MRF24XA
Low-Power, 2.4 GHz ISM-Band IEEE 802.15.4™ RF Transceiver with Extended Proprietary Features

Features
- IEEE 802.15.4™-2003 and IEEE 802.15.4-2006 Standard Compliant RF transceiver
- Multiple air-data-rates:
  - 250 kbps (IEEE 802.15.4)
  - 125, 500, 1000, 2000 kbps, co-existence with standard networks
- Configurable TX output power: -19 to 1 dBm
- Frame header duration scales with the selected data rate
- On-the-fly, per frame air-data-rate detection (link-by-link independent air-data-rates)
- Inferred destination addressing (to further save on framing overheads; optional)

Full-Featured MCU Support
- Hardware frame parser
- Hardware CSMA-CA controller, automatic Acknowledgment (ACK) and Frame Check Sequence (FCS)
- Supports all Clear Channel Assessment (CCA) modes
- Reports ED, RSSI, LQI, and CFO
- Channel Agility with acknowledgments
- Two independent 128 byte frame buffers
- Streaming mode to maximize throughput
- Automatic packet retransmit capability
- Hardware Security Engine (AES-128) and configurable Encryption/Decryption mode

Low-Power
- Extreme minimization of radio ON time
  - Highest channel-admissible data rate used
  - 20%-70% overall reduction through framing
- 2 Mbps frames can reduce radio ON time by a factor of 4 to 8 with respect to 250 kbps frames
- 27.5 mA TX current (typical at 0 dBm)
- 13.5 mA RX current in RX Listen Power-Saving mode
- 15.5-16.5 mA RX current in RX Packet Demodulation mode (data rate and device Configuration dependent)
- Deep Sleep, Sleep, Crystal ON, and RX Listen Power-Saving modes
- Memory retention in Deep Sleep mode (<40 nA typical)
- Automated functions minimize MCU ON time

General
- Low external component count
- Best-in-class battery life preservation
- Supply range: 1.5V to 3.6V
- Compact 32-pin 5x5 mm² QFN package
- Temperature range: -40°C to +85°C
- Certified turnkey-ready solutions available

Applications
- IEEE 802.15.4/ZigBee® systems (RF4CE and MIWI™ network)
- Industrial monitoring and control
- IEEE 1588 precise timing protocol networks
- Automatic meter reading
- Home building automation
- Low-power wireless sensor networks
- Consumer electronics, voice and audio
# Table of Contents

1.0 Device Overview ............................................................................................................................................................................. 5  
2.0 Hardware Description.................................................................................................................................................................................. 7  
3.0 Functional Description.................................................................................................................................................................................. 71  
4.0 General Transceiver Operations ......................................................................................................................................................... 89  
5.0 IEEE 802.15.4™ Compliant Frame Format and Frame Processing ........................................................................................................ 133  
6.0 Proprietary Frame Format and Frame Processing........................................................................................................................................ 169  
7.0 Advanced Link Behavior in Proprietary Packet Mode ........................................................................................................................................... 189  
8.0 Bridging ........................................................................................................................................................................................................... 197  
9.0 Physical layer Functions .................................................................................................................................................................................... 199  
10.0 Battery Life Optimization .......................................................................................................................................................................... 237  
11.0 Electrical Characteristics .................................................................................................................................................................................. 241  
12.0 Packaging Information .................................................................................................................................................................................... 249  
Appendix A: Revision History .............................................................................................................................................................................. 249  
The Microchip Web Site ................................................................................................................................................................................... 251  
Customer Change Notification Service .............................................................................................................................................................. 251  
Customer Support ............................................................................................................................................................................................. 251  
INDEX ............................................................................................................................................................................................................. 255

---

**TO OUR VALUED CUSTOMERS**

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at docerrors@microchip.com or fax the Reader Response Form in the back of this data sheet to (480) 792-4150. We welcome your feedback.

**Most Current Data Sheet**

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000A is version A of document DS30000).

**Errata**

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip’s Worldwide Web site; http://www.microchip.com
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include literature number) you are using.

**Customer Notification System**

Register on our web site at www.microchip.com to receive the most current information on all of our products.
1.0 DEVICE OVERVIEW

MRF24XA is an IEEE 802.15.4™ Standard compliant 2.4 GHz RF transceiver with feature extensions. MRF24XA integrates the PHY and MAC functionality in a single-chip solution. MRF24XA implements a low-cost, low-power, high data rate (125 kbps to 2 Mbps) Wireless Personal Area Network (WPAN) device. All the data rates contains the same spectral shape requiring identical bandwidth. At 125 kbps data rate Direct Sequence Spread Spectrum (DSSS) is combined with error correction and coding for maximum range and robustness against interference. The 2 Mbps data rate is used to minimize radio ON time, therefore extending battery life. Figure 1-1 illustrates a simplified block diagram of an MRF24XA wireless node. MRF24XA interfaces to many popular Microchip PIC® microcontrollers through a 4-wire serial SPI interface, interrupt, GPIO, and RESET pins. MRF24XA can also handle external Power Amplifier (PA) and Low Noise Amplifier (LNA).

MRF24XA provides hardware support for:
- Energy detection
- Carrier sense
- Four CCA modes
- CSMA-CA algorithm
- Automatic packet retransmission
- Automatic acknowledgement
- Independent transmit and receive buffers
- Security engine supports encryption and decryption for MAC sublayer and upper layer
- Inferred destination addressing
- Channel agility with ACKs
- Battery monitoring

These features reduce the processing load, allowing the use of low-cost 8-bit microcontrollers.

MRF24XA is compatible with Microchip’s ZigBee®, MiWi™ and MiWi P2P software stacks. Each software stack is available as a free download, including source code, from the Microchip web site:

FIGURE 1-1: MRF24XA WIRELESS NODE BLOCK DIAGRAM
2.0 HARDWARE DESCRIPTION

2.1 Overview

MRF24XA is an IEEE 802.15.4 Standard compliant 2.4 GHz RF transceiver with extended feature set for longer battery life, higher throughput and increased operating range.

MRF24XA integrates the PHY and MAC functionality in a single-chip solution. Figure 2-1 illustrates a block diagram of the MRF24XA circuitry.

An external 16 MHz crystal clocks the frequency synthesizer and generates a 2.4 GHz frequency RF carrier.

The receiver is a zero-IF architecture consisting of a Low-Noise Amplifier, down conversion mixers, channel filters and baseband amplifiers with a Received Signal Strength Indicator (RSSI).

The transmitter is a direct conversion architecture with a 1 dBm maximum output (typical) and 20 dB power control range.

The internal transmitter and receiver circuits contains separate RFP and RFN input/output pins that connects to impedance matching circuitry (balun) and antenna. An external Power Amplifier or Low Noise Amplifier, or both is controlled through the PA and LNA pins.

Three general purpose Input/Output (GPIO) pins are configurable for control or monitoring purposes.

The power management circuitry consists of an integrated Low Dropout (LDO) voltage regulator and a 5-bit resolution Battery Monitor Block. MRF24XA is placed into a low-current (<40 nA typical) Deep Sleep mode.

The Media Access Controller (MAC) circuitry can sequence the transmit, receive and automatically enable the security operations. The host MCU can control these mechanisms through register configurations and Frame Control (FCtrl) field embedded in the downloaded formatted frames. Three alternative frame formats are supported: IEEE 802.15.4 2003, 2006 compliant MAC frame formats and a flexible and power-efficient advanced MAC frame format, which is proprietary. Before launching transmission, the host must load the buffer with a formatted frame. The hardware can optionally perform encryption and message integrity code appending as configured, then sends the frame appending a Frame Check Sequence (FCS).

Hardware can autonomously sequence acknowledge reception and automatic retransmissions.

In reception, the format of the demodulated frame is verified. Depending on the Configuration, duplicate frames, frames with corrupted FCS or address mismatch are discarded. On reception of valid frames, automatic acknowledge sending, decryption and message integrity checking are supported.

As the default, separate buffers are reserved for transmission and reception. Alternatively, either the Transmit Streaming (TX-Streaming) or the Receive Streaming (RX-Streaming) modes are selected whereby buffers are used by alternating between the two for servicing a single direction of data flow. The AES-128 engine are governed to perform network-layer security processing and supports complete security suites such as CTR, CBC-MAC and CCM*.

Transceiver is controlled through a 4-wire SPI, interrupt and RESET pins.

2.2 Operating Modes

Table 2-1 summarizes the Operating modes of MRF24XA.

<table>
<thead>
<tr>
<th>TABLE 2-1: MRF24XA POWER MODES</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Operating Mode</th>
<th>1.2V LDO</th>
<th>Crystal Oscillator</th>
<th>Synthesizer</th>
<th>RX Front End</th>
<th>RX Baseband</th>
<th>TX Chain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deep Sleep</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>Sleep</td>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>RFOFF Crystal ON</td>
<td>ON</td>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>RFOFF Synthesizer ON</td>
<td>ON</td>
<td>ON</td>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>RX Listen Power-Save</td>
<td>ON</td>
<td>ON</td>
<td>ON</td>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>RX Listen</td>
<td>ON</td>
<td>ON</td>
<td>ON</td>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>TX</td>
<td>ON</td>
<td>ON</td>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
<td>ON</td>
</tr>
</tbody>
</table>
2.3 Block Diagram

FIGURE 2-1: MRF24XA ARCHITECTURE BLOCK DIAGRAM
### 2.4 Pin Descriptions

<table>
<thead>
<tr>
<th>Pin</th>
<th>Symbol</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AVDD</td>
<td>Power</td>
<td>1.2V supply, normally connected to VREGOUT (pin 29)</td>
</tr>
<tr>
<td>2</td>
<td>RFOUTP</td>
<td>AO</td>
<td>Differential RF Output (+)</td>
</tr>
<tr>
<td>3</td>
<td>RFOUTN</td>
<td>AO</td>
<td>Differential RF Output (–)</td>
</tr>
<tr>
<td>4</td>
<td>AVDD</td>
<td>Power</td>
<td>1.2V supply, normally connected to VREGOUT (pin 29)</td>
</tr>
<tr>
<td>5</td>
<td>RFINP</td>
<td>AI</td>
<td>Differential RF Input (+)</td>
</tr>
<tr>
<td>6</td>
<td>AVSS</td>
<td>Ground</td>
<td>Ground</td>
</tr>
<tr>
<td>7</td>
<td>RFINN</td>
<td>AI</td>
<td>Differential RF Input (–)</td>
</tr>
<tr>
<td>8</td>
<td>AVDD</td>
<td>Power</td>
<td>1.2V supply, normally connected to VREGOUT (pin 29)</td>
</tr>
<tr>
<td>9</td>
<td>AVSS</td>
<td>Power</td>
<td>Ground</td>
</tr>
<tr>
<td>10</td>
<td>GPIO2</td>
<td>DIO</td>
<td>GPIO2</td>
</tr>
<tr>
<td>11</td>
<td>GPIO1</td>
<td>DIO</td>
<td>GPIO1</td>
</tr>
<tr>
<td>12</td>
<td>GPIO0</td>
<td>DIO</td>
<td>GPIO0</td>
</tr>
<tr>
<td>13</td>
<td>INT</td>
<td>DO</td>
<td>Interrupt Output, active-low</td>
</tr>
<tr>
<td>14</td>
<td>CS</td>
<td>DI</td>
<td>SPI Chip Select Pin, active-low</td>
</tr>
<tr>
<td>15</td>
<td>SCK</td>
<td>DI</td>
<td>SPI serial clock</td>
</tr>
<tr>
<td>16</td>
<td>SDI</td>
<td>DI</td>
<td>SPI serial data Input</td>
</tr>
<tr>
<td>17</td>
<td>SDO</td>
<td>DO</td>
<td>SPI serial data Output</td>
</tr>
<tr>
<td>18</td>
<td>RESET</td>
<td>DI</td>
<td>Reset Input, active-low</td>
</tr>
<tr>
<td>19</td>
<td>DVSS</td>
<td>Ground</td>
<td>Digital ground</td>
</tr>
<tr>
<td>20</td>
<td>PA</td>
<td>DO</td>
<td>External PA enable Output</td>
</tr>
<tr>
<td>21</td>
<td>LNA</td>
<td>DO</td>
<td>External LNA enable Output</td>
</tr>
<tr>
<td>22</td>
<td>DVDD</td>
<td>Power</td>
<td>Digital 1.2V supply, normally connected to VREGOUT (pin 29)</td>
</tr>
<tr>
<td>23</td>
<td>DVDDIO</td>
<td>Power</td>
<td>Digital 1.5V–3.6V supply for the IO blocks, normally connected to VREGIN (pin 30)</td>
</tr>
<tr>
<td>24</td>
<td>OSC1</td>
<td>AI</td>
<td>Crystal oscillator Pin 1, External Clock Input</td>
</tr>
<tr>
<td>25</td>
<td>OSC2</td>
<td>AO</td>
<td>Crystal oscillator Pin 2</td>
</tr>
<tr>
<td>26</td>
<td>AVSS</td>
<td>Ground</td>
<td>Ground</td>
</tr>
<tr>
<td>27</td>
<td>AVDD</td>
<td>Power</td>
<td>1.2V supply, normally connected to VREGOUT (pin 29)</td>
</tr>
<tr>
<td>28</td>
<td>RBIAS</td>
<td>AO</td>
<td>External resistor reference pin</td>
</tr>
<tr>
<td>29</td>
<td>VREGOUT</td>
<td>Power</td>
<td>1.2V regulated Output</td>
</tr>
<tr>
<td>30</td>
<td>VREGIN</td>
<td>Power</td>
<td>1.5V–3.6V regulator Input</td>
</tr>
<tr>
<td>31</td>
<td>AVSS</td>
<td>Ground</td>
<td>Ground</td>
</tr>
<tr>
<td>32</td>
<td>AVDD</td>
<td>Power</td>
<td>1.2V supply, normally connected to VREGOUT (pin 29)</td>
</tr>
</tbody>
</table>

**Legend:** A = Analog, D = Digital, I = Input, O = Output

**Note 1:** In case PCB runs out of space, disconnect these pins.
2.4.1 POWER AND GROUND PINS

Table 2-3 lists the recommended bypass capacitors. Vdd pins 29 and 30 are power pins, which require different bypass capacitors to ensure sufficient bypass decoupling and stability. Bypass capacitors must have low serial resistance. The 4.7 µF capacitors must be made of ceramic or high-performance tantalum.

On PCB layout minimize trace length from the Vdd pin to the bypass capacitors and connect capacitors to the pads as short as possible. PCB tracks must be wide enough to minimize voltage drop and serial inductance of the power line. Analog and digital power lines must follow a star topology, where the common point is the bypass capacitor on pin 30.

### TABLE 2-3: RECOMMENDED BYPASS CAPACITOR VALUES

<table>
<thead>
<tr>
<th>Vdd Pin</th>
<th>Symbol</th>
<th>Bypass Capacitor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VDD</td>
<td>3.3 pF</td>
</tr>
<tr>
<td>29</td>
<td>VREGOUT</td>
<td>4.7 µF</td>
</tr>
<tr>
<td>30</td>
<td>VREGIN</td>
<td>10 nF + 4.7 µF</td>
</tr>
</tbody>
</table>

2.4.2 16 MHz MAIN OSCILLATOR PINS

The 16 MHz oscillator is connected to OSC1 and OSC2 pins as shown in Figure 2-2, which provides the reference frequency for the internal RF, MAC and BB circuitry. Table 2-4 lists the crystal parameters.

To minimize parasitic effects on pins, the crystal must be put as close as possible to MRF24XA. It keeps the tracks short. Crystal must be surrounded with ground pour to minimize cross coupling effects. Crystal load capacitors must be placed close to the crystal.

### TABLE 2-4: 16 MHz CRYSTAL PARAMETERS\(^1\)

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>16 MHz</td>
</tr>
<tr>
<td>Frequency tolerance for 500, 250 and 125 kbps data rates (including manufacturing aging and temperature)</td>
<td>±60 ppm(^1)</td>
</tr>
<tr>
<td>Frequency tolerance for 2 and 1 Mbps data rates (including manufacturing aging and temperature)</td>
<td>±40 ppm(^2)</td>
</tr>
<tr>
<td>Mode</td>
<td>Fundamental</td>
</tr>
<tr>
<td>Load Capacitance</td>
<td>18 pF</td>
</tr>
<tr>
<td>ESR</td>
<td>80 Ohm max</td>
</tr>
</tbody>
</table>

Note 1: IEEE 802.15.4 defines ±40 ppm.
Note 2: These values are only used for design guidance.
2.4.3 RESET (RESET) PIN

An external Hardware Reset is performed by asserting the RESET pin 18 low. If the RESET pin is deasserted, MRF24XA starts the internal Calibration process. RDYIF interrupt is set when the device is ready to use. The RESET pin contains an internal weak pull-up resistor.

2.4.4 INTERRUPT (INT) PIN

The Interrupt (INT) pin 13 provides an interrupt signal to the host MCU from MRF24XA where the signal is active-low polarity. Interrupt sources must be enabled and unmasked before the INT pin becomes active.

Refer to Section 3.2 “Interrupts” for the functional description of interrupts.

2.4.5 GENERAL PURPOSE INPUT/OUTPUT (GPIO) PINS

Three GPIO pins are configured individually for control or monitoring purposes. The TRISGPIOx bits in the GPIO register (0x0D) configures the input or output selection.

GPIO data is read or written through the GPIO bits of GPIO register. The GPIO interrupt polarity is selected through GPIOxP bits in the STGPIO (0x0E) register.

GPIO lines in Input mode are used in Schmitt Trigger Input mode. STENGPIOx bits of STGPIO register enables Schmitt Triggers. GPIOs are also used to monitor the internal blocks. The GPIOMODE bits <3:0> of the PINCON (0x0C) register selects these monitoring functions.

2.4.6 SERIAL PERIPHERAL INTERFACE (SPI) PORT PINS

MRF24XA communicates with a host MCU through a 4-wire SPI port as a slave device. MRF24XA supports SPI mode 0,0, which requires that SCK idles in a low state. The CS pin must be held low while communicating with MRF24XA. Figure 2-3 illustrates timing for a read and a write operation. MRF24XA receives the data through the SDI pin and clocks in on the rising edge of SCK. MRF24XA sends data through the SDO pin and clocks out on the falling edge of SCK. The SDO lines preserve its HiZ state in Deep Sleep mode.
2.5 Application Example

Figure 2-3 illustrates the schematic of a recommended application circuit for MRF24XA.

FIGURE 2-3: MRF24XA APPLICATION CIRCUIT
2.6 Memory Organization

Table 2-5 shows that memory is functionally divided into Special Function Registers (SFR) and data buffers. The SFRs provide control, status and device Configuration addressing for MRF24XA operations. Data buffers serve as temporary buffers for data transmission and reception. Memory is accessed through two addressing methods: Short (1 byte) and Long (2 bytes).

### 2.6.1 ADDRESS OVERVIEW

MRF24XA contains two Addressing modes:
- **Short Address Mode**: Requires one byte for address, and may be used to access the first 64 on-chip control registers.
- **Long Address Mode**: Requires two bytes for address, and may be used to access all on-chip registers and data buffers. Figure 2-4 illustrates these modes.

#### TABLE 2-5: MRF24XA MEMORY MAP

<table>
<thead>
<tr>
<th>Short Addressing</th>
<th>Long Addressing</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00–0x0F</td>
<td>0x00–0x1FF</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>0x10</td>
<td>0x200–0x2FF</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>0x2F</td>
<td>0x284–0x2FF</td>
</tr>
<tr>
<td>0x30</td>
<td>0x285–0x2FF</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>0x3F</td>
<td>0x300–0x3FF</td>
</tr>
<tr>
<td>TX and EXTDEV</td>
<td>0x384–0x3FF</td>
</tr>
<tr>
<td>MAC</td>
<td>0x385–0x3FF</td>
</tr>
<tr>
<td>SYSTEM LEVEL</td>
<td>0x3FF–0x3FF</td>
</tr>
<tr>
<td>PIR4 (0x07)</td>
<td>Retained in Deep Sleep</td>
</tr>
<tr>
<td>PIE1 (0x08)</td>
<td></td>
</tr>
<tr>
<td>0x39–0x3A</td>
<td>0x1FF–0x1FF</td>
</tr>
<tr>
<td>0x3A–0x3F</td>
<td>...</td>
</tr>
<tr>
<td>0x40–0x4F</td>
<td>0x200–0x2FF</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>0x50–0x5F</td>
<td>0x284–0x2FF</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>0x60–0x6F</td>
<td>0x285–0x2FF</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>0x70</td>
<td>0x300–0x3FF</td>
</tr>
<tr>
<td>MAC</td>
<td>0x384–0x3FF</td>
</tr>
<tr>
<td>PHY</td>
<td>0x385–0x3FF</td>
</tr>
<tr>
<td>RETAINED IN DEEP SLEEP</td>
<td></td>
</tr>
<tr>
<td>TX and EXTDEV</td>
<td></td>
</tr>
<tr>
<td>MAC</td>
<td></td>
</tr>
<tr>
<td>PIR4 (0x07)</td>
<td></td>
</tr>
<tr>
<td>PIE1 (0x08)</td>
<td></td>
</tr>
</tbody>
</table>

| 0x00–0x0F                 | 0x0F–0x0F                |
| ...                       | ...                      |
| 0x10                      | 0x10–0xFF                |
| ...                       | ...                      |
| 0x2F                      | 0x200–0x2FF              |
| 0x30                      | 0x284–0x2FF              |
| ...                       | ...                      |
| 0x3F                      | 0x300–0x3FF              |
| TX and EXTDEV             | 0x384–0x3FF              |
| MAC                       | 0x385–0x3FF              |
| SYSTEM LEVEL              | 0x3FF–0x3FF              |
| PIR4 (0x07)               | Retained in Deep Sleep   |
| PIE1 (0x08)               |                          |
| 0x39–0x3A                 | 0x1FF–0x1FF              |
| 0x3A–0x3F                 | ...                      |
| 0x40–0x4F                 | 0x200–0x2FF              |
| ...                       | ...                      |
| 0x50–0x5F                 | 0x284–0x2FF              |
| ...                       | ...                      |
| 0x60–0x6F                 | 0x285–0x2FF              |
| ...                       | ...                      |
| 0x70                       | 0x300–0x3FF              |
| MAC                       | 0x384–0x3FF              |
| PHY                       | 0x385–0x3FF              |
| RETAINED IN DEEP SLEEP    |                          |
| TX and EXTDEV             |                          |
| MAC                       |                          |
| PIR4 (0x07)               |                          |
| PIE1 (0x08)               |                          |
## TABLE 2-6: SHORT ADDRESS REGISTER SUMMARY FOR MRF24XA

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Address</th>
<th>Name</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTEM LEVEL</td>
<td>0x00</td>
<td>REGRST</td>
<td>r</td>
<td>r</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>REGRST&lt;5:0&gt;</td>
<td>71</td>
</tr>
<tr>
<td></td>
<td>0x01</td>
<td>FSMRST</td>
<td>r</td>
<td>r</td>
<td>r</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>FSRST&lt;4:0&gt;</td>
<td>71</td>
</tr>
<tr>
<td></td>
<td>0x02</td>
<td>OPSTATUS</td>
<td>r</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MACOP&lt;3:0&gt;</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>0x03</td>
<td>STATUS</td>
<td>r</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>RFOF&lt;2:0&gt;</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>0x04</td>
<td>PIRQ</td>
<td>r</td>
<td>RDYIF</td>
<td>r</td>
<td>IDIEF</td>
<td>r</td>
<td>CALIOF</td>
<td>CALAIF</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0x05</td>
<td>PIR2</td>
<td>TXIF</td>
<td>TXENCIF</td>
<td>TXMAIF</td>
<td>TXACKIF</td>
<td>TXCSMIF</td>
<td>TXSZIF</td>
<td>TOVFIF</td>
<td>FIFMF</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>0x06</td>
<td>PIR3</td>
<td>RXIF</td>
<td>RXDECIF</td>
<td>RXTAGIF</td>
<td>RXIDENTIF</td>
<td>RXFLTIF</td>
<td>RXOFIF</td>
<td>STRMF</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0x07</td>
<td>PIR4</td>
<td>TXSFDF</td>
<td>RXXSFDFIF</td>
<td>ERRORIF</td>
<td>WARNIF</td>
<td>EDCAIF</td>
<td>GPIOIF</td>
<td>GPIOIF</td>
<td>GPIOIF</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>0x08</td>
<td>PIR1</td>
<td>r</td>
<td>RDYIF</td>
<td>r</td>
<td>IDIEF</td>
<td>r</td>
<td>CALIOE</td>
<td>CALAEF</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0x09</td>
<td>PIR2</td>
<td>TXIE</td>
<td>TXENCIE</td>
<td>TXMAIE</td>
<td>TXACKIE</td>
<td>TXCSMIE</td>
<td>TXZIE</td>
<td>TOVFIE</td>
<td>FRMEF</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>0x0A</td>
<td>PIR3</td>
<td>RXIE</td>
<td>RXDECIE</td>
<td>RXTAGIE</td>
<td>RXIDENTIE</td>
<td>RXFLTIE</td>
<td>RXOFIE</td>
<td>STRMEF</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0x0B</td>
<td>PIR4</td>
<td>TXSFDE</td>
<td>TXSFDFIE</td>
<td>ERRORIE</td>
<td>WARNIE</td>
<td>EDCAIE</td>
<td>GPIOIE</td>
<td>GPIOIE</td>
<td>GPIOIE</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>0x0C</td>
<td>PINCON</td>
<td>GIE</td>
<td>r</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>GIMODE&lt;3:0&gt;</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>0x0D</td>
<td>GPIO</td>
<td>GPIOEN</td>
<td>TRISGPL2</td>
<td>TRISGPL1</td>
<td>TRISGPL0</td>
<td>GPIO2</td>
<td>GPIO1</td>
<td>GPIO0</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0x0E</td>
<td>STGPIO</td>
<td>r</td>
<td>GPIO2P</td>
<td>GPIO1P</td>
<td>GPIO0P</td>
<td>STENPI2</td>
<td>STENPI1</td>
<td>STENPI0</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0x0F</td>
<td>PULLGPIO</td>
<td>r</td>
<td>PULLDRGPIO2</td>
<td>PULLDRGPIO1</td>
<td>PULLDRGPIO0</td>
<td>PULLDRGPIO2</td>
<td>PULLDRGPIO1</td>
<td>PULLDRGPIO0</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>MAC</td>
<td>0x10</td>
<td>MACCON1</td>
<td>TRXMODE&lt;1:0&gt;</td>
<td>ADDRZ&lt;2:0&gt;</td>
<td>CRCSZ</td>
<td>FRMFMT</td>
<td>SECFLAGOVR</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0x11</td>
<td>MACON2</td>
<td>CHANNEL&lt;3:0&gt;</td>
<td>SECSUITE&lt;3:0&gt;</td>
<td>34</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0x12</td>
<td>TXCON</td>
<td>TXST</td>
<td>DTSM</td>
<td>TXENC</td>
<td>TXBUFEMPTY</td>
<td>CSMAEN</td>
<td>DR&lt;2:0&gt;</td>
<td>35</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0x13</td>
<td>RXACKWAIT</td>
<td>RXACKWAIT&lt;7:0&gt;</td>
<td>36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0x14</td>
<td>RETXCOUNT</td>
<td>RETXMCNT&lt;3:0&gt;</td>
<td>37</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0x15</td>
<td>RXCON1</td>
<td>RXEN</td>
<td>NOPA</td>
<td>RXDEC</td>
<td>RXVLQIEN</td>
<td>RSVPRIEN</td>
<td>RSVCHDREN</td>
<td>RSVCFRIM</td>
<td>38</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0x16</td>
<td>RXCON2</td>
<td>RXBUFFUL</td>
<td>IDENTREJ</td>
<td>ACKRXXF</td>
<td>ACKTXF</td>
<td>AUTOPTEN</td>
<td>AUTOACKEN</td>
<td>ADPTCHEN</td>
<td>ADPTDREN</td>
<td>39</td>
</tr>
<tr>
<td></td>
<td>0x17</td>
<td>TXACKTO</td>
<td>TXACKTO&lt;7:0&gt;</td>
<td>41</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0x18</td>
<td>RXFILTER</td>
<td>PANCRDN</td>
<td>CRRCREJ</td>
<td>CMDREJ</td>
<td>DATAREJ</td>
<td>UNIREJ</td>
<td>NOTMEREJ</td>
<td>BCRFEN</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0x19</td>
<td>TMRCON</td>
<td>BOMCNT&lt;2:0&gt;</td>
<td>BASET&lt;4:0&gt;</td>
<td>43</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0x1A</td>
<td>CSMABE</td>
<td>MAXBE&lt;3:0&gt;</td>
<td>MINBE&lt;3:0&gt;</td>
<td>45</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0x1B</td>
<td>BOUNIT</td>
<td>BOUNT&lt;7:0&gt;</td>
<td>46</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0x1C</td>
<td>STRMTOL</td>
<td>STRMTO&lt;7:0&gt;</td>
<td>46</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0x1D</td>
<td>STRMTOH</td>
<td>STRMTO&lt;15:8&gt;</td>
<td>46</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0x1E</td>
<td>OFFTM</td>
<td>OFFTM&lt;7:0&gt;</td>
<td>47</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend:  
- r = Reserved, read as '0'.

© 2015 Microchip Technology Inc.
<table>
<thead>
<tr>
<th>Architecture</th>
<th>Address</th>
<th>Name</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0x1F</td>
<td>ADDR1</td>
<td>ADDR&lt;7:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>0x20</td>
<td>ADDR2</td>
<td>ADDR&lt;15:8&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>0x21</td>
<td>ADDR3</td>
<td>ADDR&lt;23:16&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>0x22</td>
<td>ADDR4</td>
<td>ADDR&lt;31:24&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>0x23</td>
<td>ADDR5</td>
<td>ADDR&lt;39:32&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>0x24</td>
<td>ADDR6</td>
<td>ADDR&lt;47:40&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>0x25</td>
<td>ADDR7</td>
<td>ADDR&lt;55:48&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>0x26</td>
<td>ADDR8</td>
<td>ADDR&lt;63:56&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>0x27</td>
<td>ADDR9</td>
<td>ADDR&lt;7:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>0x28</td>
<td>ADDR10</td>
<td>ADDR&lt;15:8&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>0x29</td>
<td>ADDR11</td>
<td>ADDR&lt;23:16&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>0x2A</td>
<td>ADDR12</td>
<td>ADDR&lt;31:24&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>0x2B</td>
<td>ADDR13</td>
<td>ADDR&lt;39:32&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>0x2C</td>
<td>ADDR14</td>
<td>ADDR&lt;47:40&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>0x2D</td>
<td>ADDR15</td>
<td>ADDR&lt;55:48&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>0x2E</td>
<td>ADDR16</td>
<td>ADDR&lt;63:56&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>0x2F</td>
<td>ADDR17</td>
<td>ADDR&lt;7:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>0x30</td>
<td>ADDR18</td>
<td>ADDR&lt;15:8&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>0x31</td>
<td>ADDR19</td>
<td>ADDR&lt;23:16&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>0x32</td>
<td>ADDR20</td>
<td>ADDR&lt;31:24&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>0x33</td>
<td>ADDR21</td>
<td>ADDR&lt;39:32&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>0x34</td>
<td>ADDR22</td>
<td>ADDR&lt;47:40&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>0x35</td>
<td>ADDR23</td>
<td>ADDR&lt;55:48&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>0x36</td>
<td>ADDR24</td>
<td>ADDR&lt;63:56&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>0x37</td>
<td>ADDR25</td>
<td>ADDR&lt;7:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>0x38</td>
<td>ADDR26</td>
<td>ADDR&lt;15:8&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>0x39</td>
<td>ADDR27</td>
<td>ADDR&lt;23:16&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>0x3A</td>
<td>ADDR28</td>
<td>ADDR&lt;31:24&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>0x3B</td>
<td>ADDR29</td>
<td>ADDR&lt;39:32&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>0x3C</td>
<td>ADDR30</td>
<td>ADDR&lt;47:40&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>0x3D</td>
<td>ADDR31</td>
<td>ADDR&lt;55:48&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>0x3E</td>
<td>ADDR32</td>
<td>ADDR&lt;63:56&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>0x3F</td>
<td>ADDR33</td>
<td>ADDR&lt;7:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>49</td>
</tr>
</tbody>
</table>

**Legend:**  
- **r** = Reserved, read as '0'.

**TABLE 2-6: SHORT ADDRESS REGISTER SUMMARY FOR MRF24XA (CONTINUED)**

**Architecture** | **Address** | **Name**      | **Bit 7** | **Bit 6** | **Bit 5** | **Bit 4** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** | **Page** |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PHY</td>
<td>0x2F</td>
<td>CCACON1</td>
<td>CCABUSY</td>
<td>CCAST</td>
<td>RSSITHR&lt;5:0&gt;</td>
<td>CCALEN&lt;1:0&gt;</td>
<td>CCAMODE&lt;1:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0x30</td>
<td>CCACON2</td>
<td></td>
<td></td>
<td>CSTRH&lt;3:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0x31</td>
<td>EDCON</td>
<td>r</td>
<td>r</td>
<td>EDMODE</td>
<td>EDST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0x32</td>
<td>EDMEAN</td>
<td></td>
<td></td>
<td>EDMEAN&lt;7:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0x33</td>
<td>EDPEAK</td>
<td></td>
<td></td>
<td>EDPEAK&lt;7:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0x34</td>
<td>CFOCON</td>
<td></td>
<td></td>
<td>CFOTX&lt;3:0&gt;</td>
<td>CFORX&lt;3:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0x35</td>
<td>CFOMEAS</td>
<td></td>
<td></td>
<td>CFOMEAS&lt;7:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0x36</td>
<td>RATECON</td>
<td>DIS2000</td>
<td>DIS1000</td>
<td>DIS500</td>
<td>DIS125</td>
<td>DIS250</td>
<td>DISSTD</td>
<td>DIS125</td>
<td>OPTIMAL</td>
<td>PSAV&lt;3:0&gt;</td>
</tr>
<tr>
<td></td>
<td>0x37</td>
<td>POWSAVE</td>
<td>DESENTHR&lt;3:0&gt;</td>
<td></td>
<td></td>
<td>PSAVTHR&lt;3:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TX AND EXTDEV</td>
<td>0x38</td>
<td>BBCON</td>
<td>RNDMOD</td>
<td>AFCOVR</td>
<td>RXGAIN&lt;1:0&gt;</td>
<td>PRMBHOLD</td>
<td>PRMBSZ&lt;2:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0x39</td>
<td>IFGAP</td>
<td>r</td>
<td>r</td>
<td>r</td>
<td>IFGAP&lt;4:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0x3A</td>
<td>TXPOW</td>
<td>CHIPBOOST&lt;2:0&gt;</td>
<td></td>
<td></td>
<td>TXPOW&lt;4:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>0x3B</td>
<td>TX2IDLE</td>
<td>r</td>
<td>r</td>
<td>r</td>
<td>TX2IDLE&lt;4:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>61</td>
</tr>
<tr>
<td></td>
<td>0x3C</td>
<td>TX2TXMA</td>
<td>r</td>
<td>r</td>
<td></td>
<td>TX2TXMA&lt;4:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>61</td>
</tr>
<tr>
<td></td>
<td>0x3D</td>
<td>EXTPA</td>
<td>EXTPA_P</td>
<td>PAEN</td>
<td></td>
<td>PA2TXMA&lt;4:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>62</td>
</tr>
<tr>
<td></td>
<td>0x3E</td>
<td>EXTLNA</td>
<td>EXTLNA_P</td>
<td>LNAEN</td>
<td></td>
<td>LNADLY&lt;4:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>62</td>
</tr>
<tr>
<td></td>
<td>0x3F</td>
<td>BATMON</td>
<td>r</td>
<td>r</td>
<td>BATMONPD</td>
<td>BATMON&lt;4:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>63</td>
</tr>
</tbody>
</table>
## TABLE 2-7: LONG ADDRESS REGISTER SUMMARY FOR MRF24XA

<table>
<thead>
<tr>
<th>Address</th>
<th>Name</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC</td>
<td>SECKEY1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>63</td>
</tr>
<tr>
<td>0x40</td>
<td>SECKEY2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>63</td>
</tr>
<tr>
<td>0x41</td>
<td>SECKEY3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>63</td>
</tr>
<tr>
<td>0x42</td>
<td>SECKEY4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>63</td>
</tr>
<tr>
<td>0x43</td>
<td>SECKEY5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>63</td>
</tr>
<tr>
<td>0x44</td>
<td>SECKEY6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>63</td>
</tr>
<tr>
<td>0x45</td>
<td>SECKEY7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>63</td>
</tr>
<tr>
<td>0x46</td>
<td>SECKEY8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>63</td>
</tr>
<tr>
<td>0x47</td>
<td>SECKEY9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>63</td>
</tr>
<tr>
<td>0x48</td>
<td>SECKEY10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>63</td>
</tr>
<tr>
<td>0x49</td>
<td>SECKEY11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>63</td>
</tr>
<tr>
<td>0x4A</td>
<td>SECKEY12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>63</td>
</tr>
<tr>
<td>0x4B</td>
<td>SECKEY13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>63</td>
</tr>
<tr>
<td>0x4C</td>
<td>SECKEY14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>63</td>
</tr>
<tr>
<td>0x4D</td>
<td>SECKEY15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>63</td>
</tr>
<tr>
<td>0x4E</td>
<td>SECKEY16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>63</td>
</tr>
<tr>
<td>0x4F</td>
<td>SECNONCE1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>65</td>
</tr>
<tr>
<td>0x50</td>
<td>SECNONCE2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>65</td>
</tr>
<tr>
<td>0x51</td>
<td>SECNONCE3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>65</td>
</tr>
<tr>
<td>0x52</td>
<td>SECNONCE4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>65</td>
</tr>
<tr>
<td>0x53</td>
<td>SECNONCE5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>65</td>
</tr>
<tr>
<td>0x54</td>
<td>SECNONCE6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>65</td>
</tr>
<tr>
<td>0x55</td>
<td>SECNONCE7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>65</td>
</tr>
<tr>
<td>0x56</td>
<td>SECNONCE8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>65</td>
</tr>
<tr>
<td>0x57</td>
<td>SECNONCE9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>65</td>
</tr>
<tr>
<td>0x58</td>
<td>SECNONCE10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>65</td>
</tr>
<tr>
<td>0x59</td>
<td>SECNONCE11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>65</td>
</tr>
<tr>
<td>0x5A</td>
<td>SECNONCE12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>65</td>
</tr>
<tr>
<td>0x5B</td>
<td>SECNONCE13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>65</td>
</tr>
<tr>
<td>0x5C</td>
<td>SECENCFLAG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>65</td>
</tr>
<tr>
<td>0x5D</td>
<td>SECAUTHFLAG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>65</td>
</tr>
<tr>
<td>0x5E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x5F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PHY</td>
<td>SFD1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>66</td>
</tr>
<tr>
<td>0x60</td>
<td>SFD2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>67</td>
</tr>
<tr>
<td>0x61</td>
<td>SFD3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>67</td>
</tr>
<tr>
<td>0x62</td>
<td>SFD4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>68</td>
</tr>
<tr>
<td>0x63</td>
<td>SFD5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>68</td>
</tr>
<tr>
<td>0x64</td>
<td>SFD6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>69</td>
</tr>
<tr>
<td>0x65</td>
<td>SFD7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>69</td>
</tr>
<tr>
<td>0x66</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x67</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x68</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x69</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x6A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x6B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x6C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x6D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x6E</td>
<td>SFDTO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>69</td>
</tr>
<tr>
<td>0x7F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2.6.2 ADDRESS
When using a Short Addressing mode, the Address field is 6 bits wide to reduce framing overhead while accessing the mostly active registers (0x00..0x3F). In Long Addressing mode, the Address field is 10 bits wide (0x00..0x3FF) thus all the address is available for SPI operation.

2.6.3 AUTOMATIC TX START FEATURE
When a write to TRXBUF is done using Long Addressing mode and the 3rd bit of Byte 2 is set, the TXST bit automatically sets after the CS pin is released, and then MRF24XA sends the packet.

2.6.4 AUTOMATIC BUFFER FLUSH FEATURE
When a read from TRXBUF is done using Long Addressing mode and the 3rd bit of Byte 2 is set, the BUFFULL bit automatically becomes cleared after the CS pin is negated.

2.6.5 ADDRESS AUTO-INCREMENT FEATURE
After the starting address is loaded, the first byte of data is read from or written to this address. The second byte (assuming the CS pin is not negated between bytes) is read from or written to the starting address plus one, and so on.

If the memory map end is reached, the effective address rolls over to the beginning of the memory map. It is the sole responsibility of the software to handle this situation correctly. Figure 2-4 illustrates the available Address modes.

FIGURE 2-4: SPI FRAMING TYPES
## 2.7 Register Details

### REGISTER 2-1: OPSTATUS (OPERATION STATUS)

<table>
<thead>
<tr>
<th>Address: 0x02</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-0</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>MACOP&lt;3:0&gt;</td>
</tr>
</tbody>
</table>

Legend:
- R = Readable bit
- W = Writable bit
- U = Unimplemented bit, read as '0'
- -n = Value at POR
- '1' = Bit is set
- '0' = Bit is cleared
- x = Bit is unknown
- r = Reserved
- HC = Hardware Clear
- HS = Hardware Set

#### bit 7
**Reserved**: Maintain as '0'

#### bit 6-3
**MACOP<3:0>**: MAC Operation Register bits

- 111 = Transmitting Acknowledge (TXACK)
- 110 = Receiving a packet (RXBUSY)
- 101 = Receiver listening to the channel waiting for packet (RX)
- 100 = Receiving (or waiting for) Acknowledge (RXACK)
- 011 = Transmitting a packet (TX)
- 010 = Performing Clear Channel Assessment (CCA)
- 001 = Back-off before repeated CCA (BO)
- 000 = MAC does not perform any operation (IDLE)

#### bit 2-0
**RFOP<2:0>**: Radio Operation Register bits

- 111 = TX with external PA is turned on (TX+PA)
- 110 = RX with external LNA is turned on (RX+LNA)
- 101 = Synthesizer and external PA or LNA is turned on (SYNTH+PA/LNA)
- 100 = Radio calibrates if the host MCU sets the CALST, otherwise, device malfunction occurs (CAL/MAL)
- 011 = Analog transmit chain is activated (TX)
- 010 = Analog receiver chain is active (RX). Digital may be partially shut off
- 001 = Synthesizer is steady or ramping up or channel change is issued (SYNTH)
- 000 = Only the crystal oscillator is ON (OFF), (except when XTALSF = 1)

**Note 1**: GPIO<2:0> is dedicated to output MACOP<3:1> or RFOP<2:0>. Refer to the PINCON register, which specifies the pin Configuration.

**Note 2**: MACOP<0> is connected to the RXBUFFUL register bit. It cannot be output over GPIO’s.

**Note 3**: The OPSTATUS register is sent on the SDO pin during the first byte of the SPI operation.
REGISTER 2-2: STATUS (DEVICE STATUS) Address: 0x03

<table>
<thead>
<tr>
<th>R/HS</th>
<th>R/HS</th>
<th>R/HS</th>
<th>R/W/HC-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/HS</th>
<th>R/W/HC</th>
</tr>
</thead>
<tbody>
<tr>
<td>INITDONESF</td>
<td>XTALSF</td>
<td>REGSF</td>
<td>CALST</td>
<td>XTALDIS</td>
<td>DSLEEP</td>
<td>IDLESF</td>
<td>POR</td>
</tr>
</tbody>
</table>

Legend:  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as ‘0’  
-n = Value at POR  
‘1’ = Bit is set  
‘0’ = Bit is cleared  
x = Bit is unknown  
r = Reserved  
HC = Hardware Clear  
HS = Hardware Set

bit 7  
INITDONESF: Device Initialization Status Flag bit  
Indicates that the ready state is reached since the LDO is on. (If VREGIF = 1). INITDONESF is asserted when RDYIF is set for the first time after VREGIF. This bit is only cleared on reset (POR, DEVFRST and PINRST).

bit 6  
XTALSF: Crystal Status Flag bit  
XTALSF = 1, indicates that 16 MHz system clock (from the crystal oscillator) is active. This bit is cleared either when XTALDIS is set or reset (POR, DEVFRST, PINRST).
XTALSF = 0, indicates that the crystal oscillator is either powered off (XTALDIS = 1) or is ramping up or is not stabilized yet, and the system clock is inactive.

bit 5  
REGSF: Configuration Registers Status Flag bit  
REGSF = 1 indicates that all the 1.2V register content is valid. Either it holds the default value after reset and the retention memory does not hold any data to restore, or the register configurations are restored from the retention memory.
REGSF = 0 indicates that registers from 0x08-0x6E are invalid. This occurs when the wake-up procedure from Deep Sleep mode did not complete the register restore operation yet. This bit is only cleared on Reset (POR, DEVFRST, and PINRST).

bit 4  
CALST: Calibration Start bit  
MCU sets this bit to start Calibration procedure after a CALSOIF or CALHAIF interrupt occurred. MCU may not clear it to abort Calibration. The device clears CALST when the Calibration is completed (CALHAIF = 0 indicates success, CALHAIF = 1 indicates failure). Issuing CALST operation without CALHAIF/CALSOIF terminates without any effect on the device.

bit 3  
XTALDIS: Crystal Disable bit  
MCU sets this bit to send the device into XTAL OFF state (reachable from ready state). XTALSF automatically gets cleared. The SPI register access is performed when crystal is not working.

bit 2  
DSLEEP: Deep-Sleep bit  
MCU sets this bit to send the device into Deep Sleep state. Following DSLEEP = 1, the SPI access to the SFR is shut off, and the SPI pins must be quite, unless the host MCU wants to wake-up the device. When DSLEEP is set, the device transitions through register backup (taking cca. 16 µs) before LDO is powered off.

bit 1  
IDLESF: Idle Status Flag bit  
Indicates Idle state of the device when all of the following bits are deasserted:
  • TXBUFEMPTY = 0 since it is transmitted (TXST)
  • Network layer security finished (TXENC)
  • Crypto engine finished (RXDEC)
  • Energy detect operation finished (EDST)
  • Clear Channel Assessment finished (CCAST)

bit 0  
POR: Power-on-Reset Flag bit  
The 3.3V POR flag status. The device sets this only on 3.3V power-up (e.g., when battery is changed). Cleared by host MCU to be able to sense a Brown-out Reset (BOR). Settable for software testing.
REGISTER 2-3: PIR1 (PERIPHERAL INTERRUPT REGISTER 1)  Address: 0x04

<table>
<thead>
<tr>
<th>Address</th>
<th>bit 7</th>
<th>bit 6</th>
<th>bit 5</th>
<th>bit 4</th>
<th>bit 3</th>
<th>bit 2</th>
<th>bit 1</th>
<th>bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>VREGIF</td>
<td>r</td>
<td>RDSF</td>
<td>IDLEF</td>
<td>r</td>
<td>CALSOF</td>
<td>CALHAIF</td>
<td>r</td>
<td></td>
</tr>
</tbody>
</table>

Legend:  
- R = Readable bit  
- W = Writable bit  
- U = Unimplemented bit, read as ‘0’  
- -n = Value at POR  
- ‘1’ = Bit is set  
- ‘0’ = Bit is cleared  
- x = Bit is unknown  
- r = Reserved  
- HC = Hardware Clear  
- HS = Hardware Set

bit 7  
**VREGIF:** Voltage Regulator On Interrupt Flag bit  
This is a nonpersistent bit. The register bit initializes to one on 1.2V reset except for PINRESET and only clears when reading PIR1. Note that the corresponding IE bit is not implemented(1).

bit 6  
Reserved: Maintain as ‘0’

bit 5  
**RDYIF:** Ready State Interrupt Flag bit  
Set each time when ready state is reached:  
- when Calibration ended (CALST = 0)  
- when initialization ended (INITDONESF = 1)  
- when crystal is ramped up (XTALSF = 1)  

This bit is cleared when PIR1 is read.

bit 4  
**IDLEF:** Idle State Interrupt Flag bit  
Set each time the IDLESF is set and if MCU did not trigger this change. This is unchanged when MCU aborts an action by clearing either of TXST, TXENC, RXDEC or EDST bits. This bit is cleared when PIR1 is read.

bit 3  
Reserved: Maintain as ‘0’

bit 2  
**CALSOIF:** Calibration Soft Interrupt Flag bit  
CALSOIF = 1 indicates that Calibration is needed (CALST) although the radio is still functional. It also warns of a possible degradation in signal quality and consumption, and a risk of CALHAIF interrupt. This bit is cleared when PIR1 is read.

bit 1  
**CALHAIF:** Calibration Hard Interrupt Flag bit  
CALHAIF = 1 indicates that immediate Calibration (CALST) is mandatory, otherwise the radio is not functional. The device enters into malfunction state. This bit is cleared when PIR1 is read.

bit 0  
Reserved: Maintain as ‘0’

**Note 1:** Generated non-maskable interrupt is gated off until the 1.2V reset is released.
## REGISTER 2-4: PIR2 (PERIPHERAL INTERRUPT REGISTER 2)

<table>
<thead>
<tr>
<th>TXIF</th>
<th>TXENCIF</th>
<th>TXMAIF</th>
<th>TXACKIF</th>
<th>TXCSMAIF</th>
<th>TXSZIF</th>
<th>TXOVFIF</th>
<th>FRMIF</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 7</td>
<td>bit 6</td>
<td>bit 5</td>
<td>bit 4</td>
<td>bit 3</td>
<td>bit 2</td>
<td>bit 1</td>
<td>bit 0</td>
</tr>
</tbody>
</table>

**Legend:**
- R = Readable bit
- W = Writable bit
- U = Unimplemented bit, read as ‘0’
- -n = Value at POR
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- x = Bit is unknown
- r = Reserved
- HC = Hardware Clear
- HS = Hardware Set

**bit 7**
- **TXIF:** Transmission Done Interrupt Flag bit
  - The current TX operation (TXST) is successfully completed. This event is unchanged when a hardware generated ACK packet completed the transmission or when a packet is repeated. Nonpersistent, cleared by SPI read.

**bit 6**
- **TXENCIF:** Transmit Encryption Interrupt Flag bit
  - The TX packet is successfully encrypted or complemented, or both with a Message Integrity Code (MIC). Set by the device after TXENC = 1, when TXENC is cleared. Nonpersistent, cleared by SPI read.

**bit 5**
- **TXMAIF:** Transmitter Medium Access Interrupt Flag bit
  - Set by the device when the medium is accessed specifically when the first sample in the preamble is transmitted into the air. Nonpersistent, cleared by SPI read.

**bit 4**
- **TXACKIF:** Transmission Unacknowledged Failure Interrupt Flag bit
  - Set by the device when Acknowledge is not received after the configured maximum number of transmission retries RETXMCNT<3:0>, provided that the Frame Control field of the transmitted frame indicates AckReq = 1. Nonpersistent, cleared by SPI read.

**bit 3**
- **TXCSMAIF:** Transmitter CSMA Failure Interrupt Flag bit
  - Set by the device when CSMA-CA finds the channel busy for BOMCNT<2:0> number of times, provided that CSMAEN = 1 is configured. Nonpersistent, cleared by SPI read.

**bit 2**
- **TXSZIF:** Transmit Packet Size Error Interrupt Flag bit
  - Set by the device if TX packet size (first byte of the TX buffer) is found to be zero or greater than the maximum size that the buffer can support. Automatic size check is performed after TXST is set by the user. Please note that the device may modify the packet length after CRC or MIC calculation. Nonpersistent, cleared by SPI read.

**bit 1**
- **TXOVFIF:** Transmitter Overflow Interrupt Flag bit
  - The Host Controller attempted to write a TX buffer that was not empty (TXBUFEMPTY = 0).
  - Nonpersistent, cleared by SPI read.

**bit 0**
- **FRMIF:** Frame Format Error Interrupt Flag bit
  - Set if the transmitter/receiver fails to parse the frame in the buffer (it is not as it must be or it is corrupted in demodulation).
REGISTER 2-5: PIR3 (PERIPHERAL INTERRUPT REGISTER 3)  ADDRESS: 0x06

<table>
<thead>
<tr>
<th>bit 7</th>
<th>bit 6</th>
<th>bit 5</th>
<th>bit 4</th>
<th>bit 3</th>
<th>bit 2</th>
<th>bit 1</th>
<th>bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>RXIF</td>
<td>RXDECIF</td>
<td>RXTAGIF</td>
<td>r</td>
<td>RXIDENTIF</td>
<td>RXFLTIF</td>
<td>RXOVFIF</td>
<td>STRMIF</td>
</tr>
</tbody>
</table>

Legend:  
- R = Readable bit  
- W = Writable bit  
- U = Unimplemented bit, read as ‘0’  
- -n = Value at POR  
- ‘1’ = Bit is set  
- ‘0’ = Bit is cleared  
- x = Bit is unknown

bit 7  
**RXIF:**Received Successful Interrupt Flag bit  
Set by the device when a frame passed packet filtering and accepted, refer to Register 2-23. This interrupt flag is only set once for a packet and is not set when the packet is the duplicate of a repeated transmission (sequence number matches with the previously received frame). Nonpersistent, cleared by SPI read.

bit 6  
**RXDECIF:**Receiver Decryption/Authentication Passed Interrupt Flag bit  
Set by the device when decryption/authentication finished without error. Nonpersistent, cleared by SPI read.

bit 5  
**RXTAGIF:**Receiver Decryption/Authentication Failure Interrupt Flag bit  
Set by the device when decryption/authentication finished with error. Nonpersistent, cleared by SPI read.

bit 4  
Reserved: Maintain as ‘0’

bit 3  
**RXIDENTIF:**Received Packet Identical Interrupt Flag bit  
Set by the device when the packet is the duplicate of a repeated transmission (sequence number and source address matches with the previously received frame). Nonpersistent, cleared by SPI read.

bit 2  
**RXFLTIF:**Received Packet Filtered Interrupt Flag bit  
Set by the device when a packet is received, but rejected by one or more RX Filters, refer to Register 2-23. Nonpersistent, cleared by SPI read.

bit 1  
**RXOVFIF:**Receiver Overflow Error Interrupt Flag bit  
Set by the device to indicate that a packet was received, but all RX buffers were full. Consequently the packet was not received, but was discarded instead\(^1\). Nonpersistent, cleared by SPI read.

bit 0  
**STRMIF:**Receive Stream Time-Out Error Interrupt Flag bit  
Set by the device to indicate that the duration specified in STRMTO elapsed since the last received packet while in RX-Streaming mode, and the MAC clears the stored sequence number. Nonpersistent, cleared by SPI read.

**Note 1:** In Packet mode, use a single buffer for received frames. In RX-Streaming mode, use both buffers for reception.
### REGISTER 2-6: PIR4 (PERIPHERAL INTERRUPT REGISTER 4)\(^{(1)}\)  
**ADDRESS:** 0x07

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description &amp; Notes</th>
</tr>
</thead>
</table>
| 7   | TXSFDIF: Transmit SFD Sent Interrupt Flag bit  
      | Set by the device when the last sample of the SFD field is sent into the air.  
      | Nonpersistent, cleared by SPI read |
| 6   | RXSFDIF: Receive SFD Detected Interrupt Flag bit  
      | Set by the device when the SFD field of the received frame is detected.  
      | Nonpersistent, cleared by SPI read. |
| 5   | ERRORIF: General Error Interrupt Flag bit  
      | Set by the device, when malfunction state is reached. |
| 4   | WARNIF: Warning Interrupt Flag bit  
      | Set by the device when one of the following occurred:  
      | • Battery voltage drops below the threshold by BATMON<4:0> at 0x3F  
      | • Indicates that resistor is missing or improperly connected. |
| 3   | EDCCAIF: Energy Detect/CCA Done Interrupt Flag bit  
      | Set by the device when Energy-detect or CCA measurement is complete (following that the host MCU sets the EDST/CCAST bit to start the measurement and the device is clearing it for completion).  
      | Nonpersistent. Cleared by SPI read. |
| 2   | GPIO2IF: GPIO2 Interrupt Flag bit  
      | Set by the device if the GPIOMODE register is set to normal operation, the GPIO is enabled and configured to input and the level matches with the polarity. |
| 1   | GPIO1IF: GPIO1 Interrupt Flag bit  
      | Set by the device if the GPIOMODE register is set to normal operation, the GPIO is enabled and configured to input and the level matches with the polarity. |
| 0   | GPIO0IF: GPIO0 Interrupt Flag bit  
      | Set by the device if the GPIOMODE register is set to normal operation, the GPIO is enabled and configured to input and the level matches with the polarity. |

**Note 1:** CFOMEAS<7:0> indication becomes valid on SFD found.
## REGISTER 2-7: PIE1 (PERIPHERAL INTERRUPT ENABLE 1)  
**ADDRESS:** 0x08

<table>
<thead>
<tr>
<th>bit 7-6</th>
<th>bit 5</th>
<th>bit 4</th>
<th>bit 3</th>
<th>bit 2</th>
<th>bit 1</th>
<th>bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reserved: Maintain as ‘0’</td>
<td>RDYIE: Ready Interrupt Enable bit</td>
<td>IDLEIE: Idle Interrupt Enable bit</td>
<td>Reserved: Maintain as ‘0’</td>
<td>CALSOIE: Calibration Soft Interrupt Enable bit</td>
<td>CALHAIE: Calibration Hard Interrupt Enable bit</td>
<td>Reserved: Maintain as ‘0’</td>
</tr>
</tbody>
</table>

### Legend:
- **R** = Readable bit  
- **W** = Writable bit  
- **U** = Unimplemented bit, read as ‘0’  
- **-n** = Value at POR  
- ‘1’ = Bit is set  
- ‘0’ = Bit is cleared  
- x = Bit is unknown  
- r = Reserved
## MRF24XA

### REGISTER 2-8: PIE2 (PERIPHERAL INTERRUPT ENABLE 2)

<table>
<thead>
<tr>
<th>Bit</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>TXIE</td>
<td>Transmit Interrupt Enable bit. This bit masks the TXIF interrupt bit.</td>
</tr>
<tr>
<td>6</td>
<td>TXENCIE</td>
<td>Transmit Encryption and Authentication Interrupt Enable bit. This bit masks the TXENCIF interrupt bit.</td>
</tr>
<tr>
<td>5</td>
<td>TXMAIE</td>
<td>Transmitter Medium Access Interrupt Enable bit. This bit masks the TXMAIF interrupt bit.</td>
</tr>
<tr>
<td>4</td>
<td>TXACKIE</td>
<td>Transmission Unacknowledged Failure Interrupt Enable bit. This bit masks the TXACKIF interrupt bit.</td>
</tr>
<tr>
<td>3</td>
<td>TXCSMAIE</td>
<td>Transmitter CSMA Failure Interrupt Enable bit. This bit masks the TXCSMAIF interrupt bit.</td>
</tr>
<tr>
<td>2</td>
<td>TXSZIE</td>
<td>Transmit Packet Size Error Interrupt Enable bit. This bit masks the TXSZIF interrupt bit.</td>
</tr>
<tr>
<td>1</td>
<td>TXOVFIE</td>
<td>Transmitter Overflow Interrupt Enable bit. This bit masks the TXOVFIF interrupt bit.</td>
</tr>
<tr>
<td>0</td>
<td>FRMIE</td>
<td>Frame Format Error Interrupt Enable bit. This bit masks the FRMIF interrupt bit.</td>
</tr>
</tbody>
</table>

**Legend:**
- **R** = Readable bit
- **W** = Writable bit
- **U** = Unimplemented bit, read as ‘0’
- **-n** = Value at POR
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- **x** = Bit is unknown
- **r** = Reserved

---

© 2015 Microchip Technology Inc.

