

**RFM31B ISM RECEIVER**

V1.0

**Features**

- Frequency Range
  - 433/868/915MHz ISM bands
- Sensitivity = -121 dBm
- Low Power Consumption
  - 18.5 mA receive
- Data Rate = 0.123 to 256 kbps
- FSK, GFSK, and OOK modulation
- Power Supply = 1.8 to 3.6 V
- Ultra low power shutdown mode
- Digital RSSI
- Wake-up timer
- Auto-frequency calibration (AFC)
- Clear channel RX BW 2.6–620 kHz
- Programmable assessment
- Programmable packet handler
- Programmable GPIOs
- Embedded antenna diversity algorithm
- Configurable packet handler
- Preamble detector
- RX 64 byte FIFO
- Low battery detector
- Temperature sensor and 8-bit ADC
- -40 to +85 °C temperature range
- Integrated voltage regulators
- Frequency hopping capability
- On-chip crystal tuning
- 14- PIN DIP & 16-PIN SMD package
- Low BOM
- Power-on-reset (POR)

**Applications**

- Remote control
- Home security & alarm
- Telemetry
- Personal data logging
- Toy control
- Tire pressure monitoring
- Wireless PC peripherals
- Remote meter reading
- Remote keyless entry
- Home automation
- Industrial control
- Sensor networks
- Health monitors
- Tag readers

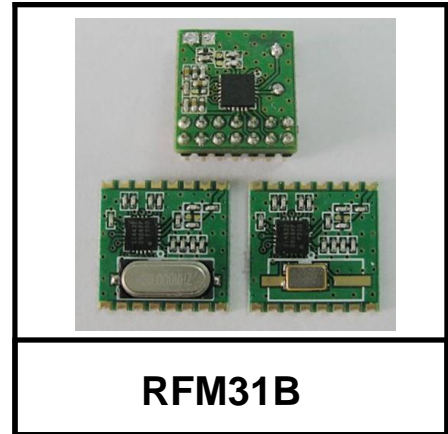
**Description**

HopeRF's RFM31B are highly integrated, low cost, 433/868/915MHz wireless ISM receiver module. The low receive sensitivity (-121dBm) ensures extended range and improved link performance. Built-in antenna diversity and support for frequency hopping can be used to further extend range and enhance performance.

Additional system features such as an automatic wake-up timer, low battery detector, 64 byte RX FIFO, automatic packet handling, and preamble detection reduce overall current consumption and allow the use of a lower-cost system MCU. An integrated temperature sensor, general purpose ADC, power-on-reset (POR), and GPIOs further reduce overall system cost and size.

The RFM31B's digital receive architecture features a high-performance ADC and DSP based modem which performs demodulation, filtering, and packet handling for increased flexibility and performance.

An easy-to-use calculator is provided to quickly configure the radio settings, simplifying customer's system design and reducing time to market.



---

**TABLE OF CONTENTS**


---

| <b><u>Section</u></b>                                   | <b><u>Page</u></b> |
|---|--------------------|
| <b>1. Electrical Specifications</b> .....               | 4                  |
| <b>2. Functional Description</b> .....                  | 10                 |
| 2.1. Operating Modes .....                              | 11                 |
| <b>3. Controller Interface</b> .....                    | 12                 |
| 3.1. Serial Peripheral Interface (SPI) .....            | 12                 |
| 3.2. Operating Mode Control.....                        | 14                 |
| 3.3. Interrupts.....                                    | 17                 |
| 3.4. System Timing .....                                | 18                 |
| 3.5. Frequency Control .....                            | 19                 |
| <b>4. Modulation Options</b> .....                      | 24                 |
| 4.1. FIFO Mode.....                                     | 24                 |
| <b>5. Internal Functional Blocks</b> .....              | 25                 |
| 5.1. RX LNA.....  | 25                 |
| 5.2. RX I-Q Mixer .....                                 | 25                 |
| 5.3. Programmable Gain Amplifier.....                   | 25                 |
| 5.4. ADC .....  | 25                 |
| 5.5. Digital Modem .....                                | 25                 |
| 5.6. Synthesizer .....                                  | 26                 |
| 5.7. Crystal Oscillator.....                            | 27                 |
| 5.8. Regulators .....                                   | 27                 |
| <b>6. Data Handling and Packet Handler</b> .....        | 28                 |
| 6.1. RX FIFO.....                                       | 28                 |
| 6.2. Packet Configuration.....                          | 29                 |
| 6.3. Packet Handler RX Mode .....                       | 29                 |
| 6.4. Data Whitening, Manchester Encoding, and CRC ..... | 32                 |
| 6.5. Preamble Detector .....                            | 32                 |
| 6.6. Preamble Length.....                               | 33                 |
| 6.7. Invalid Preamble Detector.....                     | 33                 |
| 6.8. Synchronization Word Configuration.....            | 33                 |
| 6.9. Receive Header Check.....                          | 34                 |
| <b>7. RX Modem Configuration</b> .....                  | 35                 |
| 7.1. Modem Settings for FSK and GFSK .....              | 35                 |
| <b>8. Auxiliary Functions</b> .....                     | 36                 |
| 8.1. Smart Reset.....                                   | 36                 |
| 8.2. Microcontroller Clock .....                        | 37                 |
| 8.3. General Purpose ADC .....                          | 38                 |
| 8.4. Temperature Sensor .....                           | 39                 |
| 8.5. Low Battery Detector .....                         | 41                 |

# RFM31 B

---

|  |           |
|--|-----------|
| 8.6. Wake-Up Timer and 32 kHz Clock Source ..... | 42        |
| 8.7. Low Duty Cycle Mode.....                    | 44        |
| 8.8. GPIO Configuration .....                    | 45        |
| 8.9. Antenna Diversity .....                     | 46        |
| 8.10. RSSI and Clear Channel Assessment.....     | 47        |
| <b>9. Reference Design .....</b>                 | <b>48</b> |
| <b>10. Register Table and Descriptions .....</b> | <b>49</b> |
| <b>11. Pin Descriptions: RFM31B.....</b>         | <b>52</b> |
| <b>12. Mechanical Dimension: RFM31B.....</b>     | <b>54</b> |
| <b>13. Ordering Information: RFM31B.....</b>     | <b>56</b> |
| <b>Contact Information .....</b>                 | <b>57</b> |

## 1. Electrical Specifications

**Table 1. DC Characteristics**

| Parameter            | Symbol           | Conditions  | Min | Typ  | Max | Units   |
|----------------------|------------------|---|-----|------|-----|---------|
| Supply Voltage Range | $V_{DD}$         |   | 1.8 | 3.0  | 3.6 | V       |
| Power Saving Modes   | $I_{Shutdown}$   | RC Oscillator, Main Digital Regulator, and Low Power Digital Regulator OFF                                  | —   | 15   | 50  | nA      |
|                      | $I_{Standby}$    | Low Power Digital Regulator ON (Register values retained) and Main Digital Regulator, and RC Oscillator OFF | —   | 450  | 800 | nA      |
|                      | $I_{Sleep}$      | RC Oscillator and Low Power Digital Regulator ON (Register values retained) and Main Digital Regulator OFF  | —   | 1    | —   | $\mu$ A |
|                      | $I_{Sensor-LBD}$ | Main Digital Regulator and Low Battery Detector ON, Crystal Oscillator and all other blocks OFF             | —   | 1    | —   | $\mu$ A |
|                      | $I_{Sensor-TS}$  | Main Digital Regulator and Temperature Sensor ON, Crystal Oscillator and all other blocks OFF               | —   | 1    | —   | $\mu$ A |
|                      | $I_{Ready}$      | Crystal Oscillator and Main Digital Regulator ON, all other blocks OFF. Crystal Oscillator buffer disabled  | —   | 800  | —   | $\mu$ A |
| TUNE Mode Current    | $I_{Tune}$       | Synthesizer and regulators enabled  | —   | 8.5  | —   | mA      |
| RX Mode Current      | $I_{RX}$         |   | —   | 18.5 | —   | mA      |

# RFM31 B

**Table 2. Synthesizer AC Electrical Characteristics**

| Parameter                        | Symbol                  | Conditions   | Min | Typ    | Max | Units                     |
|----------------------------------|-------------------------|--|-----|--------|-----|---------------------------|
| Synthesizer Frequency Range      | $F_{\text{SYN}}$        | 433MHz band  | 413 |        | 453 | MHz                       |
|                                  |                         | 868MHz band  | 848 |        | 888 | MHz                       |
|                                  |                         | 915MHz band  | 901 |        | 929 | MHz                       |
| Synthesizer Frequency Resolution | $F_{\text{RES-LB}}$     | 433MHz Band  | —   | 156.25 | —   | Hz                        |
|                                  | $F_{\text{RES-HB}}$     | 868/915MHz Band  | —   | 312.5  | —   | Hz                        |
| Reference Frequency Input Level  | $f_{\text{REF-LV}}$     | When using external reference signal driving XOUT pin, instead of using crystal. Measured peak-to-peak ( $V_{\text{PP}}$ ) | 0.7 | —      | 1.6 | V                         |
| Synthesizer Settling Time        | $t_{\text{LOCK}}$       | Measured from exiting Ready mode with XOSC running to any frequency. Including VCO calibration.                            | —   | 200    | —   | $\mu\text{s}$             |
| Residual FM                      | $\Delta F_{\text{RMS}}$ | Integrated over $\pm 250$ kHz bandwidth (500 Hz lower bound of integration)  | —   | 2      | 4   | $\text{kHz}_{\text{RMS}}$ |
| Phase Noise                      | $L\phi(f_M)$            | $\Delta F = 10$ kHz  | —   | -80    | —   | $\text{dBc/Hz}$           |
|                                  |                         | $\Delta F = 100$ kHz   | —   | -90    | —   | $\text{dBc/Hz}$           |
|                                  |                         | $\Delta F = 1$ MHz   | —   | -115   | —   | $\text{dBc/Hz}$           |
|                                  |                         | $\Delta F = 10$ MHz  | —   | -130   | —   | $\text{dBc/Hz}$           |

**Table 3. Receiver AC Electrical Characteristics**

| Parameter                           | Symbol        | Conditions   | Min | Typ       | Max | Units |
|-------------------------------------|---------------|--|-----|-----------|-----|-------|
| RX Frequency Range                  | $F_{RX}$      | 433MHz band  | 413 |           | 453 | MHz   |
|                                     |               | 868MHz band  | 848 |           | 888 | MHz   |
|                                     |               | 915MHz band  | 901 |           | 929 | MHz   |
| RX Sensitivity                      | $P_{RX\_2}$   | (BER < 0.1%)<br>(2 kbps, GFSK, BT = 0.5, $\Delta f = \pm 5$ kHz)<br>special crystal is used on the module  | —   | -121      | —   | dBm   |
|                                     | $P_{RX\_40}$  | (BER < 0.1%)<br>(1.2kbps,FSK, BT = 0.5,<br>$\Delta f = \pm 45$ kHz)  | —   | -114      | —   | dBm   |
|                                     | $P_{RX\_100}$ | (BER < 0.1%)<br>(100 kbps, GFSK, BT = 0.5,<br>$\Delta f = \pm 50$ kHz)   | —   | -104      | —   | dBm   |
|                                     | $P_{RX\_125}$ | (BER < 0.1%)<br>(125 kbps, GFSK, BT = 0.5,<br>$\Delta f = \pm 62.5$ kHz)   | —   | -101      | —   | dBm   |
|                                     | $P_{RX\_OOK}$ | (BER < 0.1%)<br>(4.8 kbps, 350 kHz BW, OOK)  | —   | -110      | —   | dBm   |
|                                     |               | (BER < 0.1%)<br>(40 kbps, 400 kHz BW, OOK)   | —   | -102      | —   | dBm   |
| RX Channel Bandwidth                | BW            |  | 2.6 | —         | 620 | kHz   |
| BER Variation vs Power Level        | $P_{RX\_RES}$ | Up to +5 dBm Input Level   | —   | 0         | 0.1 | ppm   |
| RSSI Resolution                     | $RES_{RSSI}$  |  | —   | $\pm 0.5$ | —   | dB    |
| $\pm 1$ -Ch Offset Selectivity      | $C/I_{1-CH}$  | Desired Ref Signal 3 dB above sensitivity, BER < 0.1%. Interferer and desired modulated with 40 kbps $\Delta F = 20$ kHz GFSK with BT = 0.5, channel spacing = 150 kHz | —   | -31       | —   | dB    |
| $\pm 2$ -Ch Offset Selectivity      | $C/I_{2-CH}$  |  | —   | -35       | —   | dB    |
| $\geq \pm 3$ -Ch Offset Selectivity | $C/I_{3-CH}$  |  | —   | -40       | —   | dB    |
| Blocking at 1 MHz Offset            | $1M_{BLOCK}$  | Desired Ref Signal 3 dB above sensitivity. Interferer and desired modulated with 40 kbps $\Delta F = 20$ kHz GFSK with BT = 0.5  | —   | -52       | —   | dB    |
| Blocking at 4 MHz Offset            | $4M_{BLOCK}$  |  | —   | -56       | —   | dB    |
| Blocking at 8 MHz Offset            | $8M_{BLOCK}$  |  | —   | -63       | —   | dB    |
| Image Rejection                     | $Im_{REJ}$    | Rejection at the image frequency.<br>IF=937 kHz  | —   | -30       | —   | dB    |
| Spurious Emissions                  | $P_{OB\_RX1}$ | Measured at RX pins  | —   | —         | -54 | dBm   |

# RFM31 B

**Table 4. Auxiliary Block Specifications**

| Parameter   | Symbol               | Conditions  | Min     | Typ  | Max | Units  |
|---|----------------------|---|---------|------|-----|--------|
| Temperature Sensor Accuracy                       | TS <sub>A</sub>      | After calibrated via sensor offset register tvoffs[7:0]                           | —       | 0.5  | —   | °C     |
| Temperature Sensor Sensitivity                    | TS <sub>S</sub>      |   | —       | 5    | —   | mV/°C  |
| Low Battery Detector Resolution                   | LBD <sub>RES</sub>   |   | —       | 50   | —   | mV     |
| Low Battery Detector Conversion Time              | LBD <sub>CT</sub>    |   | —       | 250  | —   | µs     |
| Microcontroller Clock Output Frequency            | F <sub>MC</sub>      | Configurable to 30 MHz, 15 MHz, 10 MHz, 4 MHz, 3 MHz, 2 MHz, 1 MHz, or 32.768 kHz | 32.768K | —    | 30M | Hz     |
| General Purpose ADC Resolution                    | ADC <sub>ENB</sub>   |   | —       | 8    | —   | bit    |
| General Purpose ADC Bit Resolution                | ADC <sub>RES</sub>   |   | —       | 4    | —   | mV/bit |
| Temp Sensor & General Purpose ADC Conversion Time | ADC <sub>CT</sub>    |   | —       | 305  | —   | µs     |
| 30 MHz XTAL Start-Up time                         | t <sub>30M</sub>     |   | —       | 600  | —   | µs     |
| 30 MHz XTAL Cap Resolution                        | 30M <sub>RES</sub>   |   | —       | 97   | —   | fF     |
| 32 kHz XTAL Start-Up Time                         | t <sub>32k</sub>     |   | —       | 6    | —   | sec    |
| 32 kHz XTAL Accuracy using 32 kHz XTAL            | 32K <sub>RES</sub>   |   | —       | 100  | —   | ppm    |
| 32 kHz Accuracy using Internal RC Oscillator      | 32KRC <sub>RES</sub> |   | —       | 2500 | —   | ppm    |
| POR Reset Time                                    | t <sub>POR</sub>     |   | —       | 16   | —   | ms     |
| Software Reset Time                               | t <sub>soft</sub>    |   | —       | 100  | —   | µs     |

**Table 5. Digital IO Specifications (SDO, SDI, SCLK, nSEL, and nIRQ)**

| Parameter                       | Symbol     | Conditions  | Min            | Typ | Max | Units |
|---------------------------------|------------|---|----------------|-----|-----|-------|
| Rise Time                       | $T_{RISE}$ | $0.1 \times V_{DD}$ to $0.9 \times V_{DD}$ , $C_L = 5 \text{ pF}$ | —              | —   | 8   | ns    |
| Fall Time                       | $T_{FALL}$ | $0.9 \times V_{DD}$ to $0.1 \times V_{DD}$ , $C_L = 5 \text{ pF}$ | —              | —   | 8   | ns    |
| Input Capacitance               | $C_{IN}$   |   | —              | —   | 1   | pF    |
| Logic High Level Input Voltage  | $V_{IH}$   |   | $V_{DD} - 0.6$ | —   | —   | V     |
| Logic Low Level Input Voltage   | $V_{IL}$   |   | —              | —   | 0.6 | V     |
| Input Current                   | $I_{IN}$   | $0 < V_{IN} < V_{DD}$   | -100           | —   | 100 | nA    |
| Logic High Level Output Voltage | $V_{OH}$   | $I_{OH} < 1 \text{ mA}$ source, $V_{DD} = 1.8 \text{ V}$          | $V_{DD} - 0.6$ | —   | —   | V     |
| Logic Low Level Output Voltage  | $V_{OL}$   | $I_{OL} < 1 \text{ mA}$ sink, $V_{DD} = 1.8 \text{ V}$            | —              | —   | 0.6 | V     |

**Table 6. GPIO Specifications (GPIO\_0, GPIO\_1, and GPIO\_2)**

| Parameter                            | Symbol       | Conditions   | Min            | Typ | Max | Units         |
|--------------------------------------|--------------|--|----------------|-----|-----|---------------|
| Rise Time                            | $T_{RISE}$   | $0.1 \times V_{DD}$ to $0.9 \times V_{DD}$ ,<br>$C_L = 10 \text{ pF}$ , $DRV < 1:0 > = HH$ | —              | —   | 8   | ns            |
| Fall Time                            | $T_{FALL}$   | $0.9 \times V_{DD}$ to $0.1 \times V_{DD}$ ,<br>$C_L = 10 \text{ pF}$ , $DRV < 1:0 > = HH$ | —              | —   | 8   | ns            |
| Input Capacitance                    | $C_{IN}$     |  | —              | —   | 1   | pF            |
| Logic High Level Input Voltage       | $V_{IH}$     |  | $V_{DD} - 0.6$ | —   | —   | V             |
| Logic Low Level Input Voltage        | $V_{IL}$     |  | —              | —   | 0.6 | V             |
| Input Current                        | $I_{IN}$     | $0 < V_{IN} < V_{DD}$  | -100           | —   | 100 | nA            |
| Input Current If Pullup is Activated | $I_{INP}$    | $V_{IL} = 0 \text{ V}$   | 5              | —   | 25  | $\mu\text{A}$ |
| Maximum Output Current               | $I_{OmaxLL}$ | $DRV < 1:0 > = LL$   | 0.1            | 0.5 | 0.8 | mA            |
|                                      | $I_{OmaxLH}$ | $DRV < 1:0 > = LH$   | 0.9            | 2.3 | 3.5 | mA            |
|                                      | $I_{OmaxHL}$ | $DRV < 1:0 > = HL$   | 1.5            | 3.1 | 4.8 | mA            |
|                                      | $I_{OmaxHH}$ | $DRV < 1:0 > = HH$   | 1.8            | 3.6 | 5.4 | mA            |
| Logic High Level Output Voltage      | $V_{OH}$     | $I_{OH} < I_{Omax}$ source,<br>$V_{DD} = 1.8 \text{ V}$                                    | $V_{DD} - 0.6$ | —   | —   | V             |
| Logic Low Level Output Voltage       | $V_{OL}$     | $I_{OL} < I_{Omax}$ sink,<br>$V_{DD} = 1.8 \text{ V}$                                      | —              | —   | 0.6 | V             |



# RFM31 B

---

**Table 7. Absolute Maximum Ratings**

| Parameter  | Value                       | Unit |
|--|-----------------------------|------|
| V <sub>DD</sub> to GND   | -0.3, +3.6                  | V    |
| Voltage on Digital Control Inputs  | -0.3, V <sub>DD</sub> + 0.3 | V    |
| Voltage on Analog Inputs   | -0.3, V <sub>DD</sub> + 0.3 | V    |
| RX Input Power   | +10                         | dBm  |
| Operating Temperature Range (special crystal is used on the module) T <sub>S</sub> | -40 to +85                  | °C   |
| Operating Temperature Range (Normal crystal is used on the module) T               | -20 to +60                  | °C   |
| Thermal Impedance $\theta_{JA}$  | 30                          | °C/W |
| Storage Temperature Range T <sub>STG</sub>   | -55 to +125                 | °C   |

**Note:** 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. Caution: ESD sensitive device.

# RFM31 B

## 2. Functional Description

HopeRF's RFM31B are highly integrated, low cost, 433/868/915MHz wireless ISM receivers module . The wide operating voltage range of 1.8–3.6V and low current consumption makes the RFM31B an ideal solution for battery powered applications.

