian Frequency Shift Keying
GMSK	Gaussian Minimum Shift Keying
GDPW	Gross Die Per Wafer
IF	Intermediate Frequencies
IRQ	Interrupt Request
ISM	Industrial, Scientific and Medical (radio spectrum)

List of Acronyms and their Meaning (Sheet 2 of 3)

Acronym	Meaning
LDO	Low-Dropout
LDRO	Low Data Rate Optimization
LFSR	Linear-Feedback Shift Register
LNA	Low-Noise Amplifier
LO	Local Oscillator
LoRa®	Long Range Communication <i>the LoRa® Mark is a registered trademark of the Semtech Corporation</i>
LR-FHSS	Long Range Frequency Hopping Spread Spectrum
LSB	Least Significant Bit
MISO	Master Input Slave Output
MOSI	Master Output Slave Input
MSB	Most Significant Bit
MSK	Minimum-Shift Keying
NOP	No Operation (0x00)
NRZ	Non-Return-to-Zero
NSS	Slave Select active low
OCP	Over Current Protection
PA	Power Amplifier
PER	Packet Error Rate
PHY	Physical Layer
PID	Product Identification
PLL	Phase-Locked Loop
POR	Power On Reset
RC13M	13MHz Resistance-Capacitance Oscillator
RC64k	64kHz Resistance-Capacitance Oscillator
RFO	Radio Frequency Output
RFU	Reserved for Future Use
RTC	Real-Time Clock
SCK	Serial Clock
SF	Spreading Factor

List of Acronyms and their Meaning (Sheet 3 of 3)

Acronym	Meaning
SN	Sequence Number
SNR	Signal to Noise Ratio
SPI	Serial Peripheral Interface
SSB	Single Side Bandwidth
STDBY	Standby
TCXO	Temperature-Compensated Crystal Oscillator
XOSC	Crystal Oscillator



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 2024