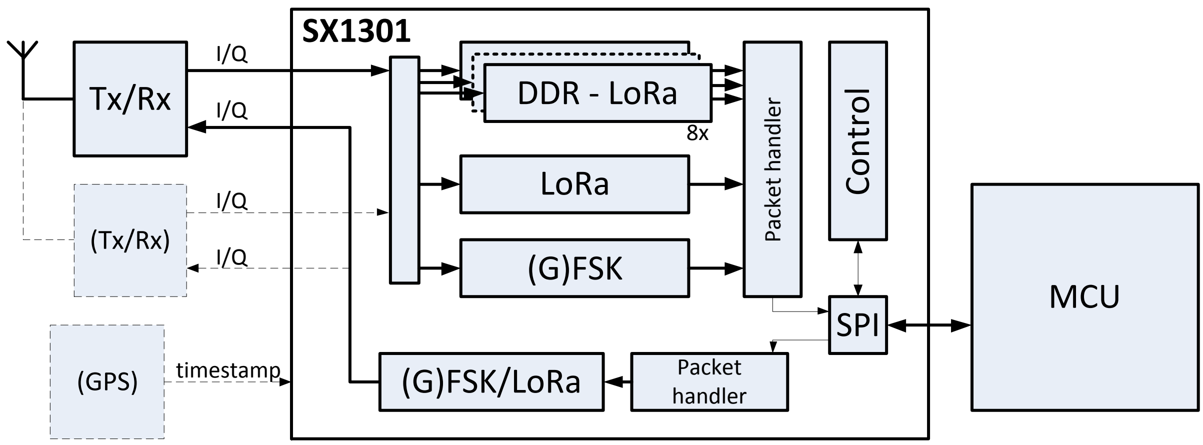


General Description

The SX1301 digital baseband chip is a massive digital signal processing engine specifically designed to offer breakthrough gateway capabilities in the ISM bands worldwide. It integrates the LoRa concentrator IP.

The LoRa concentrator is a multi-channel high performance transmitter/receiver designed to simultaneously receive several LoRa packets using random spreading factors on random channels. Its goal is to enable robust connection between a central wireless data concentrator and a massive amount of wireless end-points spread over a very wide range of distances.

The SX1301 is targeted at smart metering fixed networks and Internet of Things applications with up to 5000 nodes per km2 in moderately interfered environment.

Key product features

- Up to -142.5 dBm sensitivity with SX1257 or SX1255 Tx/Rx front-end
    - -140 dBm with included ref design
- 70 dB CW interferer rejection at 1 MHz offset
- Able to operate with negative SNR
    - CCR up to 9 dB
- Emulates 49x LoRa demodulators and 1x (G)FSK demodulator
- Dual digital Tx & Rx radio front-end interfaces
- 10 programmable parallel demodulation paths
- Dynamic data-rate adaptation (ADR)
- True antenna diversity or simultaneous dual-band operation

Ordering Information

Part Number	Conditioning
SX1301IMLTRC	Tape & Reel 3,000 parts per reel
SX1301IMLT	Trays

Applications

- Smart Metering
- Security Sensors Network
- Agricultural Monitoring
- Internet of Things (IoT)

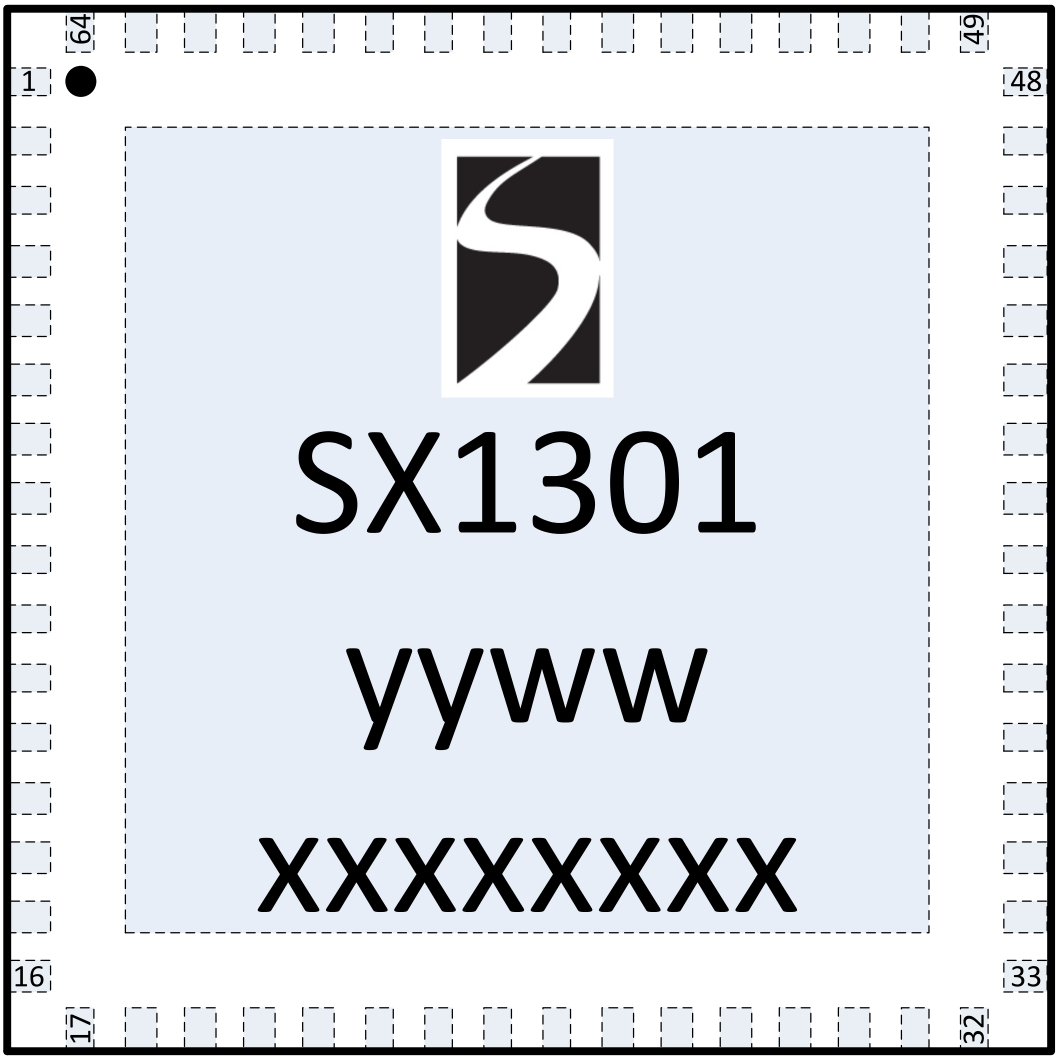
Contents

Figures

Tables

1 Pin Configuration

1.1 Pins placement and circuit marking



Legend: yyww is the date code and xxxxxxxx is the Semtech lot number.

Figure 1 Top view of SX1301 package with 64 pins and exposed ground paddle (bottom of package).

The ground paddle must be connected to ground potential through a large conductive plane that also serves for temperature dissipation.

1.2 Pins description

The table below gives the description of the pins of the circuit.

Pin	Pin Name	Type	Description
0	VSS	Power (GND)	Ground paddle – **must be connected to ground for thermal dissipation**
1	RESET	Input	Global asynchronous reset
2	HOST_SCK	Input	HOST SPI clock (max 10 MHz clock)
3	HOST_MISO	Output	HOST SPI Interface
4	HOST_MOSI	Input	HOST SPI Interface
5	HOST_CSN	Input	HOST SPI Interface
6	SCANMODE	Input	Scanmode signal (tied to 0 in normal mode)
7	VSS	Power (GND)	Ground
8	VCC18	Power (VDD)	Logic core supply
9	GPS_IN	Input	GPS 1 pps input
10	VSS	Power (GND)	Ground
11	VSS	Power (GND)	Ground
12	VCC18	Power (VDD)	Logic core supply
13	RADIO_A_EN	Output	Radio A global enable
14	LNA_A_CTRL	Output	LNA A enable
15	PA_A_CTRL	Output	PA A enable
16	NC		No connected – tie to VSS
17	PA_GAIN[1]	Output	PA gain control of both radio A/B
18	PA_GAIN[0]	Output	PA gain control of both radio A/B
19	RADIO_B_CS	Output	Radio B SPI interface
20	RADIO_B_MOSI	Output	Radio B SPI interface
21	RADIO_B_MISO	Input	Radio B SPI interface
22	RADIO_B_SCK	Output	Radio B SPI interface
23	VCC18	Power (VCC)	Logic core supply
24	VSS	Power (GND)	Ground
25	RADIO_RST	Output	Radio A/B global reset
26	PA_B_CTRL	Output	PA B enable
27	LNA_B_CTRL	Output	LNA B enable
28	RADIO_B_EN	Output	Radio B global enable
29	VCC33	Power (VCC)	Logic IO supply
30	VSS	Power (GND)	Ground
31	VSS	Power (GND)	Ground
32	NC		No connected – tie to VSS
33	NC		No connected – tie to VSS
34	SP_VALID	Input	Radio C sample valid
35	B_IQ_RX	Input	Radio B 1 bit I/Q Rx samples
36	B_QI_RX	Input	Radio B 1 bit Q/I Rx samples
37	B_IQ_TX	Output	Radio B 1 bit I/Q Tx samples
38	B_QI_TX	Output	Radio B 1 bit Q/I Tx samples
39	SP_CLK_OUT	Output	Radio C clock out (32 MHz)

Pin	Pin Name	Type	Description
40	GND	Power (GND)	Ground
41	GND	Power (GND)	Ground
42	VCC18	Power (VCC)	Logic core supply
43	CLK32M	Input	32 MHz clock from radios crystal
44	A_IQ_RX	Input	Radio A 1 bit I/Q Rx samples
45	A_QI_RX	Input	Radio A 1 bit Q/I Rx samples
46	A_IQ_TX	Output	Radio A 1 bit I/Q Tx samples
47	A_QI_TX	Output	Radio A 1 bit Q/I Tx samples
48	NC		No connected – tie to VSS
49	NC		No connected – tie to VSS
50	VSS	Power (GND)	Ground
51	VSS	Power (GND)	Ground
52	VCC33	Power (VCC)	Logic IO supply
53	CLKHS	Input	High speed digital clock
54	GPIO[4]	In/Out	General purpose GPIO[4]
55	GPIO[3]	In/Out	General purpose GPIO[3]
56	GPIO[2]	In/Out	General purpose GPIO[2]
57	GPIO[1]	In/Out	General purpose GPIO[1]
58	GPIO[0]	In/Out	General purpose GPIO[0]
59	VSS	Power (GND)	Ground
60	VCC18	Power (VCC)	Logic core supply
61	RADIO_A_SCK	Output	Radio A SPI interface
62	RADIO_A_MISO	Input	Radio A SPI interface
63	RADIO_A_MOSI	Output	Radio A SPI interface
64	RADIO_A_CS	Output	Radio A SPI interface

Table 1 Pins name and description

2 Electrical Characteristics

2.1 Absolute maximum ratings

Stresses above the values listed below may cause permanent device failure. Exposure to absolute maximum ratings for extended periods may affect device reliability. Operation outside the parameters specified in the Operating Conditions section is not implied.

Parameter	Symbol	Conditions	Value
IO power supply to VSS	$V_{DDIO,ABSMAX}$		-0.5 V to 4.0 V
Core power supply to VSS	$V_{DDCORE,ABSMAX}$		-0.5 V to 2.0 V
Storage temperature	$T_{J,STORE}$		-50 °C to 150 °C
Ambient operating temperature	$T_{J,ABSMAX}$		-40 °C to 125 °C
Pin voltage on IO and Clock pins	$V_{DPIN,ABSMAX}$		-0.3 V to VDDIO + 0.3 V
Peak reflow temperature	T_{PKG}		260 °C
Latchup	I_{LUP}	JESD78D, class I	+/-100 mA
Humidity	H_R		0 – 95 %
ESD	HBM	Human Body Model JESD22-A114 CLASS 2	2 kV
	CDM	Charged Device Model JESD22-C101 CLASS III	300 V

Table 2 Absolute maximum ratings

2.2 Constraints on external

Circuit is expected to be used with the following external conditions.

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Radio ADC samples clock input frequency	XTAL32F	Clock for data communication with Tx†		32		MHz
ADC sample clock frequency tolerance	XTAL32T		-10		+10	ppm
High speed processing clock	HSC_F	Clock for data processing	130	133	150	MHz
Load on IO pins	CLOP		0		25	pF
Notes:						
† The data communication IOs are A_I_RX, A_Q_RX, B_X_RX, B_Q_RX and clock signal is CLK32M						

Table 3 Externals

2.3 Operating conditions

The circuit will operate full specs within the following operating conditions.

Parameter†	Symbol	Conditions	Min	Typ	Max	Unit
Digital IO supply	V_{DDIO}	Operating Conditions for Electrical Specification	3.0		3.6	V
Digital core supply	V_{DDCORE}	Operating Conditions for Electrical Specification	1.75		1.85	V
Operating temperature	T_J	With chip paddle soldered to PCB ground plan with minimum 100 cm2 air exposed area and heat sink	-40		85	°C

Table 4 Operating conditions for electrical specifications

2.4 Electrical specifications

The table below gives the specifications of the circuit within the Operating Conditions as indicated in 2.3 unless otherwise specified.

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Current Consumption						
Current in idle mode	$I_{VDDCORE,IDLE}$	1.8V supply current in Idle mode[1]		120	3000	uA
	$I_{VDDIO,IDLE}$	3.3V supply current in idle mode		1	2	uA
Current in medium active	$I_{VDDCORE,MED}$	1.8V supply current with 4 active paths		330	550	mA
	$I_{VDDIO,MED}$	3.3V supply current with 4 active paths – no load		5	10	mA
Current in full active	$I_{VDDCORE,FULL}$	1.8V supply current with 8 active paths		550	750	mA
	$I_{VDDIO,FULL}$	3.3V supply current with 8 active paths – no load		5	10	mA
IO Pins levels						
Logic low input threshold	VIL	"0" logic input	0.4			V
Logic high input threshold	VIH	"1" logic input			V_{DDIO} – 0.4	V
Logic low output level	VOL	"0" logic output, 2 mA sink	VSS		VSS + 0.4	V
Logic high output level	VOH	"1" logic output, 2 mA source	V_{DDIO} – 0.4		V_{DDIO}	V

Table 5 Electrical specifications

2.5 Timing specifications

The table below gives the specifications of the circuit within the Operating Conditions as indicated in 2.3 unless otherwise specified. See chapters 3.4 and 3.5 for timing diagrams and symbol definitions.

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
SPI						
SCK frequency	F_{SCK}		-	-	10	MHz
SCK high time	t_{ch}		50	-	-	ns
SCK low time	t_{cl}		50	-	-	ns
SCK rise time	t_{rise}		-	5	-	ns
SCK fall time	t_{fall}		-	5	-	ns
MOSI setup time	t_{setup}	From MOSI change to SCK rising edge.	10	-	-	ns
MOSI hold time	t_{hold}	From SCK rising edge to MOSI change.	20	-	-	ns
CSN setup time	t_{nsetup}	From CSN falling edge to SCK rising edge	10	-	-	ns
CSN hold time	t_{nhold}	From SCK falling edge to CSN rising edge, normal mode	40	-	-	ns
NSS high time between SPI accesses	t_{nhigh}		40	-	-	ns
Clock to Rx I-Q data						
Rx IQ hold and setup time	t_{IQ}		2	-	-	ns

Table 6 Timing specifications

[1] Idle current is reached following procedure indicated in application part of datasheet (chapter 3.2.2)

3 Circuit Operation

This chapter is for information only.

3.1 General Presentation

The SX1301 is a smart baseband processor for long range ISM communication. In the receiver part, it receives I and Q digitized bitstream from one or two receivers (SX1257 as an example), demodulates these signals using several demodulators, adapting the demodulators settings to the received signal and stores the received demodulated packets in a FIFO to be retrieved from a MCU. In the transmitter part, the packets are modulated using a programmable (G)FSK/LoRa modulator and sent to one transmitter (SX1257 as an example). Received packets can be time-stamped using a GPS input.

The SX1301 has an internal control block that receives microcode from the MCU. The microcode is provided by Semtech as a binary file to load in the SX1301 at power-on (see Semtech application support for more information).

The control of the SX1301 by the MCU is made using a Hardware Abstraction Layer (HAL). The Hardware Abstraction Layer source code is provided by Semtech and can be adapted by the MCU developers. It is recommended to fully re-use the latest HAL as provided by Semtech on https://github.com/Lora-net/lora_gateway.

3.2 Power-on

3.2.1 Power-up sequence

Power-up sequence must follow the timing indicated in the figure below.

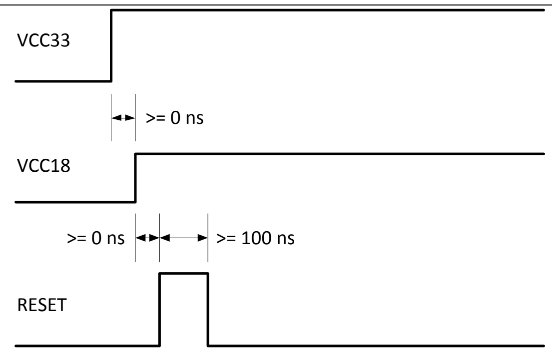


Figure 2 Power-up sequence

3.2.2 Setting the circuit is low-power mode

At power up, the circuit is in a general low-power state but some registers linked to the memory are in undefined state. To set the circuit in low-power mode, the following instructions and clocks must be provided to the circuit.

```
// Setting circuit in low-power mode after power-up
// spi_write(x, y) is a write of data "y" on address "x" on HOST SPI bus
spi_write(0,128);  Reset On
spi_write(0,0);    Reset Off
// provide at least 16 cycles on CLKHS and 16 cycles CLK32M
spi_write(18,1);   BIST 1
// provide at least 4 cycles on CLKHS and 32 cycles CLK32M and 4 cycles on HOST_SCK
spi_write(18,2);   BIST 2
```

```
// provide at least 4 cycles CLK32M and 4 cycles on HOST_SCK
spi_write(0,128);  Reset On
spi_write(0,0);    Reset Off
```

Idle mode sequence after power-up

3.3 Clocking

The SX1301 gateway requires two clocks.

- A 32MHz clock synchronous with the ADC samples. This clock is used to internally sample the ADC samples and clock all the decimation filters. When the SX1301 is used with a Semtech S1257 or SX1255 RF front-end, this clock is provided by the radio. This clock uses CMOS levels (0 – 3.3 V). If a third party radio front-end is used, this must be the clock that also clocks the ADCs and serves as a reference for the radio PLLs.
- A high speed clock whose frequency can be anywhere in the range 130 - 150 MHz. This clock uses CMOS level and must be provided from an external Oscillator. There is no constraint on this clock jitter. This clock is used for most of the demodulation blocks and data processing. This clock is never used by any of the analog/radio blocks.

3.4 SPI Interface

The SPI interface gives access to the configuration register via a synchronous full-duplex protocol. Only the slave side is implemented.