Preliminary

DS70005023C-page 25
REGISTER 2-9: PIE3 (PERIPHERAL INTERRUPT ENABLE 3)  
ADDRESS: 0x0A

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>RXIE: Received Successful Interrupt Enable bit</td>
</tr>
<tr>
<td></td>
<td>This bit masks the RXIF interrupt bit.</td>
</tr>
<tr>
<td>6</td>
<td>RXDECIE: Receiver Decryption/Authentication Passed Interrupt Enable bit</td>
</tr>
<tr>
<td></td>
<td>This bit masks the RXDECIF interrupt bit.</td>
</tr>
<tr>
<td>5</td>
<td>RXTAGIE: Receiver Decryption/Authentication Failure Interrupt Enable bit</td>
</tr>
<tr>
<td></td>
<td>This bit masks the RXTAGIF interrupt bit.</td>
</tr>
<tr>
<td>4</td>
<td>Reserved: Maintain as ‘0’</td>
</tr>
<tr>
<td>3</td>
<td>RXIDENTIE: Received Packet Identical Interrupt Enable bit</td>
</tr>
<tr>
<td></td>
<td>This bit masks the RXIDENTIF interrupt bit.</td>
</tr>
<tr>
<td>2</td>
<td>RXFLTIE: Received Packet Filtered Interrupt Enable bit</td>
</tr>
<tr>
<td></td>
<td>This bit masks the RXFLTIF interrupt bit.</td>
</tr>
<tr>
<td>1</td>
<td>RXOVFIE: Receiver Overflow Interrupt Enable bit</td>
</tr>
<tr>
<td></td>
<td>This bit masks the RXOVFIF interrupt bit.</td>
</tr>
<tr>
<td>0</td>
<td>STRMIE: Receive Stream Time-Out Error Interrupt Enable bit</td>
</tr>
<tr>
<td></td>
<td>This bit masks the STRMIF interrupt bit.</td>
</tr>
</tbody>
</table>

Legend:  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as ‘0’  
-n = Value at POR  
‘1’ = Bit is set  
‘0’ = Bit is cleared  
x = Bit is unknown  
r = Reserved
### REGISTER 2-10: PIE4 (PERIPHERAL INTERRUPT ENABLE 4)

**Address:** 0x0B

<table>
<thead>
<tr>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-1</th>
<th>RW-1</th>
<th>RW-1</th>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>TXSFDIE</td>
<td>RXSFDIE</td>
<td>ERRORIE</td>
<td>WARNIE</td>
<td>EDCCAIE</td>
<td>GPIO2IE</td>
<td>GPIO1IE</td>
<td>GPIO0IE</td>
</tr>
</tbody>
</table>

#### Bit 7
- **TXSFDIE:** Transmit SFD Sent Interrupt Enable bit
  - This bit masks the TXSFDIF interrupt bit.

#### Bit 6
- **RXSFDIE:** Receive SFD Detected Interrupt Enable bit
  - This bit masks the RXSFDIF Interrupt Enable.

#### Bit 5
- **ERRORIE:** General Error Interrupt Enable bit
  - This bit masks the ERRORIF interrupt bit.

#### Bit 4
- **WARNIE:** Warning Interrupt Enable bit
  - This bit masks the WARNIF interrupt bit.

#### Bit 3
- **EDCCAIE:** Energy Detect/CCA Done Interrupt Enable bit
  - This bit masks the EDCCAIF interrupt bit.

#### Bit 2
- **GPIO2IE:** GPIO2 Interrupt Enable bit
  - This bit masks the GPIO2IF interrupt bit.

#### Bit 1
- **GPIO1IE:** GPIO1 Interrupt Enable bit
  - This bit masks the GPIO1IF interrupt bit.

#### Bit 0
- **GPIO0IE:** GPIO0 Interrupt Enable bit
  - This bit masks the GPIO0IF interrupt bit.
### REGISTER 2-11: PINCON (PIN CONFIGURATION REGISTER)  
**ADDRESS:** 0x0C

<table>
<thead>
<tr>
<th>R-0</th>
<th>R/W-1</th>
<th>R-0</th>
<th>R-1</th>
<th>R/W-0000</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>GIE</td>
<td>r</td>
<td>IRQIF</td>
<td>GPIOMODE&lt;3:0&gt;</td>
</tr>
</tbody>
</table>

**Legend:**
- R = Readable bit
- W = Writable bit
- U = Unimplemented bit, read as '0'
- -n = Value at POR
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- x = Bit is unknown
- r = Reserved

**bit 7**
- **Reserved:** Maintain as ‘0’

**bit 6**
- **GIE:** General Interrupt Enable bit
  - This bit enables to output IRQIF on INT pin. Note that the polarity of INT pin is active-low.

**bit 5**
- **Reserved:** Maintain as ‘0’

**bit 4**
- **IRQIF:** Interrupt Request Pending bit
  - This bit is the OR relationship of the enabled interrupt flags.

**bit 3-0**
- **GPIOMODE <3:0>:** GPIO Mode Field bits
  - This field enables redefining the functionality of the GPIO pins
    - **Encoding:**
      - **11xx** = Reserved
      - **1011** = GPIO pins are used for Receive streaming (RXSTREAM). Pins GPIO<2:0> are used to output \{RXWRBUF, BUSRDBUF, RXBUFFUL\}
      - **1010** = GPIO pins are used for Transmit streaming (TXSTREAM). Pins GPIO<2:0> are used to output \{TXRDBUF, BUSWRBUF, TXBUFFIELD\}
      - **1001** = Reserved
      - **1000** = Reserved
      - **0111** = Reserved
      - **0110** = Reserved
      - **0101** = Intended for supporting Precise Network Time Synchronization (TIMESYN). GPIO<0> is used to output TX, while GPIO<1> to output RX SFD indication pulses. GPIO<2> is used in "NORMAL" operation mode.
      - **0100** = GPIO pins are used for Radio monitoring (RFMON). Pins GPIO<2:0> are used to output RFOP<2:0>.
      - **0011** = GPIO pins are used for MAC monitoring (MACMON). Pins GPIO<2:0> are used to output MACOP<3:1>.
      - **0010** = GPIO pins are used for RXFSM monitoring (RXFSMMON). Pins GPIO<2:0> are used to output receiver state-machine.
        - **000** = Preamble search
        - **001** = Hi-rate SFD search
        - **010** = Mid-rate SFD search
        - **011** = Low-rate SFD search
        - **100** = Legacy Length field processing
        - **101** = Payload processing
      - **0001** = GPIO pins are used for AGC monitoring (AGCMON). Pins GPIO<2:0> are used to output \{AGC-HOLD, GAIN<1:0>\} where AGCHOLD is an internal flag set when a receiver detects a preamble and clears when the AGC is set free after the end of the frame.
      - **0000** = GPIO pins are used as General Purpose I/O’s by the host MCU (NORMAL).
<table>
<thead>
<tr>
<th>Bit</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>GPIOEN</td>
<td>GPIO Enable bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This bit enables the GPIO’s control, if GPIOMODE is configured into Normal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>mode. The other GPIOMODE Configuration automatically controls GPIO pins.</td>
</tr>
<tr>
<td>6</td>
<td>TRISGPIO2</td>
<td>Tri-state Control for GPIO 2 Pin bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If set, the pin is configured into Input mode. Value reads from GPIO2 bit.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If cleared, the pin is configured into Output mode. Value sets through the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GPIO2 bit.</td>
</tr>
<tr>
<td>5</td>
<td>TRISGPIO1</td>
<td>Tri-state Control for GPIO 1 Pin bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If set, the pin is configured into Input mode. Value reads from GPIO1 bit.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If cleared, the pin is configured into Output mode. Value sets through the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GPIO1 bit.</td>
</tr>
<tr>
<td>4</td>
<td>TRISGPIO0</td>
<td>Tri-state Control for GPIO 0 Pin bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If set, the pin is configured into Input mode. Value reads from GPIO0 bit.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If cleared, the pin is configured into Output mode. Value sets through the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GPIO0 bit.</td>
</tr>
<tr>
<td>3</td>
<td>Reserved</td>
<td>Maintain as ‘0’</td>
</tr>
<tr>
<td>2</td>
<td>GPIO2</td>
<td>GPIO 2 Value bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This bit represents the value on the GPIO 2 pin.</td>
</tr>
<tr>
<td>1</td>
<td>GPIO1</td>
<td>GPIO 1 Value bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This bit represents the value on the GPIO 1 pin.</td>
</tr>
<tr>
<td>0</td>
<td>GPIO0</td>
<td>GPIO 0 Value bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This bit represents the value on the GPIO 0 pin.</td>
</tr>
</tbody>
</table>

Legend:
- **R** = Readable bit
- **W** = Writable bit
- **U** = Unimplemented bit, read as ‘0’
- **-n** = Value at POR
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- **x** = Bit is unknown
- **r** = Reserved
## REGISTER 2-13: STGPIO (SCHMITT TRIGGER GENERAL PURPOSE I/O REGISTER)

**ADDRESS:** 0x0E

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Value at POR</th>
<th>Readable/Writeable</th>
<th>R/W-0</th>
<th>Reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td><strong>Reserved</strong>: Maintain as ‘0’</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>GPIO2P: GPIO 2 Polarity bit</td>
<td></td>
<td>R/W-0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>This bit controls GPIO2IF polarity when configured into Input mode.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = Rising edge</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Falling edge</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>GPIO1P: GPIO 1 Polarity bit</td>
<td></td>
<td>R/W-0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>This bit controls GPIO1IF polarity when configured into Input mode.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = Rising edge</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Falling edge</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>GPIO0P: GPIO 0 Polarity bit</td>
<td></td>
<td>R/W-0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>This bit controls GPIO0IF polarity when configured into Input mode.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = Rising edge</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Falling edge</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td><strong>Reserved</strong>: Maintain as ‘0’</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>STENGPIO2: Schmitt Trigger Enable GPIO 2 bit</td>
<td></td>
<td>R/W-0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>This bit enables Schmitt-trigger circuit on GPIO 2 pad and turns off by default.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = Schmitt trigger enabled</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Schmitt trigger disabled</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>STENGPIO1: Schmitt Trigger Enable GPIO 1 bit</td>
<td></td>
<td>R/W-0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>This bit enables Schmitt-trigger circuit on GPIO 1 pad and turns off by default.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = Schmitt trigger enabled</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Schmitt trigger disabled</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>STENGPIO0: Schmitt Trigger Enable GPIO 0 bit</td>
<td></td>
<td>R/W-0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>This bit enables Schmitt-trigger circuit on GPIO 0 pad and turns off by default.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = Schmitt trigger enabled</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Schmitt trigger disabled</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Legend:**
- **R** = Readable bit
- **W** = Writable bit
- **U** = Unimplemented bit, read as ‘0’
- **-n** = Value at POR
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- **x** = Bit is unknown
- **r** = Reserved

### Bit 7
- **Reserved**: Maintain as ‘0’

### Bit 6
- **GPIO2P**: GPIO 2 Polarity bit
- This bit controls GPIO2IF polarity when configured into Input mode.
- 1 = Rising edge
- 0 = Falling edge

### Bit 5
- **GPIO1P**: GPIO 1 Polarity bit
- This bit controls GPIO1IF polarity when configured into Input mode.
- 1 = Rising edge
- 0 = Falling edge

### Bit 4
- **GPIO0P**: GPIO 0 Polarity bit
- This bit controls GPIO0IF polarity when configured into Input mode.
- 1 = Rising edge
- 0 = Falling edge

### Bit 3
- **Reserved**: Maintain as ‘0’

### Bit 2
- **STENGPIO2**: Schmitt Trigger Enable GPIO 2 bit
- This bit enables Schmitt-trigger circuit on GPIO 2 pad and turns off by default.
- 1 = Schmitt trigger enabled
- 0 = Schmitt trigger disabled

### Bit 1
- **STENGPIO1**: Schmitt Trigger Enable GPIO 1 bit
- This bit enables Schmitt-trigger circuit on GPIO 1 pad and turns off by default.
- 1 = Schmitt trigger enabled
- 0 = Schmitt trigger disabled

### Bit 0
- **STENGPIO0**: Schmitt Trigger Enable GPIO 0 bit
- This bit enables Schmitt-trigger circuit on GPIO 0 pad and turns off by default.
- 1 = Schmitt trigger enabled
- 0 = Schmitt trigger disabled
**REGISTER 2-14: PULL_GPIO (PULL CONTROL GENERAL PURPOSE I/O REGISTER)**

**ADDRESS: 0x0F**

| bit 7 | Reserved: Maintain as ‘0’ |
| bit 6 | PULLDIR_GPIO2: Pull Direction on GPIO 2 bit |
|      | These bits control the 75 kOhm weak-pull circuit direction on GPIO 2 pin. |
|      | 1 = Pull-up |
|      | 0 = Pull-down |
| bit 5 | PULLDIR_GPIO1: Pull Direction on GPIO 1 bit |
|      | These bits control the 75 kOhm weak-pull circuit direction on GPIO 1 pin. |
|      | 1 = Pull-up |
|      | 0 = Pull-down |
| bit 4 | PULLDIR_GPIO0: Pull Direction on GPIO 0 bit |
|      | These bits control the 75 kOhm weak-pull circuit direction on GPIO 0 pin. |
|      | 1 = Pull-up |
|      | 0 = Pull-down |
| bit 3 | Reserved: Maintain as ‘0’ |
| bit 2 | PULLEN_GPIO2: Pull enable on GPIO 2 bit |
|      | This bit enables the weak-pull circuit in GPIO 2 pin. Note that when pin is configured to output, weak-pull circuit is automatically disabled. |
|      | 1 = Pull enabled |
|      | 0 = Pull disabled |
| bit 1 | PULLEN_GPIO1: Pull enable on GPIO 1 bit |
|      | This bit enables the weak-pull circuit in GPIO 1 pin. Note that when pin is configured to output, weak-pull circuit is automatically disabled. |
|      | 1 = Pull enabled |
|      | 0 = Pull disabled |
| bit 0 | PULLEN_GPIO0: Pull enable on GPIO 0 bit |
|      | This bit enables the weak-pull circuit in GPIO 0 pin. Note that when pin is configured to output, weak-pull circuit is automatically disabled. |
|      | 1 = Pull enabled |
|      | 0 = Pull disabled |

**Legend:**

- **R** = Readable bit
- **W** = Writable bit
- **U** = Unimplemented bit, read as ‘0’
- ‘-n’ = Value at POR
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- ‘x’ = Bit is unknown
- **r** = Reserved
### REGISTER 2-15: MACCON1 (MAC CONTROL 1 REGISTER)  
**ADDRESS:** 0x10

<table>
<thead>
<tr>
<th>Bit 7-6</th>
<th>R/W-00</th>
<th>R/W-001</th>
<th>R/W-1</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRXMODE&lt;1:0&gt;</td>
<td>ADDRSZ&lt;2:0&gt;</td>
<td>CRCSZ</td>
<td>FRMFMT</td>
<td>SECFLAGOVR</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit 0</th>
<th></th>
</tr>
</thead>
</table>

**Legend:**  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as ‘0’  
-n = Value at POR  
‘1’ = Bit is set  
‘0’ = Bit is cleared  
x = Bit is unknown  
r = Reserved

**bit 7-6**  
**TRXMODE<1:0>:** TX/RX Mode Select Field bits  
11 = Reserved  
10 = **TX-Streaming mode.** In this mode, use both buffers for packet transmission. When issuing TRXMODE = 10, RXEN is cleared. SPI addresses 0x200 to 0x27F access Buffer 1 or Buffer 2 in alternation. Access to 0x37F through 0x383 has non-defined effect.  
01 = **RX-Streaming mode.** In this mode, use both buffers for packet reception. When issuing TRXMODE = 01, TXST and TXENC/RXDEC bits are cleared and RXEN is set. SPI addresses 0x300 to 0x383 access Buffer 1 or Buffer 2 in alternation. In this mode, Proprietary mode packets other than streaming type are automatically discarded. Access to 0x200 through 0x283 has non-defined effect.  
00 = **Packet mode.** In this mode, Buffer 1 is used as a Transmit while Buffer 2 as a Receive packet buffer. SPI addresses from 0x200 to 0x27F access Buffer 1. SPI addresses 0x300 to 0x383 access Buffer 2. TRXMODE = 00 is mandatory when FRMFMT = 0.

**bit 5-3**  
**ADDRESZ<2:0>:** Source/Destination Address Size Field bits\(^{(1, 2)}\)  
The size of the Source and Destination addresses for Proprietary packet.  
Note that this field has no effect on the processing IEEE 802.15.4 frames.  
111 = 8 octets  
110 = 7 octets  
101 = 6 octets  
100 = 5 octets  
011 = 4 octets  
010 = 3 octets  
001 = 2 octets  
000 = 1 octet

**bit 2**  
**CRCSZ:** CRC Size bit  
This bit indicates the size of the CRC field in each packet  
1 = 2 octets  
0 = 0 octet

**bit 1**  
**FRMFMT:** MAC Frame Format bit adopted by the network\(^{(3)}\)  
This bit determines the frame format used in the network.  
1 = Proprietary  
0 = IEEE 802.15.4 standard compliant.

**Note 1:** Zero-length address occurs when the corresponding DAddrPrsnt/SAddrPrsnt bits of the packet Frame Control field are set to ‘0’.  
**Note 2:** Use ADDRSZ field while receiving and transmitting, and must not be modified while RXEN or TXST is set.  
**Note 3:** Use FRMFMT field while receiving and transmitting, and must not be modified while RXEN or TXST is set. In Debug mode, use this register bit to determine the frame format for both TX/RX frame in the packet buffers.
bit 0  **SECFLAGOVR**: Security Flag Override bit

The user can override security flags used in the CCM-CTR, CBC-MAC and CCM operation, otherwise the device uses the standard (2003/2006) definition.

**Note**: 1: Zero-length address occurs when the corresponding DAddrPrsnt/SAddrPrsnt bits of the packet Frame Control field are set to '0'.

2: Use ADDRSZ field while receiving and transmitting, and must not be modified while RXEN or TXST is set.

3: Use FRMFMT field while receiving and transmitting, and must not be modified while RXEN or TXST is set. In Debug mode, use this register bit to determine the frame format for both TX/RX frame in the packet buffers.
## REGISTER 2-16: MACCON2 (MAC CONTROL 2 REGISTER)

**ADDRESS:** 0x11

<table>
<thead>
<tr>
<th>Bit 7</th>
<th>CHANNEL&lt;3:0&gt;</th>
<th>Bit 0</th>
<th>SECSUITE&lt;3:0&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>R/W-0000</td>
<td>RW/HS-0000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Legend:
- **R** = Readable bit
- **W** = Writable bit
- **U** = Unimplemented bit, read as '0'
- **-n** = Value at POR
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- **x** = Bit is unknown
- **r** = Reserved

#### bit 7-4
**CHANNEL<3:0>:** TX/RX operating channel bits

These register bits are used as the current operating channel for TX/RX operation\(^1\).
- 0x0 = Channel 11
- 0x1 = Channel 12
- ...
- ...
- 0xF = Channel 26

#### bit 3-0
**SECSUITE<3:0>:** Security suite bits\(^2\)

- 1111 = AES-CBC-MAC-32 (Authentication with a 32-bit MAC, but no Encryption/Decryption)
- 1110 = AES-CBC-MAC-64 (Authentication with a 64-bit MAC, but no Encryption/Decryption)
- 1101 = AES-CBC-MAC-128 (Authentication with a 128-bit MAC, but no Encryption/Decryption)
- 1100 = Reserved
- 1011 = Reserved
- 1010 = Reserved
- 1001 = AES-CTR (Encryption/Decryption, but no Authentication)
- 1000 = AES-ECB (Encryption only)
- 0111 = AES-ENC-MIC-128 (Authentication with a 128-bit MAC and Encryption/Decryption)
- 0110 = AES-ENC-MIC-64 (Authentication with a 64-bit MAC and Encryption/Decryption)
- 0101 = AES-ENC-MIC-32 (Authentication with a 32-bit MAC and Encryption/Decryption)
- 0100 = AES-ENC (Encryption/Decryption, but no Authentication)
- 0011 = AES-MIC-128 (Authentication with a 128-bit MAC, but no Encryption/Decryption)
- 0010 = AES-MIC-64 (Authentication with a 64-bit MAC, but no Encryption/Decryption)
- 0001 = AES-MIC-32 (Authentication with a 32-bit MAC, but no Encryption/Decryption)
- 0000 = No security services enabled, or security is handled by upper protocol layers; ignore the setting of the SecEn bit (if value is ‘0’)

### Note 1:
Use this field while receiving and transmitting, and must not be modified while RXEN or TXST is set.

### Note 2:
In 15.4-2006 Standard mode MAC layer security processing, the Register field is automatically set based on the SecLvl bits of the AuxSecHdr Control field.
REGISTER 2-17: TXCON (TRANSMIT CONTROL REGISTER)  ADDRESS: 0x12

<table>
<thead>
<tr>
<th>RW/HC-0</th>
<th>RW-0</th>
<th>RW/HC-0</th>
<th>R/HS/HC-1</th>
<th>R/W-1</th>
<th>R/W-011</th>
</tr>
</thead>
<tbody>
<tr>
<td>TXST</td>
<td>DTSM</td>
<td>TXENC</td>
<td>TXBUFEMPTY</td>
<td>CSMAEN</td>
<td>DR&lt;2:0&gt;</td>
</tr>
</tbody>
</table>

Legend:  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as ‘0’  
-n = Value at POR  
‘1’ = Bit is set  
‘0’ = Bit is cleared  
x = Bit is unknown  
r = Reserved  
HC = Hardware Clear  
HS = Hardware Set

bit 7  
TXST: Transmit Start bit  
1 = Starts the transmission of the next TX packet\(^{(1, 2)}\)  
0 = Termination of current TX operation, which may result in the transmission of an incomplete packet  

Hardware Clear:  
- Once the packet is successfully transmitted (including all attempted retransmissions, if any), the hardware clears this bit and sets the TXIF and IDLEIF.  
- If the packet transmission fails due to a CSMA failure, this bit is cleared, and TXCSMAIF is set.  
- If Acknowledge is requested (AckReq bit field in the transmitted frame is set) and not received after the configured number of retransmissions (TXRETCNT), then TXST bit is cleared, and TXACKIF is set.  
- In TX-Streaming mode (TRXMODE), TXST is set even when it is already set, resulting in a posted start. When the current TX operation completes, the posted start immediately starts afterward. Clearing of the TXST bit clears both the current and the posted (pending) TX starts. TXOVFIF is set when TXST = 1, a posted start is present and a Host Controller write to the packet buffer occurs. Outside of TX-Streaming mode, writes to TXST when TXST is already set is ignored.  

Clearing this bit aborts the current operation in these cases:  
- When transmitting a packet in Packet mode or in TX-Streaming mode  
- When waiting for an ACK packet after a transmission  
- During the CSMA CA algorithm  
- When transmitting a repeated frame  

This field is read at any time to determine if TX operation is in progress.

bit 6  
DTSM: Do Not Touch Security Materials bit\(^{(2)}\)  
1 = Device does not change the security material configured by the host MCU  
0 = Device tries to configure the security material related registers  

MCU must fill the following registers: SECNONCE, SECHDRINDEX, SECPAYINDEX and SECENDINDEX.

bit 5  
TXENC: TX Encryption  
Setting this bit starts the TX security processing (authentication or encryption, or both) of the packet in the buffer it was last written. TXENC is cleared and TXENCIF is set when the processing is complete. TXENC must be issued when NWK layer security needs to be processed. 802.15.4-2003/2006 MAC layer security operation is automatically performed when TXST bit is set. Note that this field must not be modified while TXST is set.

Note 1:  
Transmission may include automatic security processing, CRC appending, CSMA-CA channel access, Acknowledge reception and retransmissions depending on the register Configuration and the Frame Control field of the frame to be transmitted.

2: DTSM has no relevance in reception as the host can always reconfigure the security material before setting RXDEC.
bit 4  TXBUFEMPTY: TX Buffer Empty bit

TXBUFEMPTY = 1 indicates that the Host MCU can safely start writing a new frame to the buffer without overwriting any content that is in use. Writing a single byte to the buffer clears this bit. TXBUFEMPTY = 0 does not prevent the host from writing further bytes to the buffer. TXBUFEMPTY is set by the device when transmission is complete.

1 = MCU can safely start writing a new frame to the buffer
0 = Buffer is full, or being written to

When TRXMODE = 00:
Packet mode is configured then TXBUFEMPTY is set at the same time as TXST is cleared and an interrupt is generated. Therefore, this bit provides no extra information.

When TRXMODE = 10:
TX-Streaming mode is configured then TXBUFEMPTY is set at the same time as one of the buffers becomes free, while TXST may be set. Therefore, the host MCU uses TXBUFEMPTY to ensure that the next frame starts loading to the buffers, without overwriting a packet being sent (TXOVFIF).

bit 3  CSMAEN: CSMA-CA Enable bit

This bit enables CSMA-CA algorithm before transmission.

1 = CSMA-CA enabled
0 = CSMA-CA disabled

bit 2-0  DR<2:0>: Transmit Data Rate Field bits

111 = Reserved
110 = 2 Mbps
101 = 1 Mbps
100 = 500 kbps
011 = 250 kbps
010 = 125 kbps
001 = Reserved
000 = Reserved

When transmitting an Auto-ACK frame with Adaptive Data Rate in response to a received frame, the AckDataRate field in the received frame automatically determines the data rate of the PHY, and not by this Register field. In all other cases, use this Register field as the current PHY data rate when transmitting.

The PHY automatically determines the data rate for all received frames regardless of this Register field and the Adaptive Data Rate Configuration. For more information, refer to Register 2-43.

Note 1: Transmission may include automatic security processing, CRC appending, CSMA-CA channel access, Acknowledge reception and retransmissions depending on the register Configuration and the Frame Control field of the frame to be transmitted.

2: DTSM has no relevance in reception as the host can always reconfigure the security material before setting RXDEC.
REGISTER 2-18:  RXACKWAIT (RX ACKNOWLEDGE WAIT REGISTER)  
ADDRESS: 0x13

<table>
<thead>
<tr>
<th>RW-0</th>
<th>RW-1</th>
<th>RW/HC-1</th>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>RXACKWAIT&lt;7:0&gt;</th>
</tr>
</thead>
</table>

Legend:  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as ‘0’  
-n = Value at POR  
‘1’ = Bit is set  
‘0’ = Bit is cleared  
x = Bit is unknown  
r = Reserved  
HC = Hardware Clear

bit 7-0  
RXACKWAIT<7:0>: Auto Acknowledge Wait Field bits  
This field indicates the number of Base time units that the device must Wait after receiving a packet with AckReq = 1, before transmitting the corresponding ACK packet. This field is only used when AUTO-ACKEN = 1. For more information on Base time units, see Section 4.1 “MAC Architecture”.

REGISTER 2-19:  RETXCOUNT (RETRANSMISSION COUNT REGISTER)  
ADDRESS: 0x14

<table>
<thead>
<tr>
<th>RW-0011</th>
<th>R-0000</th>
</tr>
</thead>
<tbody>
<tr>
<td>RETXMCNT&lt;3:0&gt;</td>
<td>RETXCCNT&lt;3:0&gt;</td>
</tr>
</tbody>
</table>

Legend:  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as ‘0’  
-n = Value at POR  
‘1’ = Bit is set  
‘0’ = Bit is cleared  
x = Bit is unknown  
r = Reserved

bit 7-4  
RETXMCNT<3:0>: Retransmission Max Count Field bits

The maximum number of retries allowed after a transmission failure.

1111 = 15 retries  
0101 = 5 retries  
0001 = 1 retry

bit 0  
0000 = Transmitter does not Wait for ACK

bit 3-0  
RETXCCNT<3:0>: Retransmission Current Count Field bits  
This read-only field indicates the current retransmit attempt number. When RETXCCNT<3:0> = RETX-MCNT<3:0> and the TX attempt fails, the transmission is aborted, generating TXACKIF interrupt.

Note 1:  
Use this field during transmission, and must be unmodified while TXST is set.
REGISTER 2-20:  RXCON1 (MAC RECEIVE CONTROL 1 REGISTER)  

ADDRESS: 0x15

<table>
<thead>
<tr>
<th>RW/HC/HS-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>R/W-0</th>
<th>R-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>RXEN</td>
<td>NOPA</td>
<td>RXDEC</td>
<td>RSVLQIEN</td>
<td>RSVRSSIEN</td>
<td>RSVCHDREN</td>
<td>RSVCFIEN</td>
<td>r</td>
<td></td>
</tr>
</tbody>
</table>

Legend:  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as ‘0’  
-n = Value at POR  
‘1’ = Bit is set  
‘0’ = Bit is cleared  
x = Bit is unknown  
r = Reserved  
HC = Hardware Clear  
HS = Hardware Set

| bit 7 | RXEN: Receive Enable Field bit  
This bit enables/disables the packet reception. If an RX packet is currently being received, clearing this bit causes that packet to be discarded.  
1 = RX enabled  
0 = RX disabled  
Hardware clear/set when:  
• Cleared when TRXMODE is set to TX-Streaming mode  
• Set when TRXMODE is set to RX-Streaming mode  
Clearing this bit aborts the current operation in the following cases:  
• Receiving a packet in Packet mode or in RX-Streaming mode  
Changes to most RX related settings must be only done when this bit is cleared.  
Note that the clear channel assessment (CSMAEN) and ACK-frame reception does not require RXEN = 1, as the device turns the radio into RX when needed, irrespective of the status of the RXEN bit.  

| bit 6 | NOPA: No Parsing bit  
This bit disables packet parsing. Only CRC is checked, if it is enabled. This feature is useful in Sniffer mode.  
1 = Disable packet parsing  
0 = Enable packet parsing

| bit 5 | RXDEC: RX Decryption bit  
Setting this bit starts the RX security processing (authentication or decryption, or both) on the last received packet.  
1 = RX security processing started/in process. RXDECIF or RXTAGIF is set.  
0 = RX security processing inactive or complete  
This bit clears itself after RX decryption is completed.

| bit 4 | RSVLQIEN: Receive Status Vector LQI Enable bit  
If bit is set, the measured Link Quality is appended after the received frame in the packet buffer.  
1 = Append LQI field  
0 = Do not append LQI field

| bit 3 | RSVRSSIEN: Receive Status Vector RSSI Enable bit  
If bit is set, the measured RSSI is appended after the received frame in the packet buffer.  
1 = Append RSSI field  
0 = Do not append RSSI field

| bit 2 | RSVCHDREN: Receive Status Vector Channel/MAC Type/Data Rate Enable bit  
If bit is set, Channel, MAC type and Data Rate Configurations used with the received frame are appended after the received frame in the packet buffer, using the encoding specified for CH<3:0>, FRMFMT and DR<2:0> (concatenated in this order when most significant bit (MSb) is first).  
1 = Append Channel, MAC type and Data Rate fields  
0 = Do not append Channel, MAC type and Data Rate fields
REGISTER 2-20: RXCON1 (MAC RECEIVE CONTROL 1 REGISTER) (CONTINUED)

ADDRESS: 0x15

bit 1  **RSVCFOEN**: Receive Status Vector CFO Enable bit
If bit is set, the estimated Carrier Frequency Offset of the received frame is appended after the received frame in the packet buffer, using the same encoding as CFOMEAS register.

1 = Append CFO estimation
0 = Do not append estimated CFO

bit 0  **Reserved**: Maintain as '0'

REGISTER 2-21: RXCON2 (MAC RECEIVE CONTROL 2 REGISTER)

ADDRESS: 0x16

<table>
<thead>
<tr>
<th>R/C/HS-0</th>
<th>RW-0</th>
<th>R-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>R/W-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>RXBUFFUL</td>
<td>IDENTREJ</td>
<td>ACKRXFP</td>
<td>ACKTXFP</td>
<td>AUTORPTEN</td>
<td>AUTOACKEN</td>
<td>ADPTCHEN</td>
<td>ADPTDREN</td>
<td></td>
</tr>
</tbody>
</table>

Legend:  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'

- n = Value at POR  
'1' = Bit is set  
'0' = Bit is cleared  
x = Bit is unknown

r = Reserved

bit 7  **RXBUFFUL**: RX Buffer Full bit
Host MCU clears this bit to indicate that the RX packet is processed. If this bit is uncleared before the next valid RX packet is detected (packet is not a duplicate, pass RX filter, and so on), then the device sets RXOVFIF and the buffer content is unmodified, where RXBUFFUL = 1 locks write access by a new frame.
Moreover, the host can both read and write to the buffer or perform security processing.

In TRXMODE = 00 (PACKET) mode:

1 = Receive buffer content is yet to be read by the host or processed, and cannot be overwritten by a new frame
0 = Receive buffer is free for receiving a new frame

In TRXMODE = 01 (RX-STREAMING) mode:

1 = Current buffer being read from the bus contains a valid RX Packet
0 = Current buffer being read from the bus is empty

bit 6  **IDENTREJ**: Reject Identical Packet bit
Setting this bit enables the user to reject an incoming packet, in case its source address and sequence number is the same as the previously received packet.
This bit is used whenever a packet is received and ACK is transmitted, but the ACK is never received that the sender resends the TX packet. When this happens, triggers for RXIF is avoided for the second time for the same packet, thus, the second packet is ignored.
This bit is also used when a packet is repeated and the next repeater repeats the same packet back. This packet is received, but ignored.

1 = Any packet received with the same source address and sequence number as the last packet successfully received is discarded and RXIDNTIF is thrown
0 = Duplicated packets are processed further same as non-duplicated packets

**Note 1:** Use ADPTCHEN field while receiving and transmitting a packet, and must be unmodified while RXEN or TXST is set.

**2:** Use ADPTDREN field while receiving and transmitting a packet, and must be unmodified while RXEN or TXST is set.
REGISTER 2-21: RXCON2 (MAC RECEIVE CONTROL 2 REGISTER) (CONTINUED)  
ADDRESS: 0x16

bit 5  ACKRXFP: ACK RX Frame Pending bit  
This read-only Status bit reflects the value of the FrameCtrl (FramePend) bit in the last received 802.15.4 compatible ACK frame.

bit 4  ACKTXFP: ACK TX Frame Pending bit  
The value of this bit is transmitted in the FrameCtrl (FramePend) bit slot when the MAC sends out an ACK packet in 802.15.4 Compatibility mode.

bit 3  AUTORPTEN: Auto-Repeat Enable bit  
If this bit is set, the MAC automatically transmits a packet whenever a packet is received, and its Repeat bit is set.

1 = Auto-Repeat feature is enabled  
0 = Auto-Repeat feature is disabled

bit 2  AUTOACKEN: Auto-Acknowledge Enable bit  
If this bit is set, then the device automatically transmits an ACK packet whenever a packet is received, and its AckReq bit is set.

1 = Automatic Acknowledge processing enabled  
0 = Automatic Acknowledge processing disabled

bit 1  ADPTCHEN: Adaptive Channel Enable bit(1)  
Setting this bit enables the MAC in Proprietary mode to set the transmitting channel for the ACK packet based on the AckInfo field (Proprietary packet) of the received packet, rather than the CH<3:0> register bits.

1 = Adaptive Channel feature is enabled  
0 = Adaptive Channel feature is disabled

This feature is also known as Channel Agility. For more information, see Section 7.1 “Channel Agility”.

bit 0  ADPTDREN: Adaptive Data Rate Enable bit(2)  
Setting this bit enables the MAC in Proprietary mode to set the transmission data rate for the ACK packet based on the AckInfo field (Proprietary packet) of the received packet, rather than the DR<2:0> register bits.

1 = Adaptive Data Rate feature is enabled  
0 = Adaptive Data Rate feature is disabled

This feature is also known as Channel Agility. For more information, see Section 7.1 “Channel Agility”.

**Note:**  
1: Use ADPTCHEN field while receiving and transmitting a packet, and must be unmodified while RXEN or TXST is set.  
2: Use ADPTDREN field while receiving and transmitting a packet, and must be unmodified while RXEN or TXST is set.
## REGISTER 2-22: TXACKTO (TX ACKNOWLEDGE TIME-OUT REGISTER)  
ADDRESS: 0x17

<table>
<thead>
<tr>
<th>bit 7-0 TXACKTO&lt;7:0&gt;: TX Acknowledge Time-Out Field bits[^1]</th>
</tr>
</thead>
<tbody>
<tr>
<td>The maximum time in Base time units that the device must Wait for receiving an ACK packet.</td>
</tr>
<tr>
<td>0x00 = Wait 1 Base time unit before retransmitting, implying that the device continually retransmits RETXMCNT&lt;3:0&gt; times. For more information on Base time units, see Section 4.1 “MAC Architecture”.</td>
</tr>
<tr>
<td>0x01 = Wait 1 Base time unit before retransmitting</td>
</tr>
<tr>
<td>0x7F = Wait 127 Base time units before retransmitting</td>
</tr>
</tbody>
</table>

**Note 1:** Use TXACKTO field during transmission, and it must be unmodified while TXST is set.

[^1]: The value of each bit affects the maximum wait time before retransmitting the data packet.
REGISTER 2-23: RXFILTER (RX FILTER REGISTER) 
ADDRESS: 0x18

<table>
<thead>
<tr>
<th>R/W-0</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>PANCRDN</td>
<td>CRCREJ</td>
<td>CMDREJ</td>
<td>DATAREJ</td>
<td>UNIREJ</td>
<td>NOTMEREJ</td>
<td>BCREJ</td>
<td>NSTDREJ</td>
</tr>
</tbody>
</table>

Legend:  
R = Readable bit  W = Writable bit  U = Unimplemented bit, read as '0'  
-n = Value at POR  ‘1’ = Bit is set  ‘0’ = Bit is cleared  x = Bit is unknown  
r = Reserved

bit 7  
PANCRDN: PAN Coordinator bit

Setting this bit enables the node to accept DAMode = 00 type packets if it is a CMD or DATA frame.

1 = Disable rejection  
0 = Reject all DATA and CMD packets when DAMode = 00

bit 6  
CRCREJ: CRC Error Reject Enable bit

Setting this bit enables the user to reject all packets that contains an invalid CRC, provided that it is present (CRCSZ = 1). Clearing this bit enables the user to accept all packets that contains an invalid CRC, provided that it is present (CRCSZ = 1), skipping any further filtering. When CRC is not present then this bit has no effect (CRCSZ = 0).

1 = Reject all packets having an invalid CRC  
0 = Accept all packets having an invalid CRC without further filtering

bit 5  
CMDREJ: Command Frame Reject Enable bit

Setting this bit enables the user to reject all packets with FrameCtrl (Type) equal to Command.

1 = Reject all Command packets  
0 = Disable Command Frame Rejection

bit 4  
DATAREJ: Data Frame Reject Enable bit

Setting this bit enables the user to reject all packets with FrameCtrl (Type) equal to Data.

1 = Reject all Data packets  
0 = Disable Data Frame Rejection

bit 3  
UNIREJ: Unicast Reject Enable bit

Setting this bit enables the user to reject all unicast packets as in:

802.15.4 Mode: PAN Identifier matches with the PANID<15:0> or 0xFFFF, and Destination Address matches the address in the ADDR<63:0> or SHADDR<15:0> register, which the DAMode selects.

Proprietary Mode: Destination Address matches the address in ADDR<ADDRSZ<2:0>*8-1:0> register, provided that DAddrPrsnt Frame Control field is set.

1 = Reject all Unicast packets addressed to this node  
0 = Disable Unicast Rejection

Note 1: In Proprietary mode (FRMFMT = 1), when CRCREJ = 1 is used to reject unicast frames not addressed to this node. NOTMEREJ = 1 does not reject these frames.

2: UNIREJ does not affect the frames using implied destination addressing in 802.15.4 mode and inferred destination addressing in Proprietary mode.

3: NOTMEREJ does not affect the frames using implied destination addressing in 802.15.4 mode and inferred destination addressing in Proprietary mode.

4: NSTDREJ does not affect the Proprietary frames in Proprietary mode.
REGISTER 2-23: RXFILTER (RX FILTER REGISTER) (CONTINUED)  ADDRESS: 0x18

bit 2  NOTMEREJ: Not Me Unicast Reject Enable bit(3)

Setting this bit enables the user to reject all unicast packets as in:

802.15.4 Mode: Destination PAN Identifier does not match PANID<15:0> and is not 0xFFFF (broadcast) or Destination Address does not match the address in the ADDR<63:0> register or the SHADDR<15:0> register, which the DAMode selects.

Proprietary Mode: Destination Address matches the address in ADDR<ADDRSZ<2:0>*8-1:0> register, provided that DAddrPrsnt Frame Control field is set(1).

1 = Reject all Unicast packets NOT addressed to this node
0 = Disable Not Me Unicast Rejection Filtering

bit 1  BCREJ: Broadcast Rejection bit

802.15.4 Mode: Setting this bit enables the user to reject all Broadcast packets of type Data or Command. A Data or Command packet is broadcast when Short Destination Addressing is used (DAMode = 10) and Short Address is equal 0xFFFF.

Proprietary Mode: Setting this bit enables the user to reject all Broadcast packets of type Data or Command (or Streaming). A packet is broadcast when FrameCtrl[Broadcast] is set.

1 = Reject Broadcast Packets
0 = Disable Broadcast Rejection

bit 0  NSTDREJ: Non-Standard Frame Reject bit(4)

This bit enables the user to reject all 802.15.4 frames having 01 for the DAMode or SAMode fields or having the MSb (bit 2) in the Type field set (1) or having the MSb (bit 1) in the Frame Version field set to(1).

1 = Reject all Non-Standard 802.15.4 packets
0 = Disable Non-Standard Rejection

Note 1: In Proprietary mode (FRMFMT = 1), when CRCREQ = 1 is used to reject unicast frames not addressed to this node. NOTMEREJ = 1 does not reject these frames.

2: UNIREJ does not affect the frames using implied destination addressing in 802.15.4 mode and inferred destination addressing in Proprietary mode.

3: NOTMEREJ does not affect the frames using implied destination addressing in 802.15.4 mode and inferred destination addressing in Proprietary mode.

4: NSTDREJ does not affect the Proprietary frames in Proprietary mode.
REGISTER 2-24: TMRCON (TIMER CONTROL REGISTER)  
ADDRESS: 0x19

<table>
<thead>
<tr>
<th>RW-100</th>
<th>RW-00010</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOMCNT&lt;2:0&gt;</td>
<td>BASETM&lt;4:0&gt;</td>
</tr>
</tbody>
</table>

Legend:  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
-n = Value at POR  
'1' = Bit is set  
'0' = Bit is cleared  
x = Bit is unknown  
r = Reserved

bit 7-5  
**BOMCNT<2:0>:** CSMA-CA Back-off Maximum Count bits  
The maximum number of back-off attempts the CSMA-CA algorithm attempts before declaring a channel access failure.  
111 = Reserved  
110 = Reserved  
101 = 5 attempts  
100 = 4 attempts  
011 = 3 attempts  
010 = 2 attempts  
001 = 1 attempt  
000 = 0 attempt

bit 4-0  
**BASETM<4:0>:** Base Time Field bits  
The number of 1 µs clock cycles that a Base time unit represents in all register settings. For more information on Base time units, see Section 4.1 “MAC Architecture”.
# MRF24XA

## REGISTER 2-25: CSMABE (CSMA-CA BACK-OFF EXPONENT CONTROL REGISTER)

### ADDRESS: 0x1A

<table>
<thead>
<tr>
<th>RW-0101</th>
<th>RW-0011</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAXBE&lt;3:0&gt;</td>
<td>MINBE&lt;3:0&gt;</td>
</tr>
<tr>
<td>bit 7</td>
<td>bit 0</td>
</tr>
</tbody>
</table>

**Legend:**
- **R** = Readable bit
- **W** = Writable bit
- **U** = Unimplemented bit, read as '0'
- **-n** = Value at POR
- **’1’** = Bit is set
- **’0’** = Bit is cleared
- **x** = Bit is unknown
- **r** = Reserved

### bit 7-4  MAXBE<3:0>: CSMA-CA Back-off Maximum Count Field bits

The maximum value of the Back-off exponent (BE), in the CSMA-CA algorithm. The back-off time is \((2^{BE} - 1)\) units.

- **1111** = Reserved
- **1001** = Reserved
- **1000** = \(2^{8-1} = 255\) maximum units of back-off time
- **0000** = \(2^{0-1} = 0\) back-off time

### bit 3-0  MINBE<3:0>: CSMA-CA Back-off Minimum Count bits

The minimum value of the back-off exponent (BE), in the CSMA-CA algorithm. The back-off time is \((2^{BE} - 1)\) units.

- **1111** = Reserved
- **1001** = Reserved
- **1000** = \(2^{8-1} = 255\) maximum units of back-off time
- **0000** = \(2^{0-1} = 0\) back-off time
**REGISTER 2-26: BOUNIT (BACK-OFF TIME UNIT REGISTER) ADDRESS: 0x1B**

<table>
<thead>
<tr>
<th>Bit</th>
<th>R/W-10100000</th>
<th>BOUNIT&lt;7:0&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 7</td>
<td></td>
<td>bit 0</td>
</tr>
</tbody>
</table>

**Legend:**  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as ‘0’  
-n = Value at POR  
‘1’ = Bit is set  
‘0’ = Bit is cleared  
x = Bit is unknown  
r = Reserved

- **BOUNIT<7:0>: CSMA-CA Back-off Period Unit Field bits**
  - The number of Base time units for the basic back-off time unit used by CSMA-CA algorithm.
  - 11111111 = 256 Base time units
  - 00000000 = 1 Base time unit

**REGISTER 2-27: STRMTOH/STRMTOL (STREAM TIME-OUT REGISTER) ADDRESS: 0x1C – 0x1D**

<table>
<thead>
<tr>
<th>Bit</th>
<th>R/W-11111111</th>
<th>STRMTO&lt;15:8&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 15</td>
<td></td>
<td>bit 8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit</th>
<th>R/W-11111111</th>
<th>STRMTO&lt;7:0&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 7</td>
<td></td>
<td>bit 0</td>
</tr>
</tbody>
</table>

**Legend:**  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as ‘0’  
-n = Value at POR  
‘1’ = Bit is set  
‘0’ = Bit is cleared  
x = Bit is unknown  
r = Reserved

- **STRMTO<15:0>: Stream Time-Out bits**
  - The STRMTO<15:0> bits indicate the maximum number of allowed Base time units between the end of one RX Stream packet and the successful reception of the next. If no RX Stream packet is successfully received within this time, STRMIF is set.
### REGISTER 2-28: OFFTM (OFF-TIMER REGISTER)  
**ADDRESS:** 0x1E

<table>
<thead>
<tr>
<th>Bit 7-0</th>
<th>OFFTM&lt;7:0&gt;</th>
<th>OFF-Timer Field bits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>This value sets the minimum PLL OFF time in 1 µs resolution.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimum OFF Time = OFFTM&lt;7:0&gt; * 32</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If this register is set to 0xFF, PLL remains off.</td>
</tr>
</tbody>
</table>

**Legend:**  
- **R** = Readable bit  
- **W** = Writable bit  
- **U** = Unimplemented bit, read as ‘0’  
- ‘-n’ = Value at POR  
- ‘1’ = Bit is set  
- ‘0’ = Bit is cleared  
- ‘x’ = Bit is unknown  
- **r** = Reserved
## REGISTER 2-29: ADDR (ADDRESS REGISTER)

<table>
<thead>
<tr>
<th>Address</th>
<th>Description</th>
<th>Access</th>
<th>Bits</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x1F-0x26</td>
<td>ADDR&lt;63:56&gt;</td>
<td>R/W</td>
<td>63-56</td>
<td>bit 63-0: Long Address Field bits. Current device’s long address (LSB stored). For Proprietary frames, the number of address bytes is defined in ADDRSZ&lt;2:0&gt;. For addresses less than 8 octets, use the least significant bits of this register.</td>
</tr>
<tr>
<td>0x1F-0x26</td>
<td>ADDR&lt;55:48&gt;</td>
<td>R/W</td>
<td>55-48</td>
<td></td>
</tr>
<tr>
<td>0x1F-0x26</td>
<td>ADDR&lt;47:40&gt;</td>
<td>R/W</td>
<td>47-40</td>
<td></td>
</tr>
<tr>
<td>0x1F-0x26</td>
<td>ADDR&lt;39:32&gt;</td>
<td>R/W</td>
<td>39-32</td>
<td></td>
</tr>
<tr>
<td>0x1F-0x26</td>
<td>ADDR&lt;31:24&gt;</td>
<td>R/W</td>
<td>31-24</td>
<td></td>
</tr>
<tr>
<td>0x1F-0x26</td>
<td>ADDR&lt;23:16&gt;</td>
<td>R/W</td>
<td>23-16</td>
<td></td>
</tr>
<tr>
<td>0x1F-0x26</td>
<td>ADDR&lt;15:8&gt;</td>
<td>R/W</td>
<td>15-8</td>
<td></td>
</tr>
<tr>
<td>0x1F-0x26</td>
<td>ADDR&lt;7:0&gt;</td>
<td>R/W</td>
<td>7-0</td>
<td></td>
</tr>
</tbody>
</table>

Legend:  
- R = Readable bit  
- W = Writable bit  
- U = Unimplemented bit, read as ‘0’  
- -n = Value at POR  
- ‘1’ = Bit is set  
- ‘0’ = Bit is cleared  
- x = Bit is unknown  
- r = Reserved
### REGISTER 2-30: SHADDR/SHADDRL (SHORT ADDRESS REGISTER)  
**ADDRESS:** 0x27 – 0x28

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Value at POR</th>
<th>Bit Set</th>
<th>Bit Cleared</th>
<th>Bit Unknown</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>SHADDR&lt;15:8&gt;: Short Address Field bits</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>SHADDR&lt;7:0&gt;: Short Address Field bits</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Legend:
- **R** = Readable bit
- **W** = Writable bit
- **U** = Unimplemented bit, read as ‘0’
- ‘-n’ = Value at POR
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- ‘x’ = Bit is unknown

- **r** = Reserved

Current device’s short address (LSB stored). Only used in 802.15.4 mode.

### REGISTER 2-31: PANIDH/PANIDL (PAN IDENTIFIER REGISTER)  
**ADDRESS:** 0x29 – 0x2A

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Value at POR</th>
<th>Bit Set</th>
<th>Bit Cleared</th>
<th>Bit Unknown</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>PANID&lt;15:8&gt;: PAN Identifier Field bits</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>PANID&lt;7:0&gt;: PAN Identifier Field bits</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Legend:
- **R** = Readable bit
- **W** = Writable bit
- **U** = Unimplemented bit, read as ‘0’
- ‘-n’ = Value at POR
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- ‘x’ = Bit is unknown

- **r** = Reserved

Current device’s PAN Identifier (LSB stored). Only used in 802.15.4 mode.
REGISTER 2-32:  SECHDRINDEX (SECURITY HEADER INDEX REGISTER)

ADDRESS: 0x2B

<table>
<thead>
<tr>
<th>bit 7</th>
<th>Remaining bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reserved: Maintain as '0'</td>
<td></td>
</tr>
</tbody>
</table>

Legend:

- **R** = Readable bit
- **W** = Writable bit
- **U** = Unimplemented bit, read as '0'
- -n = Value at POR
- ’1’ = Bit is set
- ’0’ = Bit is cleared
- x = Bit is unknown
- r = Reserved
- HS = Hardware Set

bit 7

**Reserved:** Maintain as ‘0’

bit 6-0

**SECHDRINDEX<6:0>:** Security Header Index Field bits

This field defines the portion of the header which performs the authentication operations.

For MAC layer security, SECHDRINDEX<6:0> is defined as the address offset of the MAC header from the beginning of the frame, as stored in the buffer (0 = Length field, 1 = FrameCtrl field, and so on), and is automatically loaded for both 802.15.4 and Proprietary frames.

For Network layer security, SECHDRINDEX<6:0> is defined as the address offset of the Network Header from the beginning of the frame and the Host Controller loads it only for 802.15.4 frames. Note that for Proprietary frames, the MAC automatically loads it.

**Note 1:** Setting the DTSM bit disables the automatic computation of this field in TX mode.

REGISTER 2-33:  SECPAYINDEX (SECURITY PAYLOAD INDEX REGISTER)

ADDRESS: 0x2C

<table>
<thead>
<tr>
<th>bit 7</th>
<th>Remaining bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reserved: Maintain as '0'</td>
<td></td>
</tr>
</tbody>
</table>

Legend:

- **R** = Readable bit
- **W** = Writable bit
- **U** = Unimplemented bit, read as '0'
- -n = Value at POR
- ’1’ = Bit is set
- ’0’ = Bit is cleared
- x = Bit is unknown
- r = Reserved
- HS = Hardware Set

bit 7

**Reserved:** Maintain as ‘0’

bit 6-0

**SECPAYINDEX<6:0>:** Security Payload Index Field bits

This field defines the portion of the payload, which the encryption/decryption operations are performed.

For MAC layer security, SECPAYINDEX<6:0> is defined as the address offset of the MAC payload from the beginning of the frame, as stored in the buffer (0 = Length field, 1 = FrameCtrl field, and so on), and is automatically loaded for both 802.15.4 and Proprietary frames.

For Network layer security, SECPAYINDEX<6:0> is defined as the address offset of the payload from the beginning of the frame and the Host Controller loads it only for 802.15.4 frames. Note that for Proprietary frames, the MAC automatically loads it.

**Note 1:** Setting the DTSM bit disables the automatic computation of this field in TX mode.
REGISTER 2-34: SECENDINDEX (SECURITY END INDEX REGISTER)  ADDRESS: 0x2D

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>SECENDINDEX&lt;6:0&gt;</td>
</tr>
</tbody>
</table>

Legend:  
R = Readable bit  W = Writable bit  U = Unimplemented bit, read as ‘0’  
-n = Value at POR  ‘1’ = Bit is set  ‘0’ = Bit is cleared  x = Bit is unknown  
r = Reserved  HS = Hardware Set

**bit 7**  
Reserved: Maintain as ‘0’

**bit 6-0**  
SECENDINDEX<6:0>: Security End Index Field bits

This field defines the end of the payload, which the security operations are performed.

**Note 1:** Setting the DTSM bit disables the automatic computation of this field in TX mode.
REGISTER 2-35:  MACDEBG (MAC DEBUG CONTROL REGISTER)  ADDRESS: 0x2E

<table>
<thead>
<tr>
<th>R/W/HC-0</th>
<th>R/W/HC-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/HS/HC-0</th>
<th>R/HS/HC-0</th>
<th>R/HS/HC-0</th>
<th>R/HS/HC-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUF1TXPP</td>
<td>BUF2TXPP</td>
<td>BUF1RXPP</td>
<td>BUF2RXPP</td>
<td>TXRDBUF</td>
<td>RXWRBUF</td>
<td>BUSRDBUF</td>
<td>BUSWRBUF</td>
</tr>
</tbody>
</table>

Legend:
- R = Readable bit
- W = Writable bit
- U = Unimplemented bit, read as ‘0’
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- x = Bit is unknown
- r = Reserved
- HC = Hardware Clear
- HS = Hardware Set

bit 7  BUF1TXPP: Buffer 1 TX Process Packet bit
Setting this bit performs all of the processing (CRC generation and security) on BUF1 that is normally done before transmitting a packet, but without actually transmitting the packet.

bit 6  BUF2TXPP: Buffer 2 TX Process Packet bit
Setting this bit performs all of the processing (CRC generation and security) on BUF2 that is normally done before transmitting a packet, but without actually transmitting the packet.

bit 5  BUF1RXPP: Buffer 1 RX Process Packet bit
Setting this bit performs all of the processing (CRC checking and security) on BUF1 that is normally done when receiving a packet, but without actually receiving the packet.
This bit must be asserted while downloading security materials and so on during debug.

bit 4  BUF2RXPP: Buffer 2 RX Process Packet bit
Setting this bit performs all of the processing (CRC checking and security) on BUF2 that is normally done when receiving a packet, but without actually receiving the packet.
This bit must be asserted while downloading security materials and so on, during debug.

bit 3  TXRDBUF: TX Read Buffer Flag bit
Indicates the physical buffer number (0 = BUF1, 1 = BUF2) that the TX hardware is reading.

bit 2  RXWRBUF: RX Write Buffer Flag bit
Indicates the physical buffer number (0 = BUF1, 1 = BUF2) that the RX hardware is writing to.

bit 1  BUSRDBUF: Bus Read Buffer Flag bit
Indicates the physical buffer number (0 = BUF1, 1 = BUF2) that the SFR bus is reading. This bit is only used in RX-Streaming mode.

bit 0  BUSWRBUF: Bus Write Buffer Flag bit
Indicates the physical buffer number (0 = BUF1, 1 = BUF2) that the SFR bus is writing to. This bit is only used in TX-Streaming mode.
### REGISTER 2-36: CCACON1 (CCA CONTROL 1 REGISTER)  
**ADDRESS:** 0x2F

<table>
<thead>
<tr>
<th>R/H/S/HC-0</th>
<th>R/W/HC-0</th>
<th>R/W-001100</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCABUSY</td>
<td>R/W/HC-0</td>
<td>R/W-001100</td>
</tr>
<tr>
<td>CCAST</td>
<td>RSSITHR&lt;5:0&gt;</td>
<td></td>
</tr>
</tbody>
</table>

**Legend:**  
- **R** = Readable bit  
- **W** = Writable bit  
- **U** = Unimplemented bit, read as ‘0’  
- **-n** = Value at POR  
- ‘1’ = Bit is set  
- ‘0’ = Bit is cleared  
- **x** = Bit is unknown  
- **r** = Reserved  

**bit 7**  
**CCABUSY:** Clear Channel Assessment Busy Flag bit  
This bit represents the result of the latest CCA measurement.  
1 = Medium is busy  
0 = Medium is silent  

**bit 6**  
**CCAST:** Clear Channel Assessment Start bit\(^{(1)}\)  
Setting this register bit triggers MCU to start a new CCA measurement. The hardware clears this bit when the CCA measurement is done (EDCCAIF is set) and CCABUSY is valid.  

**bit 5-0**  
**RSSITHR<5:0>:** RSSI Threshold bits\(^{(2)}\)  
This threshold is used in CCA operation when Energy detect or Energy and Carrier Sense mode is selected.  
Representation: resolution of 2 dB/LSB, RSSITHR = 0x10 represents ca. -75 dBm noise level. Note that this threshold may be different with other matching network or antenna.  

**Note 1:** RX chain must be turned on (RXEN = 1) to perform this measurement. Packet reception is not disabled during the measurement, main purpose is testing.  
**Note 2:** In the corresponding CCA modes the radio measures EDMEAN<7:0>. If EDMEAN<7:2> is greater than RSSITHR<5:0>, CCABUSY is set. Example: To set the RSSI threshold where the chip measures 0x30 EDMEAN, RSSITHR<5:0> must be 0x0C.

### REGISTER 2-37: CCACON2 (CCA CONTROL 2 REGISTER)  
**ADDRESS:** 0x30

<table>
<thead>
<tr>
<th>R-0</th>
<th>R/W-01</th>
<th>R/W-01</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSTHR&lt;3:0&gt;</td>
<td>CCALEN&lt;1:0&gt;</td>
<td>CCAMODE&lt;1:0&gt;</td>
</tr>
</tbody>
</table>

**Legend:**  
- **R** = Readable bit  
- **W** = Writable bit  
- **U** = Unimplemented bit, read as ‘0’  
- **-n** = Value at POR  
- ‘1’ = Bit is set  
- ‘0’ = Bit is cleared  
- **x** = Bit is unknown  
- **r** = Reserved  

**bit 7-4**  
**CSTHR<3:0>:** Carrier Sense Threshold Field bits. This threshold is used in CCA operation when Carrier Sense mode is selected.  