The RFM31B uses a single-conversion mixer to downconvert the 2-level FSK/GFSK/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 then output to the system MCU through a programmable GPIO or via the standard SPI bus by reading the 64-byte RX FIFO.

A high precision local oscillator (LO) is generated by an integrated VCO and  $\Delta\Sigma$  Fractional-N PLL synthesizer. The synthesizer is designed to support configurable data rates, output frequency and frequency deviation .

The RFM31B is designed to work with a microcontroller to create a very low cost system. Voltage regulators are integrated on-chip which allows for a wide operating supply voltage range from +1.8 to +3.6V. A standard 4-pin SPI bus is used to communicate with an external microcontroller. Three configurable general purpose I/Os are available. A complete list of the available GPIO functions is shown in "8. Auxiliary Functions" and includes microcontroller clock output, Antenna Diversity, Antenna Switch, POR, and various interrupts. A complete list of the available GPIO functions is shown in "RFM31B Register Descriptions."

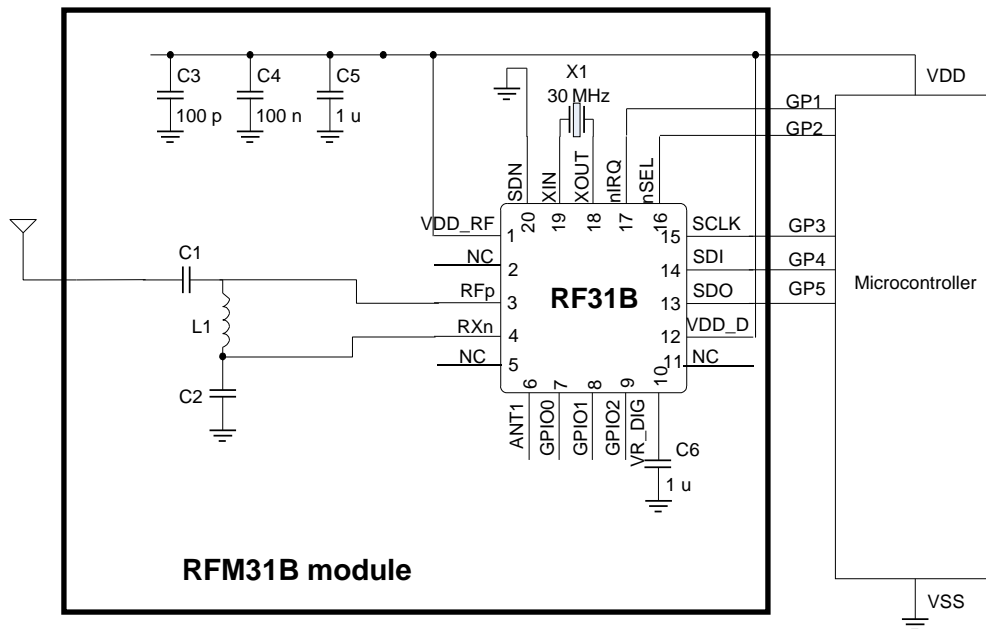


Figure 1.RFM31B Application Example

## 2.1. Operating Modes

The RFM31B provides several operating modes which can be used to optimize the power consumption for a given application. Depending upon the system communication protocol, an optimal trade-off between the radio wake time and power consumption can be achieved.

Table 8 summarizes the operating modes of the RFM31B. In general, any given operating mode may be classified as an active mode or a power saving mode. The table indicates which block(s) are enabled (active) in each corresponding mode. With the exception of the SHUTDOWN mode, all can be dynamically selected by sending the appropriate commands over the SPI operating mode. An “X” in any cell means that, in the given mode of operation, that block can be independently programmed to be either ON or OFF, without noticeably impacting the current consumption. The SPI circuit block includes the SPI interface hardware and the device register space. The 32 kHz OSC block includes the 32.768 kHz RC oscillator or 32.768 kHz crystal oscillator and wake-up timer. AUX (Auxiliary Blocks) includes the temperature sensor, general purpose ADC, and low-battery detector.

**Table 8. Operating Modes**

| Mode Name | Circuit Blocks                     |     |            |     |             |     |     | I <sub>VDD</sub> |
|-----------|------------------------------------|-----|------------|-----|-------------|-----|-----|------------------|
|           | Digital LDO                        | SPI | 32 kHz OSC | AUX | 30 MHz XTAL | PLL | RX  |                  |
| SHUTDOWN  | OFF<br>(Register contents lost)    | OFF | OFF        | OFF | OFF         | OFF | OFF | 15 nA            |
| STANDBY   | ON<br>(Register contents retained) | ON  | OFF        | OFF | OFF         | OFF | OFF | 450 nA           |
| SLEEP     |                                    | ON  | ON         | X   | OFF         | OFF | OFF | 1 μA             |
| SENSOR    |                                    | ON  | X          | ON  | OFF         | OFF | OFF | 1 μA             |
| READY     |                                    | ON  | X          | X   | ON          | OFF | OFF | 800 μA           |
| TUNING    |                                    | ON  | X          | X   | ON          | ON  | OFF | 8.5 mA           |
| RECEIVE   |                                    | ON  | X          | X   | ON          | ON  | ON  | 18.5 mA          |

## 3. Controller Interface

### 3.1. Serial Peripheral Interface (SPI)

The RFM31B communicates with the host MCU over a standard 3-wire SPI interface: SCLK, SDI, and nSEL. The host MCU can read data from the device on the SDO output pin. A SPI transaction is a 16-bit sequence which consists of a Read-Write (R/W) select bit, followed by a 7-bit address field (ADDR), and an 8-bit data field (DATA) as demonstrated in Figure 2. The 7-bit address field is used to select one of the 128, 8-bit control registers. The R/W select bit determines whether the SPI transaction is a read or write transaction. If R/W = 1 it signifies a WRITE transaction, while R/W = 0 signifies a READ transaction. The contents (ADDR or DATA) are latched into the RFM31B every eight clock cycles. The timing parameters for the SPI interface are shown in Table 9. The SCLK rate is flexible with a maximum rate of 10 MHz.

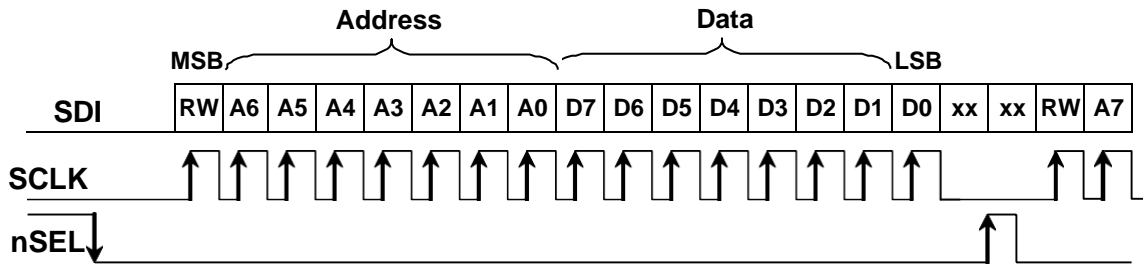


Figure 2. SPI Timing

Table 9. Serial Interface Timing Parameters

| Symbol   | Parameter              | Min (nsec) | 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 | 20         |         |
| $t_{EN}$ | Output enable time     | 20         |         |
| $t_{DE}$ | Output disable time    | 50         |         |
| $t_{SS}$ | Select setup time      | 20         |         |
| $t_{SH}$ | Select hold time       | 50         |         |
| $t_{SW}$ | Select high period     | 80         |         |

To read back data from the RFM31B, the R/W bit must be set to 0 followed by the 7-bit address of the register from which to read. The 8 bit DATA field following the 7-bit ADDR field is ignored on the SDI pin when R/W = 0. The next eight negative edge transitions of the SCLK signal will clock out the contents of the selected register. The data read from the selected register will be available on the SDO output pin. The READ function is shown in Figure 3. After the READ function is completed the SDO pin will remain at either a logic 1 or logic 0 state depending on the last data bit clocked out (D0). When nSEL goes high the SDO output pin will be pulled high by internal pullup.

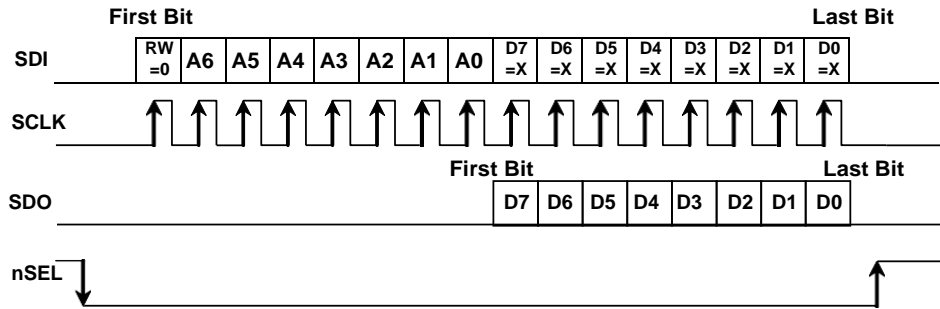


Figure 3. SPI Timing—READ Mode

The SPI interface contains a burst read/write mode which allows for reading/writing sequential registers without having to re-send the SPI address. When the nSEL bit is held low while continuing to send SCLK pulses, the SPI interface will automatically increment the ADDR and read from/write to the next address. An example burst write transaction is illustrated in Figure 4 and a burst read in Figure 5. As long as nSEL is held low, input data will be latched into the RFM31B every eight SCLK cycles.

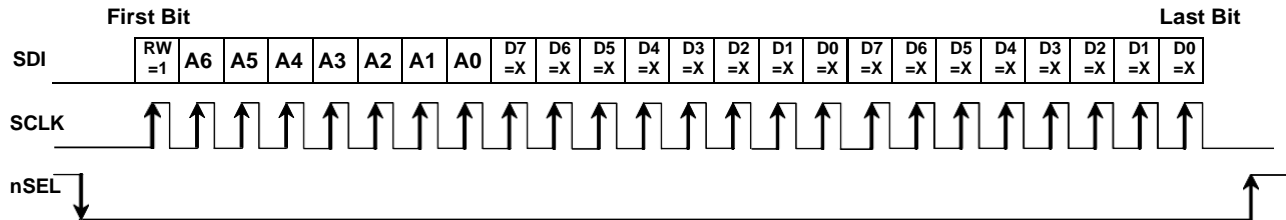


Figure 4. SPI Timing—Burst Write Mode

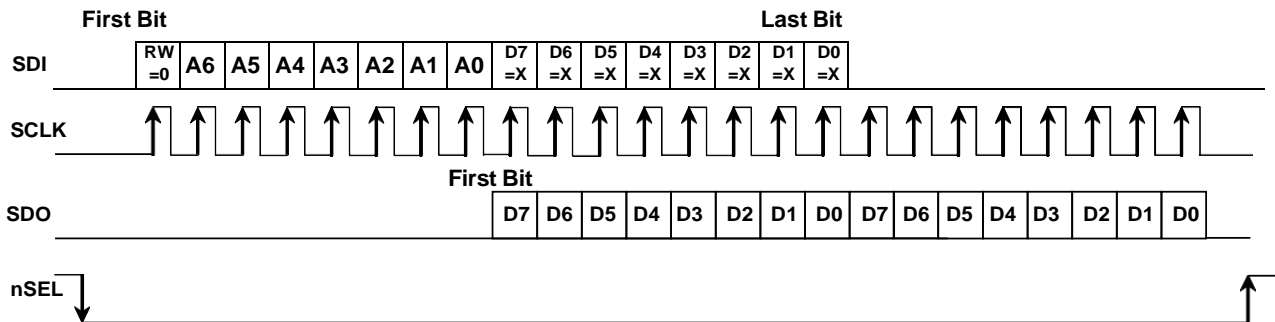
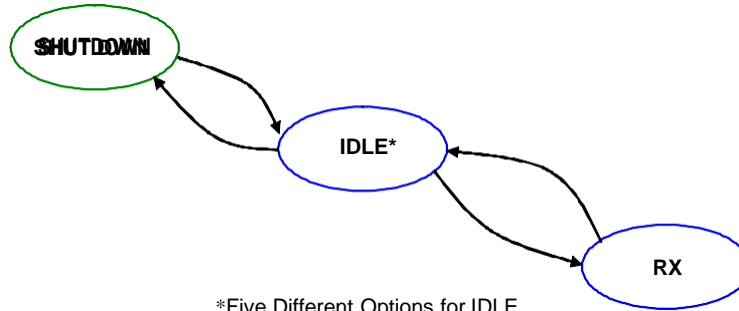


Figure 5. SPI Timing—Burst Read Mode

## 3.2. Operating Mode Control

There are three primary states in the RFM31B radio state machine: SHUTDOWN, IDLE, and RX (see Figure 6). The SHUTDOWN state completely shuts down the radio to minimize current consumption. There are five different configurations/options for the IDLE state which can be selected to optimize the chip to the applications needs. "Register 07h. Operating Mode and Function Control 1" controls which operating mode/state is selected with the exception of SHUTDOWN which is controlled by SDN pin. The RX state may be reached automatically from any of the IDLE states by setting the rxon bit in "Register 07h. Operating Mode and Function Control 1". Table 10 shows each of the operating modes with the time required to reach RX mode as well as the current consumption of each mode.

The RFM31B includes a low-power digital regulated supply (LPLDO) which is internally connected in parallel to the output of the main digital regulator. This common digital supply voltage is connected to all digital circuit blocks including the digital modem, crystal oscillator, SPI, and register space. The LPLDO has extremely low quiescent current consumption but limited current supply capability; it is used only in the IDLE-STANDBY and IDLE-SLEEP modes. The main digital regulator is automatically enabled in all other modes.



**Figure 6. State Machine Diagram**

**Table 10. Operating Modes Response Time**

| State/Mode      | Response Time to RX | Current in State /Mode [ $\mu$ A] |
|-----------------|---------------------|-----------------------------------|
| Shut Down State | 16.8 ms             | 15 nA                             |
| Idle States:    |                     |                                   |
| Standby Mode    | 800 $\mu$ s         | 450 nA                            |
| Sleep Mode      | 800 $\mu$ s         | 1 $\mu$ A                         |
| Sensor Mode     | 800 $\mu$ s         | 1 $\mu$ A                         |
| Ready Mode      | 200 $\mu$ s         | 800 $\mu$ A                       |
| Tune Mode       | 200 $\mu$ s         | 8.5 mA                            |
| RX State        | NA                  | 18.5 mA                           |

### **3.2.1. SHUTDOWN State**

The SHUTDOWN state is the lowest current consumption state of the device with nominally less than 15 nA of current consumption. The SHUTDOWN state may be entered by driving the SDN pin 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 the chip is connected to the power supply, a POR will be initiated after the falling edge of SDN.

### **3.2.2. IDLE State**

There are five different modes in the IDLE state which may be selected by "Register 07h. Operating Mode and Function Control 1". All modes have a tradeoff between current consumption and response time to RX mode. This tradeoff is shown in Table 10. After the POR event, SWRESET, or exiting from the SHUTDOWN state the chip will default to the IDLE-READY mode. After a POR event the interrupt registers must be read to properly enter the SLEEP, SENSOR, or STANDBY mode and to control the 32 kHz clock correctly.

#### **3.2.2.1. STANDBY Mode**

STANDBY mode has the lowest current consumption of the five IDLE states with only the LPLDO enabled to maintain the register values. In this mode the registers can be accessed in both read and write mode. The STANDBY mode can be entered by writing 0h to "Register 07h. Operating Mode and Function Control 1". If an interrupt has occurred (i.e., the nIRQ pin = 0) the interrupt registers must be read to achieve the minimum current consumption. Additionally, the ADC should not be selected as an input to the GPIO in this mode as it will cause excess current consumption.

#### **3.2.2.2. SLEEP Mode**

In SLEEP mode the LPLDO is enabled along with the Wake-Up-Timer, which can be used to accurately wake-up the radio at specified intervals. See "8.6. Wake-Up Timer and 32kHz Clock Source" for more information on the Wake-Up-Timer. SLEEP mode is entered by setting enwt = 1 (40h) in "Register 07h. Operating Mode and Function Control 1". If an interrupt has occurred (i.e., the nIRQ pin = 0) the interrupt registers must be read to achieve the minimum current consumption. Also, the ADC should not be selected as an input to the GPIO in this mode as it will cause excess current consumption.

#### **3.2.2.3. SENSOR Mode**

In SENSOR Mode either the Low Battery Detector, Temperature Sensor, or both may be enabled in addition to the LPLDO and Wake-Up-Timer. The Low Battery Detector can be enabled by setting enlbd = 1 in "Register 07h. Operating Mode and Function Control 1". See "8.4. Temperature Sensor" and "8.5. Low Battery Detector" for more information on these features. If an interrupt has occurred (i.e., the nIRQ pin = 0) the interrupt registers must be read to achieve the minimum current consumption.

#### **3.2.2.4. READY Mode**

READY Mode is designed to give a fast transition time to RX mode with reasonable current consumption. In this mode the Crystal oscillator remains enabled reducing the time required to switch to RX mode by eliminating the crystal start-up time. READY mode is entered by setting xton = 1 in "Register 07h. Operating Mode and Function Control 1". To achieve the lowest current consumption state the crystal oscillator buffer should be disabled in "Register 62h. Crystal Oscillator Control and Test." To exit ready mode, bufovr (bit 1) of this register must be set back to 0.

#### **3.2.2.5. TUNE Mode**

In TUNE Mode the PLL remains enabled in addition to the other blocks enabled in the IDLE modes. This will give the fastest response to RX mode as the PLL will remain locked but it results in the highest current consumption. This mode of operation is designed for frequency hopping spread spectrum systems (FHSS). TUNE mode is entered by setting pllcn = 1 in "Register 07h. Operating Mode and Function Control 1". It is not necessary to set xton to 1 for this mode, the internal state machine automatically enables the crystal oscillator.

# RFM31 B

---

## 3.2.3. RX State

The RX state may be entered from any of the IDLE modes when the rxon bit is set to 1 in "Register 07h. Operating Mode and Function Control 1". A built-in sequencer takes care of all the actions required to transition from one of the IDLE modes to the RX state. The following sequence of events will occur automatically to get the chip into RX mode when going from STANDBY mode to RX mode by setting the rxon bit:

1. Enable the main 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 (this action is skipped when the vcocal bit is "0", default value is "1").
5. Wait until PLL settles to required receive frequency (controlled by an internal timer).
6. Enable receive 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.

## 3.2.4. Device Status

| Add | R/W | Function/Description | D7     | D6     | D5     | D4      | D3      | D2 | D1     | D0     | POR Def. |
|-----|-----|----------------------|--------|--------|--------|---------|---------|----|--------|--------|----------|
| 02  | R   | Device Status        | ffovfl | ffunfl | rxffem | headerr | freqerr |    | cps[1] | cps[0] | —        |

The operational status of the chip can be read from "Register 02h. Device Status".



### 3.3. Interrupts

The RFM31B 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) shown below occur. The nIRQ pin will remain low until the microcontroller reads the Interrupt Status Register(s) (Registers 03h–04h) containing the active Interrupt Status bit. The nIRQ output signal will then be reset until the next change in status is detected. The interrupts must be enabled by the corresponding enable bit in the Interrupt Enable Registers (Registers 05h–06h). All enabled interrupt bits will be cleared when the microcontroller reads the interrupt status register. If the interrupt is not enabled when the event occurs it will not trigger the nIRQ pin, but the status may still be read at anytime in the Interrupt Status registers.

| Add | R/W | Function/Description | D7      | D6        | D5          | D4          | D3    | D2       | D1        | D0         | POR Def. |
|-----|-----|----------------------|---------|-----------|-------------|-------------|-------|----------|-----------|------------|----------|
| 03  | R   | Interrupt Status 1   | ifferr  | Reserved  | Reserved    | irxffafull  | iext  | Reserved | ipkvalid  | icrcerror  | —        |
| 04  | R   | Interrupt Status 2   | iswdet  | ipreaval  | ipreainval  | irssi       | iwut  | ilbd     | ichiprdy  | ipor       | —        |
| 05  | R/W | Interrupt Enable 1   | enfferr | Reserved  | Reserved    | enrxffafull | enext | Reserved | enpkvalid | encrcerror | 00h      |
| 06  | R/W | Interrupt Enable 2   | enswdet | enpreaval | enpreainval | enrssi      | enwut | enlbd    | enchiprdy | enpor      | 01h      |

For a complete descriptions of each interrupt, see “RFM31B Register Descriptions.”