Three access modes to the registers are provided:

- SINGLE access: 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.
- BURST access: the address byte is followed by several data bytes. The address is automatically incremented internally between each data byte. This mode is available for both read and writes accesses. The CSN pin goes low at the beginning of the frame and stay low between each byte. It goes high only after the last byte transfer.
- FIFO access: if the address byte corresponds to the address of the FIFO, then succeeding data byte will address the FIFO. The address is not automatically incremented but is memorized and does not need to be sent between each data byte. The NSS pin goes low at the beginning of the frame and stay low between each byte. It goes high only after the last byte transfer.

The figure below shows a typical SPI single access to a register.

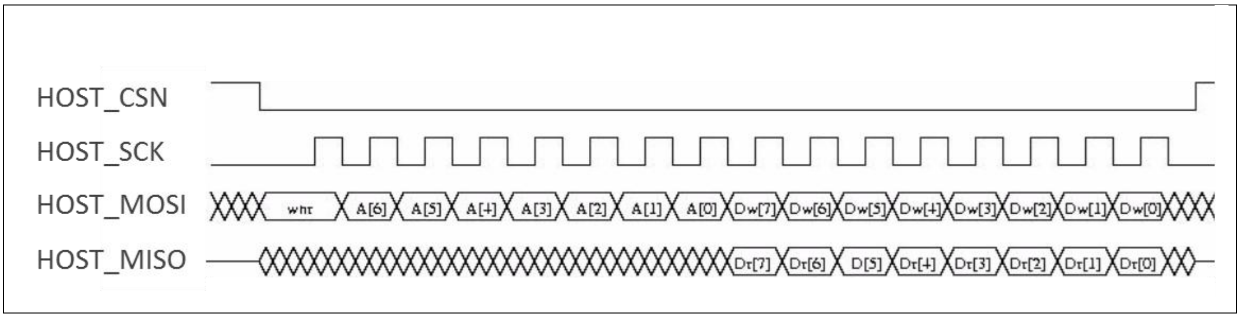


Figure 3 SPI Timing Diagram (single access)

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 CSN pin going low. MISO is high impedance when NSS is high.

The first byte is the address byte. It is comprises:

- one wnr bit, which is “1” for write access and “0” for read access.
- then seven bits of address, MSB first.

The second byte is a data byte, either sent on MOSI by the master in case of a write access or received by the master on MISO in case of read access. The data byte is transmitted MSB first.

Proceeding bytes may be sent on MOSI (for write access) or received on MISO (for read access) without a rising CSN edge and re-sending the address. In FIFO mode, if the address was the FIFO address then the bytes will be written / read at the FIFO address. In Burst mode, if the address was not the FIFO address, then it is automatically incremented for each new byte received.

The frame ends when CSN goes high. The next frame must start with an address byte. The SINGLE access mode is therefore a special case of FIFO / BURST mode with only 1 data byte transferred.

During the write access, the byte transferred from the slave to the master on the MISO line is the value of the written register before the write operation.

3.5 Rx I/Q Interface

The Rx I/Q bit stream has to be generated relative to the radio clock (32 MHz).

The SX1301 can manage I/Q generated on both clock rising and falling edges.

3.5.1 I/Q generated on clock rising edge

To relax the constraint on setup and hold time, it is recommended to use the falling edge of the clock.

To avoid internal setup and hold violation, it is mandatory to avoid I/Q change in a range of +/- 2 ns around clock falling edge

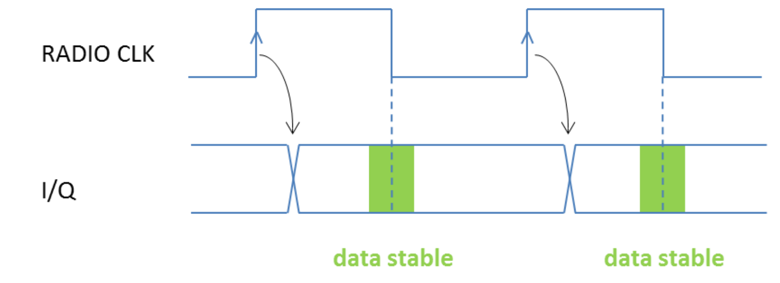


Figure 4 I/Q on clock rising edge

3.5.2 I/Q generated on clock falling edge

To relax the constraint on setup and hold time, it is recommended to use the rising edge of the clock

To avoid internal setup and hold violation, it is mandatory to avoid I/Q change in a range of +/- 2 ns around clock rising edge

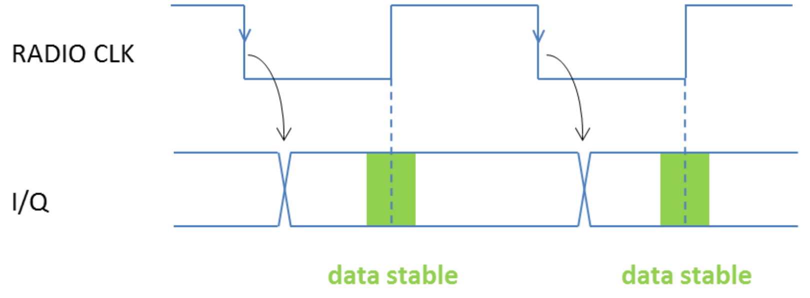


Figure 5 I/Q on clock falling edge

3.6 GPIO mapping

There are 5 general purposes I/O which can be separately configured as inputs or outputs.

The 2 registers GPIO_SELECT_OUTPUT and GPIO_SELECT_INPUT are used to define the GPIO mappings.

The 5 bits register GPIO_MODE defines the direction (input or output) for each GPIO (1 means output and 0 means input).

3.6.1 GPIO output configuration

The GPIOs configured as outputs can be driven by various internal signals. The following table gives the possible combinations selected by the GPIO_SELECT_OUTPUT register.

select	gpio[4]	gpio[3]	gpio[2]	gpio[1]	gpio[0]
0	tx_on	fsk_pkt	bh_pkt	sensor_pkt	rx_buffer_not_empty
1	1'b0	1'b0	1'b0	clk_160_div	clk_32_div
2	dbg_mcu_agc_to_gpio				
3	dbg_mcu_arb_to_gpio				
4	1'b0	1'b0	sensor_0_header_val	sensor_0_sync	sensor_0_detect
5	1'b0	1'b0	bh_header_val	bh_sync	bh_detect
6	1'b0	1'b0	fsk_header_val	fsk_sync	fsk_detect
7	radio_a_i	radio_a_q	radio_b_i	radio_b_q	1'b0
8	host_reg_to_gpio				
9	1'b0	1'b0	1'b0	bist_1_finished	bist_0_finished
10	tx_on	fsk/bh_pkt	gps_hpps	sensor_pkt	rx_buffer_not_empty

Table 7 GPIO output configuration

3.6.2 GPIO input configuration

GPIOs configured as inputs can be connected to various internal block ports. The following table gives the possible combinations selected by the GPIO_SELECT_INPUT register. NC means not connected.

SELECT	GPIO[4]	GPIO[3]	GPIO[2]	GPIO[1]	GPIO[0]
0	NC	NC	NC	NC	NC
1	NC	NC	NC	NC	NC
2	DBG_MCU_AGC_FROM_GPIO, connects GPIOs to an input port of the AGC MCU				
3	DBG_MCU_ARB_FROM_GPIO, connects GPIOs to an input port of the ARB MCU				
4	NC	NC	NC	NC	NC
5	NC	NC	NC	NC	NC
6	NC	NC	NC	NC	NC
7	NC	NC	NC	NC	NC
8	Reserved				
9	NC	NC	NC	NC	NC
10	NC	NC	NC	NC	NC

Table 8 GPIO input configuration

3.7 RX mode block diagram, reception paths characteristics

3.7.1 Block diagram

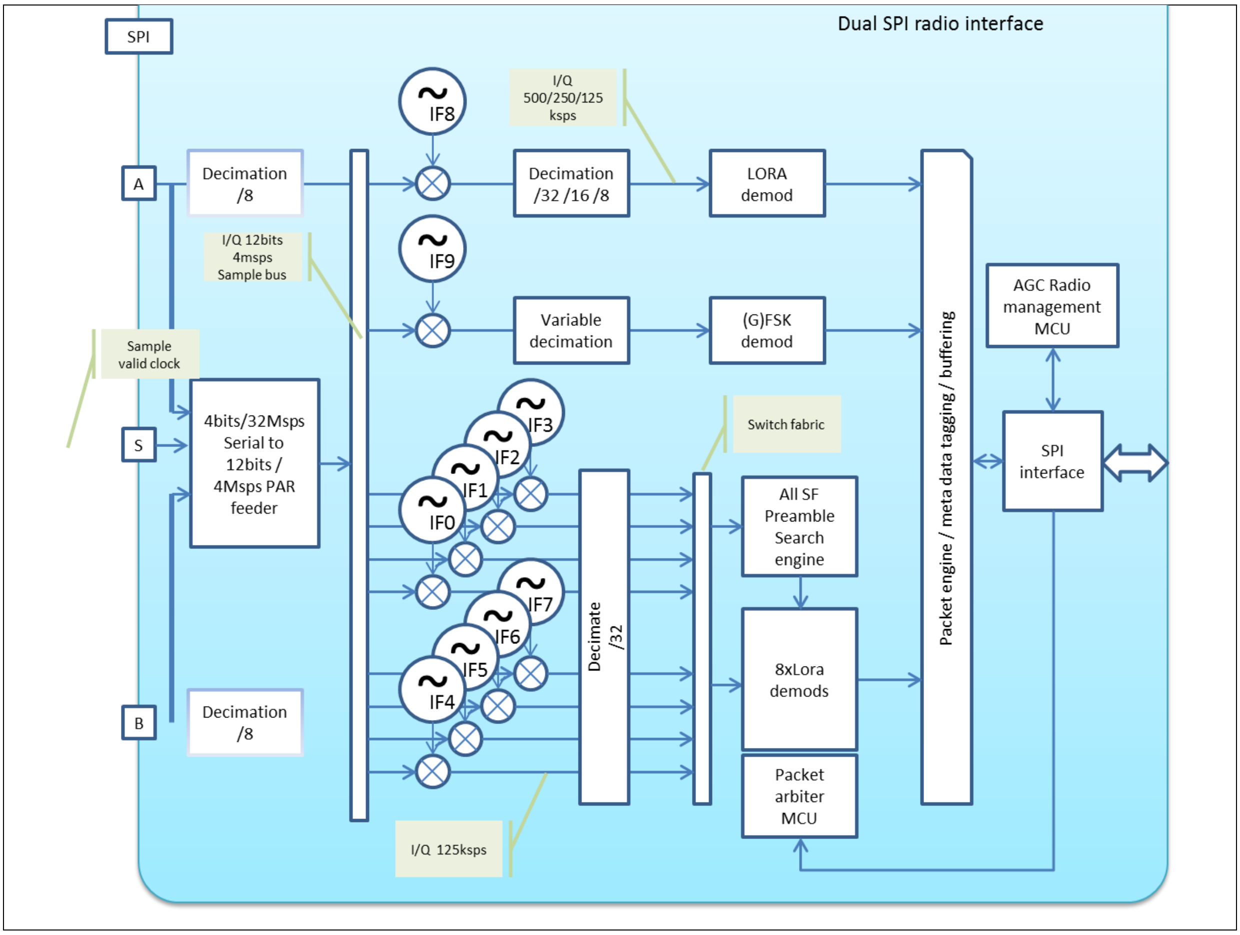


Figure 6 SX1301 digital baseband chip block diagram

All chip functionalities can be accessed through a single high speed SPI interface.

The chip integrates two dedicated micro-controllers.

1. A radio AGC MCU. Handling the real time automatic gain control of the entire chain. For this purpose this MCU can control the two radio front-ends through a dedicated SPI master interface. This MCU also handles radio calibration and RX<->TX radio switch
2. A packet arbiter MCU. Assigning the available LoRa modems to the various reception paths. This arbiter can be configured to follow different priority rules based on parameters like data rate of the incoming packet, channel, radio path or signal strength of the incoming packet.

The firmware of those 2 MCUs can be fully programmed at any time through the HOST SPI interface. This firmware is embedded in the Hardware Abstraction Layer provided by Semtech and does not need to be developed by the user.

3.7.2 Reception paths characteristics

The SX1301 digital baseband chip contains 10 programmable reception paths. Those paths have differentiated levels of programmability and allow different use cases. It is important to understand the differences between those demodulation paths to make the best possible use from the system.

IF8 LoRa channel

This channel can be connected to Radio A or B using any arbitrary intermediate frequency within the allowed range. This channel is LoRa only. The demodulation bandwidth can be configured to be 125, 250 or 500 kHz. The data rate can be configured to any of the LoRa available data rates (SF7 to SF12) but, as opposed to IF0 to 7, ONLY the configured data rate will be demodulated. This channel is intended to serve as a high speed backhaul link to other gateways or infrastructure equipment. This demodulation path is compatible with the signal transmitted by the SX1272 & SX1276 chip family. Chapter 3.13 gives a brief overview of the expected system sensitivity in LoRa mode

IF9 (G)FSK channel

Same as previous except that this channel is connected to a GFSK demodulator. The channel bandwidth and bitrate can be adjusted. This demodulator offers a very high level of configurability, going well beyond the scope of this document. The demodulator characteristics are essentially the same than the GFSK demodulator implemented on the SX1232 and SX1272 Semtech chips.

This demodulation path can demodulate any legacy FSK or GFSK formatted signal.

IF0 to IF7 LoRa channels

Those channels can be connected individually to Radio A or B. The channel bandwidth is 125 kHz and cannot be modified or configured. Each channel IF frequency can be individually configured. On each of those channels any data rate can be received without prior configuration. Several packet using different data rates may be demodulated simultaneously even on the same channel. Those channels are intended to be used for a massive asynchronous star network of 10000's of sensor nodes. Each sensor may use a random channel (amongst IF0 to 7) and a different data rate for any transmission.

Typically sensor located near the gateway will use the highest possible data rate in the fixed 125 kHz channel bandwidth (e.g. 6 kbit/s) while sensors located far away will use a lower data rate down to 300 bit/s (minimum LoRa data rate in a 125 kHz channel).

The SX1301 digital baseband chip scans the 8 channels (IF0 to IF7) for preambles of all data rates at all times. The chip is able to demodulate simultaneously up to 8 packets. Any combination of up to 8 packets is possible (e.g. one SF7 packet on IF0, one SF12 packet on IF7 and one SF9 packet on IF1 simultaneously).

The SX1301 can detect simultaneously preambles corresponding to all data rates on all IF0 to IF7 channels. However it cannot demodulate more than 8 packets simultaneously. This is because the SX1301 architecture separates the preamble detection and acquisition task from the demodulation process. The number of simultaneous demodulation (in this case 8) is an arbitrary system parameter and may be set to any value for a customer specific circuit.

The unique multi data-rate multi-channel demodulation capacity of channels 0 to 7 allow innovative network architecture to be implemented:

- End-point nodes can change frequency with each transmission in a random pattern. This provides vast improvement of the system in term of interferer robustness and radio channel diversity
- End-point nodes can dynamically perform link rate adaptation based on their link margin without adding to the protocol complexity. There is no need to maintain a table of which end point uses which data rate, because all data rates are demodulated in parallel.
- True antenna diversity can be achieved on the gateway side. Allows better performance for mobile nodes in difficult multi-path environments.

3.8 Packet engine and data buffers

3.8.1 Receiver Packet engine

Each time any of the demodulators decodes a packet, it is tagged with some additional information and stored in a shared data buffer (the data buffer size is 4096 bytes). For this purpose a specific data buffer management block reserves a segment with the necessary length in the data buffer and at the same time, stores the start address and the length of the packet field in a small FIFO type structure (named the access FIFO). The FIFO can contain up to 16 (start_addr, length) pairs.

A status register contains at any moment the number of packets currently stored in the data buffer (and in the access FIFO).

To retrieve a packet, the host micro-controller first advances 1 step in the access FIFO by writing 1 to the 'next' bit. Then reads the (start_addr, length) information. The host micro-controller can now retrieve in one SPI burst operation the entire packet and associated meta-data by reading 'length'+16 bytes starting at address 'start_addr' in the data buffer .. To do so, first position the HOST address pointer to 'start-addr', then read 'length' + 16 bytes from the 'packet_data' register . At the end of each byte the HOST address pointer is automatically incremented.

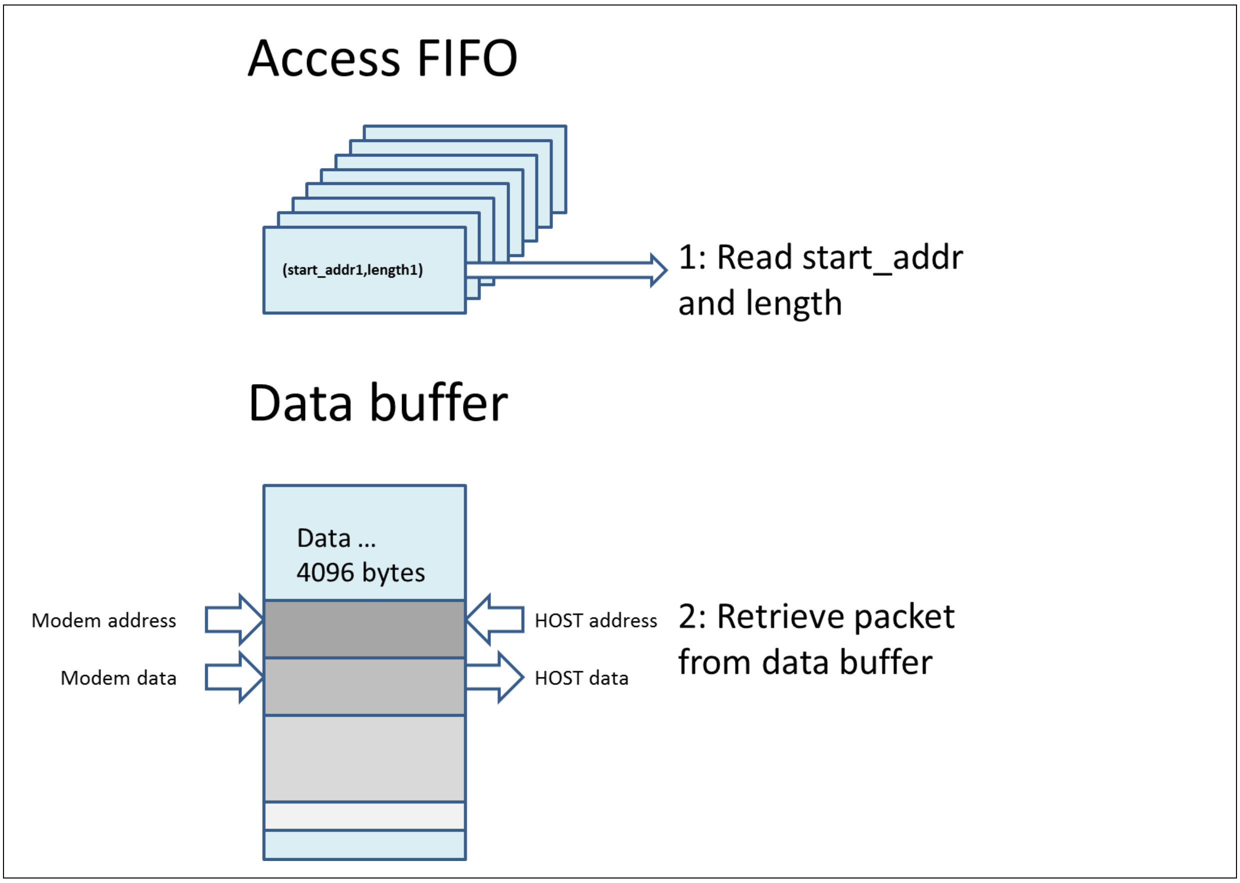


Figure 7 Access FIFO and data buffer

The packet data is organized as follows:

Packet buffer data organization		
Offset from start pointer	**Data stored**	**Comment**
0	PAYLOAD	PAYLOAD DATA
…		
…		
…		
…		
payload_size-1		
payload_size	CHANNEL	1 to 10 as described by block diagram
1+payload_size	SF[3:0],CR[2:0],CRC_EN	
2+payload_size	SNR AVERAGE	averaged SNR in dB on the packet length
3+payload_size	SNR MIN	minimum SNR (dB) recorded during packet length
4+payload_size	SNR MAX	maximum SNR recorded during packet length
5+payload_size	RSSI	channel signal strength in dB averaged during packet
6+payload_size	TIMESTAMP[7:0]	32 bits time stamp , 1 us step

7+payload_size	TIMESTAMP[15:8]	
8+payload_size	TIMESTAMP[23:16]	
9+payload_size	TIMESTAMP[31:24]	
10+payload_size	CRC_VALUE[7:0]	value of the computed CRC16
11+payload_size	CRC_VALUE[15:8]	
12+payload_size	MODEM ID	
13+payload_size	RX_MAX_BIN_POS[7:0]	Correlation peak position
14+payload_size	RX_MAX_BIN_POS[15:8]	
15+payload_size	RX_CORR_SNR	Detection correlation SNR
16+payload_size	Reserved	
17+payload_size	Reserved	

Table 9 Packet data fields

This means that the host micro-processor has to read 16 additional bytes on top of each packet to have access to all the meta-data. If the host is only interested in the payload itself + the channel and the data rate used , then payload + 2bytes is enough.

