



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



## Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



## HIGH-PERFORMANCE, LOW-CURRENT TRANSCEIVER

### Features

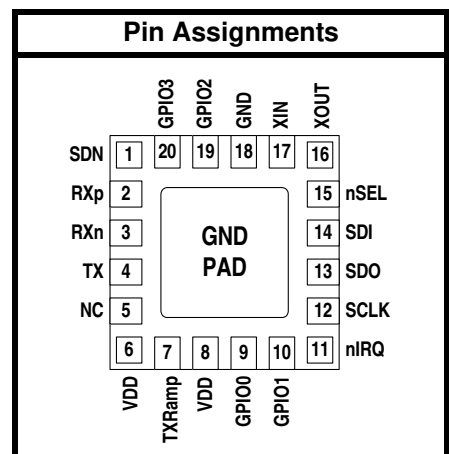
- Frequency range = 425–525 MHz
- Receive sensitivity = –124 dBm
- Modulation
  - (G)FSK
  - OOK
- Max output power
  - +20 dBm
- Low active power consumption
  - 14 mA RX
- Ultra low current powerdown modes
  - 30 nA shutdown, 50 nA standby
- Data rate = 100 bps to 500 kbps
- Fast wake and hop times
- Power supply = 1.8 to 3.6 V
- Excellent selectivity performance
  - 58 dB adjacent channel
  - 75 dB blocking at 1 MHz
- Antenna diversity and T/R switch control
- Highly configurable packet handler
- TX and RX 64 byte FIFOs
- Auto frequency control (AFC)
- Automatic gain control (AGC)
- Low BOM
- Low battery detector
- Temperature sensor
- 20-Pin QFN package
- IEEE 802.15.4g ready
- China regulatory

### Applications

- China smart meters

### Description

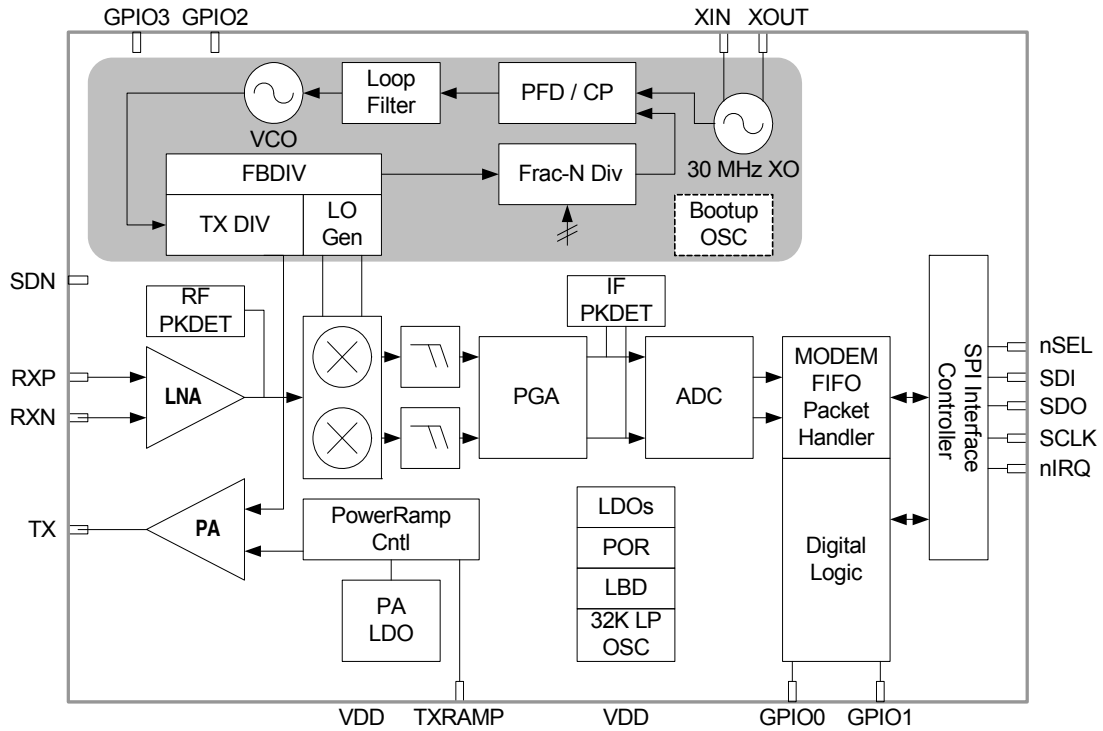
Silicon Laboratories' Si4438 is a high-performance, low-current transceivers covering the sub-GHz frequency bands from 425 to 525 MHz. The Si4438 is targeted at the Chinese smart meter market and is especially suited for electric meters. This device is footprint- and pin-compatible with the Si446x radios, which provide industry-leading performance for worldwide sub-GHz applications. The radios are part of the EZRadioPRO<sup>®</sup> family, which includes a complete line of transmitters, receivers, and transceivers covering a wide range of applications. All parts offer outstanding sensitivity of –124 dBm while achieving extremely low active and standby current consumption. The 58 dB adjacent channel selectivity with 12.5 kHz channel spacing ensures robust receive operation in harsh RF conditions. The Si4438 offers exceptional output power of up to +20 dBm with outstanding TX efficiency. The high output power and sensitivity results in an industry-leading link budget of 144 dB allowing extended ranges and highly robust communication links.



Patents pending

# Si4438

## Functional Block Diagram



Product	Freq. Range	Max Output Power	TX Current	RX Current
Si4438	425–525 MHz	+20 dBm	75 mA	14 mA

---

**TABLE OF CONTENTS**


---

<b><u>Section</u></b>	<b><u>Page</u></b>
<b>1. Electrical Specifications</b> .....	<b>4</b>
<b>2. Functional Description</b> .....	<b>11</b>
<b>3. Controller Interface</b> .....	<b>12</b>
3.1. Serial Peripheral Interface (SPI) .....	12
3.2. Fast Response Registers .....	14
3.3. Operating Modes and Timing .....	14
3.4. Application Programming Interface .....	18
3.5. Interrupts .....	18
3.6. GPIO .....	18
<b>4. Modulation and Hardware Configuration Options</b> .....	<b>19</b>
4.1. Modulation Types .....	19
4.2. Hardware Configuration Options .....	19
4.3. Preamble Length .....	20
<b>5. Internal Functional Blocks</b> .....	<b>22</b>
5.1. RX Chain .....	22
5.2. RX Modem .....	22
5.3. Synthesizer .....	24
5.4. Transmitter (TX) .....	26
5.5. Crystal Oscillator .....	29
<b>6. Data Handling and Packet Handler</b> .....	<b>30</b>
6.1. RX and TX FIFOs .....	30
6.2. Packet Handler .....	30
<b>7. RX Modem Configuration</b> .....	<b>32</b>
<b>8. Auxiliary Blocks</b> .....	<b>32</b>
8.1. Wake-up Timer and 32 kHz Clock Source .....	32
8.2. Low Duty Cycle Mode (Auto RX Wake-Up) .....	32
8.3. Temperature, Battery Voltage, and Auxiliary ADC .....	34
8.4. Low Battery Detector .....	34
8.5. Antenna Diversity .....	34
<b>9. Pin Descriptions: Si4438</b> .....	<b>35</b>
<b>10. Ordering Information</b> .....	<b>37</b>
<b>11. Package Outline: Si4438</b> .....	<b>38</b>
<b>12. PCB Land Pattern: Si4438</b> .....	<b>40</b>
<b>13. Top Marking</b> .....	<b>42</b>
13.1. Si4438 Top Marking .....	42
13.2. Top Marking Explanation .....	42
<b>Document Change List</b> .....	<b>43</b>
<b>Contact Information</b> .....	<b>44</b>

## 1. Electrical Specifications

**Table 1. DC Characteristics\***

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
Supply Voltage Range	$V_{DD}$		1.8	3.3	3.6	V
Power Saving Modes	$I_{Shutdown}$	RC Oscillator, Main Digital Regulator, and Low Power Digital Regulator OFF	—	30	—	nA
	$I_{Standby}$	Register values maintained and RC oscillator/WUT OFF	—	50	—	nA
	$I_{SleepRC}$	RC Oscillator/WUT ON and all register values maintained, and all other blocks OFF	—	900	—	nA
	$I_{SleepXO}$	Sleep current using an external 32 kHz crystal.	—	1.7	—	$\mu$ A
	$I_{Sensor-LBD}$	Low battery detector ON, register values maintained, and all other blocks OFF	—	1	—	$\mu$ A
	$I_{Ready}$	Crystal Oscillator and Main Digital Regulator ON, all other blocks OFF	—	1.8	—	mA
TUNE Mode Current	$I_{Tune\_RX}$	RX Tune	—	7.2	—	mA
	$I_{Tune\_TX}$	TX Tune	—	8	—	mA
RX Mode Current	$I_{RXH}$		—	14	—	mA
TX Mode Current (Si4438)	$I_{TX\_+20}$	+20 dBm output power, class-E match, 490 MHz, 3.3 V	—	75	—	mA

**\*Note:** All minimum and maximum values are guaranteed by production test unless otherwise stated. All typical values apply at  $V_{DD} = 3.3$  V and 25 °C unless otherwise stated.