## 3.4. System Timing

The system timing for RX mode is shown in Figure 7. The user only needs to program the desired mode, and the internal sequencer will properly transition the part from its current mode.

The VCO will automatically calibrate at every frequency change or power up. The PLL T0 time is to allow for bias settling of the VCO. The PLL TS time is for the settling time of the PLL, which has a default setting of 100  $\mu$ s. The total time for PLL T0, PLL CAL, and PLL TS under all conditions is 200  $\mu$ s. Under certain applications, the PLL T0 time and the PLL CAL may be skipped for faster turn-around time. Contact applications support if faster turnaround time is desired.

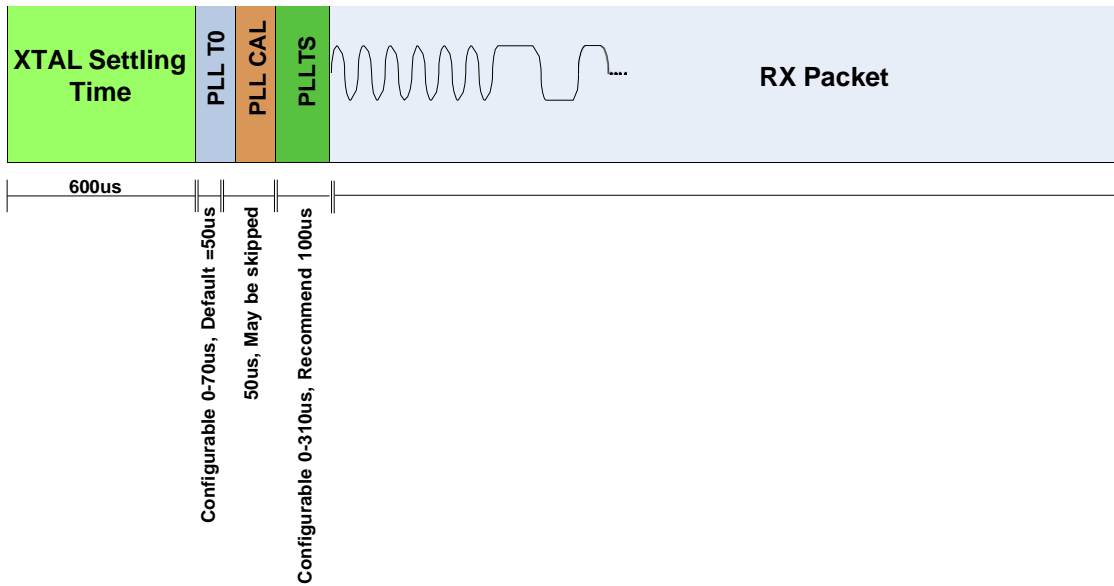


Figure 7. RX Timing

### 3.5. Frequency Control

For calculating the necessary frequency register settings it is recommended that customers use HOPERF Register Calculator worksheet (in Microsoft Excel) available on the product website. These methods offer a simple method to quickly determine the correct settings based on the application requirements. The following information can be used to calculate these values manually.

#### 3.5.1. Frequency Programming

In order to receive an RF signal, the desired channel frequency,  $f_{carrier}$ , must be programmed into the RFM31B. Note that this frequency is the center frequency of the desired channel and not an LO frequency. The carrier frequency is generated by a Fractional-N Synthesizer, using 10 MHz both as the reference frequency and the clock of the (3<sup>rd</sup> order)  $\Delta\Sigma$  modulator. This modulator uses modulo 64000 accumulators. This design was made to obtain the desired frequency resolution of the synthesizer. The overall division ratio of the feedback loop consists of an integer part (N) and a fractional part (F). In a generic sense, the output frequency of the synthesizer is as follows:

$$f_{OUT} = 10MHz \times (N + F)$$

The fractional part (F) is determined by three different values, Carrier Frequency (fc[15:0]), Frequency Offset (fo[8:0]), and Frequency Deviation (fd[7:0]). Due to the fine resolution and high loop bandwidth of the synthesizer, FSK modulation is applied inside the loop and is done by varying F according to the incoming data; this is discussed further in "3.5.4. Frequency Offset Adjustment". Also, a fixed offset can be added to fine-tune the carrier frequency and counteract crystal tolerance errors. For simplicity assume that only the fc[15:0] register will determine the fractional component. The equation for selection of the carrier frequency is shown below:

$$f_{carrier} = 10MHz \times (hbsel + 1) \times (N + F)$$

$$f_{carrier} = 10MHz * (hbsel + 1) * (fb[4 : 0] + 24 + \frac{fc[15 : 0]}{64000})$$

| Add | R/W | Function/Description        | D7       | D6       | D5       | D4       | D3       | D2       | D1    | D0    | POR Def. |
|-----|-----|-----------------------------|----------|----------|----------|----------|----------|----------|-------|-------|----------|
| 73  | R/W | Frequency Offset 1          | fo[7]    | fo[6]    | fo[5]    | fo[4]    | fo[3]    | fo[2]    | fo[1] | fo[0] | 00h      |
| 74  | R/W | Frequency Offset 2          | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | fo[9] | fo[8] | 00h      |
| 75  | R/W | Frequency Band Select       | Reserved | sbsel    | hbsel    | fb[4]    | fb[3]    | fb[2]    | fb[1] | fb[0] | 35h      |
| 76  | R/W | Nominal Carrier Frequency 1 | fc[15]   | fc[14]   | fc[13]   | fc[12]   | fc[11]   | fc[10]   | fc[9] | fc[8] | BBh      |
| 77  | R/W | Nominal Carrier Frequency 0 | fc[7]    | fc[6]    | fc[5]    | fc[4]    | fc[3]    | fc[2]    | fc[1] | fc[0] | 80h      |

The integer part (N) is determined by fb[4:0]. Additionally, the frequency can be halved by connecting a ÷2 divider to the output. This divider is not inside the loop and is controlled by the hbsel bit in "Register 75h. Frequency Band Select." This effectively partitions the entire 240–960 MHz frequency range into two separate bands: High Band (HB) for hbsel = 1, and Low Band (LB) for hbsel = 0. The valid range of fb[4:0] is from 0 to 23. If a higher value is written into the register, it will default to a value of 23. The integer part has a fixed offset of 24 added to it as shown in the formula above. Table 11 demonstrates the selection of fb[4:0] for the corresponding frequency band.

After selection of the fb (N) the fractional component may be solved with the following equation:

$$fc[15 : 0] = \left( \frac{f_{carrier}}{10MHz * (hbsel + 1)} - fb[4 : 0] - 24 \right) * 64000$$

fb and fc are the actual numbers stored in the corresponding registers.

**Table 11. Frequency Band Selection**

| fb[4:0] Value | N  | Frequency Band |               |
|---------------|----|----------------|---------------|
|               |    | hbsel=0        | hbsel=1       |
| 0             | 24 | 240–249.9 MHz  | 480–499.9 MHz |
| 1             | 25 | 250–259.9 MHz  | 500–519.9 MHz |
| 2             | 26 | 260–269.9 MHz  | 520–539.9 MHz |
| 3             | 27 | 270–279.9 MHz  | 540–559.9 MHz |
| 4             | 28 | 280–289.9 MHz  | 560–579.9 MHz |
| 5             | 29 | 290–299.9 MHz  | 580–599.9 MHz |
| 6             | 30 | 300–309.9 MHz  | 600–619.9 MHz |
| 7             | 31 | 310–319.9 MHz  | 620–639.9 MHz |
| 8             | 32 | 320–329.9 MHz  | 640–659.9 MHz |
| 9             | 33 | 330–339.9 MHz  | 660–679.9 MHz |
| 10            | 34 | 340–349.9 MHz  | 680–699.9 MHz |
| 11            | 35 | 350–359.9 MHz  | 700–719.9 MHz |
| 12            | 36 | 360–369.9 MHz  | 720–739.9 MHz |
| 13            | 37 | 370–379.9 MHz  | 740–759.9 MHz |
| 14            | 38 | 380–389.9 MHz  | 760–779.9 MHz |
| 15            | 39 | 390–399.9 MHz  | 780–799.9 MHz |
| 16            | 40 | 400–409.9 MHz  | 800–819.9 MHz |
| 17            | 41 | 410–419.9 MHz  | 820–839.9 MHz |
| 18            | 42 | 420–429.9 MHz  | 840–859.9 MHz |
| 19            | 43 | 430–439.9 MHz  | 860–879.9 MHz |
| 20            | 44 | 440–449.9 MHz  | 880–899.9 MHz |
| 21            | 45 | 450–459.9 MHz  | 900–919.9 MHz |
| 22            | 46 | 460–469.9 MHz  | 920–939.9 MHz |
| 23            | 47 | 470–479.9 MHz  | 940–960 MHz   |

The chip will automatically shift the frequency of the Synthesizer down by 937.5 kHz (30 MHz ÷ 32) to achieve the correct Intermediate Frequency (IF) when RX mode is entered. Low-side injection is used in the RX Mixing architecture.

## 3.5.2. Easy Frequency Programming for FHSS

While Registers 73h–77h may be used to program the carrier frequency of the RFM31B, it is often easier to think in terms of “channels” or “channel numbers” rather than an absolute frequency value in Hz. Also, there may be some timing-critical applications (such as for Frequency Hopping Systems) in which it is desirable to change frequency by programming a single register. Once the channel step size is set, the frequency may be changed by a single register corresponding to the channel number. A nominal frequency is first set using Registers 73h–77h, as described above. Registers 79h and 7Ah are then used to set a channel step size and channel number, relative to the nominal setting. The Frequency Hopping Step Size ( $fhs[7:0]$ ) is set in increments of 10 kHz with a maximum channel step size of 2.56 MHz. The Frequency Hopping Channel Select Register then selects channels based on multiples of the step size.

$$F_{carrier} = F_{nom} + fhs[7:0] \times (fhch[7:0] \times 10kHz)$$

For example, if the nominal frequency is set to 900 MHz using Registers 73h–77h, the channel step size is set to 1 MHz using "Register 7Ah. Frequency Hopping Step Size," and "Register 79h. Frequency Hopping Channel Select" is set to 5d, the resulting carrier frequency would be 905 MHz. Once the nominal frequency and channel step size are programmed in the registers, it is only necessary to program the  $fhch[7:0]$  register in order to change the frequency.

| Add | R/W | Function/Description             | D7      | D6      | D5      | D4      | D3      | D2      | D1      | D0      | POR Def. |
|-----|-----|----------------------------------|---------|---------|---------|---------|---------|---------|---------|---------|----------|
| 79  | R/W | Frequency Hopping Channel Select | fhch[7] | fhch[6] | fhch[5] | fhch[4] | fhch[3] | fhch[2] | fhch[1] | fhch[0] | 00h      |
| 7A  | R/W | Frequency Hopping Step Size      | fhs[7]  | fhs[6]  | fhs[5]  | fhs[4]  | fhs[3]  | fhs[2]  | fhs[1]  | fhs[0]  | 00h      |

## 3.5.3. Automatic State Transition for Frequency Change

If registers 79h or 7Ah are changed in RX mode, the state machine will automatically transition the chip back to TUNE and change the frequency. This feature is useful to reduce the number of SPI commands required in a Frequency Hopping System. This in turn reduces microcontroller activity, reducing current consumption.

### 3.5.4. Frequency Offset Adjustment

When the AFC is disabled the frequency offset can be adjusted manually by fo[9:0] in registers 73h and 74h. It is not possible to have both AFC and offset as internally they share the same register. The frequency offset adjustment and the AFC both are implemented by shifting the Synthesizer Local Oscillator frequency. This register is a signed register so in order to get a negative offset it is necessary to take the two's complement of the positive offset number. The offset can be calculated by the following:

$$DesiredOffset = 156.25Hz \times (hbsel + 1) \times fo[9 : 0]$$

$$fo[9 : 0] = \frac{DesiredOffset}{156.25Hz \times (hbsel + 1)}$$

The adjustment range in high band is ±160 kHz and in low band it is ±80 kHz. For example to compute an offset of +50 kHz in high band mode fo[9:0] should be set to 0A0h. For an offset of -50 kHz in high band mode the fo[9:0] register should be set to 360h.

| Add | R/W | Function/Description | D7       | D6       | D5       | D4       | D3       | D2       | D1    | D0    | POR Def. |
|-----|-----|----------------------|----------|----------|----------|----------|----------|----------|-------|-------|----------|
| 73  | R/W | Frequency Offset     | fo[7]    | fo[6]    | fo[5]    | fo[4]    | fo[3]    | fo[2]    | fo[1] | fo[0] | 00h      |
| 74  | R/W | Frequency Offset     | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | fo[9] | fo[8] | 00h      |

### 3.5.5. Automatic Frequency Control (AFC)

All AFC settings can be easily obtained from the settings calculator. This is the recommended method to program all AFC settings. This section is intended to describe the operation of the AFC in more detail to help understand the trade-offs of using AFC. The receiver supports automatic frequency control (AFC) to compensate for frequency differences between the transmitter and receiver reference frequencies. These differences can be caused by the absolute accuracy and temperature dependencies of the reference crystals. Due to frequency offset compensation in the modem, the receiver is tolerant to frequency offsets up to 0.25 times the IF bandwidth when the AFC is disabled. 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 0.35 times the IF bandwidth. The trade-off of receiver sensitivity (at 1% PER) versus carrier offset and the impact of AFC are illustrated in Figure 9.

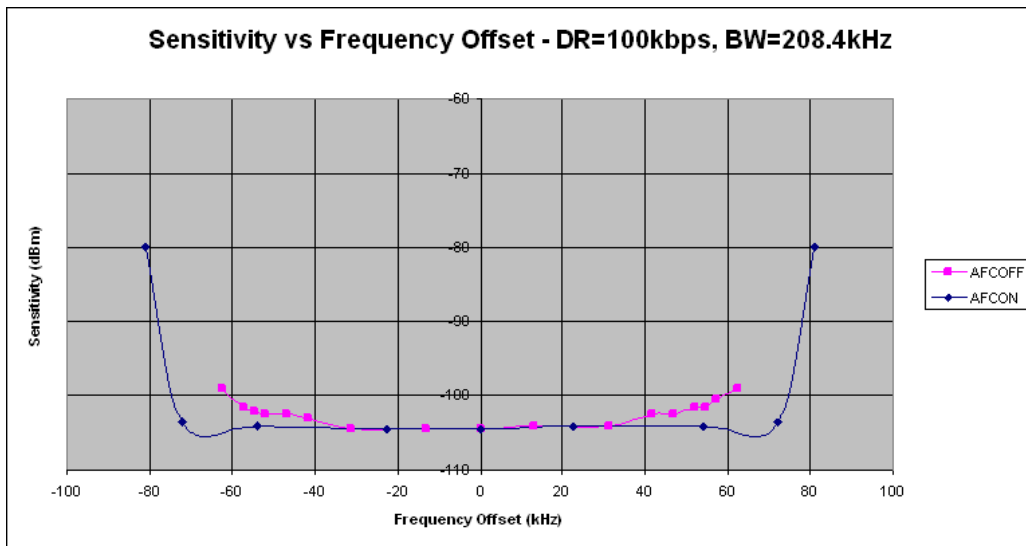


Figure 8. Sensitivity at 1% PER vs. Carrier Frequency Offset

When AFC is enabled, the preamble length needs to be long enough to settle the AFC. In general, one byte of preamble is sufficient to settle the AFC. Disabling the AFC allows the preamble to be shortened from 40 bits to 32 bits. Note that with the AFC disabled, the preamble length must still be long enough to settle the receiver and to detect the preamble (see "6.6. Preamble Length" ). The AFC corrects the detected frequency offset by changing the frequency of the Fractional-N PLL. When the preamble is detected, the AFC will freeze for the remainder of the packet. In multi-packet mode, the AFC is reset at the end of every packet and will re-acquire the frequency offset for the next packet. The AFC loop includes a bandwidth limiting mechanism improving the rejection of out of band signals. When the AFC loop is enabled, its pull-in-range is determined by the bandwidth limiter value (AFCLimiter) which is located in register 2Ah.

$$\text{AFC\_pull\_in\_range} = \pm\text{AFCLimiter}[7:0] \times (\text{hbsel}+1) \times 625 \text{ Hz}$$

The AFC Limiter register is an unsigned register and its value can be obtained from the Register Calculator spreadsheet.

|              | <b>Frequency Correction</b> |
|--------------|-----------------------------|
| AFC disabled | Freq Offset Register        |
| AFC enabled  | AFC                         |

## 4. Modulation Options

### 4.1. FIFO Mode

In FIFO mode, the receive data is stored in integrated FIFO register memory. The FIFOs are accessed via "Register 7Fh. FIFO Access," and are most efficiently accessed with burst read/write operation as discussed in "3.1. Serial Peripheral Interface (SPI)" .

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 Registers (see Table12). 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 real-time) 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 RX State when either the ipksent or ipkvalid interrupt occurs. The chip will return to any of the other states based on the settings in "Register 07h. Operating Mode and Function Control 1."

In RX mode, the rxon bit will be cleared if ipkvalid occurs and the rxmpk bit (RX Multi-Packet bit, SPI Register 08h bit [4]) is not set. When the rxmpk bit is set, the part will not exit the RX state after successfully receiving a packet, but will remain in RX mode. The microcontroller will need to decide on the appropriate subsequent action, depending upon information such as an interrupt generated by CRC, packet valid, or preamble detect.



## 5. Internal Functional Blocks

This section provides an overview some of the key blocks of the internal radio architecture.

### 5.1. RX LNA

The input frequency range for the LNA is 433/868/915MHz band. The LNA provides gain with a noise figure low enough to suppress the noise of the following stages. The LNA has one step of gain control which is controlled by the analog gain control (AGC) algorithm. The AGC algorithm adjusts the gain of the LNA and PGA so the receiver can handle signal levels from sensitivity to +5 dBm with optimal performance.

### 5.2. RX I-Q Mixer

The output of the LNA is fed internally to the input of the receive mixer. The receive mixer is implemented as an I-Q mixer that provides both I and Q channel outputs to the programmable gain amplifier. The mixer consists of two double-balanced mixers whose RF inputs are driven in parallel, local oscillator (LO) inputs are driven in quadrature, and separate I and Q Intermediate Frequency (IF) outputs drive the programmable gain amplifier. The receive LO signal is supplied by an integrated VCO and PLL synthesizer operating between 240–960 MHz. The necessary quadrature LO signals are derived from the divider at the VCO output.

### 5.3. Programmable Gain Amplifier

The programmable gain amplifier (PGA) provides the necessary gain to boost the signal level into the dynamic range of the ADC. The PGA must also have enough gain switching to allow for large input signals to ensure a linear RSSI range up to –20 dBm. The PGA has steps of 3 dB which are controlled by the AGC algorithm in the digital modem.

### 5.4. ADC

The amplified IQ IF signals are digitized using an Analog-to-Digital Converter (ADC), which allows for low current consumption and high dynamic range. The bandpass response of the ADC provides exceptional rejection of out of band blockers.

### 5.5. Digital Modem

Using high-performance ADCs allows channel filtering, image rejection, and demodulation to be performed in the digital domain, resulting in reduced area while increasing flexibility. The digital modem performs the following functions:

- Channel selection filter
- RX demodulation
- AGC
- Preamble detector
- Invalid preamble detector
- Radio signal strength indicator (RSSI)
- Automatic frequency compensation (AFC)
- Packet handling including EZMac™ features
- 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, and OOK. The channel filter can be configured to support bandwidths ranging from 620 kHz down to 2.6 kHz. A large variety of data rates are supported ranging from 0.123 up to 256 kbps. The AGC algorithm is implemented digitally using an advanced control loop optimized for fast response time.

The configurable preamble detector is used to improve the reliability of the sync-word detection. The sync-word detector is only enabled when a valid preamble is detected, significantly reducing the probability of false detection.

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), and carrier sense (CS) functionality.

Frequency mistuning caused by crystal inaccuracies can be compensated by enabling the digital automatic frequency control (AFC) in receive mode.