**bit 3-2**  
**CCALEN<1:0>:** Clear Channel Assessment Length bits\(^{(1)}\)  
Value N indicates duration of 2^N * 32 µs.  

**bit 1-0**  
**CCAMODE<1:0>:** Clear Channel Assessment Mode Field bits\(^{(2)}\)  
11 = CCA Mode 3/a in the IEEE 802.15.4 standard: Energy AND Carrier Sense Threshold  
10 = CCA Mode 2 in the IEEE 802.15.4 standard: Carrier Sense Threshold  
01 = CCA Mode 1 in the IEEE 802.15.4 standard: Energy Detect Threshold (default)  
00 = CCA Mode 3/b in the IEEE 802.15.4 standard: Energy OR Carrier Sense Threshold  

**Note 1:** The IEEE 802.15.4 standard requires 128 µs, but shorter length is recommended when using higher rates with optimized Preamble mode (RATECON.OPTIMAL = 1).  
**Note 2:** The measured RSSI result is stored in EDMEAN<7:0> register in all modes except Mode 2.
REGISTER 2-38: EDCON (ENERGY DETECT CONTROL REGISTER)(1)  ADDRESS: 0x31

<table>
<thead>
<tr>
<th>Bit 7-6</th>
<th>R-00</th>
<th>R/W-0</th>
<th>R/W/HC-0</th>
<th>R/W-1110</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>EDMODE</td>
<td>EDST</td>
<td>EDLEN&lt;3:0&gt;</td>
<td></td>
</tr>
</tbody>
</table>

Legend:  
- **R** = Readable bit  
- **W** = Writable bit  
- **U** = Unimplemented bit, read as ‘0’  
- ‘1’ = Bit is set  
- ‘0’ = Bit is cleared  
- x = Bit is unknown  
- **HC** = Hardware Clear  
- **HS** = Hardware Set  

bit 7-6  
**Reserved**: Maintain as ‘0’

bit 5  
**EDMODE**: Energy Detect Mode Select bit  
1 = Energy Detect Sampling Mode. ED duration is 128 µs. A single atomic RSSI-peak measurement is accomplished. The result is stored in EDPEAK<7:0> register.  
0 = Energy Detect Scan Mode. EDLEN<3:0> sets the ED duration. The result is stored in EDMEAN<7:0> register.

bit 4  
**EDST**: Energy Detect Measurement Start bit  
Setting this register bit triggers MCU to start a new ED measurement. The hardware clears this bit when the ED measurement is done (EDCCAIF is unchanged) and values in EDMEAN<7:0> and EDPEAK<7:0> are valid.  
If the ED measurement is aborted (RX state changes, or the MCU clears the EDST bit), then EDCCAIF is unchanged.

bit 3-0  
**EDLEN<3:0>**: Energy Detect Measurement Length Field bits(2)  
Value M indicates a sequence of (M + 1) * 8 atomic RSSI-peak measurements, each having the duration of 128 µs. At the end of the aggregate measurement, the mean and the peak value of the sequence are available in EDMEAN<7:0> and EDPEAK<7:0>.

Note 1:  
The RX chain must be turned on (RXEN = 1) to perform this measurement. Packet reception is disabled during the measurement.

2:  
When EDLEN<3:0> = M = 0xE, the 128 µs atomic measurements are performed 120 times, which is equal to the a BaseSuperFrameDuration parameter in the IEEE 802.15.4 standard.

REGISTER 2-39: EDMEAN (ENERGY DETECT MEAN INDICATION REGISTER)  ADDRESS: 0x32

<table>
<thead>
<tr>
<th>Bit 7-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDMEAN&lt;7:0&gt;</td>
</tr>
</tbody>
</table>

Legend:  
- **R** = Readable bit  
- **W** = Writable bit  
- **U** = Unimplemented bit, read as ‘0’  
- ‘1’ = Bit is set  
- ‘0’ = Bit is cleared  
- x = Bit is unknown  
- **HC** = Hardware Clear  
- **HS** = Hardware Set  

bit 7-0  
**EDMEAN<7:0>**: Energy Detect Mean Indication Field bits  
Measured mean signal strength during ED/CCA measurement.
REGISTER 2-40: EDPEAK (ENERGY DETECT PEAK INDICATION REGISTER)  ADDRESS: 0x33

<table>
<thead>
<tr>
<th>R/HS/HC-00000000</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>EDPEAK&lt;7:0&gt;</td>
<td></td>
</tr>
</tbody>
</table>

bit 7-0  **EDPEAK<7:0>:** Energy Detect Peak Indication Field bits

Measured peak signal strength during ED measurement.

Computation: The gain-compensated RSSI value is averaged over intervals of 128 μs. The peak value obtained from a sequence of such measurements is stored in EDPEAK when EDMODE = 1.

REGISTER 2-41: CFOCON (CFO PRE COMPENSATION REGISTER)  ADDRESS: 0x34

<table>
<thead>
<tr>
<th>RW-0000</th>
<th>RW-0000</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFOTX&lt;3:0&gt;</td>
<td>CFORX&lt;3:0&gt;</td>
</tr>
</tbody>
</table>

Legend:  
- R = Readable bit  
- W = Writable bit  
- U = Unimplemented bit, read as '0'  
- ‘1’ = Bit is set  
- ‘0’ = Bit is cleared  
- x = Bit is unknown

bit 7-4  **CFOTX<3:0>:** TX Carrier Frequency Offset Field bits

The host writes this value to compensate for the Carrier Frequency Offset of the node during transmission. Pre-compensation allows using crystals with wider tolerances.

Frequency Offset Unit is: 13 ppm/LSB. Two's complement encoding.

bit 3-0  **CFORX<3:0>:** RX Carrier Frequency Offset Field bits

The host writes this value to pre-compensate the Carrier Frequency Offset estimation window (±55 ppm).

Frequency Offset Unit is: 13 ppm/LSB. Two's complement encoding.
REGISTER 2-42: CFOMEAS (CFO MEASUREMENT INDICATION REGISTER)  ADDRESS: 0x35

<table>
<thead>
<tr>
<th>bit 7-0</th>
<th>CFOMEAS&lt;7:0&gt;: CFO Measurement Field bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 7-0</td>
<td>CFO Measurement Field bits</td>
</tr>
</tbody>
</table>

If AFCOVR bit is cleared, then this register is written and valid when RXSFDIF is set with the value of the Carrier Frequency Offset that was estimated during the acquisition of the packet. The host may use this value together with the LQI as a preamble quality indication (LQI is measured over the CFO compensated payload).

If AFCOVR bit is set, this receiver compensates the Carrier Frequency Offset. Note that in this case, the CFO estimation algorithm is disabled, thus ±13 ppm CFO is tolerated. CFORX has no effect when AFCOVR is set.

Frequency Offset Unit is: ~1.62 ppm/LSB of the 2.4 GHz carrier. Two's complement encoding is used.
## REGISTER 2-43: RATECON (RATE CONFIGURATION REGISTER)  
**ADDRESS:** 0x36

<table>
<thead>
<tr>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-1</th>
<th>RW-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIS2000</td>
<td>DIS1000</td>
<td>DIS500</td>
<td>DIS250</td>
<td>DISSTD</td>
<td>DIS125</td>
<td>OPTIMAL</td>
<td>PSAV</td>
</tr>
</tbody>
</table>

### bit 7
**DIS2000:** Disable 2 Mbps Frame Reception bit  
If this bit is set, then reception of 2 Mbps frames is disabled.

### bit 6
**DIS1000:** Disable 1 Mbps Frame Reception bit  
If this bit is set, then reception of 1 Mbps frames is disabled.

### bit 5
**DIS500:** Disable 500 kbps Frame Reception bit  
If this bit is set, then reception of 500 kbps frames is disabled.

### bit 4
**DIS250:** Disable 250 kbps Frame Reception bit  
If this bit is set, then reception of 250 kbps frames with non-standard-compliant SFD patterns is disabled.

### bit 3
**DISSTD:** Disable IEEE 802.15.4 compliant Frame Reception bit  
If this bit is set, then reception of 250 kbps frames with IEEE 802.15.4 compliant SFD patterns is disabled.

### bit 2
**DIS125:** Disable 125 kbps Frame Reception bit  
If this bit is set, then reception of 125 kbps frames is disabled.

### bit 1
**OPTIMAL:** Optimized Preamble Selection bit  
When this bit is set, then optimized preamble is used instead of legacy.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Optimized preamble</td>
</tr>
<tr>
<td>0</td>
<td>Legacy preamble</td>
</tr>
</tbody>
</table>

### bit 0
**PSAV:** Power-Save Mode Selection bit  
If this bit is set, frame detection is dependent on the RSSI signal, and the receive signal processor is turned on when a sudden and significant increase (PSAVTHR<3:0>) is detected in the signal strength or the signal strength is above an absolute level (DESENSTHR<3:0>).

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Power-Save mode</td>
</tr>
<tr>
<td>0</td>
<td>Hi-Sensitivity mode</td>
</tr>
</tbody>
</table>
## REGISTER 2-44: POWSAVE (POWER-SAVE CONFIGURATION REGISTER)  ADDRESS 0x37

<table>
<thead>
<tr>
<th></th>
<th>R/W-1010</th>
<th>R/W-1010</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESENSTHR&lt;3:0&gt;</td>
<td></td>
<td>PSAVTHR&lt;3:0&gt;</td>
</tr>
<tr>
<td>bit 7</td>
<td></td>
<td>bit 0</td>
</tr>
</tbody>
</table>

**Legend:**  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
-n = Value at POR  
'1' = Bit is set  
'0' = Bit is cleared  
x = Bit is unknown  
r = Reserved

**bit 7-4**  
**DESENSTHR<3:0>:** Desensitization Threshold Field bits  
This field defines an absolute level on the RSSI signal to activate receive signal processor if PSAV = 1.  
Unit is 4 dB/LSB. Unsigned encoding is used.

**bit 3-0**  
**PSAVTHR<3:0>:** Frame Detection Threshold Register Field bits  
This field defines a relative (relative to the last 4 µs RSSI value) threshold level on the RSSI signal to activate receive signal processor, if PSAV = 1.  
Unit is 0.5 dB/LSB. Unsigned encoding is used.
REGISTER 2-45: BBCON (BASEBAND CONFIGURATION REGISTER)  

ADDRESS 0x38

<table>
<thead>
<tr>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-11</th>
<th>RW-0</th>
<th>RW-001</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNDMOD</td>
<td>AFCOVR</td>
<td>RXGAIN&lt;1:0&gt;</td>
<td>PRMBHOLD</td>
<td>PRMBSZ&lt;2:0&gt;</td>
</tr>
</tbody>
</table>

Legend:

W = Writable bit  
R = Readable bit  
U = Unimplemented bit, read as ‘0’  
-n = Value at POR  
‘1’ = Bit is set  
‘0’ = Bit is cleared  
x = Bit is unknown  
r = Reserved

bit 7  
**RNDMOD**: Random modulation bit  
If this bit is set, the transmitter randomly transmits DSSS symbols or MSK chips if PRMBHOLD bit is set. The purpose of this register is only for testing.

bit 6  
**AFCOVR**: AFC override bit  
If this bit is set, the receiver uses CFOMEAS register as the CFO in reception.

bit 5-4  
**RXGAIN<1:0>**: Receiver Gain Register Field bits  
If this bit is set, the AGC operation is inhibited in the receiver and the receiver radio gain configuration is selected between three different gain levels. Encoding:

- **11**: AGC operation is enabled (default value)  
- **10**: High gain  
- **01**: Middle gain  
- **00**: Low gain  

This feature is used for testing and streaming purposes. To reduce the required interframe-gap, the RXGAIN must be set to one of the fixed gain options when the MAC is in Streaming mode.

bit 3  
**PRMBHOLD**: Preamble Hold Enable bit  
Effect: Appends extra bytes to the transmitted preamble in endless repetition until it is cleared.  
Details: The hardware checks this bit during transmission before finishing the preamble. The DR<2:0> and the register OPTIMAL determine the appropriate preamble byte and applies the modulation format. When this flag is released, the transmission of the current preamble byte is completed followed by transmitting the LENGTH field and the payload.

- **1**: Enable endless preamble repetition  
- **0**: Disable/stop endless preamble repetition

bit 2-0  
**PRMBSZ<2:0>**: Preamble Size Adjustment Field bits  
Allows adjusting the transmitted preamble length when OPTIMAL = 1. Encoding:

- **500 kbps preamble length** = (PRMBSZ<2> + 4) units, where unit = 16 μs (1 octet at 500 kbps)  
- **1 Mbps preamble length** = (PRMBSZ<1:0> + 8) units, where unit = 4 μs (1 octet at 2 Mbps)  
- **2 Mbps preamble length** = (PRMBSZ<1:0> + 8) units, where unit = 4 μs (1 octet at 2 Mbps)  

Legacy frames and 125/250 kbps optimized frames are not affected by this Register field.
REGISTER 2-46: IFGAP (INTER FRAME CONFIGURATION REGISTER)    ADDRESS 0x39

<table>
<thead>
<tr>
<th>R-000</th>
<th>R/W-10111</th>
</tr>
</thead>
<tbody>
<tr>
<td>IFGAP&lt;4:0&gt;</td>
<td>bit 7-5</td>
</tr>
</tbody>
</table>

Legend:  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
-n = Value at POR  
'1' = Bit is set  
'0' = Bit is cleared  
x = Bit is unknown  
r = Reserved

bit 7-5  
Reserved: Maintain as ‘0’

bit 4-0  
IFGAP<4:0>: TX Interframe-Gap Field bits
This field allows configuring a TX interframe-gap ranging from 0 to 30 μs. This duration is enforced as a minimum separation between the last sample of a transmitted frame and the start of the preamble for a potential subsequent frame transmission. Unit is 2 μs/LSB.

REGISTER 2-47: TXPOW (TRANSMIT POWER CONFIGURATION REGISTER)    ADDRESS 0x3A

<table>
<thead>
<tr>
<th>R/W-000</th>
<th>R/W-11111</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHIPBOOST&lt;2:0&gt;</td>
<td>TXPOW&lt;4:0&gt;</td>
</tr>
</tbody>
</table>

Legend:  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
-n = Value at POR  
'1' = Bit is set  
'0' = Bit is cleared  
x = Bit is unknown  
r = Reserved

bit 7-5  
CHIPBOOST<2:0>: TX Chip Boosting Field bits
This field modifies the spectrum of the OQPSK transmission.

bit 4-0  
TXPOW<4:0>: TX Power Register Field bits
This field allows configuring a TX power ranging from -19 to 1 dBm. Encoding:

11111 = +1 dBm
.
.
.
.
00001 = -19 dBm
00000 = PA OFF
REGISTER 2-48: TX2IDLE (TRANSMIT POWER-DOWN TO IDLE CONFIGURATION REGISTER)  
ADDRESS 0x3B

<table>
<thead>
<tr>
<th>R-0</th>
<th>RW-00011</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>r</strong></td>
<td><strong>TX2IDLE&lt;4:0&gt;</strong></td>
</tr>
</tbody>
</table>

**Legend:**  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as ‘0’  
- = Value at POR  
‘1’ = Bit is set  
‘0’ = Bit is cleared  
x = Bit is unknown  
r = Reserved

- **bit 7-5**  
  Reserved: Maintain as ‘0’

- **bit 4-0**  
  **TX2IDLE<4:0>:** Transmit Power-Down to Idle Duration Field bits  
  Defines the duration of the interval while PLL cannot be tuned (turned off or change channel) following that the transmitter and external PA (if PAE = 1) are turned down together.  
  Representation: 1 µs/1 LSB. No offset.

REGISTER 2-49: TX2TXMA (TRANSMIT POWER-UP TO MEDIUM ACCESS CONFIGURATION REGISTER)  
ADDRESS 0x3C

<table>
<thead>
<tr>
<th>R-0</th>
<th>RW-00011</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>r</strong></td>
<td><strong>TX2TXMA&lt;4:0&gt;</strong></td>
</tr>
</tbody>
</table>

**Legend:**  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as ‘0’  
- = Value at POR  
‘1’ = Bit is set  
‘0’ = Bit is cleared  
x = Bit is unknown  
r = Reserved

- **bit 7-5**  
  Reserved: Maintain as ‘0’

- **bit 4-0**  
  **TX2TXMA<4:0>:** Transmit Power-Up to Medium Access Configuration Field bits  
  Defines the time interval between turning on the internal transmitter of the device and the start time of medium access (start of the PHY-layer frame).  
  **TX_TO_TXMA = The transient time of the transmitter, in the following scenarios:**  
  PAEN = 0  
  PAEN = 1, but the PA is turned on first. PA_TO_TXMA = TX_TO_TXMA + PA transient time.  
  PAEN = 1, but the TX and PA transients are NOT sequenced.  
  **TX_TO_TXMA = The transient time of the transmitter + PA_TO_TXMA:**  
  PAEN = 1, and the transmitter is turned on first (transients are sequenced).  
  Representation: 1 µs/1 LSB. No offset.
REGISTER 2-50: EXTPA (EXTERNAL POWER AMPLIFIER CONFIGURATION REGISTER)  
ADDRESS 0x3D

<table>
<thead>
<tr>
<th>R-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-0100</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>EXTPAP</td>
<td>PAEN</td>
<td>PA2TXMA&lt;4:0&gt;</td>
</tr>
</tbody>
</table>

Legend:  
R = Readable bit  W = Writable bit  U = Unimplemented bit, read as ‘0’  
-n = Value at POR  ‘1’ = Bit is set  ‘0’ = Bit is cleared  x = Bit is unknown  
r = Reserved

bit 7  
Reserved: Maintain as ‘0’

bit 6  
EXTPAP: External Power Amplifier Polarity bit  
1 = 3.3V turns Power Amplifier ON  
0 = GND turns Power Amplifier ON

bit 5  
PAEN: External Power Amplifier Enable bit  
This bit enables the PA pin to output the control signal for external Power Amplifier.

bit 4-0  
PA2TXMA<4:0>: External Power Amplifier Power-Up to Medium Access Configuration Field bits  
Defines the time interval between turning on the external PA of the device and the start time of medium access (start of the PHY-layer frame).  
PA_TO_TXMA = The transient time of the external PA, in the following scenarios:  
PAEN = 1, and the transmitter is turned on first. TX_TO_TXMA = PA_TO_TXMA + TX transient time.  
PAEN = 1, but the TX and PA transients are NOT sequenced.  
PA_TO_TXMA = The transient time of the PA + TX_TO_TXMA:  
PAEN = 1, and the external power amplifier is turned on first (transients are sequenced).  
Representation: 1 \( \mu \)s/1 LSB. No offset

REGISTER 2-51: EXTLNA (EXTERNAL LOW-NOISE AMPLIFIER CONFIGURATION REGISTER)  
ADDRESS 0x3E

<table>
<thead>
<tr>
<th>R-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-0100</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>EXTLNAP</td>
<td>LNAEN</td>
<td>LNADLY&lt;4:0&gt;</td>
</tr>
</tbody>
</table>

Legend:  
R = Readable bit  W = Writable bit  U = Unimplemented bit, read as ‘0’  
-n = Value at POR  ‘1’ = Bit is set  ‘0’ = Bit is cleared  x = Bit is unknown  
r = Reserved

bit 7  
Reserved: Maintain as ‘0’

bit 6  
EXTLNAP: External Low Noise Amplifier Polarity bit  
1 = 3.3V turns Low-Noise Amplifier ON  
0 = GND turns Low-Noise Amplifier ON

bit 5  
LNAEN: External Low-Noise Power Amplifier Enable bit  
This bit enables the LNA pin to output the control signal for external Low-Noise Amplifier.

bit 4-0  
LNADLY<4:0>: External Low-Noise Amplifier Power-Up Transient Delay Field bits  
Defines the duration between the LNA is turned on and the reception is valid.  
LNA and internal receiver are turned on together. The longer transient is awaited before input signal is accepted as valid.  
Representation: 1 \( \mu \)s/1 LSB. No offset.
REGISTER 2-52: BATMON (BATTERY MONITOR CONFIGURATION REGISTER) ADDRESS: 0x3F

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Value at POR</th>
</tr>
</thead>
<tbody>
<tr>
<td>7-6</td>
<td>Reserved: Maintain as ‘0’</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>BATMONPD: Battery Monitor Power-Down bit</td>
<td></td>
</tr>
<tr>
<td></td>
<td>If battery monitor is working and battery voltage drops below the threshold by BATMON&lt;4:0&gt;, then WARNIF is set.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = Battery monitor is OFF</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Battery monitor is working</td>
<td></td>
</tr>
<tr>
<td>4-0</td>
<td>BATMON&lt;4:0&gt;: Battery Monitor Threshold Field bits</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$V_{\text{THRESHOLD}} = 3.6 - 0.071 * \text{BATMON}&lt;4:0&gt; \ (V)$</td>
<td></td>
</tr>
</tbody>
</table>

REGISTER 2-53: SECKEY (SECURITY KEY REGISTER) ADDRESS: 0x40 – 0x4F

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Value at POR</th>
</tr>
</thead>
<tbody>
<tr>
<td>127</td>
<td>SECKEY&lt;127:120&gt;</td>
<td></td>
</tr>
<tr>
<td>119</td>
<td>SECKEY&lt;119:112&gt;</td>
<td></td>
</tr>
<tr>
<td>111</td>
<td>SECKEY&lt;111:104&gt;</td>
<td></td>
</tr>
<tr>
<td>103</td>
<td>SECKEY&lt;103:96&gt;</td>
<td></td>
</tr>
<tr>
<td>95</td>
<td>SECKEY&lt;95:88&gt;</td>
<td></td>
</tr>
<tr>
<td>87</td>
<td>SECKEY&lt;87:80&gt;</td>
<td></td>
</tr>
<tr>
<td>79</td>
<td>SECKEY&lt;79:72&gt;</td>
<td></td>
</tr>
</tbody>
</table>
## REGISTER 2-53: SECKEY (SECURITY KEY REGISTER) (CONTINUED)  
ADDRESS: 0x40 – 0x4F

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Readable</th>
<th>Writable</th>
<th>Unimplemented</th>
<th>Reserved</th>
<th>Security Key Field bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>71</td>
<td>SECKEY&lt;71:64&gt;</td>
<td>R/W</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>SECKEY&lt;63:56&gt;</td>
<td>R/W</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>SECKEY&lt;55:48&gt;</td>
<td>R/W</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>SECKEY&lt;47:40&gt;</td>
<td>R/W</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>SECKEY&lt;39:32&gt;</td>
<td>R/W</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>SECKEY&lt;31:24&gt;</td>
<td>R/W</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>SECKEY&lt;23:16&gt;</td>
<td>R/W</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>SECKEY&lt;15:8&gt;</td>
<td>R/W</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>SECKEY&lt;7:0&gt;</td>
<td>R/W</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Legend:**
- **R** = Readable bit
- **W** = Writable bit
- **U** = Unimplemented bit, read as '0'
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- **x** = Bit is unknown
- **r** = Reserved

**bit 127-0 SECKEY<128:0>:** Security Key Field bits

Security key used in security operation.
REGISTER 2-54: SECNONCE (SECURITY NONCE REGISTER)

ADDRESS: 0x50 – 0x5C

<table>
<thead>
<tr>
<th>Bit Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>103:96</td>
<td>SECNONCE&lt;103:96&gt;</td>
</tr>
<tr>
<td>95:88</td>
<td>SECNONCE&lt;95:88&gt;</td>
</tr>
<tr>
<td>87:80</td>
<td>SECNONCE&lt;87:80&gt;</td>
</tr>
<tr>
<td>79:72</td>
<td>SECNONCE&lt;79:72&gt;</td>
</tr>
<tr>
<td>71:64</td>
<td>SECNONCE&lt;71:64&gt;</td>
</tr>
<tr>
<td>63:56</td>
<td>SECNONCE&lt;63:56&gt;</td>
</tr>
<tr>
<td>55:48</td>
<td>SECNONCE&lt;55:48&gt;</td>
</tr>
<tr>
<td>47:40</td>
<td>SECNONCE&lt;47:40&gt;</td>
</tr>
<tr>
<td>39:32</td>
<td>SECNONCE&lt;39:32&gt;</td>
</tr>
<tr>
<td>31:24</td>
<td>SECNONCE&lt;31:24&gt;</td>
</tr>
<tr>
<td>23:16</td>
<td>SECNONCE&lt;23:16&gt;</td>
</tr>
<tr>
<td>15:8</td>
<td>SECNONCE&lt;15:8&gt;</td>
</tr>
</tbody>
</table>

R/W/HS/HC-00000000

bit 103
R/W-00000000
bit 95
R/W-00000000
bit 87
R/W-00000000
bit 79
R/W-00000000
bit 71
R/W-00000000
bit 63
R/W-00000000
bit 55
R/W-00000000
bit 47
R/W-00000000
bit 39
R/W-00000000
bit 31
R/W-00000000
bit 23
R/W-00000000
bit 15
REGISTER 2-54: SECNONCE (SECURITY NONCE REGISTER) (CONTINUED)

ADDRESS: 0x50 – 0x5C

R/W-00000000

<table>
<thead>
<tr>
<th>bit 7</th>
<th>bit 0</th>
</tr>
</thead>
</table>

**Legend:**

- **R** = Readable bit
- **W** = Writable bit
- **U** = Unimplemented bit, read as '0'
- **-n** = Value at POR
- **'1'** = Bit is set
- **'0'** = Bit is cleared
- **x** = Bit is unknown
- **r** = Reserved
- **HC** = Hardware Clear
- **HS** = Hardware Set

*bit 103-0 SECNONCE<103:0>: Security Nonce Field bits*

The register represents security nonce used in security operation.

This field is deterministic in both 802.15.4-2003 and 802.15.4-2006 standards. Device can automatically calculate this field.

REGISTER 2-55: SFD1 (START FRAME DELIMITER PATTERN 1 CONFIGURATION REGISTER)

ADDRESS: 0x60

R/W-00100001

<table>
<thead>
<tr>
<th>bit 7</th>
<th>bit 0</th>
</tr>
</thead>
</table>

**Legend:**

- **R** = Readable bit
- **W** = Writable bit
- **U** = Unimplemented bit, read as '0'
- **-n** = Value at POR
- **'1'** = Bit is set
- **'0'** = Bit is cleared
- **x** = Bit is unknown
- **r** = Reserved

*bit 7-0 SFD1<7:0>: Start Frame Delimiter Pattern 1 Register Field bits*

This octet is used as SFD pattern with 2 Mbps rate when OPTIMAL = 0, and as the MSB of the SFD pattern with 2 Mbps rate when OPTIMAL = 1.

**When OPTIMAL = 0:**

The hexadecimal digits must be different from 0x0 and different from the corresponding digits in SFD<k>, k = 2, 3, 4, 6, and the value 0xA7 is forbidden.

**When OPTIMAL = 1:**

The hexadecimal digits must be different from 0x0 and different from the corresponding digits of SFD2.
**REGISTER 2-56: SFD2 (START FRAME DELIMITER PATTERN 2 CONFIGURATION REGISTER)**

**ADDRESS: 0x61**

<table>
<thead>
<tr>
<th>RW-11110001</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SFD2&lt;7:0&gt;</td>
<td></td>
</tr>
<tr>
<td>bit 7</td>
<td>bit 0</td>
</tr>
</tbody>
</table>

**Legend:**
- R = Readable bit
- W = Writable bit
- U = Unimplemented bit, read as ‘0’
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- x = Bit is unknown
- -n = Value at POR
- r = Reserved

**bit 7-0** SFD2<7:0>: Start Frame Delimiter Pattern 2 Register Field bits

This octet is used as SFD pattern with 1 Mbps rate when OPTIMAL = 0, and as the MSB of the SFD pattern with 1 Mbps rate when OPTIMAL = 1.

When OPTIMAL = 0:

The hexadecimal digits must be different from 0x0 and different from the corresponding digits in SFD<k>, k = 1, 3, 4, 6, and the value 0xA7 is forbidden.

When OPTIMAL = 1:

The hexadecimal digits must be different from 0x0 and different from the corresponding digits of SFD2.

---

**REGISTER 2-57: SFD3 (START FRAME DELIMITER PATTERN 3 CONFIGURATION REGISTER)**

**ADDRESS: 0x62**

<table>
<thead>
<tr>
<th>RW-00111011</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SFD3&lt;7:0&gt;</td>
<td></td>
</tr>
<tr>
<td>bit 7</td>
<td>bit 0</td>
</tr>
</tbody>
</table>

**Legend:**
- R = Readable bit
- W = Writable bit
- U = Unimplemented bit, read as ‘0’
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- x = Bit is unknown
- -n = Value at POR
- r = Reserved

**bit 7-0** SFD3<7:0>: Start Frame Delimiter Pattern 3 Register Field bits

This octet is used as SFD pattern with 500 kbps rate.

When OPTIMAL = 0:

The hexadecimal digits must be different from 0x0 and different from the correspond digits in SFD<k>, k = 1, 2, 4, 6, and the value 0xA7 is forbidden.

When OPTIMAL = 1:

The hexadecimal digits must be different from 0x0.
REGISTER 2-58: SFD4 (START FRAME DELIMITER PATTERN 4 CONFIGURATION REGISTER)
ADDRESS: 0x63

<table>
<thead>
<tr>
<th>RW-1100101</th>
<th>SFD4&lt;7:0&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 7</td>
<td>bit 0</td>
</tr>
</tbody>
</table>

**Legend:**
- R = Readable bit
- W = Writable bit
- U = Unimplemented bit, read as ‘0’
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- x = Bit is unknown
- r = Reserved

**bit 7-0 SFD4<7:0>:** Start Frame Delimiter Pattern 4 Register Field bits
This octet is used as SFD pattern with 250 kbps rate when proprietary MAC is in use. Otherwise, the 0xA7 pattern defined in the standard is used instead.

The hexadecimal digits must be different from 0x0 and from the corresponding digits in SFD<k>, where k = 6 or 1, 2, 3. When OPTIMAL = 0, the value 0xA7 is forbidden.

REGISTER 2-59: SFD5 (START FRAME DELIMITER PATTERN 5 CONFIGURATION REGISTER)
ADDRESS: 0x64

<table>
<thead>
<tr>
<th>RW-01001101</th>
<th>SFD5&lt;7:0&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 7</td>
<td>bit 0</td>
</tr>
</tbody>
</table>

**Legend:**
- R = Readable bit
- W = Writable bit
- U = Unimplemented bit, read as ‘0’
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- x = Bit is unknown
- r = Reserved

**bit 7-0 SFD5<7:0>:** Start Frame Delimiter Pattern 5 Register Field bits
This octet is used as the MSB of the SFD pattern with 125 kbps rate.
REGISTER 2-60: SFD6 (START FRAME DELIMITER PATTERN 6 CONFIGURATION REGISTER)  
ADDRESS: 0x65

<table>
<thead>
<tr>
<th>Bit</th>
<th>Address</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>R/W-10101000</td>
<td>SFD6&lt;7:0&gt;</td>
</tr>
</tbody>
</table>

Legend:  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as ‘0’  
- = Value at POR  
1 = Bit is set  
0 = Bit is cleared  
x = Bit is unknown  
r = Reserved

bit 7-0  
SFD6<7:0>: Start Frame Delimiter Pattern 6 Register Field bits  
When OPTIMAL = 1:  
This octet is used as the LSB of the SFD pattern with 2 Mbps rate. This octet is used as the LSB of the SFD pattern with 125 kbps rate.  
When OPTIMAL = 0:  
The value 0xA7 is forbidden. The hexadecimal digits must be different from 0x0 and different from the corresponding digits in SFD<k>, k = 4 or 1, 2, 3.

REGISTER 2-61: SFD7 (START FRAME DELIMITER PATTERN 7 CONFIGURATION REGISTER)  
ADDRESS: 0x66

<table>
<thead>
<tr>
<th>Bit</th>
<th>Address</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>R/W-11001000</td>
<td>SFD7&lt;7:0&gt;</td>
</tr>
</tbody>
</table>

Legend:  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as ‘0’  
- = Value at POR  
1 = Bit is set  
0 = Bit is cleared  
x = Bit is unknown  
r = Reserved

bit 7-0  
SFD7<7:0>: Start Frame Delimiter Pattern 7 Register Field bits  
When OPTIMAL = 1, this octet is used as the LSB of the SFD pattern with 1 Mbps rate.
3.0 FUNCTIONAL DESCRIPTION

3.1 Reset

MRF24XA has three reset types:

- Power-On Reset (POR) – MRF24XA has built-in POR circuitry that automatically resets all control registers when power is applied. After POR, MRF24XA starts the internal Calibration process. RDYIF interrupt is set when the device is ready to use.

- RESET Pin – The host MCU can reset MRF24XA by asserting the RESET pin low. All control registers are reset to default value. If the RESET pin is deasserted, MRF24XA starts the internal Calibration process. RDYIF interrupt is set when the device is ready to use.

- Software Reset – The host MCU can perform the Software Reset through the SPI interface. REGRST register (0x00) provides reset signals for the Configuration registers, while FSMRST register (0x01) provides reset functionality for the internal state machines. The reset signals are asynchronous and the level is evaluated immediately without any internal synchronization.

The recommended reset sequences:
- FSMRST = 0x1F
- REGRST = 0x3F
- REGRST = 0x00

REGISTER 3-1: REGRST (CONFIGURATION RESET)(1) ADDRESS: 0x00

<table>
<thead>
<tr>
<th>R-00</th>
<th>R/W-000000</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 7</td>
<td>REGRST&lt;5:0&gt;</td>
</tr>
</tbody>
</table>

Legend:  
- R = Readable bit  
- W = Writable bit  
- U = Unimplemented bit, read as ‘0’  
- -n = Value at POR  
- ‘1’ = Bit is set  
- ‘0’ = Bit is cleared  
- x = Bit is unknown  
- r = Reserved

bit 7-6 Reserved: Maintain as ‘0’
bit 5-0 REGRST<5:0>: Asynchronous Register Reset Field bits
000000 = Release from reset
111111 = Reset Configuration registers to default

Note 1: After setting the field, the host MCU must also clear it to release the device from reset.

REGISTER 3-2: FSMRST (CONTROLLER RESET)(1) ADDRESS: 0x01

<table>
<thead>
<tr>
<th>R-000</th>
<th>R/W-00000</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 7</td>
<td>FSMRST&lt;4:0&gt;</td>
</tr>
</tbody>
</table>

Legend:  
- R = Readable bit  
- W = Writable bit  
- U = Unimplemented bit, read as ‘0’  
- -n = Value at POR  
- ‘1’ = Bit is set  
- ‘0’ = Bit is cleared  
- x = Bit is unknown  
- r = Reserved

bit 7-5 Reserved: Maintain as ‘0’
bit 4-0 FSMRST<4:0>: Asynchronous Functional Reset Field bits
000000 = Release from reset
111111 = Reset state machines to default

Note 1: After setting the field, the host MCU must clear it to release the device from reset.
### 3.2 Interrupts

MRF24XA has one interrupt (INT), pin 13 that signals interrupt events to the host MCU. Interrupt sources are enabled through PIE1 (0x08) to PIE4 (0x0B) register bits. All interrupts are enabled or disabled using GIE bit (PINCON<6>). If GIE bit is cleared, all interrupts are disabled and INT pin remains in inactive state. Despite having the interrupts cleared by GIE bit clearing, the interrupt flags of the enabled interrupt sources are set.

Interrupt flags are located in the PIR1 (0x04) to PIR4 (0x07) registers. The PIRX register bits clears-to-zero upon read.

Therefore, the host MCU must read and store the value of the PIRX registers and check the bits to determine which interrupt occurred. The INT pin continues to signal an interrupt until all active interrupts flags in PIRX registers are read.

<table>
<thead>
<tr>
<th>REGISTER 3-3: PINCON (PIN CONFIGURATION REGISTER)</th>
<th>ADDRESS: 0x0C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Bit 7</td>
</tr>
<tr>
<td>REGRST</td>
<td>r</td>
</tr>
<tr>
<td>FSMRST</td>
<td>r</td>
</tr>
</tbody>
</table>

Legend:  
r = Reserved, read as ‘0’.

<table>
<thead>
<tr>
<th>Address</th>
<th>R-0</th>
<th>R/W-1</th>
<th>R-0</th>
<th>R-x</th>
<th>R/W-0000</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>GIE</td>
<td>r</td>
<td>IRQIF</td>
<td>GPIOMODE&lt;3:0&gt;</td>
<td></td>
</tr>
</tbody>
</table>

Legend:  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as ‘0’  
-n = Value at POR  
‘1’ = Bit is set  
‘0’ = Bit is cleared  
x = Bit is unknown  
r = Reserved

- **bit 7**  
  **Reserved:** Maintain as ‘0’

- **bit 6**  
  **GIE:** General Interrupt Enable bit  
  This bit enables to output IRQIF on \( \text{INT} \) pin. Note that the polarity of \( \text{INT} \) pin is active-low.

- **bit 5**  
  **Reserved:** Maintain as ‘0’

- **bit 4**  
  **IRQIF:** Interrupt Request Pending bit  
  This bit is the OR relationship of the enabled interrupt flags.

- **bit 3-0**  
  **GPIOMODE <3:0>:** GPIO Mode Field bits  
  This bit field is out of scope.
3.2.1 PIEx - INTERRUPT ENABLE
REGISTERS
Register bits of PIE1 to PIE4 registers enable the appropriate interrupt sources to generate interrupts to the host MCU through INT pin. The interrupt is enabled when the appropriate bit is set to ‘1’.

**REGISTER 3-4: PIE1 (PERIPHERAL INTERRUPT ENABLE 1) ADDRESS: 0x08**

<table>
<thead>
<tr>
<th></th>
<th>R-0</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R-0</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 7</td>
<td>r</td>
<td>RDYIE</td>
<td>IDLEIE</td>
<td>r</td>
<td>CALSOIE</td>
<td>CALHAIE</td>
<td>r</td>
</tr>
</tbody>
</table>

**Legend:**

- **R** = Readable bit
- **W** = Writable bit
- **U** = Unimplemented bit, read as ‘0’
- ‘-n’ = Value at POR
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- ‘x’ = Bit is unknown
- **r** = Reserved

- **bit 7-6 Reserved:** Maintain as ‘0’
- **bit 5 RDYIE:** Ready Interrupt Enable bit
  This bit masks the RDYIF interrupt bit.
- **bit 4 IDLEIE:** Idle Interrupt Enable bit
  This bit masks the IDLEIF interrupt bit.
- **bit 3 Reserved:** Maintain as ‘0’
- **bit 2 CALSOIE:** Calibration Soft Interrupt Enable bit
  This bit masks the CALSOIF interrupt bit.
- **bit 1 CALHAIE:** Calibration Hard Interrupt Enable bit
  This bit masks the CALHAIF interrupt bit.
- **bit 0 Reserved:** Maintain as ‘0’
## REGISTER 3-5: PIE2 (PERIPHERAL INTERRUPT ENABLE 2)  
**ADDRESS:** 0x09

<table>
<thead>
<tr>
<th>Bit 7</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>TXIE</td>
<td>TXIE</td>
</tr>
</tbody>
</table>

### Legend:
- **R** = Readable bit
- **W** = Writable bit
- **U** = Unimplemented bit, read as '0'
- **-n** = Value at POR
- **‘1’** = Bit is set
- **‘0’** = Bit is cleared
- **x** = Bit is unknown
- **r** = Reserved

- **bit 7**: **TXIE**: Transmit Interrupt Enable
  - This bit masks the TXIF interrupt register.
- **bit 6**: **TXENCIE**: Transmit Encryption and Authentication Interrupt Enable bit
  - This bit masks the TXENCIF interrupt register.
- **bit 5**: **TXMAIE**: Transmitter Medium Access Interrupt Enable bit
  - This bit masks the TXMAIF interrupt register.
- **bit 4**: **TXACKIE**: Transmission Unacknowledged Failure Interrupt Enable bit
  - This bit masks the TXACKIF interrupt register.
- **bit 3**: **TXCSMAIE**: Transmitter CSMA Failure Interrupt Enable bit
  - This bit masks the TXCSMAIF interrupt register.
- **bit 2**: **TXSZIE**: Transmit Packet Size Error Interrupt Enable bit
  - This bit masks the TXSZIF interrupt register.
- **bit 1**: **TXOVFIE**: Transmitter Overflow Interrupt Enable bit
  - This bit masks the TXOVFIF interrupt register.
- **bit 0**: **FRMIE**: Frame Format Error Interrupt Flag bit
  - This bit masks the FRMIF interrupt register.
### REGISTER 3-6: PIE3 (PERIPHERAL INTERRUPT ENABLE 3)

<table>
<thead>
<tr>
<th>Bit</th>
<th>Name</th>
<th>Description</th>
<th>Readable</th>
<th>Writable</th>
<th>Unimplemented</th>
<th>Reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>RXIE</td>
<td>Received Successful Interrupt Enable bit</td>
<td>R</td>
<td>W</td>
<td>U</td>
<td>r</td>
</tr>
<tr>
<td></td>
<td>This bit masks the RXIF interrupt register.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>RXDECIE</td>
<td>Receiver Decryption/Authentication Passed Interrupt Enable bit</td>
<td>R</td>
<td>W</td>
<td>U</td>
<td>r</td>
</tr>
<tr>
<td></td>
<td>This bit masks the RXDECIF interrupt register.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>RXTAGIE</td>
<td>Receiver Decryption/Authentication Failure Interrupt Enable bit</td>
<td>R</td>
<td>W</td>
<td>U</td>
<td>r</td>
</tr>
<tr>
<td></td>
<td>This bit masks the RXTAGIF interrupt register.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Reserved</td>
<td>Maintain as ‘0’</td>
<td>R</td>
<td>W</td>
<td>U</td>
<td>r</td>
</tr>
<tr>
<td>3</td>
<td>RXIDENTIE</td>
<td>Received Packet Identical Interrupt Enable bit</td>
<td>R</td>
<td>W</td>
<td>U</td>
<td>r</td>
</tr>
<tr>
<td></td>
<td>This bit masks the RXIDENTIF interrupt register.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>RXFLTIE</td>
<td>Received Packet Filtered Interrupt Enable bit</td>
<td>R</td>
<td>W</td>
<td>U</td>
<td>r</td>
</tr>
<tr>
<td></td>
<td>This bit masks the RXFLTIF interrupt register.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>RXOVFIE</td>
<td>Receiver Overflow Interrupt Enable bit</td>
<td>R</td>
<td>W</td>
<td>U</td>
<td>r</td>
</tr>
<tr>
<td></td>
<td>This bit masks the RXOVFIF interrupt register.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>STRMIE</td>
<td>Receive Stream Time-Out Error Interrupt Enable bit</td>
<td>R</td>
<td>W</td>
<td>U</td>
<td>r</td>
</tr>
<tr>
<td></td>
<td>This bit masks the STRMIF interrupt register.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Legend:**
- **R** = Readable bit
- **W** = Writable bit
- **U** = Unimplemented bit, read as ‘0’
- **-n** = Value at POR
- **‘1’** = Bit is set
- **‘0’** = Bit is cleared
- **x** = Bit is unknown
- **r** = Reserved
### REGISTER 3-7: PIE4 (PERIPHERAL INTERRUPT ENABLE 4)  
**ADDRESS: 0x0B**

<table>
<thead>
<tr>
<th>bit</th>
<th>bit 7</th>
<th>bit 6</th>
<th>bit 5</th>
<th>bit 4</th>
<th>bit 3</th>
<th>bit 2</th>
<th>bit 1</th>
<th>bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TXSFDIE: Transmit SFD Sent Interrupt Enable bit</td>
<td>RXSFDIE: Receive SFD Detected Interrupt Enable bit</td>
<td>ERRORIE: General Error Interrupt Enable bit</td>
<td>WARNIE: Warning Interrupt Enable bit</td>
<td>EDCCAIE: Energy Detect/CCA Done Interrupt Enable bit</td>
<td>GPIO2IE: GPIO2 Interrupt Enable bit</td>
<td>GPIO1IE: GPIO1 Interrupt Enable bit</td>
<td>GPIO0IE: GPIO0 Interrupt Enable bit</td>
</tr>
<tr>
<td></td>
<td>This bit masks the TXSFDIF interrupt register.</td>
<td>This bit masks the RXSFDIF Interrupt Enable.</td>
<td>This bit masks the ERRORIF interrupt register.</td>
<td>This bit masks the WARNIF interrupt register.</td>
<td>This bit masks the EDCCAIF interrupt register.</td>
<td>This bit masks the GPIO2IF interrupt register.</td>
<td>This bit masks the GPIO1IF interrupt register.</td>
<td>This bit masks the GPIO0IF interrupt register.</td>
</tr>
</tbody>
</table>

**Legend:**  
- **R** = Readable bit  
- **W** = Writable bit  
- **U** = Unimplemented bit, read as ‘0’  
- **-n** = Value at POR  
- **‘1’** = Bit is set  
- **‘0’** = Bit is cleared  
- **x** = Bit is unknown  
- **r** = Reserved
3.2.2 PIRX- PERIPHERAL INTERRUPT
REGISTERS

Register bits of PIR1 to PIR4 registers indicates the source of the interrupt. The interrupt must be enabled when the appropriate bit is set to ‘1’ in the corresponding PIEx register. MRF24XA automatically clears the contents of the PIRX registers when the host MCU reads the content of the register. MCU must store the PIRX register values in the firmware as needed.

REGISTER 3-8: PIR1 (PERIPHERAL INTERRUPT REGISTER 1) ADDRESS: 0x04

<table>
<thead>
<tr>
<th>R/HS-1</th>
<th>R-0</th>
<th>R/HS-0</th>
<th>R/W/HC-0</th>
<th>R-0</th>
<th>R/W/HS-0</th>
<th>R/W/HS-0</th>
<th>R-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>VREGIF</td>
<td>r</td>
<td>RDYIF</td>
<td>IDLEIF</td>
<td>r</td>
<td>CALSOIF</td>
<td>CALHAIF</td>
<td>r</td>
</tr>
</tbody>
</table>

Legend:  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as ‘0’  
- = Value at POR  
‘1’ = Bit is set  
‘0’ = Bit is cleared  
x = Bit is unknown  
r = Reserved  
HC = Hardware Clear  
HS = Hardware Set

bit 7  
VREGIF: Voltage regulator On Interrupt Flag bit(1)  
This is a nonpersistent bit. The register bit initializes to 1 on 1.2V reset except when RESET is used and only cleared when PIR1 is read. Note that the corresponding IE bit is not implemented.

bit 6  
Reserved: Maintain as ‘0’

bit 5  
RDYIF: Ready state Interrupt Flag bit  
Set each time when READY state is reached:  
• When Calibration ended (CALST = 0)  
• When initialization ended (INITDONEF = 1)  
• When crystal is ramped up (XTALSF = 1)  
This bit is cleared when PIR1 is read.

bit 4  
IDLEIF: Idle state Interrupt Flag bit  
Set each time the IDLESF is set and if MCU did not trigger this change. This is unchanged when MCU aborts an action by clearing either of TXST, TXENC, RXDEC, EDST or CCA bits. This bit is cleared when PIR1 is read.

bit 3  
Reserved: Maintain as ‘0’

bit 2  
CALSOIF: Calibration Soft Interrupt Flag bit  
This flag indicates that maybe Calibration is needed (CALST) although the radio is still functional. It also warns of a possible degradation in signal quality and current consumption, and a risk of CALHAIF interrupt. This bit is cleared when PIR1 is read.

bit 1  
CALHAIF: Calibration Hard Interrupt Flag bit  
This flag indicates that immediate Calibration (CALST) is mandatory, otherwise the radio is not functional. The device enters into malfunction state. This bit is cleared when PIR1 is read.

bit 0  
Reserved: Maintain as ‘0’

Note 1: Generated non-maskable interrupt is gated off until the 1.2V reset is released.
REGISTER 3-9: PIR2 (PERIPHERAL INTERRUPT REGISTER 2)  ADDRESS: 0x05

<table>
<thead>
<tr>
<th>bit 7</th>
<th>bit 6</th>
<th>bit 5</th>
<th>bit 4</th>
<th>bit 3</th>
<th>bit 2</th>
<th>bit 1</th>
<th>bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>TXIF</td>
<td>TXENCIF</td>
<td>TXMAIF</td>
<td>TXACKIF</td>
<td>TXCSMAIF</td>
<td>TXSZIF</td>
<td>TXOVFIF</td>
<td>FRMIF</td>
</tr>
</tbody>
</table>

Legend:  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
-n = Value at POR  
'1' = Bit is set  
'0' = Bit is cleared  
x = Bit is unknown  
r = Reserved

bit 7  
**TXIF**: Transmission Done Interrupt Flag bit  
The current TX operation (TXST) is successfully completed. This event becomes unchanged when a hardware generated ACK packet completed the transmission or when a packet is repeated. Nonpersistent, cleared by SPI read.

bit 6  
**TXENCIF**: Transmit Encoding Interrupt Flag bit  
The TX packet was successfully encrypted or complemented, or both with a Message Integrity Code (MIC). Set by the device after TXENC = 1, when TXENC is cleared. Nonpersistent, cleared by SPI read.

bit 5  
**TXMAIF**: Transmitter Medium Access Interrupt Flag bit  
Set by the device when the medium is accessed, specifically when the first sample in the preamble is transmitted into the air. Nonpersistent, cleared by SPI read.

bit 4  
**TXACKIF**: Transmission Unacknowledged Failure Interrupt Flag bit  
Set by the device when Acknowledge is not received after the configured maximum number of transmission retries RETXMCNT<3:0>, provided that the Frame Control field of the transmitted frame indicates AckReq = 1 and AUTOACKEN = 1. Nonpersistent, cleared by SPI read.

bit 3  
**TXCSMAIF**: Transmitter CSMA Failure Interrupt Flag bit  
Set by the device when CSMA-CA finds the channel busy for BOMCNT<2:0> number of times, provided that CSMAEN = 1 is configured. Nonpersistent, cleared by SPI read.

bit 2  
**TXSZIF**: Transmit Packet Size Error Interrupt Flag bit  
Following TXST is set the packet size (including MIC tags and CRC) is found to be zero or to be greater than the maximum size that the buffers can support. Nonpersistent, cleared by SPI read.

bit 1  
**TXOVFIF**: Transmitter Overflow Interrupt Flag bit  
The Host Controller attempted to write a TX buffer that was not empty (TXBUFEMPTY = 0). Nonpersistent, cleared by SPI read.

bit 0  
**FRMIF**: Frame Format Error Interrupt Flag bit  
Set if the transmitter/receiver fails to parse the frame in the buffer (it is not as it must be or it is corrupted in demodulation). Nonpersistent, cleared by SPI read.
### REGISTER 3-10: PIR3 (PERIPHERAL INTERRUPT REGISTER 3)  
**ADDRESS:** 0x06

<table>
<thead>
<tr>
<th>Bit</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>RXIF</td>
<td>Received Successful Interrupt Flag bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Set by the device when a frame passed packet filtering and accepted, refer to Register 5-1. This interrupt flag is only set once for a packet and is not set when the packet is the duplicate of a repeated transmission (sequence number matches with the previously received frame). Nonpersistent, cleared by SPI read.</td>
</tr>
<tr>
<td>6</td>
<td>RXDECIF</td>
<td>Receiver Decryption/Authentication Passed Interrupt Flag bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Set by the device when decryption/authentication finished without error. Nonpersistent, cleared by SPI read.</td>
</tr>
<tr>
<td>5</td>
<td>RXTAGIF</td>
<td>Receiver Decryption/Authentication Failure Interrupt Flag bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Set by the device when decryption/authentication finished with error. Nonpersistent, cleared by SPI read.</td>
</tr>
<tr>
<td>4</td>
<td>Reserved</td>
<td>Maintain as ‘0’</td>
</tr>
<tr>
<td>3</td>
<td>RXIDENTIF</td>
<td>Received Packet Identical Interrupt Flag bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Set by the device when the packet is the duplicate of a repeated transmission (sequence number, source address matches with the previously received frame). Nonpersistent, cleared by SPI read.</td>
</tr>
<tr>
<td>2</td>
<td>RXFLTIF</td>
<td>Received Packet Filtered Interrupt Flag bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Set by the device when a packet was received, but rejected by one or more RX filters, refer to Register 5-1. Nonpersistent, cleared by SPI read.</td>
</tr>
<tr>
<td>1</td>
<td>RXOVFIF</td>
<td>Receiver Overflow Error Interrupt Flag bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Set by the device to indicate that a packet was received, but all RX buffers were full. Consequently the packet was not received, but was discarded instead(1). Nonpersistent, cleared by SPI read.</td>
</tr>
<tr>
<td>0</td>
<td>STRMIF</td>
<td>Receive Stream Time-Out Error Interrupt Flag bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Set by the device to indicate the duration specified in STRMTO elapsed since the last received packet while in RX-Streaming mode, and the MAC clears the stored sequence number. Nonpersistent, cleared by SPI read.</td>
</tr>
</tbody>
</table>

**Legend:**  
- R = Readable bit  
- W = Writable bit  
- U = Unimplemented bit, read as ‘0’  
- -n = Value at POR  
- ‘1’ = Bit is set  
- ‘0’ = Bit is cleared  
- x = Bit is unknown  
- r = Reserved  
- HC = Hardware Clear  
- HS = Hardware Set

**Note 1:**  
In Packet mode, use a single buffer for received frames. In RX-Streaming mode, use both buffers for reception.
REGISTER 3-11: PIR4 (PERIPHERAL INTERRUPT REGISTER 4)  ADDRESS: 0x07

<table>
<thead>
<tr>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>TXSFDIF</td>
<td>RXSFDIF</td>
<td>ERRORIF</td>
<td>WARNIF</td>
<td>EDCCAIF</td>
<td>GPIO2IF</td>
<td>GPIO1IF</td>
<td>GPIO0IF</td>
</tr>
</tbody>
</table>

Legend:  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
-n = Value at POR  
'1' = Bit is set  
'0' = Bit is cleared  
x = Bit is unknown  
r = Reserved  
HC = Hardware Clear  
HS = Hardware Set

bit 7  TXSFDIF: Transmit SFD Sent Interrupt Flag bit  
Set by the device when the last sample of the SFD field is sent into the air.  
Nonpersistent, cleared by SPI read.

bit 6  RXSFDIF: Receive SFD Detected Interrupt Flag bit  
Set by the device when the SFD field of the received frame is detected\(^1\).  
Nonpersistent, cleared by SPI read.

bit 5  ERRORIF: General Error Interrupt Flag bit  
Set by the device when a malfunction state is reached.

bit 4  WARNIF: Warning Interrupt Flag bit  
Set by the device when one of the following is occurred:
• Battery voltage drops below the threshold by BATMON<4:0> at 0x3F  
• Indicating that resistor is missing or not connected well

bit 3  EDCCAIF: Energy Detect/CCA Done Interrupt Flag bit  
Set by the device when Energy-detect or CCA measurement is complete (following that the host MCU sets the EDST/CCAST bit to start the measurement and the device is clearing it in on completion).  
Nonpersistent. Cleared by SPI read.

bit 2  GPIO2IF: GPIO2 Interrupt Flag bit  
Set by the device if the GPIOMODE register is set to normal operation, the GPIO is enabled and configured to input and the level matches with the polarity.

bit 1  GPIO1IF: GPIO1 Interrupt Flag bit  
Set by the device if the GPIOMODE register is set to normal operation, the GPIO is enabled and configured to input and the level matches with the polarity.

bit 0  GPIO0IF: GPIO0 Interrupt Flag bit  
Set by the device if the GPIOMODE register is set to normal operation, the GPIO is enabled and configured to input and the level matches with the polarity.

Note 1:  The detection latency (0…1 µs after the last sample of the SFD). Note that the SFD may trigger on noise or interference. Note that the CFOMEAS<7:0> indication becomes valid when RXSFDIF is asserted.

TABLE 3-2: REGISTERS ASSOCIATED WITH INTERRUPTS

<table>
<thead>
<tr>
<th>Name</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIR1</td>
<td>VREGIF</td>
<td>r</td>
<td>RDYIF</td>
<td>IDLEIF</td>
<td>r</td>
<td>CALSOIF</td>
<td>CALHAIF</td>
<td>r</td>
</tr>
<tr>
<td>PIR2</td>
<td>TXIF</td>
<td>TXENCIF</td>
<td>TXMAIF</td>
<td>TXACKIF</td>
<td>TXCSMAIF</td>
<td>TXS2IF</td>
<td>TXOVFIF</td>
<td>FRMIF</td>
</tr>
<tr>
<td>PIR3</td>
<td>RXIF</td>
<td>RXDECIF</td>
<td>RXTAGIF</td>
<td>r</td>
<td>RXIDENTIF</td>
<td>RXFLTIF</td>
<td>RXOVFIF</td>
<td>STRMIF</td>
</tr>
<tr>
<td>PIR4</td>
<td>TXSFDIF</td>
<td>RXSFDIF</td>
<td>ERRORIF</td>
<td>WARNIF</td>
<td>EDCCAIF</td>
<td>GPIO2IF</td>
<td>GPIO1IF</td>
<td>GPIO0IF</td>
</tr>
<tr>
<td>PIE1</td>
<td>r</td>
<td>RDYIE</td>
<td>IDLEIE</td>
<td>r</td>
<td>CALSOIE</td>
<td>CALHAIE</td>
<td>r</td>
<td></td>
</tr>
<tr>
<td>PIE2</td>
<td>TXIE</td>
<td>TXENCIE</td>
<td>TXMAIE</td>
<td>TXACKIE</td>
<td>TXCSMAIE</td>
<td>TXS2IE</td>
<td>TXOVFIE</td>
<td>FRMIE</td>
</tr>
<tr>
<td>PIE3</td>
<td>RXIE</td>
<td>RXDECIE</td>
<td>RXTAGIE</td>
<td>r</td>
<td>RXIDENTIE</td>
<td>RXFLTIE</td>
<td>RXOVFIE</td>
<td>STRMIE</td>
</tr>
<tr>
<td>PIE4</td>
<td>TXSFDIE</td>
<td>RXSFDIE</td>
<td>ERRORIE</td>
<td>WARNIE</td>
<td>EDCCAIE</td>
<td>GPIO2IE</td>
<td>GPIO1IE</td>
<td>GPIO0IE</td>
</tr>
<tr>
<td>PINCON</td>
<td>r</td>
<td>GIE</td>
<td>r</td>
<td>IRQIF</td>
<td>GPIOMODE&lt;3:0&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend:  
r = Reserved.
3.3 GPIO Functions and GPIO Interrupts

MRF24XA has three GPIO pins, GPIO2 pin 12, GPIO1 pin 11 and GPIO0 pin 10. GPIO pins are used as general purpose IO pins or GPIOs can monitor internal states.

Refer to Register 3-17 for more information on GPIO monitoring.

3.3.1 GPIO GENERAL IO FUNCTIONALITIES

To operate MRF24XA GPIOx pins in general purpose IO mode, set GPIOEN bit (0x0D<7>) to ‘1’ and set GPIOOMODE<3:0> bits (0x0C<3:0>) to ‘0000’.

The TRISGPIOx bits (0x0D<6:4>) configures the input or output selection of GPIOs. Clearing the TRISGPIOx bit sets the appropriate GPIO line to Output mode. Input the default GPIO line direction after POR.

GPIO lines in Input mode are used with Schmitt Trigger input buffers. STENGPIOx bits (0x0E<2:0>) enables the Schmitt Triggers. Setting the STENGPIOx bit to ‘1’ enables Schmitt Trigger input of the appropriate pin.

GPIO data is read or written to through the GPIO bits (0x0D<2:0>).