Table 2. Synthesizer AC Electrical Characteristics<sup>1</sup>

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
Synthesizer Frequency Range (Si4438)	$F_{\text{SYN}}$		425	—	525	MHz
Synthesizer Frequency Resolution <sup>2</sup>	$F_{\text{RES-525}}$	425–525 MHz	—	14.3	—	Hz
Synthesizer Settling Time	$t_{\text{LOCK}}$	Measured from exiting Ready mode with XOSC running to any frequency. Including VCO Calibration.	—	50	—	$\mu\text{s}$
Phase Noise	$L\phi(f_M)$	$\Delta F = 10 \text{ kHz}, 460 \text{ MHz}$	—	-106	—	dBc/Hz
		$\Delta F = 100 \text{ kHz}, 460 \text{ MHz}$	—	-110	—	dBc/Hz
		$\Delta F = 1 \text{ MHz}, 460 \text{ MHz}$	—	-123	—	dBc/Hz
		$\Delta F = 10 \text{ MHz}, 460 \text{ MHz}$	—	-130	—	dBc/Hz

**Notes:**

- All minimum and maximum values are guaranteed by production test unless otherwise stated. All typical values apply at  $V_{\text{DD}} = 3.3 \text{ V}$  and  $25 \text{ }^\circ\text{C}$  unless otherwise stated.
- Default API setting for modulation deviation resolution is double the typical value specified.

**Table 3. Receiver AC Electrical Characteristics<sup>1</sup>**

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
RX Frequency Range (Si4438)	$F_{RX}$		425	—	525	MHz
RX Sensitivity	$P_{RX\_0.5}$	(BER < 0.1%) (500 bps, GFSK, BT = 0.5, $\Delta f = \pm 250\text{Hz}$ ) <sup>2</sup>	—	-124	—	dBm
	$P_{RX\_40}$	(BER < 0.1%) (40 kbps, GFSK, BT = 0.5, $\Delta f = \pm 20\text{ kHz}$ ) <sup>2</sup>	—	-108	—	dBm
	$P_{RX\_100}$	(BER < 0.1%) (100 kbps, GFSK, BT = 0.5, $\Delta f = \pm 50\text{ kHz}$ ) <sup>1</sup>	—	-103	—	dBm
	$P_{RX\_9.6}$	(BER < 0.1%) (9.6 kbps, GFSK, BT = 0.5, $\Delta f = \pm 4.8\text{ kHz}$ ) <sup>2</sup>	—	-114	—	dBm
	$P_{RX\_OOK}$	(BER < 0.1%, 4.8 kbps, 350 kHz BW, OOK, PN15 data) <sup>2</sup>	—	-108	—	dBm
		(BER < 0.1%, 40 kbps, 350 kHz BW, OOK, PN15 data) <sup>2</sup>	—	-102	—	dBm
		(BER < 0.1%, 120 kbps, 350 kHz BW, OOK, PN15 data) <sup>2</sup>	—	-98	—	dBm
RX Channel Bandwidth <sup>4</sup>	BW		1.1	—	850	kHz
BER Variation vs Power Level <sup>2</sup>	$P_{RX\_RES}$	Up to +5 dBm Input Level	—	0	0.1	ppm
RSSI Resolution	$RES_{RSSI}$		—	$\pm 0.5$	—	dB
$\pm 1$ -Ch Offset Selectivity, 450 MHz <sup>2</sup>	$C/I_{1-CH}$	Desired Ref Signal 3 dB above sensitivity, BER < 0.1%. Interferer is CW, and desired is modulated with 2.4 kbps $\Delta F = 1.2\text{ kHz}$ GFSK with BT = 0.5, RX channel BW = 4.8 kHz, channel spacing = 12.5 kHz	—	-58	—	dB
Blocking 1 MHz Offset <sup>2</sup>	$1M_{BLOCK}$	Desired Ref Signal 3 dB above sensitivity, BER = 0.1%. Interferer is CW, and desired is modulated with 2.4 kbps, $\Delta F = 1.2\text{ kHz}$ GFSK with BT = 0.5, RX channel BW = 4.8 kHz	—	-75	—	dB
Blocking 8 MHz Offset <sup>2</sup>	$8M_{BLOCK}$		—	-84	—	dB
Image Rejection <sup>2</sup>	$Im_{REJ}$	Rejection at the image frequency. IF = 468 kHz	—	35	—	dB

**Notes:**

1. All minimum and maximum values are guaranteed by production test unless otherwise stated. All typical values apply at VDD = 3.3 V and 25 °C unless otherwise stated.
2. BER is measured in the 450–470 MHz frequency band on RF evaluation boards.
3. For PER tests, 48 preamble symbols, 4 byte sync word, 10 byte payload and CRC-32 was used.
4. Guaranteed by bench characterization.

Table 4. Transmitter AC Electrical Characteristics<sup>1</sup>

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
TX Frequency Range	$F_{TX}$		425	—	525	MHz
(G)FSK Data Rate <sup>2,3</sup>	$DR_{FSK}$		0.1	—	500	kbps
OOK Data Rate <sup>2,3</sup>	$DR_{OOK}$		0.1	—	120	kbps
Modulation Deviation Range	$\Delta f_{525}$	425–525 MHz	—	750	—	kHz
Modulation Deviation Resolution <sup>4</sup>	$F_{RES-525}$	425–525 MHz	—	14.3	—	Hz
Output Power Range <sup>5</sup>	$P_{TX}$		–20	—	+20	dBm
TX RF Output Steps	$\Delta P_{RF\_OUT}$	Using switched current match within 6 dB of max power	—	0.1	—	dB
TX RF Output Level Variation vs. Temperature	$\Delta P_{RF\_TEMP}$	–40 to +85 °C	—	1	—	dB
TX RF Output Level Variation vs. Frequency	$\Delta P_{RF\_FREQ}$		—	0.5	—	dB
Transmit Modulation Filtering	$B*T$	Gaussian Filtering Bandwidth Time Product	—	0.5	—	

**Notes:**

1. All minimum and maximum values are guaranteed by production test unless otherwise stated. All typical values apply at  $V_{DD} = 3.3\text{ V}$  and  $25\text{ °C}$  unless otherwise stated.
2. Guaranteed by design.
3. The maximum data rate is dependent on the XTAL frequency and is calculated as per the formula:  
Maximum Symbol Rate =  $F_{xtal}/60$ , where  $F_{xtal}$  is the XTAL frequency (typically 30 MHz).
4. Default API setting for modulation deviation resolution is double the typical value specified.
5. Output power is dependent on matching components and board layout.



**Table 5. Auxiliary Block Specifications<sup>1</sup>**

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
Temperature Sensor Sensitivity	TS <sub>S</sub>		—	4.5	—	ADC Codes/ °C
Low Battery Detector Resolution	LBD <sub>RES</sub>		—	50	—	mV
Microcontroller Clock Output Frequency Range <sup>2</sup>	F <sub>MC</sub>	Configurable to Fxtal or Fxtal divided by 2, 3, 7.5, 10, 15, or 30 where Fxtal is the reference XTAL frequency. In addition, 32.768 kHz is also supported.	32.768K	—	Fxtal	Hz
Temperature Sensor Conversion	TEMP <sub>CT</sub>	Programmable setting	—	3	—	ms
XTAL Range <sup>3</sup>	XTAL <sub>Range</sub>		25	—	32	MHz
30 MHz XTAL Start-Up Time	t <sub>30M</sub>	Using XTAL and board layout in reference design. Start-up time will vary with XTAL type and board layout.	—	250	—	µs
30 MHz XTAL Cap Resolution	30M <sub>RES</sub>		—	70	—	fF
32 kHz XTAL Start-Up Time	t <sub>32k</sub>		—	2	—	sec
32 kHz Accuracy using Internal RC Oscillator	32KRC <sub>RES</sub>		—	2500	—	ppm
POR Reset Time	t <sub>POR</sub>		—	—	5	ms

**Notes:**

1. All minimum and maximum values are guaranteed by production test unless otherwise stated. All typical values apply at VDD = 3.3 V and 25 °C unless otherwise stated.
2. Microcontroller clock frequency tested in production at 1 MHz, 30 MHz and 32.768 kHz. Other frequencies tested in bench characterization.
3. XTAL Range tested in production using an external clock source (similar to using a TCXO).