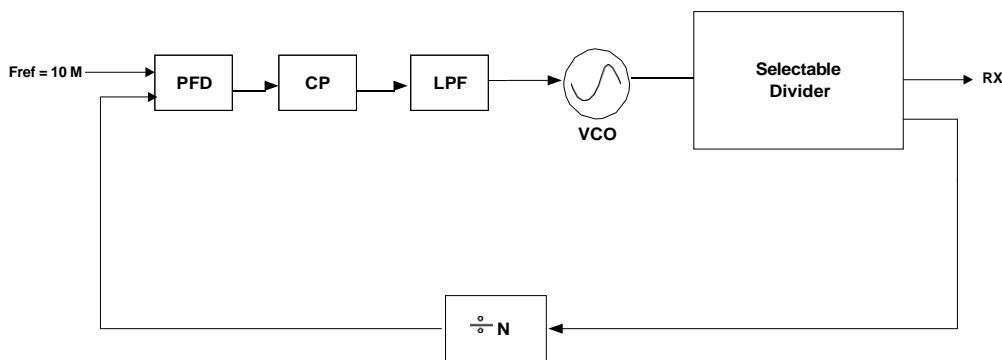
A comprehensive programmable packet handler including key features of EZMac™ is integrated to create a variety of communication topologies ranging from peer-to-peer networks to mesh networks. The extensive programmability of the packet header allows for advanced packet filtering which in turn enables a mix of broadcast, group, and point-to-point communication.

A wireless communication channel can be corrupted by noise and interference, and it is therefore 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.

## 5.6. Synthesizer

An integrated Sigma Delta ( $\Sigma\Delta$ ) Fractional-N PLL synthesizer capable of operating from 240–960 MHz is provided on-chip. Using a  $\Sigma\Delta$  synthesizer has many advantages; it provides flexibility in choosing data rate, deviation, channel frequency, and channel spacing.

The PLL and  $\Delta$ - $\Sigma$  modulator scheme is designed to support any desired frequency and channel spacing in the range from 240–960 MHz with a frequency resolution of 156.25 Hz (Low band) or 312.5 Hz (High band).



**Figure 9. PLL Synthesizer Block Diagram**

The reference frequency to the PLL is 10 MHz. The PLL utilizes a differential L-C VCO, with integrated on-chip inductors. The output of the VCO is followed by a configurable divider which will divide down the signal to the desired output frequency band. The modulus of the variable divide-by-N divider stage is controlled dynamically by the output from the  $\Delta$ - $\Sigma$  modulator. The tuning resolution is sufficient to tune to the commanded frequency with a maximum accuracy of 312.5 Hz anywhere in the range between 240–960 MHz.

### 5.6.1. VCO

The output of the VCO is automatically divided down to the correct output frequency depending on the `hbsel` and `fb[4:0]` fields in "Register 75h. Frequency Band Select." In receive mode, the LO frequency is automatically shifted downwards by the IF frequency of 937.5 kHz, allowing receive operation on the same frequency. The VCO integrates the resonator inductor and tuning varactor, so no external VCO components are required.

The VCO uses a capacitance bank to cover the wide frequency range specified. The capacitance bank will automatically be calibrated every time the synthesizer is enabled. In certain fast hopping applications this might not be desirable so the VCO calibration may be skipped by setting the appropriate register.

## 5.7. Crystal Oscillator

The RFM31B includes an integrated 30MHz crystal oscillator with a fast start-up time of less than 600 s. A parallel resonant 30MHz crystal is used on the module. The design is differential with the required crystal load capacitance integrated on-chip to minimize the number of external components.

The crystal load capacitance can be digitally programmed to accommodate crystals with various load capacitance requirements and to adjust the frequency of the crystal oscillator. The tuning of the crystal load capacitance is programmed through the xlc[6:0] field of "Register 09h. 30 MHz Crystal Oscillator Load Capacitance." The total internal capacitance is 12.5 pF and is adjustable in approximately 127 steps (97fF/step). The xtalshift bit provides a coarse shift in frequency but is not binary with xlc[6:0].

The crystal frequency adjustment can be used to compensate for crystal production tolerances. Utilizing the on-chip temperature sensor and suitable control software, the temperature dependency of the crystal can be canceled.

The typical value of the total on-chip capacitance C<sub>int</sub> can be calculated as follows:

$$C_{int} = 1.8 \text{ pF} + 0.085 \text{ pF} \times xlc[6:0] + 3.7 \text{ pF} \times xtalshift$$

Note that the coarse shift bit xtalshift is not binary with xlc[6:0]. The total load capacitance C<sub>load</sub> seen by the crystal can be calculated by adding the sum of all external parasitic PCB capacitances C<sub>ext</sub> to C<sub>int</sub>. If the maximum value of C<sub>int</sub> (16.3 pF) is not sufficient, an external capacitor can be added for exact tuning. Additional information on calculating C<sub>ext</sub> and crystal selection guidelines is provided.

If AFC is disabled then the synthesizer frequency may be further adjusted by programming the Frequency Offset field fo[9:0] in "Register 73h. Frequency Offset 1" and "Register 74h. Frequency Offset 2", as discussed in "3.5. Frequency Control"

The crystal oscillator frequency is divided down internally and may be output to the microcontroller through one of the GPIO pins for use as the System Clock. In this fashion, only one crystal oscillator is required for the entire system and the BOM cost is reduced. The available clock frequencies and GPIO configuration are discussed further in "8.2. Microcontroller Clock"

| Add | R/W | Function/Description                | D7        | D6     | D5     | D4     | D3     | D2     | D1     | D0     | POR Def. |
|-----|-----|-------------------------------------|-----------|--------|--------|--------|--------|--------|--------|--------|----------|
| 09  | R/W | Crystal Oscillator Load Capacitance | xtalshift | xlc[6] | xlc[5] | xlc[4] | xlc[3] | xlc[2] | xlc[1] | xlc[0] | 7Fh      |

## 5.8. Regulators

There are a total of six regulators integrated onto the RFM31B . With the exception of the digital regulator, all regulators are designed to operate with only internal decoupling. All regulators are designed to operate with an input supply voltage from +1.8 to +3.6V. A supply voltage should only be connected to the VDD pins.

## 6. Data Handling and Packet Handler

The internal modem is designed to operate with a packet including a 10101... preamble structure. To configure the modem to operate with packet formats without a preamble or other legacy packet structures contact customer support.

### 6.1. RX FIFO

A 64 byte FIFO is integrated into the chip for RX, as shown in Figure 11. "Register 7Fh. FIFO Access" is used to access the FIFO. A burst read, as described in "3.1. Serial Peripheral Interface (SPI)" , from address 7Fh will read data from the RX FIFO.

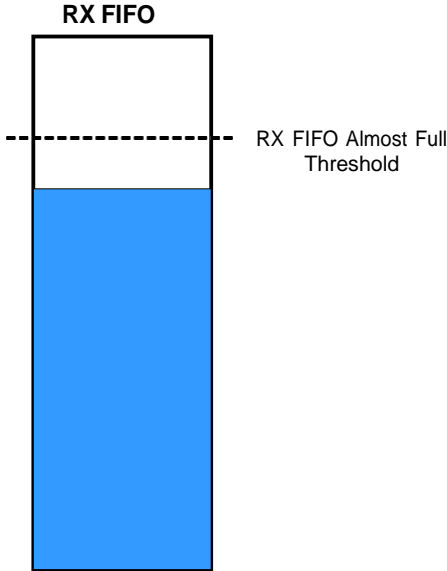


Figure 10. FIFO Threshold

| Add | R/W | Function/Description           | D7        | D6        | D5        | D4    | D3       | D2    | D1     | D0       | POR Def. |
|-----|-----|--------------------------------|-----------|-----------|-----------|-------|----------|-------|--------|----------|----------|
| 08  | R/W | Operating & Function Control 2 | antdiv[2] | antdiv[1] | antdiv[0] | rxmpk | Reserved | enldm | ffclrx | Reserved | 00h      |

The RX FIFO has one programmable threshold called the FIFO Almost Full Threshold, rxafthr[5:0]. When the incoming RX data crosses the Almost Full Threshold an interrupt will be generated to the microcontroller via the nIRQ pin. The microcontroller will then need to read the data from the RX FIFO.

| Add | R/W | Function/Description | D7       | D6       | D5         | D4         | D3         | D2         | D1         | D0         | POR Def. |
|-----|-----|----------------------|----------|----------|------------|------------|------------|------------|------------|------------|----------|
| 7E  | R/W | RX FIFO Control      | Reserved | Reserved | rxafthr[5] | rxafthr[4] | rxafthr[3] | rxafthr[2] | rxafthr[1] | rxafthr[0] | 37h      |

The RX FIFO may be cleared or reset with the ffclrx bit in "Register 08h. Operating Mode and Function Control 2,". All interrupts may be enabled by setting the Interrupt Enabled bits in "Register 05h. Interrupt Enable 1" and "Register 06h. Interrupt Enable 2,". If the interrupts are not enabled the function will not generate an interrupt on the nIRQ pin but the bits will still be read correctly in the Interrupt Status registers.

## 6.2. Packet Configuration

When using the FIFO, automatic packet handling may be enabled for the RX mode. "Register 30h. Data Access Control" through "Register 39h.

The general packet structure is shown in Figure 12. The length of each field is shown below the field. The preamble pattern is always a series of alternating ones and zeroes, starting with a zero. All the fields have programmable lengths to accommodate different applications. The most common CRC polynomials are available for selection.



**Figure 11. Packet Structure**

An overview of the packet handler configuration registers is shown in Table 13.

## 6.3. Packet Handler RX Mode

### 6.3.1. Packet Handler Disabled

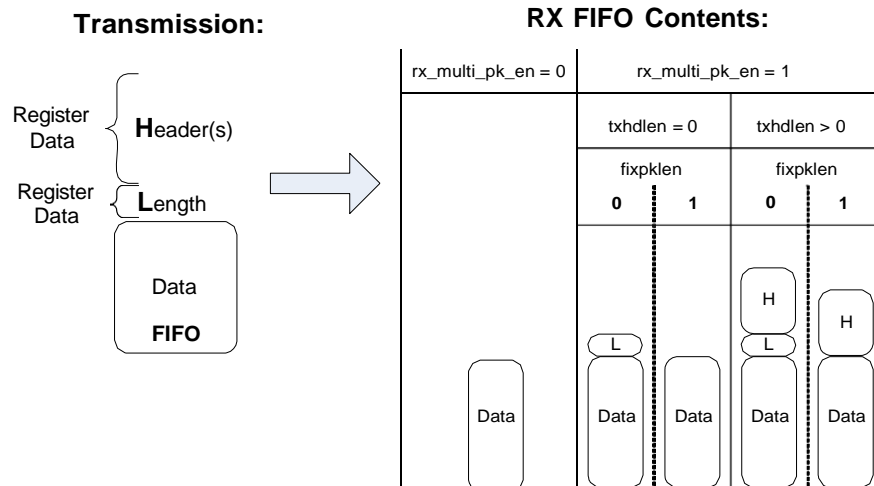
When the packet handler is disabled certain fields in the received packet are still required. Proper modem operation requires preamble and sync when the FIFO is being used, as shown in Figure 14. Bits after sync will be treated as raw data with no qualification. This mode allows for the creation of a custom packet handler when the automatic qualification parameters are not sufficient. Manchester encoding is supported but data whitening, CRC, and header checks are not



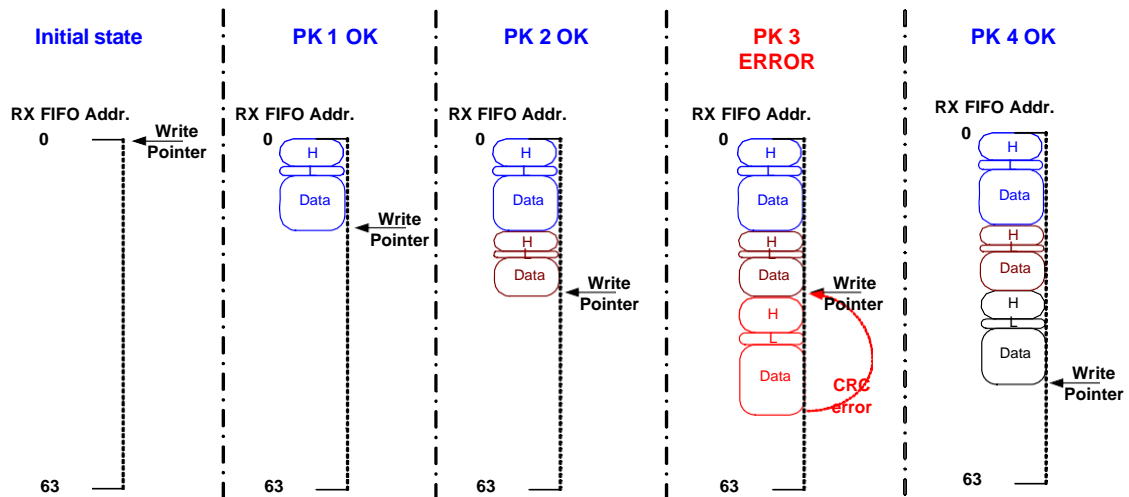
**Figure 12. Required RX Packet Structure with Packet Handler Disabled**

### 6.3.2. Packet Handler Enabled

When the packet handler is enabled, all the fields of the packet structure need to be configured. The receive FIFO can be configured to handle packets of fixed or variable length with or without a header. If multiple packets are desired to be stored in the FIFO, then there are options available for the different fields that will be stored into the FIFO. Figure 15 demonstrates the options and settings available when multiple packets are enabled. Figure 16 demonstrates the operation of fixed packet length and correct/incorrect packets.



**Figure 13. Multiple Packets in RX Packet Handler**



**Figure 14. Multiple Packets in RX with CRC or Header Error**

Table 12. Packet Handler Registers

| Add   | R/W | Function/Description       | D7         | D6         | D5         | D4         | D3         | D2         | D1         | D0         | POR Def. |
|-------|-----|----------------------------|------------|------------|------------|------------|------------|------------|------------|------------|----------|
| 30    | R/W | Data Access Control        | enpacrx    | lsbfrst    | crocdonly  | *Reserved  | Reserved   | encrc      | crc[1]     | crc[0]     | 1Dh      |
| 31    | R   | EzMAC status               | Reserved   | rxcrcl     | pkscrch    | pkrx       | pkvalid    | crccerror  | Reserved   | Reserved   | —        |
| 32    | R/W | Header Control 1           | bcen[3]    | enbcast[2] | enbcast[1] | enbcast[0] | hdch[3]    | hdch[2]    | hdch[1]    | hdch[0]    | 0Ch      |
| 33    | R/W | Header Control 2           | skipsyn    | hdlen[2]   | hdlen[1]   | hdlen[0]   | fixpklen   | syncden[1] | syncden[0] | prealen[8] | 22h      |
| 34    | R/W | Preamble Length            | prealen[7] | prealen[6] | prealen[5] | prealen[4] | prealen[3] | prealen[2] | prealen[1] | prealen[0] | 07h      |
| 35    | R/W | Preamble Detection Control | preath[4]  | preath[3]  | preath[2]  | preath[1]  | preath[0]  | Reserved   | Reserved   | Reserved   | 20h      |
| 36    | R/W | Sync Word 3                | sync[31]   | sync[30]   | sync[29]   | sync[28]   | sync[27]   | sync[26]   | sync[25]   | sync[24]   | 2Dh      |
| 37    | R/W | Sync Word 2                | sync[23]   | sync[22]   | sync[21]   | sync[20]   | sync[19]   | sync[18]   | sync[17]   | sync[16]   | D4h      |
| 38    | R/W | Sync Word 1                | sync[15]   | sync[14]   | sync[13]   | sync[12]   | sync[11]   | sync[10]   | sync[9]    | sync[8]    | 00h      |
| 39    | R/W | Sync Word 0                | sync[7]    | sync[6]    | sync[5]    | sync[4]    | sync[3]    | sync[2]    | sync[1]    | sync[0]    | 00h      |
| 3A–3E | R/W | Reserved                   | Reserved   |            |            |            |            |            |            |            |          |
| 3F    | R/W | Check Header 3             | chhd[31]   | chhd[30]   | chhd[29]   | chhd[28]   | chhd[27]   | chhd[26]   | chhd[25]   | chhd[24]   | 00h      |
| 40    | R/W | Check Header 2             | chhd[23]   | chhd[22]   | chhd[21]   | chhd[20]   | chhd[19]   | chhd[18]   | chhd[17]   | chhd[16]   | 00h      |
| 41    | R/W | Check Header 1             | chhd[15]   | chhd[14]   | chhd[13]   | chhd[12]   | chhd[11]   | chhd[10]   | chhd[9]    | chhd[8]    | 00h      |
| 42    | R/W | Check Header 0             | chhd[7]    | chhd[6]    | chhd[5]    | chhd[4]    | chhd[3]    | chhd[2]    | chhd[1]    | chhd[0]    | 00h      |
| 43    | R/W | Header Enable 3            | hden[31]   | hden[30]   | hden[29]   | hden[28]   | hden[27]   | hden[26]   | hden[25]   | hden[24]   | FFh      |
| 44    | R/W | Header Enable 2            | hden[23]   | hden[22]   | hden[21]   | hden[20]   | hden[19]   | hden[18]   | hden[17]   | hden[16]   | FFh      |
| 45    | R/W | Header Enable 1            | hden[15]   | hden[14]   | hden[13]   | hden[12]   | hden[11]   | hden[10]   | hden[9]    | hden[8]    | FFh      |
| 46    | R/W | Header Enable 0            | hden[7]    | hden[6]    | hden[5]    | hden[4]    | hden[3]    | hden[2]    | hden[1]    | hden[0]    | FFh      |
| 47    | R   | Received Header 3          | rxhd[31]   | rxhd[30]   | rxhd[29]   | rxhd[28]   | rxhd[27]   | rxhd[26]   | rxhd[25]   | rxhd[24]   | —        |
| 48    | R   | Received Header 2          | rxhd[23]   | rxhd[22]   | rxhd[21]   | rxhd[20]   | rxhd[19]   | rxhd[18]   | rxhd[17]   | rxhd[16]   | —        |
| 49    | R   | Received Header 1          | rxhd[15]   | rxhd[14]   | rxhd[13]   | rxhd[12]   | rxhd[11]   | rxhd[10]   | rxhd[9]    | rxhd[8]    | —        |
| 4A    | R   | Received Header 0          | rxhd[7]    | rxhd[6]    | rxhd[5]    | rxhd[4]    | rxhd[3]    | rxhd[2]    | rxhd[1]    | rxhd[0]    | —        |
| 4B    | R   | Received Packet Length     | rxplen[7]  | rxplen[6]  | rxplen[5]  | rxplen[4]  | rxplen[3]  | rxplen[2]  | rxplen[1]  | rxplen[0]  | —        |

## 6.4. Data Whitening, Manchester Encoding, and CRC

Data whitening can be used to avoid extended sequences of 0s or 1s in the transmitted data stream to achieve a more uniform spectrum. When enabled, the payload data bits are XORed with a pseudorandom sequence output from the built-in PN9 generator. The generator is initialized at the beginning of the payload. The receiver recovers the original data by repeating this operation. Manchester encoding can be used to ensure a dc-free transmission and good synchronization properties. When Manchester encoding is used, the effective datarate is unchanged but the actual datarate (preamble length, etc.) is doubled due to the nature of the encoding. The effective datarate when using Manchester encoding is limited to 128 kbps. The implementation of Manchester encoding is shown in Figure 16. Data whitening and Manchester encoding can be selected with "Register 70h. Modulation Mode Control 1". The CRC is configured via "Register 30h. Data Access Control." Figure 15 demonstrates the portions of the packet which have Manchester encoding, data whitening, and CRC applied. CRC can be applied to only the data portion of the packet or to the data, packet length and header fields. Figure 16 provides an example of how the Manchester encoding is done and also the use of the Manchester invert (enmaniv) function.