3.8.2 Transmitter packet engine

The SX1301 gateway transmitter can be used to send packets. The following parameters can be dynamically programmed with each packet:

- Radio channel
- FSK or LoRa modulation
- Bandwidth ,data rate, coding rate (in LoRa mode) , bit rate and Fdev (in FSK mode)
- RF output power
- Radio path (A or B)
- Time of departure (immediate or differed based on the gateway hardware clock with 1us accuracy)

All those dynamic parameter fields are sent alongside the payload in the same data buffer.

The data buffer can only hold a single packet at a time (next packet to be sent). The scheduling and ordering task is let to the host micro-processor.

The host micro-processor can program the exact time of departure of each packet relative to the gateway hardware clock. The same clock is used to tag each packet received with a 32bits timestamp. The same 32bits time stamp principle is used in TX mode to indicate when to transmit exactly. This removes the real time constraint from the host micro-processor and allows very precise protocol timing.(For example, if the protocol running on the end point expects and acknowledge exactly one sec after the end of each packet of its uplink). The host micro-processor pulls the uplink packet from the RX packet engine , realizes that it must send an acknowledge, takes the uplink packet time stamp, simply increments it by 1 sec and uses that value to program the time of departure of the acknowledge packet. Exactly one second (+/- 1us) after the uplink packet was received, the gateway will transmit the desired acknowledge packet. This allows very tight reception interval windows on the battery powered end points hence improved battery life.

The packet structure for transmission is as follow:

Byte	Subfield	Description	comment
0	23:16	Channel frequency	Fchan/32MHz*2^19
1	15:8		
2	7:0		
3	31:24	Start time	Value of the timer at which the modem has to start (in us)
4	23:16		
5	15:8		
6	7:0		
7	7:6	Reserved	
	5:5	Radio select	Select radio A (0) or B (1)
	4:4	Modulation type	0:LoRa, 1:FSK
	3:0	Tx power	>7: 20dBm, otherwise 14dBm
8		Reserved	
LoRa:			
9	7:7	Payload CRC16 enable	Enables CRC16
	6:4	Coding rate	Coding rate = 4/(4+CR)
	3:0	SF	6 to 12
10	7:0	Payload length	number of bytes
11	7:3	Reserved	
	2:2	Implicit header enable	
	1:0	Modulation bandwidth	2:500, 1:250, 0:125 kHz
12	15:8	Preamble symbol number	Number of symbols in the preamble
13	7:0		
14		Reserved	
15		Reserved	
FSK:			
9	7:0	FSK frequency deviation	Frequency deviation in KHz
10	7:0	Payload length	number of bytes
11	0	Packet Mode	0 -> fixed length 1 -> variable length
	1	CRC enable	0 -> No CRC 1 -> CRC
	3:2	Dcfree Enc	00 -> DC free encoding off 01 -> Manchester encoding 10 -> Whitening Encoding 11 -> reserved
	4	Crc IBM	0 -> CCITT CRC 1 -> IBM CRC
12	15:8	FSK Preamble Size	The number of preamble bytes sent over the air before the sync pattern.
13	7:0	FSK Preamble Size	
14	15:8	FSK bit rate	Bit rate = 32e6/(FSK bit rate)
15	7:0	FSK bit rate	

16		Payload first byte	up to 128 bytes

Table 10 Packet structure for transmission

For words of more than 1 byte, MSBs are sent first.

Bytes 9 to 15 vary depending whether the FSK or the LoRa TX modem is being used.

The user payload starts at byte 16. This is the first byte that will be received by the end point. Bytes 0 to 15 are not transmitted and are just used to dynamically configure the gateway prior to emission.

3.9 Receiver IF frequencies configuration

Each IF path intermediate frequency can be programmed independently from -2 to +2 MHz. the following sections give a few programming example for various use cases.

3.9.1 Configuration using 2 x SX1257 radios

The SX1257 RX PLLs can be configured to any frequency inside the 868/900 MHz ISM band with a 61 Hz step. The SX1257 streams I/Q samples through a 2 wire digital interface. The bits stream corresponds directly to the I/Q sigma delta ADCs outputs sampled at 32 MSps. This delta sigma stream must be low-passed and decimated to recover the available 80dB dynamic of the ADCs. After decimation the usable spectrum bandwidth is ±400 kHz centered on the RX PLL carrier frequency.

The following plot gives the spectral power content of the I/Q bit stream.

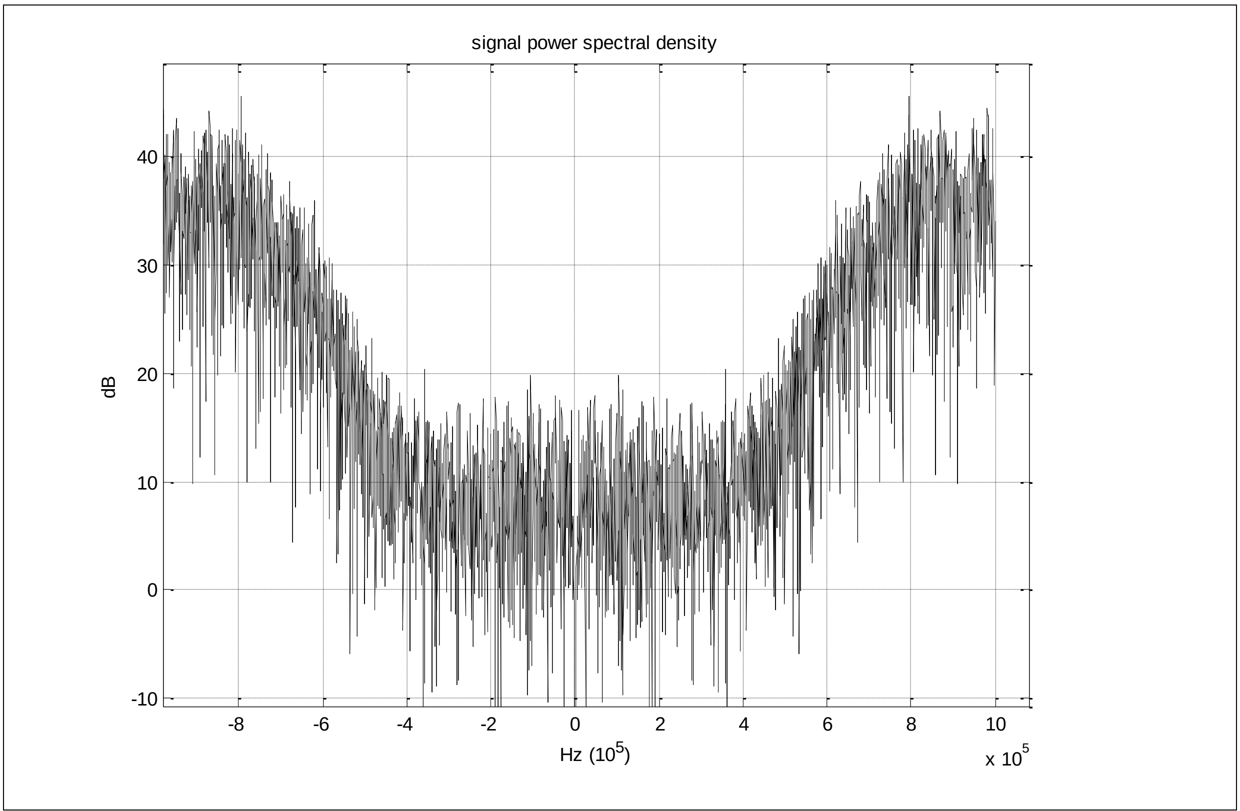


Figure 8 SX1255/57 digital I/Q power spectral density

The quantization noise raises sharply outside the -400 to +400 kHz range. For more details on the SX1257/55 radio specifications please consult the specific product datasheet.

The following plot represents a possible use case where

- Radio A PLL is set to 867.0 MHz
- Radio B PLL is set to 868.4 MHz
- The system uses 8 separate 125 kHz LoRa channels for star connection to sensors
- One high speed 250 kHz LoRa channel for connection to a relay
- One high speed 200 kHz GFSK channel for meshing

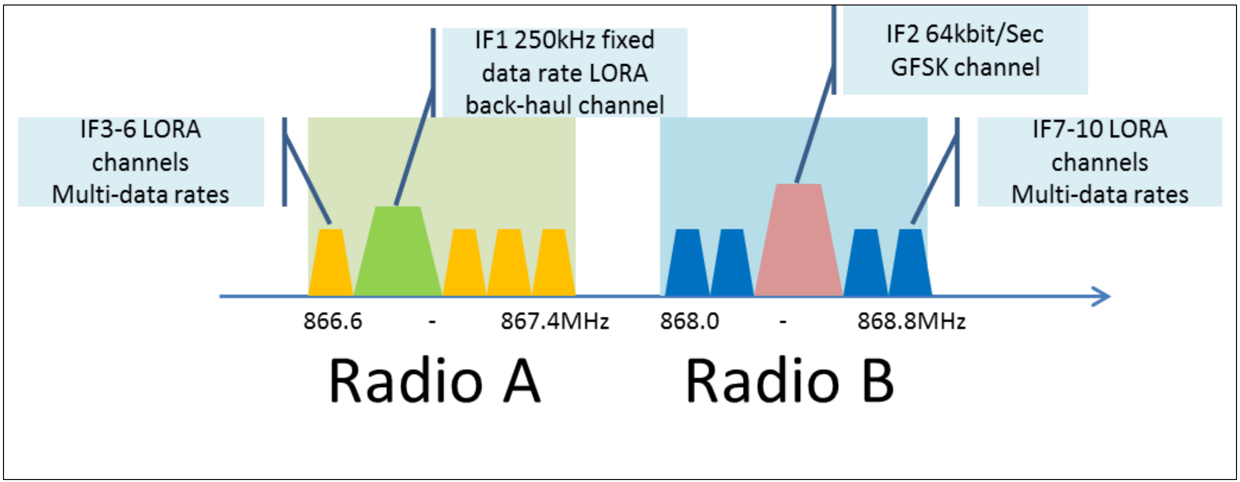


Figure 9 Radio spectrum

In the previous example the various IF frequencies would be set as follow:

IF8	A: -125kHz	Lora backhaul , fixed data-rate
IF9	B: 0kHz	GFSK backhaul
IF0	A: -312.5kHz	LoRa multi-data rate channel
IF1	A: 62.5kHz	"
IF2	A: 187.5kHz	"
IF3	A: 312.5kHz	"
IF4	B: -312.5kHz	"
IF5	B: -187.5kHz	"
IF6	B: 187.5kHz	"
IF7	B: 312.5kHz	"

Table 11 IF frequencies set

If for example, 8 contiguous 125 kHz LoRa channels are desired the following configuration may be used:

- Radio A PLL is set to 867 MHz
- Radio B PLL is set to 876.5 MHz

The two radio baseband spectrum overlap a little bit.

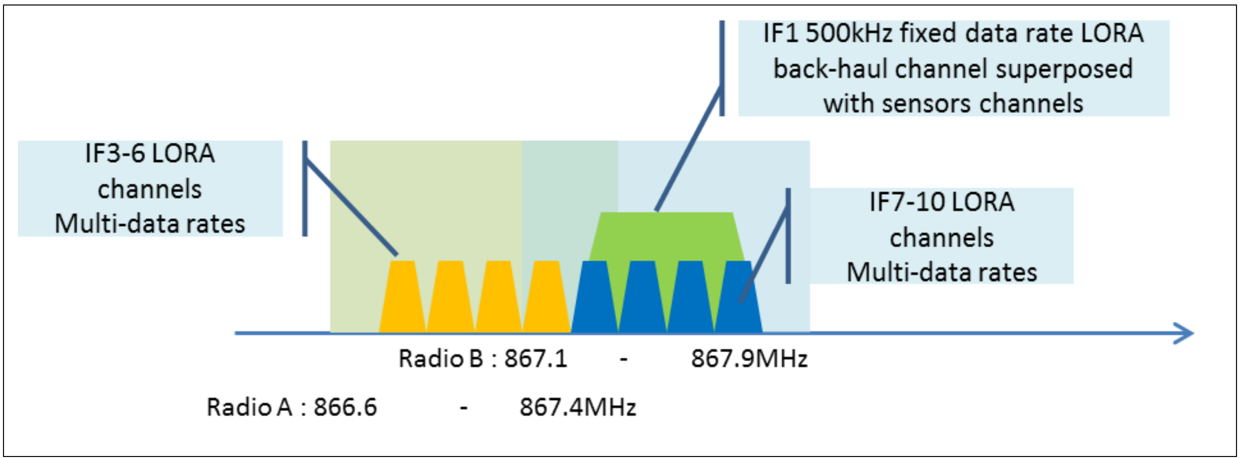


Figure 10 Radio spectrum

The following IF frequencies are used:

IF8	A: 0 kHz	Lora backhaul , fixed data-rate
IF9	Not used	GFSK backhaul
IF0	B: -187.5 kHz	LoRa multi-data rate channel
IF1	B: -62.5 kHz	"
IF2	B: 62.5 kHz	"
IF3	B: 187.5 kHz	"
IF4	A: -187.5 kHz	"
IF5	A: -62.5 kHz	"
IF6	A: 62.5 kHz	"
IF7	A: 187.5 kHz	"

Table 12 IF frequency used

Note : As shown in this example the 500 or 250 kHz IF1 LoRa channel may overlap with the multi-data rate IF3 to 10 channels. Transmissions happening in the IF7 to 10 channels will be noise like for the IF1 LoRa demodulator and reciprocally. It is however better from a performance point of view to separate as much as possible different channels mainly when the associated signal powers are very different (like between a backhaul link which usually enjoys line-of-sight attenuation and sensor link with very low signal levels).

3.9.2 Two SX1255 : 433 MHz band

The circuit will behave exactly as described in the previous section except that everything can be transposed in the 433 MHz ISM band using SX1255 front-end radios instead of SX1257.

3.9.3 One SX1257 and one SX1255

In that case dual band simultaneous reception is possible. The following configuration is a typical example of the possible system configuration.

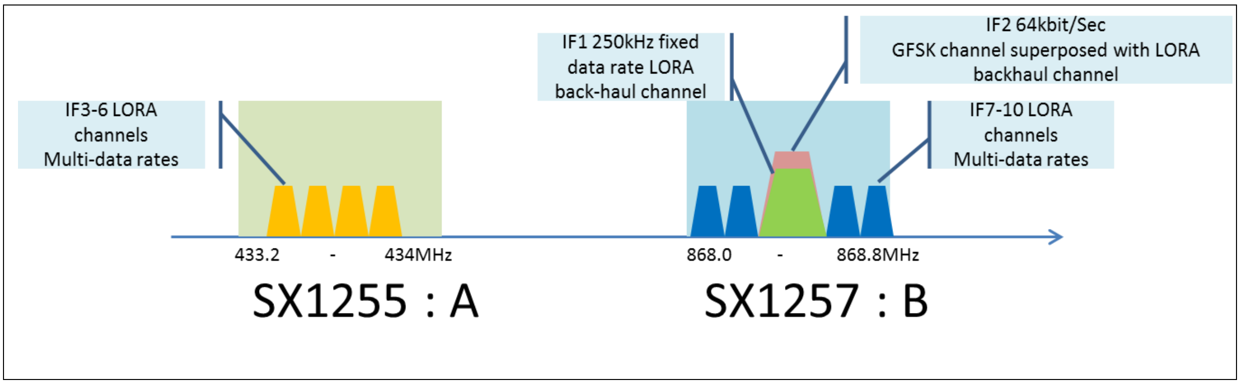


Figure 11 Radio spectrum

- Radio A is an SX1255 configured on 433.6 MHz
- Radio B is a SX1257 configured on 866.4 MHz
- 4 multi data-rates 125 kHz LoRa channel in the low –band
- 4 multi data-rates 125 kHz LoRa channel in the high –band
- One 250 kHz LoRa fixed data-rate channel superposed with a 200 kHz GFSK channel in the high band

As can be seen the system is extremely flexible and allows any arbitrary set of channel configuration.

3.10 Connection to RF front-end

3.10.1 Connection to Semtech SX1255 or SX1257 components

The SX1301 digital baseband chip is designed to be preferably interfaced with either:

1. 2x SX1257 radio front-ends for the 868 MHz band with antenna diversity support
2. 2x SX1255 radio front-ends for the 433 MHz band with antenna diversity support
3. 1x SX1257 & 1x SX1255 , enabling simultaneous dual-band operation

All modems Intermediate Frequencies may be adjusted independently within the allowed radio baseband bandwidth, e.g. ±400 kHz. Optimized firmware is provided to optimally setup the SX1257 / 1255 radios and perform real time automatic gain control.