Table 6. Digital IO Specifications (GPIO\_x, SCLK, SDO, SDI, nSEL, nIRQ, SDN)<sup>1</sup>

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
Rise Time <sup>2,3</sup>	$T_{RISE}$	$0.1 \times V_{DD}$ to $0.9 \times V_{DD}$ , $C_L = 10$ pF, DRV<1:0> = LL	—	2.3	—	ns
Fall Time <sup>3,4</sup>	$T_{FALL}$	$0.9 \times V_{DD}$ to $0.1 \times V_{DD}$ , $C_L = 10$ pF, DRV<1:0> = LL	—	2	—	ns
Input Capacitance	$C_{IN}$		—	2	—	pF
Logic High Level Input Voltage	$V_{IH}$		$V_{DD} \times 0.7$	—	—	V
Logic Low Level Input Voltage	$V_{IL}$		—	—	$V_{DD} \times 0.3$	V
Input Current	$I_{IN}$	$0 < V_{IN} < V_{DD}$	-10	—	10	$\mu$ A
Input Current If Pullup is Activated	$I_{INP}$	$V_{IL} = 0$ V	1	—	10	$\mu$ A
Drive Strength for Output Low Level	$I_{OmaxLL}$	DRV[1:0] = LL <sup>3</sup>	—	6.66	—	mA
	$I_{OmaxLH}$	DRV[1:0] = LH <sup>3</sup>	—	5.03	—	mA
	$I_{OmaxHL}$	DRV[1:0] = HL <sup>3</sup>	—	3.16	—	mA
	$I_{OmaxHH}$	DRV[1:0] = HH <sup>3</sup>	—	1.13	—	mA
Drive Strength for Output High Level	$I_{OmaxLL}$	DRV[1:0] = LL <sup>3</sup>	—	5.75	—	mA
	$I_{OmaxLH}$	DRV[1:0] = LH <sup>3</sup>	—	4.37	—	mA
	$I_{OmaxHL}$	DRV[1:0] = HL <sup>3</sup>	—	2.73	—	mA
	$I_{OmaxHH}$	DRV[1:0] = HH <sup>3</sup>	—	0.96	—	mA
Drive Strength for Output High Level for GPIO0	$I_{OmaxLL}$	DRV[1:0] = LL <sup>3</sup>	—	2.53	—	mA
	$I_{OmaxLH}$	DRV[1:0] = LH <sup>3</sup>	—	2.21	—	mA
	$I_{OmaxHL}$	DRV[1:0] = HL <sup>3</sup>	—	1.7	—	mA
	$I_{OmaxHH}$	DRV[1:0] = HH <sup>3</sup>	—	0.80	—	mA
Logic High Level Output Voltage	$V_{OH}$	DRV[1:0] = HL	$V_{DD} \times 0.8$	—	—	V
Logic Low Level Output Voltage	$V_{OL}$	DRV[1:0] = HL	—	—	$V_{DD} \times 0.2$	V

**Notes:**

1. All minimum and maximum values are guaranteed by production test unless otherwise stated. All typical values apply at  $V_{DD} = 3.3$  V and  $25$  °C unless otherwise stated.
2. 8 ns is typical for GPIO0 rise time.
3. Assuming  $V_{DD} = 3.3$  V, drive strength is specified at  $V_{oh}(\min) = 2.64$  V and  $V_{ol}(\max) = 0.66$  V at room temperature.
4. 2.4 ns is typical for GPIO0 fall time.

**Table 7. Thermal Operating Characteristics**

Parameter	Value	Unit
Operating Ambient Temperature Range $T_A$	-40 to +85	°C
Thermal Impedance $\theta_{JA}$	30	°C/W
Junction Temperature $T_{JMAX}$	+125	°C
Storage Temperature Range $T_{STG}$	-55 to +125	°C

**Table 8. Absolute Maximum Ratings\***

Parameter	Value	Unit
$V_{DD}$ to GND	-0.3, +3.6	V
Instantaneous $V_{RF-peak}$ to GND on TX Output Pin	-0.3, +8.0	V
Sustained $V_{RF-peak}$ to GND on TX Output Pin	-0.3, +6.5	V
Voltage on Digital Control Inputs	-0.3, $V_{DD} + 0.3$	V
Voltage on Analog Inputs	-0.3, $V_{DD} + 0.3$	V
Voltage on XIN Input when using a TCXO	-0.7, $V_{DD} + 0.3$	V
RX Input Power	+10	dBm
Operating Ambient Temperature Range $T_A$	-40 to +85	°C
Thermal Impedance $\theta_{JA}$	30	°C/W
Junction Temperature $T_J$	+125	°C
Storage Temperature Range $T_{STG}$	-55 to +125	°C
<p><b>*Note:</b> Stresses beyond those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. These are stress ratings only and functional operation of the device at or beyond these ratings in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability. Power Amplifier may be damaged if switched on without proper load or termination connected. TX matching network design will influence TX <math>V_{RF-peak}</math> on TX output pin. Caution: ESD sensitive device.</p>		

## 2. Functional Description

The Si4438 devices are high-performance, low-current, wireless ISM transceivers that cover the sub-GHz bands. The wide operating voltage range of 1.8–3.6 V and low current consumption make the Si4438 an ideal solution for battery powered applications. The Si4438 operates as a time division duplexing (TDD) transceiver where the device alternately transmits and receives data packets. The device uses a single-conversion mixer to downconvert the 2-level FSK/GFSK or OOK modulated receive signal to a low IF frequency. Following a programmable gain amplifier (PGA) the signal is converted to the digital domain by a high performance  $\Delta\Sigma$  ADC allowing filtering, demodulation, slicing, and packet handling to be performed in the built-in DSP increasing the receiver's performance and flexibility versus analog based architectures. The demodulated signal is output to the system MCU through a programmable GPIO or via the standard SPI bus by reading the 64-byte RX FIFO.

A single high precision local oscillator (LO) is used for both transmit and receive modes since the transmitter and receiver do not operate at the same time. The LO is generated by an integrated VCO and  $\Delta\Sigma$  Fractional-N PLL synthesizer. The synthesizer is designed to support configurable data rates from 100 bps to 500 kbps. The transmit FSK data is modulated directly into the  $\Delta\Sigma$  data stream and can be shaped by a Gaussian low-pass filter to reduce unwanted spectral content.

The Si4438 contains a power amplifier (PA) that supports output power up to +20 dBm with very high efficiency, consuming only 75 mA. The integrated +20 dBm power amplifier can also be used to compensate for the reduced performance of a lower cost, lower performance antenna or antenna with size constraints due to a small form-factor. Competing solutions require large and expensive external PAs to achieve comparable performance. The PA is single-ended to allow for easy antenna matching and low BOM cost. The PA incorporates automatic ramp-up and ramp-down control to reduce unwanted spectral spreading. The Si4438 family supports TX/RX switch control, and antenna diversity switch control to extend the link range and improve performance. Built-in antenna diversity can be used to further extend range and enhance performance. Antenna diversity is completely integrated into the Si4438 and can improve the system link budget by 8–10 dB, resulting in substantial range increases under adverse environmental conditions. A highly configurable packet handler allows for autonomous encoding/decoding of nearly any packet structure. Additional system features, such as an automatic wake-up timer, low battery detector, 64 byte TX/RX FIFOs, and preamble detection, reduce overall current consumption and allows for the use of lower-cost system MCUs. An integrated temperature sensor, power-on-reset (POR), and GPIOs further reduce overall system cost and size. The Si4438 is designed to work with an MCU, crystal, and a few passive components to create a very low-cost system.

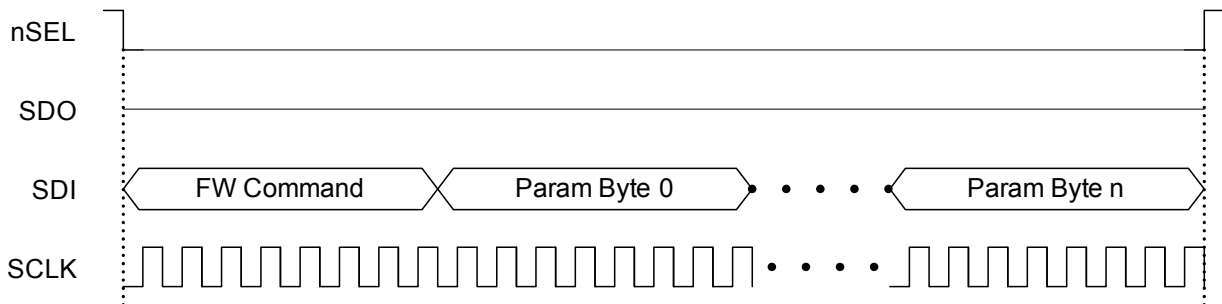
## 3. Controller Interface

### 3.1. Serial Peripheral Interface (SPI)

The Si4438 communicates with the host MCU over a standard 4-wire serial peripheral interface (SPI): SCLK, SDI, SDO, and nSEL. The SPI interface is designed to operate at a maximum of 10 MHz. The SPI timing parameters are demonstrated in Table 9. The host MCU writes data over the SDI pin and can read data from the device on the SDO output pin. Figure 1 demonstrates an SPI write command. The nSEL pin should go low to initiate the SPI command. The first byte of SDI data will be one of the firmware commands followed by n bytes of parameter data which will be variable depending on the specific command. The rising edges of SCLK should be aligned with the center of the SDI data.