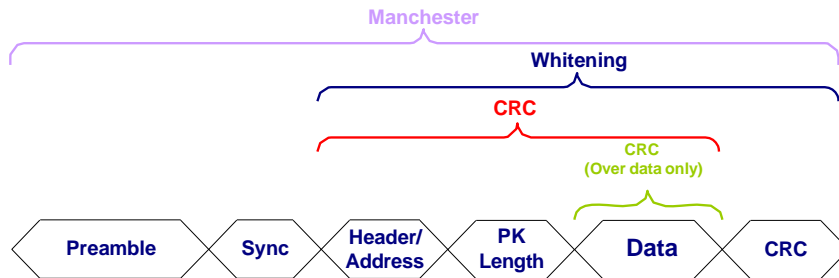


Figure 15. Operation of Data Whitening, Manchester Encoding, and CRC

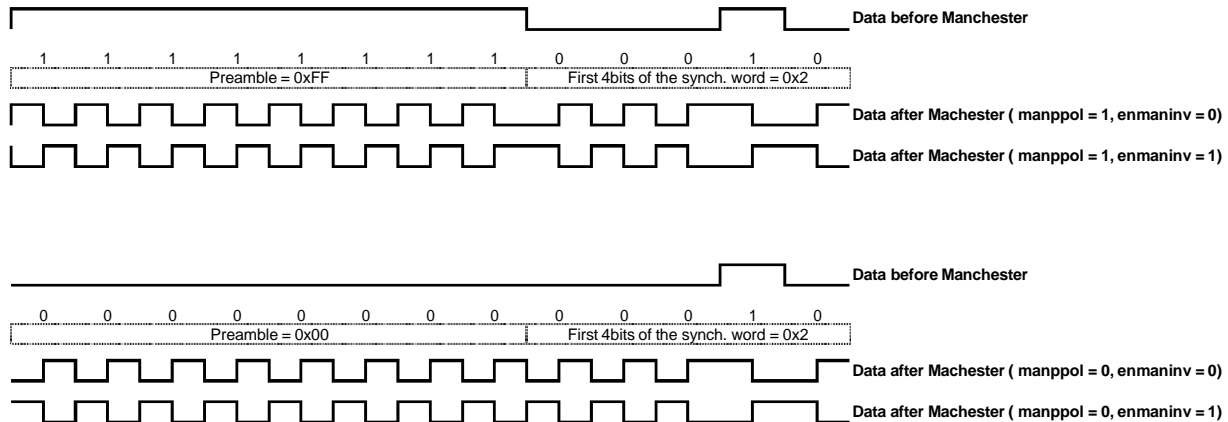


Figure 16. Manchester Coding Example

## 6.5. Preamble Detector

The RFM31B has integrated automatic preamble detection. The preamble length is configurable from 1–256 bytes using the prealen[7:0] field in "Register 33h. Header Control 2" and "Register 34h. Preamble Length," as described in "6.2. Packet Configuration." The preamble detection threshold, preath[4:0] as set in "Register 35h. Preamble Detection Control 1", is in units of 4 bits. The preamble detector searches for a preamble pattern with a length of preath[4:0].

If a false preamble detect occurs, the receiver will continue searching for the preamble when no sync word is detected. When a false preamble detect occurs, the receiver will continue searching for the preamble when no sync word is detected. Once preamble is detected (false or real) then the part will then start searching for sync. If no sync occurs then a timeout will occur and the device will initiate search for preamble again. The timeout period is defined as the sync word length plus four bits and will start after a non-preamble pattern is recognized after a valid preamble detection. The preamble detector output may be programmed onto one of the GPIO or read in the interrupt status registers.



## 6.6. Preamble Length

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 will depend 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 is programmed in register 35h. 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.

It is possible to use the RFM31B in a raw mode without the requirement for a 101010 preamble. Contact customer support for further details.

**Table 13. Minimum Receiver Settling Time**

| Mode   | Approximate Receiver Settling Time | Recommended preamble length with 8-bit detection threshold | Recommended preamble length with 20-bit detection threshold |
|--|------------------------------------|--|---|
| (G)FSK AFC Disabled                            | 1 byte                             | 20 bits  | 32 bits   |
| (G)FSK AFC Enabled                             | 2 byte                             | 28 bits  | 40 bits   |
| (G)FSK AFC Disabled +Antenna Diversity Enabled | 1 byte                             | —  | 64 bits   |
| (G)FSK AFC Enabled +Antenna Diversity Enabled  | 2 byte                             | —  | 8 byte  |
| OOK  | 2 byte                             | 3 byte   | 4 byte  |
| OOK + Antenna Diversity Enabled                | 8 byte                             | —  | 8 byte  |

**Note:** The recommended preamble length and preamble detection threshold listed above are to achieve 0% PER. They may be shortened when occasional packet errors are tolerable.

## 6.7. Invalid Preamble Detector

When scanning channels in a frequency hopping system it is desirable to determine if a channel is valid in the minimum amount of time. The preamble detector can output an invalid preamble detect signal which can be used to identify the channel as invalid. After a configurable time set in Register 60h[7:4], an invalid preamble detect signal is asserted indicating an invalid channel. The period for evaluating the signal for invalid preamble is defined as  $(inv\_pre\_th[3:0] \times 4) \times \text{Bit Rate Period}$ . The preamble detect and invalid preamble detect signals are available in "Register 03h. Interrupt/Status 1" and "Register 04h. Interrupt/Status 2" .

## 6.8. Synchronization Word Configuration

The synchronization word length for RX can be configured in Reg 33h, `synclen[1:0]`. The expected or transmitted sync word can be configured from 1 to 4 bytes as defined below:

- `synclen[1:0] = 00`—Expected Synchronization Word (sync word) 3.
- `synclen[1:0] = 01`—Expected Synchronization Word 3 first, followed by sync word 2.
- `synclen[1:0] = 10`—Expected Synchronization Word 3 first, followed by sync word 2, followed by sync word 1.
- `synclen[1:0] = 11`—Expected Synchronization Word 3 first, followed by sync word 2, followed by sync word 1, followed by sync word 0.

The sync is transmitted or expected in the following sequence: sync 3→sync 2→sync 1→sync 0. The sync word values can be programmed in Registers 36h–39h. After preamble detection, the part will search for sync for a fixed period of time. If a sync is not recognized in this period, a timeout will occur, and the search for preamble will be re-

initiated. The timeout period after preamble detections is defined as the value programmed into the sync word length plus four additional bits.

## 6.9. Receive Header Check

The header check is designed to support 1–4 bytes and broadcast headers. The header length needs to be set in register 33h, `hdlen[2:0]`. The headers to be checked need to be set in register 32h, `hdch[3:0]`. For instance, there can be four bytes of header in the packet structure but only one byte of the header is set to be checked (i.e., header 3). For the headers that are set to be checked, the expected value of the header should be programmed in `chhd[31:0]` in Registers 3F–42. The individual bits within the selected bytes to be checked can be enabled or disabled with the header enables, `hden[31:0]` in Registers 43–46. For example, if you want to check all bits in header 3 then `hden[31:24]` should be set to FF but if only the last 4 bits are desired to be checked then it should be set to 00001111 (0F). Broadcast headers can also be programmed by setting `bcen[3:0]` in Register 32h. For broadcast header check the value may be either “FFh” or the value stored in the Check Header register. A logic equivalent of the header check for Header 3 is shown in Figure 17. A similar logic check will be done for Header 2, Header 1, and Header 0 if enabled.

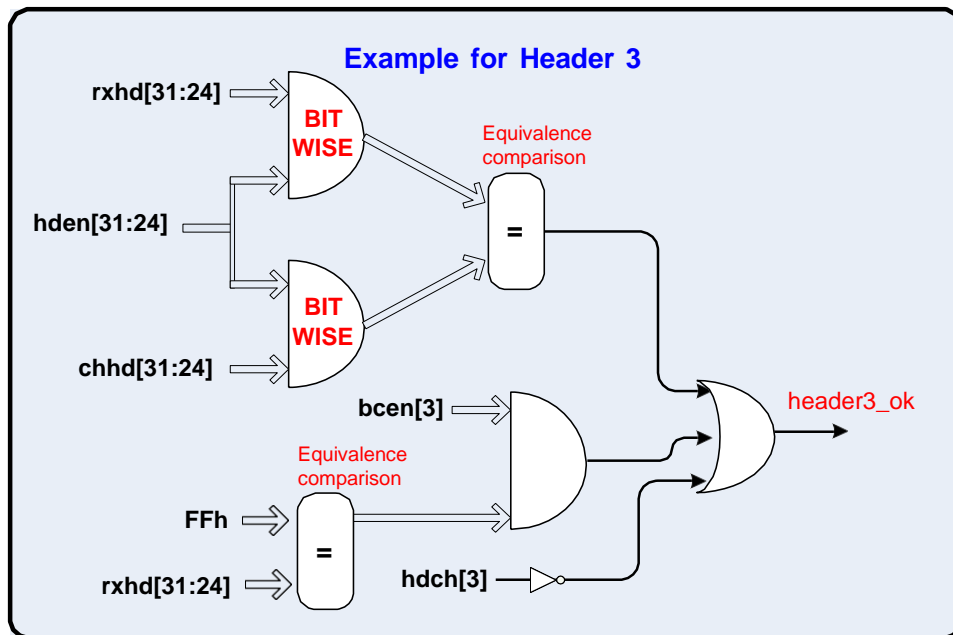


Figure 17. Header

## **7. RX Modem Configuration**

A Microsoft Excel (WDS) parameter calculator or Wireless Development Suite (WDS) calculator is provided to determine the proper settings for the modem. The calculator can be found on [www.silabs.com](http://www.silabs.com) or on the CD provided with the demo kits. An application note is available to describe how to use the calculator and to provide advanced descriptions of the modem settings and calculations.

### **7.1. Modem Settings for FSK and GFSK**

The modem performs channel selection and demodulation in the digital domain. The channel filter bandwidth is configurable from 2.6 to 620 kHz. The receiver channel bandwidth is set depending on the data rate and modulation index via registers 1C–25h. The modulation index is equal to 2 times the peak deviation divided by the data rate ( $R_b$ ).

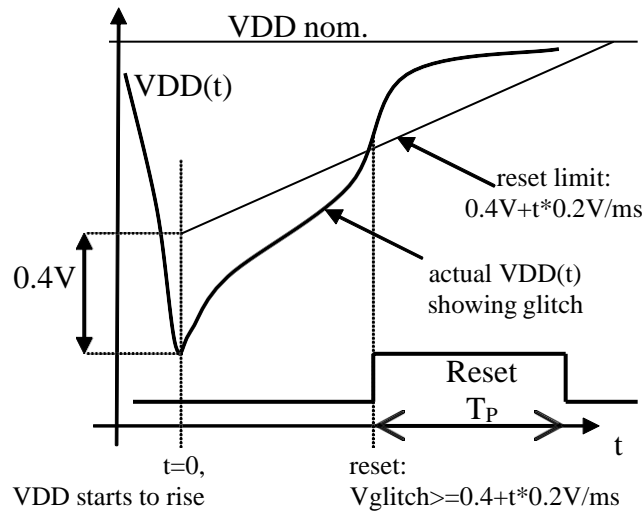
When Manchester coding is disabled, the required channel filter bandwidth is calculated as  $BW = 2F_d + R_b$  where  $F_d$  is the frequency deviation and  $R_b$  is the data rate.

## 8. Auxiliary Functions

### 8.1. Smart Reset

The RFM31B contains an enhanced integrated SMART RESET or POR circuit. The POR circuit contains both a classic level threshold reset as well as a slope detector POR. This reset circuit was designed to produce a reliable reset signal under any circumstances. Reset will be initiated if any of the following conditions occur:

- Initial power on, VDD starts from gnd: reset is active till VDD reaches  $V_{RR}$  (see table);
- When VDD decreases below  $V_{LD}$  for any reason: reset is active till VDD reaches  $V_{RR}$ ;
- A software reset via “Register 08h. Operating Mode and Function Control 2,” : reset is active for time  $T_{SWRST}$
- On the rising edge of a VDD glitch when the supply voltage exceeds the following time functioned limit:



**Figure 18. POR Glitch Parameters**

**Table 14. POR Parameters**

| Parameter              | Symbol | Comment                                     | Min  | Typ | Max  | Unit |
|------------------------|--------|---|------|-----|------|------|
| Release Reset Voltage  | VRR    |   | 0.85 | 1.3 | 1.75 | V    |
| Power-On VDD Slope     | SVDD   | tested VDD slope region                     | 0.03 |     | 300  | V/ms |
| Low VDD Limit          | VLD    | $VLD < VRR$ is guaranteed                   | 0.7  | 1   | 1.3  | V    |
| Software Reset Pulse   | TSWRST |   | 50   |     | 470  | us   |
| Threshold Voltage      | VTSD   |   |      | 0.4 |      | V    |
| Reference Slope        | k      |   |      | 0.2 |      | V/ms |
| VDD Glitch Reset Pulse | TP     | Also occurs after SDN, and initial power on | 5    | 16  | 40   | ms   |

The reset will initialize all registers to their default values. The reset signal is also available for output and use by the microcontroller by using the default setting for GPIO\_0. The inverted reset signal is available by default on GPIO\_1.

## 8.2. Microcontroller Clock

The 30 MHz crystal oscillator frequency is divided down internally and may be output to the microcontroller through GPIO2. This feature is useful to lower BOM cost by using only one crystal in the system. The system clock frequency is selectable from one of 8 options, as shown below. Except for the 32.768 kHz option, all other frequencies are derived by dividing the crystal oscillator frequency. The 32.768 kHz clock signal is derived from an internal RC oscillator or an external 32 kHz crystal. The default setting for GPIO2 is to output the microcontroller clock signal with a frequency of 1 MHz.

| Add | R/W | Function/Description         | D7 | D6 | D5      | D4      | D3    | D2      | D1      | D0      | POR Def. |
|-----|-----|------------------------------|----|----|---------|---------|-------|---------|---------|---------|----------|
| 0A  | R/W | Microcontroller Output Clock |    |    | clkt[1] | clkt[0] | enlfc | mclk[2] | mclk[1] | mclk[0] | 06h      |

| mclk[2:0] | Clock Frequency |
|-----------|-----------------|
| 000       | 30 MHz          |
| 001       | 15 MHz          |
| 010       | 10 MHz          |
| 011       | 4 MHz           |
| 100       | 3 MHz           |
| 101       | 2 MHz           |
| 110       | 1 MHz           |
| 111       | 32.768 kHz      |

If the microcontroller clock option is being used there may be the need of a system clock for the microcontroller while the RFM31B is in SLEEP mode. Since the crystal oscillator is disabled in SLEEP mode in order to save current, the low-power 32.768 kHz clock can be automatically switched to become the microcontroller clock. This feature is called enable low frequency clock and is enabled by the enlfc bit in "Register 0Ah. Microcontroller Output Clock." When enlfc = 1 and the chip is in SLEEP mode then the 32.768 kHz clock will be provided to the microcontroller as the system clock, regardless of the setting of mclk[2:0]. For example, if mclk[2:0] = 000, 30 MHz will be provided through the GPIO output pin to the microcontroller as the system clock in all IDLE or RX states. When the chip enters SLEEP mode, the system clock will automatically switch to 32.768 kHz from the RC oscillator or 32.768 XTAL.

Another available feature for the microcontroller clock is the clock tail, clkt[1:0] in "Register 0Ah. Microcontroller Output Clock." If the low frequency clock feature is not enabled (enlfc = 0), then the system clock to the microcontroller is disabled in SLEEP mode. However, it may be useful to provide a few extra cycles for the microcontroller to complete its operation prior to the shutdown of the system clock signal. Setting the clkt[1:0] field will provide additional cycles of the system clock before it shuts off.

| clkt[1:0] | Clock Tail |
|-----------|------------|
| 00        | 0 cycles   |
| 01        | 128 cycles |
| 10        | 256 cycles |
| 11        | 512 cycles |

If an interrupt is triggered, the microcontroller clock will remain enabled regardless of the selected mode. As soon as the interrupt is read the state machine will then move to the selected mode. The minimum current consumption will not be achieved until the interrupt is read. For instance, if the chip is commanded to SLEEP mode but an interrupt has occurred the 30 MHz XTAL will not be disabled until the interrupt has been cleared.

# RFM31 B

## 8.3. General Purpose ADC

An 8-bit SAR ADC is integrated for general purpose use, as well as for digitizing the on-chip temperature sensor reading. Registers 0Fh "ADC Configuration", 10h "Sensor Offset" and 4Fh "Amplifier Offset" can be used to configure the ADC operation.

Every time an ADC conversion is desired, bit 7 "adcstart/adcbusy" in "Register 1Fh. Clock Recovery Gearshift Override" must be set to 1. This is a self clearing bit that will be reset to 0 at the end of the conversion cycle of the ADC. The conversion time for the ADC is 350  $\mu$ s. After this time or when the "adcstart/adcbusy" bit is cleared, then the ADC value may be read out of register 11h "ADC Value".

The architecture of the ADC is shown in Figure 19. The signal and reference inputs of the ADC are selected by `adcsel[2:0]` and `adcref[1:0]` in "Register 0Fh. ADC Configuration," respectively. The default setting is to read out the temperature sensor using the bandgap voltage (VBG) as reference. With the VBG reference the input range of the ADC is from 0–1.02 V with an LSB resolution of 4 mV (1.02/255). Changing the ADC reference will change the LSB resolution accordingly.

A differential multiplexer and amplifier are provided for interfacing external bridge sensors. The gain of the amplifier is selectable by `adcgain[1:0]` in Register 0Fh. The majority of sensor bridges have supply voltage ( $V_{DD}$ ) dependent gain and offset. The reference voltage of the ADC can be changed to either  $V_{DD}/2$  or  $V_{DD}/3$ . A programmable  $V_{DD}$  dependent offset voltage can be added using `soffs[3:0]` in register 10h.

See "General Purpose ADC Configuration" for more details on the usage of the general purpose ADC.

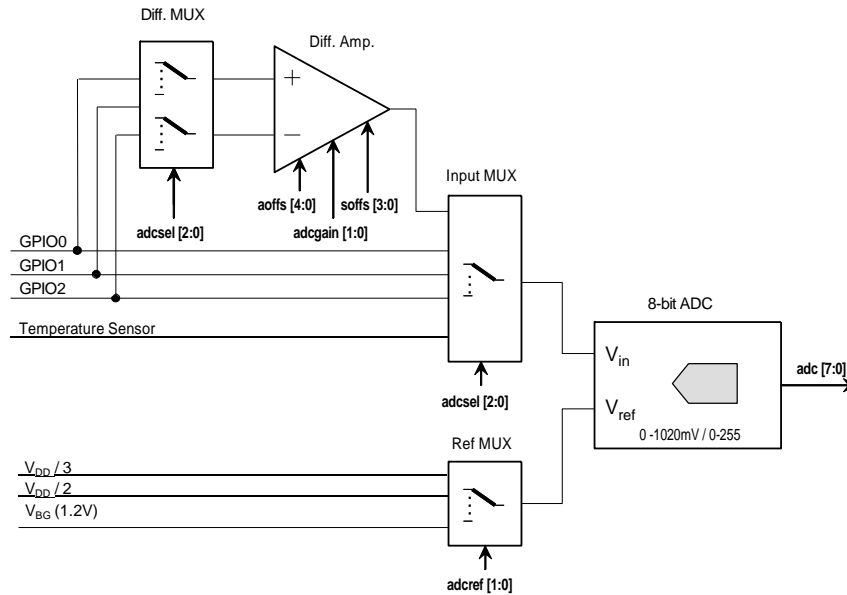


Figure 19. General Purpose ADC Architecture

| Add | R/W | Function/Description | D7               | D6        | D5        | D4        | D3        | D2        | D1         | D0         | POR Def. |
|-----|-----|----------------------|------------------|-----------|-----------|-----------|-----------|-----------|------------|------------|----------|
| 0F  | R/W | ADC Configuration    | adcstart/adcbusy | adcsel[2] | adcsel[1] | adcsel[0] | adcref[1] | adcref[0] | adcgain[1] | adcgain[0] | 00h      |
| 10  | R/W | Sensor Offset        |                  |           |           |           | soffs[3]  | soffs[2]  | soffs[1]   | soffs[0]   | 00h      |
| 11  | R   | ADC Value            | adc[7]           | adc[6]    | adc[5]    | adc[4]    | adc[3]    | adc[2]    | adc[1]     | adc[0]     | —        |

## 8.4. Temperature Sensor

An integrated on-chip analog temperature sensor is available. The temperature sensor will be automatically enabled when the temperature sensor is selected as the input of the ADC or when the analog temp voltage is selected on the analog test bus. The temperature sensor value may be digitized using the general-purpose ADC and read out over the SPI through "Register 10h. ADC Sensor Amplifier Offset." The range of the temperature sensor is configurable. Table 15 lists the settings for the different temperature ranges and performance.