GPIO lines can have active pull-up or pull-down. PULLENGPIOx (0x0F<2:0>) bits enable line pulling function. Setting PULLENGPIOx bit to ‘1’ enables active pull-up or pull-down circuit. PULLDIRGPIOx bit (0x0F<6:4>) sets the pull direction. Setting PULLDIRGPIOx bit to ‘1’ defines pull-up, while clearing the bit defines pull-down on the appropriate GPIO line.

3.3.2 GPIO INTERRUPT HANDLING

GPIO lines can also generate interrupts. To use GPIO interrupts, set the appropriate GPIOxIE bit (0x0B<2:0>) to ‘1’ to enable the interrupt generation. To enable interrupt generation on INT pin, set GIE bit (0x0C<6>) to ‘1’.

The GPIO interrupt polarity is selected through GPIOxP bits (0x0E<6:4>). Setting GPIOxP bit to ‘1’ triggers interrupt logic at the rising edge of the input signal. While clearing the bit enables interrupt generation on the falling edge of the input pin.
REGISTER 3-12:  PIR4 (PERIPHERAL INTERRUPT REGISTER 4)  ADDRESS: 0x07

<table>
<thead>
<tr>
<th>bit 7-3</th>
<th>Out of scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 2</td>
<td>GPIO2IF: GPIO2 Interrupt Flag bit</td>
</tr>
<tr>
<td></td>
<td>Set by the device if the GPIOMODE register is set to normal operation, the GPIO is enabled and configured to input and the level matches with the polarity.</td>
</tr>
<tr>
<td>bit 1</td>
<td>GPIO1IF: GPIO1 Interrupt Flag bit</td>
</tr>
<tr>
<td></td>
<td>Set by the device if the GPIOMODE register is set to normal operation, the GPIO is enabled and configured to input and the level matches with the polarity.</td>
</tr>
<tr>
<td>bit 0</td>
<td>GPIO0IF: GPIO0 Interrupt Flag bit</td>
</tr>
<tr>
<td></td>
<td>Set by the device if the GPIOMODE register is set to normal operation, the GPIO is enabled and configured to input and the level matches with the polarity.</td>
</tr>
</tbody>
</table>

REGISTER 3-13:  PIE4 (PERIPHERAL INTERRUPT ENABLE 4)  ADDRESS: 0x0B

<table>
<thead>
<tr>
<th>bit 7-3</th>
<th>Out of scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 2</td>
<td>GPIO2IE: GPIO2 Interrupt Enable bit</td>
</tr>
<tr>
<td></td>
<td>This bit masks the GPIO2IF interrupt register.</td>
</tr>
<tr>
<td>bit 1</td>
<td>GPIO1IE: GPIO1 Interrupt Enable bit</td>
</tr>
<tr>
<td></td>
<td>This bit masks the GPIO1IF interrupt register.</td>
</tr>
<tr>
<td>bit 0</td>
<td>GPIO0IE: GPIO0 Interrupt Enable bit</td>
</tr>
<tr>
<td></td>
<td>This bit masks the GPIO0IF interrupt register.</td>
</tr>
</tbody>
</table>
### REGISTER 3-14: GPIO (GENERAL PURPOSE I/O REGISTER)

ADDRESS: **0x0D**

<table>
<thead>
<tr>
<th>Bit</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>GPIOEN</td>
<td>GPIO Enable bit. This bit enables the GPIO's control, if GPIOMODE is only configured into Normal mode. The other GPIOMODE Configuration automatically controls GPIO pins.</td>
</tr>
<tr>
<td>6</td>
<td>TRISGPIO2</td>
<td>Tri-state Control for GPIO 2 Pin bit. If set, the pin is configured into Input mode. Value reads from GPIO2 bit. If cleared, the pin is configured into Output mode. Value sets through the GPIO2 bit.</td>
</tr>
<tr>
<td>5</td>
<td>TRISGPIO1</td>
<td>Tri-state Control for GPIO 1 Pin bit. If set, the pin is configured into Input mode. Value reads from GPIO1 bit. If cleared, the pin is configured into Output mode. Value sets through the GPIO1 bit.</td>
</tr>
<tr>
<td>4</td>
<td>TRISGPIO0</td>
<td>Tri-state Control for GPIO 0 Pin bit. If set, the pin is configured into Input mode. Value reads from GPIO0 bit. If cleared, the pin is configured into Output mode. Value sets through the GPIO0 bit.</td>
</tr>
<tr>
<td>3</td>
<td>Reserved</td>
<td>Maintain as '0'.</td>
</tr>
<tr>
<td>2</td>
<td>GPIO2</td>
<td>GPIO 2 Value bit. This bit represents the value on the GPIO 2 pin.</td>
</tr>
<tr>
<td>1</td>
<td>GPIO1</td>
<td>GPIO 1 Value bit. This bit represents the value on the GPIO 1 pin.</td>
</tr>
<tr>
<td>0</td>
<td>GPIO0</td>
<td>GPIO 0 Value bit. This bit represents the value on the GPIO 0 pin.</td>
</tr>
</tbody>
</table>

Legend:
- **R** = Readable bit
- **W** = Writable bit
- **U** = Unimplemented bit, read as '0'
- **-n** = Value at POR
- **'1'** = Bit is set
- **'0'** = Bit is cleared
- **x** = Bit is unknown
- **r** = Reserved

**Example:**
- GPIO2: Setting GPIO2 to '1' will configure the pin into Input mode.
- GPIO1: Setting GPIO1 to '0' will configure the pin into Output mode.
### REGISTER 3-15: STGPIO (SCHMITT TRIGGER GENERAL PURPOSE I/O REGISTER)

**ADDRESS:** 0x0E

<table>
<thead>
<tr>
<th>bit 7</th>
<th>bit 6</th>
<th>bit 5</th>
<th>bit 4</th>
<th>bit 3</th>
<th>bit 2</th>
<th>bit 1</th>
<th>bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
</tr>
<tr>
<td>r</td>
<td>GPIO2P</td>
<td>GPIO1P</td>
<td>GPIO0P</td>
<td>r</td>
<td>STENGPIO2</td>
<td>STENGPIO1</td>
<td>STENGPIO0</td>
</tr>
</tbody>
</table>

**Legend:**
- **R** = Readable bit
- **W** = Writable bit
- **U** = Unimplemented bit, read as ‘0’
- -n = Value at POR
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- x = Bit is unknown

- **r** = Reserved

**bit 7**
- **Reserved:** Maintain as ‘0’

**bit 6**
- **GPIO2P:** GPIO 2 Polarity bit
  - This bit controls GPIO2IF polarity when configured into Input mode.
  - 1 = Rising edge
  - 0 = Falling edge

**bit 5**
- **GPIO1P:** GPIO 1 Polarity bit
  - This bit controls GPIO1IF polarity when configured into Input mode.
  - 1 = Rising edge
  - 0 = Falling edge

**bit 4**
- **GPIO0P:** GPIO 0 Polarity bit
  - This bit controls GPIO0IF polarity when configured into Input mode.
  - 1 = Rising edge
  - 0 = Falling edge

**bit 3**
- **Reserved:** Maintain as ‘0’

**bit 2**
- **STENGPIO2:** Schmitt Trigger Enable GPIO 2
  - This bit enables Schmitt-trigger circuit on GPIO 2 pad and turns off by default.
  - 1 = Schmitt trigger enabled
  - 0 = Schmitt trigger disabled

**bit 1**
- **STENGPIO1:** Schmitt Trigger Enable GPIO 1
  - This bit enables Schmitt-trigger circuit on GPIO 1 pad and turns off by default.
  - 1 = Schmitt trigger enabled
  - 0 = Schmitt trigger disabled

**bit 0**
- **STENGPIO0:** Schmitt Trigger Enable GPIO 0
  - This bit enables Schmitt-trigger circuit on GPIO 0 pad and turns off by default.
  - 1 = Schmitt trigger enabled
  - 0 = Schmitt trigger disabled
MRF24XA

REGISTER 3-16:  PULLGPIO (PULL CONTROL GENERAL PURPOSE I/O REGISTER)

ADDRESS: 0x0F

<table>
<thead>
<tr>
<th>R-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>PULLDIRGPIO2</td>
<td>PULLDIRGPIO1</td>
<td>PULLDIRGPIO0</td>
<td>r</td>
<td>PULLENGPIO2</td>
<td>PULLENGPIO1</td>
<td>PULLENGPIO0</td>
</tr>
</tbody>
</table>

bit 7  Reserved: Maintain as ‘0’

bit 6  PULLDIRGPIO2: Pull Direction on GPIO 2 bit
These bits control the 75 kOhm weak-pull circuit direction on GPIO 2 pin.
1 = Pull-up
0 = Pull-down

bit 5  PULLDIRGPIO1: Pull Direction on GPIO 1 bit
These bits control the 75 kOhm weak-pull circuit direction on GPIO 1 pin.
1 = Pull-up
0 = Pull-down

bit 4  PULLDIRGPIO0: Pull Direction on GPIO 0 bit
These bits control the 75 kOhm weak-pull circuit direction on GPIO 0 pin.
1 = Pull-up
0 = Pull-down

bit 3  Reserved: Maintain as ‘0’

bit 2  PULLENGPIO2: Pull Enable on GPIO 2 bit
This bit enables to weak-pull circuit in GPIO 2 pin. Note that when pin is configured to output, weak-pull circuit automatically disables.
1 = Pull enabled
0 = Pull disabled

bit 1  PULLENGPIO1: Pull Enable on GPIO 1 bit
This bit enables to weak-pull circuit in GPIO 1 pin. Note that when pin is configured to output, weak-pull circuit automatically disables.
1 = Pull enabled
0 = Pull disabled

bit 0  PULLENGPIO0: Pull Enable on GPIO 0 bit
This bit enables to weak-pull circuit in GPIO 0 pin. Note that when pin is configured to output, weak-pull circuit automatically disables.
1 = Pull enabled
0 = Pull disabled
## REGISTER 3-17:  PINCON (PIN CONFIGURATION REGISTER)

<table>
<thead>
<tr>
<th>Address</th>
<th>Address</th>
<th>R-0</th>
<th>R/W-1</th>
<th>R-0</th>
<th>R-1</th>
<th>R/W-0000</th>
<th>GPIOMODE&lt;3:0&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 7</th>
<th>bit 6</th>
<th>bit 5</th>
<th>bit 4</th>
<th>bit 3-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>GIE</td>
<td>IRQIF</td>
<td>GPIOMODE&lt;3:0&gt;</td>
<td>GPIOMODE&lt;3:0&gt;</td>
<td>GPIOMODE&lt;3:0&gt;</td>
</tr>
</tbody>
</table>

### Legend:
- **R** = Readable bit  
- **W** = Writable bit  
- **U** = Unimplemented bit, read as ‘0’  
- **-n** = Value at POR  
- **‘1’** = Bit is set  
- **‘0’** = Bit is cleared  
- **x** = Bit is unknown  
- **r** = Reserved

**bit 7**  
Reserved: Maintain as ‘0’

**bit 6**  
GIE: General Interrupt Enable bit  
This bit enables to output IRQIF on INT pin. Note that the polarity of INT pin is active-low.

**bit 5**  
Reserved: Maintain as ‘0’

**bit 4**  
IRQIF: Interrupt Request Pending bit  
This bit is the OR relationship of the enabled interrupt flags.

**bit 3-0**  
GPIOMODE<3:0>: GPIO Mode Field bit  
This field allows redefining the functionality of the GPIO pins Encoding:

- **11xx** = Reserved
- **1011** = GPIO pins are used for Receive streaming (RXSTREAM). Pins GPIO<2:0> are used to output \{RXWRBUF, BUSRDBUF, RXBUFFUL\}.
- **1010** = GPIO pins are used for Transmit streaming (TXSTREAM). Pins GPIO<2:0> are used to output \{TXRDBUF, BUSWRBUF, TXBUFEMPTY\}.
- **1001** = Reserved
- **1000** = Reserved
- **0111** = Reserved
- **0110** = Reserved
- **0101** = Intended for supporting Precise Network Time Synchronization (TIMESYN). GPIO<0> is used to output TX, while GPIO<1> to output RX SFD indication pulses. GPIO<2> is used in “NORMAL” operation mode.
- **0100** = GPIO pins are used for Radio monitoring (RFMON). Pins GPIO<2:0> are used to output RFOP<2:0>.
- **0011** = GPIO pins are used for MAC monitoring (MACMON). Pins GPIO<2:0> are used to output MACOP<3:1>.
- **0010** = GPIO pins are used for RXFSM monitoring (RXFSMMON). Pins GPIO<2:0> are used to output receiver state-machine  
  - **000** = Preamble search  
  - **001** = Hi-rate SFD search  
  - **010** = Mid-rate SFD search  
  - **011** = Low-rate SFD search  
  - **100** = Legacy length field processing  
  - **101** = Payload processing
- **0001** = GPIO pins are used for AGC monitoring (AGCMON). Pins GPIO<2:0> are used to output \{AGCHOLD, GAIN<1:0>\} where AGCHOLD is an internal flag set when a receiver detects a preamble and clears when the AGC is set free after the end of the frame.
- **0000** = GPIO pins are used as General Purpose I/O’s by the host MCU (NORMAL).
MRF24XA

3.4 PA and LNA Outputs

MRF24XA has a Power Amplifier (PA) control pin (pin 20) and a Low Noise Amplifier (LNA) control pin (pin 21). These pins are capable of handling external PAs and LNAs or external antenna switch circuits. MRF24XA can also tolerate different start-up times of different external circuits by sending or accepting data if the external circuits completes their ramp up. MRF24XA can handle both active-high or active-low control signal sensitive circuits.

For more information, refer to Section 9.13 “External Power Amplifier (PA)/Low-Noise Amplifier (LNA)”.

3.5 Battery Monitor

The voltage level on the battery is monitored. If the battery monitoring is enabled and the voltage level drops below a threshold, voltage interrupt (WARNIF) is asserted. Refer to Register 3-18 for more information on the battery.

REGISTER 3-18: BATMON (BATTERY MONITOR CONFIGURATION REGISTER) ADDRESS: 0x3F

<table>
<thead>
<tr>
<th>Bit 7-6</th>
<th>Bit 5</th>
<th>Bit 4-0</th>
<th>R-0</th>
<th>R/W-1</th>
<th>R/W-11111</th>
</tr>
</thead>
<tbody>
<tr>
<td>BATMONPD</td>
<td>BATMON&lt;4:0&gt;</td>
<td>BATMONPD Bit</td>
<td>BATMON&lt;4:0&gt;</td>
<td>BATMONPD Bit</td>
<td></td>
</tr>
</tbody>
</table>

Legend:  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as ‘0’  
-n = Value at POR  
‘1’ = Bit is set  
‘0’ = Bit is cleared  
x = Bit is unknown

bit 7-6  
Reserved: Maintain as ‘0’

bit 5  
BATMONPD: Battery Monitor Power-Down bit

If battery monitor is working and battery voltage drops below the threshold by BATMON<4:0> then WARNIF is set.

1 = Battery monitor is OFF  
0 = Battery monitor is working

bit 4-0  
BATMON<4:0>: Battery Monitor Threshold Field bits

\[ V_{\text{THRESHOLD}} = 3.6 - 0.071 \times \text{BATMON<4:0>} \ (V) \]
4.0 GENERAL TRANSCEIVER OPERATIONS

4.1 MAC Architecture

The architecture of MAC layer processing is illustrated in Figure 4-1.

In reception, the receive signal processor acquires the synchronization header of the frame on-air, and demodulates the frame starting from the LENGTH field. The demodulated data is written directly into the Receiver Buffer (Default, Buffer 2) if the targeted buffer is declared empty (RXBUFFUL = 0). After LENGTH number of bytes are received into the buffer and RSV data are appended, the frame is parsed according to the selected Framing mode (IEEE 802.15.4 or proprietary). The Frame Control Sequence (FCS) is checked to detect corruption by noise. Corrupted frames or frames not addressed to this node are rejected (discarded), which the host configures. Rejection means that reception is completed now and the Receive Buffer status remains empty (RXBUFFUL = 0). It is configurable whether the frame is discarded silently or generates an interrupt to the host.

If a frame is accepted and Acknowledge is requested for the frame, then the radio turns to transmit and sends an Acknowledgment. As other features, automatic ACK-sending are enabled or bypassed. If the frame is the duplicate of a previously received and accepted frame then the frame is discarded (following Acknowledgment). Otherwise, the frame is the first copy of an accepted frame, which must be reported to the host. To lock the buffer from being overwritten by a new frame, RXBUFFUL is automatically set (1). RXIF interrupt is generated for the host, which completes the reception.

The host MCU can only access the Receive Buffer when RXBUFFUL is set (1). To free up the buffer, the host clears RXBUFFUL (0).

If the frame is encrypted or contains an authentication tag (MIC), the host MCU must run the decrypt/authenticate operation before it reads the payload and frees up the buffer.

When sending, the host MCU constructs the frame and downloads it to the transmit buffer (Default, Buffer 0), and triggers transmission after the last byte. The device processes the content of the buffer in-place. After parsing, a security processing takes place if required, finally an FCS is generated and appended to the frame. The LENGTH is adjusted each time an authentication tag (MIC) or FCS is appended to the frame.

After in-place frame processing the medium is accessed using the Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA). The RF transmit chain is only enabled when the channel is free, or if CSMA is bypassed. As soon as the RF can transmit, the Transmit Signal Processor starts sending the Synchronization Header (SHR) and followed by the buffer content up the FCS. If an Acknowledgment is requested then the RF chain is turned into receive. If ACK is not received before the expiration of a time-out, the transmission can automatically start over from CSMA through SHR-transmission, and then transmitting the SHR-transmission frame if configured. After successful sending an interrupt is generated to the host MCU. Only either the TX MAC or the RX MAC is active at a time.
4.2 Operations Overview

4.2.1 TERMINOLOGY
Node denotes the wireless communication node that is formed by a MRF24XA device and a host MCU. Device denotes the MRF24XA device. Software/SW denotes the software running in the Host MCU. The device does not contain a processor core that runs software. Frame and Packet are used interchangeably.

4.2.2 HOST INTERFACE
The host MCU controls the device over SPI (max. 10 MHz), whereas the device indicates task completion or failure events, and frames received over the air by raising an interrupt. Most interrupt flags are masked, which means that these are still set on the respective event, but cannot activate the interrupt pin on the device. The host services the interrupts through reading the interrupt register. The interrupt bytes are self-cleared on SPI-read. Using the convention, the “IF” suffix in mnemonics refers to “Interrupt Flag”. For example, TXIF and RXIF. For software troubleshooting, the host can set the interrupt flags.

4.2.3 BUFFERS
The device has two frame buffers (128 bytes each). SPI allows accessing each byte in the frame buffer at its own address. As the default, the buffer starting at address 0x200 is used for transmission (BUF1) and the buffer starting at address 0x300 is used for reception (BUF2).

4.2.4 OPERATING STATES
The Figure 4-3 chart describes the top-level state machine of MRF24XA, including the valid state transitions. Note that the register access can also perform the other transitions, however these may result in an unexpected behavior.

For more information on the MRF24XA Power modes on power consumption in each state, refer to Table 2-1.

4.2.4.1 Description of Each State
- RADIO IN PIN RESET: MCU keeps nRST pin low. Radio is in reset, functions are unavailable.
- POWER OFF: Voltage on 3V3 pin is 0V. Radio is shut down.
• INIT (RST = 1): Internal state. Radio performs automatic initialization after Power or Pin Reset. In this state, SPI is active and TX buffer can be pre-loaded. Register access is limited to 0x00 - 0x07.
• INIT (RST = 0): Internal state. Radio performs automatic initialization to recover from Deep Sleep. In this state, SPI is active and TX buffer can be pre-loaded. Register access is limited to 0x00 - 0x07.
• DEEP SLEEP: The radio is in an extremely low-power state. Current consumption is 40 nA. All parts, including the 1V2 on-chip regulator, is turned off. 3V3 backup memory is only powered to keep the register settings. To maintain the settings, the chip must not be powered down completely.
• INITIAL CALIBRATION: After reset, some internal circuits must be calibrated. The radio automatically performs these calibrations. In case some problem occurs during Calibration, the chip remains in this state.
• RADIO OFF: This is the first stable state after power cycle or pin reset. The radio is ready for all operations, every register is accessible. Even if crystal is running, it takes 50 µs for the synthesizer to ramp up before RX or TX mode gets enabled.
• IDLE: Radio is up and all registers are accessible. Crystal and synthesizer are up and running to reduce transition to RX or TX.
• SLEEP: Only the regulator is powered on, crystal is not running. This state may be used for pre-loading register values or TX buffer. No other functionality are available.
• RX Listen and RX Listen Power Save: Radio awaits for packets to arrive and to be received. All registers are accessible and TX buffer may be loaded. Even if PSAV is enabled, the operation remains the same. The difference displays in power consumption and sensitivity.
• RX FRAME: Temporary internal state for the actual packet reception process.
• TX FRAME: Temporary internal state for sending out a packet from TX buffer.

4.2.4.2 Detailed Transition Description
1: MCU releases nRST. Radio moves into INIT (RST = 1) mode. In case there were no power cycles, POR bit retains its state.
2: Radio gets powered on. POR bit sets to 1 and VREGIF is set when the 1V2 regulator is up. VREGIF is not maskable.
3: To start recovery from Deep Sleep mode, a dummy SPI read operation (at least four changes on SDI line) is required from the MCU. The first interrupt after wake-up is VREGIF, which indicates the start of the on-chip 1.2V regulator. The SPI is operational at this point that enables the MCU to service the interrupt by reading the interrupt source register. POR flag retains its state before Deep Sleep mode. Transmit buffers are accessible for preloads. After the crystal oscillator becomes stabilized (1-3 ms; 1,4 ms typical), the device sets the RDYIF interrupt without going through recalibration, and then all the registers become accessible. Previously stored Configuration is retained during Deep Sleep, thus Calibration is not necessary.
4: In case radio was started with any kind of reset, Calibration is needed. Transition from INIT state and the initial Calibration are automatically done.
5: Theoretically, it is possible that Calibration fails. In that case, the chip remains in this internal state. Further information in STATUS register description.
7: Radio automatically turns to OFF mode after the register values are restored and RX is disabled before Deep Sleep.
8: In case RX is enabled before Deep Sleep, radio automatically turns back to RX or RX Power Save mode.
9, 12: To initiate transition to Sleep mode, MCU must set XTALDIS bit. Note that Sleep mode is only accessible from Idle and Radio OFF mode.
10: To recover from Sleep mode MCU must clear XTALDIS bit. Afterwards crystal ramps up and radio turns back to OFF modes.
11: Transferring to Deep Sleep mode. MCU must clear POR bit and all interrupt registers and set DSLEEP bit afterwards.
13, 14: RX mode is accessible from Radio OFF and Idle mode. The only difference is in transition time: from OFF mode user must Wait for synthesizer to ramp up.
15: MCU clears RXEN bit to go back to Idle. For further information on transition, see check point 20.
16, 17 and 18: MCU sets TXST to transmit packet from buffer. The difference is in transition time: from OFF mode user must Wait for synthesizer to ramp up.
19, 21: Radio turns back to Idle mode (if started from Idle or OFF mode) after the packet is sent or the MCU clears the TXST bit. Transition to RX takes place if RXEN bit is set on high before setting TXST.
20: Transition depends on OFFTM register settings. When TXST or RXEN bits are cleared, radio waits OFFTM number of Base time units and then it turns to OFF mode to save power. To disable automatic transition, set OFFTM to 0xFF.
22: Automatic transition if radio detects a packet being transmitted over air.
23: Transition after packet reception is over or MCU clears RXEN bit.

**TABLE 4-1: TRANSITION TIMES**

<table>
<thead>
<tr>
<th>Starting State</th>
<th>Destination State</th>
<th>Radio internal tasks</th>
<th>Typ. Transition Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power OFF</td>
<td>Radio OFF</td>
<td>Regulator, XTAL ramp up and Calibration</td>
<td>1.6 ms&lt;sup&gt;(1)&lt;/sup&gt;</td>
</tr>
<tr>
<td>Pin Reset</td>
<td>Radio OFF</td>
<td>XTAL ramp up, Calibration</td>
<td>1.4 ms&lt;sup&gt;(1)&lt;/sup&gt;</td>
</tr>
<tr>
<td>Radio OFF, IDLE, RX</td>
<td>DEEP SLEEP</td>
<td>Save registers, shut down 1V2 LDO</td>
<td>45 ms&lt;sup&gt;(2)&lt;/sup&gt;</td>
</tr>
<tr>
<td>DEEP SLEEP</td>
<td>Radio OFF</td>
<td>Regulator and XTAL ramp up, register recovery</td>
<td>1.6 ms&lt;sup&gt;(1)&lt;/sup&gt;</td>
</tr>
<tr>
<td>DEEP SLEEP</td>
<td>RX</td>
<td>Regulator and XTAL ramp up, register recovery, synthesizer on</td>
<td>1.65 ms&lt;sup&gt;(1)&lt;/sup&gt;</td>
</tr>
<tr>
<td>Radio OFF</td>
<td>RX Listen</td>
<td>Synthesizer ramp up, Power on RX front end</td>
<td>57 µs</td>
</tr>
<tr>
<td>Radio OFF</td>
<td>TX Frame</td>
<td>Synthesizer ramp up, Power on TX front end</td>
<td>52 µs</td>
</tr>
<tr>
<td>IDLE</td>
<td>RX Listen</td>
<td>Power on RX front end</td>
<td>6.8 µs</td>
</tr>
<tr>
<td>IDLE</td>
<td>TX Frame</td>
<td>Power on TX front end</td>
<td>1.3 µs</td>
</tr>
<tr>
<td>Radio OFF</td>
<td>SLEEP</td>
<td>Crystal power off</td>
<td>25 µs</td>
</tr>
<tr>
<td>SLEEP</td>
<td>Radio OFF</td>
<td>Crystal ramp up</td>
<td>1.4 ms&lt;sup&gt;(1)&lt;/sup&gt;</td>
</tr>
</tbody>
</table>

**Note 1:** XTAL ramp up time is depending on temperature. Minimum values 1 ms, typical value 1.4 ms, maximum value 3 ms.

**Note 2:** This time is referring to the complete discharge time on the internally regulated 1V2 bus and may depend on the actual capacitance on this network.

**FIGURE 4-2:** TRANSITION FROM DEEP SLEEP TO RX LISTEN POWER-SAVE MODE
FIGURE 4-3: MRF24XA TOP LEVEL STATE MACHINE (SIMPLIFIES STATE TRANSITION CHART)

Note: See Section 4.2.4.2 “Detailed Transition Description” for details on the interrupts in connection with state transitions.
Frame sending/reception involves the following steps as shown in the Figure 4-4. (Note: Sending = originator, Receiving = recipient)

1. Originator and Recipient Nodes apply the previously shared (negotiated/global) configurations.
2. Recipient MCU enables reception in the device when the RXEN (1) control bit in the RXCON1 register is set.
3. Originator MCU constructs the payload.
4. Originator MCU constructs the MAC frame header applying the per frame configurations.
5. Originator MCU loads the MAC frame to the device.
6. Originator MCU starts transmit operation when the TXST (1) control bit (MACCON1 register) in the device is set. MAC layer encryption is automatically applied while for network layer encryption
7. Originator device executes in-place processing on the frame. For example, encryption and FCS appending, and checks the Configuration of the frame header that affects the per frame device behavior for this frame (whether an Acknowledge is requested from the recipient).
8. Originator device attempts to send the message to the receiver device. Prior accessing the medium, ensure that no other device is using it on the same channel frequency, or jammed by interferers before sending. The applied procedure is called CSMA-CA. When the channel is clear, it sends the frame. Finally, the device waits for an Acknowledge as the frame is configured to request one.
9. Recipient device receives the frame and parses it. The frame is accepted. The frame can also be rejected due to destination address or FCS mismatch.
10. Recipient device sends an ACK.
11. Recipient device generates an RXIF (1) interrupt to its host MCU when ACK-sending is complete. Since the frame is accepted and acknowledged, RXBUFFUL is set (1) by the device. This protects the frame from being overwritten by a subsequent different frame.
12. Recipient MCU may trigger in-place processing (for example, decryption by setting RXDEC) on the frame after servicing the interrupt.
13. Originator device fails to receive the ACK-frame. Therefore, it starts over transmitting the same frame (retransmission) through CSMA-CA first and then sending follows.
14. Recipient device receives the retransmitted frame and finds out it is a duplicate. It still sends another ACK-frame to it, but discards the duplicate frame.
15. Originator device receives the ACK and generates a TXIF (1) interrupt to confirm the successful sending to its host MCU. It also clears the TXST (0) control/Status bit. Originator device returns to Reception mode if RXEN = 1 is configured, otherwise it turns to TRXOFF.
16. MCU services the interrupt (TXIF) to learn the confirmation. Transaction is successfully completed.
17. Recipient device completes the in-place decryption of the frame and generates a RXDECIF (1) interrupt to indicate status to its host MCU.
18. Recipient MCU reads the decrypted frame from the buffer and unlocks the buffer when the control/status flag RXBUFFUL (0) of the device is cleared.
In-place processing (whether transmitting or receiving-side) is tested using a single node. This is useful in device testing or software troubleshooting. Figure 4-5 illustrates the procedure of in-place test using a single node. Note that the Originator and the Recipient node can be the same hardware.

TXBUF1PP triggers the processing that TXST originally performs. TXIF interrupt is generated at the end of in-place processing, without attempting to physically send the frame. The device clears TXBUF1PP at the same time.
Buffer 1 holds the processed (encrypted, FCS-appended) frame. If RXBUF1PP is set, frame filtering is performed and RXIF interrupt is generated. Alternatively, the processing is performed in Buffer 2 (the normal receive buffer) using TXBUF2PP and RXBUF2PP. Note that unlike TXBUF1PP and TXBUF2PP, RXBUF1PP and RXBUF2PP do not automatically clear when RXIF is set but select which buffer must be processed when RXDEC is issued.

**FIGURE 4-5: IN-PLACE TEST USING A SINGLE NODE: EXAMPLE SCENARIO (MESSAGE SEQUENCE CHART)**
4.3 Global vs. Per Node vs. Per Packet Configurations

For certain configurable parameters, the selected options apply to all nodes in a network. The following are the global attributes:
- The MAC frame format used in the network (FRMFMT register bit configuration)
- The FCS appending and checking method (refer to CRCSZ register bit)
- The medium access (CSMAEN register bit configuration)

4.3.1 GLOBAL CONFIGURATIONS

Global attributes are negotiated through management information travelling in the MAC payload (of the current frame or a previous frame). For example,
- Address size in Proprietary MAC framing mode
- Network Layer security enabling and security material in IEEE 802.15.4-compliant MAC framing mode

These global attributes are shared between all nodes of the network as all the nodes can access the same medium, and the sending and recipient sides must process the frame consistently.

4.3.2 PER PACKET CONFIGURATIONS

As opposed to global configurations, per packet attributes vary from packet to packet. For example,
- Acknowledge Requested for the current frame (AckReq)
- Security Processing enabled for the current frame (SecEn)

These attributes must be shared between the originator and the recipient of the frame and must travel with the packet. Prior sending a frame, the originator MCU applies the desired attributes when the respective frame control bits (for example, AckReq, SecEn) in the MAC header of the frame are configured. When the send operation is triggered, the sending device checks these attributes and adapts its (frame processing and sending) behavior accordingly. The receiver device does the same on reception.

4.3.3 PER NODE CONFIGURATION

It applies to specific nodes in the network, having a specific role. The auto-repeater functionality (in proprietary MAC mode) is a typical example. Another case is how a node filters frames. A sniffer node must have different Configuration than an ordinary node. This Configuration may also be different since the sniffer must not send acknowledge.

4.4 Features Overview

The device supports two Framing modes:
- IEEE 802-15.4 standard compliant, see Section 5.0 “IEEE 802.15.4™ Compliant Frame Format and Frame Processing” format
- Proprietary format, see Section 6.0 “Proprietary Frame Format and Frame Processing” format

Hardware support is provided for both the features, but a network must only use one of these in all the nodes. A compromise is offered through Bridging, see Section 8.0 “Bridging”.

For the discussion it is helpful to distinguish between Protocol Agnostic and Protocol Dependent.

Configuration Options:
- The availability and Configuration format of Protocol Dependent options are conditioned on the selected framing protocol whether IEEE 802.15.4-compliant or proprietary network operation is required.
- In contrast, the availability, behavior and Configuration format of other options are Protocol Agnostic from a device point of view, which means that the Configuration occurs similarly for framing protocol.

All Protocol Agnostic options can freely combined with any of the “protocol-dependent” configurations as far as the device constrains it. To comply with the IEEE 802.15.4 protocol, the constraints specified in the standards must be followed, see Table 4-3.

Figure 4-6 lists the higher level (MAC layer) Configuration features.

- FCS method: Hardware supports the 2-byte long CRC sequence adopted by IEEE 802-15.4. This must be adequate for most applications. In the contrary case, the CRC appending and checking are disabled if CRCSZ = 0. If CRCSZ = 0 then AUTOACKEN = 0 and RXFILTER = 0x00 is required. CRCSZ = 1 is assumed in the discussion.
- CSMA is described in Section 4.11 “Carrier Sense Multiple Access-Collision Avoidance (CSMA-CA)” and only requires Packet mode.
- Automatic Acknowledgement Reception and Sending is configured as specified in Section 4.12 “Clear Channel Assessment (CCA)”, Section 4.14 “Acknowledge Sending by Recipient” and Section 4.15 “Acknowledge Reception by Originator”.

© 2015 Microchip Technology Inc. Preliminary DS70005023C-page 97
• In both framing configurations, the device offers support for: frame parsing, frame filtering, frame types, addressing modes applicable to multi-cast and uni-cast frames and security processing. Standard mode operation is described in Section 5.0 “IEEE 802.15.4™ Compliant Frame Format and Frame Processing”. Proprietary mode is described in Section 6.0 “Proprietary Frame Format and Frame Processing” through Section 7.0 “Advanced Link Behavior in Proprietary Packet Mode”.

• The following modes are only available in Proprietary modes: Bridging, Link Agility, see Section 7.1 “Channel Agility”, and Auto-Repeater, see Section 7.3 “Auto-Repeater”.

Table 4-2 lists a summary of possible node behaviors and node types. Section 5.0 “IEEE 802.15.4™ Compliant Frame Format and Frame Processing” focuses on the general processing of IEEE 802.15.4-compliant and proprietary-format non-streaming frames when the node is non-streaming configured in Packet mode. Differences to this behavior are specified for the other scenarios in the respective sections.

![FIGURE 4-6: MAC LAYER CONFIGURATION OVERVIEW](image-url)

<table>
<thead>
<tr>
<th>COMMUNICATION ASPECTS</th>
<th>AVAILABLE OPTIONS</th>
<th>PER PACKET or PER NODE or GLOBAL?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PROTOCOLAGNOSTIC:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Frame Error Detection</td>
<td>Trailer FCS by HW: yes/no?</td>
<td>Global option (typically, but not enforced)</td>
</tr>
<tr>
<td>Link Reliability</td>
<td>Acknowledge Request: yes/no? (if yes, # of retransmissions?)</td>
<td>Per packet option</td>
</tr>
<tr>
<td></td>
<td>Automatic ACK response: yes/no?</td>
<td>Per node option</td>
</tr>
<tr>
<td>Multiple Access</td>
<td>CSMA: yes/no?</td>
<td>Per packet option</td>
</tr>
<tr>
<td><strong>PROTOCOLDEPENDENT:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Frame Parsing</td>
<td>IEEE 802.15.4 vs. Proprietary?</td>
<td>Global option (except for „bridging”)</td>
</tr>
<tr>
<td>Frame Filtering</td>
<td>Filters by validity, type, address</td>
<td>Per node option</td>
</tr>
<tr>
<td>Frame Type</td>
<td>Type: Data/CMD/ACK/Beacon...?</td>
<td>Per packet option</td>
</tr>
<tr>
<td>Multi-cast Frames</td>
<td>Broadcast/Unicast Destination?</td>
<td>Per packet option</td>
</tr>
<tr>
<td>Address Format</td>
<td>Dest. Address/Src. Address format?</td>
<td>Per packet – in IEEE 802.15.4 Global – in Proprietary mode</td>
</tr>
<tr>
<td>MAC Layer Security</td>
<td>Privacy: yes/no? Frame authenticity: yes/no, MIC-tag size?</td>
<td>Per packet enabling option</td>
</tr>
<tr>
<td><strong>PROPRIETARY MODE ONLY:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Link Agility</td>
<td>Air-Data-Rate Adaptation: yes/no? Adapt channel between TX/RX?</td>
<td>Per packet option</td>
</tr>
<tr>
<td>Buffer Handling</td>
<td>Packet vs. Streaming Buffer Mode</td>
<td>Per packet option</td>
</tr>
<tr>
<td>Auto-Repeater</td>
<td>Auto-Repeat Mode</td>
<td>Per packet and per RX node option</td>
</tr>
</tbody>
</table>
### TABLE 4-2: SUMMARY OF NODE CONFIGURATION OPTIONS

<table>
<thead>
<tr>
<th>NODE CONFIGURATION</th>
<th>Packet Mode</th>
<th>Proprietary</th>
<th>Proprietary Repeater</th>
<th>Propr. TX Stream</th>
<th>Propr. RX Stream</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRXMODE (00: Packet Mode)</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>10</td>
<td>01</td>
</tr>
<tr>
<td>FRMFMT (Std:0, Proprietary:1)</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>AUTORPTEN (Repeater Node:1)</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

#### FEATURE/FRAME NODE CAPABILITY

<table>
<thead>
<tr>
<th>FEATURE/FRAME</th>
<th>NODE CAPABILITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSMAEN (CSMA enable:1)</td>
<td>0 or 1</td>
</tr>
<tr>
<td>CRCSZ (CRC 2 bytes: 1, none: 0)</td>
<td>1 (or 0)</td>
</tr>
<tr>
<td>AUTOACKEN (Auto-ACK enable:1)</td>
<td>0 or 1</td>
</tr>
</tbody>
</table>

| Retransmission capability | Yes | Yes | No | No | No |
| Buffer Handling | 1 RX, 1 TX | 1 RX, 1 TX | 2 TRX | 2 TX | 2 RX |

With IEEE 802.15.4 Frames, capability to:

- TX, RX
- RX (TX) (bridging)
- discard
- n/a
- discard

With Proprietary Non-Streaming Type of Frames (Repeat:0/1), capability to:

- discard
- TX, RX
- Repeat if Repeat = 1
- n/a
- discard

With Proprietary Streaming Type of Frames, capability to:

- discard
- RX (sets on RX-Streaming)
- discard
- TX
- RX

<table>
<thead>
<tr>
<th>Security Processing</th>
<th>Available</th>
<th>Available</th>
<th>None</th>
<th>available but impractical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel Agility Support for ACK-ing</td>
<td>No</td>
<td>Yes</td>
<td>n/a</td>
<td>n/a</td>
</tr>
</tbody>
</table>

| PHY-features | Same for all |
| Available Data Rates | All(2) | All | All | All | All |
| On-the-fly receiver rate adaptation | Yes | Yes | Yes | Yes | Yes |
| DSSS | Yes | Yes | Yes | Yes |

**Note 1:** Proprietary frames requesting both Acknowledge and repeat are not recommended if any of the repeaters has AUTOACKEN = 0, and vice versa. Setting all of these may cause issues.

**Note 2:** Although 250 kbps is the only data rate that ensures compliance to the IEEE 802.15.4 standard, however as an extension, the other data rates can also be used with the standard MAC format, which may be easier to integrate with the legacy software.

**Note 3:** CRCSZ = 0 has no practical use in Standard-Format mode. If CRCSZ = 0 then AUTOACKEN = 0 and RXFILTER = 0x00 is required.

**Note 4:** Acknowledge sending is solved. To send a frame, the transmitter changes FRMFMT for the sending.
## REGISTER 4-1: MACCON1 (MAC CONTROL 1 REGISTER)

<table>
<thead>
<tr>
<th>Bit Field</th>
<th>Description</th>
<th>Value</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRXMODE&lt;1:0&gt;</td>
<td>TX/RX Mode Select Field bits</td>
<td>11 = Reserved, 10 = TX-Streaming mode, 01 = RX-Streaming mode, 00 = Packet mode</td>
<td></td>
</tr>
<tr>
<td>ADDRSZ&lt;2:0&gt;</td>
<td>Source/Destination Address Size Field bits</td>
<td>111 = 8 octets, 110 = 7 octets, 101 = 6 octets, 100 = 5 octets, 011 = 4 octets, 010 = 3 octets, 001 = 2 octets, 000 = 1 octet</td>
<td></td>
</tr>
<tr>
<td>CRCSZ</td>
<td>CRC Size bit</td>
<td>1 = 2 octets, 0 = 0 octet</td>
<td></td>
</tr>
<tr>
<td>FRMFMT</td>
<td>MAC Frame Format Adopted by the Network bit</td>
<td>1 = Proprietary, 0 = IEEE 802.15.4 standard compliant</td>
<td></td>
</tr>
<tr>
<td>SECFLAGOVR</td>
<td>Security Flag Override bit</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Legend:
- **R** = Readable bit
- **W** = Writable bit
- **U** = Unimplemented bit, read as ‘0’
- **-n** = Value at POR
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- **x** = Bit is unknown
- **r** = Reserved

1. Zero-length address occurs when the corresponding DAddrPrsnt/SAddrPrsnt bits of the packet Frame Control field are set to ‘0’.
2. Use the ADDRSZ field while receiving and transmitting, and must not be modified while RXEN or TXST is set.
3. Use the FRMFMT field while receiving and transmitting, and must not be modified while RXEN or TXST is set.

In Debug mode, this register bit is used to determine the frame format for both Tx/Rx frame in the packet buffers.
4.5 Protocol Selection and Constraints

Applications typically fall into two categories:

- Category 1: Standard compliant operation is required exclusively
- Category 2: Standard compliant operation is not required, capability to form a gateway to standard network is sufficient:
  - Green-field development. Proprietary MAC framing is optimal.
  - Legacy software is better served when the standard MAC frame format is applied, although the network is not required to use IEEE 802.15.4 standard modulation formats over the air.

In category 1, the Proprietary Features are not used. The constraints listed in Table 4-3 must be applied. A significant limitation is that IEEE 802.15.4 only allows using a single air-data-rate, 250 kbps.

In category 2, these constraints are relieved and the network may either adopt the proprietary (FRMFMT = 1) or the IEEE 802.15.4 MAC (FRMFMT = 0) frame formats which fits better with the conditions. In either options, the network can use all the air-data-rates. A gateway to a standard network is formed through Bridging, see Section 8.0 “Bridging”. The Physical Layer Configuration is described in Section 9.0 “Physical layer Functions”.

### Table 4-3: THE IEEE 802.15.4™ STANDARD CONSTRAINTS

<table>
<thead>
<tr>
<th>Parameter Description</th>
<th>Register Field</th>
<th>Default on Reset</th>
<th>IEEE 802.15.4™ constraint/recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame Format</td>
<td>FRMFMT</td>
<td>0</td>
<td>0 (std. frame format)</td>
</tr>
<tr>
<td>Buffer Handling</td>
<td>TRXMODE</td>
<td>00</td>
<td>00 (Packet mode)</td>
</tr>
<tr>
<td>Sender Data Rate</td>
<td>DR&lt;2:0&gt;</td>
<td>011</td>
<td>011 (TX 250 kbps)</td>
</tr>
<tr>
<td>Receiver Data Rate Reject Filter (otherwise data rate is adapted on-the-fly, per frame)</td>
<td>RATECON&lt;7:2&gt;</td>
<td>000000 (all enabled)</td>
<td>111101 (only enable: RX 250 kbps, SFD = 0xA7)</td>
</tr>
<tr>
<td>FCS (CRC) size (0 or 2 bytes)</td>
<td>CRCSZ</td>
<td>1</td>
<td>1 (2 byte CRC appended)</td>
</tr>
<tr>
<td>Frame rejection on CRC mismatch</td>
<td>CRCREJ</td>
<td>1</td>
<td>1 (CRC match enforced)</td>
</tr>
<tr>
<td>Frame rejection filter</td>
<td>RXFILTER&lt;7:0&gt;</td>
<td>0x7F</td>
<td>0x40 (frame rejection only on CRC mismatch)</td>
</tr>
<tr>
<td>Duplicate Rejection</td>
<td>IDENTREJ</td>
<td>0</td>
<td>1 (discard duplicates)</td>
</tr>
<tr>
<td>Automatic Acknowledge Handling (send/receive)</td>
<td>AUTOACKEN</td>
<td>0</td>
<td>1 (Auto ACK enabled)</td>
</tr>
<tr>
<td>Base time units applied by TXACKWAIT and RXACKWAIT</td>
<td>BASETM&lt;4:0&gt;</td>
<td>000010 (2 µs)</td>
<td>10000 (must be a divisor of 16 µs)</td>
</tr>
<tr>
<td>Wait duration (in base units) before Acknowledge sending (by the data frame recipient)</td>
<td>RXACKWAIT&lt;7:0&gt;</td>
<td>0x60</td>
<td>0x0C (must be &gt;= 192 µs)</td>
</tr>
<tr>
<td>Time-out duration (in base units) for Acknowledge reception (by the data originator)</td>
<td>TXACKTO&lt;7:0&gt;</td>
<td>0x80 (256 µs)</td>
<td>0x36 (must be &gt;= 864 µs)</td>
</tr>
<tr>
<td>CSMA medium access enabled</td>
<td>CSMAEN</td>
<td>1</td>
<td>1 (CSMA enabled)</td>
</tr>
<tr>
<td>CCA Mode (energy vs. carrier)</td>
<td>CCAMODE&lt;1:0&gt;</td>
<td>01</td>
<td>01 (4 options allowed)</td>
</tr>
<tr>
<td>CCA Measurement Duration</td>
<td>CCALEN&lt;1:0&gt;</td>
<td>1</td>
<td>10 (128 µs)</td>
</tr>
<tr>
<td>CCA Energy Threshold</td>
<td>EDthreshold&lt;1:0&gt;</td>
<td>0x32 (-88 dBm)</td>
<td>0x46 (-78 dBm; must be &lt;-75 dBm)</td>
</tr>
<tr>
<td>Energy Detect Mode (1: 128 us, 0: variable duration)</td>
<td>EDMODE&lt;1:0&gt;</td>
<td>0</td>
<td>0 (EDLEN applies)</td>
</tr>
<tr>
<td>Energy Detect Duration</td>
<td>EDLEN&lt;1:0&gt;</td>
<td>0x0E</td>
<td>0xE (15.360 ms; must be repeated multiple times)</td>
</tr>
</tbody>
</table>

**Note 1:** For more information on Physical Layer Configuration, see Section 9.0 “Physical layer Functions”.

**Note:** “Proprietary” is not equivalent to “full-custom”. The LENGTH field and Frame Control field must be used as described. (The transmitter processing of the FrameCtrl.SecEn bit cannot be disabled.) The payload portion can still carry customized management information that is processed in software. It is recommended that FRMFMT = 1 be used with NSTDREJ = 1 if a gateway is not required.
4.6 Frame-On-Air/Air-Data-Rate

In the Originator of the frame, FRMFMT and DR<2:0> select the air-data-rate and the frame format. The PRMBHLD bit indefinitely holds out the preamble. For more information on register definitions, refer to Register 4-2.

The recipient of the frame can receive the frame formats that FRMFMT and RATECON<7:2> selects.

The Recipient sends the Acknowledge using the same frame format as the acknowledged frame, except for the agility, see Section 7.1 “Channel Agility”. The Originator must set RATECON<7:2> adequately to receive the ACK frame.

RATECON<1> selects between Legacy and Optimal PHY frame format. This is set independently from the rate or the MAC protocol using the MAC operation.

**FIGURE 4-7: FRAME-ON-AIR/AIR-DATA-RATE**

<table>
<thead>
<tr>
<th>Preamble type:</th>
<th>FRMFMT:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legacy: 192 µs fix</td>
<td>0 IEEE 802.15.4™ standard compliant</td>
</tr>
<tr>
<td>Optimal: Duration scales reciprocally with DR</td>
<td>1 Proprietary</td>
</tr>
</tbody>
</table>
## REGISTER 4-2: RATECON (RATE CONFIGURATION REGISTER)

<table>
<thead>
<tr>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-1</th>
<th>RW-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIS2000</td>
<td>DIS1000</td>
<td>DIS500</td>
<td>DIS250</td>
<td>DISSTD</td>
<td>DIS125</td>
<td>OPTIMAL</td>
<td>PSAV</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 7</th>
<th>bit 6</th>
<th>bit 5</th>
<th>bit 4</th>
<th>bit 3</th>
<th>bit 2</th>
<th>bit 1</th>
<th>bit 0</th>
</tr>
</thead>
</table>

### Legend:
- **R** = Readable bit
- **W** = Writable bit
- **U** = Unimplemented bit, read as ‘0’
- **-n** = Value at POR
- **‘1’** = Bit is set
- **‘0’** = Bit is cleared
- **x** = Bit is unknown
- **r** = Reserved

#### bit 7  DIS2000: Disable 2 Mbps Frame Reception bit
If this bit is set, then reception of 2 Mbps frames is disabled.

#### bit 6  DIS1000: Disable 1 Mbps Frame Reception bit
If this bit is set, then reception of 1 Mbps frames is disabled.

#### bit 5  DIS500: Disable 500 kbps Frame Reception bit
If this bit is set, then reception of 500 kbps frames is disabled.

#### bit 4  DIS250: Disable 250 kbps Frame Reception bit
If this bit is set, then reception of 250 kbps frames with non-standard-compliant SFD patterns is disabled.

#### bit 3  DISSTD: Disable IEEE 802.15.4 compliant Frame Reception bit
If this bit is set, then reception of 250 kbps frames with IEEE 802.15.4 compliant SFD patterns is disabled.

#### bit 2  DIS125: Disable 125 kbps Frame Reception bit
If this bit is set, then reception of 125 kbps frames is disabled.

#### bit 1  OPTIMAL: Optimized Preamble Selection bit
When this bit is set, then optimized preamble is used instead of legacy.
- **1** = Optimized preamble
- **0** = Legacy preamble

#### bit 0  PSAV: Power-Save Mode Selection bit
When this bit is set, frame detection is dependent on the RSSI signal, and the receive signal processor is turned on when a sudden and significant increase (PSAVTHR<3:0>) is detected in the signal strength or the signal strength is above an absolute level (DESENSTHR<3:0>).
- **1** = Power-Save mode
- **0** = Hi-Sensitivity mode
### REGISTER 4-3: MACCON1 (MAC CONTROL 1 REGISTER)  ADDRESS: 0x10

<table>
<thead>
<tr>
<th></th>
<th>R/W-00</th>
<th>R/W-001</th>
<th>R/W-1</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRXMODE&lt;1:0&gt;</td>
<td></td>
<td>ADDR_SZ&lt;2:0&gt;</td>
<td>CRCSZ</td>
<td>FRMFMT</td>
<td>SECFLAGOVR</td>
</tr>
<tr>
<td>bit 7-6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Legend:</td>
<td>R = Readable bit</td>
<td>W = Writable bit</td>
<td>U = Unimplemented bit, read as '0'</td>
<td>-n = Value at POR</td>
<td>1 = Bit is set</td>
</tr>
</tbody>
</table>

#### bit 7-6 TRXMODE<1:0>: TX/RX Mode Select Field bits
- **11** = Reserved
- **10** = **TX-Streaming mode**. In this mode, use both buffers for packet transmission. When issuing TRXMODE = 10, RXEN is cleared. SPI addresses 0x200 to 0x27F access Buffer 1 or Buffer 2 in alternation. Access to 0x37F through 0x383 has non-defined effect.
- **01** = **RX-Streaming mode**. In this mode, use both buffers for packet reception. When issuing TRXMODE = 01, TXST and TXENC/RXDEC bits are cleared and RXEN is set. SPI addresses 0x300 to 0x383 access Buffer 1 or Buffer 2 in alternation. In this mode, Proprietary mode packets other than streaming type are automatically discarded. Access to 0x200 through 0x283 has non-defined effect.
- **00** = **Packet mode**. In this mode, Buffer 1 is used as a Transmit while Buffer 2 as a Receive packet buffer. SPI addresses from 0x200 to 0x27F access Buffer 1. SPI addresses 0x300 to 0x383 access Buffer 2. TRXMODE = 00 is mandatory when FRMFMT = 0.

#### bit 5-3 ADDR_SZ<2:0>: Source/Destination Address Size Fields bits\(^{(1, 2)}\)
The size of the Source and Destination addresses for Proprietary packet. Note that this field has no effect on the processing IEEE 802.15.4 frames.
- **111** = 8 octets
- **110** = 7 octets
- **101** = 6 octets
- **100** = 5 octets
- **011** = 4 octets
- **010** = 3 octets
- **001** = 2 octets
- **000** = 1 octet

#### bit 2 CRCSZ: CRC Size bit
This bit indicates the size of the CRC field in each packet.
- **1** = 2 octets
- **0** = 0 octet

#### bit 1 FRMFMT: MAC frame format bit adopted by the network bit\(^{(3)}\)
This bit determines the frame format used in the network.
- **1** = Proprietary
- **0** = IEEE 802.15.4 standard compliant.

#### bit 0 SECFLAGOVR: Security Flag Override bit
The user can override security flags used in the CCM-CTR, CBC-MAC and CCM operations, otherwise the device uses the standard (2003/2006) definition.

### Note
1. Zero-length address occurs when the corresponding DAddrPrsnt/SAddrPrsnt bits of the packet Frame Control field are set to '0'.
2. Use ADDR_SZ field while receiving and transmitting, and must not be modified while RXEN or TXST is set.
3. Use FRMFMT field while receiving and transmitting, and must not be modified while RXEN or TXST is set. In Debug mode, this register bit is used to determine the frame format for both Tx/Rx frame in the packet buffers.
### REGISTER 4-4: BBCON (BASEBAND CONFIGURATION REGISTER)  
ADDRESS 0x38

<table>
<thead>
<tr>
<th>Bit</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>RNDMOD</td>
<td>Random Modulation bit. If this bit is set, the transmitter randomly transmits DSSS symbols or MSK chips if PRMBHOLD bit is set. The purpose of this register is only for testing.</td>
</tr>
<tr>
<td>6</td>
<td>AFCOVR</td>
<td>AFC Override bit. If this bit is set, the receiver uses CFOMEAS register as the CFO in reception.</td>
</tr>
</tbody>
</table>
| 5-4  | RXGAIN<1:0>        | Receiver Gain Register Field bits. If this bit is set, the AGC operation is inhibited in the receiver and the receiver radio gain Configuration is selected between three different gain levels. Encoding:  
|      |                    | 11 = AGC operation is enabled (default value) |
|      |                    | 10 = High gain |
|      |                    | 01 = Middle gain |
|      |                    | 00 = Low gain  |
|      |                    | This feature is used for test and streaming purposes. To reduce the required interframe-gap, the RXGAIN must set to one of the fixed gain options when the MAC is in Streaming mode. |
| 3    | PRMBHOLD           | Preamble Hold Enable bit. Effect: Appends extra bytes to the transmitted preamble in endless repetition until it is cleared. Details: The hardware checks this bit during transmission before finishing the preamble. The DR<2:0> and the register OPTIMAL determine the appropriate preamble byte and applies the modulation format. When this flag is released the transmission of the current preamble byte is completed followed by transmitting the LENGTH field and the payload.  
|      |                    | 1 = Enable endless preamble repetition  |
|      |                    | 0 = Disable/stop endless preamble repetition |  
| 2-0  | PRMBSZ<2:0>        | Preamble Size Adjustment Field bits. Enables adjusting the transmitted preamble length when OPTIMAL = 1. Encoding:  
|      |                    | 500 kbps preamble length = (PRMBSZ<2> + 4) units, where unit = 16 μs (1 octet at 500 kbps)  |
|      |                    | 1 Mbps preamble length = (PRMBSZ<1:0> + 8) units, where unit = 4 μs (1 octet at 2 Mbps)  |
|      |                    | 2 Mbps preamble length = (PRMBSZ<1:0> + 8) units, where unit = 4 μs (1 octet at 2 Mbps)  |
|      |                    | This register field does not affect the Legacy frames and 125/250 kbps optimized frames. |
4.7 Security Suites

MRF24XA provides extensive hardware support for security suites defined in 802.15.4-2003/2006 standard. The security suites are based on the AES-128 block cipher transformation. Block ciphers are ciphers that work on a plaintext block of a fixed length to produce a ciphertext block of the same length. Given a particular Key (K), there is a 1-to-1 correspondence between the Plaintext Block (P) and the Ciphertext Block (C).

- Encryption operation: \( C_i = E_K(P_i) \)
  - Ciphertext block \((i)\) is produced by Encrypting Plaintext block \((i)\) using key K.
- Decryption operation: \( P_i = D_K(C_i) \)
  - Plaintext block \((i)\) is produced by Decrypting Ciphertext block \((i)\) using key K.

4.7.1 ELECTRONIC CODE BOOK MODE (ECB)

The simple usage of the block cipher is known as ECB mode. There are several issues in using a block cipher in ECB mode to encrypt data. Each plaintext block is encrypted to the same ciphertext block and it is possible to associate the ciphertext block with an event without knowing that the plaintext block itself. To trigger the event, user can resend the ciphertext block, a process known as Replay Attack. In addition, most block cipher algorithms in ECB mode do nothing to scramble repetitive data, making the plaintext block reversible from the ciphertext block.

4.7.2 COUNTER MODE (CTR)

In CTR mode, each ciphertext block is produced by XOR'ing the plaintext block with the encrypted version of a counter input. The initial value of the counter serves as the Initialization Vector (IV) for the message block, and may be changed for each message block. Although the term “counter” is used, this does not mandate the use of a true counter. An easy-to-compute function, which is practically non-repeating (at least for a long time) may be used. CTR mode may be 100% parallelized for both encryption and decryption.

802.15.4-2003 defines CTR mode as follows:

\[
\begin{align*}
CTR_0 &= 0 \\
\text{Loop on each 16 byte block of plaintext} \\
C_i &= P_i \ xor \ E_K(\{ENCFLAGS, SECNONCE, CTR_i\}) \\
CTR_{i+1} &= CTR_i + 1
\end{align*}
\]

Where, \( ENCFLAGS \) is defined by the standard (0x82), \( SECNONCE \) is defined in the standards and \( CTR_i \) is a 2 byte block counter. If the last block of the plaintext is not 16 byte, it is zero-padded and only the required number of MSB bytes is used in the XOR operation. The host can override ENCFLAGS field when the SECFLAGOVR bit in MACCON1 register is enabled, and the new flags through SECENCFLAG register is set. The host can overwrite the SECNONCE register anytime through SECNONCE1..13 registers.

802.15.4-2006 does not define CTR mode.

4.7.3 CIPHER BLOCK CHAINING MODE (CBC)

In CBC mode, each plaintext block is XOR’ed with the result of the previous block encryption operation before being encrypted. In this way all plaintext blocks depend on the previous block, making it difficult to remove, add or change individual blocks without detection. In addition, the encryption for the first block is performed using XOR of the plaintext block and a Initial Value (IV). This initial value changes for each message, making it more resistant to Replay Attacks.

MRF24XA does not support CBC mode.

4.7.4 CIPHER BLOCK CHAINING MESSAGE AUTHENTICATION CODE MODE (CBC-MAC)

A CBC-MAC protects the authenticity of a message, and therefore implicitly its integrity. The algorithm takes the variable length input message and a secret key, and produces a MIC TAG (MIC and MAC terms are often used interchangeably). Any change to the content of the message results in a change of the MIC tag, which guarantees the integrity of the message. As the same message with a different secret key also produces a different MIC tag, a MAC mode also provides protection of authenticity.