**Table 9. Serial Interface Timing Parameters**

Symbol	Parameter	Min (ns)	Max (ns)	Diagram
$t_{CH}$	Clock high time	40		
$t_{CL}$	Clock low time	40		
$t_{DS}$	Data setup time	20		
$t_{DH}$	Data hold time	20		
$t_{DD}$	Output data delay time		43	
$t_{DE}$	Output disable time		45	
$t_{SS}$	Select setup time	20		
$t_{SH}$	Select hold time	50		
$t_{SW}$	Select high period	80		
<p><b>*Note:</b> CL = 10 pF; VDD = 1.8 V; SDO Drive strength setting = 10.</p>				



**Figure 1. SPI Write Command**

The Si4438 contains an internal MCU which controls all the internal functions of the radio. For SPI read commands a typical MCU flow of checking clear-to-send (CTS) is used to make sure the internal MCU has executed the command and prepared the data to be output over the SDO pin. Figure 2 demonstrates the general flow of an SPI read command. Once the CTS value reads FFh then the read data is ready to be clocked out to the host MCU. The typical time for a valid FFh CTS reading is 20  $\mu$ s. Figure 3 demonstrates the remaining read cycle after CTS is set to FFh. The internal MCU will clock out the SDO data on the negative edge so the host MCU should process the SDO data on the rising edge of SCLK.

### Firmware Flow

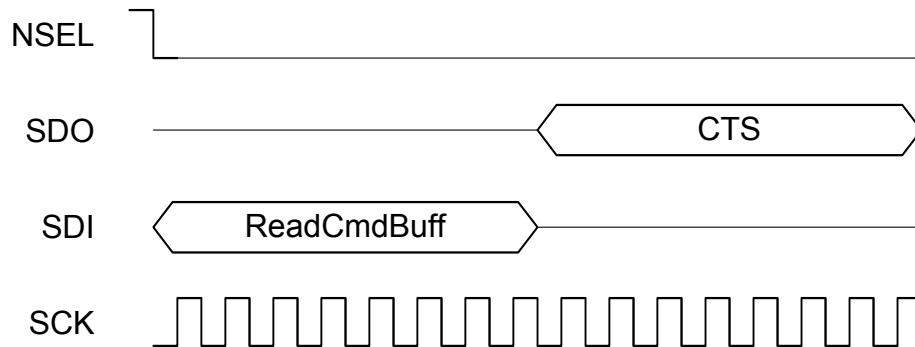
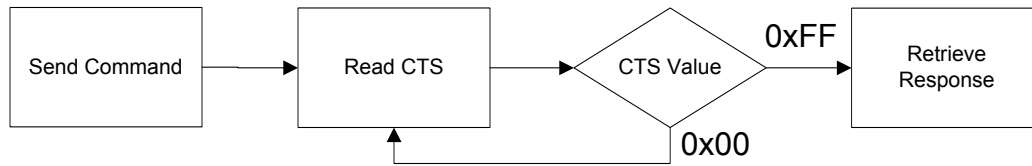


Figure 2. SPI Read Command—Check CTS Value

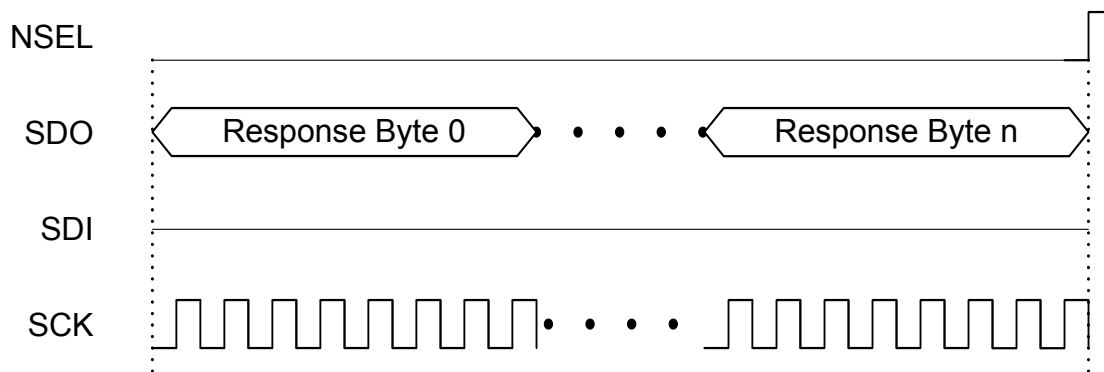


Figure 3. SPI Read Command—Clock Out Read Data

## 3.2. Fast Response Registers

The fast response registers are registers that can be read immediately without the requirement to monitor and check CTS. There are four fast response registers that can be programmed for a specific function. The fast response registers can be read through API commands, 0x50 for Fast Response A, 0x51 for Fast Response B, 0x53 for Fast Response C, and 0x57 for Fast Response D. The fast response registers can be configured by the “FRR\_CTL\_X\_MODE” properties.

The fast response registers may be read in a burst fashion. After the initial 16 clock cycles, each additional eight clock cycles will clock out the contents of the next fast response register in a circular fashion. The value of the FRRs will not be updated unless NSEL is toggled.

## 3.3. Operating Modes and Timing

The primary states of the Si4438 are shown in Figure 4. The shutdown state completely shuts down the radio to minimize current consumption. Standby/Sleep, SPI Active, Ready, TX Tune, and RX tune are available to optimize the current consumption and response time to RX/TX for a given application. API commands START\_RX, START\_TX, and CHANGE\_STATE control the operating state with the exception of shutdown which is controlled by SDN, pin 1. Table 10 shows each of the operating modes with the time required to reach either RX or TX mode as well as the current consumption of each mode. The times in Table 9 are measured from the rising edge of nSEL until the chip is in the desired state. Note that these times are indicative of state transition timing but are not guaranteed and should only be used as a reference data point. An automatic sequencer will put the chip into RX or TX from any state. It is not necessary to manually step through the states. To simplify the diagram it is not shown but any of the lower power states can be returned to automatically after RX or TX.

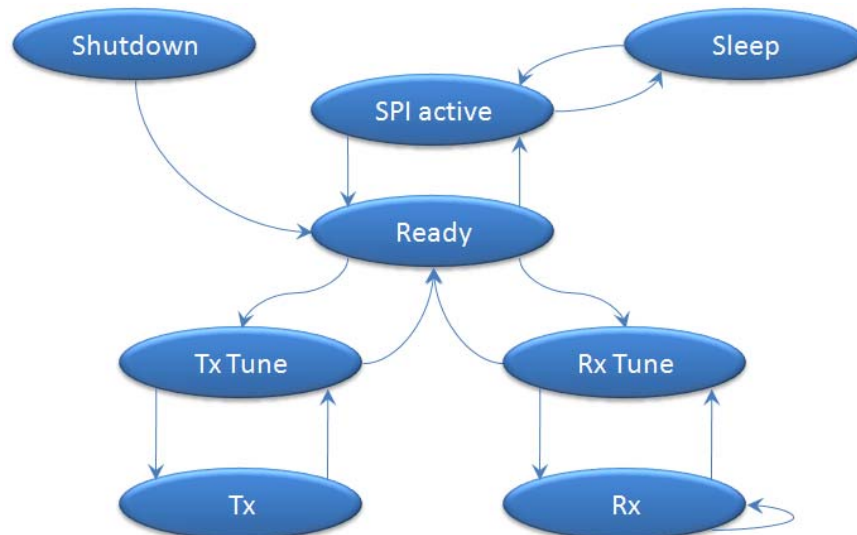


Figure 4. State Machine Diagram

Table 10. Operating State Response Time and Current Consumption

State/Mode	Response Time to		Current in State /Mode
	TX	RX	
Shutdown State	15 ms	15 ms	30 nA
Standby State	440 $\mu$ s	440 $\mu$ s	50 nA
Sleep State	440 $\mu$ s	440 $\mu$ s	900 nA
SPI Active State	340 $\mu$ s	340 $\mu$ s	1.35 mA
Ready State	126 $\mu$ s	122 $\mu$ s	1.8 mA
TX Tune State	58 $\mu$ s	—	8 mA
RX Tune State	—	74 $\mu$ s	7.2 mA
TX State	—	138 $\mu$ s	75 mA @ +20 dBm
RX State	130 $\mu$ s	75 $\mu$ s	14 mA

Figure 5 shows the POR timing and voltage requirements. The power consumption (battery life) depends on the duty cycle of the application or how often the part is in either Rx or Tx state. In most applications the utilization of the standby state will be most advantageous for battery life but for very low duty cycle applications shutdown will have an advantage. For the fastest timing the next state can be selected in the START\_RX or START\_TX API commands to minimize SPI transactions and internal MCU processing.