To use the Temp Sensor:

1. Set the input for ADC to the temperature sensor, "Register 0Fh. ADC Configuration"—`adcsel[2:0] = 000`
2. Set the reference for ADC, "Register 0Fh. ADC Configuration"—`adcref[1:0] = 00`
3. Set the temperature range for ADC, "Register 12h. Temperature Sensor Calibration"—`tsrange[1:0]`
4. Set `entsoffs = 1`, "Register 12h. Temperature Sensor Calibration"
5. Trigger ADC reading, "Register 0Fh. ADC Configuration"—`adcstart = 1`
6. Read temperature value—Read contents of "Register 11h. ADC Value"

| Add | R/W | Function/Description       | D7                      | D6                      | D5                     | D4                     | D3                     | D2                     | D1                      | D0                      | POR Def. |
|-----|-----|----------------------------|-------------------------|-------------------------|------------------------|------------------------|------------------------|------------------------|-------------------------|-------------------------|----------|
| 12  | R/W | Temperature Sensor Control | <code>tsrange[1]</code> | <code>tsrange[0]</code> | <code>entsoffs</code>  | <code>entstrim</code>  | <code>tstrim[3]</code> | <code>tstrim[2]</code> | <code>vbgtrim[1]</code> | <code>vbgtrim[0]</code> | 20h      |
| 13  | R/W | Temperature Value Offset   | <code>tvoffs[7]</code>  | <code>tvoffs[6]</code>  | <code>tvoffs[5]</code> | <code>tvoffs[4]</code> | <code>tvoffs[3]</code> | <code>tvoffs[2]</code> | <code>tvoffs[1]</code>  | <code>tvoffs[0]</code>  | 00h      |

**Table 15. Temperature Sensor Range**

| <code>entoff</code> | <code>tsrange[1]</code> | <code>tsrange[0]</code> | Temp. range | Unit | Slope   | ADC8 LSB |
|---------------------|-------------------------|-------------------------|-------------|------|---------|----------|
| 1                   | 0                       | 0                       | -64 ... 64  | °C   | 8 mV/°C | 0.5 °C   |
| 1                   | 0                       | 1                       | -64 ... 192 | °C   | 4 mV/°C | 1 °C     |
| 1                   | 1                       | 0                       | 0 ... 128   | °C   | 8 mV/°C | 0.5 °C   |
| 1                   | 1                       | 1                       | -40 ... 216 | °F   | 4 mV/°F | 1 °F     |
| 0*                  | 1                       | 0                       | 0 ... 341   | °K   | 3 mV/°K | 1.333 °K |

**\*Note:** Absolute temperature mode, no temperature shift. This mode is only for test purposes. POR value of `EN_TOFF` is 1.

The slope of the temperature sensor is very linear and monotonic. For absolute accuracy better than 10 °C calibration is necessary. The temperature sensor may be calibrated by setting `entsoffs = 1` in "Register 12h. Temperature Sensor Control" and setting the offset with the `tvoffs[7:0]` bits in "Register 13h. Temperature Value Offset." This method adds a positive offset digitally to the ADC value that is read in "Register 11h. ADC Value." The other method of calibration is to use the `tstrim` which compensates the analog circuit. This is done by setting `entstrim = 1` and using the `tstrim[2:0]` bits to offset the temperature in "Register 12h. Temperature Sensor Control." With this method of calibration, a negative offset may be achieved. With both methods of calibration better than ±3 °C absolute accuracy may be achieved.

The different ranges for the temperature sensor and ADC8 are demonstrated in Figure 20. The value of the ADC8 may be translated to a temperature reading by  $ADC8Value \times ADC8\ LSB + \text{Lowest Temperature in Temp Range}$ . For instance for a `tsrange = 00`,  $Temp = ADC8Value \times 0.5 - 64$ .

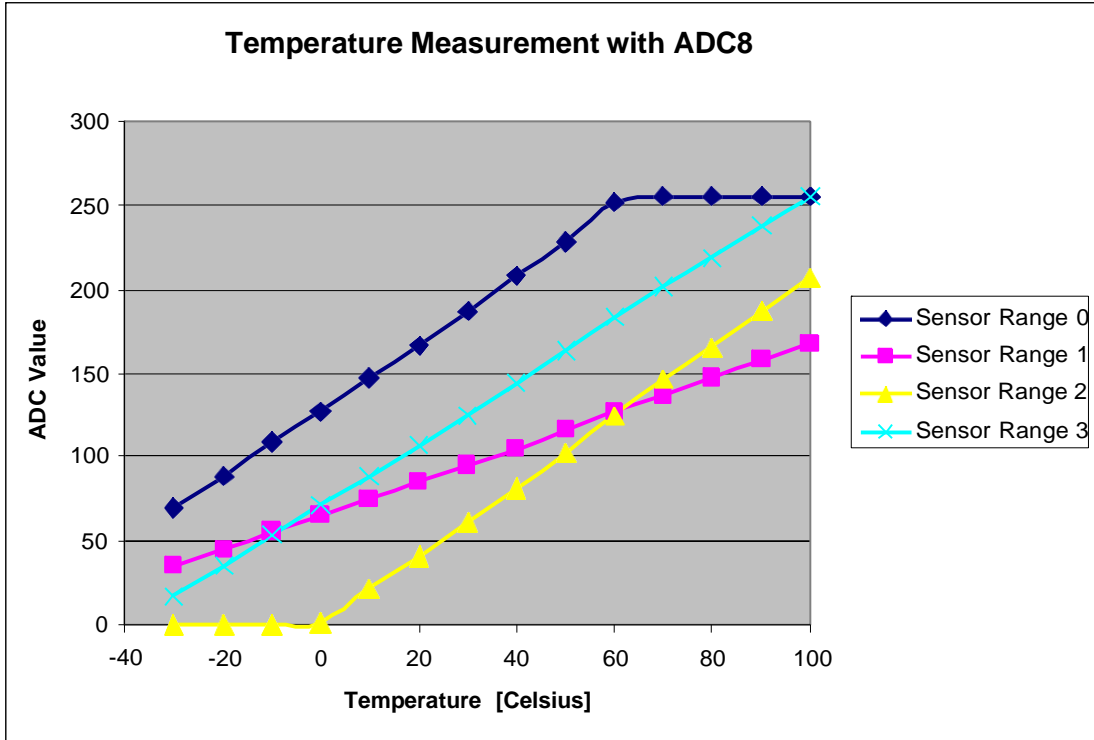


Figure 20. Temperature Ranges using ADC8



## 8.5. Low Battery Detector

A low battery detector (LBD) with digital read-out is integrated into the chip. A digital threshold may be programmed into the lbd[4:0] field in "Register 1Ah. Low Battery Detector Threshold." When the digitized battery voltage reaches this threshold an interrupt will be generated on the nIRQ pin to the microcontroller. The microcontroller can confirm source of the interrupt by reading "Register 03h. Interrupt/Status 1" and "Register 04h. Interrupt/Status 2,".

If the LBD is enabled while the chip is in SLEEP mode, it will automatically enable the RC oscillator which will periodically turn on the LBD circuit to measure the battery voltage. The battery voltage may also be read out through "Register 1Bh. Battery Voltage Level" at any time when the LBD is enabled. The low battery detect function is enabled by setting enlbd=1 in "Register 07h. Operating Mode and Function Control 1".

| Ad | R/W | Function/Description           | D7 | D6 | D5 | D4      | D3      | D2      | D1      | D0      | POR Def. |
|----|-----|--------------------------------|----|----|----|---------|---------|---------|---------|---------|----------|
| 1A | R/W | Low Battery Detector Threshold |    |    |    | lbd[4]  | lbd[3]  | lbd[2]  | lbd[1]  | lbd[0]  | 14h      |
| 1B | R   | Battery Voltage Level          | 0  | 0  | 0  | vbat[4] | vbat[3] | vbat[2] | vbat[1] | vbat[0] | —        |

The LBD output is digitized by a 5-bit ADC. When the LBD function is enabled, enlbd = 1 in "Register 07h. Operating Mode and Function Control 1", the battery voltage may be read at anytime by reading "Register 1Bh. Battery Voltage Level." A battery voltage threshold may be programmed in "Register 1Ah. Low Battery Detector Threshold." When the battery voltage level drops below the battery voltage threshold an interrupt will be generated on the nIRQ pin to the microcontroller if the LBD interrupt is enabled in "Register 06h. Interrupt Enable 2,". The microcontroller will then need to verify the interrupt by reading the interrupt status register, addresses 03 and 04h. The LSB step size for the LBD ADC is 50 mV, with the ADC range demonstrated in the table below. If the LBD is enabled the LBD and ADC will automatically be enabled every 1 s for approximately 250 μs to measure the voltage which minimizes the current consumption in Sensor mode. Before an interrupt is activated four consecutive readings are required.

$$\text{BatteryVoltage} = 1.7 + 50\text{mV} \times \text{ADCValue}$$

| ADC Value | VDD Voltage [V] |
|-----------|-----------------|
| 0         | < 1.7           |
| 1         | 1.7–1.75        |
| 2         | 1.75–1.8        |
| ...       | ...             |
| 29        | 3.1–3.15        |
| 30        | 3.15–3.2        |
| 31        | > 3.2           |

## 8.6. Wake-Up Timer and 32 kHz Clock Source

The chip contains an integrated wake-up timer which can be used to periodically wake the chip from SLEEP mode. The wake-up timer runs from the internal 32.768 kHz RC Oscillator. The wake-up timer can be configured to run when in SLEEP mode. If  $enwt = 1$  in "Register 07h. Operating Mode and Function Control 1" when entering SLEEP mode, the wake-up timer will count for a time specified defined in Registers 14–16h, "Wake Up Timer Period". At the expiration of this period an interrupt will be generated on the nIRQ pin if this interrupt is enabled. The microcontroller will then need to verify the interrupt by reading the Registers 03h–04h, "Interrupt Status 1 & 2". The wake-up timer value may be read at any time by the  $wtv[15:0]$  read only registers 13h–14h.

The formula for calculating the Wake-Up Period is the following:

$$WUT = \frac{4 \times M \times 2^R}{32.768} ms$$

| WUT Register | Description        |
|--------------|--------------------|
| wtr[3:0]     | R Value in Formula |
| wtd[1:0]     | D Value in Formula |
| wtm[15:0]    | M Value in Formula |

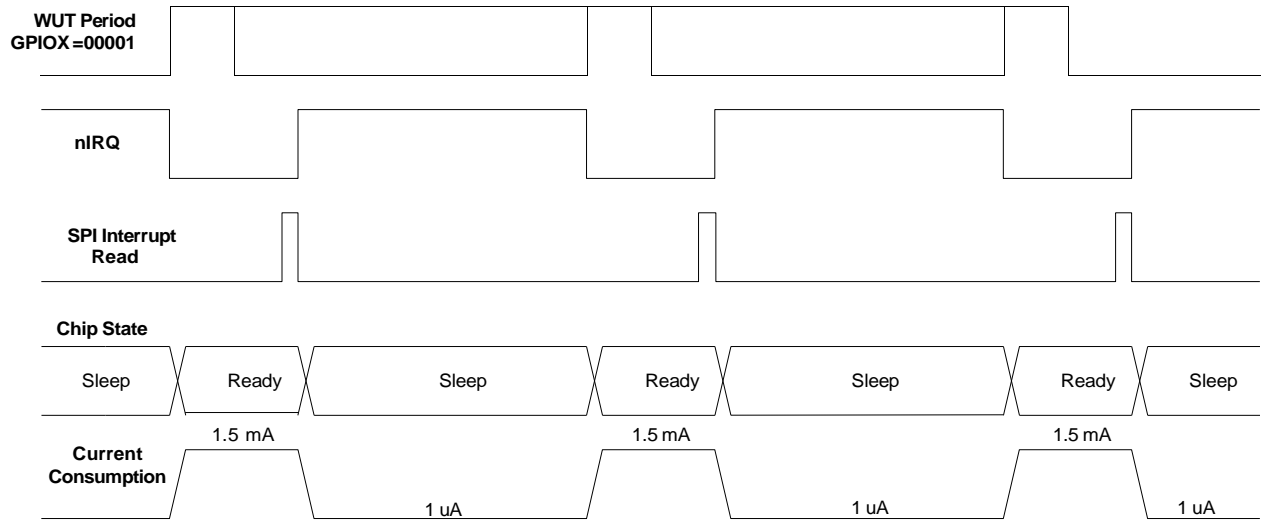
Use of the D variable in the formula is only necessary if finer resolution is required than can be achieved by using the R value.

| Add | R/W | Function/Description   | D7      | D6      | D5      | D4      | D3      | D2      | D1     | D0     | POR Def. |
|-----|-----|------------------------|---------|---------|---------|---------|---------|---------|--------|--------|----------|
| 14  | R/W | Wake-Up Timer Period 1 |         |         | wtr[3]  | wtr[2]  | wtr[1]  | wtr[0]  | wtd[1] | wtd[0] | 00h      |
| 15  | R/W | Wake-Up Timer Period 2 | wtm[15] | wtm[14] | wtm[13] | wtm[12] | wtm[11] | wtm[10] | wtm[9] | wtm[8] | 00h      |
| 16  | R/W | Wake-Up Timer Period 3 | wtm[7]  | wtm[6]  | wtm[5]  | wtm[4]  | wtm[3]  | wtm[2]  | wtm[1] | wtm[0] | 00h      |
| 17  | R   | Wake-Up Timer Value 1  | wtv[15] | wtv[14] | wtv[13] | wtv[12] | wtv[11] | wtv[10] | wtv[9] | wtv[8] | —        |
| 18  | R   | Wake-Up Timer Value 2  | wtv[7]  | wtv[6]  | wtv[5]  | wtv[4]  | wtv[3]  | wtv[2]  | wtv[1] | wtv[0] | —        |

There are two different methods for utilizing the wake-up timer (WUT) depending on if the WUT interrupt is enabled in "Register 06h. InterruptEnable 2,". If the WUT interrupt is enabled then nIRQ pin will go low when the timer expires. The chip will also change state so that the 30 MHz XTAL is enabled so that the microcontroller clock output is available for the microcontroller to use to process the interrupt. The other method of use is to not enable the WUT interrupt and use the WUT GPIO setting. In this mode of operation the chip will not change state until commanded by the microcontroller. The different modes of operating the WUT and the current consumption impacts are demonstrated in Figure 21.

A 32 kHz XTAL may also be used for better timing accuracy. By setting the  $x32\ ksel$  bit in "Register 07h. Operating & Function Control 1," GPIO0 is automatically reconfigured so that an external 32 kHz XTAL may be connected to this pin. In this mode, the GPIO0 is extremely sensitive to parasitic capacitance, so only the XTAL should be connected to this pin with the XTAL physically located as close to the pin as possible. Once the  $x32\ ksel$  bit is set, all internal functions such as WUT, micro-controller clock, and LDC mode will use the 32 kHz XTAL and not the 32 kHz RC oscillator.

Interrupt Enable  $enwut = 1$  (Reg 06h)



Interrupt Enable  $enwut = 0$  (Reg 06h)

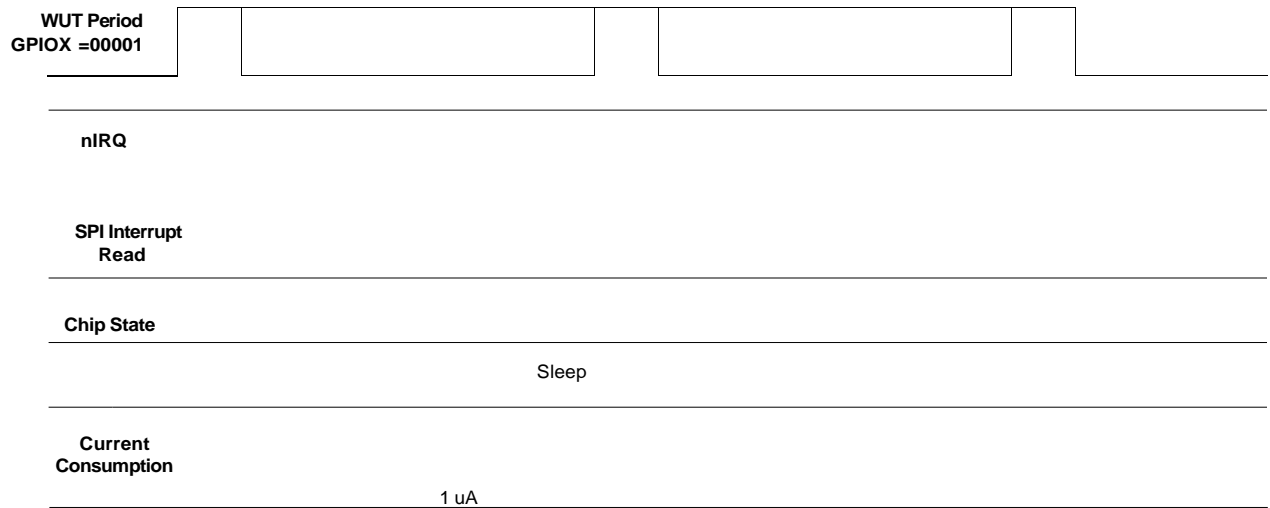
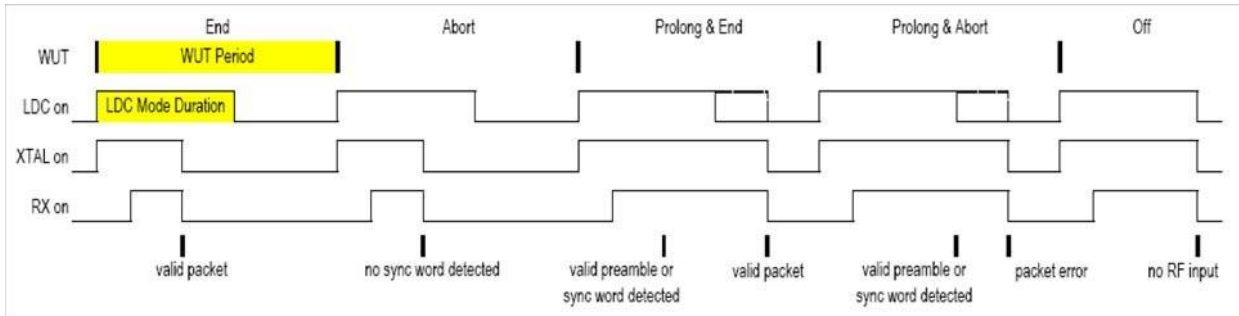


Figure 21. WUT Interrupt and WUT Operation

## 8.7. Low Duty Cycle Mode

The Low Duty Cycle Mode is available to automatically wake-up the receiver to check if a valid signal is available. The basic operation of the low duty cycle mode is demonstrated in the figure below. If a valid preamble or sync word is not detected the chip will return to sleep mode until the beginning of a new WUT period. If a valid preamble and sync are detected the receiver on period will be extended for the low duty cycle mode duration (TLDC) to receive all of the packet. The WUT period must be set in conjunction with the low duty cycle mode duration. The R value (Reg 14h) is shared between the WUT and the TLDC. The ldc[7:0] bits are located in “Register 19h. Low Duty Cycle Mode Duration.” The time of the TLDC is determined by the formula below:

$$TLDC = ldc [7 : 0] \times \frac{4 \times 2^R}{32.768} ms$$



**Figure 22. Low Duty Cycle Mode**

## 8.8. GPIO Configuration

Three general purpose IOs (GPIOs) are available. Numerous functions such as specific interrupts, Antenna Diversity Switch control, Microcontroller Output, etc. can be routed to the GPIO pins as shown in the tables below. When in Shutdown mode all the GPIO pads are pulled low.

**Note:** The ADC should not be selected as an input to the GPIO in Standby or Sleep Modes and will cause excess current consumption.

| Add | R/W | Function/Description   | D7          | D6          | D5         | D4         | D3       | D2       | D1       | D0       | POR Def. |
|-----|-----|------------------------|-------------|-------------|------------|------------|----------|----------|----------|----------|----------|
| 0B  | R/W | GPIO0 Configuration    | gpio0drv[1] | gpio0drv[0] | pup0       | gpio0[4]   | gpio0[3] | gpio0[2] | gpio0[1] | gpio0[0] | 00h      |
| 0C  | R/W | GPIO1 Configuration    | gpio1drv[1] | gpio1drv[0] | pup1       | gpio1[4]   | gpio1[3] | gpio1[2] | gpio1[1] | gpio1[0] | 00h      |
| 0D  | R/W | GPIO2 Configuration    | gpio2drv[1] | gpio2drv[0] | pup2       | gpio2[4]   | gpio2[3] | gpio2[2] | gpio2[1] | gpio2[0] | 00h      |
| 0E  | R/W | I/O Port Configuration |             | extitst[2]  | extitst[1] | extitst[0] | itsdo    | dio2     | dio1     | dio0     | 00h      |

The GPIO settings for GPIO1 and GPIO2 are the same as for GPIO0 with the exception of the 00000 default setting. The default settings for each GPIO are listed below:

| GPIO  | 00000—Default Setting |
|-------|-----------------------|
| GPIO0 | POR                   |
| GPIO1 | POR Inverted          |
| GPIO2 | Microcontroller Clock |

This application uses antenna diversity so a GPIO is used to control the antenna switch. For a complete list of the available GPIO's see "RFM31B Register Descriptions."

The GPIO drive strength may be adjusted with the gpioXdrv[1:0] bits. Setting a higher value will increase the drive strength and current capability of the GPIO by changing the driver size. Special care should be taken in setting the drive strength and loading on GPIO2 when the microcontroller clock is used. Excess loading or inadequate drive may contribute to increased spurious emissions.