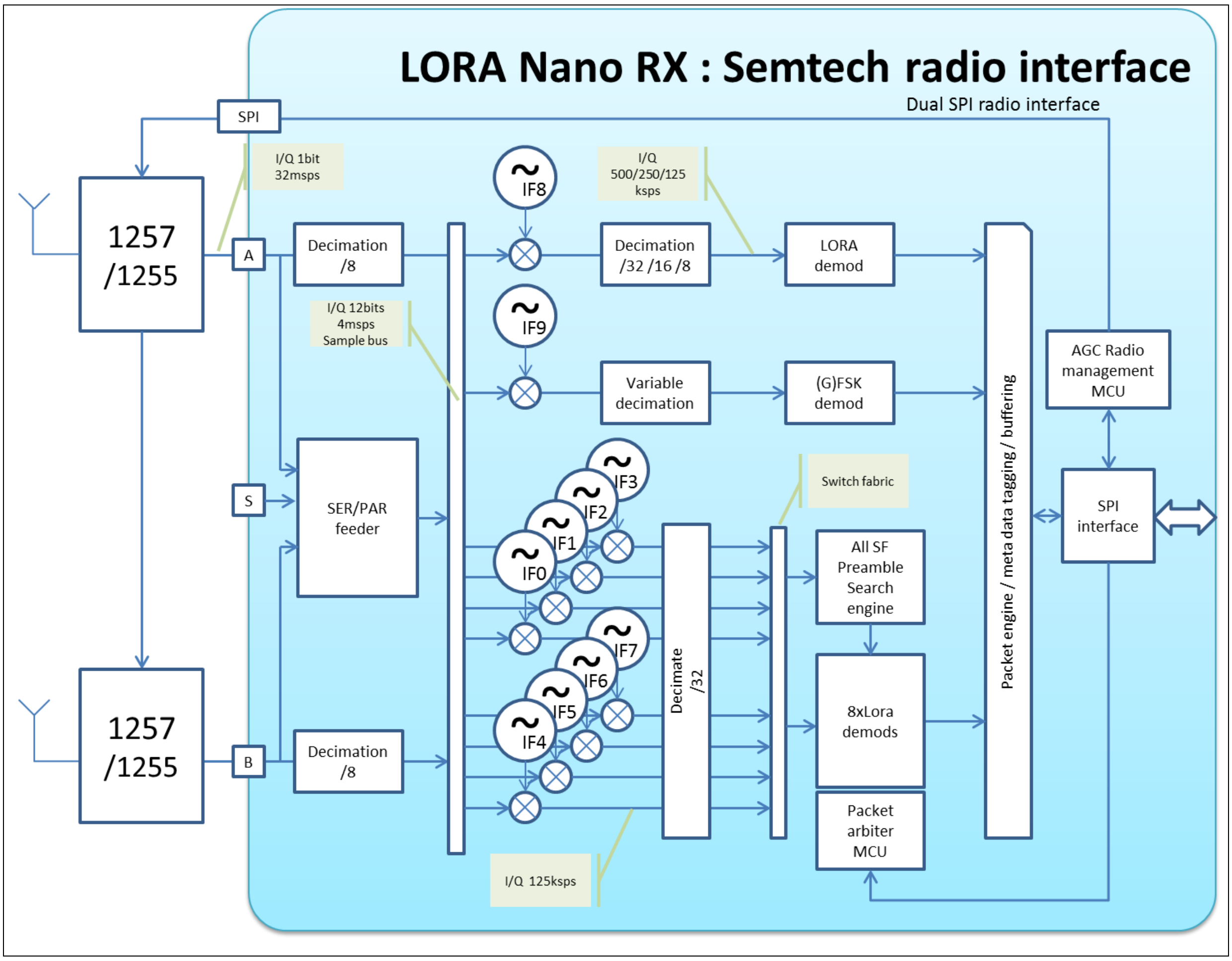


Figure 12 Dual band operation

3.10.2 SX1301 RX operation using a third party RF front-end

In that case a third party RF front-end may be used. The digitized I/Q stream must be adapted to the specific format required by the SX1301 digital baseband using an FPGA/CPLD or any other suitable programmable component.

In that mode the SX1301 expects a stream of 4 bits samples at a 32 MSps rate. The “Sample valid” input should pulse every 8 clock cycles to delimit packets of 8 samples. From those 8 samples representing 32 bits, the first 24 MSB are kept as I/Q 12bits sample information and fed to the internal sample 4 MSps sample bus.

All modems Intermediate Frequencies may be adjusted independently within the allowed radio baseband bandwidth up to 2 MHz (third party radio and FPGA/CPLD digital filtering dependent)

The 32MHz clock input is not represented for the sake of clarity.

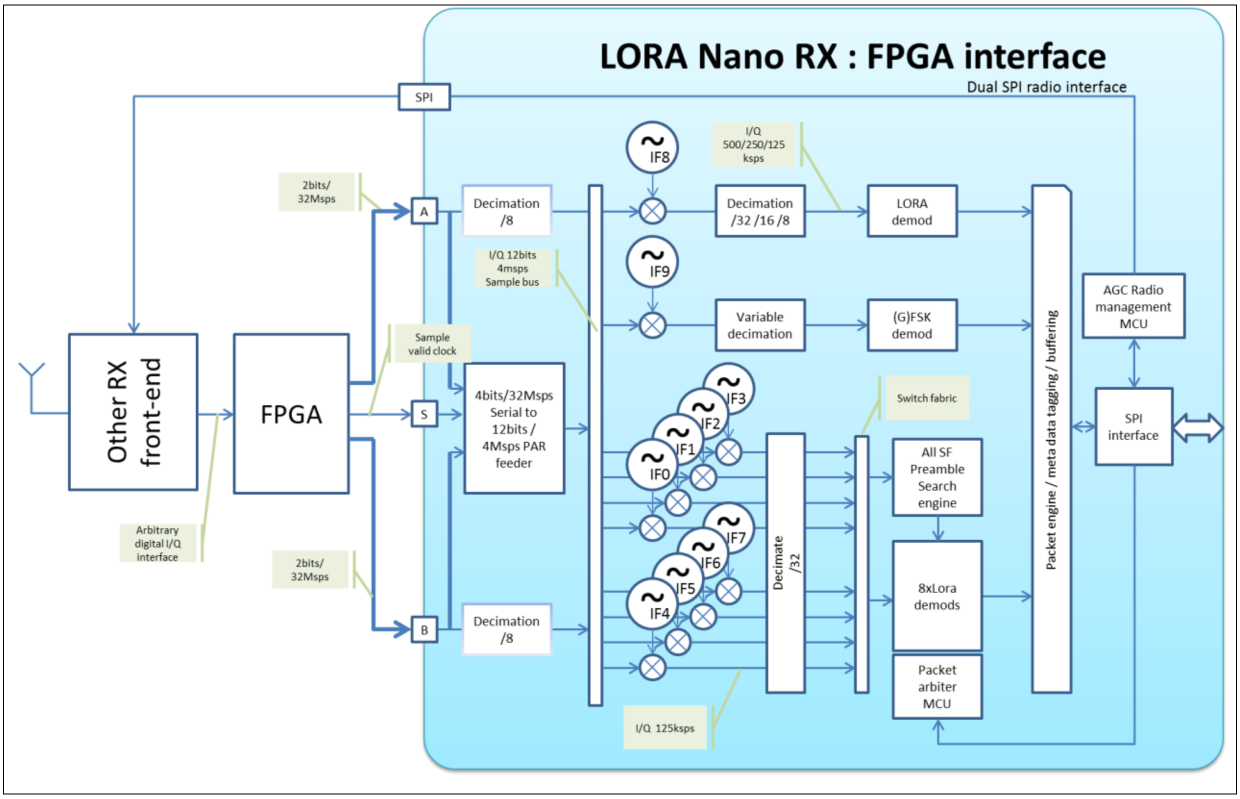


Figure 13 SX1301 with third party frontend

The digital interface to third party radio works as follow:

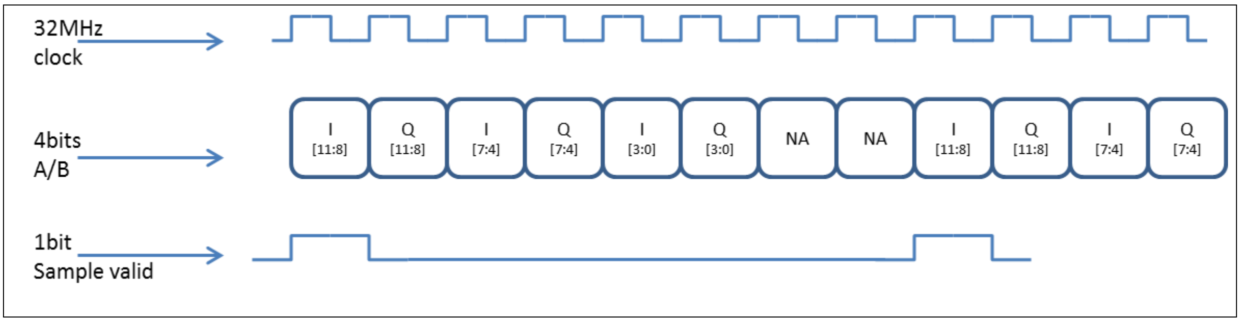


Figure 14 Digital interface for third party radio

The RF front-end must provide a 32 MHz clock. "Sample valid" and data bits must change state on the rising edge of the clock. They are sampled internally in the SX1301 digital IC on the falling edge of the 32 MHz clock.

The "sample valid" signal signals the start of a new I/Q sample. The I/Q 'bits chunks are time interleaved.

When the SX1301 digital baseband chip is connected to a third party radio front-end, the firmware running on the AGC MCU can be changed to perform dynamic gain adaptation of the external radio chip through an SPI interface. The radio SPI interface must fulfill the following conditions:

1. 7 bits address width and 1 W/R bit
2. 8 bits data width

The "Chip select" signal polarity is programmable.

3.10.3 Radio calibration

All calibrations required are performed by uploading the calibration firmware to the integrated radio controller MCU. This specific firmware runs entirely on the SX1301 gateway without intervention of the host micro-processor and performs the following calibrations on both radio channels:

- Carrier leakage cancellation in TX mode
- IQ gain (better than 0.1 dB) and phase imbalance (better than 1 deg) in RX mode

All corrections are applied digitally inside the SX1301 gateway at the appropriate place in the TX & RX processing chains.

During the duration of the calibration (500 ms), no RX or TX operation is possible.

3.10.4 SX1301 connection to RF front-end for TX operation

In TX mode , the SX1301 digital baseband must be connected either to:

1. At least one SX1255 or SX1257
2. Any combination of both radios

Third party radios are not supported for TX operations. Any LoRa or (G)FSK packet may be transmitted on any of the two radios. Only a single packet may be transmitted at any given time. Transmit operation interrupts all current reception operations.

The digital radio interfaces are separated between RX & TX, therefore the SX1301 may accommodate a third party radio front-end for RX operations and any combination of SX1255/57 for TX operation without problem.

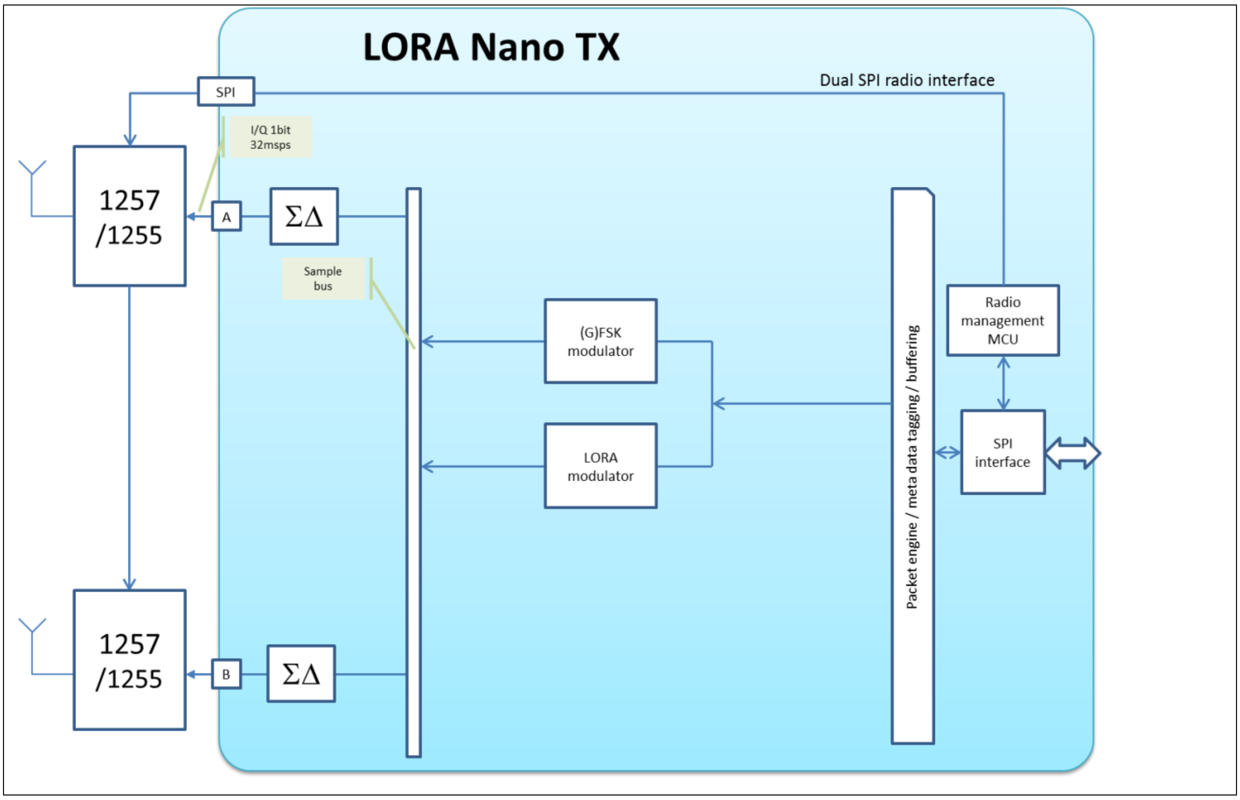


Figure 15 Transmission schematics

3.11 Reference application

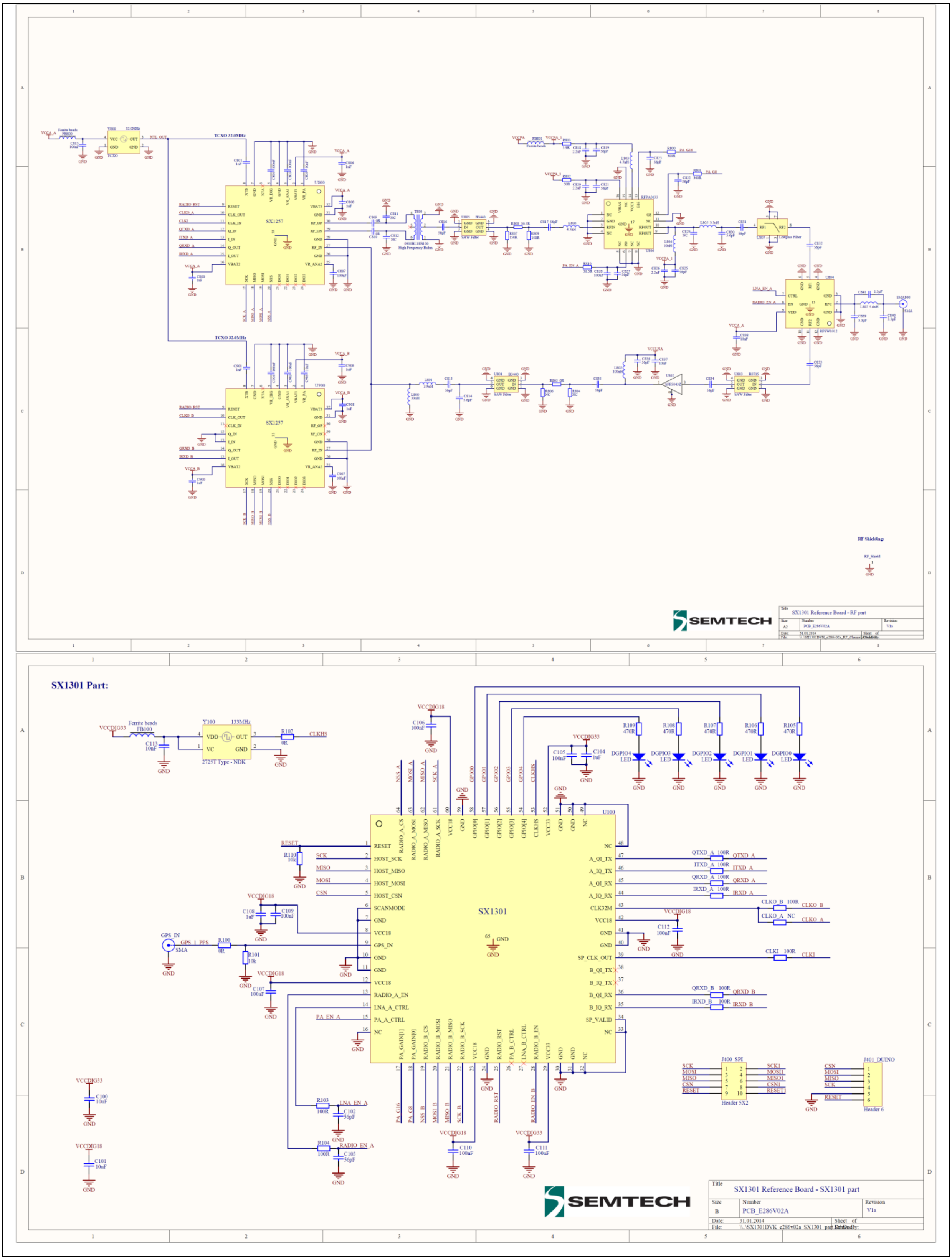


Figure 16 Reference application

3.12 SX1301 sensitivity performance in reference application

Sensitivities are given for 32 bytes payload, 10% PER.

Symbol	Descriptions	Conditions	Typ	Unit
RFS_SF12_0	LoRa sensitivity at SF12 : IF8 path	BW = 125 kHz BW = 250 kHz BW = 500 kHz	-140 -137 -134	dBm
RFS_SF12_07	LoRa sensitivity at SF12 : IF0 to 7 paths	BW = 125 kHz	-140	dBm
ACR_SF12_1M	Receiver CW interferer rejection at 1 MHz offset at SF12	BW = 125 kHz	+80	dB
CCR_SF12	Co-channel rejection at SF12	Wanted signal 10 dB above sensitivity	+25*	dB
RFS_SF7	LoRa sensitivity at SF7 : IF8 path	BW = 125 kHz BW = 250 kHz BW = 500 kHz	-126 -123 -120	dBm
RFS_SF7	LoRa sensitivity at SF7 : IF0 to 7 paths	BW = 125 kHz	-126	dBm
ACR_SF7_1M	Receiver CW interferer rejection at 1 MHz offset	BW = 125 kHz	+70	dB
CCR_SF7	Co-channel rejection at SF7	Wanted signal 10 dB above sensitivity	+9*	dB
RFS_F	FSK sensitivity	FDA = 50 kHz , BT = 100 kb/s	-103	dBm
BRF	Bit rate FSK	Programmable : limited by SSB: FDA + BRF/2 < 250 kHz	1.2 to 100	kb/s
FDA	Frequency deviation, FSK	Programmable	0.6 to 200	kHz

Note: * CCR>0 means that interferer level is greater than wanted signal level. LoRa modulation works with a negative S(N+I)R

Table 13 SX1301 performance in reference application

3.13 SX1301 sensitivity vs data rate in LoRa mode

The data rates and sensitivities are only function of the modulation bandwidth and the spreading factor. They are not function of the carrier frequency (868 or 433 MHz)

The sensitivities given are typical measurements done around 867 MHz using a SX1257 front-end with an external low-noise amplifier, SAW filter, and TRX switch as described in the “reference design” section.