802.15.4-2003 defines CBC-MAC mode as follows:

\[
\begin{align*}
P &= \{LENGTH, MACHDR, MACPAYLOAD\} \\
O_1 &= E_K(P) \\
\text{Loop on each 16 byte block of plaintext} \\
O_i &= E_K(P_i \ xor \ O_{i-1}) \\
\text{End}
\end{align*}
\]

Where \( LENGTH \) is the number of bytes to be authenticated (MAC header and payload).

802.15.4-2006 does not define CBC-MAC mode.
This mode is a combination of CTR mode (encryption) and CBC-MAC mode (authentication). Initially, CBC-MAC is applied to compute the MIC tag. CTR mode (encryption) is only performed on a selected portion of the authenticated message and the MIC tag. Different combination of authentication and encryption are formed.

802.15.4-2003/2006 defines CCM* mode as follows:

\[ P = \{\text{AUTHENTICATION FLAG, NONCE, LENGTH, MAC-HEADER, MACPAYLOAD}\} \]

\[ O_i = EK(P_i) \]

Loop on each 16 byte block of plaintext

\[ O_i = EK(P_i \ xor \ O_{i-1}) \]

End

MICTAG is the leftmost M bit of \( O_{\text{end}} \)

AUTHENTICATION FLAG: Reserved || Adata || M || L

---

**4.8 Buffer Processing in Non-Secured Sending**

General Frame processing as shown in Figure 4-8, applies when AUTORPTEN = 0. Unsecured frame (SecEn = 0) is not an Acknowledgement and CRCSZ = 1. Acknowledgement is transmitted or received by software (AUTOACKEN = 0) and CRCSZ = 1 (ACK may contain piggyback data in the payload, at the discretion of the host software).

In Figure 4-8, note the following:

- Transmit buffer is the buffer starting at 0x200, whereas the Receive Buffer is the buffer starting at address 0x300 (this also applies for Streaming mode).
- In the Transmit Buffer, FCS is automatically appended to the frame, and the LENGTH field is also automatically incremented.
- The Receive buffer holds the FCS appended frame, and the according LENGTH. RSV is appended to the frame. Note the ordering of the RSV fields. RSV appending does not affect the LENGTH field.

The following cases are not described in this data sheet:

- Processing of secured frames (SecEn = 1) is presented in Section 5.3 “Security Material”.
- When AckReq = 1 (implying TXRXMODE = 00, CRCSZ = 1), then the Acknowledge frame is generated or accepted on-the-fly: without writing or reading the buffers (If CRCSZ = 0 then the CRC appending does not take place and Acknowledge is always processed by software: AUTOACKEN = 0).
- Repeater mode is described in Section 7.3 “Auto-Repeater”.

---

**FIGURE 4-8: GENERAL FRAME PROCESSING**

<table>
<thead>
<tr>
<th>TX buffer loaded:</th>
<th>Length</th>
<th>MHR</th>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>TXST set by host</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TX buffer processed:</td>
<td>Length</td>
<td>MHR</td>
<td>Payload</td>
</tr>
<tr>
<td>RXIF interrupt received, RXBUFFUL is</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>set. (RSV appending enabled in: RXCON2)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RX buffer holds</td>
<td>Length</td>
<td>MHR</td>
<td>Payload</td>
</tr>
<tr>
<td>MAC Frame</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

/LENGTH up to payload

/LENGTH up to CRC

/ if CRCSZ = 1 in Transmitter

/LENGTH up to CRC

/ if CRCSZ = 1 in Receiver
4.9 Frame Transmission in Packet Mode

FIGURE 4-9: TRANSMITTER PROCESSING IN PACKET MODE

- TXST ← 1
  - Select BUF1
  - TX MHR parsing
    - Parsable?
      - Fail (invalid format)
      - FRMIF ← 1
      - if CRCSZ = 1
        - Size run over 127 bytes: TXSZIF = 1
    - Pass
      - Tx Security Processing
        - if required
      - FCS appending
        - if CRCSZ = 1
      - TXSZIF
        - 0 (In-place test only)
          -TXST
            - 1
              - TXRETCNT ← 0
                - Reset retransmission count
      - CSMA-CA
        - TXCSMAIF
          - 1
            - Medium access failure: TXCSMAIF = 1
          - 0
            - ACK required?
              - Yes
                - ACK received?
                  - Yes (success)
                    - TXIF ← 1
                  - No
                    - Increment TXRETCNT
                      - No
                        - TXRETCNT ≥ TXRETMCNT
                          - Yes (failure)
                            - Max. retransmissions reached
                            - BUF1TXPP ← 0
                            - BUF2TXPP ← 0
                            - TXST ← 0
                            - TXBUFSIZE ← 1
                            - Interrupt Service
                          - Yes (success)
                            - TXIF ← 1
                          - No
                            - TXIF ← 1
          - No (Sending complete)
            - TXIF ← 1

- Figure 5-7
- Figure 5-9
- Figure 4-12

Pass
## TXCON (TRANSMIT CONTROL REGISTER)

**Address:** 0x12

<table>
<thead>
<tr>
<th>R/W/HC-0</th>
<th>R/W-0</th>
<th>R/W/HC-0</th>
<th>R/HS/HC-1</th>
<th>RW-1</th>
<th>R/W-011</th>
</tr>
</thead>
<tbody>
<tr>
<td>TXST</td>
<td>DTSM</td>
<td>TXENC</td>
<td>TXBUFEMPTY</td>
<td>CSMAEN</td>
<td>DR&lt;2:0&gt;</td>
</tr>
</tbody>
</table>

### Legend:
- **R** = Readable bit
- **W** = Writable bit
- **U** = Unimplemented bit, read as ‘0’
- **-n** = Value at POR
- **‘1’** = Bit is set
- **‘0’** = Bit is cleared
- **x** = Bit is unknown
- **r** = Reserved
- **HC** = Hardware Clear
- **HS** = Hardware Set

#### bit 7
**TXST:** Transmit Start bit

- **1** = Starts the transmission of the next TX packet\(^{(1,2)}\)
- **0** = Termination of current TX operation, which may result in the transmission of an incomplete packet

**Hardware clear:**
- After the packet is successfully transmitted (including all attempted retransmissions, if any), the hardware clears this bit and sets the TXIF and IDLEIF.
- If the packet transmission fails due to a CSMA failure, then this bit is cleared, and TXCSMAIF is set.
- If Acknowledge is requested (AckReq bit field in the transmitted frame is set) and not received after the configured number of retransmissions (TXRETMCNT), then TXST bit is cleared and a TXACKIF is set.
- In TX-Streaming mode (TRXMODE), TXST is set even when it is already set, resulting in a posted start. When the current TX operation completes, the posted start immediately starts afterward. Clearing of the TXST bit clears both the current and the posted (pending) TX starts. TXOVFIF is set when TXST = 1, a posted start is present and a Host Controller write to the packet buffer occurs. Outside of TX-Streaming mode, writes to TXST when TXST is already set is ignored.

Clearing this bit aborts the current operation in the following cases:
- When transmitting a packet in Packet mode or in TX-Streaming mode
- When waiting for an ACK packet after a transmission
- During the CSMA CA algorithm
- When transmitting a repeated frame

This field is read at any time to determine if the TX operation is in progress.

#### bit 6
**DTSM:** Do Not Touch Security Material bits\(^{(2)}\)

- **1** = Device do not change the security material configured by the host MCU
- **0** = Device tries to configure the security material related registers

MCU must fill the following registers: SECNONCE, SECHDRINDX, SECPAYINDX and SECEND-INDX.

#### bit 5
**TXENC:** TX Encryption

Setting this bit starts TX security processing (authentication or encryption, or both) of the packet in the buffer it was last written. TXENC is cleared and TXENCIF is set when the processing is complete. TXENC must be issued when NWK layer security must be processed. 802.15.4-2003/2006 MAC layer security operation is automatically performed when TXST bit is set. This field must not be modified while TXST is set.

### Note 1:
Transmission may include automatic security processing, CRC appending, CSMA-CA channel access, Acknowledge reception and retransmissions depending on the register Configuration and the Frame Control field of the frame to be transmitted.

### Note 2:
Setting TXST bit in either Sleep/RFOFF state, device transits to TX state for packet transmission.
REGISTER 4-5:  TXCON (TRANSMIT CONTROL REGISTER) (CONTINUED)  ADDRESS: 0x12

bit 4  TXBUFEMPTY: TX Buffer Empty bit

TXBUFEMPTY = 1 indicates that the host MCU can safely start writing a new frame to the buffer without overwriting any content that is in use. Writing a single byte to the buffer causes this bit to be cleared. TXBUFEMPTY = 0 does not prevent the host from writing further bytes to the buffer. TXBUFEMPTY is set by the device when transmission is complete.

1 = MCU can safely start writing a new frame to the buffer
0 = Buffer is full, or being written to

When TRXMODE = 00:
(PACKET) mode is configured then TXBUFEMPTY is set at the same time as TXST is cleared. An interrupt is also generated. Therefore, this bit provides no extra information.

When TRXMODE = 10:
(TXSTREAMING) mode is configured then TXBUFEMPTY is set at the same time as one of the buffers becomes free, while TXST may be set. Therefore, the host MCU uses TXBUFEMPTY to ensure that it can start loading the next frame to the buffers without overwriting a packet being sent (TXOVFIF).

bit 3  CSMAEN: CSMA-CA Enable bit

This bit enables CSMA-CA algorithm before transmission.

1 = CSMA-CA enabled
0 = CSMA-CA disabled

bit 2-0  DR<2:0>: Transmit Data Rate Field bits

111 = Reserved
110 = 2 Mbps
101 = 1 Mbps
100 = 500 kbps
011 = 250 kbps
010 = 125 kbps
001 = Reserved
000 = Reserved

When transmitting an Auto-ACK frame with Adaptive Data Rate in response to a received frame, the AckDataRate field in the received frame automatically determines the data rate of the PHY, and not by this Register field. In all other cases, use this Register field as the current PHY data rate when transmitting.

The PHY determines the data rate for all received frames regardless of this Register field and the Adaptive Data Rate Configuration. For more information, refer to Register 4-2.

Note  1:  Transmission may include automatic security processing, CRC appending, CSMA-CA channel access, Acknowledge reception and retransmissions depending on the register Configuration and the Frame Control field of the frame to be transmitted.

2:  Setting TXST bit in either Sleep/RFOFF state, device transits to TX state for packet transmission.
REGISTER 4-6: PIR2 (PERIPHERAL INTERRUPT REGISTER 2)                  ADDRESS: 0x05

<table>
<thead>
<tr>
<th>bit 7</th>
<th>bit 6</th>
<th>bit 5</th>
<th>bit 4</th>
<th>bit 3</th>
<th>bit 2</th>
<th>bit 1</th>
<th>bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>TXIF</td>
<td>TXENCIF</td>
<td>TXMAIF</td>
<td>TXACKIF</td>
<td>TXCSMAIF</td>
<td>TXSZIF</td>
<td>TXOVFIF</td>
<td>FRMIF</td>
</tr>
</tbody>
</table>

Legend:

- R = Readable bit
- W = Writable bit
- U = Unimplemented bit, read as ‘0’
- -n = Value at POR
- ’1’ = Bit is set
- ‘0’ = Bit is cleared
- x = Bit is unknown
- r = Reserved
- HC = Hardware Clear
- HS = Hardware Set

bit 7  **TXIF**: Transmission Done Interrupt Flag bit

The current TX operation (TXST) has successfully completed. This event is unchanged when a hardware generated ACK packet is transmitted or when a packet is repeated. Nonpersistent, cleared by SPI read.

bit 6  **TXENCIF**: Transmit Encoding Interrupt Flag bit

The TX packet was successfully encrypted or complemented, or both with a Message Integrity Code (MIC). Set by the device after TXENC = 1, when TXENC is cleared. Nonpersistent, cleared by SPI read.

bit 5  **TXMAIF**: Transmitter Medium Access Interrupt Flag bit

Set by the device when the medium is accessed, specifically when the first sample in the preamble is transmitted into the air. Nonpersistent, cleared by SPI read.

bit 4  **TXACKIF**: Transmission Unacknowledged Failure Interrupt Flag bit

Set by the device when Acknowledge is not received after the configured maximum number of transmission retries RETXMCNT<3:0>, provided that the Frame Control field of the transmitted frame indicates AckReq = 1. Nonpersistent, cleared by SPI read.

bit 3  **TXCSMAIF**: Transmitter CSMA Failure Interrupt Flag bit

Set by the device when CSMA-CA finds the channel busy for BOMCNT<2:0> number of times, provided that CSMAEN = 1 is configured. Nonpersistent, cleared by SPI read.

bit 2  **TXSZIF**: Transmit Packet Size Error Interrupt Flag bit

Following TXST is set the packet size (including MIC tags and CRC) is found to be zero or to be greater than the maximum size that the buffers can support. Nonpersistent, cleared by SPI read.

bit 1  **TXOVFIF**: Transmitter Overflow Interrupt Flag bit

The Host Controller attempted to write a TX buffer that was not empty (TXBUFEMPTY = 0). Nonpersistent, cleared by SPI read.

bit 0  **FRMIF**: Frame Format Error Interrupt Flag bit

Set if the transmitter/receiver fails to parse the frame in the buffer (it is not as it must be or it is corrupted in demodulation).
REGISTER 4-7: PIR4 (PERIPHERAL INTERRUPT REGISTER 4)  ADDRESS: 0x07

<table>
<thead>
<tr>
<th>Bit</th>
<th>Bit 0</th>
<th>Bit 1</th>
<th>Bit 2</th>
<th>Bit 3</th>
<th>Bit 4</th>
<th>Bit 5</th>
<th>Bit 6</th>
<th>Bit 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>TXSFDIF</td>
<td>RXSFDIF</td>
<td>ERRORIF</td>
<td>WARNIF</td>
<td>EDCCAIF</td>
<td>GPIO2IF</td>
<td>GPIO1IF</td>
<td>GPIO0IF</td>
<td></td>
</tr>
</tbody>
</table>

Legend:  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as ‘0’  
-n = Value at POR  
‘1’ = Bit is set  
‘0’ = Bit is cleared  
x = Bit is unknown  
r = Reserved  
HC = Hardware Clear  
HS = Hardware Set

- **TXSFDIF**: Transmit SFD Sent Interrupt Flag bit  
  Set by the device when the last sample of the SFD field is sent into the air. Nonpersistent, cleared by SPI read.

- **RXSFDIF**: Receive SFD Detected Interrupt Flag bit  
  Set by the device when the SFD field of the received frame is detected. Nonpersistent, cleared by SPI read.

- **ERRORIF**: General Error Interrupt Flag bit  
  Set by the device, when malfunction state is reached.

- **WARNIF**: Warning Interrupt Flag bit  
  Set by the device when one of the following occurred:  
  - Battery voltage drops below the threshold given by BATMON<4:0>  
  - Resistor on pin 28 is missing or not connected well

- **EDCCAIF**: Energy Detect/CCA Done Interrupt Flag bit  
  Set by the device when Energy-detect or CCA measurement is complete (following that the host MCU has set the EDST/CCAST bit to start the measurement and the device is clearing it on completion). Nonpersistent. Cleared by SPI read.

- **GPIO2IF**: GPIO2 Interrupt Flag bit  
  Set by the device if the GPIOMODE register is set to normal operation, the GPIO is enabled and configured to input and the level matches with the polarity.

- **GPIO1IF**: GPIO1 Interrupt Flag bit  
  Set by the device if the GPIOMODE register is set to normal operation, the GPIO is enabled and configured to input and the level matches with the polarity.

- **GPIO0IF**: GPIO0 Interrupt Flag bit  
  Set by the device if the GPIOMODE register is set to normal operation, the GPIO is enabled and configured to input and the level matches with the polarity.
4.10 Frame Reception in Packet Mode

FIGURE 4-10: RECEIVER OPERATION (IF AUTORPTEN = 0)

- MCU:RXEN ← 1 while TXST = 0
- TXST ← 0 while RXEN = 1

RXListen Operation

Frame acquired?

No

Yes

RXSFDIF ← 1

Frame Reception

Abort Events

- MCU:EDST/CCAST ← 1 while in RxListen
- MCU:RXEN ← 0
- MCU:TXST ← 1 while in RxListen

Abort RXLISTEN or Frame Reception

Abort RXLISTEN or Frame Reception

Abort RXLISTEN or Frame Reception

Enter RXListen Operation disabling Acquisition

Do ED/CCA measurement

EDCCAIF ← 1

Do Transmission

TXST ← 0 by device
Or abortion:
TXST ← 0 by host
FIGURE 4-11: FRAME RECEPTION IN PACKET MODE (TRXMODE = 00)

Frame received over the Medium

RXSFIDIF ← 1 by device

BUF2 selected

RXBUFFUL?

0

LENTH and MAC frame written into buffer

RSV appending

MHR parsing

Pass

FRMIF ← 1

Fail

RXFLTIF ← 1

Acknowledgment is always bypassed in debug

Exit with failure. Packet discarded. Earlier packet kept. (RXBUFFUL = 1)

RXOVFIF ← 1

Yes

duplicate?

No

RXIDENTIF ← 1

Yes (discard)

RXIDENTIF ← -1 if duplicate

RX IDENTIF

ACK Required?

No

Auto ACK-sending

Yes

Acknowledge must be sent for duplicated frames if AckReq = 1.

RX IDENTIF

AckReq = 0 OR AUTOACKEN = 0 OR debug (BUF1RXPP or BUF2RXPP)

Security Material Retrieval support

RXBUFFUL ← 1

RXIF ← 1

Exit with success (RXBUFFUL = 1)

Figure 4-13

Software reads the buffer content, may decide on proceeding to security processing (Figure 5-16); finally, frees up the buffer by clearing RXBUFFUL.

Figure 5-15
REGISTER 4-8: RXCON1 (MAC RECEIVE CONTROL 1 REGISTER)  
ADDRESS: 0x15

<table>
<thead>
<tr>
<th>R/W/HC/HS-0</th>
<th>R/W-0</th>
<th>R/W/HC-0</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>RXEN</td>
<td>NOPA</td>
<td>RXDEC</td>
<td>RSVLQIEN</td>
<td>RSVRSSIEN</td>
<td>RSVCHDREN</td>
<td>RSVCF0EN</td>
<td>r</td>
</tr>
</tbody>
</table>

Legend:  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
-<n> = Value at POR  
'1' = Bit is set  
'0' = Bit is cleared  
x = Bit is unknown  
r = Reserved  
HC = Hardware Clear  
HS = Hardware Set

bit 7  
**RXEN**: Receive Enable Field bit  
This bit Enables/Disables the packet reception. If an RX packet is currently being received, clearing this bit causes that packet to be discarded.  
1 = RX enabled  
0 = RX disabled  
Hardware clear/set when:  
• Cleared when TRXMODE is set to TX-Streaming mode  
• Set when TRXMODE is set to RX-Streaming mode  
Clearing this bit aborts the current operation in the following cases:  
• Receiving a packet in Packet mode or in RX-Streaming mode  
Changes to most RX related settings must be only done when this bit is cleared.  
The clear channel assessment (CSMAEN) and ACK-frame reception does not require RXEN = 1 as the device turns the radio into RX when needed, irrespective of the status of the RXEN bit.

bit 6  
**NOPA**: No Parsing bit  
This bit disables packet parsing. Only CRC is checked if it is enabled. This feature is useful in Sniffer mode.  
1 = Disable packet parsing  
0 = Enable packet parsing

bit 5  
**RXDEC**: RX Decryption bit  
Setting this bit starts RX security processing (authentication or decryption, or both) on the last received packet.  
1 = RX security processing started/in process. RXDECIF or RXTAGIF is set.  
0 = RX security processing inactive or complete  
This bit clears itself after RX decryption is completed.

bit 4  
**RSVLQIEN**: Receive Status Vector LQI Enable bit  
If this bit is set, the measured Link Quality is appended after the received frame in the packet buffer.  
1 = Append LQI field  
0 = Do not append LQI field

bit 3  
**RSVRSSIEN**: Receive Status Vector RSSI Enable bit  
If this bit is set, the measured RSSI is appended after the received frame in the packet buffer.  
1 = Append RSSI field  
0 = Do not append RSSI field

bit 2  
**RSVCHDREN**: Receive Status Vector Channel/MAC Type/Data Rate Enable bit  
If this bit is set, Channel, MAC type and Data Rate configurations used with the received frame are appended after the received frame in the packet buffer, using the encoding specified for CH<3:0>, FRMFMT and DR<2:0> (concatenated in this order when most significant bit (MSb) is first).  
1 = Append Channel, MAC type and Data Rate fields  
0 = Do not append Channel, MAC type and Data Rate fields

bit 1  
**RSVCFOEN**: Receive Status Vector CFO Enable bit  
If this bit is set, the estimated Carrier Frequency Offset of the received frame is appended after the received frame in the packet buffer, using the same encoding as CFOMEAS register.  
1 = Append CFO estimation  
0 = Do not append estimated CFO

bit 0  
**Reserved**: Maintain as '0'
### REGISTER 4-9: RXCON2 (MAC RECEIVE CONTROL 2 REGISTER)

**ADDRESS:** 0x16

<table>
<thead>
<tr>
<th>R/HS-0</th>
<th>RW-0</th>
<th>R-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>RXBUFFUL</td>
<td>IDENTREJ</td>
<td>ACKRXFP</td>
<td>ACKTXFP</td>
<td>AUTORPTEN</td>
<td>AUTOACKEN</td>
<td>ADPTCHEN</td>
<td>ADPTDREN</td>
<td></td>
</tr>
</tbody>
</table>

**Legend:**
- R = Readable bit
- W = Writable bit
- U = Unimplemented bit, read as ‘0’
- -n = Value at POR
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- x = Bit is unknown
- r = Reserved

**bit 7**

**RXBUFFUL:** RX Buffer Full bit

Host MCU clears this bit to indicate that the RX packet is processed. If this bit is uncleared before the next valid RX packet is detected (packet is not a duplicate, pass RX filter, and so on), then the device sets RXOVFIF and the buffer content is unmodified, where RXBUFFUL = 1 locks write access by a new frame. Moreover, the host can both read and write to the buffer or perform security processing.

In TRXMODE = 00 (PACKET) mode:
- 1 = Receive buffer content is yet to be read by the host or processed, and cannot be overwritten by a new frame
- 0 = Receive buffer is free for receiving a new frame

In TRXMODE = 01 (RXSTREAMING) mode:
- 1 = Current buffer being read from the bus contains a valid RX Packet
- 0 = Current buffer being read from the bus is empty

**bit 6**

**IDENTREJ:** Reject Identical Packet bit

Setting this bit enables the user to reject an incoming packet, in case its source address and sequence number is the same as the previously received packet.

This bit is used whenever a packet is received and ACK is transmitted, but the ACK is never received that the sender resends the TX packet. When this happens, triggers for RXIF is avoided for the second time for the same packet, thus, the second packet is ignored.

This bit is also used when a packet is repeated and the next repeater repeats the same packet back. This packet is received, but ignored.

- 1 = Any packet received with the same source address and sequence number as the last packet successfully received is discarded and RXIDNTIF is thrown
- 0 = Duplicated packets are processed further same as non-duplicated packets

**bit 5**

**ACKRXFP:** ACK RX Frame Pending bit

This read-only Status bit reflects the value of the FrameCtrl (FramePend) bit in the last received 802.15.4 compatible ACK frame.

**bit 4**

**ACKTXFP:** ACK TX Frame Pending bit

The value of this bit is transmitted in the FrameCtrl (FramePend) bit slot when the MAC sends an ACK packet in 802.15.4 Compatibility mode.

**bit 3**

**AUTORPTEN:** Auto-Repeat Enable bit

If this bit is set, the MAC automatically transmits a packet whenever a packet is received and its Repeat bit is set.

- 1 = Auto-Repeat feature is enabled
- 0 = Auto-Repeat feature is disabled

**Note 1:** Use ADPTCHEN field while receiving and transmitting, and must be unmodified while RXEN or TXST is set.

**Note 2:** Use ADPTDREN field while receiving and transmitting, and must be unmodified while RXEN or TXST is set.
REGISTER 4-9: RXCON2 (MAC RECEIVE CONTROL 2 REGISTER) (CONTINUED)

ADDRESS: 0x16

bit 2  AUTOACKEN: Auto-Acknowledge Enable bit

Setting this bit enables the device to automatically transmit an ACK packet whenever a packet is received and its AckReq bit is set.

1 = Automatic Acknowledge processing enabled
0 = Automatic Acknowledge processing disabled

bit 1  ADPTCHEN: Adaptive Channel Enable bit

Setting this bit enables the MAC in Proprietary mode to set the transmitting channel for the ACK packet based on the AckInfo field (Proprietary packet) of the received packet, rather than the CH<3:0> register bits.

1 = Adaptive Channel feature is enabled
0 = Adaptive Channel feature is disabled

This feature is also known as Channel Agility. For more information, see Section 7.1 “Channel Agility”.

bit 0  ADPTDREN: Adaptive Data Rate Enable bit

Setting this bit enables the MAC in Proprietary mode to set the transmission data rate for the ACK packet based on the AckInfo field (Proprietary packet) of the received packet, rather than the DR<2:0> register bits.

1 = Adaptive Data Rate feature is enabled
0 = Adaptive Data Rate feature is disabled

This feature is also known as Channel Agility. For more information, see Section 7.1 “Channel Agility”.

Note 1: Use ADPTCHEN field while receiving and transmitting, and must be unmodified while RXEN or TXST is set.

2: Use ADPTDREN field while receiving and transmitting, and must be unmodified while RXEN or TXST is set.
REGISTER 4-10:  PIR3 (PERIPHERAL INTERRUPT REGISTER 3)  ADDRESS: 0x06

<table>
<thead>
<tr>
<th>bit 7</th>
<th>bit 6</th>
<th>bit 5</th>
<th>bit 4</th>
<th>bit 3</th>
<th>bit 2</th>
<th>bit 1</th>
<th>bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>RXIF</td>
<td>RXDECIF</td>
<td>RXTAGIF</td>
<td>RXIDENTIF</td>
<td>RXFLTIF</td>
<td>RXOVFIF</td>
<td>STRMIF</td>
<td></td>
</tr>
</tbody>
</table>

Legend:  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as ‘0’  
-n = Value at POR  
‘1’ = Bit is set  
‘0’ = Bit is cleared  
x = Bit is unknown  
r = Reserved  
HC = Hardware Clear  
HS = Hardware Set

bit 7  
RXIF: Received Successful Interrupt Flag bit  
Set by the device when a frame passed the packet filtering and accepted, refer to Register 2-23. This interrupt flag is set only once for a packet and is not set when the packet is the duplicate of a repeated transmission (sequence number matches with the previously received frame). Nonpersistent, cleared by SPI read.

bit 6  
RXDECIF: Receiver Decryption/Authentication Passed Interrupt Flag bit  
Set by the device when decryption/authentication finished without error. Nonpersistent, cleared by SPI read.

bit 5  
RXTAGIF: Receiver Decryption/Authentication Failure Interrupt Flag bit  
Set by the device when decryption/authentication finished with error. Nonpersistent, cleared by SPI read.

bit 4  
Reserved: Maintain as ‘0’

bit 3  
RXIDENTIF: Received Packet Identical Interrupt Flag bit  
Set by the device when the packet is the duplicate of a repeated transmission (sequence number and source address matches with the previously received frame). Nonpersistent, cleared by SPI read.

bit 2  
RXFLTIF: Received Packet Filtered Interrupt Flag bit  
Set by the device when a packet was received, but rejected by one or more RX Filters, refer to Register 2-23. Nonpersistent, cleared by SPI read.

bit 1  
RXOVFIF: Receiver Overflow Error Interrupt Flag bit  
Set by the device to indicate that a packet was received, but all RX buffers are full. Consequently, the packet was not received, but was discarded instead(1). Nonpersistent, cleared by SPI read.

bit 0  
STRMIF: Receive Stream Time-Out Error Interrupt Flag bit  
Set by the device to indicate that the duration specified in STRMTO elapsed since the last received packet while in RX-Streaming mode, and the MAC clears the stored sequence number. Nonpersistent, cleared by SPI read.

Note 1:  
In Packet mode, use a single buffer for received frames. In RX-Streaming mode, use both buffers for reception.
**REGISTER 4-11: PIR4 (PERIPHERAL INTERRUPT REGISTER 4)**  
**ADDRESS: 0x07**

<table>
<thead>
<tr>
<th>Bit</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>TXSFDIF</td>
<td>Transmit SFD Sent Interrupt Flag bit. Set by the device when the last sample of the SFD field has been sent into the air. Nonpersistent, cleared by SPI read.</td>
</tr>
<tr>
<td>6</td>
<td>RXSFDIF</td>
<td>Receive SFD Detected Interrupt Flag bit. Set by the device when the SFD field of the received frame is detected. Nonpersistent, cleared by SPI read.</td>
</tr>
<tr>
<td>5</td>
<td>ERRORIF</td>
<td>General Error Interrupt Flag bit. Set by the device, when malfunction state is reached.</td>
</tr>
</tbody>
</table>
| 4   | WARNIF                | Warning Interrupt Flag bit. Set by the device when one of the following occurred:  
|     |                       | • Battery voltage drops below the threshold given by BATMON<4:0>  
|     |                       | • Resistor on pin 28 is missing or not connected well |
| 3   | EDCCAIF               | Energy Detect/CCA Done Interrupt Flag bit. Set by the device when Energy-detect or CCA measurement is complete (following that the host MCU has set the EDST/CCAST bit to start the measurement and the device is clearing it in on completion). Nonpersistent. Cleared by SPI read. |
| 2   | GPIO2IF               | GPIO2 Interrupt Flag bit. Set by the device if the GPIOMODE register is set to normal operation, the GPIO is enabled and configured to input and the level matches with the polarity. |
| 1   | GPIO1IF               | GPIO1 Interrupt Flag bit. Set by the device if the GPIOMODE register is set to normal operation, the GPIO is enabled and configured to input and the level matches with the polarity. |
| 0   | GPIO0IF               | GPIO0 Interrupt Flag bit. Set by the device if the GPIOMODE register is set to normal operation, the GPIO is enabled and configured to input and the level matches with the polarity. |

**Legend:**  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as ‘0’  
-n = Value at POR  
‘1’ = Bit is set  
‘0’ = Bit is cleared  
x = Bit is unknown  
r = Reserved  
HC = Hardware Clear  
HS = Hardware Set
## REGISTER 4-12: PIR2 (PERIPHERAL INTERRUPT REGISTER 2)

**ADDRESS:** 0x05

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>R/W/HS/HC-0</th>
<th>R/W/HS/HC-0</th>
<th>R/W/HS/HC-0</th>
<th>R/W/HS/HC-0</th>
<th>R/W/HS/HC-0</th>
<th>R/W/HS/HC-0</th>
<th>R/W/HS/HC-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>TXIF</td>
<td>TXIF</td>
<td>R</td>
<td>W</td>
<td>R</td>
<td>W</td>
<td>R</td>
<td>W</td>
<td>R</td>
</tr>
<tr>
<td>TXENCIF</td>
<td>TXENCIF</td>
<td>R</td>
<td>U</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>U</td>
</tr>
<tr>
<td>TXMAIF</td>
<td>TXMAIF</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>TXACKIF</td>
<td>TXACKIF</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>TXCSMAIF</td>
<td>TXCSMAIF</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>TXSZIF</td>
<td>TXSZIF</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>TXOVFIF</td>
<td>TXOVFIF</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>FRMIF</td>
<td>FRMIF</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
</tbody>
</table>

**Legend:**  
- **R** = Readable bit  
- **W** = Writable bit  
- **U** = Unimplemented bit, read as ‘0’  
- **-n** = Value at POR  
- **1** = Bit is set  
- **0** = Bit is cleared  
- **x** = Bit is unknown  
- **r** = Reserved  
- **HC** = Hardware Clear  
- **HS** = Hardware Set

**bit 7-1**  
Out of scope

**bit 0**  
**FRMIF**: Frame Format Error Interrupt Flag bit

Set if the transmitter/receiver fails to parse the frame in the buffer (it is not as it must be or it is corrupted in demodulation). For example, reserved values found in the MAC header fields.

Nonpersistent, cleared by SPI read.
4.11 Carrier Sense Multiple Access-Collision Avoidance (CSMA-CA)

Carrier Sense Multiple Access-Collision Avoidance (CSMA-CA) is performed before transmitting a packet to increase the odds that the packet is successfully received without interference from other transmitting devices nearby.

When enabled (CSMAEN = 1), the MAC automatically performs CSMA-CS using the underlying Clear Channel Assessment (CCA) operation. CSMA-CA is only performed before transmitting a packet (excluding ACK packets automatically transmitted during Auto-Acknowledge) in Packet and Repeater mode.

4.11.1 CSMA-CA CONFIGURATION

Setting the CSMAEN register bit enables CSMA-CA. CSMA-CA is automatically executed when the TXST register bit is set before the packet is transmitted. CSMA-CA is considered part of a transmission operation and it is aborted when the TXST register bit is cleared, and not when the RXEN register bit is cleared.

The following register bits are used in the Configuration of CSMA-CA:

- CSMAEN
- BOMCNT<2:0>
- BOUNIT<7:0>
- MINBE<3:0>
- MAXBE<3:0>

4.11.2 CSMA-CA BACK-OFF ALGORITHM

1. Wait a random number of Base time units between 0 and \((2^{\text{MINBE}-1}) \times \text{BOUNIT}<7:0>\).

   **Note:** If \(\text{MINBE} = 0\), the first iteration of the CSMA algorithm performs a CCA operation immediately without any backoff time.

2. Perform a Clear Channel Assessment (CCA) operation.

3. If CCA fails, then wait for a random number of Base time units between 0 and \((2^{(\text{MINBE}+1)-1}) \times \text{BOUNIT}<7:0>\).

4. Repeat above two steps until CCA passes, incrementing the back-off exponent each time, until the maximum back-off time becomes \((2^{\text{MAXBE}-1}) \times \text{BOUNIT}<7:0>\), or until the number of attempts is greater than BOMCNT<2:0>.

5. If CCA is failed, but the number of attempts is less than BOMCNT<2:0>, keep trying with a back-off time of \((2^{\text{MAXBE}-1}) \times \text{BOUNIT}<7:0>\) Base time units until the number of attempts is greater than BOMCNT<2:0>, or CCA passes.

6. If CCA still fails, the TX CSMA Error event is generated.

MRF24XA automatically controls an external LNA if enabled.
REGISTER 4-13: TMRCON (TIMER CONTROL REGISTER)  ADDRESS: 0x19

<table>
<thead>
<tr>
<th>Bit 7-5</th>
<th>BOMCNT&lt;2:0&gt;: CSMA-CA Back-off Maximum Count Field bits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The maximum number of back-off attempts that the CSMA-CA algorithm tries before declaring a channel access failure.</td>
</tr>
<tr>
<td>111</td>
<td>Reserved</td>
</tr>
<tr>
<td>110</td>
<td>Reserved</td>
</tr>
<tr>
<td>101</td>
<td>5 attempts</td>
</tr>
<tr>
<td>100</td>
<td>4 attempts</td>
</tr>
<tr>
<td>011</td>
<td>3 attempts</td>
</tr>
<tr>
<td>010</td>
<td>2 attempts</td>
</tr>
<tr>
<td>001</td>
<td>1 attempt</td>
</tr>
<tr>
<td>000</td>
<td>0 attempt</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit 4-0</th>
<th>BASETM&lt;4:0&gt;: Base Time Field bits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The number of 1 µs clock cycles that a Base time unit represents in all register settings. For more information, see Section 4.1 “MAC Architecture”.</td>
</tr>
</tbody>
</table>
REGISTER 4-14: CSMABE (CSMA-CA BACK-OFF EXPONENT CONTROL REGISTER)  
ADDRESS: 0x1A

<table>
<thead>
<tr>
<th>RW-0101</th>
<th>RW-0011</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAXBE&lt;3:0&gt;</td>
<td>MINBE&lt;3:0&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 7</th>
<th>bit 6</th>
<th>bit 5</th>
<th>bit 4</th>
<th>bit 3</th>
<th>bit 2</th>
<th>bit 1</th>
<th>bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAXBE&lt;3:0&gt;</td>
<td>CSMA-CA Back-off Maximum Count Fields</td>
<td>CSMA-CA Back-off Minimum Count bit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The maximum value of the Back-off exponent (BE) is in the CSMA-CA algorithm. The back-off time is ((2^{BE} - 1)) units.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1111 = Reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1110 = Reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1101 = Reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1100 (2^8 - 1) = 255 maximum units of back-off time</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1011 = Reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1010 = Reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1001 = Reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000 (2^8 - 1) = 255 maximum units of back-off time</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0111 = Reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0110 = Reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0101 = Reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0100 (2^8 - 1) = 255 maximum units of back-off time</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0011 = Reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0010 = Reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0001 = Reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000 (2^8 - 1) = No back-off time</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend:  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
-n = Value at POR  
'1' = Bit is set  
'0' = Bit is cleared  
x = Bit is unknown  
r = Reserved
REGISTER 4-15: BOUNIT (BACK-OFF TIME UNIT REGISTER)  ADDRESS: 0x1B

<table>
<thead>
<tr>
<th>Bit 7</th>
<th>R/W-10100000</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOUNIT&lt;7:0&gt;</td>
<td></td>
</tr>
</tbody>
</table>

Legend:  R = Readable bit   W = Writable bit   U = Unimplemented bit, read as ‘0’
         -n = Value at POR   ‘1’ = Bit is set   ‘0’ = Bit is cleared   x = Bit is unknown
         r = Reserved

bit 7-0  BOUNIT<7:0>: CSMA-CA Back-off Period Unit Field bits
The number of Base time units for the basic back-off time unit used by CSMA-CA algorithm.

11111111 = 256 Base time units
.
.
.
00000000 = 1 Base time unit

FIGURE 4-12: CSMA-CA ALGORITHM

```
        NB = 0
        BE = MINBE <3:0>

        Delay for a Random number of Back-off
        Periods between 0 and 2^BE - 1

        Perform CCA

        Channel Idle?
          Yes
          No

        NB = NB +1
        BE = min (BE + 1, MAXBE <3:0>)

        Yes
        NB > BOMCNT <2:0>?
          No  (Failure)
          Yes  Transmit Pending Packet  (Success)
```
4.12 Clear Channel Assessment (CCA)

Clear Channel Assessment (CCA) is a function within CSMA/CA to determine whether the wireless medium is ready and able to receive data, thus the transmitter can start sending it.

CCA is implemented outside of the MAC. This enables the radio to transmit in the presence of interference from other wireless protocols that operate on the same frequency.

CCA may be performed using either Energy Detection (ED), Carrier Sense (CS), or a combination of both. For more information on register description, see Section 9.6 “Clear Channel Assessment (CCA)”.

4.13 Condition for Hardware Acknowledgement

Figure 4-12 illustrates the condition for hardware acknowledgement as examined in Figure 4-9 and Figure 4-11. The AUTOACKEN = 0 case, when Acknowledgement is done by software. Both acknowledgement mechanisms (AUTOACKEN = 0/1) are described for the originator and the recipient.

**FIGURE 4-13: ACK REQUIREMENT DECISION**

```
“ACK Required?”

AckReq = 1

No

[ The sender of the frame does not request an ACK to it. ]

Yes

Packet Mode (TRXMODE = 00) and non-streaming frames are only considered since streaming type of frames always contain AckReq, and RX-Streaming nodes discard non-streaming type of frames.

AUTOACKEN = 1

No

[ ACK is handled by software. ]

Yes

AUTOACKEN = 1 is forbidden in Repeater Nodes, thus AUTORPTEN = 0 is implied.

[ The required and sufficient condition for Acknowledge sending is TRXMODE = 00 AND AUTORPTEN = 0 AND AUTOACKEN = 1 AND AckReq = 1. ]

Evaluates to TRUE

Evaluates to FALSE

[ If the transmitter (TXST = 1) of the frame having AckReq = 1 is configured as TRXMODE = 00 AND AUTOACK = 1 then it must receive a valid ACK to the sent frame to report successful sending. ]

[ If the receiver (TXST = 0) of the frame having AckReq = 1 is configured as TRXMODE = 00 AND AUTOACK = 1 then it must transmit an ACK to the received frame and all of the subsequent duplicates that arrive before the MHR is overwritten in the frame buffer. ]
```
### REGISTER 4-16: RXCON2 (MAC RECEIVE CONTROL REGISTER 2)  
**ADDRESS: 0x16**

<table>
<thead>
<tr>
<th>Bit 7-3</th>
<th>Bit 2</th>
<th>Bit 1-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Out of scope</td>
<td><strong>AUTOACKEN</strong>: Auto-Acknowledge Enable bit</td>
<td>Out of scope</td>
</tr>
</tbody>
</table>

**Legend:**  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as ‘0’  
-n = Value at POR  
‘1’ = Bit is set  
‘0’ = Bit is cleared  
x = Bit is unknown  
r = Reserved  
HS = Hardware Set  
C = Clearable bit  

- **bit 2**  
  - **AUTOACKEN**: Auto-Acknowledge Enable bit  
    
    **Recipient of a data frame:** If this bit is set, the device automatically transmits an ACK packet whenever a packet is received and its AckReq bit is set.

    **Originator of a data frame:** If this bit is set, the device awaits an ACK packet after the transmission of a packet (and after each retransmissions), and automatically processes the received ACKnowledge packet without writing it to the buffer. Setting this bit is required to enable automatic retransmissions of the device. The host MCU must clear AUTOACKEN to disable the automatic processing of acknowledge frames and enable writing to the buffer.

    1 = Automatic Acknowledge processing enabled  
    0 = Automatic Acknowledge processing disabled  

- **bit 1-0**  
  - Out of scope
4.14 Acknowledge Sending by Recipient

ACK sending must never use CSMA whether AUTO-ACKEN = 1 or 0.

FIGURE 4-14: AUTOMATIC ACKNOWLEDGE SENDING (AUTOACKEN = 1 AND ACKREQ = 1)

- RXEN = 1
- FrameCtrl.AckReq = 1 must hold for entry
- (Default) RXChannel = TXChannel = CH<3:0>.
  For the exception, refer to Section 7.1 “Channel Agility” on agility.

- Proprietary frame carrying AckInfo.

- FRMFMT = 1
  - Yes
  - No
  - Await RXACKWAIT Base time units. (start transition to TX just in time)
  - Start transmitting the preamble in the frame (without doing CSMA)
  - (when SFD is sent: TXSFDIF — 1)
  - Generate ACKNOWLEDGE frame MHR without writing to buffer. If FRMFMT = 0 then FrameCtrl.FramePend — ACKTXFP.
    SEQUENCE — received frame sequence
    Append FCS (on-the-fly)
    CH<3:0> selected as receive channel
    Return to RX
  - Exit

- ADPTCHEN
  - 0
    - Use AckInfo field of the received frame to select the transmitting channel for the ACK frame
  - 1
- ADPTDREN
  - 0
    - Use AckInfo field of the received data rate for the ACK frame
## TABLE 4-4: IDENTICAL PACKET REJECTION SCENARIO

<table>
<thead>
<tr>
<th>RXBUFFULL</th>
<th>IDENTREJ</th>
<th>AutoAck</th>
<th>AckReg</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>No ACK is sent, RXBUFFULL&lt;1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>ACK is sent, RXBUFFULL&lt; 1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Store sequence number and source address (SA + PID) No ACK is sent, RXBUFFULL&lt;- 1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Store sequence number and source address (SA + PID) ACK is sent, RXBUFFULL&lt;- 1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Store sequence number and source address (SA + PID) ACK is sent, RXBUFFULL&lt;- 1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>No ACK is sent, RXOVFIF&lt;- 1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>No ACK is sent, if stored sequence number and source address match with the received one, then</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>RXIDENTIF&lt;- 1 otherwise, RXOVFIF&lt;- 1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>If stored sequence number and source address match with the received one, ACK is sent and RXIDENTIF &lt;- 1 Otherwise, RXOVFIF&lt;- 1</td>
</tr>
</tbody>
</table>
4.15 Acknowledge Reception by Originator

After the reception of a valid ACK packet (Sequence field matches with transmitted Frame Sequence field), RXSFDIF and TXIF interrupts are generated (the RXIF is not generated while receiving an ACK frame). If the maximum number of retransmissions is reached (TXRETCNT >= TXRETMCNT), for example, no valid acknowledge received, TXACKIF interrupt is generated.

FIGURE 4-15: ACKNOWLEDGE RECEPTION AND RETRANSMISSION CONTROL

- **Figure 4-9 TRANSMITTER PROCESSING IN PACKET MODE**
  - TXST = 1
  - Entered when ACK Required

- **(Default) RXChannel = TXChannel = CH<3:0>.**
  - For the exception, refer to Section 7.1 “Channel Agility” on agility.
  - TXST = 1

- **Receiver turned OFF RXChannel selected**
  - Radio to RX
  - Start time-out counter
  - Frame received?
    - Yes
      - RXSFDIF = 1
      - Frame parsed without writing to buffer
      - Valid ACK frame received with SEQUENCE field matching the transmitted frame?
        - Yes
          - Increment TXRETCNT
        - No
          - TXRETCNT >= TXRETMCNT?
            - Yes
              - Update ACKRXFP
              - Success
            - No
              - Increment TXRETCNT
              - Retransmit
              - Receiver turned OFF TXChannel selected
        - No
          - Timer reached TXACKTO Base time units?
            - Yes
              - Receiver turned OFF TXChannel selected
            - No
              - TXACKIF = 1
              - Update ACKRXFP

- **No**
  - Receiver turned OFF TXChannel selected
  - TXRETCNT >= TXRETMCNT?
    - Yes
      - Receiver turned OFF TXChannel selected
    - No
      - Increment TXRETCNT

- **Figure 4-9 on agility.**
  - Section 7.1 “Channel Agility”
4.16 Base Time Units

Writing the BASETM<4:0> register bits selects the desired Base time. Each increment of BASETM<4:0> is equal to 1 µs.

The RXACKWAIT<7:0>, TXACKTO<7:0>, BOUNIT<7:0>, STRMTO<15:0> and OFFTM<7:0> fields are specified in terms of Base time units.

The BASETM<4:0> bits must be unchanged while RXEN = 1 or TXST = 1. The Base time is used in all modes for all types of packets.

### TABLE 4-5: BASE TIME UNITS

<table>
<thead>
<tr>
<th>Function/Timer</th>
<th>Range of Timer with BASETM&lt;4:0&gt; = 0x01 (1 µs resolution)</th>
<th>Range of Timer with BASETM&lt;4:0&gt; = 0x02 (2 µs resolution)</th>
<th>Range of Timer with BASETM&lt;4:0&gt; = 0x04 (4 µs resolution)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time to Wait before transmitting an ACK packet (RXACKWAIT&lt;7:0&gt;)</td>
<td>0-128 µs</td>
<td>0-256 µs</td>
<td>0-512 µs</td>
</tr>
<tr>
<td>Maximum time to look for an ACK packet before issuing a TX ACK Error or before retransmitting (TXACKTO&lt;7:0&gt;)</td>
<td>0-256 µs</td>
<td>0-512 µs</td>
<td>0-1024 µs</td>
</tr>
<tr>
<td>CSMA Backoff Time (0 - (2BE - 1) * BOUNIT&lt;7:0&gt;)</td>
<td>—</td>
<td>BOUNIT&lt;7:0&gt; = 160 (320 µs)</td>
<td>BOUNIT&lt;7:0&gt; = 80 (320 µs)</td>
</tr>
<tr>
<td>BE = 0</td>
<td>—</td>
<td>—</td>
<td>0</td>
</tr>
<tr>
<td>BE = 1</td>
<td>—</td>
<td>—</td>
<td>0-320 µs</td>
</tr>
<tr>
<td>BE = 2</td>
<td>—</td>
<td>—</td>
<td>0-960 µs</td>
</tr>
<tr>
<td>BE = 3</td>
<td>—</td>
<td>—</td>
<td>0-2.24 ms</td>
</tr>
<tr>
<td>BE = 4</td>
<td>—</td>
<td>—</td>
<td>0-4.8 ms</td>
</tr>
<tr>
<td>BE = 5</td>
<td>—</td>
<td>—</td>
<td>0-9.92 ms</td>
</tr>
<tr>
<td>BE = 6</td>
<td>—</td>
<td>—</td>
<td>0-20.16 ms</td>
</tr>
<tr>
<td>BE = 7</td>
<td>—</td>
<td>—</td>
<td>0-40.64 ms</td>
</tr>
<tr>
<td>BE = 8</td>
<td>—</td>
<td>—</td>
<td>0-81.6 ms</td>
</tr>
<tr>
<td>RX Stream Timeout (STRMTO&lt;15:0&gt;)</td>
<td>0-65 ms</td>
<td>0-131 ms</td>
<td>0-131 ms</td>
</tr>
<tr>
<td>Minimum OFF Time (OFFTM&lt;7:0&gt; * 32)</td>
<td>0-8 ms</td>
<td>0-16 ms</td>
<td>0-32 ms</td>
</tr>
</tbody>
</table>

**Note 1:** The maximum delay that MAC can support is 131 ms. Values outside this range may be set, but results in truncation of the number to one that is less than or equal to 131 ms.

**Note 2:** The value of 320 µs is chosen as it is the value referenced in the 802.15.4-2006 specification. Other values are possible, but may break 802.15.4 compliance.

**Note 3:** Note that the OFFTM<7:0> register is the only timer value that is expressed not directly in BASETM units, but rather is expressed in BASETM * 32 units.
4.17 Initialization sequence

To reach optimal RX and TX parameters, some private register values must be changed after every reset. Note that all results in this document were measured with this initialization sequence.

Initiate HW Reset;
Wait for Ready IF;
MRF24XA_WriteByte(0xFF, 0x5A);
MRF24XA_WriteByte(0xB7, 0xC0);
MRF24XA_WriteByte(0x80, 0x87);
MRF24XA_WriteByte(0x81, 0x84);
MRF24XA_WriteByte(0x82, 0x8A);
MRF24XA_WriteByte(0x83, 0x5E);
MRF24XA_WriteByte(0x84, 0xA5);
MRF24XA_WriteByte(0x8A, 0x2B);
MRF24XA_WriteByte(0x8B, 0x30);
MRF24XA_WriteByte(0x8C, 0x30);
MRF24XA_WriteByte(0x97, 0xD0);
MRF24XA_WriteByte(0xA3, 0x37);
MRF24XA_WriteByte(0xA5, 0x14);
MRF24XA_WriteByte(0xAA, 0x54);
MRF24XA_WriteByte(0xAB, 0x0B);

Note: MRF24XA_WriteByte (unsigned short address, unsigned char value) writes the given value to the selected address.
5.0 IEEE 802.15.4™ COMPLIANT FRAME FORMAT AND FRAME PROCESSING

Figure 5-1 shows the general MAC header structure.
The specific format of the Acknowledge frame is pro-
vided in Figure 5-2. The frame buffer is written in the
following sequence: (1) LENGTH field byte, (2) Byte 0
of the FrameCtrl field, (3) Byte 1 of the FrameCtrl field,
and (4) SEQUENCE.

**FIGURE 5-1: IEEE.802.15.4™ MAC HEADER STRUCTURE**

- **Type<2:0>:** Indicates the frame type. For more
  information, see Section 5.1 “Frame Types in
  IEEE 802.15.4-Compliant Framing Mode”.
- **SecEn:** Security Enable bit. For more information,
  see Section 5.3 “Security Material” and
  Section 5.4 “Security Material Retrieval with
  IEEE 802.15.4 Compliant Frames”.
- **FramePend:** This bit of the ACK frame must be
  set when ACKTXFP = 1 and the frame being
  ACK’d is an 802.15.4 Data Request Command
  Frame (FrameCtrl.Type = 011 AND CMDCmd =
  0x04), and cleared otherwise. CSMA-CA is not
  performed before sending out ACK packets.
- **AckReq:** ACK Request. For more information, see
  Section 4.12 “Clear Channel Assessment
  (CCA)”.
- **PIDCmp:** PAN Identifier Compare. For more
  information, see Section 5.2 “Addressing in
  IEEE 802.15.4 Compliant Framing Mode”.
- **SAMode:** Source Address Mode. For more
  information, see Section 5.2 “Addressing in
  IEEE 802.15.4 Compliant Framing Mode”.
- **DAMode:** Destination Address Mode. For more
  information, see Section 5.2 “Addressing in
  IEEE 802.15.4 Compliant Framing Mode”.

© 2015 Microchip Technology Inc. Preliminary DS70005023C-page 133
FIGURE 5-2: IEEE.802.15.4™ ACKNOWLEDGE FRAME STRUCTURE

<table>
<thead>
<tr>
<th>Acknowledge Frame (IEEE 802.15.4™)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MHR</td>
</tr>
<tr>
<td>Length (1 octet)</td>
</tr>
</tbody>
</table>

FrameCtrl (IEEE 802.15.4 ACK frame)

- Byte 0 = 02h or 12h 0x00 or 0x10
- Byte 1 = 00h or 0x00 0x02 or 0x12

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2:0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rsvd.</td>
<td>PIDCmp</td>
<td>AckReq</td>
<td>FramePend</td>
<td>SecEn</td>
<td>Type</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7:6</th>
<th>5:4</th>
<th>3:2</th>
<th>1:0</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAMode</td>
<td>FrameVer</td>
<td>DAMode</td>
<td>Rsvd.</td>
</tr>
</tbody>
</table>

FIGURE 5-3: EXAMPLE: PACKET MODE WITHOUT SECURITY

Legend:
- Data Flow direction
- Initiator
- Event or Control Flow
- Responder

Sending Node
- TX Host MCU
- SPI, INT
- TX Device
- TX Configuration
- Construct frame
- Unsecured Frame (FR)
- TXBUFEMPTY=0
- Set TXST
- CRC appended
- TXSFDF, TXMAF
- Frame Sending:
  - Frame started on Air
  - SHR (SFO) finished
  - Buffer is written to Frame Complete
- TXIF, TXBUFEMPTY=1

Receiving Node
- RX Host MCU
- SPI, INT
- RX Device
- RX Configuration
- Clear RXBUFFULL
- RXSFDF=1
- RX-Frame Filter
   - (Duplicates silently discarded)
   - RXFLTIF or RXIF or FRMF
   - Unsecured Frame (FR)
### Transmitter Side:

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>TX Configuration:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>SECEN = 0&lt;br&gt;CSMAEN = 0</td>
</tr>
</tbody>
</table>

**Example**

- **ShortAddress:** 0x1A1B
- **PID:** 0x2C2D

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Construct, download unprocessed frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td><strong>Example</strong> 0C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Length = 12_d + 2_d (CRC) = 14_d = 0x0E</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FrameCtrl = 0x01 0x98 = lsb_0000_0001_1001_1000 (Data, SEC = 0, DA, SA Short Addresses, ver2006)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sequence = 0xA8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SA/DA PID = 0x2D2C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DA = 0xFFFF (broadcast)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SA = 0x1B1A (unicast)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAC Payload = 0xFF</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Set TXST: Launches transmission. CRC is appended. Length is incremented accordingly.</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
<td><strong>Example</strong> 0E</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CRC: 0xA7 0x8E</td>
</tr>
</tbody>
</table>

### Receiver Side:

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>RX Configuration: SECEN = 0 (NWK), Security Suite</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td><strong>Example</strong> Address: Don’t care.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>RXSFDIF = 1 unless RXBUFFUL = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td>(If RXBUFFUL= 1 then RXSFDIF= 1; RXOVFIF = 1; no writing to buffer)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>RX Parsing and Filtering when frame reception is complete.</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
<td>If duplicated packet then silently discarded</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Else if packet filtered then RXFLTIF,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Otherwise RXIF = 1 (since SECEN = 0)</td>
</tr>
</tbody>
</table>

**Example** RXFILTER(@0x18) = 0x45

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>RXIF = 1. CRC is not valid for the decrypted frame.</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td></td>
<td><strong>Example</strong> 0E</td>
</tr>
</tbody>
</table>
FIGURE 5-4: EXAMPLE: PACKET MODE WITH NKW-LAYER SECURITY

Legend: Data Flow direction Initiator Event or Control Flow Responder

Sending Node

1. Construct frame
2. TX Configuration
3. Key, Nonce, Indices
4. Set TXENC
5. TXENCIF=1, TXENC=0
6. TX security processing
7. TXF=1, TXBUFEMPTY=1

Receiving Node

1. RX Configuration
2. RX Frame Filter
3. Buffer is written to if no RXOVFIF Frame Complete
4. Check FR
5. RX security processing
6. Read RX buffer
7. RX DEC
8. RXDECIF=1 or RXTAGIF=1

Unprocessed Frame (FR)
MediumSPI, INT

TX Device

RX Device

TXBUFEMPTY=0
TXF=1, TXBUFEMPTY=1

RX Host MCU

TX Host MCU

Medium

TXSFDIF=1

RXSFDIF=1

RX Frame Filter

Read TX buffer

Set TXST

SHR (SFD) finished

TXSFDIF=1

Check PRFR optional

Read RX buffer

If RXBUFFULL=1 when RXSFDIF=1 then RXOVFIF=1 and drops packet

Set RXDEC

Retrieve Key, Nonce, Indices

Check PRFR

Retrieve Key, Nonce, Indices

Key, Nonce, Indices

Set RXDEC
## Transmitter Side:

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Description Construct, download unprocessed frame</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>Length = 18d + 4d (MIC-32) + 2d (CRC) = 24d = 0x18</td>
<td>Length = 18d + 4d (MIC-32) + 2d (CRC) = 24d = 0x18</td>
</tr>
<tr>
<td></td>
<td>FrameCtrl = 0x01 0x98 = lsb_0000_0001_1001_1000 (SEC = 0, DA, SA Short Addresses, ver2006)</td>
<td>FrameCtrl = 0x01 0x98 = lsb_0000_0001_1001_1000 (SEC = 0, DA, SA Short Addresses, ver2006)</td>
</tr>
<tr>
<td></td>
<td>Sequence = 0xA8</td>
<td>Sequence = 0xA8</td>
</tr>
<tr>
<td></td>
<td>SA/DA PID = 0x2D2C</td>
<td>SA/DA PID = 0x2D2C</td>
</tr>
<tr>
<td></td>
<td>DA = 0xFFFF (broadcast)</td>
<td>DA = 0xFFFF (broadcast)</td>
</tr>
<tr>
<td></td>
<td>SA = 0x1B1A (unicast)</td>
<td>SA = 0x1B1A (unicast)</td>
</tr>
<tr>
<td></td>
<td>Network Header = 0x001 0x02 0x03 0x04 0x05 0x06</td>
<td>Network Header = 0x001 0x02 0x03 0x04 0x05 0x06</td>
</tr>
<tr>
<td></td>
<td>Network Payload = 0xFF</td>
<td>Network Payload = 0xFF</td>
</tr>
<tr>
<td>2</td>
<td>Description TX Configuration: NWK layer security</td>
<td>Security Suite = MIC-32</td>
</tr>
<tr>
<td>3</td>
<td>Description TX Configuration: Key, Nonce, Payload Index, Header Index</td>
<td>Short Address = 0x1B1A</td>
</tr>
<tr>
<td></td>
<td>Example</td>
<td>PID = 0x2D2C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Header Index (@0x2B) = 12d</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Payload Index (@0x2C) = 18d</td>
</tr>
<tr>
<td></td>
<td>Key&lt;i&gt; = 0x0F0E0D0C0B0A09080706050403020100</td>
<td>Key&lt;i&gt; = 0x0F0E0D0C0B0A09080706050403020100</td>
</tr>
<tr>
<td></td>
<td>Nonce&lt;i&gt; = 0x50 + &lt;i&gt;, i = 0...12</td>
<td>Nonce&lt;i&gt; = 0x50 + &lt;i&gt;, i = 0...12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MRF24XA register content</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x20 22 33 44 55 66 77 88 1A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MRF24XA register content</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x28 1B 2C 2D 0C 12 00 00 00</td>
</tr>
<tr>
<td>4</td>
<td>Description Issue TXENC: Launches CCM authentication and encryption.</td>
<td>Issue TXENC: Launches CCM authentication and encryption.</td>
</tr>
<tr>
<td>5</td>
<td>Description Security Processing Done: TXENCIF = 1, TXENC = 0.</td>
<td>Security Processing Done: TXENCIF = 1, TXENC = 0.</td>
</tr>
<tr>
<td></td>
<td>Optionally, TX buffer is read. Processed Frame PRFR is compared to the result of the receiver security processing or to the calculated expected outcome.</td>
<td>Optionally, TX buffer is read. Processed Frame PRFR is compared to the result of the receiver security processing or to the calculated expected outcome.</td>
</tr>
<tr>
<td></td>
<td>Example</td>
<td>Expected buffer content</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Expected buffer content</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Encrypted payload (0xFF): 0x46</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MIC-32: 0x78 C3 22 32</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CRC: 0xA7 0x8E</td>
</tr>
<tr>
<td>6</td>
<td>Description Set TXST: Launches transmission. CRC is appended.</td>
<td>Set TXST: Launches transmission. CRC is appended.</td>
</tr>
<tr>
<td></td>
<td>Example</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CRC: 0xA7 0x8E</td>
</tr>
<tr>
<td>7</td>
<td>Description End of Transmission (No ACK Request, No CSMA): TXIF received.</td>
<td>End of Transmission (No ACK Request, No CSMA): TXIF received.</td>
</tr>
<tr>
<td></td>
<td>TXBUFEMPTY= 1</td>
<td>TXBUFEMPTY= 1</td>
</tr>
</tbody>
</table>
Receiver Side:

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RX Configuration: Security Suite</td>
<td>Security Suite = MIC-32</td>
</tr>
<tr>
<td>2</td>
<td>If RXBUFFUL is 0 then RXSFDIF = 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>If RXBUFFUL is 1 then RXSFDIF = 1 RXOVFIF = 1 (no writing to buffer)</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>RX Parsing and Filtering when frame reception is complete.</td>
<td>RXFILTER(@0x18) = 0x45</td>
</tr>
<tr>
<td></td>
<td>If packet filtered then RXFLTIF,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Otherwise = 1 (Network secured frame received since SECEN = 0)</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Read RX buffer containing PRFR + CRC. CRC is valid for the encrypted frame</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>and, LQI, RSSI (RSVs) appended to the frame.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example</td>
<td>RxBuffer = 0x45</td>
</tr>
<tr>
<td></td>
<td>If RXBUFFUL is 1 then RXSFDIF = 1 RXOVFIF = 1 (no writing to buffer)</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>RXDEC = 1 launches decryption and authenticity checking.</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>If authenticity is approved (success): RXDECIF = 1. CRC is not valid for the</td>
<td>1A</td>
</tr>
<tr>
<td></td>
<td>decrypted frame. Otherwise RXTAGIF = 1.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example</td>
<td>RxTag = 0x42</td>
</tr>
</tbody>
</table>

5.1 Frame Types in IEEE 802.15.4-Compliant Framing Mode

The Type<2:0> bit field in FrameCtrl uses the encoding in Table 5-1.