### 3.3.1. Power on Reset (POR)

A Power On Reset (POR) sequence is used to boot the device up from a fully off or shutdown state. To execute this process, VDD must ramp within 1ms and must remain applied to the device for at least 10ms. If VDD is removed, then it must stay below 0.15V for at least 10ms before being applied again. Please see Figure x and Table x for details.

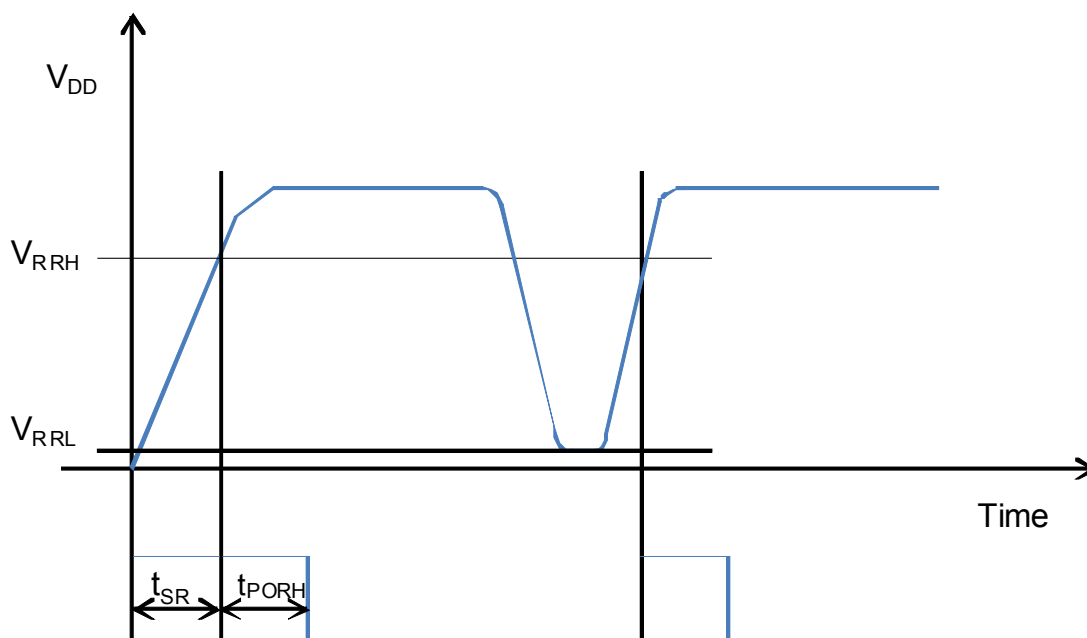


Figure 5. POR Timing Diagram



Table 11. POR Timing

Variable	Description	Min	Typ	Max	Units
$t_{PORH}$	High time for VDD to fully settle POR circuit	10			ms
$t_{PORL}$	Low time for VDD to enable POR	10			ms
$V_{RRH}$	Voltage for successful POR	90%*Vdd			V
$V_{RRL}$	Starting Voltage for successful POR	0		150	mV
$t_{SR}$	Slew rate of VDD for successful POR			1	ms

### 3.3.2. Shutdown State

The shutdown state is the lowest current consumption state of the device with nominally less than 30 nA of current consumption. The shutdown state may be entered by driving the SDN pin (Pin 1) high. The SDN pin should be held low in all states except the shutdown state. In the shutdown state, the contents of the registers are lost and there is no SPI access. When coming out of the shutdown state a power on reset (POR) will be initiated along with the internal calibrations. After the POR the POWER\_UP command is required to initialize the radio. The SDN pin needs to be held high for at least 10us before driving low again so that internal capacitors can discharge. Not holding the SDN high for this period of time may cause the POR to be missed and the device to boot up incorrectly. If POR timing and voltage requirements cannot be met, it is highly recommended that SDN be controlled using the host processor rather than tying it to GND on the board.

### 3.3.3. Standby State

Standby state has the lowest current consumption with the exception of shutdown but has much faster response time to RX or TX mode. In most cases standby should be used as the low power state. In this state the register values are maintained with all other blocks disabled. The SPI is accessible during this mode but any SPI event, including FIFO R/W, will enable an internal boot oscillator and automatically move the part to SPI active state. After an SPI event the host will need to re-command the device back to standby through the “Change State” API command to achieve the 50 nA current consumption. If an interrupt has occurred (i.e., the nIRQ pin = 0) the interrupt registers must be read to achieve the minimum current consumption of this mode.

### 3.3.4. Sleep State

Sleep state is the same as standby state but the wake-up-timer and a 32 kHz clock source are enabled. The source of the 32 kHz clock can either be an internal 32 kHz RC oscillator which is periodically calibrated or a 32 kHz oscillator using an external XTAL. The SPI is accessible during this mode but an SPI event will enable an internal boot oscillator and automatically move the part to SPI active mode. After an SPI event the host will need to re-command the device back to sleep. If an interrupt has occurred (i.e., the nIRQ pin = 0) the interrupt registers must be read to achieve the minimum current consumption of this mode.

### 3.3.5. SPI Active State

In SPI active state the SPI and a boot up oscillator are enabled. After SPI transactions during either standby or sleep the device will not automatically return to these states. A “Change State” API command will be required to return to either the standby or sleep modes.

### 3.3.6. Ready State

Ready state is designed to give a fast transition time to TX or RX state with reasonable current consumption. In this mode the Crystal oscillator remains enabled reducing the time required to switch to TX or RX mode by eliminating the crystal start-up time.

### 3.3.7. TX State

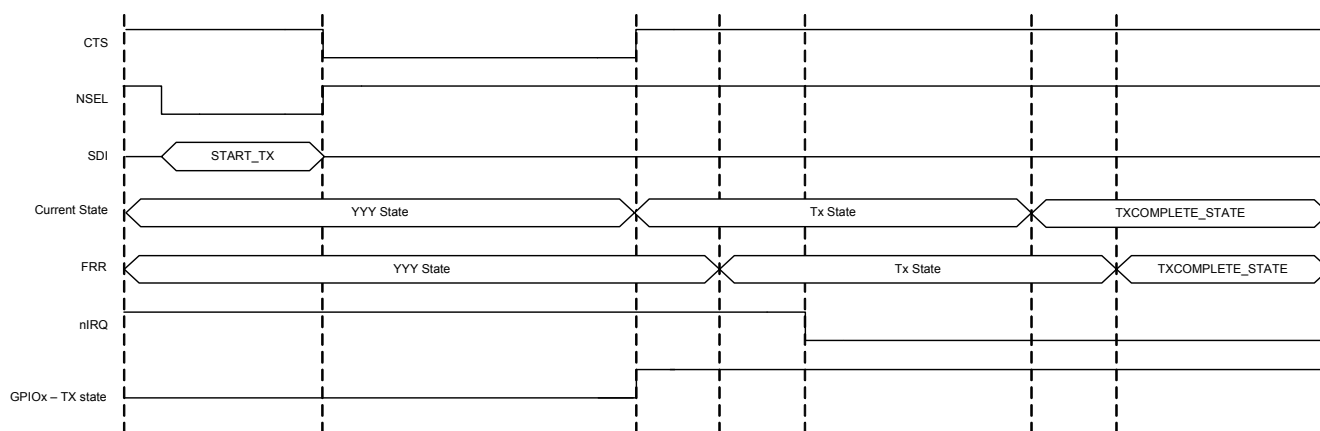
The TX state may be entered from any of the state with the “Start TX” or “Change State” API commands. A built-in sequencer takes care of all the actions required to transition between states from enabling the crystal oscillator to ramping up the PA. The following sequence of events will occur automatically when going from standby to TX state.

1. Enable internal LDOs.
2. Start up crystal oscillator and wait until ready (controlled by an internal timer).

3. Enable PLL.
4. Calibrate VCO/PLL.
5. Wait until PLL settles to required transmit frequency (controlled by an internal timer).
6. Activate power amplifier and wait until power ramping is completed (controlled by an internal timer).
7. Transmit packet.

Steps in this sequence may be eliminated depending on which state the chip is configured to prior to commanding to TX. By default, the VCO and PLL are calibrated every time the PLL is enabled. When the START\_TX API command is utilized the next state may be defined to ensure optimal timing and turnaround.

Figure 6 shows an example of the commands and timing for the START\_TX command. CTS will go high as soon as the sequencer puts the part into TX state. As the sequencer is stepping through the events listed above, CTS will be low and no new commands or property changes are allowed. If the Fast Response (FRR) or nIRQ is used to monitor the current state there will be slight delay caused by the internal hardware from when the event actually occurs to when the transition occurs on the FRR or nIRQ. The time from entering TX state to when the FRR will update is 5  $\mu$ s and the time to when the nIRQ will transition is 13  $\mu$ s. If a GPIO is programmed for TX state or used as control for a transmit/receive switch (TR switch) there is no delay.