3.13.1 125kHz mode: IF8, IF[0 to 7] paths

SF	Data rate (bit/sec)	Sensitivity (dBm)
7	5469	-130.0
8	3125	-132.5
9	1758	-135.0
10	977	-137.5
11	537	-140.0
12	293	-142.5

Table 14 Sensitivity with 125 kHz mode

3.13.2 250 & 500 kHz mode: IF8 only

SF	Data rate (bit/sec)	Sensitivity (dBm)
7	10938	-127.0
8	6250	-129.5
9	3516	-132.0
10	1953	-134.5
11	1074	-137.0
12	586	-139.5

Table 15 Sensitivity with 250 kHz mode

SF	Data rate (bit/sec)	Sensitivity (dBm)
7	21875	-124.0
8	12500	-126.5
9	7031	-129.0
10	3906	-131.5
11	2148	-134.0
12	1172	-136.5

Table 16 Sensitivity with 500 kHz mode

3.14 SX1301 interference rejection

The following graphs show typical measurement results at 870 MHz with 125 kHz bandwidth measured on the SX1301 with the documented front-end “reference design”.

The frequency asymmetry in the interferer rejection comes from the SAW filter transfer function. This measure is taken on the upper-most channels of the design; therefore the SAW filter starts to attenuate interferences above the wanted signal frequencies. The interferer rejection is arbitrarily limited to 100dB by the measurement setup.

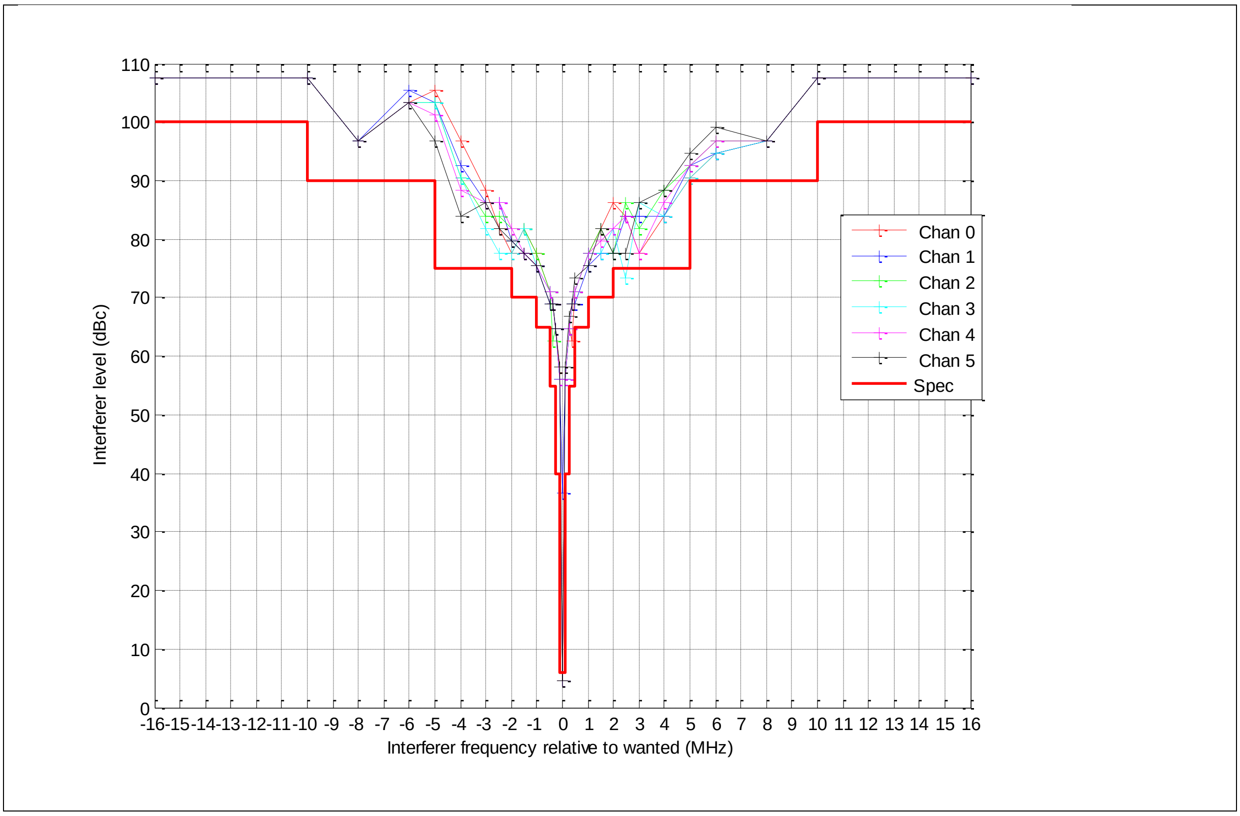


Figure 17 CW interferer rejection @ SF7

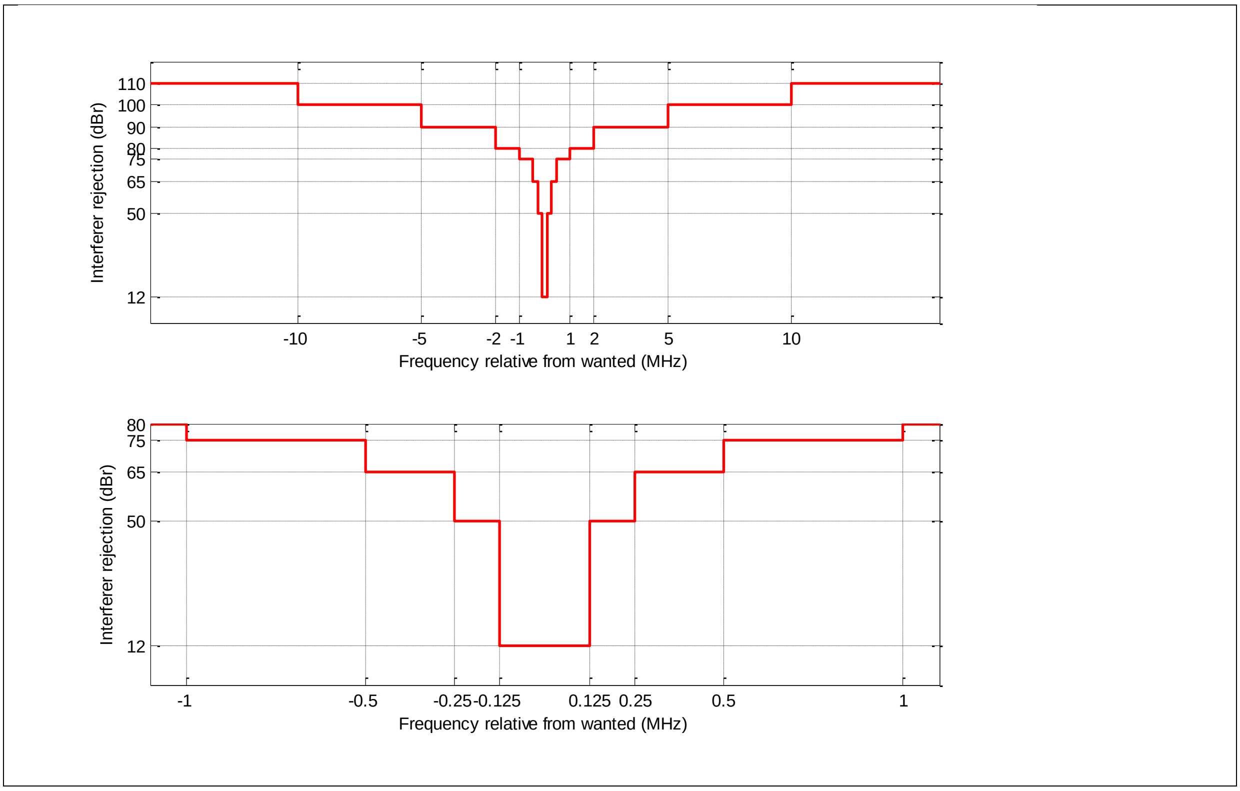


Figure 18 CW interferer rejection @ SF12

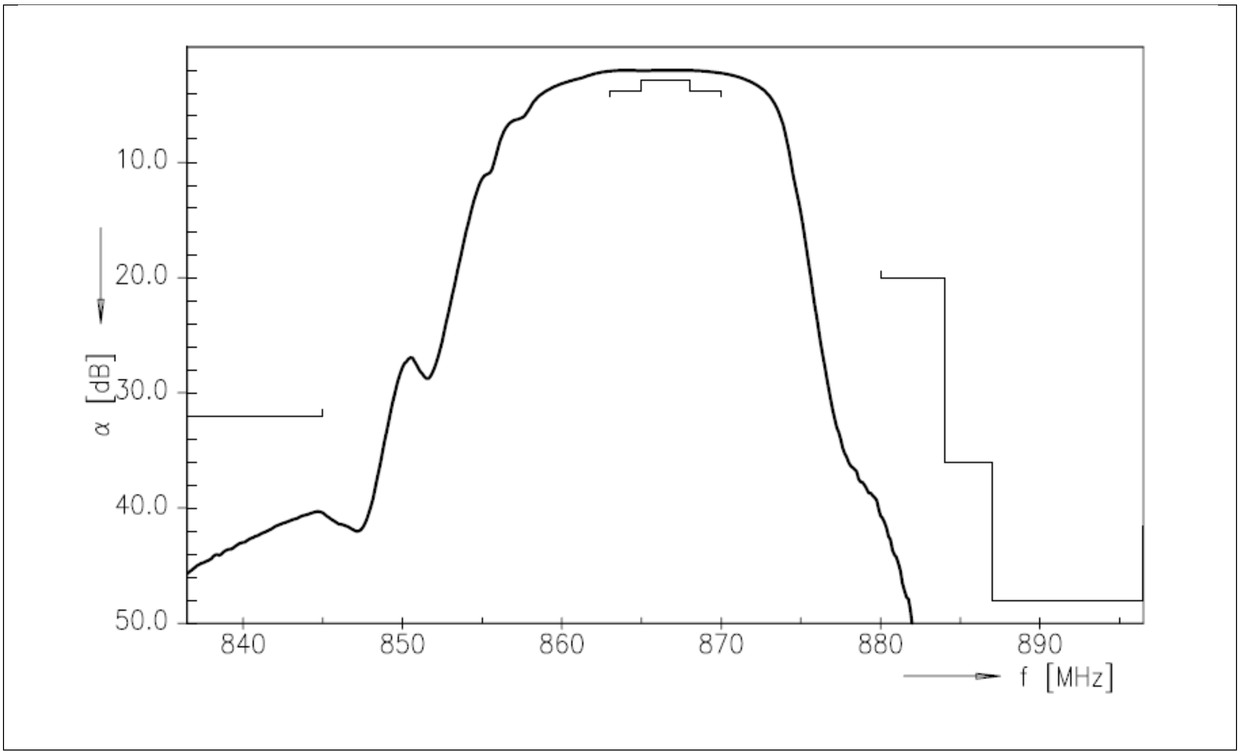


Figure 19 EPCOS B3117 SAW filter transfer function

3.15 Hardware Abstraction Layer (HAL)

3.15.1 Introduction

The Semtech SX1301 is an all-digital half-duplex radio modem capable of receiving multiple modulations, multiple radio channels, and multiple data rates simultaneously. This SX1301 is highly configurable.

Because of the variable number (and types) of radio channels, modems and transceivers, and because the different hardware implementations can be quite different (typically, not the same register mapping, naming and various features), presenting a unified hardware abstraction to the user can greatly simplify writing an application and porting an application between different hardware.

The first chapter of this document describes the Lora gateway abstraction presented to the user and what functions can the user use to interact with it.

The second chapter describes how the HAL manipulates the hardware. That chapter assumes that the hardware has the following characteristics:

- SX1301 based board
- Two SX1257 radios
- A native SPI link between the gateway host and the SX1301 LoRa concentrator

3.15.2 Abstraction presented to the gateway host

The system composed of a SX1301 and one or more radio transceivers is represented to the user as the following entities:

- 1 or more radio chains,
- 1 or more RX modems with a settable Intermediate Frequency (IF),
- A unified RX packet buffer,

- A single TX chain.

The link between the SX1301 and the gateway host is transparent for the user.

Radio chain
A radio chain selects and amplifies a limited portion of the RF spectrum, and digitizes it to be used by the modem chains.

A radio chain is characterized by its bandwidth, maximum and minimum allowed RF frequency in RX, maximum and minimum allowed RF frequency in TX.

For each radio chain, the settings are:

- Enabled or disabled: a radio chain must be enabled to receive or send packets through it.
- Center RF frequency: the portion of the RF spectrum available for RX modems will be [Fc - (BW/2) ; Fc + (BW/2)] if Fc is the center frequency and BW is the radio bandwidth.

The user configures the radio chains by calling the `lgw_rxrf_setconf` function one time for each chain with parameters contained in a `lgw_conf_rxrf_s` structure.

The present Lora gateway abstraction has 2 radio chains, numbered starting from 0.

Modem chain
A modem chain demodulates a small portion (a RF channel) of the RF spectrum digitized by a radio chain. Each modem chain RF channel can be placed individually inside the bandwidth of a radio using the IF (for Intermediate Frequency) setting, that's why they are designated in the abstraction as "IF+modem" chains.

The modem demodulates packets according to it intrinsic capabilities (e.g. the modulations it can process) and user-selected settings (e.g. what is the channel bandwidth for modems that supports multiple bandwidths) and send the receive packets to the RX buffer.

An IF+modem chain is characterized by its type (e.g. Lora "multi", FSK "standard"). That type define chat sort of signal can be demodulated and how the settings are interpreted.

For each IF+modem, the settings are:

- Enable or disabled.
- Source radio chain & IF frequency: selects from which radio chain the digitized RF spectrum will come. The center frequency of the demodulated RF channel is the sum of the center frequency of the selected radio chain and the IF. The IF can be positive or negative.
- Bandwidth: for modems that supports multiple channel bandwidth.
- Data rate: for modems that supports multiple data rates. It can be a combination for modems that supports multiples data rates *simultaneously*.

The user configures the IF+modem chains by calling the `lgw_rxif_setconf` function one time for each chain with parameters contained in a `lgw_conf_rxif_s` structure.

The present gateway has 4 Lora "simultaneous multi data rate" modem chain, and one Lora "standard" modem chain.

RX buffer
Packets that are received by all the modem chains are stored in the RX packet buffer until the gateway host come and fetch them.

There is no setting for that entity.

The user get packets stored in the RX buffer by calling the `lgw_receive` function that fill an array of `lgw_pkt_rx_s` structure for the packet metadata and allocate dynamic memory space to chain the packets payloads.

The present gateway can store up to 8 packets in its RX buffer.

TX chain

The TX chain is composed of a single multi-standard, multi-bandwidth, multi-data-rate modem and is used to send the single packet waiting in the TX packet buffer through one of the radio chains.

The settings are:

- Center channel frequency: must be in the range supported by the selected radio chain.
- TX Mode: to send the packet immediately, or to synchronize the sending with a specific event.
- Destination radio chain: selects through which radio chain the packet will be sent.
- The RF power at which the packet will be sent.
- The modulation used and several parameters that are shared by all modulations (e.g. the data rate) and parameters that are modulation-specific (e.g. the frequency deviation for FSK).
- The size and the content of the payload.

There is no need to configure the TX chain before starting the Lora gateway IP, all the settings to send a packet are contained in a `lgw_pkt_tx_s` structure and can be changed for each packet.

3.15.3 Composition of the software library

Configuration functions

Two functions are used to configure system:

- `lgw_rxrf_setconf` to configure the radio chains
- `lgw_rxif_setconf` to configure the modems and their Ifs

The other parts of the system are either not configurable or configured indirectly (using elements from the radio, IF and modem configuration).

The configuration is committed to the hardware when it is started. The hardware must be stopped before changing the configuration using the two functions.

Start and stop functions

Two functions are used to start and stop the hardware:

- `lgw_start` to initialize the hardware, configure it, load the firmware, and get it ready to send and receive packets.
- `lgw_stop` to stop the hardware, clear the configuration and put back the hardware to the post-reset state.

When the hardware is stopped (its initial state) the configuration functions can be called. Any attempt to send or receive packets will return an error.

Once the hardware is started, packets can be sent and received. Any attempt to alter the configuration will return an error.

Send and receive functions

The **`lgw_receive`** function is used to receive up to N packets. If there are no packet available in the RX packet buffer the function returns immediately indicating that 0 packets have be retrieved.

If there are packets waiting in the RX buffer, the function returns up to N packets. Every packet returned by the function is removed from the RX packet buffer, but if the function returns N packets (the maximum number allowed), and that number is less than the size of the RX packet buffer, there might still be packets left in the buffer after the function returns. The function can be called repeatedly to empty the RX packet buffer.

The **`lgw_send`** function is used to put a packet in the TX packet buffer. Only one packet can be in that buffer, being sent immediately or waiting for a triggered event to send it.

3.15.4 Interaction with the Lora hardware

This chapter describes the interaction between the different software components of the Lora gateway Hardware Abstraction Layer library and the hardware components that constitute the Lora gateway.

Configuration phase

During the configuration phase, the user program calls the two configuration functions **`lgw_rxrf_setconf`** and **`lgw_rxif_setconf`** any number of times.

Each time they are called, the two configuration functions:

- Check that the hardware is not started,
- Check the parameters sent by the user program are correct,
- If the parameters are within the range supported by the hardware, they are stored in a set of static variables in the HAL.

Initialization phase

After configuration, when the **`lgw_start`** function is called, the following actions are taken:

- The connection to the hardware is initialized and version is checked,
- The hardware is resetted,
- All radios are configured according to user settings,
- Some calibration parameters and static parameters are configured,
- All IF and modems are configured according to user settings,
- The firmware is loaded and controllers are started.

Receiving packets

After initialization, the Lora gateway will receive all the packets it is configured to receive and will put them in the RX packet buffer. Even packets that are received with corrupted data (error during the CRC checking) are available in the RX buffer.