TABLE 5-1: IEEE 802.15.4™ FRAME TYPES

<table>
<thead>
<tr>
<th>TYPE Field</th>
<th>Frame Type</th>
<th>Related Hardware Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>Beacon</td>
<td>Beacons are a specific type of broadcast frames. This device does not provide support for MHR-parsing on Beacon frames. Beacon frames are always accepted as valid frames.</td>
</tr>
<tr>
<td>001</td>
<td>Data</td>
<td>This is filtered by setting DATAREJ.</td>
</tr>
<tr>
<td>010</td>
<td>Acknowledge</td>
<td>Must be generated by the receiver (from SW or HW), if AckReq = 1 in the last received frame, and must contain the same Sequence value. This is generated by hardware (AUTOACKEN = 1). In this case it is not loaded to the TX frame buffer. AUTOACKEN = 1 requires CRCSZ = 1 on both the transmitter and the receiver side.</td>
</tr>
<tr>
<td>011</td>
<td>Command</td>
<td>This is filtered by CMDREJ. First byte of payload (Command) is never encrypted. See command encoding in Table 82 in Section 7.3 of IEEE 802.15.4™-2006.</td>
</tr>
<tr>
<td>1xx</td>
<td>Reserved</td>
<td>—</td>
</tr>
</tbody>
</table>

If SecEn bit in FrameCtrl is set, the hardware parses the frame to construct the security material (both at sending and after reception). In the case of Beacon frames, it is the responsibility of the host MCU to set the security materials before transmission (TXST) and after reception (RXIF).

For Beacon frames, the Frame Version subfield must be set to ‘1’ if the Security Enabled subfield is set to ‘1’.
**REGISTER 5-1: RXFILTER (RX FILTER REGISTER)**

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7-6</td>
<td>Out of Scope</td>
</tr>
<tr>
<td>5</td>
<td><strong>CMDREJ</strong>: Command Frame Reject Enable bit</td>
</tr>
<tr>
<td></td>
<td>Setting this bit enables the user to reject all packets with FrameCtrl&lt;Type&gt; equal to Command.</td>
</tr>
<tr>
<td></td>
<td>1 = Reject all Command packets</td>
</tr>
<tr>
<td></td>
<td>0 = Disable Command Frame Rejection</td>
</tr>
<tr>
<td>4</td>
<td><strong>DATAREJ</strong>: Data Frame Reject Enable bit</td>
</tr>
<tr>
<td></td>
<td>Setting this bit enables the user to reject all packets with FrameCtrl&lt;Type&gt; equal to Data.</td>
</tr>
<tr>
<td></td>
<td>1 = Reject all Data packets</td>
</tr>
<tr>
<td></td>
<td>0 = Disable Data Frame Rejection</td>
</tr>
<tr>
<td>0-2</td>
<td>Out of Scope</td>
</tr>
</tbody>
</table>
5.2 Addressing in IEEE 802.15.4 Compliant Framing Mode

The Destination Addressing Mode (DAMode) and Source Addressing Mode (SAMode) bit fields of FrameCrtl defines the Address format used in MHR, see Figure 5-1. DAMode subfield encodes the length of the DestPID and DestAddr fields as listed in Table 5-3. SAMode subfield encodes the length of the SrcPID and SrcAddr fields as listed in Table 5-3.

**TABLE 5-2: IEEE 802.15.4™ DESTINATION ADDRESSING MODES**

<table>
<thead>
<tr>
<th>DAMode b1, b0</th>
<th>Destination Addressing Mode</th>
<th>DestPID</th>
<th>DestAddr format</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>16-bit DestPID and 64-bit Dest. Long Address</td>
<td>XXXXh</td>
<td>XXXX_ XXXX_ XXXX_ XXXXh</td>
</tr>
<tr>
<td>10</td>
<td>16-bit DestPID and 16-bit Dest. Short Address</td>
<td>XXXXh</td>
<td>XXXXh</td>
</tr>
<tr>
<td>01</td>
<td>Reserved</td>
<td></td>
<td>—</td>
</tr>
<tr>
<td>00</td>
<td>DestPID and DestAddr are not present</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

**TABLE 5-3: IEEE 802.15.4™ SOURCE ADDRESSING MODES**

<table>
<thead>
<tr>
<th>SAMode b1, b0</th>
<th>Source Addressing Mode</th>
<th>SrcPID</th>
<th>SrcAddr Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>If DAMode&lt;1&gt; = 1 and PIDCmp = 1, then only 64-bit Source Long Address (SrcPID is implied by DestPID) else, 16-bit SrcPID and 64-bit Source Long Address</td>
<td>XXXXh</td>
<td>XXXX_ XXXX_ XXXX_ XXXXh</td>
</tr>
<tr>
<td>10</td>
<td>If DAMode&lt;1&gt; = 1 and PIDCmp = 1, then only 16-bit Source Short Address (SrcPID is implied by DestPID) else, 16-bit SrcPID and 16-bit Source Short Address</td>
<td>XXXXh</td>
<td>XXXXh</td>
</tr>
<tr>
<td>01</td>
<td>Reserved</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>00</td>
<td>SrcPID and SrcAddr are not present</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

On reception of a frame, each node compares its own SHADDR, ADDR, PANID Configuration, see Table 5-4, to the appropriate destination addressing fields in the received frame. A valid frame is identified if a match is found. Additionally, rules apply for broadcast frames and for implied unicast addressing as explained in the sequel.

**TABLE 5-4: RELEVANT REGISTERS FOR IEEE 802.15.4™-MODE ADDRESSING**

<table>
<thead>
<tr>
<th>ADDR. RESGISTER</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x1F</td>
<td>ADDR1</td>
<td>ADDR&lt;7:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x20</td>
<td>ADDR2</td>
<td>ADDR&lt;15:8&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x21</td>
<td>ADDR3</td>
<td>ADDR&lt;23:16&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x22</td>
<td>ADDR4</td>
<td>ADDR&lt;31:24&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x23</td>
<td>ADDR5</td>
<td>ADDR&lt;39:32&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x24</td>
<td>ADDR6</td>
<td>ADDR&lt;47:40&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x25</td>
<td>ADDR7</td>
<td>ADDR&lt;55:48&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x26</td>
<td>ADDR8</td>
<td>ADDR&lt;63:56&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x27</td>
<td>SHADDR</td>
<td>SHADDR&lt;7:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x28</td>
<td>SHADDR</td>
<td>SHADDR&lt;15:8&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x29</td>
<td>PANIDL</td>
<td>PANID&lt;7:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x2A</td>
<td>PANIDH</td>
<td>PANID&lt;15:8&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
If DAMode subfield is equal to zero and the Frame Type subfield does not specify that this frame is an acknowledgment or Beacon frame, then the SAMode subfield must be non-zero, implying that the frame is directed to the PAN coordinator with the PAN identifier as specified in the Source PAN Identifier field. This addressing option is referred to as 'implied'.

Broadcast frames of type data or command must always use DAMode = 01 and DestAddr = FFFFh.

Acknowledge frames are broadcast frames and always use DAMode = 00 and SAMode = 00.

Beacon frames are broadcast frames and always use DAMode = 00, PIDCmp = 0 with SAMode = 01 or 10. Table 5-6 and Table 5-7 show the examples for destination and source addressing, using the TX and RX node configurations in Table 5-5.

### TABLE 5-5: EXAMPLE CONFIGURATION

<table>
<thead>
<tr>
<th>TX (Source) Configuration</th>
<th>ADDR = 0x080706050403020100</th>
<th>SHADDR = 0x1211</th>
<th>PANID = 0xD2D1</th>
</tr>
</thead>
<tbody>
<tr>
<td>RX (Destination) Configuraion</td>
<td>ADDR = 0xA8A7A6A5A4A3A2A1A0</td>
<td>SHADDR = 0xB2B1</td>
<td>PANID = 0xB2B1</td>
</tr>
<tr>
<td>MHR</td>
<td>FrameCtrl</td>
<td>Sequence</td>
<td>DestPID</td>
</tr>
<tr>
<td>FrameCtrl</td>
<td>FrameCtrl&lt;7:0&gt; = 0</td>
<td>PIDCmp</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>FrameCtrl&lt;15:8&gt; = SAMode&lt;1:0&gt;</td>
<td>0</td>
<td>DAMode&lt;1:0&gt;</td>
</tr>
</tbody>
</table>

where,

*Type* is not Acknowledge and X is either of {0,1}

### TABLE 5-6: DESTINATION ADDRESSING OPTIONS (IEEE 802.15.4™) USING THE EXAMPLE

<table>
<thead>
<tr>
<th>Options</th>
<th>Broadcast</th>
<th>Unicast</th>
<th>Implied to Coordin.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DestPID</td>
<td>Command (or Data)</td>
<td>Beacon</td>
<td>Long</td>
</tr>
<tr>
<td>DestAddr</td>
<td>D1, D2</td>
<td>A1, A2, A3, A4</td>
<td>A1, A2, A3, A4</td>
</tr>
<tr>
<td>TYPE</td>
<td>000</td>
<td>xxx</td>
<td>xxx</td>
</tr>
<tr>
<td>DAMode</td>
<td>10</td>
<td>11</td>
<td>00</td>
</tr>
<tr>
<td>Address Filter</td>
<td>BCREJ</td>
<td>NOTMEREJ, UNIREJ</td>
<td></td>
</tr>
</tbody>
</table>

Note 1: DAMode = 01 is reserved and is rejected by NSTDREJ = 1.

### TABLE 5-7: SOURCE ADDRESSING OPTIONS (IEEE 802.15.4™) USING THE EXAMPLE

<table>
<thead>
<tr>
<th>Options</th>
<th>Long (Explicit SrcPID)</th>
<th>Long (Implied SrcPID)</th>
<th>Short (Explicit SrcPID)</th>
<th>Short (Implied SrcPID)</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>SrcPID</td>
<td>D1, D2</td>
<td>01, 02, 03, 04, 05, 06, 07, 08</td>
<td>D1, D2</td>
<td>11, 12</td>
<td></td>
</tr>
<tr>
<td>SrcAddr</td>
<td>01, 02, 03, 04, 05, 06, 07, 08</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TYPE</td>
<td>xxx</td>
<td>xxx</td>
<td>xxx</td>
<td>xxx</td>
<td>xxx</td>
</tr>
<tr>
<td>SAMode</td>
<td>11</td>
<td>11</td>
<td>10</td>
<td>10</td>
<td>00</td>
</tr>
<tr>
<td>DAMode</td>
<td>xx</td>
<td>1x</td>
<td>xx</td>
<td>1x</td>
<td>xx</td>
</tr>
<tr>
<td>PIDCmp</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>x</td>
</tr>
</tbody>
</table>

Note 1: SAMode = 01, is reserved, and is rejected by NSTDREJ = 1.

The valid address formats are summarized in Table 5-8 for all frame types. Broadcast and unicast cases are distinguished in the case of command and data frames. Unicast frames are either addressed to the receiving node or to a different node. UNIREJ and NOTMEREJ are sensitive to the former or the latter case, respectively. Broadcast command and data frames are filtered when BCREJ is set. The parser does not filter the Beacon frames.
### REGISTER 5-2: RXFILTER (RX FILTER) – WHEN IEEE 802.15.4™ MODE

<table>
<thead>
<tr>
<th>Bit 7-4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
</tr>
</thead>
</table>

#### Legend:
- **R** = Readable bit
- **W** = Writable bit
- **U** = Unimplemented bit, read as '0'
- **-n** = Value at POR
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- **x** = Bit is unknown
- **r** = Reserved

### bit 7-4
Out of scope

### bit 3
#### UNIREJ:
Setting this bit enables the user to reject all unicast packets as in:

**802.15.4 Mode:** PAN Identifier matches with the PANID<15:0> or 0xFFFF, and Destination Address matches the address in the ADDR<63:0> or SHADDR<15:0> register, which the DAMode selects.

**Proprietary Mode:** Destination Address matches the address in ADDR<ADDRSZ<2:0>*8-1:0> register, provided that DAddrPrsnt Frame Control field is set(1).

- 1 = Reject all Unicast packets addressed to this node
- 0 = Disable Unicast Rejection

### bit 2
#### NOTMEREJ:
Setting this bit enables the user to reject all unicast packets as in:

**802.15.4 Mode:** Destination PAN Identifier does not match PANID<15:0> and is not 0xFFFF (broadcast) or Destination Address does not match the address in the ADDR<63:0> register or the SHADDR<15:0> register, which the DAMode selects.

**Proprietary Mode:** Destination Address matches the address in ADDR<ADDRSZ<2:0>*8-1:0> register, provided that DAddrPrsnt Frame Control field is set(1).

- 1 = Reject all Unicast packets NOT addressed to this node
- 0 = Disable Not Me Unicast Rejection Filtering

### bit 1
#### BCREJ:
**802.15.4 Mode:** Setting this bit enables the user to reject all Broadcast packets of type Data or Command. A Data or Command packet is broadcast when Short Destination Addressing is used (DAMode = 10) and Short Address is equal 0xFFFF.

**Proprietary Mode:** Setting this bit enables the user to reject all Broadcast packets of type Data or Command (or Streaming). A packet is broadcast when FrameCtrl[Broadcast] is set.

- 1 = Reject Broadcast Packets
- 0 = Disable Broadcast Rejection

### bit 0
#### NSTDREJ:
This bit enables the user to reject all 802.15.4 frames having 01 for the DAMode or SAMode fields or having the most significant bit (MSb) (bit 2) in the Type field set (1) or having the MSb (bit 1) in the Frame Version field set(1).

- 1 = Reject all Non-Standard 802.15.4 packets
- 0 = Disable Non-Standard Rejection

#### Note 1:
In Proprietary mode (FRMFMT = 1), when CRCREJ = 1 is used to reject unicast frames not addressed to this node. NOTMEREJ = 1 does not reject these frames.

#### Note 2:
UNIREJ does not affect the frames using implied destination addressing in 802.15.4 mode and inferred destination addressing in Proprietary mode.

#### Note 3:
NOTMEREJ does not affect the frames using implied destination addressing in 802.15.4 mode and inferred destination addressing in Proprietary mode.

#### Note 4:
NSTDREJ does not affect the Proprietary frames in Proprietary mode.
When a valid frame gets filtered, RXFLTIF is set, otherwise, RXIF terminates the successful reception. For more information, refer to Register 5-1.

Invalid addressing formats are produced if:

- Either DAMode or SAMode are set to the reserved value of '01'.
- DAMode or SAMode values are used with an incompatible Type field value. For example,
  - Beacon with DAMode = 1x
  - DAMode = 00, SAMode = 00 used with Type of Beacon/Command / Data.
- PIDCmp is set on an inconsistent way to DAMode and SAMode.
- SrcPID or SrcAddr holds 'FFFF', or if DestPID holds 'FFFF' while DAMode = 11.
- LENGTH field is less than the MHR length computed from FrameCtrl.

The device checks the first condition and FRMIF is generated. The device does not check the second condition, therefore one out of RXIF, RXFLTIF, FRMIF is expected. The hardware checks the third condition and PIDCmp value is handled as 0. The hardware does not check the fourth condition and RXIF is expected. All other invalid formats also produces one out of RXIF, RXFLTIF, and FRMIF.

<table>
<thead>
<tr>
<th>DA</th>
<th>PID COMP</th>
<th>SA</th>
<th>DEST(4)</th>
<th>TYPES(3)</th>
<th>Field Sizes in Octets(4)</th>
<th>Description(5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0(1)</td>
<td>00</td>
<td>BC3</td>
<td>A</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0(1)</td>
<td>00</td>
<td>BC1</td>
<td>C,D</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>0(1)</td>
<td>00</td>
<td>BC2</td>
<td>C,D</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>0(1)</td>
<td>00</td>
<td>UNI</td>
<td>C,D</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>0(1)</td>
<td>00</td>
<td>NOTME</td>
<td>C,D</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>0(1)</td>
<td>00</td>
<td>UNI</td>
<td>C,D</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>11</td>
<td>0(1)</td>
<td>00</td>
<td>NOTME</td>
<td>C,D</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>00</td>
<td>0(1)</td>
<td>10</td>
<td>BC3</td>
<td>B</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>00</td>
<td>0(1)</td>
<td>10</td>
<td>UNI2, PANCN</td>
<td>C,D</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>00</td>
<td>0(1)</td>
<td>10</td>
<td>NOTME2, PANCN</td>
<td>C,D</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>00</td>
<td>0(1)</td>
<td>11</td>
<td>BC3</td>
<td>B</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>00</td>
<td>0(1)</td>
<td>11</td>
<td>—</td>
<td>C,D</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>00</td>
<td>0(1)</td>
<td>11</td>
<td>NOTME2, PANCN</td>
<td>C,D</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0(1)</td>
<td>10</td>
<td>BC1</td>
<td>C,D</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>0(1)</td>
<td>10</td>
<td>BC2</td>
<td>C,D</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>10</td>
<td>UNI</td>
<td>C,D</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>10</td>
<td>NOTME</td>
<td>C,D</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>10</td>
<td>BC1</td>
<td>C,D</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>10</td>
<td>UNI</td>
<td>C,D</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>10</td>
<td>NOTME</td>
<td>C,D</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>11</td>
<td>BC1</td>
<td>C,D</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>11</td>
<td>BC2</td>
<td>C,D</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>11</td>
<td>UNI</td>
<td>C,D</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>11</td>
<td>NOTME</td>
<td>C,D</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>11</td>
<td>BC1</td>
<td>C,D</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>11</td>
<td>UNI</td>
<td>C,D</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
5.3 Security Material

The security material required for CBC-MAC, CTR, and CCM are the inputs configured to the registers as listed in Table 5-10.

- SECSUITE<3:0> selects the security suite consisting of encryption or authentication, or both, see Table 5-9.
- SECHDRINDX<6:0> is the byte index where authentication must start.
- SECPAYINDX<6:0> is the byte index where encryption/decryption must start.
- SECENDINDX<6:0> points at the last byte of the payload (before MIC and FCS).
- SECKEY<127:0> holds the symmetric Key.
- SECNONCE<103:0> holds a Nonce value that is unique for each frame while a specific Key is in use. This ensures sequence freshness (for protection against repeat-attack) and protects the key from being deciphered based on the encoded messages. The transmitter generates the information required to generate the Nonce and then sends to the Receiver as plain text as part of the frame.

Section 5.4 “Security Material Retrieval with IEEE 802.15.4 Compliant Frames” describes how the security level is selected and whether the device or software fills out the above registers before the security operation is launched. DEVICE/HOST fills in these registers and the Authentication appends a MIC tag to the frame (before FCS is appended), after the position pointed at by SECENDINDX. Encryption/decryption alters the “payload” stored in the buffer from SECPAYINDX through SECENDINDX. The range defined for “Payload” does not necessarily coincide with the MAC payload as explained in the sequel.

Figure 5-2 to Figure 5-9 illustrate the order of all the security operations, which is valid for both 2003/2006 Compliant Framing modes.

5.4 Security Material

The security material required for CBC-MAC, CTR, and CCM are the inputs configured to the registers as listed in Table 5-10.

- SECSUITE<3:0> selects the security suite consisting of encryption or authentication, or both, see Table 5-9.
- SECHDRINDX<6:0> is the byte index where authentication must start.
- SECPAYINDX<6:0> is the byte index where encryption/decryption must start.
- SECENDINDX<6:0> points at the last byte of the payload (before MIC and FCS).
- SECKEY<127:0> holds the symmetric Key.
- SECNONCE<103:0> holds a Nonce value that is unique for each frame while a specific Key is in use. This ensures sequence freshness (for protection against repeat-attack) and protects the key from being deciphered based on the encoded messages. The transmitter generates the information required to generate the Nonce and then sends to the Receiver as plain text as part of the frame.

### Table 5-9: Security Level: Mode of Operation

<table>
<thead>
<tr>
<th>Security Level(1)</th>
<th>Payload</th>
<th>MIC Tag of Octets</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>Plain text</td>
<td>No Authentication</td>
<td>—</td>
</tr>
<tr>
<td>0001</td>
<td>Plain text</td>
<td>4 bytes</td>
<td>CCM operation. Only defined in 2006.</td>
</tr>
<tr>
<td>0010</td>
<td>Plain text</td>
<td>8 bytes</td>
<td>CCM operation. Only defined in 2006.</td>
</tr>
<tr>
<td>0011</td>
<td>Plain text</td>
<td>16 bytes</td>
<td>CCM operation. Only defined in 2006.</td>
</tr>
</tbody>
</table>
### TABLE 5-9: SECURITY LEVEL: MODE OF OPERATION (CONTINUED)

<table>
<thead>
<tr>
<th>Security Level(1)</th>
<th>Payload</th>
<th>MIC Tag of Octets</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0100</td>
<td>Encrypted</td>
<td>No Authentication</td>
<td>CCM operation. Only defined in 2006.</td>
</tr>
<tr>
<td>1000</td>
<td>Encrypted</td>
<td>No Authentication</td>
<td>ECB operation. Not defined in 2003/2006 (only encryption)</td>
</tr>
<tr>
<td>1001</td>
<td>Encrypted</td>
<td>No Authentication</td>
<td>CTR operation. Only defined in 2003.</td>
</tr>
<tr>
<td>1010</td>
<td>Reserved</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>1011</td>
<td>Reserved</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>1100</td>
<td>Reserved</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>1101</td>
<td>Plain text</td>
<td>16 bytes</td>
<td>CBC-MAC operation. Only defined in 2003.</td>
</tr>
<tr>
<td>1110</td>
<td>Plain text</td>
<td>8 bytes</td>
<td>CBC-MAC operation. Only defined in 2003.</td>
</tr>
<tr>
<td>1111</td>
<td>Plain text</td>
<td>4 bytes</td>
<td>CBC-MAC operation. Only defined in 2003.</td>
</tr>
</tbody>
</table>

**Note 1:** In 2006 compliant framing, the security level is traveling with the frame, while in 2003 it must be set globally.

### TABLE 5-10: SECURITY MATERIAL INPUTS TO CBC-MAC, CTR AND CCM

<table>
<thead>
<tr>
<th>ADDR.</th>
<th>REGISTER</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x10</td>
<td>MACCON1</td>
<td>TRXMODE&lt;1:0&gt;</td>
<td>ADDRZ&lt;2:0&gt;</td>
<td>CRCSZ</td>
<td>FRMFMT</td>
<td>SECFLAGOVR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x11</td>
<td>MACCON2</td>
<td>CHANNEL&lt;3:0&gt;</td>
<td>SECSUITE&lt;3:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x2B</td>
<td>SECHRDINDEX</td>
<td></td>
<td>SECHRDINDEX&lt;6:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x2C</td>
<td>SECPAYINDEX</td>
<td></td>
<td>SECPAYINDEX&lt;6:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x2D</td>
<td>SECENDINDEX</td>
<td></td>
<td>SECENDINDEX&lt;6:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x40 through 0x4F</td>
<td>SECKEY1</td>
<td></td>
<td></td>
<td>SECKEY&lt;7:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;2,3,4,...,15&gt;</td>
<td></td>
<td></td>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SECKEY16</td>
<td></td>
<td></td>
<td>SECKEY&lt;127:120&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x50 through 0x5C</td>
<td>SECNONCE1</td>
<td></td>
<td>SECNONCE&lt;7:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;2,3,4,...,12&gt;</td>
<td></td>
<td></td>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SECNONCE13</td>
<td></td>
<td>SECNONCE&lt;103:96&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x5D</td>
<td>SECENCFLAG</td>
<td></td>
<td>SECENCFLAG&lt;7:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x5E</td>
<td>SECAUTHFLAG</td>
<td></td>
<td></td>
<td>SECAUTHFLAG&lt;7:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
FIGURE 5-5: CCM*/CBC-MAC AUTHENTICATION OPERATION (TX)

Exception Handling:

**TXSZIF:** Transmit Packet Size Error Interrupt Flag

TXST is set when the packet size (including MIC tags and CRC) is found to be zero or to be greater than the maximum size that the buffers can support.

**FRMIF:** Frame Format Error Interrupt Flag

Set if the transmitter/receiver fails to parse the frame in the buffer (it is not as it must be or it is corrupted in demodulation). For example, reserved values are found in the MAC header fields.
**FIGURE 5-6: CCM*/CBC-MAC DE-AUTHENTICATION OPERATION (RX)**\(^{(1, 2)}\)

<table>
<thead>
<tr>
<th>Received from the air</th>
<th>Length</th>
<th>MHR</th>
<th>MAC Payload</th>
<th>Tag</th>
<th>CRC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**RX Buffer**

<table>
<thead>
<tr>
<th>Length</th>
<th>MHR</th>
<th>Data/CMD Payload</th>
<th>Tag</th>
<th>CRC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note 1:** The Length field above refers to the total number of octets as reported by the Baseband, and therefore includes the Tag (MIC), but excludes any RSV octets.

**2:** If present, RSV octets are placed after the Tag, as these are only received when the complete frame is received. It is the responsibility of software to determine the address of the RSV in the buffer (RSV Address = Length + 1), and to discard/ignore the Tag octets.
FIGURE 5-7: CCM*/CTR/ECB ENCRYPTION OPERATION (TX)

Exception handling:

**TXSZIF:** Transmit Packet Size Error Interrupt Flag
TXST is set when the packet size (including MIC tags and CRC) is found to be zero or to be greater than the maximum size that the buffers can support.

**FRMIF:** Frame Format Error Interrupt Flag
Set if the transmitter/receiver fails to parse the frame in the buffer (it is not as it must be or it is corrupted in demodulation). For example, reserved values are found in the MAC header fields.
FIGURE 5-8: CCM*/CTR/ECB DECRYPTION OPERATION (TX)\(^{(1, 2)}\)

<table>
<thead>
<tr>
<th>Exception Handling:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RXTAGIF</strong>: Receiver Decryption/Authentication Failure Interrupt Flag</td>
</tr>
<tr>
<td>Set by the device when decryption/authentication finished with error.</td>
</tr>
<tr>
<td><strong>Note 1</strong>: The Length field above refers to the total number of octets as reported by the Baseband, and therefore includes the Tag (MIC), but excludes any RSV octets.</td>
</tr>
<tr>
<td><strong>Note 2</strong>: If present, RSV octets are placed after the Tag, as these are only received when the complete frame is received. It is the responsibility of software to determine the address of the RSV in the buffer (RSV Address = Length + 1), and to discard/ignore the Tag octets.</td>
</tr>
</tbody>
</table>
FIGURE 5-9: CCM* ENCRYPTION AND AUTHENTICATION OPERATION (TX)

Exception handling:

TXSZIF: Transmit Packet Size Error Interrupt Flag
TXST is set when the packet size (including MIC tags and CRC) is found to be zero or to be greater than the maximum size that the buffers can support.

FRMIF: Frame Format Error Interrupt Flag
Set if the transmitter/receiver fails to parse the frame in the buffer (it is not as it must be or it is corrupted in demodulation). For example, reserved values are found in the MAC header fields.
Exception Handling:

**RXTAGIF:** Receiver Decryption/Authentication Failure Interrupt Flag
Set by the device when decryption/authentication finished with error.

**Note 1:** The Length field above refers to the total number of octets as reported by the Baseband, and therefore includes the Tag (MIC), but excludes any RSV octets.

**2:** If present, RSV octets are placed after the Tag, as these are only received when the complete frame is received. It is the responsibility of software to determine the address of the RSV in the buffer (RSV Address = Length + 1), and to discard/ignore the Tag octets.

**3:** The Message and Tag decryption operations do not depend on each other, and may be computed in any order. All other factors being equal, the Tag decryption operation must be performed first, as it uses the starting counter value.
5.4 Security Material Retrieval with IEEE 802.15.4 Compliant Frames

This section explains how the security material, see Section 5.3 “Security Material”, is retrieved when the MAC frame is formatted to the IEEE 802.15.4 specification (either FRMFMT = 0 or “bridging”) and security is applied either at the MAC layer or at the NWK layer, or both.

Table 5-10 indicates the relevant Configuration registers (SECSUITE<3:0>). Figure 5-11 represents the relevant security fields (SecEn, SecLvl<2:0>, FrameVer<1:0>, FrameCnt, KeyIDMode, KeySrc, KeyIndex).

The IEEE 802.15.4 standard enables five different security scenarios. The difference among these scenarios are shown in Table 5-13. If SecEn = 1, then the MAC layer security is enabled. In case NWK layer security is also enabled, then it is calculated before MAC layer security.

MAC layer security material retrieval differs in the 2006 and the 2003 versions of the standard. Distinction is possible based on the FrameVer<1:0> field. The device does not support the Security Material retrieval for Beacon frames. The Type<2:0> field distinguishes the Beacon frames.

KeyIDMode, KeySrc, KeyIndex in the AuxSecHdr are done by software for the retrieval of the MAC layer Symmetric Key, the details are out of scope.

If 2006-MAC layer security is applied, use the FrameCnt, SecLvl, and 8-byte source address to construct the Nonce, see Figure 5-12. If 2003 MAC layer security is applied, use the FrameCnt, KeySeqCnt, and source address to construct the Nonce field, see Figure 5-13. When the frame contains a short source address, the device cannot set the Nonce correctly. Similarly, if the frame is of Type = Beacon, then the SECPAYINDEX is set incorrectly. In these cases, the registers must be configured from software. On the RX side, this is easily done before launching the security processing (RXDEC = 1). On the TX-side, set DTSM to prevent the device from overwriting the Nonce and Indexes that the software configures.

Only the MAC layer security contains specific indexes as the Network layer security configures the software:

In MAC layer security, SECHDRINDEX is always the first byte of the MHR.

In MAC layer security applied for frames of type Data and Streaming, SECPAYINDEX is the first byte of the payload. For Command frames, SECPAYINDEX is the second byte of the payload. SECPAYINDEX can take different values for Beacon frames, which the software must always specify.
REGISTER 5-3: SECHDRINDX (SECURITY HEADER INDEX REGISTER)  ADDRESS: 0x2B

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Reserved</td>
</tr>
</tbody>
</table>
| 6-0  | SECHDRINDX<6:0>: Security Header Index bits  
This field defines the portion of the header which performs the authentication operations.  
For MAC layer security, SECHDRINDX<6:0> is defined as the address offset of the MAC header from the beginning of the frame, as stored in the buffer (0 = Length field, 1 = FrameCtrl field, and so on), and is automatically loaded for both 802.15.4 and Proprietary frames.  
For Network layer security, SECHDRINDX<6:0> is defined as the address offset of the Network Header from the beginning of the MAC Payload, and the Host Controller loads it only for 802.15.4 frames. Note that for Proprietary frames, the MAC automatically loads it\(^{(1)}\). |

**Note 1:** The setting DTSM in TX mode disables automatic computation of this field.

---

REGISTER 5-4: SECPAYINDX (SECURITY PAYLOAD INDEX REGISTER)  ADDRESS: 0x2C

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Reserved</td>
</tr>
</tbody>
</table>
| 6-0  | SECPAYINDX<6:0>: Security Payload Index bits  
This field defines the portion of the payload which performs the Encryption/Decryption operations.  
For MAC layer security, SECPAYINDX<6:0> is defined as the address offset of the MAC Payload from the beginning of the frame, as stored in the buffer (0 = Length field, 1 = FrameCtrl field, and so on), and is automatically loaded for both 802.15.4 and Proprietary frames.  
For Network layer security, SECPAYINDX<6:0> is defined as the address offset of the Network Header from the beginning of the MAC Payload, and the Host Controller loads it only for 802.15.4 frames. Note that for Proprietary frames, the MAC automatically loads it\(^{(1)}\). |

**Note 1:** The setting DTSM in TX mode disables automatic computation of this field.
**REGISTER 5-5: SECENDINDX (SECURITY END INDEX REGISTER) **

ADDRESS: 0x2D

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>R/W/HS</th>
<th>Value at POR</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Reserved</td>
<td>R</td>
<td>'0'</td>
</tr>
<tr>
<td>6-0</td>
<td>SECENDINDX&lt;6:0&gt;</td>
<td>W</td>
<td>'0'</td>
</tr>
</tbody>
</table>

Legend:  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
-n = Value at POR  
'1' = Bit is set  
'0' = Bit is cleared  
x = Bit is unknown  
r = Reserved  
HS = Hardware Set

Note 1: The setting DTSM in TX mode disables automatic computation of this field.

This field defines the end of the payload which performs the security operations.

Note 1: The setting DTSM in TX mode disables automatic computation of this field.
Figure 5-11 illustrates the construction of the Nonce in 802.15.4-mode.

**FIGURE 5-11: IEEE.802.15.4™ SECURITY CONTROL FIELDS**

```
<table>
<thead>
<tr>
<th>MHR: MAC Header (IEEE 802.15.4™)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FrameCtrl (2 octets)</td>
</tr>
<tr>
<td>Sequence (1 octet)</td>
</tr>
<tr>
<td>DestPID (0/2 octets)</td>
</tr>
<tr>
<td>DestAddr (0/2/8 octets)</td>
</tr>
<tr>
<td>SrcPID (0/2 octets)</td>
</tr>
<tr>
<td>SrcAddr (0/2/8 octets)</td>
</tr>
<tr>
<td>AuxSecHdr (0-14 octets)</td>
</tr>
<tr>
<td>Destination (0/4/10 octets)</td>
</tr>
<tr>
<td>Source(0/2/4/8/10 octets)</td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>FrameCtrl (IEEE 802.15.4 2003)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Byte 0</td>
</tr>
<tr>
<td>7:6</td>
</tr>
<tr>
<td>Rsvd. (1 bit)</td>
</tr>
<tr>
<td>0</td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>AuxSecHdr (IEEE 802.15.4™–2006)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SecCtr 1 octet</td>
</tr>
<tr>
<td>FrameCnt 4 octets</td>
</tr>
<tr>
<td>KeySrc 0/4/8 octets</td>
</tr>
<tr>
<td>KeyIndex 0/1 octet</td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>AuxSecHdr (IEEE 802.15.4™–2003)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FrameCnt 4 octets</td>
</tr>
<tr>
<td>KeySeqCnt 1 octet</td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>SecCtr (IEEE 802.15.4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7:5</td>
</tr>
<tr>
<td>Reserved (3 bits)</td>
</tr>
<tr>
<td>00</td>
</tr>
<tr>
<td>01</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>11</td>
</tr>
</tbody>
</table>
```

8-byte SrcAddr

NONCE<103:0>

SECSUITE<2:0> in MACCON2 (SECSUITE<3> ← 0)
FIGURE 5-12: 802.15.4 CCM NONCE (ONLY MAC LAYER SECURITY)-2006

<table>
<thead>
<tr>
<th>FrameCtr (4 Octets)</th>
<th>(5'b00000, SecLvl) (1 Octet)</th>
</tr>
</thead>
<tbody>
<tr>
<td>13 Octets</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** The originator device automatically fills in SrcAddr field using the values of registers ADDR8 through ADDR1 irrespective of the SAMode. The recipient host must fill in the nonce if SAMode is different from ‘11’.

---

FIGURE 5-13: 802.15.4 CCM NONCE (ONLY MAC LAYER SECURITY)-2003

<table>
<thead>
<tr>
<th>KeySeqCnt (1 Octet)</th>
</tr>
</thead>
<tbody>
<tr>
<td>13 Octets</td>
</tr>
</tbody>
</table>

**Note:** The originator device automatically fills in SrcAddr field using the values of registers ADDR8 through ADDR1 irrespective of the SAMode. The recipient host must fill in the nonce if SAMode is different from ‘11’.
5.5 Transmit Security Processing of IEEE 802.15.4 Compliant Frames

Setting TXST triggers automatic MAC layer security processing and frame sending as an uninterrupted sequence, see Figure 5-14. (network layer) which is triggered by TXENC is applied, where TXENCIF must be awaited before other operation.

BUF1TXPP, BUF2TXPP, TXENC, and TXST can trigger security functions as shown in Figure 5-12 and Figure 5-14, where BUF1TXPP and BUF2TXPP are both used for debug. The respective interrupts are generated on completion and the device automatically clears the aforementioned triggering bits. In Figure 5-12, the device shows the conditions for security material retrieval and the operation of the DTSM bit.

FIGURE 5-14: TRANSMIT SECURITY PROCESSING WHEN FRMFMT = 0 (IEEE 802.15.4™ FORMAT)

Before launching the transmit processing (Figure 4-9):
• SW always configures SECKEY
• SW may configure SECSUITE, SECNONCE, SECHDRINDX, SECPAYINDEX, SECENDINDEX

BUF1TXPP, BUF2TXPP, TXENC, and TXST can trigger security functions as shown in Figure 5-12 and Figure 5-14, where BUF1TXPP and BUF2TXPP are both used for debug. The respective interrupts are generated on completion and the device automatically clears the aforementioned triggering bits. In Figure 5-12, the device shows the conditions for security material retrieval and the operation of the DTSM bit.
FIGURE 5-15: TRANSMITTER TXENC PROCESSING WHEN FRMFMT = 0 (IEEE 802.15.4™ FORMAT)

Length is affected. MAC MAC/NWK.

Exception Handling:

**TXSZIF**: Transmit Packet Size Error Interrupt Flag

TXST is set when the packet size (including MIC tags and CRC) is found to be zero or to be greater than the maximum size that the buffers can support.

**FRMIF**: Frame Format Error Interrupt Flag

SW configured SECKEY, SECSUITE, SECNONCE, SECHDRINDX, SECPAYINDEX and SECENDINDEX.

TXENC is only required for NWK layer security processing.

LENGTH is incremented when a MIC tag is attached. New LENGTH must not exceed 0x7F. Otherwise TXSZIF is set.
5.6 Security Processing of Received IEEE 802.15.4 Compliant Frames

Receive security is always performed when RXDEC is set and awaits RXDECIF or RXTAGIF as it is not automatically triggered. If both MAC and NWK layer security are applied, then both must be processed in this order when RXDEC is set for a second time after updating the security material correctly.

Figure 5-17 shows the security functions triggered by RXDEC. The device automatically clears the respective interrupts generated on completion and RXDEC.

FIGURE 5-16: SECURITY MATERIAL RETRIEVAL SUPPORT IN RECEIVE PROCESSING WHEN FRMFMT = 0 (IEEE 802.15.4™ FORMAT)

After RXIF is asserted:
- SW always configures SECKEY.
- SW may need to configure SECSUITE, SECNONCE, SECHDRINDX, SECPAYINDX, SECENDINDX.

Length is affected. MAC MAC/NWK.

Exception Handling:
FRMFIF: Frame Format Error Interrupt Flag
The bit is set if the transmitter/receiver fails to parse the frame in the buffer (it is not as it must be or it is corrupted in demodulation). For example, reserved values found in the MAC header fields.
The valid frame is available in BUF2 in Mission mode, or in the buffer selected by BUF1RXPP or BUF2RXPP during debug. Device and SW have both parsed the frame. Figure 5-16 illustrates the security material retrieval support. SW has configured SECKEY as required for the frame, and can also overwrite the configurations in SECSUITE, SECNONCE, SECHDRINDX, SECPAYINDX, SECENDINDX.

**FIGURE 5-17: RECEIVER RXDEC PROCESSING WHEN FRMFMT = 0 (IEEE 802.15.4™-MODE)**

- RXDEC ← 1
- AES-CCM
- MIC tag mismatch?
  - Authentication failed
  - RXDEC ← 0
  - RXDECIF ← 1
  - RXTAGIF ← 1
- Success
- Interrupt Service
5.7 Security Procedure for IEEE 802.15.4 Compliant Frames

For more information about the frame format, refer to Table 5-12 through Table 5-14 and Figure 5-3.

**TABLE 5-11: RELEVANT REGISTER BITS FOR SECURITY CONTROL WITH IEEE 802.15.4™ FRAMES**

<table>
<thead>
<tr>
<th>Name</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>MACCON1</td>
<td>TRXMODE&lt;1:0&gt;</td>
<td>ADDRSZ&lt;2:0&gt;</td>
<td>CRCSZ</td>
<td>FRMFMT</td>
<td>SECFLAGOVR</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend:  
- *r* = Reserved, read as ‘0’.

**TABLE 5-12: DEFINITION OF SECURITY SUPPORT CATEGORIES (IEEE 802.15.4™ FRAMES)**

<table>
<thead>
<tr>
<th>Security Support Category</th>
<th>SecEn</th>
<th>FrameVer &lt;1:0&gt;</th>
<th>Type&lt;2:0&gt; (and SAMode)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0x</td>
<td>Data/CMD/Beacon/ACK</td>
<td>No security</td>
</tr>
<tr>
<td>A</td>
<td>1</td>
<td>0x</td>
<td>Data/CMD</td>
<td>2003/2006 MAC layer security</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>0x</td>
<td>Beacon</td>
<td>2003/2006 MAC layer secured beacon</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0x</td>
<td>Data/(CMD)/Beacon</td>
<td>NWK layer security</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>0x</td>
<td>Data/CMD</td>
<td>NWK + 2003/2006 MAC layer security</td>
</tr>
<tr>
<td>E</td>
<td>1</td>
<td>0x</td>
<td>Beacon</td>
<td>NWK + 2003/2006 MAC layer security for Beacon frames</td>
</tr>
</tbody>
</table>
### TABLE 5-13: SECURED FRAME TRANSMISSION (IEEE 802.15.4™ MAC FORMAT)

<table>
<thead>
<tr>
<th>#</th>
<th>Processing Step</th>
<th>Steps per each Security Case</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>A</td>
</tr>
<tr>
<td>1</td>
<td>Host MCU constructs the frame and loads the buffer.</td>
<td>SecEn = 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FrameVer = 0x (either)</td>
</tr>
<tr>
<td>2</td>
<td>For NWK security processing, Host MCU configures:</td>
<td>No NWK layer security.</td>
</tr>
<tr>
<td>3</td>
<td>Host MCU triggers security processing without sending.</td>
<td>TXENC ← 1</td>
</tr>
<tr>
<td>4</td>
<td>Device performs the security processing for NWK layer if TXENC is set. LENGTH and SECENDINDX are updated if MIC tag is appended. TXSZIF if size runs over 127 bytes.</td>
<td>TXENCIF ← 1</td>
</tr>
<tr>
<td>5</td>
<td>Host MCU awaits TXENCIF interrupt, indicating completion. Device clears TXENC.</td>
<td>SECKEY (+NONCE, if SAMode is not 11) (+SECSUITE, if FRAMEVER=2003)</td>
</tr>
<tr>
<td>6</td>
<td>For MAC security processing, Host MCU configures:</td>
<td>DTSM = 0 (=1, if SAMode is not 11)</td>
</tr>
<tr>
<td>7</td>
<td>Host MCU sets DTSM to inhibit the hardware from overwriting just configured SECSUITE, SEC*INDX and NONCE registers.</td>
<td>TXST ← 1</td>
</tr>
<tr>
<td>8</td>
<td>Host MCU triggers Security processing and sending.</td>
<td>MAC layer security LENGTH, if MIC added</td>
</tr>
<tr>
<td>9</td>
<td>If SecEn = 1 and DTSM = 0, then the device configures the SECSUITE, SEC*INDX and NONCE registers.</td>
<td>SECSUITE SEC*INDX NONCE</td>
</tr>
<tr>
<td>10</td>
<td>Device performs the security processing for MAC layer: LENGTH is adjusted if MIC tag is appended. TXSZIF if size runs over 127 bytes.</td>
<td>LENGTH is adjusted as CRC is appended (if CRCSZ = 1). TXSZIF if size runs over 127 bytes.</td>
</tr>
<tr>
<td>11</td>
<td>LENGTH is adjusted as CRC is appended (if CRCSZ = 1). TXSZIF if size runs over 127 bytes.</td>
<td>TXIF (if no TXSZIF or FRMIF)</td>
</tr>
</tbody>
</table>

**Note 1:** SEC*INDX denotes SECHDRINDX, SECPAYINDX and SECENDINDX.
### TABLE 5-14: SECURED FRAME RECEPTION (IEEE 802.15.4™ MAC FORMAT)

<table>
<thead>
<tr>
<th>#</th>
<th>Processing Step</th>
<th>Steps per each security case</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>A</td>
</tr>
<tr>
<td>-2</td>
<td>Device parses the SecEn bit in the FrameCtrl.</td>
<td>SecEn = 1</td>
</tr>
<tr>
<td>-1</td>
<td>For MAC security processing, the device configures (correctly or incorrectly) the following:</td>
<td>SECSUITE SEC*INDX NONCE</td>
</tr>
<tr>
<td>0</td>
<td>Valid frame received on air and accepted by RXFILTER.</td>
<td>RXIF = 1, RXBUFFUL = 1, (RXSFDIF = 1)</td>
</tr>
<tr>
<td>1</td>
<td>Host MCU has the opportunity to check the SecEn, FrameVer and SAMode bits in the MAC header.</td>
<td>FrameVer = 0x (either)</td>
</tr>
<tr>
<td>2</td>
<td>For MAC security processing, the Host MCU must load the following:</td>
<td>SECKEY (+NONCE if SAMode is not 11)</td>
</tr>
<tr>
<td>3</td>
<td>Host MCU starts MAC security processing by setting RXDEC.</td>
<td>RXDEC ← 1</td>
</tr>
<tr>
<td>4</td>
<td>Device performs MAC layer security processing as illustrated in Figure 5-4 through Figure 5-9.</td>
<td>MAC layer</td>
</tr>
<tr>
<td>5</td>
<td>If Authentication fails then RXTAGIF is generated. Otherwise, the security operation is successful and RXDECIF is generated.</td>
<td>RXDECIF (or RXTAGIF)</td>
</tr>
<tr>
<td>6</td>
<td>SW examines RXTAGIF, if set, SW aborts further processing and frees the buffer by clearing RXBUFFUL.</td>
<td>RXTAGIF ← 1</td>
</tr>
<tr>
<td>7</td>
<td>For NWK security processing, the Host MCU must load the following:</td>
<td>No NWK layer security</td>
</tr>
<tr>
<td>8</td>
<td>Host MCU starts NWK security processing by setting RXDEC.</td>
<td>RXDEC ← 1</td>
</tr>
<tr>
<td>9</td>
<td>Device performs NWK layer security processing. (No figure)</td>
<td>NWK layer security</td>
</tr>
<tr>
<td>10</td>
<td>If Authentication fails, RXTAGIF is generated. Otherwise, the security operation is successful and RXDECIF is generated. The device clears RXDEC.</td>
<td>RXDECIF (or RXTAGIF) RXDEC ← 0</td>
</tr>
<tr>
<td>11</td>
<td>SW examines RXTAGIF, if set, SW aborts further processing and frees the buffer by clearing RXBUFFUL. For the length</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>SW reads the entire frame from the buffer.</td>
<td>—</td>
</tr>
<tr>
<td>13</td>
<td>SW clears the RXBUFFUL to free the buffer.</td>
<td>—</td>
</tr>
</tbody>
</table>
5.8 Security Examples

The following section provides examples for the usage of MRF24XA security.

5.8.1 802.15.4-2006 COMPLIANT FRAME

ANNEX C.2.2 (TYPE A)

Configuration:
- Network Configuration: Extended address, PAN Compression, and ACKReq
- Source address: 0xACDE480000000001, where 01 is at address 0x1F
- Destination address: 0xACDE480000000002
- PANID 0x4321, where 21 is at address 0x29
- Payload: 61 62 63 64
- Frame counter: 0x00000005
- Security level: 0x04
- Packet: Data packet.

5.8.1.1 Transmission

For 802.15.4-2006 compliant, follow these transmission flow:

1. Host MCU constructs the frame and loads the buffer:
   1E || 69 DC 84 21 43 02 00 00 00 00 48 DE AC 01 00 00 00 00 48 DE AC || 04 05 00 00 00 || 61 62 63 64
2. —
3. —
4. —
5. —
6. Host MCU configures SECKEY.
   0xC0C1C2C3C4C5C6C7C8C9CACBCCCD-CECF, where LSB (0xCF) is at address 0x40
7. Host MCU clears DTSM.
8. Host MCU issues TXST.
9. MRF24XA configures:
   - SECSUITE to 0x04
   - SECNONCE to 0xAC-DE48000000001000000000504, where MSB (0xAC) is at address 0x5C
   - SECHDRINDX to 0x01
   - SECPAYINDX to 0x1B
   - SECENDINDX to 0x1E.
10. MRF24XA performs CCM* encryption, where 61 62 63 64 is encrypted to D4 3E 02 2B.
11. MRF24XA appends CRC: 0x18E0.
12. MRF24XA transmits the packet to the medium. MRF24XA is waiting for an ACKnowledge frame. Different IF is received based on the register settings (for example, TX with CSMA). TX Buffer (0x200) content:
   20 || 69 DC 84 21 43 02 00 00 00 00 48 DE AC 01 00 00 00 00 48 DE AC || 04 05 00 00 00 || D4 3E 02 2B || E0 18

5.8.1.2 Reception

1. MRF24XA receives the following packet through the antenna:
   20 || 69 DC 84 21 43 02 00 00 00 00 48 DE AC 01 00 00 00 00 48 DE AC || 04 05 00 00 00 || D4 3E 02 2B || E0 18
2. MRF24XA configures:
   - SECSUITE to 0x04
   - SECNONCE to 0xAC-DE48000000001000000000504, where MSB (0xAC) is at address 0x5C
   - SECHDRINDX to 0x01
   - SECPAYINDX to 0x1B
   - SECENDINDX to 0x1E.
3. MRF24XA asserts RXIF (RXSFDIF):
   - Packet accepted by RX filter
   - ACK frame: 05 || 02 10 84 || 05 E2 sent to medium (asserts TXSFD, TXMAIF).
4. —
5. Host MCU downloads SECKEY
   0xC0C1C2C3C4C5C6C7C8C9CACBCCCD-CECF, where LSB (0xCF) is at address 0x40.
6. Host MCU issues RXDEC.
7. MRF24XA performs CCM* decryption, D4 3E 02 2B is decrypted to 61 62 63 64.
8. MRF24XA asserts RXDECIF (and IDLEIF).
9. —
10. —
11. —
12. —
13. —
14. —
15. SW read the entire frame from the Rx Buffer (0x300): 
   20 || 69 DC 84 21 43 02 00 00 00 00 48 DE AC 01 00 00 00 00 48 DE AC || 04 05 00 00 00 || 61 62 63 64 || E0 18 || RSVs
5.8.2 802.15.4-2006 COMPLIANT FRAME ANNEX C.2.3 (TYPE A)

- Network Configuration: Extended address, ACKReq
- Source address: 0xACDE480000000001, where 01 is at address 0x1F
- Source PANID: 0x4321, where 21 is at address 0x29
- Destination address: 0xACDE480000000002
- Destination PANID: 0xFFFF
- Payload: 01 CE
- Frame counter: 0x00000005
- Security level: 0x06
- Packet: Command packet

5.8.2.1 Transmission

1. Host MCU constructs the frame and loads the buffer:
   1E || 2B DC 84 21 43 02 00 00 00 00 48 DE AC
   FF FF 01 00 00 00 00 48 DE AC || 06 05 00 00
   00 || 01 CE

2. —
3. —
4. —
5. —
6. Host MCU configures SECKEY:
   0xC0C1C2C3C4C5C6C7C8C9CACBCCCD-
   CECF, where LSB (0xCF) is at address 0x40.
7. Host MCU clears DTS register.
8. Host MCU issues TXST.
9. MRF24XA configures:
   - SECSUITE to 0x06
   - SECNONCE to 0xAC-
     DE4800000000100000000504, where
     MSB (0xAC) is at address 0x5C
   - SECHDRINDX to 0x01
   - SECPAYINDX to 0x1E
   - SECENDINDX to 0x26.
10. MRF24XA performs CCM* authentication with encryption, where 01 CE is encrypted to 01 D8, and the following MIC tag is attached:
    4F DE 52 90 61 F9 C6 F1
11. MRF24XA appends CRC: 0x4FE4.
12. MRF24XA transmits the packet to the medium. MRF24XA is waiting for an ACK frame. Different IF is received based on the register settings (for example, TX with CSMA).
   TX Buffer (0x200) content:
   28 || 2B DC 84 21 43 02 00 00 00 00 48 DE AC
   FF FF 01 00 00 00 00 48 DE AC || 06 05 00 00
   00 || 01 D8 || 4F DE 52 90 61 F9 C6 F1 || E4 4F

5.8.2.2 Reception

1. MRF24XA receives the following packet through the antenna:
   28 || 2B DC 84 21 43 02 00 00 00 00 48 DE AC
   FF FF 01 00 00 00 00 48 DE AC || 06 05 00 00
   00 || 01 D8 || 4F DE 52 90 61 F9 C6 F1 || E4 4F
2. MRF24XA configures:
   - SECSUITE to 0x06
   - SECNONCE to 0xAC-
     DE4800000000100000000504, where
     MSB (0xAC) is at address 0x5C
   - SECHDRINDX to 0x01
   - SECPAYINDX to 0x1E
   - SECENDINDX to 0x26.
3. MRF24XA asserts RXIF (RXSFDIF):
   - Packet accepted by RX filter
   - ACK frame: 05 || 02 10 84 || 05 E2 sent to
     medium (asserts TXSFD, TXMAIF).
4. —
5. Host MCU downloads SECKEY:
   0xC0C1C2C3C4C5C6C7C8C9CACBCCCD-
   CECF, where LSB (0xCF) is at address 0x40.
6. Host MCU issues RXDEC.
7. MRF24XA performs CCM* de-authentication and decryption, where the MIC tag is compared against the received one, and 01 D8 is decrypted to 01 CE.
8. MRF24XA asserts RXDECIF (and IDLEIF).
9. —
10. —
11. —
12. —
13. —
14. —
15. SW can read the entire frame from Rx Buffer (0x300):
    28 || 2B DC 84 21 43 02 00 00 00 00 48 DE AC
    FF FF 01 00 00 00 00 48 DE AC || 06 05 00 00
    00 || 01 CE || 4F DE 52 90 61 F9 C6 F1 || E4 4F
    || RSVs
5.8.3 NWK LAYER SECURITY (TYPE C)

- Network Configuration: Extended address
- Source address: N/A
- Source PANID: N/A
- Destination address: 0x9897969594939291
- Destination PANID: 0xD2D1
- Network header: 41 41
- Network payload: 14 14
- Network security level: 0x06
- Packet: Data packet

5.8.3.1 Transmission

1. Host MCU constructs the frame and loads the buffer:
   11 || 01 0C 14 D1 D2 91 92 93 94 95 96 97 98 || 41 41 14 14
2. Host MCU configures security materials:
   - SECSUITE register to 0x06
   - SECNONCE register to 0xFDFCFB-FAF9F8F7F6F5F4F3F2F1, where MSB (0xFD) is at address 0x5C
   - SECKEY register to 0x0F0E0D0C0B0A0908070605040302010, where LSB (0x00) is at address 0x40
   - SECHDRINDX register to 0x0E
   - SECPAYINDX register to 0x10
   - SECENDINDX register to 0x11.
3. Host MCU issues TXENC.
4. MRF24XA performs CCM* authentication with encryption, where 14 14 is encrypted to 14 DA, and the following MIC tag is attached:
   53 99 39 A1 55 C5 D3 F6
5. MRF24XA asserts TXENCIF (and IDLEIF).
6. —
7. —
8. Host MCU issues TXST.
9. —
10. —
11. MRF24XA appends CRC: 0x9BC9.
12. MRF24XA transmits the packet to the medium. Different IF is received based on the register settings (for example, TX with CSMA).