**Figure 6. Start\_TX Commands and Timing**

### 3.3.8. RX State

The RX state may be entered from any of the other states by using the “Start RX” or “Change State” API command. A built-in sequencer takes care of all the actions required to transition between states. The following sequence of events will occur automatically to get the chip into RX mode when going from standby to RX state:

1. Enable the digital LDO and the analog LDOs.
2. Start up crystal oscillator and wait until ready (controlled by an internal timer).
3. Enable PLL.
4. Calibrate VCO
5. Wait until PLL settles to required receive frequency (controlled by an internal timer).
6. Enable receiver circuits: LNA, mixers, and ADC.
7. Enable receive mode in the digital modem.

Depending on the configuration of the radio, all or some of the following functions will be performed automatically by the digital modem: AGC, AFC (optional), update status registers, bit synchronization, packet handling (optional) including sync word, header check, and CRC. Similar to the TX state, the next state after RX may be defined in the “Start RX” API command. The START\_RX commands and timing will be equivalent to the timing shown in Figure 6.

## 3.4. Application Programming Interface

The host MCU communicates with an application programming interface (API) embedded inside the device. The API is divided into two sections, commands and properties. The commands are used to control the chip and retrieve its status. The properties are general configurations which will change infrequently. For API description details, refer to the EZRadioPRO API Documentation.zip file available on [www.silabs.com](http://www.silabs.com).

## 3.5. Interrupts

The Si4438 is capable of generating an interrupt signal when certain events occur. The chip notifies the microcontroller that an interrupt event has occurred by setting the nIRQ output pin LOW = 0. This interrupt signal will be generated when any one (or more) of the interrupt events (corresponding to the Interrupt Status bits) occur. The nIRQ pin will remain low until the microcontroller clears all the interrupts. The nIRQ output signal will then be reset until the next change in status is detected.

The interrupts sources are grouped into three groups: packet handler, chip status, and modem. The individual interrupts in these groups can be enabled/disabled in the interrupt property registers. An interrupt must be enabled for it to trigger an event on the nIRQ pin. The interrupt group must be enabled as well as the individual interrupts in API properties described in the API documentation. Once an interrupt event occurs and the nIRQ pin is low there are two ways to read and clear the interrupts. All of the interrupts may be read and cleared in the "GET\_INT\_STATUS" API command. By default all interrupts will be cleared once read. If only specific interrupts want to be read in the fastest possible method the individual interrupt groups (Packet Handler, Chip Status, Modem) may be read and cleared by the "GET\_MODEM\_STATUS", "GET\_PH\_STATUS" (packet handler), and "GET\_CHIP\_STATUS" API commands. The instantaneous status of a specific function maybe read if the specific interrupt is enabled or disabled. The status results are provided after the interrupts and can be read with the same commands as the interrupts. The status bits will give the current state of the function whether the interrupt is enabled or not. The fast response registers can also give information about the interrupt groups but reading the fast response registers will not clear the interrupt and reset the nIRQ pin.

## 3.6. GPIO

Four general purpose IO pins are available to utilize in the application. The GPIO are configured by the GPIO\_PIN\_CFG command in address 13h. For a complete list of the GPIO options please see the API guide. GPIO pins 0 and 1 should be used for active signals such as data or clock. GPIO pins 2 and 3 have more susceptibility to generating spurious in the synthesizer than pins 0 and 1. The drive strength of the GPIOs can be adjusted with the GEN\_CONFIG parameter in the GPIO\_PIN\_CFG command. By default the drive strength is set to minimum. The default configuration for the GPIOs and the state during SDN is shown below in Table 12. The state of the IO during shutdown is also shown in Table 12. As indicated previously in Table 6, GPIO 0 has lower drive strength than the other GPIOs.

Table 12. GPIOs

Pin	SDN State	POR Default
GPIO0	0	POR
GPIO1	0	CTS
GPIO2	0	POR
GPIO3	0	POR
nIRQ	resistive VDD pull-up	nIRQ
SDO	resistive VDD pull-up	SDO
SDI	High Z	SDI
SCLK	High Z	SCLK
NSEL	High Z	NSEL

## 4. Modulation and Hardware Configuration Options

The Si4438 supports three different modulation options and can be used in various configurations to tailor the device to any specific application or legacy system for drop in replacement. The modulation and configuration options are set in property, MODEM\_MOD\_TYPE. Refer to the EZRadioPRO API Documentation.zip file available on [www.silabs.com](http://www.silabs.com) for details.

### 4.1. Modulation Types

The Si4438 supports five different modulation options: Gaussian frequency shift keying (GFSK), frequency-shift keying (FSK), on-off keying (OOK). Minimum shift keying (MSK) can also be created by using GFSK settings. GFSK is the recommended modulation type as it provides the best performance and cleanest modulation spectrum. The modulation type is set by the "MOD\_TYPE[2:0]" registers in the "MODEM\_MOD\_TYPE" API property. A continuous-wave (CW) carrier may also be selected for RF evaluation purposes. The modulation source may also be selected to be a pseudo-random source for evaluation purposes.

### 4.2. Hardware Configuration Options

There are different receive demodulator options to optimize the performance and mutually-exclusive options for how the RX/TX data is transferred from the host MCU to the RF device.

#### 4.2.1. Receive Demodulator Options

There are multiple demodulators integrated into the device to optimize the performance for different applications, modulation formats, and packet structures. The calculator built into WDS will choose the optimal demodulator based on the input criteria.

##### 4.2.1.1. Synchronous Demodulator

The synchronous demodulator's internal frequency error estimator acquires the frequency error based on a 101010 preamble structure. The bit clock recovery circuit locks to the incoming data stream within four transactions of a "10" or "01" bit stream. The synchronous demodulator gives optimal performance for 2-level FSK or GFSK modulation that has a modulation index less than 2.

##### 4.2.1.2. Asynchronous Demodulator

The asynchronous demodulator should be used OOK modulation and for FSK/GFSK under one or more of the following conditions:

- Modulation index  $\geq 2$
- Non-standard preamble (not 1010101... pattern)

When the modulation index exceeds 2, the asynchronous demodulator has better sensitivity compared to the synchronous demodulator. An internal deglitch circuit provides a glitch-free data output and a data clock signal to simplify the interface to the host. There is no requirement to perform deglitching in the host MCU. The asynchronous demodulator will typically be utilized for legacy systems and will have many performance benefits over devices used in legacy designs. Unlike the Si4432/31 solution for non-standard packet structures, there is no requirement to perform deglitching on the data in the host MCU. Glitch-free data is output from Si4438 devices, and a sample clock for the asynchronous data can also be supplied to the host MCU; so, oversampling or bit clock recovery is not required by the host MCU. There are multiple detector options in the asynchronous demodulator block, which will be selected based upon the options entered into the WDS calculator. The asynchronous demodulator's internal frequency error estimator is able to acquire the frequency error based on any preamble structure.

#### 4.2.2. RX/TX Data Interface With MCU

There are two different options for transferring the data from the RF device to the host MCU. FIFO mode uses the SPI interface to transfer the data, while direct mode transfers the data in real time over GPIO.

##### 4.2.2.1. FIFO Mode

In FIFO mode, the transmit and receive data is stored in integrated FIFO register memory. The TX FIFO is accessed by writing Command 66h followed directly by the data/clock that the host wants to write into the TX FIFO. The RX FIFO is accessed by writing command 77h followed by the number of clock cycles of data the host would like to read out of the RX FIFO. The RX data will be clocked out onto the SDO pin.

In TX mode, if the packet handler is enabled, the data bytes stored in FIFO memory are “packaged” together with other fields and bytes of information to construct the final transmit packet structure. These other potential fields include the Preamble, Sync word, Header, CRC checksum, etc. The configuration of the packet structure in TX mode is determined by the Automatic Packet Handler (if enabled), in conjunction with a variety of Packet Handler properties. If the Automatic Packet Handler is disabled, the entire desired packet structure should be loaded into FIFO memory; no other fields (such as Preamble or Sync word) will be automatically added to the bytes stored in FIFO memory. For further information on the configuration of the FIFOs for a specific application or packet size, see “6. Data Handling and Packet Handler” on page 30. In RX mode, only the bytes of the received packet structure that are considered to be “data bytes” are stored in FIFO memory. Which bytes of the received packet are considered “data bytes” is determined by the Automatic Packet Handler (if enabled) in conjunction with the Packet Handler configuration. If the Automatic Packet Handler is disabled, all bytes following the Sync word are considered data bytes and are stored in FIFO memory. Thus, even if Automatic Packet Handling operation is not desired, the preamble detection threshold and Sync word still need to be programmed so that the RX Modem knows when to start filling data into the FIFO. When the FIFO is being used in RX mode, all of the received data may still be observed directly (in realtime) by properly programming a GPIO pin as the RXDATA output pin; this can be quite useful during application development. When in FIFO mode, the chip will automatically exit the TX or RX State when either the PACKET\_SENT or PACKET\_RX interrupt occurs. The chip will return to the IDLE state programmed in the argument of the “START TX” or “START RX” API command, TXCOMPLETE\_STATE[3:0] or RXVALID\_STATE[3:0]. For example, the chip may be placed into TX mode by sending the “START TX” command and by writing the 30h to the TXCOMPLETE\_STATE[3:0] argument. The chip will transmit all of the contents of the FIFO, and the ipksent interrupt will occur. When this event occurs, the chip will return to the ready state as defined by TXCOMPLETE\_STATE[3:0] = 30h.