When the **`lwg_receive`** function is called by the user program to get packet from the RX packet buffer, it first checks that the hardware is started and that the parameters are correct. After that, the following actions are taken:

- The function try to fetch a packet from the FIFO,
- If the FIFO is empty, the function returns nothing,
- If there was a packet in the FIFO, the function get the packet metadata and payload using the address that was in the FIFO,
- The payload is stored in a dynamically allocated memory space,

- The metadata are stored in a **lgw_pkt_rx_s** structure that was allocated by the user program,
- The FIFO is advanced to the next step,
- All the following steps are repeated until the FIFO is empty or the maximum number of packet to fetch has been reached.

When a packet is emitted, reception is automatically suspended. Reception is resumed when TX is finished without any intervention of the host. Packets that were being received at the beginning of the TX will be lost or their data will be corrupted.

Sending packets
When the **lwg_send** function is called by the user program to send a packet, it first checks that the hardware is started. After that, the following actions are taken:

- The function check that the parameters contained in the **lgw_pkt_tx_s** structure are correct,
- It then use the parameters in the structure to encode the metadata,
- It then copy the metadata and the payload to the TX packet buffer,
- Finally, it initializes the TX cycle and returns.

Depending on the settings, the packets might be sent immediately, or after a trigger event.

Stopping the gateway
When the **lgw_stop** function is called, the hardware is first resetted, and then the connection is closed.

3.15.5 Important HAL functions

HAL (Hardware Abstraction Layer software) can be found on GitHub at the following address:

https://github.com/Lora-net/lora_gateway

Function name	Used by function	Variables of the structure
`lgw_conf_rxrf_s`	`lgw_rxrf_setconf`	Configuration for an RF chain: center frequency, and whether it's enabled or not (disabled by default).
`lgw_conf_rxif_s`	`lgw_rxif_setconf`	Configuration for an IF chain and associated modem: center frequency, RF chain selection, and modem parameters.
`lgw_pkt_rx_s`	`lgw_receive`	Full payload of a received packet (contained in an byte array) and all the metadata this is available regarding this packet.
`lgw_pkt_tx_s`	`lgw_send`	All the parameters needed to send a packet (including frequency, modem parameters, etc) and the content of the packet payload.

Table 17 HAL main data structures

Function name	Function group	Parameters	Effect
`lgw_rxrf_setconf`	Configuration	Chain number & configuration structure.	Check user-provided configuration for a given RF chain, and commit it to an internal configuration data structure if correct. The configuration is not applied to hardware directly.
`lgw_rxif_setconf`			Check user-provided configuration for a given IF chain (+modem), and commit it to an internal configuration data structure if correct. The configuration is not applied to hardware directly.
`lgw_start`	Hardware management	None.	Connect with a SX1301-based concentrator board and apply the internal configuration.
`lgw_stop`			Stop the hardware, to save power or to be able to reconfigure it.
`lgw_send`	Packet handling	Max number of packets to retrieve, and pointer to an array of RX metadata & payload structures.	Retrieve up to a certain number of packets from the RX FIFO. Function is non-blocking and will return with as much packets as possible if less than the maximum are availables.
`lgw_receive`		TX payload & parameters structure.	Attempt to send a single packet using the user-provided parameters.

Table 18 HAL main functions

4 Memory map

The circuit registers are managed by the HAL software. The following register list and description is not exhaustive. It is presented here for information only, to support customers modifying the HAL for specific purposes.

Bits and registers that are not documented are reserved. They may include calibration values. It is important not to modify these bits and registers. If specific bits must be changed in a register with reserved bits, the register must be read first, specific bits modified while masking reserved bits and then the register can be written.

4.1 Registers list

The registers are paginated. Some registers are accessed independently on the page selection. These are listed on Table 19. Paginated registers are on Table 20, Table 21 and Table 22.

Page	Address	Name
All	0x00	RegPage
All	0x01	RegVer
All	0x02	RegRdbal
All	0x03	RegRdbah
All	0x04	RegRdbd
All	0x05	RegTdba
All	0x06	RegTdbd
All	0x0A	RegMpd
All	0x0B	RegRpns
All	0x0C	RegRpapl
All	0x0D	RegRpaph
All	0x0E	RegRps
All	0x0F	RegRpps
All	0x10	RegGen
All	0x11	RegCken
All	0x1B	RegGpsi
All	0X1C	RegGpso
All	0X1D	RegGpmode
All	0x1E	RegGpregi
All	0x1F	RegGprego
All	0x20	RegAgcsts
All	0x7D	RegArbsts
All	0x7F	RegId

Table 19 List of registers that are accessed without paging

Page	Address	Name
0x00	0x21	RegIqcfg
0x00	0x22	RegDeccfg
0x00	0x23	RegChrs
0x00	0x24	RegIf0l
0x00	0x25	RegIf0h
0x00	0x26	RegIf1l
0x00	0x27	RegIf1h
0x00	0x28	RegIf2l
0x00	0x29	RegIf2h
0x00	0x2A	RegIf3l
0x00	0x2B	RegIf3h
0x00	0x2C	RegIf4l
0x00	0x2D	RegIf4h
0x00	0x2E	RegIf5l
0x00	0x2F	RegIf5h
0x00	0x30	RegIf6l
0x00	0x31	RegIf6h
0x00	0x32	RegIf7l
0x00	0x33	RegIf7h
0x00	0x34	RegIf8l
0x00	0x35	RegIf8h
0x00	0x36	RegIf9l
0x00	0x37	RegIf9h
0x00	0x41	RegCor0deten
0x00	0x42	RegCor1deten
0x00	0x43	RegCor2deten
0x00	0x44	RegCor3deten
0x00	0x45	RegCor4deten
0x00	0x46	RegCor5deten
0x00	0x47	RegCor6deten
0x00	0x48	RegCor7deten
0x00	0x54	RegAmso124h
0x00	0x5D	RegTimtrak2
0x00	0x60	RegPrsymbnbl
0x00	0x61	RegPrsymbnbh
0x00	0x64	RegMisc_cfg2
0x00	0x65	RegHeader_cfg1
0x00	0x66	RegHeader_cfg2
0x00	0x6A	RegMcu_ctrl
0x00	0x6B	RegChann_select_rssi

Table 20 List of registers on page 0

Page	Address	Name
0x01	0x21	RegTxtrig
0x01	0x27	RegTx_offset_i
0x01	0x28	RegTx_offset_q
0x01	0x2B	RegBhimpcfg1
0x01	0x2C	RegBhimpcfg2
0x01	0x2E	RegBhsyncpos
0x01	0x2F	RegBhprsymnbl
0x01	0x3A	RegMbwssf_misc_cfg1
0x01	0x3B	RegMbwssf_misc_cfg2
0x01	0x3C	RegMbwssf_misc_cfg3
0x01	0x3D	RegMbwssf_misc_cfg4
0x01	0x3E	RegTx_status
0x01	0x3F	RegFsk_cfg1
0x01	0x40	RegFsk_cfg2
0x01	0x41	RegFsk_error_osr_tol
0x01	0x42	RegFsk_br_ratiol
0x01	0x43	RegFsk_br_ratioh
0x01	0x44	RegFsk_ref_pattern_0
0x01	0x45	RegFsk_ref_pattern_1
0x01	0x46	RegFsk_ref_pattern_2
0x01	0x47	RegFsk_ref_pattern_3
0x01	0x48	RegFsk_ref_pattern_4
0x01	0x49	RegFsk_ref_pattern_5
0x01	0x4A	RegFsk_ref_pattern_6
0x01	0x4B	RegFsk_ref_pattern_7
0x01	0x4C	RegFsk_pkt_length
0x01	0x52	RegFsk_aafc
0x01	0x53	RegFsk_pattern_timeout_cfgl
0x01	0x54	RegFsk_pattern_timeout_cfgh

Table 21 List of registers on page 1

Page	Address	Name
0x02	0x21	RegRadio_a_spi_1
0x02	0x22	RegRadio_a_spi_2
0x02	0x23	RegRadio_a_spi_3
0x02	0x25	RegRadio_a_spi_4
0x02	0x26	RegRadio_b_spi_1
0x02	0x27	RegRadio_b_spi_2
0x02	0x28	RegRadio_b_spi_3
0x02	0x2A	RegRadio_b_spi_4
0x02	0x2B	RegRadio_cfg
0x02	0x2C	RegPa_gain
0x02	0x2D	RegFe_a_ctrl_lut
0x02	0x2E	RegFe_b_ctrl_lut
0x02	0x38	RegValid_header_counter_m_ wssf
0x02	0x39	RegValid_header_counter_fsk
0x02	0x3A	RegValid_packet_counter_mb_ ssf
0x02	0x3B	RegValid_packet_counter_fsk
0x02	0x3C	RegChann_rssi
0x02	0x3D	RegBb_rssi
0x02	0x3E	RegDec_rssi
0x02	0x46	RegTimestamp_0
0x02	0x47	RegTimestamp_1
0x02	0x48	RegTimestamp_2
0x02	0x49	RegTimestamp_3
0x02	0x52	RegSpi_master_cfg
0x02	0x59	RegGps_cfg

Table 22 List of registers on page 2

4.2 Registers Description

Next tables list the detail content of each registers. Bit mode signification is r: read only, w: write only, rw: read/write. Reserved bits should be written with their reset state, they may be read different states.

4.2.1 All pages registers

All pages registers apply independently of the selected page.

Bits	Name	Mode	Reset	Description
[1:0]	PAGE_REG	RW	0x00	select the page of registers : valid range 0 to 2
[7]	SOFT_RESET	W	0x00	do a soft reset of all chip

Table 23 RegPage definition

Bits	Name	Mode	Reset	Description
[7:0]	VERSION	RW	0x67	Version of the current chip

Table 24 RegVer definition

Bits	Name	Mode	Reset	Description
[7:0]	RX_DATA_BUF_ADDR	RW	0x00	Memory address pointer in the RX data buffer, The databuffer is 4096 byte long

Table 25 RegRdbal definition

Bits	Name	Mode	Reset	Description
[7:0]	RX_DATA_BUF_ADDR	RW	0x00	(MSB)

Table 26 RegRdbah definition

Bits	Name	Mode	Reset	Description
[7:0]	RX_DATA_BUF_DATA	RW	0x00	Rx data buffer word. Each read operation on this register increments the RX_DATA_BUF_ADDR register.

Table 27 RegRdbd definition

Bits	Name	Mode	Reset	Description
[7:0]	TX_DATA_BUF_ADDR	RW	0x00	Address pointer in the TX data buffer

Table 28 RegTdba definition

Bits	Name	Mode	Reset	Description
[7:0]	TX_DATA_BUF_DATA	RW	0x00	Tx data buffer word. Each write operation on this register increments the TX_DATA_BUF_ADDR register. Readback of data is also possible

Table 29 RegTdbd definition

Bits	Name	Mode	Reset	Description
[7:0]	MCU_PROM_DATA	RW	0x00	data word to the MCU program memory. Each write operation increments MCU_PROM_ADDR register

Table 30 RegMpd definition

Bits	Name	Mode	Reset	Description
[7:0]	RX_PACKET_DATA_FIFO_NUM_STORED	RW	0x00	number of packets available in the Rx data buffer. Any write operation on this registers updates the value of : RX_PACKET_DATA_FIFO_ADDR_POINTER,RX_PACKET_DATA_FIFO_STATUS and RX_PACKET_DATA_FIFO_PAYLOAD_SIZE to point to the next packet in rx data buffer

Table 31 RegRpns definition

Bits	Name	Mode	Reset	Description
[7:0]	RX_PACKET_DATA_FIFO_ADDR_POINTER	R	0x00	Start address of the current packet in the Rx data buffer (LSB)

Table 32 RegRpapl definition

Bits	Name	Mode	Reset	Description
[7:0]	RX_PACKET_DATA_FIFO_ADDR_POINTER	R	0x00	Start address of the current packet in the Rx data buffer (MSB). Reading this register will automaticaly set the RX_DATA_BUF_ADDR pointer to RX_PACKET_DATA_FIFO_ADDR_POINTER value

Table 33 RegRpaph definition

Bits	Name	Mode	Reset	Description
[7:0]	RX_PACKET_DATA_FIFO_STATUS	R	0x00	CRC Status of the current packet. 1 -> packet valid, 3-> packet with payload CRC error

Table 34 RegRps definition

Bits	Name	Mode	Reset	Description
[7:0]	RX_PACKET_DATA_FIFO_PAYLOAD_SIZE	R	0x00	Size of the current packet payload in byte. To read the full packet + attached metadata, the user will have to read RX_PACKET_DATA_FIFO_PAYLOAD_SIZE+16 bytes

Table 35 RegRpps definition

Bits	Name	Mode	Reset	Description
[0]	MBWSSF_MODEM_ENABLE	RW	0x00	Enables backhaul Lora rx modem
[1]	CONCENTRATOR_MODEM_ENABLE	RW	0x00	enables the 8 sensor Lora rx modems
[2]	FSK_MODEM_ENABLE	RW	0x00	enables the FSK rx modem
[3]	GLOBAL_EN	RW	0x00	Enables everything else (except modems)

Table 36 RegGen definition

Bits	Name	Mode	Reset	Description
[0]	CLK32M_EN	RW	0x01	enables 32 MHz radio clock input
[1]	CLKHS_EN	RW	0x01	enables the correlators fast clock input

Table 37 RegCken definition

Bits	Name	Mode	Reset	Description
[3:0]	GPIO_SELECT_INPUT	RW	0x00	defines the routing of GPIO inputs to chip blocks : see GPIO mapping paragraph

Table 38 RegGpsi definition

Bits	Name	Mode	Reset	Description
[3:0]	GPIO_SELECT_OUTPUT	RW	0x00	defines signals routed to GPIO outpus : see GPIO mapping paragraph

Table 39 RegGpso definition

Bits	Name	Mode	Reset	Description
[4:0]	GPIO_MODE	RW	0x00	Sets GPIO direction : bit0=1 => GPIO0 is an output , bit0=0 => GPIO0 is an input

Table 40 RegGpmode definition

Bits	Name	Mode	Reset	Description
[4:0]	GPIO_PIN_REG_IN	R	0x00	allows to readback the gpio pin value thought the SPI interface. A GPIO defined as output will read as 0

Table 41 RegGpregi definition

Bits	Name	Mode	Reset	Description
[4:0]	GPIO_PIN_REG_OUT	RW	0x00	sets the 5 gpio pins output value. Only active if GPIO_SELECT_OUTPUT = 8

Table 42 RegGprego definition

Bits	Name	Mode	Reset	Description
[7:0]	MCU_AGC_STATUS	R	0x00	AGC MCU status , this status value is provided by the running AGC firmware loop

Table 43 RegAgcsts definition

Bits	Name	Mode	Reset	Description
[7:0]	MCU_ARB_STATUS	R	0x00	ARB MCU status , provided by the ARB firmware

Table 44 RegArbsts definition

Bits	Name	Mode	Reset	Description
[7:0]	CHIP_ID	RW	0x01	chip_id number

Table 45 RegId definition

4.2.2 Page 0 registers

Bits	Name	Mode	Reset	Description
[3]	RX_EDGE_SELECT	RW	0x00	Selects the clock edge used to sample the 1bit I/Q samples from the SX1257 0: falling edge, 1: rising edge
[4]	MISC_RADIO_EN	RW	0x00	Selects the radio interface, 0: SX1255/7 are used, 1: third-party radio interface used

Table 46 RegIqcfg definition

Bits	Name	Mode	Reset	Description
[3:0]	FILTER_GAIN	RW	0x07	Decimation filter gain in forced mode, 0: max gain, 15: min gain. each step equal to a -6dB atten

Table 47 RegDeccfg definition

Bits	Name	Mode	Reset	Description
[7:0]	RADIO_SELECT	RW	0xF0	selects radio source for each channel : LSB = channel 0 , MSB = channel7 : bit = 0: source is RADIO A , 1: RADIO B

Table 48 RegChrs definition

Bits	Name	Mode	Reset	Description
[7:0]	IF_FREQ_0	RW	0x80	define the IF frequency of sensor channel 0 (LSB) if_freq_Hz = (IF_FREQ_0/2^13)x4MHz

Table 49 RegIf0l definition

Bits	Name	Mode	Reset	Description
[4:0]	IF_FREQ_0	RW	0x1E	(MSB) , this is a 13bit signed number

Table 50 RegIf0h definition

Bits	Name	Mode	Reset	Description
[7:0]	IF_FREQ_1	RW	0x80	Channel 1 if_frequency

Table 51 RegIf1l definition

Bits	Name	Mode	Reset	Description
[4:0]	IF_FREQ_1	RW	0x1F	

Table 52 RegIf1h definition

Bits	Name	Mode	Reset	Description
[7:0]	IF_FREQ_2	RW	0x80	Channel 2 if_frequency

Table 53 RegIf2l definition

Bits	Name	Mode	Reset	Description
[4:0]	IF_FREQ_2	RW	0x00	

Table 54 RegIf2h definition

Bits	Name	Mode	Reset	Description
[7:0]	IF_FREQ_3	RW	0x80	Channel 3 if_frequency

Table 55 RegIf3l definition

Bits	Name	Mode	Reset	Description
[4:0]	IF_FREQ_3	RW	0x01	

Table 56 RegIf3h definition

Bits	Name	Mode	Reset	Description
[7:0]	IF_FREQ_4	RW	0x80	Channel 4 if_frequency

Table 57 RegIf4l definition

Bits	Name	Mode	Reset	Description
[4:0]	IF_FREQ_4	RW	0x1E	

Table 58 RegIf4h definition

Bits	Name	Mode	Reset	Description
[7:0]	IF_FREQ_5	RW	0x80	Channel 5 if_frequency

Table 59 RegIf5l definition

Bits	Name	Mode	Reset	Description
[4:0]	IF_FREQ_5	RW	0x1F	

Table 60 RegIf5h definition

Bits	Name	Mode	Reset	Description
[7:0]	IF_FREQ_6	RW	0x80	Channel 6 if_frequency

Table 61 RegIf6l definition

Bits	Name	Mode	Reset	Description
[4:0]	IF_FREQ_6	RW	0x00	

Table 62 RegIf6h definition

Bits	Name	Mode	Reset	Description
[7:0]	IF_FREQ_7	RW	0x80	Channel 7 if_frequency

Table 63 RefIf7l definition

Bits	Name	Mode	Reset	Description
[4:0]	IF_FREQ_7	RW	0x01	

Table 64 RegIf7h definition

Bits	Name	Mode	Reset	Description
[7:0]	IF_FREQ_8	RW	0x00	LoRa backhaul Channel if_frequency

Table 65 RegIf8l definition

Bits	Name	Mode	Reset	Description
[4:0]	IF_FREQ_8	RW	0x00	

Table 66 RegIf8h definition

Bits	Name	Mode	Reset	Description
[7:0]	IF_FREQ_9	RW	0x00	FSK chanel if_frequency

Table 67 RegIf9l definition

Bits	Name	Mode	Reset	Description
[4:0]	IF_FREQ_9	RW	0x00	

Table 68 RegIf9h definition

Bits	Name	Mode	Reset	Description
[6:0]	CORR0_DETECT_EN	RW	0x00	Selects active spreading factors for sensor channel 0 , LSB = SF6, MSB = SF12, bit : 0-> SF disabled, 1-> SF enabled