   TX Buffer (0x200) content:
   1B || 01 0C 14 D1 D2 91 92 93 94 95 96 97 98 || 41 41 14 DA 53 99 39 A1 55 C5 D3 F6 || C9 9B

5.8.3.2 Reception

1. MRF24XA receives the following packet through the antenna:
   1B || 01 0C 14 D1 D2 91 92 93 94 95 96 97 98 || 41 41 14 DA 53 99 39 A1 55 C5 D3 F6 || C9 9B
2. —
3. MRF24XA asserts RXIF (RXSFDIF): Packet accepted by RX filter.
4. —
5. —
6. —
7. —
8. —
9. —
10. Host MCU configures security materials:
   - SECSUITE to 0x06
   - SECNONCE to 0xFDFCFB-FAF9F8F7F6F5F4F3F2F1, where MSB (0xFD) is at address 0x5C
   - SECKEY to 0x0F0E0D0C0B0A0908070605040302010, where 00 is at address 0x40
   - SECHDRINDX to 0x0E
   - SECPAYINDX to 0x10.
11. Host MCU issues RXDEC.
12. MRF24XA performs CCM* de-authentication and decryption, where the MIC tag is compared against the received one and 14 DA is decrypted to 14 14.
13. MRF24XA asserts RXDECIF (and IDLEIF).
14. —
15. SW can read the entire frame from Rx Buffer (0x300):
   1B || 01 0C 14 D1 D2 91 92 93 94 95 96 97 98 || 41 41 14 DA 53 99 39 A1 55 C5 D3 F6 || C9 9B || RSVS
5.8.4 802.15.4-2006 COMPLIANT FRAME WITH NWK LAYER SECURITY (TYPE D)

- Network Configuration: Extended address
- Source address: 0x0807060504030201, where LSB (0x01) is at address 0x1F
- Source PANID: 0xC2C1, where LSB (0xC1) is at address 0x29
- Destination address: 0x9897969594939291
- Destination PANID: 0xD2D1
- Network header: 41 41
- Network payload: 14 14
- Network security level: 0x06
- Frame counter: 0x55555555
- Security level: 0x07
- Packet: Command packet

5.8.4.1 Transmission
1. Host MCU constructs the frame and loads the buffer:
   20 || 09 DC 14 D1 D2 91 92 93 94 95 96 97 98
   C1 C2 01 02 03 04 05 06 07 08 || 07 55 55 55 55
   || 41 41 14 14
2. Host MCU configures security materials for NWK:
   - SECSUITE register to 0x06
   - SECNONCE register to 0xFDFFFFFAFAF9FAF
     where MSB (0xFD) is at address 0x5C
   - SECKEY register to
     0x0F0E0D0C0B0A09080706050403020100,
     where LSB (0x00) is at address 0x40
   - SECHDRINDX register to 0x1D
   - SECPAYINDX register to 0x1F
   - SECENDINDX register to 0x20.
3. Host MCU issues TXENC.
4. MRF24XA performs CCM* authentication with encryption, where 41 41 14 14 is encrypted to 41 41 14 DA, and the following MIC tag is attached:
   53 99 39 A1 55 C5 D3 F6 MIC-TAG.
5. MRF24XA asserts TXENCIF (and IDLEIF).
6. Host MCU downloads SECKEY 0x0F0E0D0C0B0A09080706050403020100,
   where LSB (0x00) is at address 0x40.
7. Host MCU clears DTSM register.
8. Host MCU issues TXST.
9. MRF24XA configures:
   - SECSUITE to 0x07
   - SECNONCE to 0x0807060504030201555555555
     where MSB (0x08) is at address 0x5C
   - SECHDRINDX to 0x01
   - SECPAYINDX to 0x1D
   - SECENDINDX to 0x38.
10. MRF24XA performs CCM* authentication with encryption, where 41 41 14 DA 53 99 39 A1 55
    C5 D3 F6 is encrypted to C9 87 C6 D8 7F E4 BD
    A2 A4 00 89 9F, and the following MIC tag is attached:
    B4 E6 9C B1 54 7F 9B B3 40 89 77 FB 93 34 E2 D6
    11. MRF24XA appends CRC: 0x1AA8.
12. MRF24XA transmits the packet to the medium. Different IF is received based on the register settings (for example, TX with CSMA).

5.8.4.2 Reception
1. MRF24XA receives the following packet through the antenna:
   3A || 09 DC 14 D1 D2 91 92 93 94 95 96 97 98
   C1 C2 01 02 03 04 05 06 07 08 || 07 55 55 55 55
   || C9 87 C6 D8 7F E4 BD A2 A4 00 89 9F B4 E6
   9C B1 54 7F 9B B3 40 89 77 FB 93 34 E2 D6 ||
   A8 1A
2. MRF24XA configures:
   - SECSUITE to 0x07
   - SECNONCE to 0x0807060504030201555555555
     where MSB (0x08) is at address 0x5C
   - SECHDRINDX to 0x01
   - SECPAYINDX to 0x1D
   - SECENDINDX to 0x38.
3. MRF24XA asserts RXIF (RXSFDIF): Packet accepted by RX filter.
4. —
5. Host MCU configures SECKEY 0xC0C1C2C3C4C5C6C7C8C9CABCCCD
    where LSB (0xCF) is at address 0x40.
6. Host MCU issues RXDEC.
7. MRF24XA performs CCM* de-authentication and decryption, where the MIC tag is compared against the received one, and C9 87 C6 D8 7F
    E4 BD A2 A4 00 89 9F is decrypted to 41 41 14 DA 53 99 39 A1 55
    C5 D3 F6
8. MRF24XA asserts RXDECIF (and IDLEIF).
9. —
10. Host MCU configures security materials:
    - SECSUITE to 0x06
    - SECNONCE to 0xFDFFFFF9F8F7F6F5F4F3F2F1, where MSB (0xFD) is at address 0x5C
    - SECKEY to 0x0F0E0D0C0B0A09080706050403020100, where LSB (0x00) is at address 0x40
    - SECHDRINDX to 0x1D
    - SECPAYINDX to 0x1F.
11. Host MCU issues RXDEC.
12. MRF24XA performs CCM* de-authentication and decryption, where the MIC tag is compared against the received one, and 14 DA is decrypted to 14 14.
13. MRF24XA asserts RXDECIF (and IDLEIF).
14. —
15. SW can read the entire frame from the RxBuffer (0x300):
    3A || 09 DC 14 D1 D2 91 92 93 94 95 96 97 98 C1 C2 01 02 03 04 05 06 07 08 || 07 55 55 55 55 || 41 41 14 14 53 99 39 A1 55 C5 D3 F6 B4 E6 9C B1 54 7F 98 B3 40 89 77 FB 93 34 E2 D6 || A8 1A || RSVs

5.8.5 802.15.4-2003 COMPLIANT FRAME (TYPE A)
• Network Configuration: Extended address, PAN compression
• Source address: 0x0807060504030201, where 01 is at address 0x1F
• Destination address: 0xAAAAAAAAAAAAAAAA
• PANID: 0x3412 where 12 is at address 0x29
• Payload: FF
• Frame counter: 0x0403020100
• Key sequence counter: 0x12
• Security level: CCM-32 (SecLevel: 0x05)
• Packet: Data packet

5.8.5.1 Transmission
1. Host MCU constructs the frame and loads the buffer:
    1B || 49 CC 01 12 34 AA AA AA AA AA AA AA AA AA AA AA AA AA 01 02 03 04 05 06 07 08 || 01 02 03 04 05 || FF
2. —
3. —
4. —
5. —
6. Host MCU configures SECKEY:
    0x000102030405060708090A0B0C0D0E0F, where LSB (0x0F) is at address 0x40.
7. Host MCU clears DTSM.
8. Host MCU issues TXST.
9. MRF24XA configures:
    - SECNONCE to 0x01020304050607080102030405, where MSB (0x01) is at address 0x5C
    - SECHDRINDX to 0x01
    - SECPAYINDX to 0x1B
    - SECENDINDX to 0x1B.
10. MRF24XA performs CCM* authentication with encryption, where FF is encrypted to AC, and the following MIC tag is attached:
    FC 30 DB BD
11. MRF24XA appends CRC: 0xEB32.
12. MRF24XA transmits the packet to the medium.
    Different IF is received based on the register settings (for example, TX with CSMA).
    TX Buffer (0x200) content:
    21 || 49 CC 01 12 34 AA AA AA AA AA AA AA AA AA AA AA AA 01 02 03 04 05 06 07 08 01 02 03 04 05 || AC FC 30 DB BD || 32 EB
6.0 PROPRIETARY FRAME FORMAT AND FRAME PROCESSING

6.1 Proprietary MAC Frame Configuration

Figure 6-1 shows the proprietary MAC header structure. Figure 6-2 shows the specific format of Acknowledge frame. The frame buffer is written first with the LENGTH field byte, and then followed by the FrameCtrl field. An optional Acknowledge Info field is sent before the SEQUENCE.

**FIGURE 6-1: PROPRIETARY MAC HEADER STRUCTURE**

<table>
<thead>
<tr>
<th>FrameCtrl (1 octets)</th>
<th>AckInfo (0-1 octets)</th>
<th>Sequence</th>
<th>DestAddr (0-8 octets)</th>
<th>SrcAddr (0-8 octets)</th>
<th>AuxSecHdr (0-3 octets)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Destination</th>
<th>Source</th>
</tr>
</thead>
</table>

1 byte of FrameCtrl (Proprietary)

<table>
<thead>
<tr>
<th>SAddrPrsnt (1 bit)</th>
<th>DAddrPrsnt (1 bit)</th>
<th>AckReq (1 bit)</th>
<th>Repeat (1 bit)</th>
<th>SecEn (1 bit)</th>
<th>Broadcast (1 bit)</th>
<th>Type (2 bits)</th>
</tr>
</thead>
</table>

6.2 Frame Types

**TABLE 6-1: FRAME TYPES (BOTH PROTOCOLS)**

<table>
<thead>
<tr>
<th>Frame Type</th>
<th>IEEE</th>
<th>Mi-Wi</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>001</td>
<td>01</td>
<td>Filtered by DATAREJ</td>
</tr>
<tr>
<td>Command</td>
<td>011</td>
<td>11</td>
<td>Filtered by CMDREJ First byte of payload (Command) is never encrypted.</td>
</tr>
<tr>
<td>Ack</td>
<td>010</td>
<td>10</td>
<td>Acknowledge Frame Must be generated by the receiver (from SW or HW) if AckReq = 1 in the last received frame, using the same sequence number. Acknowledge Frame is generated by hardware (AUTOACKEN = 1). If this is the case, it is not loaded to the TX frame buffer. AUTOACKEN = 1 requires CRCSZ = 1 on both the transmitter and the receiver side.</td>
</tr>
<tr>
<td>Beacon</td>
<td>000</td>
<td>N/A</td>
<td>Filtered by BCREJ Otherwise, this device does not provide support for parsing on Beacon frames. Security processing requires adjusting the payload index (SECPAYINDX). Beacon frames are only used with broadcast addressing.</td>
</tr>
</tbody>
</table>

(0x0) Sequence DestAddr
(0-8 octets)
(0-8 octets)
(0-3 octets)

(0x0) Address
(0-8 octets)
(0-8 octets)
6.3 Addressing in Proprietary Framing Mode

The following fields are handled using these examples:

- Header
- Sequence number
- Address
- Data/Command
- CRC
- Inferred Destination Addressing: The Destination Address participates in the CRC computation as part of the frame, but it is omitted from the frame that is sent into the air.

TABLE 6-1: FRAME TYPES (BOTH PROTOCOLS) (CONTINUED)

<table>
<thead>
<tr>
<th>Frame Type</th>
<th>IEEE</th>
<th>Mi-Wi</th>
<th>Description</th>
</tr>
</thead>
</table>
| Streaming  | N/A  | 00    | TRXMODE= 01 by transmitter  
TRXMODE= 10 by receiver  
The two buffers are handled by alternating between the two to service a single direction.  
Security parsing makes no distinction between Streaming frames and Data frames.  
Streaming frames are never acknowledged. |

FIGURE 6-2: PROPRIETARY MAC HEADER STRUCTURE: ACKNOWLEDGE FRAME

<table>
<thead>
<tr>
<th>Frame Type</th>
<th>IEEE</th>
<th>Mi-Wi</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AckReq</td>
<td>00</td>
<td>00</td>
<td></td>
</tr>
<tr>
<td>Repeat</td>
<td>00</td>
<td>00</td>
<td></td>
</tr>
<tr>
<td>SecEn</td>
<td>00</td>
<td>00</td>
<td></td>
</tr>
<tr>
<td>Broadcast</td>
<td>00</td>
<td>00</td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>00</td>
<td>00</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FrameCtrl</td>
<td>0x06</td>
</tr>
<tr>
<td>SAddrPrsnt</td>
<td>0</td>
</tr>
<tr>
<td>DAddrPrsnt</td>
<td>0</td>
</tr>
<tr>
<td>AckReq</td>
<td>0</td>
</tr>
<tr>
<td>Repeat</td>
<td>0</td>
</tr>
<tr>
<td>SecEn</td>
<td>0</td>
</tr>
<tr>
<td>Broadcast</td>
<td>1</td>
</tr>
<tr>
<td>Type</td>
<td>10</td>
</tr>
</tbody>
</table>
### TABLE 6-2: RELEVANT REGISTERS FOR PROPRIETARY MODE ADDRESSING

<table>
<thead>
<tr>
<th>ADDR.</th>
<th>REGISTER</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x10</td>
<td>MACCON1</td>
<td>TRXMODE&lt;1:0&gt;</td>
<td>ADDRSZ&lt;2:0&gt;</td>
<td>CRCSZ</td>
<td>FRMFMT</td>
<td>SECFLAGOVR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x1F</td>
<td>ADDR1</td>
<td></td>
<td>ADDR&lt;7:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x20</td>
<td>ADDR2</td>
<td></td>
<td>ADDR&lt;15:8&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x21</td>
<td>ADDR3</td>
<td></td>
<td>ADDR&lt;23:16&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x22</td>
<td>ADDR4</td>
<td></td>
<td>ADDR&lt;31:24&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x23</td>
<td>ADDR5</td>
<td></td>
<td>ADDR&lt;39:32&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x24</td>
<td>ADDR6</td>
<td></td>
<td>ADDR&lt;47:40&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x25</td>
<td>ADDR7</td>
<td></td>
<td>ADDR&lt;55:48&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x26</td>
<td>ADDR8</td>
<td></td>
<td>ADDR&lt;63:56&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend: \( r \) = Reserved, read as ‘0’.

### TABLE 6-3: EXAMPLE CONFIGURATION

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TX and RX Common</td>
<td>ADDRSZ&lt;2:0&gt;= 101 =&gt; (means that ADDR&lt;63:40&gt; is not used)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TX (Source) Configuration</td>
<td>ADDR&lt;63:0&gt;= 0xxxxx060504030201 (4 MSBs not used)</td>
<td>SHADDR&lt;15:0&gt; = xx xx (not used)</td>
<td>PANID &lt;15:0&gt; = xx xx (not used)</td>
</tr>
<tr>
<td>RX (Destination) Configuration</td>
<td>ADDR&lt;63:0&gt;= 0xxxxx969594939291 (4 MSBs not used)</td>
<td>SHADDR&lt;15:0&gt; = xx xx (not used)</td>
<td>PANID &lt;15:0&gt; = xx xx (not used)</td>
</tr>
<tr>
<td>FrameCtrl</td>
<td>Type&lt;1:0&gt;</td>
<td>Broadcast</td>
<td>0</td>
</tr>
<tr>
<td>Frame</td>
<td>Length</td>
<td>FrameCtrl</td>
<td>Sequence++</td>
</tr>
</tbody>
</table>

### TABLE 6-4: LEGAL DESTINATION ADDRESSING OPTIONS USING THE EXAMPLE

<table>
<thead>
<tr>
<th>Options:</th>
<th>Broadcast</th>
<th>Unicast</th>
<th>Inferred</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example DEST.</td>
<td>—</td>
<td>0x969594939291</td>
<td>—</td>
</tr>
<tr>
<td>Type</td>
<td>xx</td>
<td>xx</td>
<td>xx</td>
</tr>
<tr>
<td>Broadcast</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DAPrsnt</td>
<td>x</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>ADDRSZ&lt;2:0&gt;</td>
<td>xxx</td>
<td>3’b101</td>
<td>xxx</td>
</tr>
<tr>
<td>CRCSZ</td>
<td>x</td>
<td>xx</td>
<td>1</td>
</tr>
</tbody>
</table>

### TABLE 6-5: LEGAL SOURCE ADDRESSING OPTIONS USING THE EXAMPLE

<table>
<thead>
<tr>
<th>Options:</th>
<th>Unicast</th>
<th>Inferred</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example SRC.</td>
<td>0x060504030201</td>
<td>—</td>
</tr>
<tr>
<td>Type</td>
<td>xx</td>
<td>xx</td>
</tr>
<tr>
<td>SAPrsnt</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ADDRSZ&lt;2:0&gt;</td>
<td>3’b101</td>
<td>xxx</td>
</tr>
</tbody>
</table>
**REGISTER 6-1: RXFILTER (RX FILTER REGISTER) ADDRESS: 0x18**

<table>
<thead>
<tr>
<th>Bit</th>
<th>Legend</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>R</td>
<td>Readable bit</td>
</tr>
<tr>
<td>6</td>
<td>r</td>
<td>Reserved</td>
</tr>
<tr>
<td>5</td>
<td>CMDREJ</td>
<td>Command Frame Reject Enable bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Setting this bit enables the user to reject all packets with FrameCtrl&gt;Type equal to Command.</td>
</tr>
<tr>
<td>4</td>
<td>DATAREJ</td>
<td>Data Frame Reject Enable bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Setting this bit enables the user to reject all packets with FrameCtrl&gt;Type equal to Data.</td>
</tr>
<tr>
<td>0-3</td>
<td></td>
<td>Out of Scope</td>
</tr>
</tbody>
</table>

- **Legend:**
  - R = Readable bit
  - W = Writable bit
  - U = Unimplemented bit, read as ‘0’
  - -n = Value at POR
  - ‘1’ = Bit is set
  - ‘0’ = Bit is cleared
  - x = Bit is unknown
  - r = Reserved

- **Bit 5:**
  - **CMDREJ:** Command Frame Reject Enable bit
    - Setting this bit enables the user to reject all packets with FrameCtrl<Type> equal to Command.
    - 1 = Reject all Command packets
    - 0 = Disable Command Frame Rejection

- **Bit 4:**
  - **DATAREJ:** Data Frame Reject Enable bit
    - Setting this bit enables the user to reject all packets with FrameCtrl<Type> equal to Data.
    - 1 = Reject all Data packets
    - 0 = Disable Data Frame Rejection
6.3.1 INFERRED DESTINATION ADDRESSING

Inferred destination addressing is indicated in the Proprietary frame format through the combination of DAddrPresent = 0 and Broadcast = 0 flags in the frame header (FrameControl field) and CRCSZ = 1 in the MACCON1 register.

The transmitter calculates the CRC over the complete frame, see Figure 6-3, but drops the Destination Address (DAddr) from the transmitted one, see Figure 6-4. The receiver checks the CRC with its own address inserted. In the case of a match the frame is accepted, otherwise it is silently discarded. This way CRC filtering takes over the role of Address-Match filtering.

As the framing overhead becomes shorter, the duty-cycle of the radio gets decreased or the throughput gets increased. Therefore, the energy consumed by sending a single byte can outweigh the energy budget of hundreds of MCU byte-operations, the impact on battery life is straightforward.

Note that in case of Inferred DA, the ACKINFO field is mandatory when AckReq = 1. Otherwise, the ACKINFO field is only mandatory if ADPTDREN = 1 or ADPTCHEN = 1.

**FIGURE 6-3: INFERRED DESTINATION ADDRESS MODE**

Transmitter constructs the frame\(^{(1)}\)

<table>
<thead>
<tr>
<th>FrameCtrl</th>
<th>SeqNo</th>
<th>DAddr</th>
<th>SAddr</th>
<th>Payload</th>
<th>CRC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>True</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Frame sent into the air without the DAddr field

<table>
<thead>
<tr>
<th>FrameCtrl</th>
<th>SeqNo</th>
<th>SAddr</th>
<th>Payload</th>
<th>CRC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Receiver checks the CRC using its own node address

<table>
<thead>
<tr>
<th>FrameCtrl</th>
<th>SeqNo</th>
<th>NAddr</th>
<th>SAddr</th>
<th>Payload</th>
<th>CRC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note 1:** The indicated frame format is arbitrary to some extent. SeqNo and SAddr refer to sequence number and source address respectively.
### 6.4 Security Material Retrieval
Support with Proprietary Frames

#### FIGURE 6-4: PROPRIETARY MAC AUXSECHDR OCTETS (ONLY PRESENT WHEN SECEN = 1)

<table>
<thead>
<tr>
<th>SecLayer = 00 (MAC layer security only)</th>
<th>7:2</th>
<th>1:0</th>
</tr>
</thead>
<tbody>
<tr>
<td>MacPayIndex (6 bits)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SecLayer (2 bits)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1-byte</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SecLayer = 01 (NWK layer security only)</th>
</tr>
</thead>
<tbody>
<tr>
<td>15:9</td>
</tr>
<tr>
<td>NetPayIndex (7 bits)</td>
</tr>
<tr>
<td>NetHdrIndex (7 bits)</td>
</tr>
<tr>
<td>SecLayer (2 bits)</td>
</tr>
<tr>
<td>2 bytes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SecLayer = 10 (MAC and NWK layer security)</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>3 bytes</td>
</tr>
</tbody>
</table>

**Note:** Always encode Security Indices with DA present (Inferred DA mode).

As the MacPayIndex and NetPayIndex fields can point anywhere in the frame (within the range of pointer representation), it is the arbitrary choice of the application weather the Nonce and the Security Suite is included in the frame or not.
### 6.5 Security Processing of Transmitted Proprietary Frames

Setting TXST triggers automatic security processing and frame sending as an uninterrupted sequence as shown in **Figure 6-5**, which is triggered by TXENC is only required when both NWK layer and MAC layer security are applied, see **Figure 6-6**. In this case, TXENC is set to perform NWK layer security and TXENCIF must be awaited, then MAC security is configured and TXST is set to launch the MAC security processing and sending.

BUF1TXPP, BUF2TXPP, TXENC, and TXST can trigger security functions as shown in **Figure 6-5** and **Figure 6-6**, where BUF1TXPP and BUF2TXPP are both used for debug. The respective interrupts are generated on completion and the device automatically clears the aforementioned triggering bits. **Figure 6-5** illustrates the conditions for security material retrieval.

**FIGURE 6-5: TRANSMIT SECURITY PROCESSING WHEN FRMFMT = 1 (PROPRIETARY–FORMAT)**

- Before launching the transmit processing (Figure 4-9)
  - SW always configures SECKEY, SECSUITE, SECNONCE.
  - SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.

- SW always configures SECKEY, SECSUITE, SECNONCE.

- SW must never configure SECHDRINDX, SECPAYINDEX, SECENDINDEX, except if DTSM = 1.
FIGURE 6-6: TRANSMITTER TXENC PROCESSING WHEN FRMFMT = 1 (PROPRIETARY FORMAT)

- SW configured SECKEY, SECSUITE and SECNONCE.
- TXENC ← 1
- SecLayer = 10
- DTSM
- SECHDRINDX ← NetHdrIndex
- SECPAYINDEX ← NetPayIndex
- CTR, CBC-MAC, CCM*
- TXENC ← 0
- Interrupt Service

- No
- Yes
- 1
- 0

- TXENC is only required for NWK layer security processing in the case when SecLayer = 10 (implying that MAC layer security is applied on setting TXST).

Unintended use: SECHDRINDEX, SECPAYINDEX and SESENDINDEX configured for MAC layer

- Do NWK layer

SW configured SECKEY, SECSUITE and SECNONCE.

NWK+MAC layer

Do NWK layer

No

Yes

1

0

Unintended use: SECHDRINDEX, SECPAYINDEX and SESENDINDEX configured for MAC layer

SW configured SECKEY, SECSUITE and SECNONCE.

NWK+MAC layer

Do NWK layer

No

Yes

1

0

Unintended use: SECHDRINDEX, SECPAYINDEX and SESENDINDEX configured for MAC layer

SW configured SECKEY, SECSUITE and SECNONCE.
6.6 Security Processing of Received Proprietary Frames

Receive security is always performed when RXDEC is set and awaits RXDECIF or RXTAGIF as it is not automatically triggered. If both MAC and NWK layer security are applied, then both must be processed in this order when RXDEC is set for a second time after updating the security material correctly.

Figure 6-7 shows the security functions triggered by RXDEC. The respective interrupts are generated on completion and the device automatically clears the RXDEC.

**FIGURE 6-7: RECEIVER RXDEC PROCESSING WHEN FRMFMT = 1 (PROPRIETARY)**

- The valid frame is available in BUF2 in Mission mode, or in the buffer selected by BUF1RXPP or BUF2RXPP during debug.
- Device and SW have both parsed the frame.
- Refer to Figure 6-4 for security material retrieval support.
- SW has configured SECKEY, SECSUITE, SECNONCE as required for the frame, and can also overwrite the configurations in SECHDRINDEX, SECPAYINDEX.
- Upon successful MAC layer processing, if Network layer is also secured, the indexes are retrieved. SW must update SECKEY, SECNONCE, SECSUITE.

The diagram illustrates the processing steps:

1. RXDEC ← 1
2. CTR, CBC-MAC, CCM*
3. MIC tag match?
   - Yes
   - No
   - SecLayer = 10?
     - Yes
     - SECHDRINDEX ← NetHdrIndex
     - SECPAYINDEX ← NetPayIndex
     - SECENDINDEX ← MIC-tag length
     - RXDEC ← 0
     - RXDECIF ← 1
     - RXTAGIF ← 1
     - Interrupt Service
   - No Match
4. Authentication failed

The flowchart demonstrates the processing sequence for both MAC and Network layer security.
6.7 Security Procedure for Proprietary Frames

Provided that SecEn is set (1) in the MHR, three levels of security processing are possible, based on the SecLayer<1:0> bits carried in the AuxSecHdr<1:0> field of a given frame: 00-MAC only, 01-NWK only, 10-MAC and NWK.

**TABLE 6-6: SECURED FRAME TRANSMISSION (PROPRIETARY MAC FORMAT)**

<table>
<thead>
<tr>
<th>#</th>
<th>Processing Step</th>
<th>MECH only</th>
<th>NWK only</th>
<th>MAC + NWK layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Host MCU constructs the frame and loads the buffer.</td>
<td>SecEn = 1</td>
<td>SecEn = 1</td>
<td>SecEn = 1</td>
</tr>
<tr>
<td>2</td>
<td>For NWK security processing, Host MCU configures:</td>
<td>No NWK layer security</td>
<td>SECKEY SECSUITE NONCE</td>
<td>SECKEY SECSUITE NONCE</td>
</tr>
<tr>
<td>3</td>
<td>Host MCU triggers security processing without sending.</td>
<td>No TXENC.</td>
<td>Host: TXENC ← 1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Device performs the security processing for NWK layer if TXENC is set. SECPAYINDEX, SECHDRINDEX are filled in from NetHdrIndex, NetPayIndex, respectively. SECENDINDEX initially points at the last payload byte. LENGTH and SECENDINDEX are updated if MIC tag is appended. TXSZIF if size runs over 127 bytes.</td>
<td></td>
<td>NWK layer security</td>
<td>SECPAYINDEX SECHDRINDEX LENGTH, SECENDINDEX</td>
</tr>
<tr>
<td>5</td>
<td>Host MCU awaits TXENCIF interrupt, indicating completion. Device clears TXENC.</td>
<td></td>
<td>TXENCIF ← 1</td>
<td>TXENC ← 0</td>
</tr>
<tr>
<td>6</td>
<td>For MAC security processing, Host MCU configures:</td>
<td>SECKEY SECSUITE NONCE</td>
<td>No MAC security.</td>
<td>SECKEY SECSUITE NONCE</td>
</tr>
<tr>
<td>7</td>
<td>Host MCU triggers Security processing and sending.</td>
<td>Host: TXST ← 1</td>
<td>Host: TXST ← 1</td>
<td>Host: TXST ← 1</td>
</tr>
<tr>
<td>8</td>
<td>If SecEn = 1 and DTSM = 0 then the device configures the SEC*INDEX registers using MacPayIndex, MacHdrIndex and the LENGTH field.</td>
<td>SECHDRINDEX SECPAYINDEX SECENDINDEX</td>
<td>—</td>
<td>SECHDRINDEX SECPAYINDEX SECENDINDEX</td>
</tr>
<tr>
<td>9</td>
<td>Device performs the security processing for the indicated layer: LENGTH is adjusted if MAC or NWK layer MIC tag is appended. TXSZIF if size runs over 127 bytes.</td>
<td>MAC layer security LENGTH if MIC added</td>
<td>NWK layer security LENGTH if MIC added</td>
<td>MAC layer security LENGTH if MIC added</td>
</tr>
<tr>
<td>10</td>
<td>LENGTH is adjusted as CRC is appended (if CRCSZ = 1). TXSZIF if size runs over 127 bytes.</td>
<td>LENGTH, CRC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Frame is sent.</td>
<td>TXIF (if no TXSZIF or FRMIF)</td>
<td></td>
<td>TXST ← 0</td>
</tr>
<tr>
<td>#</td>
<td>Processing Step</td>
<td>Steps per each Security Case</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----</td>
<td>---------------------------------------------------------------------------------</td>
<td>----------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2</td>
<td>Device parses the SecEn bit in the FrameCtrl.</td>
<td>MAC only: SecEn = 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>NWK only: SecEn = 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAC + NWK layer: SecEn = 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>For MAC security processing, the device configures the SECHDRINDEX, SECPAYINDEX based on the Auxiliary Security Header, as well as the SECENDINDEX based on the LENGTH field.</td>
<td>MAC only: SEC*INDX from MacHdrIndex, MacPayIndex and the LENGTH</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>NWK only: SEC*INDX from NetHdrIndex, NetPayIndex and the LENGTH</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAC + NWK layer: SEC*INDX from MacHdrIndex, MacPayIndex and the LENGTH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Valid frame received and accepted by RXFILTER.</td>
<td>RXIF = 1, RXBUFFUL = 1, (RXSFDIF = 1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Host MCU reads the frame header to check SecEn, SecLayer, source address, and so on.</td>
<td>Read frame header from buffer.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>For MAC security processing, the Host MCU must load the following:</td>
<td>SECKEY NONCE SECSUITE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>No MAC layer security</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SECKEY NONCE SECSUITE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Host MCU starts MAC security processing by setting RXDEC.</td>
<td>Host: RXDEC ← 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Device performs MAC layer security.</td>
<td>MAC layer security</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAC layer security</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>If SecLayer = 10 then SEC*INDX are filled in preparation for network layer security processing following in the sequel.</td>
<td>—</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>If Authentication fails then RXTAGIF is generated otherwise the security operation is successful and RXDECIF is generated. Device clears RXDEC.</td>
<td>RXDECIF (or RXTAGIF) RXDEC ← 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>SW examines RXTAGIF; If set, SW aborts further processing and frees the buffer by clearing RXBUFFUL.</td>
<td>RXTAGIF ← 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>For NWK security processing, the Host MCU must load the following:</td>
<td>No NWK layer security</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SECKEY SECSUITE NONCE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SECKEY SECSUITE NONCE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Host MCU starts NWK security processing by setting RXDEC.</td>
<td>Host: RXDEC ← 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Device performs NWK layer security processing. (No figure).</td>
<td>NWK layer security</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>NWK layer security</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>If Authentication fails then RXTAGIF is generated. Otherwise, the security operation is successful and RXDECIF is generated. Device clears RXDEC.</td>
<td>RXDECIF (or RXTAGIF) RXDEC ← 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>SW examines RXTAGIF; If set, SW aborts further processing and frees the buffer by clearing RXBUFFUL.</td>
<td>RXTAGIF ← 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>SW reads the payload from the buffer.</td>
<td>—</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>SW clears the RXBUFFUL to free the buffer.</td>
<td>RXBUFFUL ← 0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
6.8 Security Examples

This section provides examples for Proprietary mode framing.

6.8.1 MAC LAYER SECURITY EXAMPLE 1

- Network Configuration: Address size is 8 bytes
- Source address: 0x0807060504030201, where LSB (0x01) is at address 0x1F
- Destination address: 0x9897969594939291
- Payload: BA BA
- MAC security level: 0x04
- MAC security indices: Only encode from the second payload
- Packet: Data packet

6.8.1.1 Transmission

1. Host MCU constructs the frame and loads the buffer:
   15 || C9 55 91 92 93 94 95 96 97 98 01 02 03 04 05 06 07 08 || 54 || BA BA

2. —
3. —
4. —
5. —
6. Host MCU configures:
   - SECSUITE to 0x04
   - SECKEY to 0x0F0E0D0C0B0A090807060504030201, where LSB (0x00) is at address 0x40
   - SECNONCE to 0x08070605040302015555555506, where MSB (0x08) is at address 0x5c.
7. Host MCU sets the TXST register
8. MRF24XA configures:
   - SECHDRINDX to 0x01
   - SECPAYINDX to 0x15
   - SECENDINDX to 0x15.
9. MRF24XA performs CCM* encryption, where BA BA is encrypted to BA F7.
10. MRF24XA appends CRC: 0x9D0A.
11. MRF24XA transmits the packet to the medium.
    Different IF is received based on the register settings (for example, TX with CSMA).
    TX Buffer (0x200) content:
    17 || C9 55 91 92 93 94 95 96 97 98 01 02 03 04 05 06 07 08 || 54 || BA F7 || 0A 9D

6.8.1.2 Reception

1. MRF24XA receives the following packet through the antenna:
   17 || C9 55 91 92 93 94 95 96 97 98 01 02 03 04 05 06 07 08 || 54 || BA F7 || 0A 9D
2. MRF24XA configures:
   - SECHDRINDX to 0x01
   - SECPAYINDX to 0x15
   - SECENDINDX to 0x15.
3. MRF24XA asserts RXIF (RXSFDIF): Packet accepted by RX filter.
4. —
5. Host MCU configures:
   - SECSUITE to 0x04
   - SECKEY to 0x0F0E0D0C0B0A090807060504030201, where LSB (0x00) is at address 0x40
   - SECNONCE to 0x08070605040302015555555506, where MSB (0x08) is at address 0x5c.
6. Host MCU issues RXDEC.
7. MRF24XA performs CCM* decryption, where BA F7 is decrypted to BA BA.
8. MRF24XA asserts RXDECIF (and IDLEIF).
9. —
10. —
11. —
12. —
13. —
14. —
15. —
16. SW read the entire frame from RX Buffer (0x300):
    17 || C9 55 91 92 93 94 95 96 97 98 01 02 03 04 05 06 07 08 || 54 || BA BA || 0A 9D || RSVs
6.8.2  MAC LAYER SECURITY EXAMPLE 2

- Network Configuration: Address size is 8 bytes, Inferred destination addressing
- Source address: 0x0807060504030201, where LSB (0x01) is at address 0x1F
- Destination address: 0x9897969594939291
- Payload: BA BA
- MAC security level: 0x07
- MAC security indices: Only encode from the second payload
- Packet: Data packet

6.8.2.1  Transmission
1. Host MCU constructs the frame and loads the buffer:
   15 || 09 55 91 92 93 94 95 96 97 98 || 34 || BA BA
   Always encode Security Indices with DA present in AUXSECHDR!
2. —
3. —
4. —
5. —
6. Host MCU configures:
   - SECSUITE to 0x07
   - SECKEY to 0x0F0E0D0C0B0A09080706050403020100, where LSB (0x00) is at address 0x40
   - SECNONCE to 0x08070605040302015555555506, where MSB (0x08) is at address 0x5c.
7. Host MCU issues TXST.
8. MRF24XA configures
   - SECHDRINDEX to 0x01
   - SECPAYINDEX to 0x0D (inferred DA is considered)
   - SECENDINDEX to 0x15.
9. MRF24XA performs CCM* authentication with encryption, where BA BA is encrypted to BA F7, and the following MIC tag is attached:
   00 11 6C 8C 59 02 66 AC 5B DC 2D 30 21 1E D0 0C
10. MRF24XA appends CRC: 0xA2D2.
11. MRF24XA transmits the packet to the medium. Different IF is received based on the register settings (for example, TX with CSMA). Packet transmitted to the medium:
   17 || 09 55 || 34 || BA F7 || 00 11 6C 8C 59 02 66 AC 5B DC 2D 30 21 1E D0 0C || D2 A2
   TX Buffer (0x200) content:
   1F || 09 55 91 92 93 94 95 96 97 98 || 34 || BA F7 || 00 11 6C 8C 59 02 66 AC 5B DC 2D 30 21 1E D0 0C || D2 A2

6.8.2.2  Reception
1. MRF24XA receives the following packet through the antenna:
   17 || 09 55 || 34 || BA F7 || 00 11 6C 8C 59 02 66 AC 5B DC 2D 30 21 1E D0 0C || D2 A2
2. MRF24XA configures:
   - SECHDRINDEX to 0x01
   - SECPAYINDEX to 0x0D (inferred DA is considered)
   - SECENDINDEX to 0x15.
3. MRF24XA asserts RXIF (RXSFDIF): Packet accepted by RX filter.
4. —
5. Host MCU configures:
   - SECSUITE to 0x07
   - SECKEY to 0x0F0E0D0C0B0A09080706050403020100, where LSB (0x00) is at address 0x40
   - SECNONCE to 0x08070605040302015555555506, where MSB (0x08) is at address 0x5c.
6. Host MCU issues RXDEC.
7. MRF24XA performs CCM* de-authentication and decryption, where the MIC tag is compared against the received one, and BA F7 is decrypted to BA BA.
8. MRF24XA asserts RXDECIF (and IDLEIF).
9. —
10. —
11. —
12. —
13. —
14. —
15. SW read the entire frame from RX Buffer (0x300):
   17 || 09 55 || 34 || BA BA || 00 11 6C 8C 59 02 66 AC 5B DC 2D 30 21 1E D0 0C || D2 A2 || RSVs
6.8.3 MAC LAYER SECURITY EXAMPLE 3

- Network Configuration: Address size is 1 byte, Inferred destination addressing
- Source address: 0x01 (is at address 0x1F)
- Destination address: 0x91
- Payload: BA BA
- MAC security level: 0x07
- MAC security indices: Only encode from the second payload
- Packet: Data packet

6.8.3.1 Transmission

1. Host MCU constructs the frame and loads the buffer: 06 || 09 55 91 || 18 || BA BA. Always encode security indices with DA present in AUXSECHDR!
2. —
3. —
4. —
5. —
6. Host MCU configures:
   - SECSUITE to 0x07
   - SECKEY to 0x0F0E0D0C0B0A09080706050403020100, where LSB (0x00) is at address 0x40
   - SECNONCE 0x08070605040302155555555506, where MSB (0x08) is at address 0x5c.
7. Host MCU issues TXST.
8. MRF24XA configures:
   - SECHDRINDX to 0x01
   - SECPAYINDX to 0x06 (inferred DA is considered)
   - SECENDINDX to 0x15.
9. MRF24XA performs CCM* authentication with encryption, where BA BA is encrypted to BA F7, and the following MIC tag is attached:
   35 84 FC 4F 1B 92 36 D2 8F D5 D8 B6 68 79 6A 13
10. MRF24XA appends CRC: 0x23A9.
11. MRF24XA transmits the packet to the medium. Different IF is received based on the register settings (for example, TX with CSMA). Packet transmitted to the medium:
    17 || 09 55 || 18 || BA BA || 35 84 FC 4F 1B 92 36 D2 8F D5 D8 B6 68 79 6A 13 || A9 23
    TX Buffer (0x200) content:
    18 || 09 55 91 || 18 || BA BA || 35 84 FC 4F 1B 92 36 D2 8F D5 D8 B6 68 79 6A 13 || A9 23

6.8.3.2 Reception

1. MRF24XA receives the following packet through the antenna:
   17 || 09 55 || 18 || BA F7 || 35 84 FC 4F 1B 92 36 D2 8F D5 D8 B6 68 79 6A 13 || A9 23
2. MRF24XA configures:
   - SECHDRINDX to 0x01
   - SECPAYINDX to 0x06 (inferred DA is considered)
   - SECENDINDX to 0x15.
3. MRF24XA asserts RXIF (RXSFDIF): Packet accepted by RX filter.
4. —
5. Host MCU configures:
   - SECSUITE to 0x07
   - SECKEY to 0x0F0E0D0C0B0A09080706050403020100, where LSB (0x00) is at address 0x40
   - SECNONCE to 0x08070605040302015555555506, where MSB (0x08) is at address 0x5c.
6. Host MCU issues RXDEC.
7. MRF24XA performs CCM* de-authentication and decryption, where the MIC tag is compared against the received one, and BA F7 is decrypted to BA BA.
8. MRF24XA asserts RXDECIF (and IDLEIF).
9. —
10. —
11. —
12. —
13. —
14. —
15. SW read the entire frame from RX Buffer (0x300):
   17 || 09 55 || 18 || BA BA || 35 84 FC 4F 1B 92 36 D2 8F D5 D8 B6 68 79 6A 13 || A9 23 || RSVs
6.8.4 NWK LAYER SECURITY

EXAMPLE 1

- Network Configuration: Address size is 8 bytes
- Source address: 0x0807060504030201, where LSB (0x01) is at address 0x1F
- Destination address: 0x9897969594939291
- Network header: BA BA
- Network payload: AB AB
- NET security level: 0x01
- Packet: Data packet

6.8.4.1 Transmission

1. Host MCU constructs the frame and loads the buffer:
   15 || C9 55 91 92 93 94 95 96 97 98 01 02 03 04 05 06 07 08 || 55 2E || BA BA AB AB

2. Host MCU configures:
   - SECSUITE to 0x01
   - SECKEY to 0x0F0E0D0C0B0A09080706050403020100, where LSB (0x00) is at address 0x40
   - SECNONCE 0x08070605040302015555555506, where MSB (0x08) is at address 0x5c.

3. —
4. —
5. —
6. —
7. Host MCU issues TXST.
8. MRF24XA configures:
   - SECHDRINDX to 0x15
   - SECPAYINDX to 0x17
   - SECENDINDX to 0x18.
9. MRF24XA performs CCM* authentication, where the following MIC tag is attached:
   FB 17 32 26
10. MRF24XA appends CRC: 0xAA70.
11. MRF24XA transmits the packet to the medium. Different IF is received based on the register settings (for example, TX with CSMA).
   TX Buffer (0x200) content:
   1E || C9 55 91 92 93 94 95 96 97 98 01 02 03 04 05 06 07 08 || 55 2E || BA BA AB FB 17 32 26 || 70 AA

6.8.4.2 Reception

1. MRF24XA receives the following packet through the antenna:
   1E || C9 55 91 92 93 94 95 96 97 98 01 02 03 04 05 06 07 08 || 55 2E || BA BA FB 17 32 26 || 70 AA
2. MRF24XA configures:
   - SECHDRINDX to 0x15
   - SECPAYINDX to 0x17
   - SECENDINDX to 0x18.
3. MRF24XA asserts RXIF (RXSFDIF): Packet accepted by RX filter.
4. —
5. —
6. —
7. —
8. —
9. —
10. —
11. Host MCU configures:
    - SECSUITE to 0x04
    - SECKEY to 0x0F0E0D0C0B0A09080706050403020100, where LSB (0x00) is at address 0x40
    - SECNONCE to 0x08070605040302015555555506, where MSB (0x08) is at address 0x5c.
12. Host MCU issues RXDEC.
13. MRF24XA performs CCM* de-authentication, where the MIC tag is compared against the received one.
14. MRF24XA asserts RXDECIF (and IDLEIF).
15. —
16. SW read the entire frame from RX Buffer (0x300):
    1E || C9 55 91 92 93 94 95 96 97 98 01 02 03 04 05 06 07 08 || 55 2E || BA BA FB 17 32 26 || 70 AA || RSVs
6.8.5 NWK LAYER SECURITY

EXAMPLE 2

- Network Configuration: Address size is 8 bytes, Inferred destination addressing
- Source address: 0x0807060504030201 where LSB (0x01) is at address 0x1F
- Destination address: 0x9897969594939291
- Network header: BA BA
- Network payload: AB AB
- Security level: 0x07
- Packet: Data packet

6.8.5.1 Transmission
1. Host MCU constructs the frame and loads the buffer:
   15 || 89 55 91 92 93 94 95 96 97 98 01 02 03 04 05 06 07 08 || 55 2E || BA BA AB AB
2. Host MCU configures:
   - SECSUITE to 0x07
   - SECKEY to 0x0F0E0D0C0B0A09080706050403020100, where LSB (0x00) is at address 0x40
   - SECNONCE 0x08070605040302015555555506, where MSB (0x08) is at address 0x5c.
3. —
4. —
5. —
6. —
7. Host MCU issues TXST.
8. MRF24XA configures:
   - SECHDRINDX to 0x15 (inferred DA is considered)
   - SECPAYINDX to 0x17 (inferred DA is considered)
   - SECENDINDX to 0x20.
9. MRF24XA performs CCM* authentication and encryption, where BA BA AB AB is encrypted to BA BA E6 E5, and the following MIC tag is attached:
   77 FE 46 E2 D4 0E 1D C6 34 D9 34 36 4F 28 2F D8
10. MRF24XA appends CRC: 0x55C1.
11. MRF24XA transmits the packet to the medium. Different IF is received based on the register settings (for example, TX with CSMA). Packet transmitted to the medium:
   22 || 89 55 01 02 03 04 05 06 07 08 || 55 2E || BA BA E6 E5 77 FE 46 E2 D4 0E 1D C6 34 D9 34 36 4F 28 2F D8 || C1 55
   TX Buffer (0x200) content:

2A || 89 55 91 92 93 94 95 96 97 98 01 02 03 04 05 06 07 08 || 55 2E || BA BA E6 E5 77 FE 46 E2 D4 0E 1D C6 34 D9 34 36 4F 28 2F D8 || C1 55

6.8.5.2 Reception
1. MRF24XA receives the following packet through the antenna:
   22 || 89 55 01 02 03 04 05 06 07 08 || 55 2E || BA BA E6 E5 77 FE 46 E2 D4 0E 1D C6 34 D9 34 36 4F 28 2F D8 || C1 55
2. MRF24XA configures:
   - SECHDRINDX to 0x15 (inferred DA is considered)
   - SECPAYINDX to 0x17 (inferred DA is considered)
   - SECENDINDX to 0x20.
3. MRF24XA asserts RXIF (RXSFDIF): Packet accepted by RX filter.
4. —
5. —
6. —
7. —
8. —
9. —
10. —
11. Host MCU configures:
   - SECSUITE to 0x07
   - SECKEY to 0x0F0E0D0C0B0A09080706050403020100, where LSB (0x00) is at address 0x40
   - SECNONCE 0x08070605040302015555555506, where MSB (0x08) is at address 0x5c.
12. Host MCU issues RXDEC.
13. MRF24XA performs CCM* de-authentication and decryption, where the MIC tag is compared against the received one, and BA BA E6 E5 is decrypted to BA BA AB AB.
14. MRF24XA asserts RXDECIF (and IDLEIF).
15. —
16. SW read the entire frame from RX Buffer (0x300):
   22 || 89 55 01 02 03 04 05 06 07 08 || 55 2E || BA BA AB AB 77 FE 46 E2 D4 0E 1D C6 34 D9 34 36 4F 28 2F D8 || C1 55 || RSVs
6.8.6 NWK LAYER EXAMPLE 3

- Network Configuration: Address size is 1 byte, Inferred destination addressing
- Source address: 0x01 (is at address 0x1F)
- Destination address: 0x91
- Network header: BA BA
- Network payload: AB AB
- Security level: 0x07
- Packet: Data packet

6.8.6.1 Transmission

1. Host MCU constructs the frame and loads the buffer: 0A || 89 55 91 01 || 1D 12 || BA BA AB AB
2. Host MCU configures:
   - SECSUITE to 0x07
   - SECKEY to 0x0F0E0D0C0B0A09080706050403020100, where LSB (0x00) is at address 0x40
   - SECNONCE 0x08070605040302015555555506, where MSB (0x08) is at address 0x5c.
3. —
4. —
5. —
6. —
7. Host MCU issues TXST.
8. MRF24XA configures:
   - SECHDRINDX to 0x07 (inferred DA is considered)
   - SECPAYINDX to 0x09 (inferred DA is considered)
   - SECENDINDX to 0x19.
9. MRF24XA performs CCM* authentication and encryption, where BA BA AB AB is encrypted to BA BA E6 E5, and the following MIC tag is attached:
   77 FE 46 E2 D4 0E 1D C6 34 D9 34 36 4F 28 2F D8
10. MRF24XA appends CRC: 0xCD05.
11. MRF24XA transmits the packet to the medium. Different IF is received based on the register settings (for example, TX with CSMA). Packet transmitted to the medium:
   1B || 89 55 01 || 1D 12 || BA BA E6 E5 77 FE 46 E2 D4 0E 1D C6 34 D9 34 36 4F 28 2F D8 || 05 CD
   TX Buffer (0x200) content:
   1C || 89 55 91 01 || 1D 12 || BA BA E6 E5 77 FE 46 E2 D4 0E 1D C6 34 D9 34 36 4F 28 2F D8 || 05 CD

6.8.6.2 Reception

1. MRF24XA receives the following packet through the antenna:
   1B || 89 55 01 || 1D 12 || BA BA E6 E5 77 FE 46 E2 D4 0E 1D C6 34 D9 34 36 4F 28 2F D8 || 05 CD
2. MRF24XA configures:
   - SECHDRINDX to 0x07 (inferred DA is considered)
   - SECPAYINDX to 0x09 (inferred DA is considered)
   - SECENDINDX to 0x19.
3. MRF24XA asserts RXIF (RXSFDIF): Packet accepted by RX filter.
4. —
5. —
6. —
7. —
8. —
9. —
10. —
11. Host MCU configures:
   - SECSUITE to 0x07
   - SECKEY to 0x0F0E0D0C0B0A09080706050403020100, where LSB (0x00) is at address 0x40
   - SECNONCE 0x08070605040302015555555506, where MSB (0x08) is at address 0x5c.
12. Host MCU issues RXDEC.
13. MRF24XA performs CCM* de-authentication and decryption, where the MIC tag is compared against the received one, and BA BA E6 E5 is decrypted to BA BA AB AB.
14. MRF24XA asserts RXDECIF (and IDLEIF).
15. —
16. SW read the entire frame from the RX Buffer (0x300):
   1B || 89 55 01 || 1D 12 || BA BA BA BA 77 FE 46 E2 D4 0E 1D C6 34 D9 34 36 4F 28 2F D8 || 05 CD || RSVs
6.8.7 MAC AND NWK LAYER SECURITY
EXAMPLE 1

1. Host MCU constructs the frame and loads the buffer:
   19 || C9 55 91 92 93 94 95 96 97 98 01 02 03 04 05 06 07 08 || 5E 18 19 || BA BA AB AB

2. Host MCU configures:
   - SECSUITE to 0x07
   - SECKEY to 0x0F0E0D0C0B0A09080706050403020100, where LSB (0x00) is at address 0x40
   - SECNONCE 0x08070605040302015555555506, where MSB (0x08) is at address 0x5c.

3. Host MCU issues TXENC.

4. MRF24XA configures:
   - SECHDRINDX to 0x18
   - SECPAYINDX to 0x19
   - SECENDINDX to 0x19.

5. MRF24XA performs CCM* authentication with encryption, where AB AB is encrypted to AB E6, and the following MIC tag is attached:

   AB 4B 03 7B B7 30 98 B1 E5 93 CA D7 86 81 8A 2D

6. MRF24XA asserts TXENCIF (and IDLEIF).

7. Host MCU configures:
   - SECSUITE to 0x18
   - SECKEY to 0xFFFEFDFCFB-FAF9F8F7F6F5F4F3F2F1F0, where LSB (0xF0) is at address 0x40
   - SECNONCE 0xFFFEFDFCFB-FAF9F8F7F6F5F4F3F2F1F0, where LSB (0xF0) is at address 0x40

8. Host MCU issues TXST.

9. MRF24XA configures:
   - SECHDRINDX to 0x01
   - SECPAYINDX to 0x17
   - SECENDINDX to 0x29.

10. MRF24XA performs CCM* authentication, where the following MIC tag is attached:

    05 15 AB 5F 6C 7D 5C 70 6C 96 91 C0 34 E5 18 0D

11. MRF24XA appends CRC: 0x717B.

12. MRF24XA transmits the packet to the medium. Different IF is received based on the register settings (for example, TX with CSMA).

   TX Buffer (0x200) content:

   3B || C9 55 91 92 93 94 95 96 97 98 01 02 03 04 05 06 07 08 || 5E 18 19 || BA BA AB AB

6.8.7.2 Reception

1. MRF24XA receives the following packet through the antenna:

   3B || C9 55 91 92 93 94 95 96 97 98 01 02 03 04 05 06 07 08 || 5E 18 19 || BA BA AB AB

2. MRF24XA configures:
   - SECHDRINDX to 0x01
   - SECPAYINDX to 0x17
   - SECENDINDX to 0x39.

3. MRF24XA asserts RXIF (RXSFDIF): Packet accepted by RX filter.

4. —

5. Host MCU configures:
   - SECSUITE to 0x18
   - SECKEY to 0xFFFEFDFCFB-FAF9F8F7F6F5F4F3F2F1F0, where LSB (0xF0) is at address 0x40
   - SECNONCE 0xFFFEFDFCFB-FAF9F8F7F6F5F4F3F2F1F0, where LSB (0xF0) is at address 0x40

6. MRF24XA asserts RXDECIF (and IDLEIF).

7. MRF24XA configures:
   - SECHDRINDX to 0x18
   - SECPAYINDX to 0x19
   - SECENDINDX to 0x29.

8. Host MCU issues RXDEC.

9. MRF24XA asserts RXDECIIF (and IDLEIF).

10. —
11. Host MCU configures:
   - SECSUITE to 0x07
   - SECKEY to
     0x0F0E0D0C0B0A09080706050403020100, where LSB (0x00) is at address 0x40
   - SECNONCE to
     0x08070605040302015555555506, where MSB (0x08) is at address 0x5c.
12. Host MCU issues RXDEC.
13. MRF24XA performs CCM* de-authentication and decryption, where the MIC tag is compared against the received one, and AB E6 is decrypted to AB AB.
14. MRF24XA asserts RXDECIF (and IDLEIF).
15. SW read the entire frame from the RX Buffer (0x300):
   3B || C9 55 91 92 93 94 95 96 97 98 01 02 03 04 05 06 07 08 || 5E 18 19 || BA BA AB AB
   03 7B B7 30 98 B1 E5 93 CA D7 86 81 8A 2D
   05 15 AB 5F 6C 7D 5C 70 6C 96 91 C0 34 E5 18 0D || 7B 71 || RSVs

6.8.8 MAC AND NWK LAYER SECURITY
EXAMPLE 2

- Network Configuration: Address size is 4 bytes, Inferred destination addressing
- Source address: 0x04030201 where LSB (0x01) is at address 0x1F
- Destination address: 0x94939291
- Payload: BA BA
- Network header: AB
- Network payload: AB
- Network security level: 0x07
- MAC security level: 0x07
- MAC security indices: Only encode from the second payload
- Packet: Data packet

6.8.8.1 Transmission

1. Host MCU constructs the frame and loads the buffer:
   11 || 89 55 91 92 93 94 01 02 03 04 || 3E 10 11 || BA BA AB AB
2. Host MCU configures:
   - SECSUITE to 0x07
   - SECKEY to
     0x0F0E0D0C0B0A09080706050403020100, where LSB (0x00) is at address 0x40
   - SECNONCE
     0x08070605040302015555555506, where MSB (0x08) is at address 0x5c.
3. Host MCU issues TXENC.
4. MRF24XA configures:
   - SECHDRINDX to 0x10
   - SECPAYINDX to 0x11
   - SECENDINDX to 0x11.
5. MRF24XA performs CCM* authentication with encryption, where AB AB is encrypted to AB E6, and the following MIC tag is attached:
   AB 4B 03 7B B7 30 98 B1 E5 93 CA D7 86 81 8A 2D
6. MRF24XA asserts TXENCIF (and IDLEIF).
7. Host MCU configures:
   - SECSUITE to 0x03
   - SECKEY to 0xFFEFDFCFBF-FAF9F8F7F6F4F3F2F1F0, where LSB (0x00) is at address 0x40
   - SECNONCE
     0xFFEFDFCFBF-FAF9F8F7F6F4F3F2F15555555506, where (0xF0) is at address 0x5c.
8. Host MCU issues TXST.
9. MRF24XA configures:
   - SECHDRINDEX to 0x01
   - SECPAYINDEX to 0x0F
   - SECENDINDEX to 0x21

10. MRF24XA performs CCM* authentication with encryption, where BA BA AB E6 AB 4B 03 7B B7 30 98 B1 E5 93 CA D7 86 81 8A 2D is encrypted to BA A2 6D 3C 78 90 8F 99 BB E6 6B 29 CC AF A1 6F 14 9B OD 7A, and the following MIC tag is attached:
    23 EB C5 73 E8 44 DA 0E 8D D7 9C E7 06 E1 BD C2

11. MRF24XA appends CRC: 0xF9B3.

12. MRF24XA transmits the packet to the medium.
    Different IF is received based on the register settings (for example, TX with CSMA). Packet transmitted to the medium:
    2F || 89 55 01 02 03 04 || 3E 10 11 || BA A2 6D 3C 78 90 8F 99 BB E6 6B 29 CC AF A1 6F 14 9B OD 7A 23 EB C5 73 E8 44 DA 0E 8D D7 9C E7 06 E1 BD C2 || B3 F9
    TX Buffer (0x200) content:
    33 || 89 55 91 92 93 94 01 02 03 04 || 3E 10 11 || BA A2 6D 3C 78 90 8F 99 BB E6 6B 29 CC AF A1 6F 14 9B OD 7A 23 EB C5 73 E8 44 DA 0E 8D D7 9C E7 06 E1 BD C2 || B3 F9

### 6.8.8.2 Reception

1. MRF24XA receives the following packet through the antenna:
   2F || 89 55 01 02 03 04 || 3E 10 11 || BA A2 6D 3C 78 90 8F 99 BB E6 6B 29 CC AF A1 6F 14 9B OD 7A 23 EB C5 73 E8 44 DA 0E 8D D7 9C E7 06 E1 BD C2 || B3 F9

2. MRF24XA configures:
   - SECHDRINDEX to 0x01
   - SECPAYINDEX to 0x0F (Inferred DA is considered)
   - SECENDINDEX to 0x2D.

3. MRF24XA asserts RXIF (RXSFDIF): Packet accepted by RX filter.

4. —

5. Host MCU configures:
   - SECSUITE to 0x07
   - SECKEY to 0x0706050403020100
   - SECNONCE to 0x0F0E0D0C0B0A09080706050403020100, where LSB (0x00) is at address 0x40

6. SW read the entire frame from RX Buffer (0x300):
   2F || 89 55 01 02 03 04 || 3E 10 11 || BA A2 6D AB AB 4B 03 7B B7 30 98 B1 E5 93 CA D7 86 81 8A 2D 23 EB C5 73 E8 44 DA 0E 8D D7 9C E7 06 E1 BD C2 || B3 F9 || RSVs

7. MRF24XA performs CCM* de-authentication, where the MIC tag is compared against the received one, and BA A2 6D 3C 78 90 8F 99 BB E6 6B 29 CC AF A1 6F 14 9B OD 7A is decrypted to BA BA AB E6 AB 4B 03 7B B7 30 98 B1 E5 93 CA D7 86 81 8A 2D

8. MRF24XA configures:
   - SECHDRINDEX to 0x10
   - SECPAYINDEX to 0x11
   - SECENDINDEX to 0x1D.

9. MRF24XA asserts RXDECIF (and IDLEIF).

10. —

11. Host MCU configures:
   - SECSUITE to 0x07
   - SECKEY to 0x0706050403020100
   - SECNONCE to 0x0F0E0D0C0B0A09080706050403020100, where MSB (0x08) is at address 0x5c.

12. Host MCU issues RXDEC.

13. MRF24XA performs CCM* de-authentication and decryption, where the MIC tag is compared against the received one, and AB E6 is decrypted to AB AB.

14. MRF24XA asserts RXDECIF (and IDLEIF).

15. —

16. SW read the entire frame from RX Buffer (0x300):
   2F || 89 55 01 02 03 04 || 3E 10 11 || BA A2 6D AB AB AB AB 4B 03 7B B7 30 98 B1 E5 93 CA D7 86 81 8A 2D 23 EB C5 73 E8 44 DA 0E 8D D7 9C E7 06 E1 BD C2 || B3 F9 || RSVs
7.0 ADVANCED LINK BEHAVIOR IN PROPRIETARY PACKET MODE

7.1 Channel Agility

In some communication environments, where several nodes use the same medium, it is necessary to choose different channels for the communicating nodes. It introduces difficulties in ACK receiving. Figure 7-1 illustrates the example of channel agility. To prevent higher MCU load on channel changing, MRF24XA handles the channel change for the automatic sending of ACK. This feature is enabled by ADPTCHEN bit that must be enabled for all the nodes within the same network.