#### 4.2.2.2. Direct Mode

For legacy systems that perform packet handling within the host MCU or other baseband chip, it may not be desirable to use the FIFO. For this scenario, a Direct mode is provided, which bypasses the FIFOs entirely. In TX Direct mode, the TX modulation data is applied to an input pin of the chip and processed in “real time” (i.e., not stored in a register for transmission at a later time). Any of the GPIOs may be configured for use as the TX Data input function. Furthermore, an additional pin may be required for a TX Clock output function if GFSK modulation is desired (only the TX Data input pin is required for FSK). To achieve direct mode, the GPIO must be configured in the “GPIO\_PIN\_CFG” API command as well as the “MODEM\_MOD\_TYPE” API property. For GFSK, “TX\_DIRECT\_MODE\_TYPE” must be set to Synchronous. For 2FSK or OOK, the type can be set to asynchronous or synchronous. The MOD\_SOURCE[1:0] should be set to 01h for all direct mode configurations. In RX Direct mode, the RX Data and RX Clock can be programmed for direct (real-time) output to GPIO pins. The microcontroller may then process the RX data without using the FIFO or packet handler functions of the RFIC.

### 4.3. Preamble Length

The preamble length requirement is only relevant if using the synchronous demodulator. If the asynchronous demodulator is being used, then there is no requirement for a conventional 101010 pattern.

The preamble detection threshold determines the number of valid preamble bits the radio must receive to qualify a valid preamble. The preamble threshold should be adjusted depending on the nature of the application. The required preamble length threshold depends on when receive mode is entered in relation to the start of the transmitted packet and the length of the transmit preamble. With a shorter than recommended preamble detection threshold, the probability of false detection is directly related to how long the receiver operates on noise before the transmit preamble is received. False detection on noise may cause the actual packet to be missed. The preamble detection threshold may be adjusted in the modem calculator by modifying the “PM detection threshold” in the “RX parameters tab” in the radio control panel. For most applications with a preamble length longer than 32 bits, the default value of 20 is recommended for the preamble detection threshold. A shorter Preamble Detection Threshold may be chosen if occasional false detections may be tolerated. When antenna diversity is enabled, a 20-bit preamble detection threshold is recommended. When the receiver is synchronously enabled just before the start of the packet, a shorter preamble detection threshold may be used. Table 13 demonstrates the recommended preamble detection threshold and preamble length for various modes.

Table 13. Recommended Preamble Length

Mode	AFC	Antenna Diversity	Preamble Type	Recommended Preamble Length	Recommended Preamble Detection Threshold
(G)FSK	Disabled	Disabled	Standard	4 Bytes	20 bits
(G)FSK	Enabled	Disabled	Standard	5 Bytes	20 bits
(G)FSK	Disabled	Disabled	Non-standard	2 Bytes	0 bits
(G)FSK	Enabled		Non-standard	Not Supported	
(G)FSK	Disabled	Enabled	Standard	7 Bytes	24 bits
(G)FSK	Enabled	Enabled	Standard	8 Bytes	24 bits
OOK	Disabled	Disabled	Standard	4 Bytes	20 bits
OOK	Disabled	Disabled	Non-standard	2 Bytes	0 bits
OOK	Enabled			Not Supported	

**Notes:**

1. The recommended preamble length and preamble detection thresholds listed above are to achieve 0% PER. They may be shortened when occasional packet errors are tolerable.
2. All recommended preamble lengths and detection thresholds include AGC and BCR settling times.
3. "Standard" preamble type should be set for an alternating data sequence at the max data rate (...10101010...)
4. "Non-standard" preamble type can be set for any preamble type including ...10101010...
5. When preamble detection threshold = 0, sync word needs to be 3 Bytes to avoid false syncs. When only a 2 Byte sync word is available the sync word detection can be extended by including the last preamble Byte into the RX sync word setting.

## 5. Internal Functional Blocks

The following sections provide an overview to the key internal blocks and features.

### 5.1. RX Chain

The internal low-noise amplifier (LNA) is designed to be a wide-band LNA that can be matched with three external discrete components to cover any common range of frequencies in the sub-GHz band. The LNA has extremely low noise to suppress the noise of the following stages and achieve optimal sensitivity; so, no external gain or front-end modules are necessary. The LNA has gain control, which is controlled by the internal automatic gain control (AGC) algorithm. The LNA is followed by an I-Q mixer, filter, programmable gain amplifier (PGA), and ADC. The I-Q mixers downconvert the signal to an intermediate frequency. The PGA then boosts the gain to be within dynamic range of the ADC. The ADC rejects out-of-band blockers and converts the signal to the digital domain where filtering, demodulation, and processing is performed. Peak detectors are integrated at the output of the LNA and PGA for use in the AGC algorithm.

### 5.2. RX Modem

Using high-performance ADCs allows channel filtering, image rejection, and demodulation to be performed in the digital domain, which allows for flexibility in optimizing the device for particular applications. The digital modem performs the following functions:

- Channel selection filter
- TX modulation
- RX demodulation
- Automatic Gain Control (AGC)
- Preamble detection
- Invalid preamble detection
- Radio signal strength indicator (RSSI)
- Automatic frequency compensation (AFC)
- Cyclic redundancy check (CRC)

The digital channel filter and demodulator are optimized for ultra-low-power consumption and are highly configurable. Supported modulation types are GFSK, FSK, GMSK, and OOK. The channel filter can be configured to support bandwidths ranging from 850 down to 1.1 kHz. A large variety of data rates are supported ranging from 100 bps up to 500 kbps. The configurable preamble detector is used with the synchronous demodulator to improve the reliability of the sync-word detection. Preamble detection can be skipped using only sync detection, which is a valuable feature of the asynchronous demodulator when very short preambles are used in protocols, such as MBus. The received signal strength indicator (RSSI) provides a measure of the signal strength received on the tuned channel. The resolution of the RSSI is 0.5 dB. This high-resolution RSSI enables accurate channel power measurements for clear channel assessment (CCA), carrier sense (CS), and listen before talk (LBT) functionality. A comprehensive programmable packet handler including key features of Silicon Labs' EZMAC is integrated to create a variety of communication topologies ranging from peer-to-peer networks to mesh networks. The extensive programmability of the packet header allows for advanced packet filtering, which, in turn enables a mix of broadcast, group, and point-to-point communication. A wireless communication channel can be corrupted by noise and interference, so it is important to know if the received data is free of errors. A cyclic redundancy check (CRC) is used to detect the presence of erroneous bits in each packet. A CRC is computed and appended at the end of each transmitted packet and verified by the receiver to confirm that no errors have occurred. The packet handler and CRC can significantly reduce the load on the system microcontroller allowing for a simpler and cheaper microcontroller. The digital modem includes the TX modulator, which converts the TX data bits into the corresponding stream of digital modulation values to be summed with the fractional input to the sigma-delta modulator. This modulation approach results in highly accurate resolution of the frequency deviation. A Gaussian filter is implemented to support GFSK, considerably reducing the energy in adjacent channels.

### 5.2.1. Automatic Gain Control (AGC)

The AGC algorithm is implemented digitally using an advanced control loop optimized for fast response time. The AGC occurs within a single bit or in less than 2  $\mu$ s. Peak detectors at the output of the LNA and PGA allow for optimal adjustment of the LNA gain and PGA gain to optimize IM3, selectivity, and sensitivity performance.

### 5.2.2. Auto Frequency Correction (AFC)

Frequency mistuning caused by crystal inaccuracies can be compensated for by enabling the digital automatic frequency control (AFC) in receive mode. There are two types of integrated frequency compensation: modem frequency compensation, and AFC by adjusting the PLL frequency. With AFC disabled, the modem compensation can correct for frequency offsets up to  $\pm 0.25$  times the IF bandwidth. When the AFC is enabled, the received signal will be centered in the pass-band of the IF filter, providing optimal sensitivity and selectivity over a wider range of frequency offsets up to  $\pm 0.35$  times the IF bandwidth. When AFC is enabled, the preamble length needs to be long enough to settle the AFC. As shown in Table 13 on page 21, an additional byte of preamble is typically required to settle the AFC.