## 8.9. Antenna Diversity

To mitigate the problem of frequency-selective fading due to multi-path propagation, some radio systems use a scheme known as antenna diversity. In this scheme, two antennas are used. Each time the radio enters RX mode the receive signal strength from each antenna is evaluated. This evaluation process takes place during the preamble portion of the packet. The antenna with the strongest received signal is then used for the remainder of that RX packet.

This chip fully supports antenna diversity with an integrated antenna diversity control algorithm. The required signals needed to control an external SPDT RF switch (such as PIN diode or GaAs switch) are available on the GPIOx pins. The operation of these GPIO signals is programmable to allow for different antenna diversity architectures and configurations. The antdiv[2:0] bits are found in register 08h “Operating & Function Control 2.” The GPIO pins are capable of sourcing up to 5 mA of current, so it may be used directly to forward-bias a PIN diode if desired.

The antenna diversity algorithm will automatically toggle back and forth between the antennas until the packet starts to arrive. The recommended preamble length for optimal antenna selection is 8 bytes. A special antenna diversity algorithm (antdiv[2:0] = 110 or 111) is included that allows for shorter preamble lengths for beacon mode in TDMA-like systems where the arrival of the packet is synchronous to the receiver enable. The recommended preamble length to obtain optimal antenna selection for synchronous mode is 4 bytes.

| Add | R/W | Function/Description           | D7        | D6        | D5        | D4    | D3       | D2    | D1     | D0       | POR Def. |
|-----|-----|--------------------------------|-----------|-----------|-----------|-------|----------|-------|--------|----------|----------|
| 08  | R/W | Operating & Function Control 2 | antdiv[2] | antdiv[1] | antdiv[0] | rxmpk | Reserved | enldm | ffclrx | Reserved | 00h      |

**Table 16. Antenna Diversity Control**

| antdiv[2:0] | RX State                                   |           | Non RX State |           |
|-------------|--|-----------|--------------|-----------|
|             | GPIO Ant1                                  | GPIO Ant2 | GPIO Ant1    | GPIO Ant2 |
| 000         | 0  | 1         | 0            | 0         |
| 001         | 1  | 0         | 0            | 0         |
| 010         | 0  | 1         | 1            | 1         |
| 011         | 1  | 0         | 1            | 1         |
| 100         | Antenna Diversity Algorithm                |           | 0            | 0         |
| 101         | Antenna Diversity Algorithm                |           | 1            | 1         |
| 110         | Antenna Diversity Algorithm in Beacon Mode |           | 0            | 0         |
| 111         | Antenna Diversity Algorithm in Beacon Mode |           | 1            | 1         |

### 8.10. RSSI and Clear Channel Assessment

Received signal strength indicator (RSSI) is an estimate of the signal strength in the channel to which the receiver is tuned. The RSSI value can be read from an 8-bit register with 0.5 dB resolution per bit. Figure 23 demonstrates the relationship between input power level and RSSI value. The absolute value of the RSSI will change slightly depending on the modem settings. The RSSI may be read at anytime, but an incorrect error may rarely occur. The RSSI value may be incorrect if read during the update period. The update period is approximately 10 ns every 4 Tb. For 10 kbps, this would result in a 1 in 40,000 probability that the RSSI may be read incorrectly. This probability is extremely low, but to avoid this, one of the following options is recommended: majority polling, reading the RSSI value within 1 Tb of the RSSI interrupt, or using the RSSI threshold described in the next paragraph for Clear Channel Assessment (CCA).

| Add | R/W | Function/Description                       | D7        | D6        | D5        | D4        | D3        | D2        | D1        | D0        | POR Def. |
|-----|-----|--|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|----------|
| 26  | R   | Received Signal Strength Indicator         | rssif[7]  | rssif[6]  | rssif[5]  | rssif[4]  | rssif[3]  | rssif[2]  | rssif[1]  | rssif[0]  | —        |
| 27  | R/W | RSSI Threshold for Clear Channel Indicator | rssith[7] | rssith[6] | rssith[5] | rssith[4] | rssith[3] | rssith[2] | rssith[1] | rssith[0] | 00h      |

For CCA, threshold is programmed into rssith[7:0] in "Register 27h. RSSI Threshold for Clear Channel Indicator." After the RSSI is evaluated in the preamble, a decision is made if the signal strength on this channel is above or below the threshold. If the signal strength is above the programmed threshold then the RSSI status bit, irssi, in "Register 04h. Interrupt/Status 2" will be set to 1. The RSSI status can also be routed to a GPIO line by configuring the GPIO configuration register to GPIOx[3:0] = 1110.



Figure 23. RSSI Value vs. Input Power

# RFM31B

## 9. Reference Design

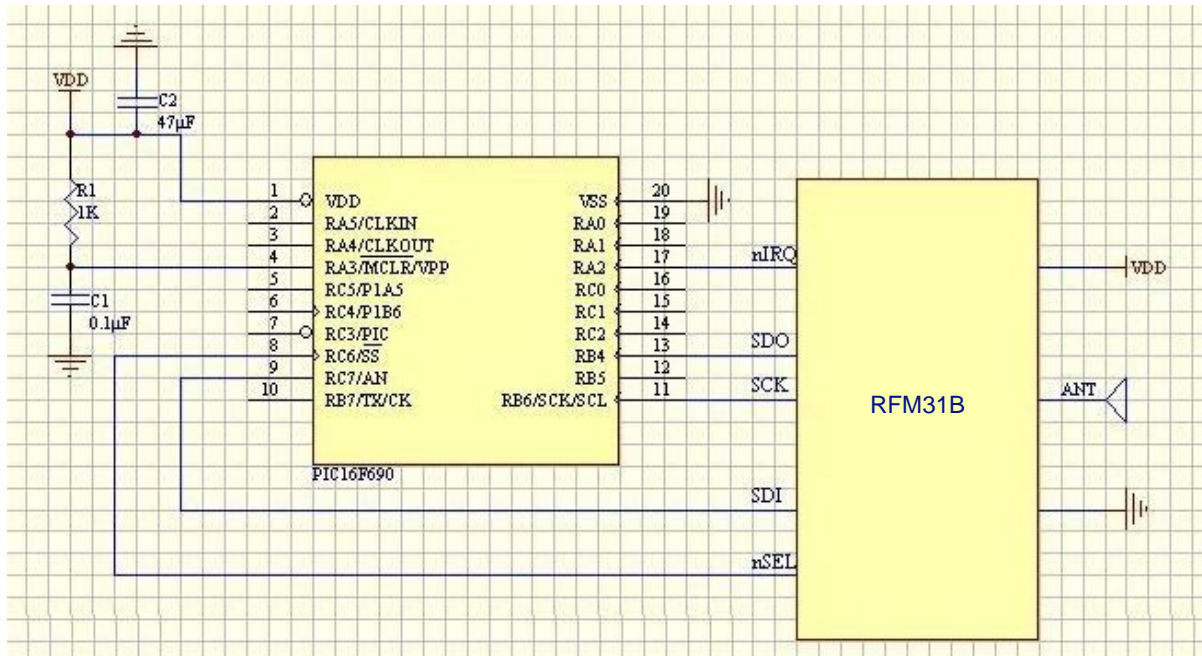


Figure 24.RFM31B Reference Design Schematic



## 10. Register Table and Descriptions

Table 17. Register Descriptions

| Add | R/W | Function/Desc                              | Data                      |                |             |             |             |            |            |            | POR Default |
|-----|-----|--|---------------------------|----------------|-------------|-------------|-------------|------------|------------|------------|-------------|
|     |     |  | D7                        | D6             | D5          | D4          | D3          | D2         | D1         | D0         |             |
| 01  | R   | Device Version                             | 0                         | 0              | 0           | vc[4]       | vc[3]       | vc[2]      | vc[1]      | vc[0]      | 06h         |
| 02  | R   | Device Status                              | ffovfl                    | ffunfl         | rxffem      | headerr     | reserved    | reserved   | cps[1]     | cps[0]     | —           |
| 03  | R   | Interrupt Status 1                         | ifferr                    | Reserved       | Reserved    | irxffaull   | iext        | Reserved   | ipkvalid   | icrcerror  | —           |
| 04  | R   | Interrupt Status 2                         | iswdet                    | ipreaval       | ipreainval  | irssi       | iwut        | ilbd       | ichiprdy   | ipor       | —           |
| 05  | R/W | Interrupt Enable 1                         | enfferr                   | Reserved       | Reserved    | enrxffaull  | enext       | Reserved   | enpkvalid  | encrcerror | 00h         |
| 06  | R/W | Interrupt Enable 2                         | enswdet                   | enpreaval      | enpreainval | enrssi      | enwut       | enlbd      | enchiprdy  | enpor      | 03h         |
| 07  | R/W | Operating & Function Control 1             | swres                     | enlbd          | enwt        | x32ksel     | Reserved    | rxon       | pllon      | xton       | 01h         |
| 08  | R/W | Operating & Function Control 2             | antdiv[2]                 | antdiv[1]      | antdiv[0]   | rxmpk       | Reserved    | enldm      | ffclrx     | Reserved   | 00h         |
| 09  | R/W | Crystal Oscillator Load Capacitance        | xtalshft                  | xlcl[6]        | xlcl[5]     | xlcl[4]     | xlcl[3]     | xlcl[2]    | xlcl[1]    | xlcl[0]    | 7Fh         |
| 0A  | R/W | Microcontroller Output Clock               | Reserved                  | Reserved       | clkt[1]     | clkt[0]     | enlfc       | mclk[2]    | mclk[1]    | mclk[0]    | 06h         |
| 0B  | R/W | GPIO0 Configuration                        | gpio0drv[1]               | gpio0drv[0]    | pup0        | gpio0[4]    | gpio0[3]    | gpio0[2]   | gpio0[1]   | gpio0[0]   | 00h         |
| 0C  | R/W | GPIO1 Configuration                        | gpio1drv[1]               | gpio1drv[0]    | pup1        | gpio1[4]    | gpio1[3]    | gpio1[2]   | gpio1[1]   | gpio1[0]   | 00h         |
| 0D  | R/W | GPIO2 Configuration                        | gpio2drv[1]               | gpio2drv[0]    | pup2        | gpio2[4]    | gpio2[3]    | gpio2[2]   | gpio2[1]   | gpio2[0]   | 00h         |
| 0E  | R/W | I/O Port Configuration                     | Reserved                  | extitst[2]     | extitst[1]  | extitst[0]  | itsdo       | dio2       | dio1       | dio0       | 00h         |
| 0F  | R/W | ADC Configuration                          | adcstart/ <i>adc-done</i> | adcsl[2]       | adcsl[1]    | adcsl[0]    | adcref[1]   | adcref[0]  | adcgain[1] | adcgain[0] | 00h         |
| 10  | R/W | ADC Sensor Amplifier Offset                | Reserved                  | Reserved       | Reserved    | Reserved    | adcoffs[3]  | adcoffs[2] | adcoffs[1] | adcoffs[0] | 00h         |
| 11  | R   | ADC Value                                  | adc[7]                    | adc[6]         | adc[5]      | adc[4]      | adc[3]      | adc[2]     | adc[1]     | adc[0]     | —           |
| 12  | R/W | Temperature Sensor Control                 | tsrange[1]                | tsrange[0]     | entsoffs    | entstrim    | tstrim[3]   | tstrim[2]  | tstrim[1]  | tstrim[0]  | 20h         |
| 13  | R/W | Temperature Value Offset                   | tvoffs[7]                 | tvoffs[6]      | tvoffs[5]   | tvoffs[4]   | tvoffs[3]   | tvoffs[2]  | tvoffs[1]  | tvoffs[0]  | 00h         |
| 14  | R/W | Wake-Up Timer Period 1                     | Reserved                  | Reserved       | Reserved    | wtr[4]      | wtr[3]      | wtr[2]     | wtr[1]     | wtr[0]     | 03h         |
| 15  | R/W | Wake-Up Timer Period 2                     | wtm[15]                   | wtm[14]        | wtm[13]     | wtm[12]     | wtm[11]     | wtm[10]    | wtm[9]     | wtm[8]     | 00h         |
| 16  | R/W | Wake-Up Timer Period 3                     | wtm[7]                    | wtm[6]         | wtm[5]      | wtm[4]      | wtm[3]      | wtm[2]     | wtm[1]     | wtm[0]     | 01h         |
| 17  | R   | Wake-Up Timer Value 1                      | wtv[15]                   | wtv[14]        | wtv[13]     | wtv[12]     | wtv[11]     | wtv[10]    | wtv[9]     | wtv[8]     | —           |
| 18  | R   | Wake-Up Timer Value 2                      | wtv[7]                    | wtv[6]         | wtv[5]      | wtv[4]      | wtv[3]      | wtv[2]     | wtv[1]     | wtv[0]     | —           |
| 19  | R/W | Low-Duty Cycle Mode Duration               | ldc[7]                    | ldc[6]         | ldc[5]      | ldc[4]      | ldc[3]      | ldc[2]     | ldc[1]     | ldc[0]     | 00h         |
| 1A  | R/W | Low Battery Detector Threshold             | Reserved                  | Reserved       | Reserved    | lbd[4]      | lbd[3]      | lbd[2]     | lbd[1]     | lbd[0]     | 14h         |
| 1B  | R   | Battery Voltage Level                      | 0                         | 0              | 0           | vbat[4]     | vbat[3]     | vbat[2]    | vbat[1]    | vbat[0]    | —           |
| 1C  | R/W | IF Filter Bandwidth                        | dwn3_bypass               | ndec[2]        | ndec[1]     | ndec[0]     | filset[3]   | filset[2]  | filset[1]  | filset[0]  | 01h         |
| 1D  | R/W | AFC Loop Gearshift Override                | afcbd                     | enafc          | afcgearh[2] | afcgearh[1] | afcgearh[0] | 1p5 bypass | matap      | ph0size    | 40h         |
| 1E  | R/W | AFC Timing Control                         | swait_timer[1]            | swait_timer[0] | shwait[2]   | shwait[1]   | shwait[0]   | anwait[2]  | anwait[1]  | anwait[0]  | 0Ah         |
| 1F  | R/W | Clock Recovery Gearshift Override          | Reserved                  | Reserved       | crfast[2]   | crfast[1]   | crfast[0]   | crslow[2]  | crslow[1]  | crslow[0]  | 03h         |
| 20  | R/W | Clock Recovery Oversampling Ratio          | rxosr[7]                  | rxosr[6]       | rxosr[5]    | rxosr[4]    | rxosr[3]    | rxosr[2]   | rxosr[1]   | rxosr[0]   | 64h         |
| 21  | R/W | Clock Recovery Offset 2                    | rxosr[10]                 | rxosr[9]       | rxosr[8]    | stallctrl   | ncoff[19]   | ncoff[18]  | ncoff[17]  | ncoff[16]  | 01h         |
| 22  | R/W | Clock Recovery Offset 1                    | ncoff[15]                 | ncoff[14]      | ncoff[13]   | ncoff[12]   | ncoff[11]   | ncoff[10]  | ncoff[9]   | ncoff[8]   | 47h         |
| 23  | R/W | Clock Recovery Offset 0                    | ncoff[7]                  | ncoff[6]       | ncoff[5]    | ncoff[4]    | ncoff[3]    | ncoff[2]   | ncoff[1]   | ncoff[0]   | AEh         |
| 24  | R/W | Clock Recovery Timing Loop Gain 1          | Reserved                  | Reserved       | Reserved    | rxnocomp    | crgain2x    | crgain[10] | crgain[9]  | crgain[8]  | 02h         |
| 25  | R/W | Clock Recovery Timing Loop Gain 0          | crgain[7]                 | crgain[6]      | crgain[5]   | crgain[4]   | crgain[3]   | crgain[2]  | crgain[1]  | crgain[0]  | 8Fh         |
| 26  | R   | Received Signal Strength Indicator         | rssl[7]                   | rssl[6]        | rssl[5]     | rssl[4]     | rssl[3]     | rssl[2]    | rssl[1]    | rssl[0]    | —           |
| 27  | R/W | RSSI Threshold for Clear Channel Indicator | rssith[7]                 | rssith[6]      | rssith[5]   | rssith[4]   | rssith[3]   | rssith[2]  | rssith[1]  | rssith[0]  | 1Eh         |

Table 17. Register Descriptions (Continued)