Table 69 RegCore0deten definition

Bits	Name	Mode	Reset	Description
[6:0]	CORR1_DETECT_EN	RW	0x00	Selects active spreading factors for sensor channel 1

Table 70 RegCore1deten definition

Bits	Name	Mode	Reset	Description
[6:0]	CORR2_DETECT_EN	RW	0x00	idem channel 2

Table 71 RegCore2deten definition

Bits	Name	Mode	Reset	Description
[6:0]	CORR3_DETECT_EN	RW	0x00	idem channel 3

Table 72 RegCore3deten definition

Bits	Name	Mode	Reset	Description
[6:0]	CORR4_DETECT_EN	RW	0x00	idem channel 4

Table 73 RegCore4deten definition

Bits	Name	Mode	Reset	Description
[6:0]	CORR5_DETECT_EN	RW	0x00	idem channel 5

Table 74 RegCore5deten definition

Bits	Name	Mode	Reset	Description
[6:0]	CORR6_DETECT_EN	RW	0x00	idem channel 6

Table 75 RegCore6deten definition

Bits	Name	Mode	Reset	Description
[6:0]	CORR7_DETECT_EN	RW	0x00	idem channel 7

Table 76 RegCore7deten definition

Bits	Name	Mode	Reset	Description
[3:0]	ADJUST_MODEM_START_OFFSET_SF12_RDX4	RW	0x0F	do not change it

Table 77 RegAmso124h definition

Bits	Name	Mode	Reset	Description
[5:0]	FREQ_TO_TIME_DRIFT	RW	0x09	Converts frequency offset into XTAL offset , depends on the frequency band used ,Should be 19 for 433MHz operation, and 17 for 470Mhz , exact value is round(8092/Frf)

Table 78 RegTimtrak2 definition

Bits	Name	Mode	Reset	Description
[7:0]	PREAMBLE_SYMB1_NB	RW	0x0A	Max preamble symbol number expected (MSB) , a longer preamble will trig a Header Error , reception will be dropped

Table 79 RegPrsymbnbl definition

Bits	Name	Mode	Reset	Description
[7:0]	PREAMBLE_SYMB1_NB	RW	0x00	

Table 80 RegSymbnbh definition

Bits	Name	Mode	Reset	Description
[6:0]	PPM_OFFSET	RW	0x00	Selects reduced encoding for the various SF on sensor channels, LSB=SF6 , MSB=SF12 , a 1 means reduced encoding is used, must match the encoding used by the transmitter

Table 81 RegMisc_cfg2 definition

Bits	Name	Mode	Reset	Description
[7:0]	MAX_PAYLOAD_LEN	RW	0xFF	Sets the maximum payload length expected on the sensor rx modem. longer packets will be dropped

Table 82 RegHeader_cfg1 definition

Bits	Name	Mode	Reset	Description
[0]	ONLY_CRC_EN	RW	0x01	0-> no CRC_EN filtering performed, 1-> only packets with CRC enabled are accepted

Table 83 RegHeader_cfg2 definition

Bits	Name	Mode	Reset	Description
[0]	MCU_RST_0	RW	0x01	Puts the ARB MCUs in reset , write 0 to restart
[1]	MCU_RST_1	RW	0x01	Puts the AGC MCUs in reset , write 0 to restart
[2]	MCU_SELECT_MUX_0	RW	0x00	Enables ARB MCU program RAM upload through the SPI interface
[3]	MCU_SELECT_MUX_1	RW	0x00	Enables AGC MCU program RAM upload through the SPI interface
[4]	MCU_CORRUPTION_DETECTED_0	R	0x00	Status, 1 if a program memory parity error occured on the ARB MCU, this means that the code ram content is incorrect
[5]	MCU_CORRUPTION_DETECTED_1	R	0x00	Status, 1 if a program memory parity error occured on the AGC MCU, this means that the code ram content is incorrect

Table 84 RegMcu_ctrl definition

Bits	Name	Mode	Reset	Description
[7:0]	CHANN_SELECT_RSSI	RW	0x01	Selects the channel RSSI to monitor : values 0 to 7

Table 85 RegChann_select_rssi definition

4.2.3 Page 1 registers

Bits	Name	Mode	Reset	Description
[0]	TX_TRIG_IMMEDIATE	W	0x00	A 1 written on this register trigs an immediate transmission , this registers automatically resets to 0
[1]	TX_TRIG_DELAYED	W	0x00	A 1 written on this register trigs a time triggered transmission, the transmission will start at the programmed time
[2]	TX_TRIG_GPS	W	0x00	A 1 written on this register trigs a transmission on the next rising edge of the GPS PPS signal

Table 86 RegTrig definition

Bits	Name	Mode	Reset	Description
[7:0]	TX_OFFSET_I	RW	0x00	Transmiter carrier leakage cancellation : I

Table 87 RegTx_offset_i definition

Bits	Name	Mode	Reset	Description
[7:0]	TX_OFFSET_Q	RW	0x00	Transmiter carrier leakage cancellation : Q

Table 88 RegTx_offset_q definition

Bits	Name	Mode	Reset	Description
[0]	MBWSSF_IMPLICIT_HEADER	RW	0x00	Enables implict header mode for the backhaul lora RX modem
[1]	MBWSSF_IMPLICIT_CRC_EN	RW	0x00	Sets CRC in implicit header mode
[4:2]	MBWSSF_IMPLICIT_CODING_RATE	RW	0x00	Sets coding rate in implicit header mode

Table 89 RegBhimpcfg1 definition

Bits	Name	Mode	Reset	Description
[7:0]	MBWSSF_IMPLICIT_PAYLOAD_LENGHT	RW	0x00	Sets payload length in implicit header mode

Table 90 RegBhimpcfg2 definition

Bits	Name	Mode	Reset	Description
[3:0]	MBWSSF_FRAME_SYNCH_PEAK1_POS	RW	0x01	Reserved: End of preamble marker position
[7:4]	MBWSSF_FRAME_SYNCH_PEAK2_POS	RW	0x02	Reserved: End of preamble marker position

Table 91 RegBhsyncpos definition

Bits	Name	Mode	Reset	Description
[7:0]	MBWSSF_PREAMBLE_SYMB1_NB	RW	0x0A	Sets the preamble maximum symbol number expected , a longer preamble will trigger a header error , demoudulation will be dropped

Table 92 RegBhprsymnbl definition

Bits	Name	Mode	Reset	Description
[1:0]	MBWSSF_MODEM_BW	RW	0x00	Sets the LoRa RX backhaul modem bandwidth , 0 =125kHZ, 1=250kHZ, 2=500kHz
[2]	MBWSSF_RADIO_SELECT	RW	0x00	Selects the radio source for the backhaul modem
[3]	MBWSSF_RX_CHIRP_INVERT	RW	0x01	Reserved: inverts chirp slope

Table 93 RegMbwssf_misc_cfg1 definition

Bits	Name	Mode	Reset	Description
[3:0]	MBWSSF_LLR_SCALE	RW	0x08	Reserved: same as SX1272
[5:4]	MBWSSF_SNR_AVG_CST	RW	0x03	Reserved: same as SX1272
[6]	MBWSSF_PPM_OFFSET	RW	0x00	Enables reduced encoding for improoved timing tracking robustness , should match transmitter

Table 94 RegMbwssf_misc_cfg2 definition

Bits	Name	Mode	Reset	Description
[3:0]	MBWSSF_RATE_SF	RW	0x07	sets backhaul spreading factor: 7 to 12
[4]	MBWSSF_ONLY_CRC_EN	RW	0x01	filters out packets without CRC field

Table 95 RegMbwssf_misc_cfg3 definition

Bits	Name	Mode	Reset	Description
[7:0]	MBWSSF_MAX_PAYLOAD_LEN	RW	0xFF	filters out packet with payload length longer than this value

Table 96 RegMbwssf_misc_cfg4 definition

Bits	Name	Mode	Reset	Description
[7:0]	TX_STATUS	R	0x80	contains the Transmit modem status : {receiving,tx_modem running,ramping,tx trigger armed,0,0,waiting for GPS,waiting for timer }

Table 97 RegTx_status definition

Bits	Name	Mode	Reset	Description
[2:0]	FSK_CH_BW_EXPO	RW	0x00	FSK channel bandwidth SSB = 32MHz/(16 * 2^(2+ch_bw_expo) ,0 forbidden , 1=+/-250kHz , 7=+/-3.9kHz
[5:3]	FSK_RSSI_LENGTH	RW	0x00	RSSI average window length : window length = 2^(1+fsk_rssi_length)
[6]	FSK_RX_INVERT	RW	0x00	Inverts FSK modem I/q samples
[7]	FSK_PKT_MODE	RW	0x00	0=fixed length packet , length is set by FSK_PKT_LENGTH register, 1=packet length is contained in header

Table 98 RegFsx_cfg definition

Bits	Name	Mode	Reset	Description
[2:0]	FSK_PSIZE	RW	0x00	length of the Sync pattern is 1+FSK_PSIZE bytes
[3]	FSK_CRC_EN	RW	0x00	Enables CRC16 calculation&check
[5:4]	FSK_DCFREE_ENC	RW	0x00	Encoding mode: 00=dc free encoding, 01=Manchester 10=Whitening 11=forbiden
[6]	FSK_CRC_IBM	RW	0x00	defines CRC type : 0=CCITT CRC16 , 1=IBM CRC as in CC1100
[7]	FSK_PREAMBLE_SEQ	RW	0x00	Reserved: unknown ??

Table 99 RegFsk_cfg2 definition

Bits	Name	Mode	Reset	Description
[4:0]	FSK_ERROR_OSR_TOL	RW	0x00	number of tolerated chip error during preamble correlation
[7]	FSK_RADIO_SELECT	RW	0x00	Selects the FSK modem sample radio source, 0=radio A , 1=radio B

Table 100 RegFsk_error_osr_tol definition

Bits	Name	Mode	Reset	Description
[7:0]	FSK_BR_RATIO[7..0]	RW	0x00	defines the FSK bitrate , bitrate = 32MHz/br_ratio (LSB)

Table 101 RegFsk_br_ratiol definition

Bits	Name	Mode	Reset	Description
[7:0]	FSK_BR_RATIO[15..8]	RW	0x00	(MSB)

Table 102 RegFsk_br_ratioh definition

Bits	Name	Mode	Reset	Description
[7:0]	FSK_REF_PATTERN[7..0]	RW	0x00	8 LSBs of the programmable sync word , sync word is sent MSB first in the air, should match the transmitter setting

Table 103 RegFsk_ref_pattern_0 definition

Bits	Name	Mode	Reset	Description
[7:0]	FSK_REF_PATTERN[15..8]	RW	0x00	

Table 104 RegFsk_ref_pattern_1 definition

Bits	Name	Mode	Reset	Description
[7:0]	FSK_REF_PATTERN[23..16]	RW	0x00	

Table 105 RegFsk_ref_pattern_2 definition

Bits	Name	Mode	Reset	Description
[7:0]	FSK_REF_PATTERN[31..24]	RW	0x00	

Table 106 RegFsk_ref_pattern_3 definition

Bits	Name	Mode	Reset	Description
[7:0]	FSK_REF_PATTERN[39..32]	RW	0x00	

Table 107 RegFsk_ref_pattern_4 definition

Bits	Name	Mode	Reset	Description
[7:0]	FSK_REF_PATTERN[47..40]	RW	0x00	

Table 108 RegFsk_ref_pattern_5 definition

Bits	Name	Mode	Reset	Description
[7:0]	FSK_REF_PATTERN[55..48]	RW	0x00	

Table 109 RegFsk_ref_pattern_6 definition

Bits	Name	Mode	Reset	Description
[7:0]	FSK_REF_PATTERN[63..56]	RW	0x00	first byte of sync word , if a single byte is used as sync word , it will be matched to this register

Table 110 RegFsk_ref_pattern_7 definition

Bits	Name	Mode	Reset	Description
[7:0]	FSK_PKT_LENGTH	RW	0x00	length of the packet in byte in fixed packet length mode , in variable length mode this register is used to filter out packet payloads longer than FSK_PKT_LENGTH

Table 111 RegFsk_pkt_length definition

Bits	Name	Mode	Reset	Description
[0]	FSK_AUTO_AFC_ON	RW	0x01	enables Automatic frequency offset compensation, active by default

Table 112 RegFsk_aafc definition

Bits	Name	Mode	Reset	Description
[7:0]	FSK_PATTERN_TIMEOUT_CFG	RW	0x00	time out following preamble detection expressed in bits (LSB). If no sync word is found after FSK_PATTERN_TIMEOUT_CFG bit periods , the modem goes back in preamble search & AFC

Table 113 RegFsk_pattern_timeout_cfg definition

Bits	Name	Mode	Reset	Description
[1:0]	FSK_PATTERN_TIMEOUT_CFG	RW	0x00	(MSB)

Table 114 RegFsk_pattern_timeout_cfg definition

4.2.4 Page 2 registers

Bits	Name	Mode	Reset	Description
[7:0]	SPI_RADIO_A__DATA	RW	0x00	Data sent to radio A SPI interface when HOST controls the radio SPI interface directly

Table 115 RegRadio_a_spi_1 definition

Bits	Name	Mode	Reset	Description
[7:0]	SPI_RADIO_A__DATA_READBACK	R	0x00	Byte read from radio A during a SPI transaction when HOST controls the radio SPI interface directly

Table 116 RegRadio_a_spi_2 definition

Bits	Name	Mode	Reset	Description
[7:0]	SPI_RADIO_A__ADDR	RW	0x00	Addr word sent to radio A SPI interface when HOST controls the radio SPI interface directly

Table 117 RegRadio_a_spi_3 definition

Bits	Name	Mode	Reset	Description
[0]	SPI_RADIO_A__CS	RW	0x00	A rising edge on this register triggers a single byte SPI transaction on the radio A SPI interface

Table 118 RegRadio_a_spi_4 definition

Bits	Name	Mode	Reset	Description
[7:0]	SPI_RADIO_B__DATA	RW	0x00	idem

Table 119 RegRadio_b_spi_1 definition

Bits	Name	Mode	Reset	Description
[7:0]	SPI_RADIO_B__DATA_READBACK	R	0x00	idem

Table 120 RegRadio_b_spi_2 definition

Bits	Name	Mode	Reset	Description
[7:0]	SPI_RADIO_B__ADDR	RW	0x00	idem

Table 121 RegRadio_b_spi_3 definition

Bits	Name	Mode	Reset	Description
[0]	SPI_RADIO_B__CS	RW	0x00	idem for radio B

Table 122 RegRadio_b_spi_4 definition

Bits	Name	Mode	Reset	Description
[0]	RADIO_A_EN	RW	0x00	Drives the Enable pin of the Radio A
[1]	RADIO_B_EN	RW	0x00	Drives the Enable pin of the Radio A
[2]	RADIO_RST	RW	0x01	Drive the RST pin of both radios
[3]	LNA_A_EN	RW	0x00	when controlled by the HOST ,used to control the external LNA&PA state , used as index LSB of control logic LUT ,see LNA_A_CTRL_LUT&PA_A_CTRL_LUT
[4]	PA_A_EN	RW	0x00	used as MSB of control logic LUT , only active when HOST directly controls external front-end
[5]	LNA_B_EN	RW	0x00	idem
[6]	PA_B_EN	RW	0x00	idem

Table 123 RegRadio_cfg definition

Bits	Name	Mode	Reset	Description
[1:0]	PA_GAIN	RW	0x00	External PA optionnal gain control signal

Table 124 RegPa_gain definition

Bits	Name	Mode	Reset	Description
[3:0]	LNA_A_CTRL_LUT	RW	0x02	LNA_A_CTRL pin state is LNA_A_CTRL_LUT({pa_a_en,lna_a,en}) , allows arbitrary logic contorl of any external RF Front-end with 2 wires interface
[7:4]	PA_A_CTRL_LUT	RW	0x04	PA_A_CTRL pin state is PA_A_CTRL_LUT({pa_a_en,lna_a,en})

Table 125 RegFe_a_ctrl_lut definition

Bits	Name	Mode	Reset	Description
[3:0]	LNA_B_CTRL_LUT	RW	0x02	idem for radio B RF front-end
[7:4]	PA_B_CTRL_LUT	RW	0x04	idem for radio B RF front-end

Table 126 RegFe_b_ctrl_lut definition

Bits	Name	Mode	Reset	Description
[7:0]	VALID_HEADER_COUNTER_MBWSSF	R	0x00	Number of valid header received by the LoRa backhaul RX modem , wraps to 0 when it reaches 255

Table 127 RegValid_header_counter_mbwssf definition

Bits	Name	Mode	Reset	Description
[7:0]	VALID_HEADER_COUNTER_FSK	R	0x00	Number of valid header received by the FSK backhaul RX modem

Table 128 RegValid_header_counter_fsk definition

Bits	Name	Mode	Reset	Description
[7:0]	VALID_PACKET_COUNTER_MBWSSF	R	0x00	Number of valid packet received by the LoRa backhaul RX modem

Table 129 RegValid_packet_counter_mbwssf definition

Bits	Name	Mode	Reset	Description
[7:0]	VALID_PACKET_COUNTER_FSK	R	0x00	Number of valid header received by the FSK backhaul RX modem

Table 130 RegValid_packet_counter_fsk definition

Bits	Name	Mode	Reset	Description
[7:0]	CHANN_RSSI	R	0x00	Instant RSSI measured in the digital channelizer , channel is selected with the CHANN_SELECT_RSSI register

Table 131 RegChann_rssi definition

Bits	Name	Mode	Reset	Description
[7:0]	BB_RSSI	R	0x00	Instant RSSI measured after the first decimation stage with ~8MHz bandwidth , detects analog ADC saturation

Table 132 RegBb_rssi definition

Bits	Name	Mode	Reset	Description
[7:0]	DEC_RSSI	R	0x00	Instant RSSI measured at the output of the decimation filter with bandwidth = +/-2MHz

Table 133 RegDec_rssi definition

Bits	Name	Mode	Reset	Description
[7:0]	TIMESTAMP[7..0]	R	0x00	32bits time counter clocked at 1Mhz , this value is latched on the falling edge of SPI CSN signal , the 32bits (4bytes) of this register should be read in 1 single burst

Table 134 RegTimestamp_0 definition

Bits	Name	Mode	Reset	Description
[7:0]	TIMESTAMP[15..8]	R	0x00	

Table 135 RegTimestamp_1 definition

Bits	Name	Mode	Reset	Description
[7:0]	TIMESTAMP[23..16]	R	0x00	

Table 136 RegTimestamp_2 definition

Bits	Name	Mode	Reset	Description
[7:0]	TIMESTAMP[31..24]	R	0x00	

Table 137 RegTimestamp_3 definition

Bits	Name	Mode	Reset	Description
[0]	SPI_MASTER_CHIP_SELECT_POLARITY	RW	0x00	Selects the polarity of the radio SPI master to accomodate third party radios if required : 0 = active low
[1]	SPI_MASTER_CPOL	RW	0x00	Radio SPI master Clock Polarity , by default ok for SX1257
[2]	SPI_MASTER_CPHA	RW	0x00	Radio SPI master CPHA parameter , by default ok for SX1257

Table 138 RegSpi_master_cfg definition

Bits	Name	Mode	Reset	Description
[0]	GPS_EN	RW	0x00	Selects the GPS input to sample the integrated time counter , if 0 the current value is read , if 1 the value at the last GPS pulse is read
[1]	GPS_POL	RW	0x01	Selects on which edge of the GPS PPS signal the time counter should be sampled: 1 means rising edge

Table 139 RegGps_cfg definition

5 External components

A decoupling capacitor (Cdec) is required to minimize the ripple on the power lines.