### 5.2.3. Received Signal Strength Indicator

The received signal strength indicator (RSSI) is an estimate of the signal strength in the channel to which the receiver is tuned. The RSSI measurement is done after the channel filter, so it is only a measurement of the desired or undesired in-band signal power. There are two different methods for reading the RSSI value and several different options for configuring the RSSI value that is returned. The fastest method for reading the RSSI is to configure one of the four fast response registers (FRR) to return a latched RSSI value. The latched RSSI value is measured once per packet and is latched at a configurable amount of time after RX mode is entered. The fast response registers can be read in 16 SPI clock cycles with no requirement to wait for CTS. The RSSI value may also be read out of the GET\_MODEM\_STATUS command. In this command, both the current RSSI and the latched RSSI are available. The current RSSI value represents the signal strength at the instant in time the GET\_MODEM\_STATUS command is processed and may be read multiple times per packet. Reading the RSSI in the GET\_MODEM\_STATUS command takes longer than reading the RSSI out of the fast response register. After the initial command, it will take 33  $\mu$ s for CTS to be set and then the four or five bytes of SPI clock cycles to read out the respective current or latched RSSI values.

The RSSI configuration options are set in the MODEM\_RSSI\_CONTROL API property. The latched RSSI value may be latched and stored based on the following events: preamble detection, sync detection, or a configurable number of bit times measured after the start of RX mode (minimum of 4 bit times). The requirement for four bit times is determined by the processing delay and settling through the modem and digital channel filter. In MODEM\_RSSI\_CONTROL, the RSSI may be defined to update every bit period or to be averaged and updated every four bit periods. If RSSI averaging over four bits is enabled, the latched RSSI value will be delayed to a minimum of 7 bits after the start of RX mode to allow for the averaging. The latched RSSI values are cleared when entering RX mode so they may be read after the packet is received or after dropping back to standby mode. If the RSSI value has been cleared by the start of RX but not latched yet, a value of 0 will be returned if it is attempted to be read.

The RSSI value read by the API could be translated to dBm by the following linear equation:

$$\text{RSSI}_{\text{dBm}} = \frac{\text{RSSI\_value}}{2} - 130$$



The RSSI values and curves may be offset by the MODEM\_RSSI\_COMP property. The default value of 7'h32 corresponds to no RSSI offset. Setting a value less than 7'h32 corresponds to a negative offset, and a value higher than 7'h32 corresponds to a positive offset. The offset value is in 1 dB steps. For example, setting a value of 7'h3A corresponds to a positive offset of 8 dB.

Clear channel assessment (CCA) or RSSI threshold detection is also available. An RSSI threshold may be set in the MODEM\_RSSI\_THRESH API property. If the RSSI value is above this threshold, an interrupt or GPIO may notify the host. Both the latched version and asynchronous version of this threshold are available on any of the GPIOs. Automatic fast hopping based on RSSI is available. See “5.3.1.2. Automatic RX Hopping and Hop Table”.

## 5.3. Synthesizer

An integrated Sigma Delta ( $\Sigma\Delta$ ) Fractional-N PLL synthesizer capable of operating over 425–525 MHz. Using a  $\Sigma\Delta$  synthesizer has many advantages; it provides flexibility in choosing data rate, deviation, channel frequency, and channel spacing. The transmit modulation is applied directly to the loop in the digital domain through the fractional divider, which results in very precise accuracy and control over the transmit deviation. The frequency resolution in the 425–525 MHz band is 14.3 Hz with more resolution in the other bands. The nominal reference frequency to the PLL is 30 MHz, but any XTAL frequency from 25 to 32 MHz may be used. The modem configuration calculator in WDS will automatically account for the XTAL frequency being used. The PLL utilizes a differential LC VCO with integrated on-chip inductors. The output of the VCO is followed by a configurable divider, which will divide the signal down to the desired output frequency band.

### 5.3.1. Synthesizer Frequency Control

The frequency is set by changing the integer and fractional settings to the synthesizer. The WDS calculator will automatically provide these settings, but the synthesizer equation is shown below for convenience. The APIs for setting the frequency are `FREQ_CONTROL_INTE`, `FREQ_CONTROL_FRAC2`, `FREQ_CONTROL_FRAC1`, and `FREQ_CONTROL_FRAC0`.

$$\text{RF\_channel} = \left( \text{fc\_inte} + \frac{\text{fc\_frac}}{2^{19}} \right) \times \frac{2 \times \text{freq\_xo}}{\text{outdiv}} (\text{Hz})$$

**Note:** The  $\text{fc\_frac}/2^{19}$  value in the above formula has to be a number between 1 and 2.

#### 5.3.1.1. EZ Frequency Programming

In applications that utilize multiple frequencies or channels, it may not be desirable to write four API registers each time a frequency change is required. EZ frequency programming is provided so that only a single register write (channel number) is required to change frequency. A base frequency is first set by first programming the integer and fractional components of the synthesizer. This base frequency will correspond to channel 0. Next, a channel step size is programmed into the `FREQ_CONTROL_CHANNEL_STEP_SIZE_1` and `FREQ_CONTROL_CHANNEL_STEP_SIZE_0` API registers. The resulting frequency will be:

$$\text{RF Frequency} = \text{Base Frerquency} + \text{Channel} \times \text{Stepsize}$$

The second argument of the `START_RX` or `START_TX` is `CHANNEL`, which sets the channel number for EZ frequency programming. For example, if the channel step size is set to 1 MHz, the base frequency is set to 490 MHz with the `INTE` and `FRAC` API registers, and a `CHANNEL` number of 5 is programmed during the `START_TX` command, the resulting frequency will be 495 MHz. If no `CHANNEL` argument is written as part of the `START_RX/TX` command, it will default to the previous value. The initial value of `CHANNEL` is 0; so, if no `CHANNEL` value is written, it will result in the programmed base frequency.

### 5.3.1.2. Automatic RX Hopping and Hop Table

The transceiver supports an automatic hopping feature that can be fully configured through the API. This is intended for RX hopping where the device has to hop from channel to channel and look for packets. Once the device is put into the RX state, it automatically starts hopping through the hop table if the feature is enabled.

The hop table can hold up to 64 entries and is maintained in firmware. Each entry is a channel number; so, the hop table can hold up to 64 channels. The number of entries in the table is set by RX HOP TABLE\_SIZE API.

The specified channels correspond to the EZ frequency programming method for programming the frequency. The receiver starts at the base channel and hops in sequence from the top of the hop table to the bottom. The table will wrap around to the base channel once it reaches the end of the table. An entry of 0xFF in the table indicates that the entry should be skipped. The device will hop to the next non 0xFF entry.

There are three conditions that can be used to determine whether to continue hopping or to stay on a particular channel. These conditions are:

- RSSI threshold
- Preamble timeout (invalid preamble pattern)
- Sync word timeout (invalid or no sync word detected after preamble)

These conditions can be used individually, or they can be enabled all together by configuring the RX\_HOP\_CONTROL API. However, the firmware will make a decision on whether or not to hop based on the first condition that is met.

The RSSI that is monitored is the current RSSI value. This is compared to the threshold, and, if it is above the threshold value, it will stay on the channel. If the RSSI is below the threshold, it will continue hopping. There is no averaging of RSSI done during the automatic hopping from channel to channel. Since the preamble timeout and the sync word timeout are features that require packet handling, the RSSI threshold is the only condition that can be used if the user is in “direct” or “RAW” mode where packet handling features are not used.

Note that the RSSI threshold is not an absolute RSSI value; instead, it is a relative value and should be verified on the bench to find an optimal threshold for the application.

The turnaround time from RX to RX on a different channel using this method is 115  $\mu$ s. The time spent in receive mode will be determined by the configuration of the hop conditions. Manual RX hopping will have the fastest turn-around time but will require more overhead and management by the host MCU.

The following are example steps for using Auto Hop:

1. Set the base frequency (inte + frac) and channel step size.
2. Define the number of entries in the hop table (RX\_HOP\_TABLE\_SIZE).
3. Write the channels to the hop table (RX\_HOP\_TABLE\_ENTRY\_n)
4. Configure the hop condition and enable auto hopping- RSSI, preamble, or sync (RX\_HOP\_CONTROL).
5. Set preamble and sync parameters if enabled.
6. Program the RSSI threshold property in the modem using “MODEM\_RSSI\_THRESH”.
7. Set the preamble threshold using “PREAMBLE\_CONFIG\_STD\_1”.
8. Program the preamble timeout property using “PREAMBLE\_CONFIG\_STD\_2”.
9. Set the sync detection parameters if enabled.
10. If needed, use “GPIO\_PIN\_CFG” to configure a GPIO to toggle on hop and hop table wrap.
11. Use the “START\_RX” API with channel number set to the first valid entry in the hop table (i.e., the first non 0xFF entry).
12. Device should now be in auto hop mode.

### 5.3.1.3. Manual RX Hopping

The RX\_HOP command provides the fastest method for hopping from RX to RX but it requires more overhead and management by the host MCU. Using the RX\_HOP command, the turn-around time is 75  $\mu$ s. The timing is faster with this method than Start\_RX or RX hopping because one of the calculations required for the synthesizer calibrations is offloaded to the host and must be calculated/stored by the host, VCO\_CNT0. For information about using fast manual hopping, contact customer support.