| Add   | R/W | Function/Desc                      | Data          |               |               |               |             |             |             |             | POR Default |
|-------|-----|------------------------------------|---------------|---------------|---------------|---------------|-------------|-------------|-------------|-------------|-------------|
|       |     |                                    | D7            | D6            | D5            | D4            | D3          | D2          | D1          | D0          |             |
| 28    | R   | Antenna Diversity Register 1       | adrssi1[7]    | adrssia[6]    | adrssia[5]    | adrssia[4]    | adrssia[3]  | adrssia[2]  | adrssia[1]  | adrssia[0]  | —           |
| 29    | R   | Antenna Diversity Register 2       | adrssib[7]    | adrssib[6]    | adrssib[5]    | adrssib[4]    | adrssib[3]  | adrssib[2]  | adrssib[1]  | adrssib[0]  | —           |
| 2A    | R/W | AFC Limiter                        | Afclim[7]     | Afclim[6]     | Afclim[5]     | Afclim[4]     | Afclim[3]   | Afclim[2]   | Afclim[1]   | Afclim[0]   | 00h         |
| 2B    | R   | AFC Correction Read                | afc_corr[9]   | afc_corr[8]   | afc_corr[7]   | afc_corr[6]   | afc_corr[5] | afc_corr[4] | afc_corr[3] | afc_corr[2] | 00h         |
| 2C    | R/W | OOK Counter Value 1                | afc_corr[9]   | afc_corr[9]   | ookfrzen      | peakdeten     | madeten     | ookcnt[10]  | ookcnt[9]   | ookcnt[8]   | 18h         |
| 2D    | R/W | OOK Counter Value 2                | ookcnt[7]     | ookcnt[6]     | ookcnt[5]     | ookcnt[4]     | ookcnt[3]   | ookcnt[2]   | ookcnt[1]   | ookcnt[0]   | BCh         |
| 2E    | R/W | Slicer Peak Hold                   | Reserved      | attack[2]     | attack[1]     | attack[0]     | decay[3]    | decay[2]    | decay[1]    | decay[0]    | 26h         |
| 2F    |     | Reserved                           |               |               |               |               |             |             |             |             |             |
| 30    | R/W | Data Access Control                | enpacrx       | lsbrst        | crdonly       | skip2ph       | Reserved    | encrc       | crc[1]      | crc[0]      | 8Dh         |
| 31    | R   | EzMAC status                       | 0             | rxrcr1        | pkscrch       | pkrx          | pkvalid     | crccrerror  | Reserved    | Reserved    | —           |
| 32    | R/W | Header Control 1                   | bcen[3:0]     |               |               |               | hdch[3:0]   |             |             |             | 0Ch         |
| 33    | R/W | Header Control 2                   | skipsyn       | hdlen[2]      | hdlen[1]      | hdlen[0]      | fixpklen    | syncden[1]  | syncden[0]  | prealen[8]  | 22h         |
| 34    | R/W | Preamble Length                    | prealen[7]    | prealen[6]    | prealen[5]    | prealen[4]    | prealen[3]  | prealen[2]  | prealen[1]  | prealen[0]  | 08h         |
| 35    | R/W | Preamble Detection Control         | preath[4]     | preath[3]     | preath[2]     | preath[1]     | preath[0]   | rssi_off[2] | rssi_off[1] | rssi_off[0] | 2Ah         |
| 36    | R/W | Sync Word 3                        | sync[31]      | sync[30]      | sync[29]      | sync[28]      | sync[27]    | sync[26]    | sync[25]    | sync[24]    | 2Dh         |
| 37    | R/W | Sync Word 2                        | sync[23]      | sync[22]      | sync[21]      | sync[20]      | sync[19]    | sync[18]    | sync[17]    | sync[16]    | D4h         |
| 38    | R/W | Sync Word 1                        | sync[15]      | sync[14]      | sync[13]      | sync[12]      | sync[11]    | sync[10]    | sync[9]     | sync[8]     | 00h         |
| 39    | R/W | Sync Word 0                        | sync[7]       | sync[6]       | sync[5]       | sync[4]       | sync[3]     | sync[2]     | sync[1]     | sync[0]     | 00h         |
| 3A-3E |     | Reserved                           |               |               |               |               |             |             |             |             |             |
| 3F    | R/W | Check Header 3                     | chhd[31]      | chhd[30]      | chhd[29]      | chhd[28]      | chhd[27]    | chhd[26]    | chhd[25]    | chhd[24]    | 00h         |
| 40    | R/W | Check Header 2                     | chhd[23]      | chhd[22]      | chhd[21]      | chhd[20]      | chhd[19]    | chhd[18]    | chhd[17]    | chhd[16]    | 00h         |
| 41    | R/W | Check Header 1                     | chhd[15]      | chhd[14]      | chhd[13]      | chhd[12]      | chhd[11]    | chhd[10]    | chhd[9]     | chhd[8]     | 00h         |
| 42    | R/W | Check Header 0                     | chhd[7]       | chhd[6]       | chhd[5]       | chhd[4]       | chhd[3]     | chhd[2]     | chhd[1]     | chhd[0]     | 00h         |
| 43    | R/W | Header Enable 3                    | hden[31]      | hden[30]      | hden[29]      | hden[28]      | hden[27]    | hden[26]    | hden[25]    | hden[24]    | FFh         |
| 44    | R/W | Header Enable 2                    | hden[23]      | hden[22]      | hden[21]      | hden[20]      | hden[19]    | hden[18]    | hden[17]    | hden[16]    | FFh         |
| 45    | R/W | Header Enable 1                    | hden[15]      | hden[14]      | hden[13]      | hden[12]      | hden[11]    | hden[10]    | hden[9]     | hden[8]     | FFh         |
| 46    | R/W | Header Enable 0                    | hden[7]       | hden[6]       | hden[5]       | hden[4]       | hden[3]     | hden[2]     | hden[1]     | hden[0]     | FFh         |
| 47    | R   | Received Header 3                  | rxhd[31]      | rxhd[30]      | rxhd[29]      | rxhd[28]      | rxhd[27]    | rxhd[26]    | rxhd[25]    | rxhd[24]    | —           |
| 48    | R   | Received Header 2                  | rxhd[23]      | rxhd[22]      | rxhd[21]      | rxhd[20]      | rxhd[19]    | rxhd[18]    | rxhd[17]    | rxhd[16]    | —           |
| 49    | R   | Received Header 1                  | rxhd[15]      | rxhd[14]      | rxhd[13]      | rxhd[12]      | rxhd[11]    | rxhd[10]    | rxhd[9]     | rxhd[8]     | —           |
| 4A    | R   | Received Header 0                  | rxhd[7]       | rxhd[6]       | rxhd[5]       | rxhd[4]       | rxhd[3]     | rxhd[2]     | rxhd[1]     | rxhd[0]     | —           |
| 4B    | R   | Received Packet Length             | rxplen[7]     | rxplen[6]     | rxplen[5]     | rxplen[4]     | rxplen[3]   | rxplen[2]   | rxplen[1]   | rxplen[0]   | —           |
| 4C-4E |     | Reserved                           |               |               |               |               |             |             |             |             |             |
| 4F    | R/W | ADC8 Control                       | Reserved      | Reserved      | adc8[5]       | adc8[4]       | adc8[3]     | adc8[2]     | adc8[1]     | adc8[0]     | 10h         |
| 50-5F |     | Reserved                           |               |               |               |               |             |             |             |             |             |
| 60    | R/W | Channel Filter Coefficient Address | Inv_pre_th[3] | Inv_pre_th[2] | Inv_pre_th[1] | Inv_pre_th[0] | chfiladd[3] | chfiladd[2] | chfiladd[1] | chfiladd[0] | 00h         |
| 61    |     | Reserved                           |               |               |               |               |             |             |             |             |             |
| 62    | R/W | Crystal Oscillator/Control Test    | pwst[2]       | pwst[1]       | pwst[0]       | clkhyst       | enbias2x    | enamp2x     | bufovr      | enbuf       | 24h         |
| 63-68 |     | Reserved                           |               |               |               |               |             |             |             |             |             |
| 69    | R/W | AGC Override 1                     | Reserved      | sgi           | agcen         | lnagain       | pga3        | pga2        | pga1        | pga0        | 20h         |
| 6A-6C |     | Reserved                           |               |               |               |               |             |             |             |             |             |
| 70    | R/W | Modulation Mode Control 1          | Reserved      | Reserved      |               | enphpwdn      | manppol     | enmaninv    | enmanch     | enwhite     | 0Ch         |
| 71    | R/W | Modulation Mode Control 2          | trclk[1]      | trclk[0]      | dtmod[1]      | dtmod[0]      | eninv       | fd[8]       | modtyp[1]   | modtyp[0]   | 00h         |
| 73    | R/W | Frequency Offset 1                 | fo[7]         | fo[6]         | fo[5]         | fo[4]         | fo[3]       | fo[2]       | fo[1]       | fo[0]       | 00h         |
| 74    | R/W | Frequency Offset 2                 | Reserved      | Reserved      | Reserved      | Reserved      | Reserved    | Reserved    | fo[9]       | fo[8]       | 00h         |

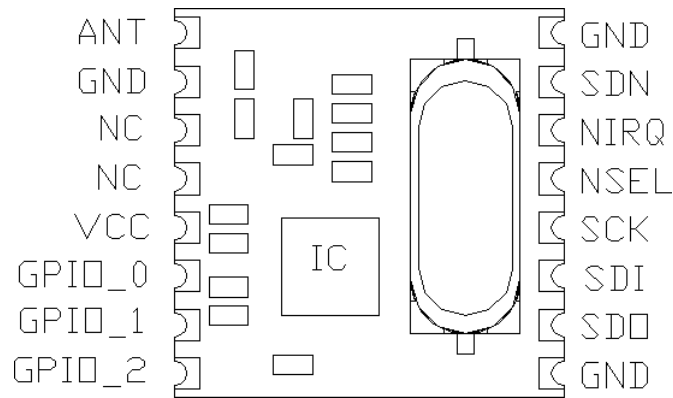
**Table 17. Register Descriptions (Continued)**

| Add | R/W      | Function/Desc                    | Data     |          |            |            |            |            |            |            | POR Default |
|-----|----------|----------------------------------|----------|----------|------------|------------|------------|------------|------------|------------|-------------|
|     |          |                                  | D7       | D6       | D5         | D4         | D3         | D2         | D1         | D0         |             |
| 75  | R/W      | Frequency Band Select            | Reserved | sbsel    | hbssel     | fb[4]      | fb[3]      | fb[2]      | fb[1]      | fb[0]      | 75h         |
| 76  | R/W      | Nominal Carrier Frequency 1      | fc[15]   | fc[14]   | fc[13]     | fc[12]     | fc[11]     | fc[10]     | fc[9]      | fc[8]      | BBh         |
| 77  | R/W      | Nominal Carrier Frequency 0      | fc[7]    | fc[6]    | fc[5]      | fc[4]      | fc[3]      | fc[2]      | fc[1]      | fc[0]      | 80h         |
| 78  | Reserved |                                  |          |          |            |            |            |            |            |            |             |
| 79  | R/W      | Frequency Hopping Channel Select | fhch[7]  | fhch[6]  | fhch[5]    | fhch[4]    | fhch[3]    | fhch[2]    | fhch[1]    | fhch[0]    | 00h         |
| 7A  | R/W      | Frequency Hopping Step Size      | fhs[7]   | fhs[6]   | fhs[5]     | fhs[4]     | fhs[3]     | fhs[2]     | fhs[1]     | fhs[0]     | 00h         |
| 7B  | Reserved |                                  |          |          |            |            |            |            |            |            |             |
| 7E  | R/W      | RX FIFO Control                  | Reserved | Reserved | rxafthr[5] | rxafthr[4] | rxafthr[3] | rxafthr[2] | rxafthr[1] | rxafthr[0] | 37h         |
| 7F  | R/W      | FIFO Access                      | fifod[7] | fifod[6] | fifod[5]   | fifod[4]   | fifod[3]   | fifod[2]   | fifod[1]   | fifod[0]   | —           |

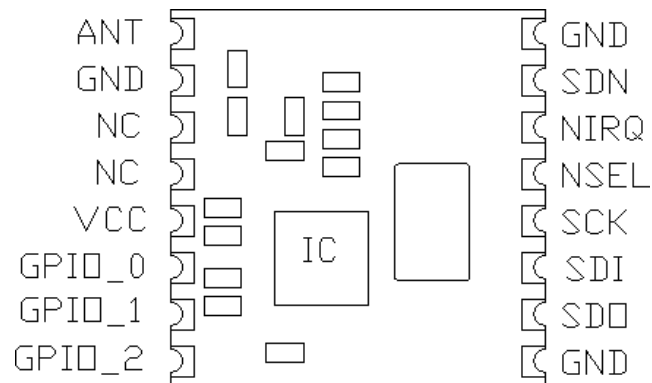
**Note:** Detailed register descriptions are available in “RFM31B Register Descriptions.”

**11. Pin Descriptions: RFM31B**

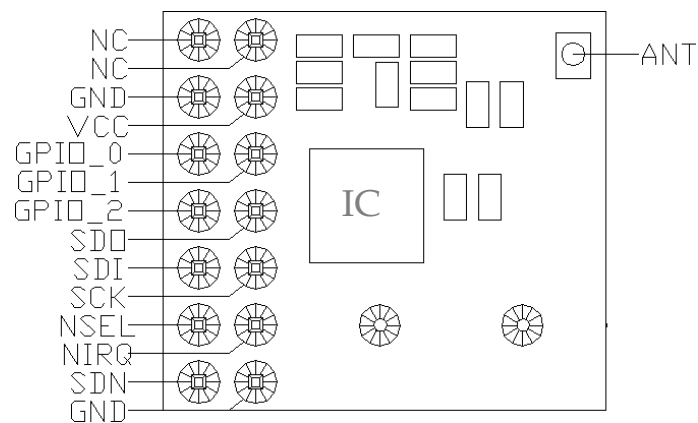
**RFM31B-S1**



**RFM31B-S2**



**RFM31B-D**



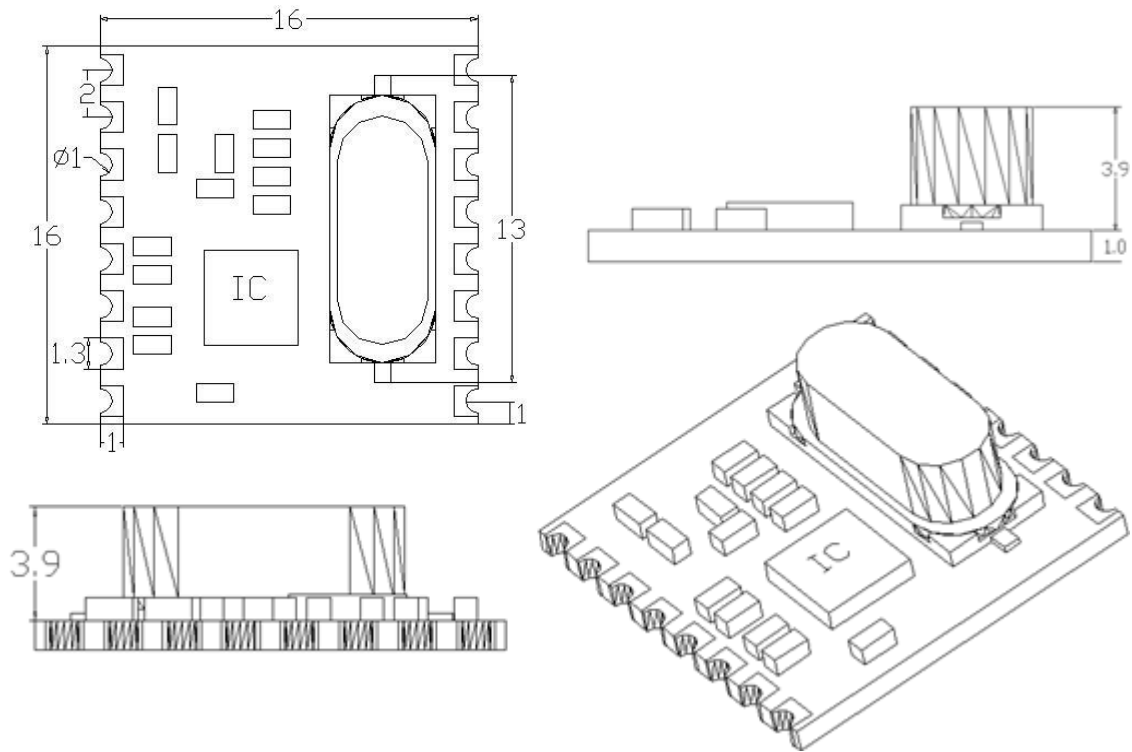
# RFM31 B

---

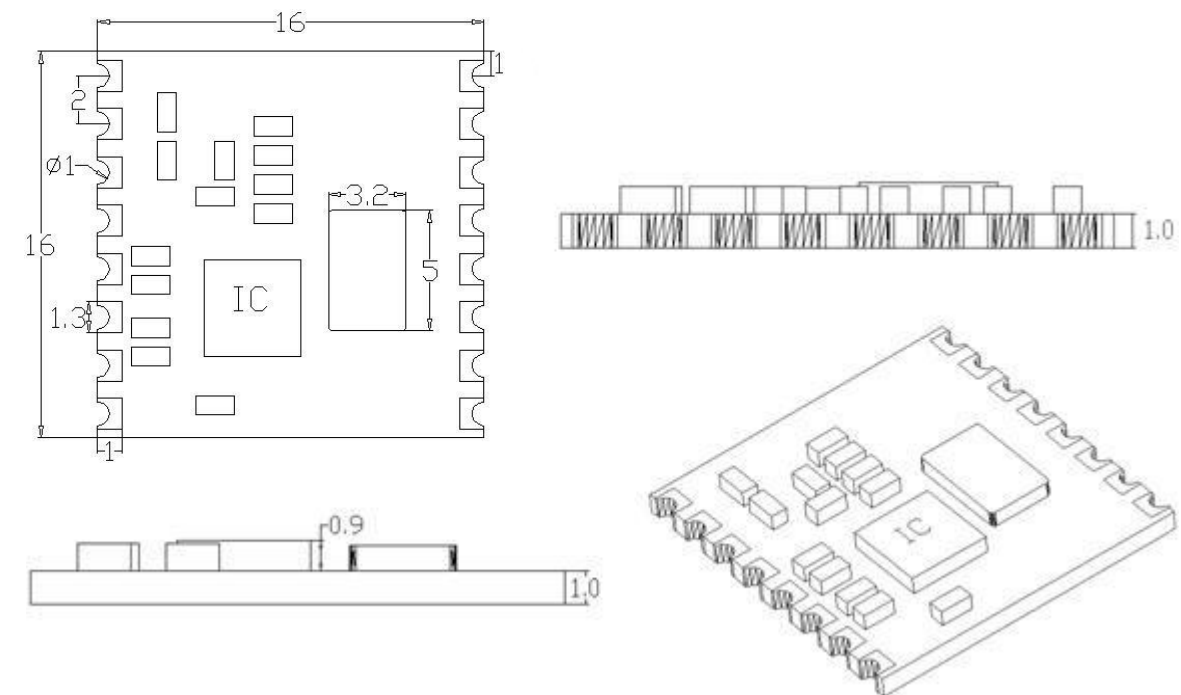
|        |     |  |
|--------|-----|--|
| VCC    | S   | +1.8 to +3.6 V supply voltage. The recommended Vcc supply voltage is +3.3 V.   |
| GND    | S   | Ground reference.  |
| GPIO_0 | I/O | General Purpose Digital I/O that may be configured through the registers to perform various functions including: Microcontroller Clock Output, FIFO status, POR, Wake-Up timer, Low Battery Detect, TRSW, AntDiversity control, etc. See the SPI GPIO Configuration Registers, Address 0Bh, 0Ch, and 0Dh for more information.   |
| GPIO_1 | I/O |  |
| GPIO_2 | I/O |  |
| SDO    | O   | 0–Vcc V digital output that provides a serial readback function of the internal control registers.   |
| SDI    | I   | Serial Data input. 0–Vcc V digital input. This pin provides the serial data stream for the 4-line serial data bus.   |
| SCLK   | I   | Serial Clock input. 0–VDD V digital input. This pin provides the serial data clock function for the 4-line serial data bus. Data is clocked into the RFM31 on positive edge transitions.   |
| nSEL   | I   | Serial Interface Select input. 0– Vcc V digital input. This pin provides the Select/Enable function for the 4-line serial data bus. The signal is also used to signify burst read/write mode.  |
| nIRQ   | O   | General Microcontroller Interrupt Status output. When the RFM31 exhibits anyone of the Interrupt Events the nIRQ pin will be set low=0. Please see the Control Logic registers section for more information on the Interrupt Events. The Microcontroller can then determine the state of the interrupt by reading a corresponding SPI Interrupt Status Registers, Address 03h and 04h. |
| SDN    | I   | Shutdown input pin. 0–Vcc V digital input. SDN should be = 0 in all modes except Shutdown mode. When SDN =1 the chip will be completely shutdown and the contents of the registers will be lost.   |
| ANT    | I/O | RF signal input.(50 OHM input Impedance)   |
| NC     |     | No Connection  |

12. Mechanical Dimension: RFM31B

SMD PACKAGE (S1)

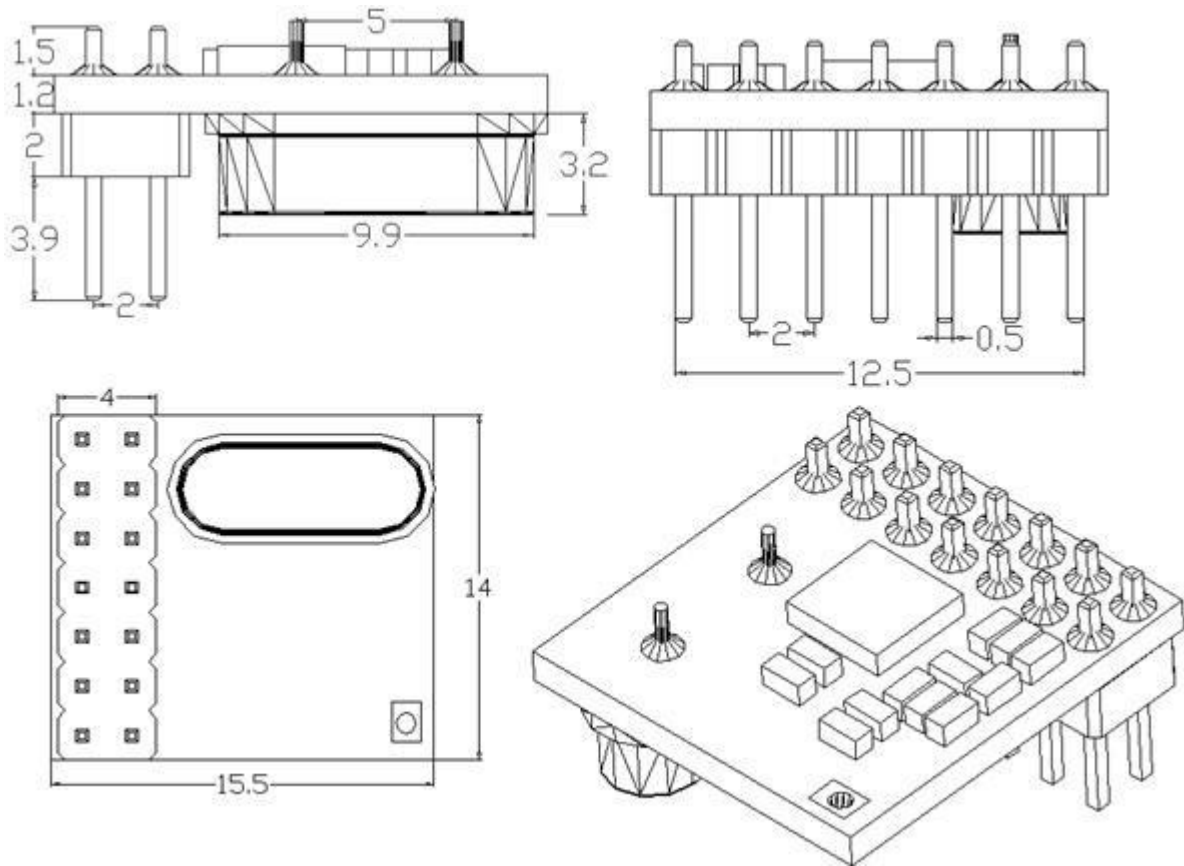


SMD PACKAGE (S2)



# RFM31B

## DIP PACKAGE (D)



## 13. Ordering Information

Part Number=module type—operation band—package type

**RFM31B—433—D**

↑                    ↑                    ↓  
module type    operation band    Package

example: 1, RFM31B module at 433MHz band, DIP : RFM31B-433-D。

2, RFM31B module at 868MHZ band, SMD, thickness at 4.9mm: RFM31B-868-S1。