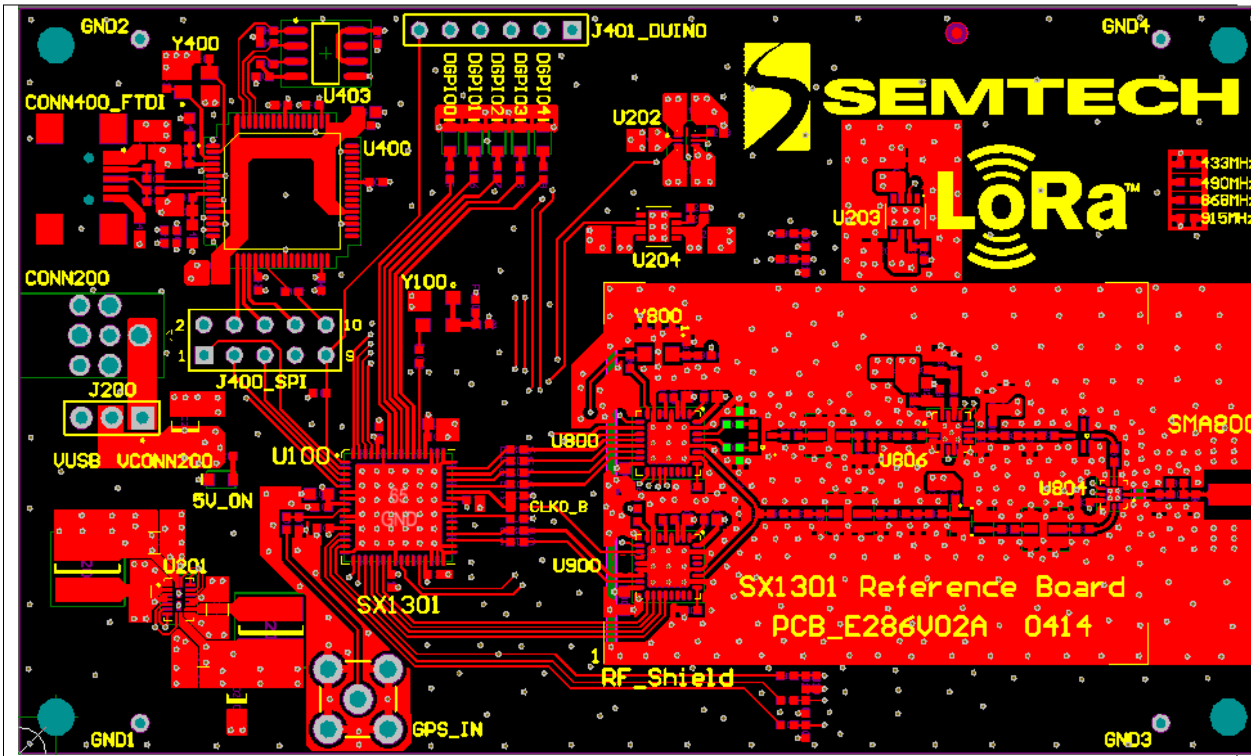
Component	Value	Manufacturer	Part number	Package
Cdec	100 nF, 10 V	TDK	C0603X5R1A104KT	0201 (0603 metric)
	100 nF, 6.3 V	Taiyo Yuden	EMK063AC6104MP-F	0201 (0603 metric)
	100 nF, 6.3 V	Murata	GRM033R60J104ME19D	0201 (0603 metric)

Table 140 Recommended external components

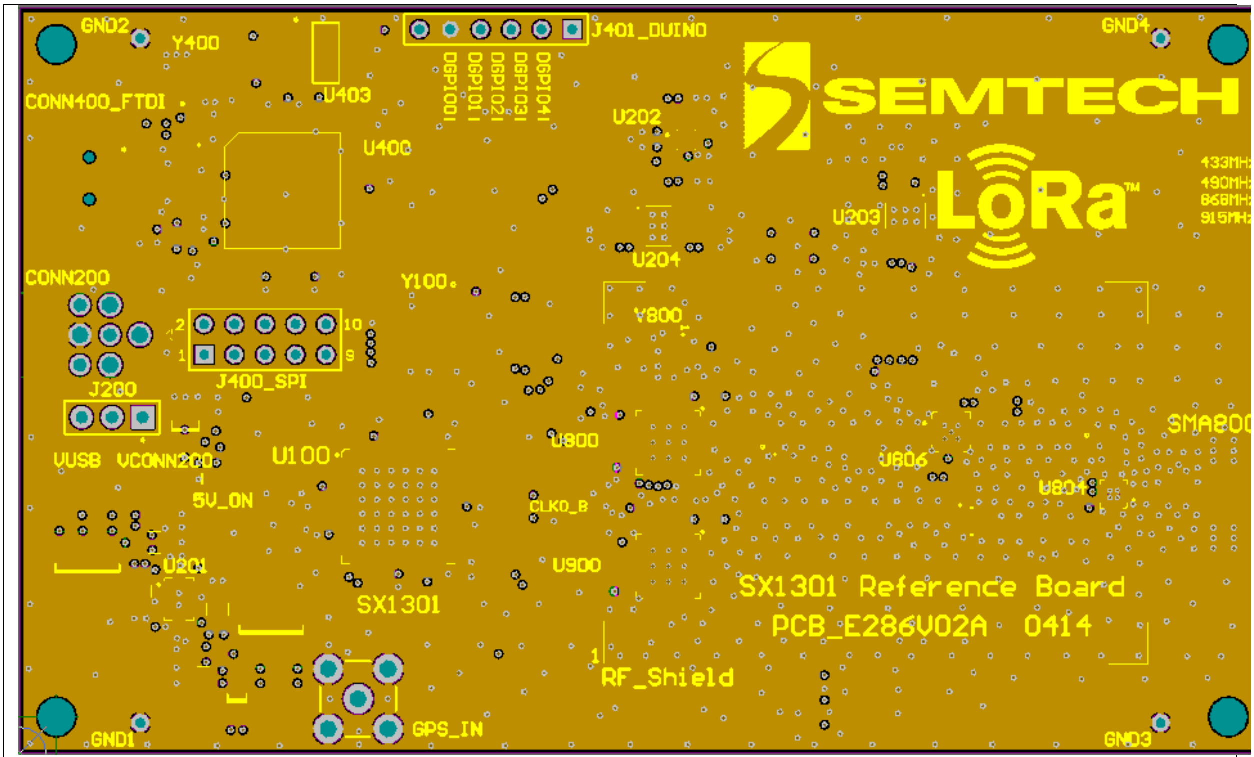
6 PCB Layout Considerations

The bottom ground paddle must be soldered to a ground plate. The ground plate must be large enough to support SX1301 power dissipation.

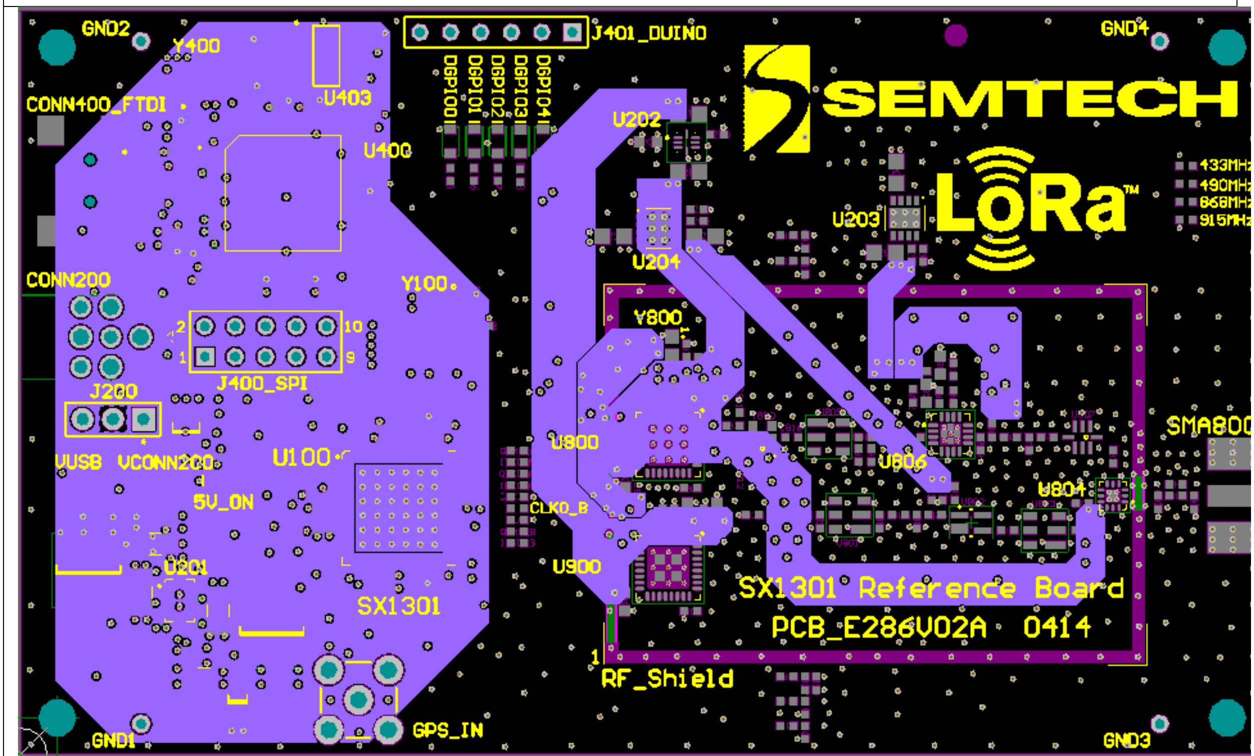
The PCB layout must minimize distances between the IC and the decoupling capacitors.



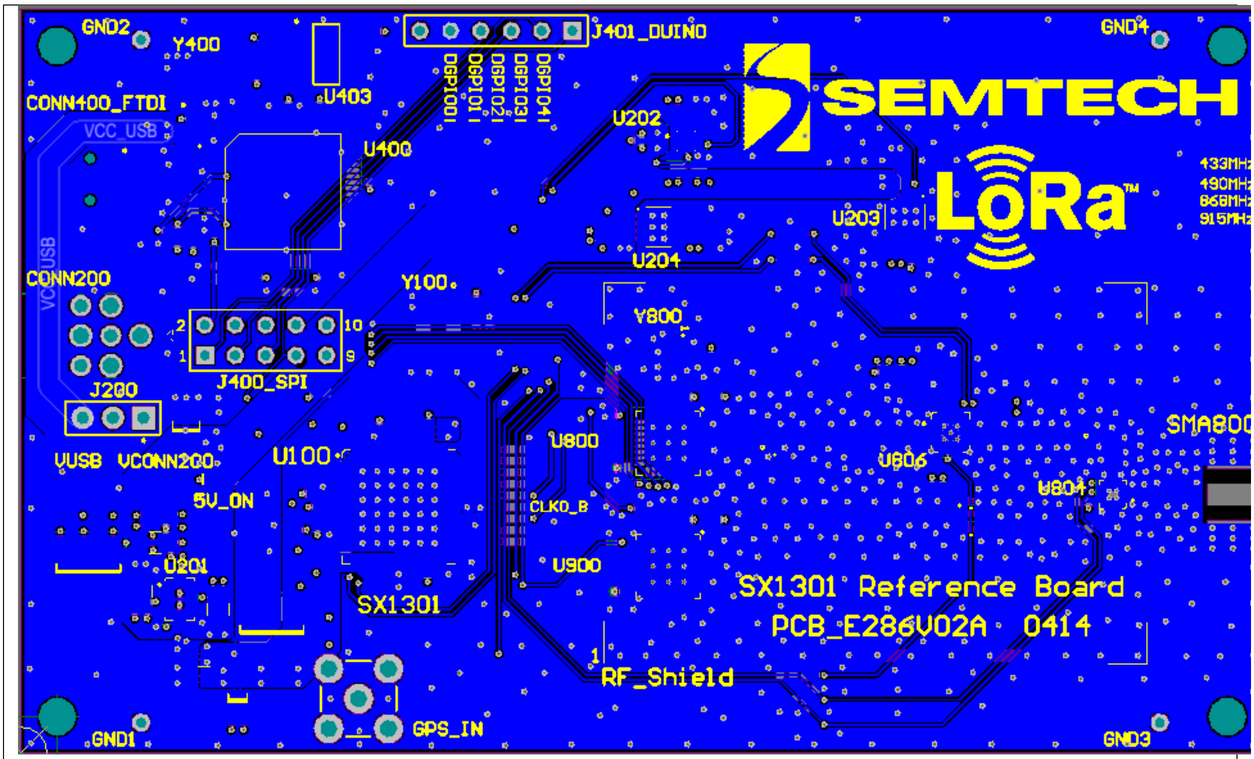
Top layer - signals



Second layer - shield



Third layer - power

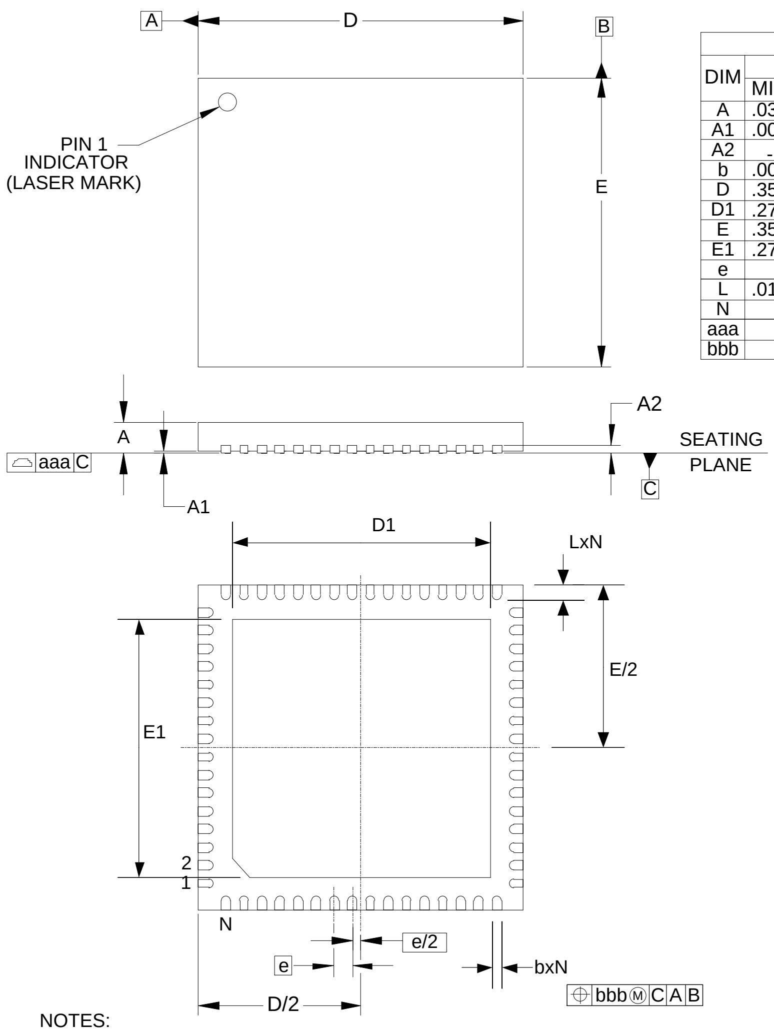


Bottom layer – shield and thermal dissipation

Figure 20 PCB layout example

7 Packaging Information

7.1 Package Outline Drawing



DIM	INCHES MIN	INCHES NOM	INCHES MAX	MILLIMETERS MIN	MILLIMETERS NOM	MILLIMETERS MAX
A	.031	-	.039	0.80	-	1.00
A1	.000	-	.002	0.00	-	0.05
A2	-	(.008)	-	-	(0.20)	-
b	.007	.010	.012	0.18	0.25	0.30
D	.350	.354	.358	8.90	9.00	9.10
D1	.276	.281	.285	7.00	7.15	7.25
E	.350	.354	.358	8.90	9.00	9.10
E1	.276	.281	.285	7.00	7.15	7.25
e		.020 BSC			0.50 BSC	
L	.012	.016	.020	0.30	0.40	0.50
N		64			64	
aaa		.003			0.08	
bbb		.004			0.10	

NOTES:

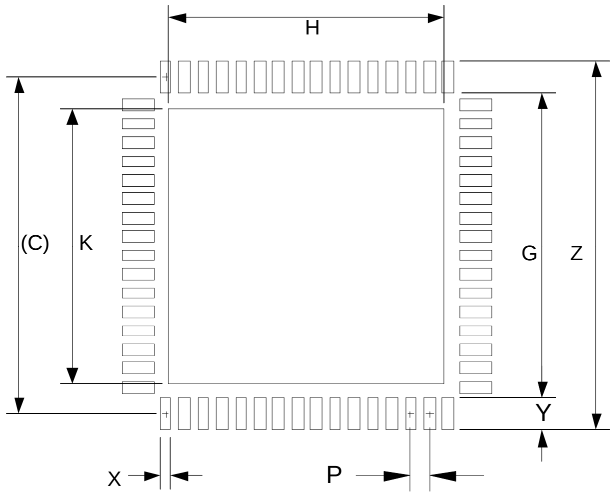
1. CONTROLLING DIMENSIONS ARE IN MILLIMETERS (ANGLES IN DEGREES).
2. COPLANARITY APPLIES TO THE EXPOSED PAD AS WELL AS THE TERMINALS.

Figure 21 Package dimensions

7.2 Thermal impedance of package

Thermal impedance with natural convection is 16.4 °C/W. Thermal impedance with heat sink on package bottom is 0.18 °C/W.

7.3 Land Pattern Drawing



DIMENSIONS		
DIM	INCHES	MILLIMETERS
C	(.352)	(8.95)
G	.319	8.10
H	.287	7.30
K	.287	7.30
P	.020	0.50
X	.012	0.30
Y	.033	0.85
Z	.386	9.80

NOTES:

1. CONTROLLING DIMENSIONS ARE IN MILLIMETERS (ANGLES IN DEGREES).
2. THIS LAND PATTERN IS FOR REFERENCE PURPOSES 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 22 Land pattern drawing

8 Revision Information

Revision	Information
V2.0	First released version

Contact information

Semtech Corporation

Wireless & Sensing Products Division

200 Flynn Road, Camarillo, CA 93012

Phone: (805) 498-2111 Fax: (805) 498-3804

www.semtech.com