Note that in case of Inferred DA, the ACKINFO field is mandatory when AckReq=1. Otherwise the ACKINFO field is mandatory if ADPTDREN = 1 or ADPTCHEN = 1.

FIGURE 7-1: CHANNEL AGILITY EXAMPLE

Example 7-1: Auto ACK Example with Channel Agility

A → B:

A: CHANNEL = 10(1)
   TXST
   CHANNEL = 7

B: CHANNEL = 10(2)
   RXIF
   CHANNEL = 7

Sending ACK back to A
   CHANNEL = 10

Note 1: Node A knows the frequency channel that Node B uses for receiving.

2: However, Node B knows the RX channel of Node A, as AUTOACK = 1 the ACK must immediately sent back, and there is no time for MCU interactions.

Channel Agility is based on AckInfo field of the Proprietary MAC header. Proprietary MAC header is described in Figure 6-1.

FIGURE 7-2: PROPRIETARY MAC ACKINFO<7:0> OCTET (ONLY PRESENT WHEN ACKREQ = 1)

<table>
<thead>
<tr>
<th>3:0</th>
<th>7:4</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACKDataRate (4 bits)</td>
<td>ACKChannel (4 bits)</td>
</tr>
</tbody>
</table>

1 Octet

AckDataRate: When ADPTDREN = 1 and AckReq = 1, this field determines the TX Data Rate for the ACK packet, regardless of the setting of the DR<3:0> Register field.

AckChannel: When ADPTCHEN = 1 and AckReq = 1, this field determines the TX Channel for the ACK packet, regardless of the setting of the CH<3:0> register field.
## REGISTER 7-1: RXCON2 (MAC RECEIVE CONTROL 2 REGISTER)  ADDRESS: 0x16

<table>
<thead>
<tr>
<th>R/C/HS-0</th>
<th>R/W-0</th>
<th>R-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>RXBUFFUL</td>
<td>IDENTREJ</td>
<td>ACKRXFP</td>
<td>ACKTXFP</td>
<td>AUTORPTEN</td>
<td>AUTOACKEN</td>
<td>ADPTCHEN</td>
<td>ADPTDREN</td>
</tr>
</tbody>
</table>

### Legend:
- **R** = Readable bit  
- **W** = Writable bit  
- **U** = Unimplemented bit, read as ‘0’
- **-n** = Value at POR  
- **1** = Bit is set  
- **0** = Bit is cleared  
- **x** = Bit is unknown
- **r** = Reserved  
- **HS** = Hardware Set  
- **C** = Clearable bit

#### bit 7-2
Out of scope

#### bit 1
**ADPTCHEN**: Adaptive Channel Enable bit\(^{(1)}\)

- Setting this bit enables the MAC to set the transmitting channel for the ACK packet based on the Ack-Info field (Proprietary packet) of the received packet, rather than the CH<3:0> register bits.
  
  - 1 = Adaptive Channel feature is enabled
  - 0 = Adaptive Channel feature is disabled

#### bit 0
Out of scope

### Note 1:
Use ADPTCHEN field while receiving and transmitting, and must be unmodified while RXEN or TXST is set.
7.2 Data Rate Agility

Similar to Channel Agility as described in Figure 7.1, ACK is sent at different data rates within different nodes. It may provide more robust ACK sending mechanism in busier networking environments. Data Rate Agility is enabled by the ADPTDREN bit that must be set for all the nodes within the same network. Data Rate Agility is based on AckInfo field of the Proprietary MAC header. Proprietary MAC header is described in Figure 6-1. AckInfo field is described in Figure 7-1. Figure 4-13 describes the ACK sending mechanism.

REGISTER 7-2: RXCON2 (MAC RECEIVE CONTROL 2 REGISTER) ADDRESS: 0x16

<table>
<thead>
<tr>
<th>R/C/HS-0</th>
<th>R/W-0</th>
<th>R-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>RXBUFFUL</td>
<td>IDENTREJ</td>
<td>ACKRXFP</td>
<td>ACKTXFP</td>
<td>AUTORPTEN</td>
<td>AUTOACKEN</td>
<td>ADPTCHEN</td>
<td>ADPTDREN</td>
<td></td>
</tr>
</tbody>
</table>

Legend:  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as ‘0’  
-n = Value at POR  
‘1’ = Bit is set  
‘0’ = Bit is cleared  
x = Bit is unknown  
r = Reserved  
HS = Hardware Set  
C = Clearable bit

bit 7-1 Out of scope  
bit 0 ADPTDREN: Adaptive Data Rate Enable bit(1)  
Setting this bit enables the MAC to set the transmission data rate for the ACK packet based on the AckInfo field (Proprietary packet) of the received packet, rather than the DR<2:0> register bits.  
1 = Adaptive Data Rate feature is enabled  
0 = Adaptive Data Rate feature is disabled

Note 1: Use ADPTDREN field while receiving and transmitting, and must be unmodified while RXEN or TXST is set.
7.3 Auto-Repeater

The Auto-Repeat feature enables to automatically (without Host Controller intervention) repeat any packets that request it (FrameCtrl[Repeat] = 1). Auto-Repeat is not available in RX- or TX- Streaming modes as these modes are designed to provide maximum throughput rather than reliable transport. Only Data and Command frames are repeated with Auto-Repeat.

If MRF24XA receives a Data or Command frame with its FrameCtrl[Repeat] bit set, the frame may be repeated without MCU intervention when AUTORPTEN register bit is set. If the packet is a Unicast packet, and its Destination Address (explicit or inferred) matches the ADDR<63:0> register, then the packet is not repeated, but received as a normal packet instead. If the packet is a Broadcast packet, or is a Unicast packet with a non-matching address, the packet is received into the buffer, and then retransmitted (repeated) without modification (no CRC generation or encryption steps are performed).

It is recommended to use this function together with the CSMA/CA algorithm to avoid the collision with the IDENTITY = 1 function. Therefore, packet received more than twice is not repeated.
7.4 Streaming

The Streaming feature provides the maximum throughput between two nodes. In this mode, the two packet buffers are used to transmit/receive packets by alternating between the two.

The main advantage of this mode is that MCU can perform buffer RW while the packet is being transmitted over the air.

Auto-ACK and Auto-Retransmission are unavailable in this mode. Additionally, CSMA-CA operation is skipped to provide the maximum throughput.

7.4.1 TX

After the initial negotiation (channel, power, security key, and so on), the TX node sets the TRXMODE<1:0> register to 2'b10. The MCU shall write the packets to address 0x200. The switching is handled internally. Note that MCU must write MRF24XA if the TXBUFFEMPTY flag is set. To maximize throughput, "WRITE and set TXST" SPI framing format is recommended.

Note that TXRDBUF, BUSWRBUF, and TXBUFFEMPTY signals are the output on the GPIO pins with GPIOMODE = 1010 settings.

Handling buffer empty flag by an additional interrupt is recommended to avoid constant polling over SPI.

FIGURE 7-3: STREAMING MODE TX TIMELINE

![Stream TIMELINE](image-url)
7.4.2 RX

After the initial negotiation (channel, power, security key, and so on), the RX node sets the TRXMODE<1:0> register to 2'b01. The MCU must read the packets from address 0x300. The switching is handled internally. Note that MCU must read MRF24XA when the RXBUFFFULL flag is set. In Streaming mode receiving packets after RXIF do not work. To maximize throughput, "READ and clear RXBUFFFULL" SPI framing format is recommended.

Note that RXWRBUF, BUSRDBUF, and RXBUFFUL signals are the output on the GPIO pins with GPIOMODE = 1011 settings.

Handling buffer full flag by an additional interrupt is recommended to avoid constant polling over SPI.

To reach optimal sensitivity, turn Power Save (PSAV) off.

FIGURE 7-4: STREAMING RX TIMELINE

![Streaming RX Timeline Diagram]
7.4.3 STREAM RX TIMEOUT

The radio sets STRMIF to indicate that the time defined in STRMTO register has elapsed.

**REGISTER 7-3: PIR3 (PERIPHERAL INTERRUPT REGISTER 3)**

<table>
<thead>
<tr>
<th>RW/HS/HC-0</th>
<th>R/W/HS/HC-0</th>
<th>R/W/HS/HC-0</th>
<th>R-0</th>
<th>R/W/HS/HC-0</th>
<th>R/W/HS/HC-0</th>
<th>R/W/HS/HC-0</th>
<th>R/W/HS/HC-0</th>
<th>STRMIF</th>
</tr>
</thead>
<tbody>
<tr>
<td>RXIF</td>
<td>RXDECIF</td>
<td>RXTAGIF</td>
<td>r</td>
<td>RXIDENTIF</td>
<td>RXFLTIF</td>
<td>RXOVFIF</td>
<td>STRMIF</td>
<td>bit 7</td>
</tr>
<tr>
<td>bit 7-0</td>
<td>bit 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend:  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as ‘0’  

-n = Value at POR  
‘1’ = Bit is set  
‘0’ = Bit is cleared  
x = Bit is unknown

bit 7-1 Out of Scope

bit 0 STRMIF: Receive Stream Time-Out Error Interrupt Flag bit

Set by the device to indicate that the duration specified in STRMTO elapsed since the last received packet while in RX-Streaming mode, and the MAC clears the stored sequence number.

Nonpersistent, cleared by SPI read.

**Note 1:** In Packet mode, use a single buffer for received frames. In RX-Streaming mode, use both buffers for reception.

**REGISTER 7-4: TRMTOH/STRMTOL (STREAM TIME-OUT REGISTER)**

<table>
<thead>
<tr>
<th>RW-11111111</th>
<th>STRMTO&lt;15:8&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 15-8</td>
<td>bit 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RW-11111111</th>
<th>STRMTO&lt;7:0&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 7-0</td>
<td></td>
</tr>
</tbody>
</table>

Legend:  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as ‘0’  

-n = Value at POR  
‘1’ = Bit is set  
‘0’ = Bit is cleared  
x = Bit is unknown

bit 15-0 STRMTO<15:0>: Stream Time-Out bits

The STRMTO<15:0> bits indicate the maximum number of allowed Base time units between the end of one RX Stream packet and the successful reception of the next. If no RX Stream packet is successfully received within this time, STRMIF is set.
8.0 BRIDGING

Any member of a Proprietary node may run two stacks and be part of a 15.4 network. This enables the node to act as a gateway between the two networks. As the default configuration, NWFRMFMT = 1. This setting is overridden when it receives a 15.4 frame, but switches back temporarily when it needs to send.

The network is configurable to use proprietary or standard (IEEE 802.15.4) MAC. However, a proprietary network must also be able to send and receive frames to/from standard-compliant networks. This capability is referred to as Bridging.

The Bridging node must implement both the proprietary and the standard compliant MAC framing protocols. Each time a 250 kbps frame is received through the Bridging node, it must decide which MAC protocol to use for parsing the incoming frame.

The problem of Bridging is that the 802.15.4 MAC Frame Control field will not allow for distinction in the case of 250 kbps frames. Therefore, the selected SFD is used for distinction. Standard compliant SFD pattern is selected when Standard MAC is used with 250 kbps frame, otherwise, a different SFD value is used.

When the Host MCU (or MAC) selects the 250 kbps air-data-rate for transmission, it also indicates which MAC is used by a sideband signal. The transmitter baseband selects the SFD accordingly. If proprietary MAC is selected, the SFD is read from a host configurable register, otherwise, the standard defined pattern is used (0xA7). Additionally, if proprietary is used, the length and payload fields are scrambled.

On the receiver side, the 250 kbps preamble and the SFD are detected first. This determines which MAC protocol is used to parse the PHY payload.

FIGURE 8-1: BRIDGING

[Diagram of Bridging network with legend: Proprietary NWK node, Standard NWK node, Bridging node belonging to both networks, Standard 250 kbps link, Proprietary PHY link]
9.0 PHYSICAL LAYER FUNCTIONS

9.1 Synthesizer Power-Up, Power Off

Table 2-1 illustrates MRF24XA power modes, while Figure 4-3 illustrates the operation states.

RFOFF state is the state when most of the RF circuits are powered off. As Table 2-1 illustrates, RFOFF state is divided into two sub-states. In Crystal ON state, only 16 MHz on-chip crystal oscillator is powered on and the synthesizer is switched off. In Synthesizer ON state, both the on-chip crystal oscillator and the synthesizer are powered on.

MRF24XA provides an OFF-Timer to optimize the power consumption by managing the ON time of the on-chip synthesizer. Before the synthesizer is switched off, RXEN or TXST turns to ‘0’ from ‘1’, the user must set OFFTM<7:0>. The value of the register is interpreted as an OFF time counter. As the counter runs out, synthesizer is started regardless of the state of RXEN and TXST bits.

REGISTER 9-1: OFFTM (OFF-TIMER REGISTER)  ADDRESS: 0x1E

<table>
<thead>
<tr>
<th>bit 7</th>
<th>bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFFTM&lt;7:0&gt;</td>
<td></td>
</tr>
</tbody>
</table>

R/W-00000000

Legend:  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as ‘0’  
-n = Value at POR  
‘1’ = Bit is set  
‘0’ = Bit is cleared  
x = Bit is unknown  
r = Reserved

bit 7-0 OFFTM<7:0>: OFF-Timer Field bits
This value sets the minimum PLL OFF time.
Minimum OFF Time = OFFTM<7:0> * 32 μs
If this register is set to 0xFF, PLL remains off.
REGISTER 9-2: RXCON1 (MAC RECEIVE CONTROL 1 REGISTER)

ADDRESS: 0x15

<table>
<thead>
<tr>
<th>RW/HC/HS-0</th>
<th>R/W-0</th>
<th>R/W/HC-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>RXEN</td>
<td>NOPA</td>
<td>RXDEC</td>
<td>RSVLQIEN</td>
<td>RSVRSSIEN</td>
<td>RSVCHDREN</td>
<td>RSVCFQEN</td>
<td>r</td>
<td></td>
</tr>
</tbody>
</table>

bit 7  
RXEN: Receive Enable Field bit
This bit enables/disables the packet reception. If an RX packet is being received, clearing this bit will causes that packet to be discarded.

1 = RX enabled
0 = RX disabled

Hardware clear/set when:
• Cleared when TRXMODE is set to TX-Streaming mode
• Set when TRXMODE is set to RX-Streaming mode

Clearing this bit aborts the current operation in the following cases:
• Receiving a packet in Packet mode or in RX-Streaming mode

Changes to most RX related settings must be only done when this bit is cleared.

The clear channel assessment (CSMAEN) and ACK-frame reception does not require RXEN = 1 as the device turns the radio into RX when required, irrespective of the status of the RXEN bit.

bit 6  
NOPA: No Parsing bit
This bit disables packet parsing. Only CRC is checked, if it is enabled. This feature is useful in Sniffer mode.

1 = Disable packet parsing
0 = Enable packet parsing

bit 5  
RXDEC: RX Decryption bit
Setting this bit starts RX security processing (authentication or decryption, or both) on the last received packet.

1 = RX security processing started/in process. RXDECIF or RXTAGIF is set.
0 = RX security processing inactive or complete

This bit clears itself after RX decryption has completed.

bit 4  
RSVLQIEN: Receive Status Vector LQI Enable bit
If this bit is set, the measured Link Quality is appended after the received frame in the packet buffer.

1 = Append LQI field
0 = Do not append LQI field

bit 3  
RSVRSSIEN: Receive Status Vector RSSI Enable bit
If this bit is set, the measured RSSI is appended after the received frame in the packet buffer.

1 = Append RSSI field
0 = Do not append RSSI field

bit 2  
RSVPCHDREN: Receive Status Vector Channel/MAC Type/Data Rate Enable bit
If this bit is set, Channel, MAC type and Data Rate configurations used with the received frame are appended after the received frame in the packet buffer, using the encoding specified for CH<3:0>, FRMFMT and DR<2:0> (concatenated in this order when most significant bit (MSb) is first).

1 = Append Channel, MAC type and Data Rate fields
0 = Do not append Channel, MAC type and Data Rate fields
REGISTER 9-3: TXCON (TRANSMIT CONTROL REGISTER)
ADDRESS: 0x12

<table>
<thead>
<tr>
<th>R/W/HC-0</th>
<th>RW-0</th>
<th>R/W/HC-0</th>
<th>R/HS/HC-1</th>
<th>R/W-1</th>
<th>RW-011</th>
</tr>
</thead>
<tbody>
<tr>
<td>TXST</td>
<td>DTSM</td>
<td>TXENC</td>
<td>TXBUFEMPTY</td>
<td>CSMAEN</td>
<td>DR&lt;2:0&gt;</td>
</tr>
</tbody>
</table>

Legend:  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as ‘0’  
-1 = Value at POR  
‘1’ = Bit is set  
‘0’ = Bit is cleared  
x = Bit is unknown  
r = Reserved  
HC = Hardware Clear  
HS = Hardware Set

bit 7  
TXST: Transmit Start bit. This is set or cleared by the host MCU bit  
1 = Starts the transmission of the next TX packet  
0 = Termination of current TX operation, which may result in the transmission of an incomplete packet.

Hardware Clear:  
• After the packet is successfully transmitted (including all attempted retransmissions, if any), the hardware clears this bit and sets the TXIF and IDLEIF.  
• If the packet transmission fails due to a CSMA failure, then this bit is cleared, and TXCSMAIF is set.  
• If Acknowledge is requested (AckReq bit field in the transmitted frame and AUTOACKEN register bit are both set) and not received after the configured number of retransmissions (TXRETMCNT), then TXST bit is cleared and a TXACKIF is set.  
• In TX-Streaming mode (TRXMODE), TXST is set even when it is already set, resulting in a “posted start”. When the current TX operation completes, the “posted start” immediately starts afterward. Clearing of the TXST bit clears both the current and the posted (pending) TX starts. TXOVFIF is unchanged when TXST = 1, a posted start is present and a Host Controller write to the packet buffer occurs. Outside of TX-Streaming mode, writes to TXST when TXST is already set is ignored.

Clearing this bit aborts the current operation in the following cases:  
• When transmitting a packet in Packet mode or in TX-Streaming mode  
• When waiting for an ACK packet after a transmission  
• During the CSMA CA algorithm  
• When transmitting a repeated frame

This field is read at any time to determine if the TX operation is in progress.

bit 6-0  
Out of scope

Note 1: Transmission may include automatic security processing, CRC appending, CSMA-CA channel access, Acknowledge reception and retransmissions depending on the register Configuration and the Frame Control field of the frame to be transmitted.

2: By setting the TXST bit in either Sleep/RFOFF state, the device transits to TX state for packet transmission.
9.2 Operating Channel

MRF24XA is capable of selecting one of sixteen channel frequencies in the 2.4 GHz band.

The desired channel is selected by configuring the CHANNEL<3:0> bits in the MACCON2 register. Refer to Table 9-2 for the MACCON2 register setting for channel number and frequency.

If Channel Agility is not used, all nodes share the same channel both in RX and TX modes. The channel is defined by CHANNEL<3:0> as indicated below.

For more information on Channel Agility, see Section 7.1 “Channel Agility”.

### REGISTER 9-4: MACCON2 (MAC CONTROL 2 REGISTER) ADDRESS: 0x11

<table>
<thead>
<tr>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>R/W-0000</td>
<td>R/W/HS-0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHANNEL&lt;3:0&gt;</td>
<td>SECSUITE&lt;3:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Legend:**
- **R** = Readable bit
- **W** = Writable bit
- **U** = Unimplemented bit, read as ‘0’
- **-n** = Value at POR
- **‘1’** = Bit is set
- **‘0’** = Bit is cleared
- **x** = Bit is unknown
- **r** = Reserved

**bit 7-4** CHANNEL<3:0>: TX/RX Operating Channel field bits

These register bits are used as the current operating channel for TX/RX operation.

- 0x0 = Channel 11
- 0x1 = Channel 12
- ...
- 0xF = Channel 26

**bit 3-0** Out of scope

**Note 1:** Use this field while receiving and transmitting, and must be unmodified while RXEN or TXST is set.

### TABLE 9-2: CHANNEL SELECTION MACCON2 (0x11) REGISTER SETTING

<table>
<thead>
<tr>
<th>CHANNEL&lt;3:0&gt; Bits in MACCON2</th>
<th>Channel Number</th>
<th>Channel Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0</td>
<td>11</td>
<td>2.405 GHz</td>
</tr>
<tr>
<td>0x1</td>
<td>12</td>
<td>2.410 GHz</td>
</tr>
<tr>
<td>0x2</td>
<td>13</td>
<td>2.415 GHz</td>
</tr>
<tr>
<td>0x3</td>
<td>14</td>
<td>2.420 GHz</td>
</tr>
<tr>
<td>0x4</td>
<td>15</td>
<td>2.425 GHz</td>
</tr>
<tr>
<td>0x5</td>
<td>16</td>
<td>2.430 GHz</td>
</tr>
<tr>
<td>0x6</td>
<td>17</td>
<td>2.435 GHz</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CHANNEL&lt;3:0&gt; Bits in MACCON2</th>
<th>Channel Number</th>
<th>Channel Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x7</td>
<td>18</td>
<td>2.440 GHz</td>
</tr>
<tr>
<td>0x8</td>
<td>19</td>
<td>2.445 GHz</td>
</tr>
<tr>
<td>0x9</td>
<td>20</td>
<td>2.450 GHz</td>
</tr>
<tr>
<td>0xA</td>
<td>21</td>
<td>2.455 GHz</td>
</tr>
<tr>
<td>0xB</td>
<td>22</td>
<td>2.460 GHz</td>
</tr>
<tr>
<td>0xC</td>
<td>23</td>
<td>2.465 GHz</td>
</tr>
<tr>
<td>0xD</td>
<td>24</td>
<td>2.470 GHz</td>
</tr>
<tr>
<td>0xE</td>
<td>25</td>
<td>2.475 GHz</td>
</tr>
<tr>
<td>0xF</td>
<td>26</td>
<td>2.480 GHz</td>
</tr>
</tbody>
</table>
REGISTER 9-5: RXCON1 (MAC RECEIVE CONTROL 1 REGISTER) ADDRESS: 0x15

<table>
<thead>
<tr>
<th>RW/HC/HS-0</th>
<th>RW-0</th>
<th>RW/HC-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>R-0</th>
<th>r</th>
</tr>
</thead>
<tbody>
<tr>
<td>RXEN</td>
<td>NOPA</td>
<td>RXDEC</td>
<td>RSVIQEN</td>
<td>RSVRORDER</td>
<td>RSVHDR</td>
<td>RSVCFOEN</td>
<td>r</td>
<td></td>
</tr>
</tbody>
</table>

Legend:  R = Readable bit  W = Writable bit  U = Unimplemented bit, read as '0'
        -n = Value at POR  ‘1’ = Bit is set  ‘0’ = Bit is cleared  x = Bit is unknown
        r = Reserved  HC = Hardware Clear  HS = Hardware Set

bit 7  RXEN: Receive Enable bit(1, 2)

This bit Enables/Disables the packet reception. If an RX packet is being received, clearing this bit causes that packet to be discarded.

1 = RX enabled
0 = RX disabled

Hardware clear/set when:

• Cleared when TRXMODE is set to TX-Streaming mode
• Set when TRXMODE is set to RX-Streaming mode

Clearing this bit aborts the current operation in the following cases:

• Receiving a packet in Packet mode or in RX-Streaming mode
• Transmitting an ACK packet for a received frame during an Auto-Acknowledge operation

bit 6-0  Out of scope

Note 1: Changes to most RX related settings must be only done when this bit is cleared.

2: Clear channel assessment (CSMAEN) and ACK-frame reception does not require RXEN = 1 as the device turns the radio into RX when needed, irrespective of the status of the RXEN bit.
9.3 RXLISTEN Operations

The air-data-rate is detected in two stages:

1. Simultaneously monitoring the presence of 2 Mbps, 500 kbps, 250 kbps modulated preambles until one of these is detected with sufficient reliability.
2. Searching for a Start Frame Delimiter that may further distinguish between air-data-rates.

Figure 9-1 and Figure 9-2 show the order of processing steps and decisions for the Optimal and Legacy frame formats.
FIGURE 9-1: DATA RATE SELECTION IN THE RECEIVER WHEN OPTIMAL FRAMING MODE IS CONFIGURED

Start RX Signal Processing

- 2 Mbps preamble?
  - Yes: Signal path to 2 Mbps bit- and byte-aligned demodulation/decoding
    - Search for two different 16-chip SFD patterns (at 2 Mbps)
      - Yes: 16-bit exact match to pattern_2000
        - No: 16-bit tolerant match to pattern_1000
          - Yes: Continue 2 Mbps reception
          - No: Switch to 1 Mbps reception
      - No: 16-bit exact match to pattern_2000
        - Yes: Continue 2 Mbps reception
        - No: Switch to 1 Mbps reception
  - No: 500 kbps preamble?
    - Yes: Signal path to 500 kbps bit- and byte-aligned demodulation/decoding
      - Search for 8-bit SFD pattern
        - Yes: 8-bit exact match to pattern_500
          - Yes: Continue 500 kbps reception
          - No: Switch to 250 kbps reception as in the standard
        - No: 8-bit exact match to pattern_500
          - Yes: Continue 500 kbps reception as in the standard
          - No: Switch to 250 kbps reception
      - No: 500 kbps preamble?
        - Yes: Signal path to 500 kbps bit- and byte-aligned demodulation/decoding
          - Search for two different 16-chip SFD patterns (at 2 Mbps)
            - Yes: 16-bit tolerant match to pattern_1000
              - Yes: Continue 2 Mbps reception
              - No: Switch to 1 Mbps reception
            - No: 16-bit tolerant match to pattern_1000
              - Yes: Continue 2 Mbps reception
              - No: Switch to 1 Mbps reception
        - No: 250 kbps preamble?
          - Yes: Signal path to 250 kbps bit- and byte-aligned demodulation/decoding
            - Search for two different 16-chip SFD patterns (at 250 kbps)
              - Yes: 8-bit exact match to pattern_250
                - Yes: Continue 250 kbps reception
                - No: Switch to 125 kbps reception
              - No: 8-bit exact match to pattern_250
                - Yes: Continue 250 kbps reception
                - No: Switch to 125 kbps reception
          - No: 250 kbps preamble?
            - Yes: Signal path to 250 kbps bit- and byte-aligned demodulation/decoding
              - Search for 8-bit SFD pattern
                - Yes: 8-bit exact match to 0xA7
                  - Yes: IEEE 802.15.4 Frame
                    - Yes: 8-bit exact match to pattern_125
                      - Yes: Switch to 125 kbps reception
                      - No: Switch to 125 kbps reception
                    - No: 8-bit exact match to pattern_125
                      - Yes: Switch to 125 kbps reception
                      - No: Switch to 125 kbps reception
                  - No: Switch to 125 kbps reception
              - No: 8-bit exact match to 0xA7
                - Yes: IEEE 802.15.4 Frame
                  - Yes: 8-bit exact match to pattern_125
                    - Yes: Switch to 125 kbps reception
                    - No: Switch to 125 kbps reception
                  - No: 8-bit exact match to pattern_125
                    - Yes: Switch to 125 kbps reception
                    - No: Switch to 125 kbps reception
        - No: 250 kbps preamble?
          - Yes: Signal path to 250 kbps bit- and byte-aligned demodulation/decoding
            - Search for 8-bit SFD pattern
              - Yes: 8-bit exact match to 0xA7
                - Yes: IEEE 802.15.4 Frame
                  - Yes: 8-bit exact match to pattern_125
                    - Yes: Switch to 125 kbps reception
                    - No: Switch to 125 kbps reception
                  - No: 8-bit exact match to pattern_125
                    - Yes: Switch to 125 kbps reception
                    - No: Switch to 125 kbps reception
                - No: Switch to 125 kbps reception
              - No: 8-bit exact match to 0xA7
                - Yes: IEEE 802.15.4 Frame
                  - Yes: 8-bit exact match to pattern_125
                    - Yes: Switch to 125 kbps reception
                    - No: Switch to 125 kbps reception
                  - No: 8-bit exact match to pattern_125
                    - Yes: Switch to 125 kbps reception
                    - No: Switch to 125 kbps reception
          - No: 250 kbps preamble?
FIGURE 9-2: DATA RATE SELECTION IN THE RECEIVER WHEN LEGACY FRAMING MODE IS CONFIGURED

1. **Start RX Signal Processing**
   - **250 kbps preamble?**
     - **Yes**
       - Switch signal path to 250 kbps bit- and byte-aligned demodulation and decoding
       - Search for two different 16-bit SFD patterns (at 2 Mbps)
     - **No**

2. **8-bit exact match to pattern_L2000**
   - **Yes**
     - Start descrambler
     - After LENGTH field switch to 2 Mbps reception
   - **No**
     - After LENGTH field switch to 1 Mbps reception

3. **8-bit exact match to pattern_L1000**
   - **Yes**
     - Start descrambler
     - After LENGTH field switch to 500 kbps reception
   - **No**
     - After LENGTH field switch to 250 kbps reception

4. **8-bit exact match to pattern_L500**
   - **Yes**
     - Continue 250 kbps reception
     - Switch to 125 kbps reception: LLR mapping, DE-interleaving, FEC decoding
     - Start descrambler
     - Receive LENGTH field
     - Receive Payload bytes
   - **No**
     - Continue 250 kbps reception
     - After LENGTH field switch to 1 Mbps reception

5. **16-bit tolerant match to pattern_125**
   - **Yes**
     - Start descrambler
     - Receive Payload bytes
   - **No**

6. **8-bit exact match to 0xA7**
   - **Yes**
     - IEEE 802.15.4 Frame
     - 8-bit exact match to pattern_250
   - **No**
     - 16-bit tolerant match to pattern_125

7. **8-bit exact match to pattern_250**
   - **Yes**
     - 16-bit tolerant match to pattern_125
   - **No**
     - Continue 250 kbps reception

8. **IEEE 802.15.4 Frame**
   - **Yes**
     - 8-bit exact match to pattern_250
   - **No**
     - 16-bit tolerant match to pattern_125
Abort/Reset from upper layer

Gain Change

Start AGC

Start Digital Demodulator

Preamble Acquired?

AGC Level Triggered?

Yes

No

Switch Signal Flow to bit- and byte-aligned demodulation at the detected Preamble Air-Data Rate; Start SFD detectors; AGC must hold the gain

Yes

SFD detected?

No

SFD time-out?

Yes

No

Init SFD time-out

Preamble Loss?

No

Yes

Switch the signal path to the detected payload data rate

Transfer bytes to MAC

Last byte done?

Yes

No

No

Yes

Yes

No
9.4 Automatic Gain Control (AGC)

AGC circuit can provide automatic gain adjustment according to the Received field strength. AGC gain is set in four steps.

REGISTER 9-7: BBCON (BASEBAND CONFIGURATION REGISTER) ADDRESS 0x38

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Out of scope</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Out of scope</td>
<td></td>
</tr>
<tr>
<td>5-4</td>
<td>RXGAIN&lt;1:0&gt;: Receiver Gain Register Field bits</td>
<td></td>
</tr>
<tr>
<td></td>
<td>By setting this bit, the AGC operation is inhibited in the receiver and the receiver radio gain Configuration is selected between three different gain levels. Encoding:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>11 = AGC operation is enabled (default value)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10 = High gain</td>
<td></td>
</tr>
<tr>
<td></td>
<td>01 = Middle gain</td>
<td></td>
</tr>
<tr>
<td></td>
<td>00 = Low gain</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This feature is used for test and streaming purposes. To reduce the required interframe-gap, the RXGAIN must be set to one of the fixed gain options when the MAC is in Streaming mode.</td>
<td></td>
</tr>
<tr>
<td>3-0</td>
<td>Out of scope</td>
<td></td>
</tr>
</tbody>
</table>
9.5 Energy Detection

The Received Signal Strength Indicator (RSSI) is an estimate of the received signal power within the bandwidth of a particular channel, and obtained by the user using Energy Detection (ED). MRF24XA has the capability to measure the received signal power for a user-defined number of symbols, and to report back the measured RSSI value.

The mapping between the RSSI value returned and the input power level is shown in Figure 9-4. This aggregates all the AGC curves, hence user does not require to calculate with any other settings. The curve is directly used.

The RSSI value associated with a received packet may also be automatically stored as part of the packet’s Receive Status Vector (RSV).

**FIGURE 9-4: RSSI VALUE VS. RECEIVED POWER**

![RSSI Characteristics Graph](image)

**EQUATION 9-1: RSSI VALUE VS. RECEIVED POWER EQUATION**

\[
Pin = 0.5 \times \text{Energy Detect Code}^{(1)} - 112 \text{ dBm}^{(2)}
\]

Note 1: Energy Detect Code is read from EDMEAN<7:0> field.
2: Equation 9-1 is valid for EDMEAN<7:0> from 40 to 184 decimal values.
**REGISTER 9-8: RXCON1 (MAC RECEIVE CONTROL 1 REGISTER)**

<table>
<thead>
<tr>
<th>Bit</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 7</td>
<td>RXEN</td>
<td>Receive Enable Field bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This bit Enables/Disables the packet reception. If an RX packet is currently</td>
</tr>
<tr>
<td></td>
<td></td>
<td>being received, clearing this bit causes that packet to be discarded.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = RX enabled</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = RX disabled</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hardware clear/set when:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Cleared when TRXMODE is set to TX-Streaming mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Set when TRXMODE is set to RX-Streaming mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Clearing this bit aborts the current operation in the following cases:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Receiving a packet in Packet mode or in RX-Streaming mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Changes to most RX related settings must be only done when this bit is</td>
</tr>
<tr>
<td></td>
<td></td>
<td>cleared.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The clear channel assessment (CSMAEN) and ACK-frame reception does not</td>
</tr>
<tr>
<td></td>
<td></td>
<td>require RXEN = 1 as the device turns the radio into RX when needed,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>irrespective of the status of the RXEN bit.</td>
</tr>
<tr>
<td>bit 6</td>
<td>NOPA</td>
<td>No Parsing bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This bit disables packet parsing. Only CRC is checked, if it is enabled.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This feature is useful in Sniffer mode.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = Disable packet parsing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = Enable packet parsing</td>
</tr>
<tr>
<td>bit 5</td>
<td>RXDEC</td>
<td>RX Decryption bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Setting this bit starts RX security processing (authentication or</td>
</tr>
<tr>
<td></td>
<td></td>
<td>decryption, or both) on the last received packet.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = RX security processing started/in process. RXDECIF or RXTAGIF is set.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = RX security processing inactive or complete</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This bit clears itself after RX decryption is completed.</td>
</tr>
<tr>
<td>bit 4</td>
<td>RSVLQIEN</td>
<td>Receive Status Vector LQI Enable bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If this bit is set, the measured Link Quality is appended after the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>received frame in the packet buffer.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = Append LQI field</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = Do not append LQI field</td>
</tr>
<tr>
<td>bit 3</td>
<td>RSVRSSIEN</td>
<td>Receive Status Vector RSSI Enable bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If this bit is set, the measured RSSI is appended after the received</td>
</tr>
<tr>
<td></td>
<td></td>
<td>frame in the packet buffer.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = Append RSSI field</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = Do not append RSSI field</td>
</tr>
</tbody>
</table>

**Legend:**

- **R** = Readable bit
- **W** = Writable bit
- **U** = Unimplemented bit, read as ‘0’
- **-n** = Value at POR
- **‘1’** = Bit is set
- **‘0’** = Bit is cleared
- **x** = Bit is unknown

- **r** = Reserved
- **HC** = Hardware Clear
- **HS** = Hardware Set
REGISTER 9-9:  CCACON1 (CCA CONTROL 1 REGISTER)  ADDRESS: 0x2F

<table>
<thead>
<tr>
<th>R/HS/HC-0</th>
<th>R/W/HC-0</th>
<th>R/W-001100</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCABUSY</td>
<td>CCAST</td>
<td>RSSITHR&lt;5:0&gt;</td>
</tr>
</tbody>
</table>

bit 7  **CCABUSY:** Clear Channel Assessment Busy Flag bit
This bit represents the result of the latest CCA measurement.

- 1 = Medium is busy
- 0 = Medium is silent

bit 6  **CCAST:** Clear Channel Assessment Start bit(1)
Setting this register bit triggers MCU to start a new CCA measurement. The hardware clears this bit when the CCA measurement is done (EDCCAIF is set) and CCABUSY is valid.

bit 5-0  **RSSITHR<5:0>:** RSSI Threshold bits(2)
This threshold is used in CCA operation when Energy detect or Energy and Carrier Sense mode is selected.

Representation: resolution of 2 dB/LSB, RSSITHR = 0x10 represents ca. -75 dBm noise level. Note that this threshold may be different with other matching network or antenna.

**Note 1:** RX chain must be turned on (RXEN = 1) to perform this measurement. Packet reception is not disabled during the measurement, its main purpose is testing.

**Note 2:** In the corresponding CCA modes the radio measures EDMEAN<7:0>. If EDMEAN<7:2> is greater than RSSITHR<5:0>, CCABUSY is set. Example: To set the RSSI threshold where the chip measures 0x30 EDMEAN, RSSITHR<5:0> must be 0x0C.
**REGISTER 9-10: EDCON (ENERGY DETECT CONTROL REGISTER)**

<table>
<thead>
<tr>
<th>R-00</th>
<th>RW</th>
<th>RW/HC-0</th>
<th>R/W-1110</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>EDMODE</td>
<td>EDST</td>
<td>EDLEN&lt;3:0&gt;</td>
</tr>
</tbody>
</table>

Legend:  
- **R** = Readable bit  
- **W** = Writable bit  
- **U** = Unimplemented bit, read as ‘0’  
- ‘-n’ = Value at POR  
- ‘1’ = Bit is set  
- ‘0’ = Bit is cleared  
- **x** = Bit is unknown  
- **r** = Reserved  
- **HC** = Hardware Clear

| bit 7-6 | Reserved: Maintain as ‘0’ |
| bit 5 | **EDMODE**: Energy Detect Mode Select bit |
|       | 1 = Energy Detect Sampling Mode. ED duration is 128 µs. A single atomic RSSI-peak measurement is accomplished. The result is stored in EDPEAK<7:0> register. |
|       | 0 = Energy Detect Scan Mode. EDLEN<3:0> sets the ED duration. The result is stored in EDMEAN<7:0> register. |
| bit 4 | **EDST**: Energy Detect Measurement Start bit |
|       | Setting this register bit triggers MCU to start a new CCA measurement. The hardware clears this register bit when the ED measurement is done (EDCCAIF is unchanged) and values in EDMEAN<7:0> and EDPEAK<7:0> are valid. |
|       | If the ED measurement is aborted (RX state is changed, or the MCU clears the EDST bit) then EDCCAIF is unchanged. |
| bit 3-0 | **EDLEN<3:0>**: Energy Detect Measurement Length Field bits |
|       | Value \( M \) indicates a sequence of \((M + 1) \times 8\) atomic RSSI-peak measurements, each having the duration of 128 µs. At the end of the aggregate measurement, the mean and the peak value of the sequence are available in EDMEAN<7:0> and EDPEAK<7:0>. |

**Note 1:** The RX chain must be turned on (RXEN = 1) to perform this measurement. Packet reception is disabled during the measurement.

**Note 2:** When EDLEN<3:0> = \( M = 0xE \), then the 128 µs atomic measurements are performed 120 times, which is equal to the a BaseSuperFrameDuration parameter in the IEEE 802.15.4 standard.

---

**REGISTER 9-11: EDMEAN (ENERGY DETECT MEAN INDICATION REGISTER)**

<table>
<thead>
<tr>
<th>R/W/HS/HC-00000000</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDMEAN&lt;7:0&gt;</td>
</tr>
</tbody>
</table>

Legend:  
- **R** = Readable bit  
- **W** = Writable bit  
- **U** = Unimplemented bit, read as ‘0’  
- ‘-n’ = Value at POR  
- ‘1’ = Bit is set  
- ‘0’ = Bit is cleared  
- **x** = Bit is unknown  
- **r** = Reserved  
- **HC** = Hardware Clear  
- **HS** = Hardware Set

| bit 7-0 | **EDMEAN<7:0>**: Energy Detect Mean Indication Field bits |
|        | Measured mean signal strength during ED/CCA measurement. |
REGISTER 9-12:  EDPEAK (ENERGY DETECT PEAK INDICATION REGISTER)  ADDRESS: 0x33

<table>
<thead>
<tr>
<th>R/HS/HC-00000000</th>
<th>EDPEAK&lt;7:0&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 7</td>
<td>bit 0</td>
</tr>
</tbody>
</table>

Legend:  R = Readable bit  W = Writable bit  U = Unimplemented bit, read as ‘0’
-n = Value at POR  ‘1’ = Bit is set  ‘0’ = Bit is cleared  x = Bit is unknown
r = Reserved  HC = Hardware Clear  HS = Hardware Set

bit 7-0  EDPEAK<7:0>: Energy Detect Peak Indication Field bits
Measured peak signal strength during ED measurement.
Computation: The gain-compensated RSSI value is averaged over intervals of 128 μs. The peak value obtained from a sequence of such measurements is stored in EDPEAK, when EDMODE = 1.

TABLE 9-4:  REGISTERS ASSOCIATED WITH RSSI AND ED

<table>
<thead>
<tr>
<th>Names</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>RXCON1</td>
<td>RXEN</td>
<td>NOPA</td>
<td>RXDEC</td>
<td>RXVLQIEN</td>
<td>RSVRSSIEN</td>
<td>RSVCHDREN</td>
<td>RSVCF0EN</td>
<td>r</td>
</tr>
<tr>
<td>CCACON1</td>
<td>CCABUSY</td>
<td>CCAST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EDCON</td>
<td></td>
<td></td>
<td>EDMODE</td>
<td>EDST</td>
<td>EDLEN&lt;3:0&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EDMEN</td>
<td></td>
<td></td>
<td>EDMEN&lt;7:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EDPEAK</td>
<td></td>
<td></td>
<td>EDPEAK&lt;7:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend:  r = Reserved, read as ‘0’.
9.6 Clear Channel Assessment (CCA)

Clear Channel Assessment is a function within CSMA/CA to determine whether the wireless medium is ready and able to receive data, therefore the transmitter can start sending the data.

CCA is implemented outside of the MAC. This enables the radio to transmit in the presence of interference from other wireless protocols that operate on the same frequency.

CCA may be performed using either Energy Detection (ED), Carrier Sense (CS) or a combination of both.

9.6.1 CCA CONFIGURATION

CCA is automatically executed (potentially multiple times) as part of the CSMA-CA procedure when the TXST register bit is set and CSMA-CA is enabled.

The following register bits are used in the Configuration of CCA:
- CCAMODE<1:0>
- CCALEN<1:0>
- CSTHR<3:0>
- RSSITHR<5:0>

9.6.1.1 Energy Detection (ED)

When CCAMODE<1:0> = 10, the CCA reports a busy medium upon detecting energy above the energy detection threshold defined in the RSSITHR<5:0> register bits.

To use this method of CCA, the following Configuration must be used:
- CCALEN<1:0> = Measurement duration
- CCAMODE<1:0> = 10
- RSSITHR<5:0> = RSSI threshold value

The mapping between the RSSITHR threshold and the power level is shown in Figure 9-4 and Equation 9-1.

9.6.1.2 Carrier Sense (CS)

When CCAMODE<1:0> = 01, the CCA reports a busy medium upon detecting of a signal with particular modulation and spreading characteristics.

To use this method of CCA, the following Configuration must be used:
- CCALEN<1:0> = Measurement duration
- CCAMODE<1:0> = 01
- CSTHR<3:0> = Carrier sense threshold

9.6.1.3 Carrier Sense with Energy Detection

When CCAMODE<1:0> = 11, the CCA reports a busy medium upon detecting of a signal with particular modulation and spreading characteristics and energy above the energy detection threshold defined in the RSSITHR<5:0> register bits. To use this method of CCA, the following Configuration must be used:
- CCALEN<1:0> = Measurement duration
- CCAMODE<1:0> = 11
- RSSITHR<5:0> = RSSI threshold value
- CSTHR<3:0> = Carrier sense threshold

Note: In the corresponding CCA modes the radio measures EDMEAN<7:0>. If EDMEAN<7:2> is greater than RSSI-THR<5:0>, CCABUSY is set. Example: To set the RSSI threshold where the chip measures 0x30 EDMEAN, RSSI-THR<5:0> must be 0x0C.

9.6.2 CCA OPERATION

MRF24XA automatically initiates CCA as part of the CSMA-CA algorithm. CCA operation is requested independently for software CSMA-CA, or for test purpose through the CCAST bit.
REGISTER 9-13: OPSTATUS (OPERATION STATUS)\(^{(3)}\)  
ADDRESS: 0x2

<table>
<thead>
<tr>
<th>R-0</th>
<th>R/HS/HC-0</th>
<th>R/HS/HC-0</th>
<th>R/HS/HC-0</th>
<th>R/HS/HC-0</th>
<th>R/HS/HC-0</th>
<th>R/HS/HC-0</th>
<th>R/HS/HC-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>MACOP&lt;3:0&gt;</td>
<td></td>
<td>RFOP&lt;2:0&gt;</td>
<td>bit 7</td>
<td>bit 0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend:  
- R = Readable bit  
- W = Writable bit  
- U = Unimplemented bit, read as ‘0’  
- -n = Value at POR  
- ’1’ = Bit is set  
- ’0’ = Bit is cleared  
- x = Bit is unknown  
- r = Reserved  
- HC = Hardware Clear  
- HS = Hardware Set

bit 7  
Reserved: Maintain as ‘0’

bit 6-3  
MACOP <3:0>: MAC Operation Register Field bits\(^{(1, 2)}\)  
Provides status information on the current state of the MAC state machine. Encoding on MACOP<3:1>:

- 111 = Transmitting Acknowledge (TXACK)  
- 110 = Receiving a packet (RXBUSY)  
- 101 = Receiver listening to the channel waiting for packet (RX)  
- 100 = Receiving (or waiting for) Acknowledge (RXACK)  
- 011 = Transmitting a packet (TX)  
- 010 = Performing Clear Channel Assessment (CCA)  
- 001 = Back-off before repeated CCA (BO)  
- 000 = MAC does not perform any operation (IDLE)

bit 2-0  
RFOP <2:0>: Radio Operation Register Field bits  
Provides status information on the current Radio state. Encoding on RFOP<2:0>:

- 111 = TX with external PA is turned on (TX+PA)  
- 110 = RX with external LNA is turned on (RX+LNA)  
- 101 = Synthesizer and external PA or LNA is turned on (SYNTH+PA/LNA)  
- 100 = Radio calibrates if the host MCU sets the CALST, otherwise, device malfunction occurs (CAL/MAL)  
- 011 = Analog transmit chain is activated (TX)  
- 010 = Analog receiver chain is active (RX). (Digital may be partially shut off)  
- 001 = Synthesizer is steady or ramping up or channel change is issued (SYNTH)  
- 000 = Only the crystal oscillator is ON(OFF), (except when XTALSF = 1)

Note 1:  
GPIO<2:0> is dedicated to output MACOP<3:1> or RFOP<2:0>. Refer to PINCON register, which specifies the pin Configuration.

2:  
MACOP<0> is connected to RXBUFFUL register bit. It cannot be output over GPIO's.

3:  
OPSTATUS register is sent on the SDO pin during all SPI operation.
REGISTER 9-14:  CCACON1 (CCA CONTROL 1 REGISTER)  

<table>
<thead>
<tr>
<th>Bit</th>
<th>R/W/HC-0</th>
<th>R/W/HC-0</th>
<th>R/RW01</th>
<th>CCABUSY</th>
<th>CCAST</th>
<th>RSSITHR&lt;5:0&gt;</th>
</tr>
</thead>
</table>

Legend:  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as ‘0’  
-n = Value at POR  
‘1’ = Bit is set  
‘0’ = Bit is cleared  
x = Bit is unknown  
r = Reserved  
HC = Hardware Clear  
HS = Hardware Set

bit 7  
**CCABUSY**: Clear Channel Assessment Busy Flag bit  
This bit represents the result of the latest CCA measurement.  
1 = Medium is busy  
0 = Medium is silent

bit 6  
**CCAST**: Clear Channel Assessment Start bit  
Setting this register bit triggers MCU to start a new CCA measurement. The hardware clears this bit when the CCA measurement is done (EDCCAIF is set) and CCABUSY is valid.

bit 5-0  
**RSSITHR<5:0>**: RSSI Threshold bits  
This threshold is used in CCA operation when Energy detect or Energy and Carrier Sense mode is selected.  
Representation: resolution of 2 dB/LSB, RSSITHR = 0x10 represents ca. -75 dBm noise level. Note that this threshold may be different with other matching network or antenna.

Note 1:  
RX chain must be turned on (RXEN = 1) to perform this measurement. Packet reception is not disabled during the measurement, and main purpose is testing.

Note 2:  
In the corresponding CCA modes the radio measures EDMEAN<7:0>. If EDMEAN<7:2> is greater than RSSITHR<5:0>, CCABUSY is set. Example: To set the RSSI threshold where the chip measures 0x30 EDMEAN, RSSITHR<5:0> must be 0x0C.

REGISTER 9-15:  CCACON2 (CCA CONTROL 2 REGISTER)  

<table>
<thead>
<tr>
<th>Bit</th>
<th>R/W-01</th>
<th>R/RW01</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCTHRH&lt;3:0&gt;</td>
<td>CCASTL&lt;1:0&gt;</td>
<td>CCAMODEL&lt;1:0&gt;</td>
</tr>
</tbody>
</table>

Legend:  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as ‘0’  
-n = Value at POR  
‘1’ = Bit is set  
‘0’ = Bit is cleared  
x = Bit is unknown  
r = Reserved

bit 7  
**CCSTHR<3:0>**: Carrier Sense Threshold Field bits

bit 3-2  
**CCALEN<1:0>**: Clear Channel Assessment Length bits

Value N indicates duration of $2^N \times 32 \ \mu s$.

bit 1-0  
**CCAMODEL<1:0>**: Clear Channel Assessment Mode Field bits

11 = CCA Mode 3/a in the std. <1>: Energy AND Carrier Sense Threshold  
10 = CCA Mode 2 in the std. <1>: Carrier Sense Threshold  
01 = CCA Mode 1 in the std. <1>: Energy Detect Threshold (default)  
00 = CCA Mode 3/b in the std.<1>: Energy OR Carrier Sense Threshold

Note 1:  
The IEEE 802.15.4 standard requires 128 \mu s. But shorter length is recommended when using higher rates with optimized Preamble mode (RATECON.OPTIMAL = 1).

Note 2:  
The measured RSSI result is stored in EDMEAN<7:0> register in all modes except in Mode 2.
REGISTER 9-16: EDMEAN (ENERGY DETECT MEAN INDICATION REGISTER) ADDRESS: 0x32

<table>
<thead>
<tr>
<th>Bit 7</th>
<th>EDMEAN&lt;7:0&gt;</th>
</tr>
</thead>
</table>

Legend:  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as ‘0’  
-n = Value at POR  
‘1’ = Bit is set  
‘0’ = Bit is cleared  
x = Bit is unknown  
r = Reserved  
HC = Hardware Clear  
HS = Hardware Set

bit 7-0 EDMEAN<7:0>: Energy Detect Mean Indication Field bits  
Measured mean signal strength during ED/CCA measurement.

TABLE 9-5: REGISTERS ASSOCIATED WITH CCA

<table>
<thead>
<tr>
<th>Names</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPSTATUS</td>
<td>r</td>
<td>MACOP&lt;3:0&gt;</td>
<td>RFOP&lt;2:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCACON1</td>
<td>CCABUSY</td>
<td>CCAST</td>
<td>RSSITHR&lt;5:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCACON2</td>
<td>CSTHR&lt;3:0&gt;</td>
<td>CCALEN&lt;1:0&gt;</td>
<td>CCAMODE&lt;1:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EDMEAN</td>
<td></td>
<td>EDMEAN&lt;7:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend:  
r = Reserved, read as ‘0’.
9.7 Physical Framing

Physical frame durations for the different data rates are shown in Table 9-6. Duration is expressed in payload byte time.

**TABLE 9-6: FRAME DURATION**

<table>
<thead>
<tr>
<th>Frame Formats</th>
<th>T [µs/byte]</th>
<th>Duration expressed in payload byte time (T)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Preamble</td>
<td>SFD</td>
</tr>
<tr>
<td>Proprietary 125 kbps</td>
<td>64</td>
<td>4</td>
</tr>
<tr>
<td>Standard 250 kbps</td>
<td>32</td>
<td>4</td>
</tr>
<tr>
<td>Proprietary 500 kbps</td>
<td>16</td>
<td>4</td>
</tr>
<tr>
<td>Proprietary 1 Mbps</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>Proprietary 2 Mbps</td>
<td>4</td>
<td>8</td>
</tr>
</tbody>
</table>

Different frame data rates are recognized and processed based on the recognized SFD field of the PHY frame. Figure 4-7 describes the basic PHY frame structure. RATECON<7:2> bits can disable the reception of the unwanted data rate frames.

**TABLE 9-7: USED SFD FIELDS FOR VARIOUS DATA RATES**

<table>
<thead>
<tr>
<th>Preamble Type</th>
<th>Data Rate Pattern</th>
<th>Used SFD Field</th>
<th>Fault Tolerance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimal</td>
<td>Pattern_2000</td>
<td>&lt;SFD1, SFD6&gt;</td>
<td>Exact match required</td>
</tr>
<tr>
<td></td>
<td>Pattern_1000</td>
<td>&lt;SFD2, SFD7&gt;</td>
<td>Maximally two non-contiguous two-element burst error</td>
</tr>
<tr>
<td></td>
<td>Pattern_500</td>
<td>SFD3</td>
<td>Exact match required</td>
</tr>
<tr>
<td></td>
<td>Pattern_250 proprietary</td>
<td>SFD4</td>
<td>Exact match required</td>
</tr>
<tr>
<td>Legacy</td>
<td>Pattern_2000</td>
<td>SFD1</td>
<td>Exact match required</td>
</tr>
<tr>
<td></td>
<td>Pattern_1000</td>
<td>SFD2</td>
<td>Exact match required</td>
</tr>
<tr>
<td></td>
<td>Pattern_500</td>
<td>SFD3</td>
<td>Exact match required</td>
</tr>
<tr>
<td></td>
<td>Pattern_250 proprietary</td>
<td>SFD4</td>
<td>Exact match required</td>
</tr>
<tr>
<td></td>
<td>Pattern_250 standard</td>
<td>0xA7</td>
<td>Exact match required</td>
</tr>
<tr>
<td></td>
<td>Pattern_125</td>
<td>&lt;SFD5, SFD6&gt;</td>
<td>Maximally two faulty nibbles from four</td>
</tr>
<tr>
<td></td>
<td>Legacy</td>
<td>Pattern_2000</td>
<td>SFD1</td>
</tr>
<tr>
<td></td>
<td>Pattern_1000</td>
<td>SFD2</td>
<td>Exact match required</td>
</tr>
<tr>
<td></td>
<td>Pattern_500</td>
<td>SFD3</td>
<td>Exact match required</td>
</tr>
<tr>
<td></td>
<td>Pattern_250 proprietary</td>
<td>SFD4</td>
<td>Exact match required</td>
</tr>
<tr>
<td></td>
<td>Pattern_250 standard</td>
<td>0xA7</td>
<td>Exact match required</td>
</tr>
<tr>
<td></td>
<td>Pattern_125</td>
<td>&lt;SFD5, SFD6&gt;</td>
<td>—</td>
</tr>
</tbody>
</table>
REGISTER 9-17: RATECON (RATE CONFIGURATION REGISTER)  
ADDRESS: 0x36

<table>
<thead>
<tr>
<th>bit 7</th>
<th>bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIS2000: Disable 2 Mbps Frame Reception bit</td>
<td>Out of scope</td>
</tr>
<tr>
<td>DIS1000: Disable 1 Mbps Frame Reception bit</td>
<td></td>
</tr>
<tr>
<td>DIS500: Disable 500 kbps Frame Reception bit</td>
<td></td>
</tr>
<tr>
<td>DIS250: Disable 250 kbps Frame Reception bit</td>
<td></td>
</tr>
<tr>
<td>DISSTD: Disable IEEE 802.15.4 compliant Frame Reception bit</td>
<td></td>
</tr>
<tr>
<td>DIS125: Disable 125 kbps Frame Reception bit</td>
<td></td>
</tr>
<tr>
<td>OPTIMAL: Optimized Preamble Selection bit</td>
<td></td>
</tr>
</tbody>
</table>

Legend:
R = Readable bit  W = Writable bit  U = Unimplemented bit, read as ’0’
- = Value at POR  ’1’ = Bit is set  ’0’ = Bit is cleared  x = Bit is unknown
r = Reserved

- If this bit is set, then reception of 2 Mbps frames is disabled.
- If this bit is set, then reception of 1 Mbps frames is disabled.
- If this bit is set, then reception of 500 kbps frames is disabled.
- If this bit is set, the reception of 250 kbps frames with non-standard-compliant SFD patterns is disabled.
- If this bit is set, then reception of 250 kbps frames with IEEE 802.15.4 compliant SFD patterns is disabled.
- If this bit is set, then reception of 125 kbps frames is disabled.
- When this bit is set, then optimized preamble is used instead of legacy.
  - 1 = Optimized preamble
  - 0 = Legacy preamble
REGISTER 9-18: SFD1 (START FRAME DELIMITER PATTERN 1 CONFIGURATION REGISTER)
ADDRESS: 0x60

<table>
<thead>
<tr>
<th>Bit</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>SFD1&lt;7:0&gt;</td>
<td>Start Frame Delimiter Pattern 1 Register Field bits</td>
</tr>
</tbody>
</table>

Legend: R = Readable bit, W = Writable bit, U = Unimplemented bit, read as ‘0’
-n = Value at POR, ‘1’ = Bit is set, ‘0’ = Bit is cleared, x = Bit is unknown
r = Reserved

When OPTIMAL = 0:
The hexadecimal digits must be different from 0x0 and different from the corresponding digits in SFD<k>, k = 2, 3, 4, 6, and the value 0xA7 is forbidden.

When OPTIMAL = 1:
The hexadecimal digits must be different from 0x0 and different from the corresponding digits of SFD2.

REGISTER 9-19: SFD2 (START FRAME DELIMITER PATTERN 2 CONFIGURATION REGISTER)
ADDRESS: 0x61

<table>
<thead>
<tr>
<th>Bit</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>SFD2&lt;7:0&gt;</td>
<td>Start Frame Delimiter Pattern 2 Register Field bits</td>
</tr>
</tbody>
</table>

Legend: R = Readable bit, W = Writable bit, U = Unimplemented bit, read as ‘0’
-n = Value at POR, ‘1’ = Bit is set, ‘0’ = Bit is cleared, x = Bit is unknown
r = Reserved

When OPTIMAL = 0:
The hexadecimal digits must be different from 0x0 and different from the corresponding digits in SFD<k>, k = 1, 3, 4, 6, and the value 0xA7 is forbidden.

When OPTIMAL = 1:
The hexadecimal digits must be different from 0x0 and different from the corresponding digits of SFD1.
MRF24XA

REGISTER 9-20: SFD3 (START FRAME DELIMITER PATTERN 3 CONFIGURATION REGISTER)
ADDRESS: 0x62

<table>
<thead>
<tr>
<th>RW-00111011</th>
</tr>
</thead>
<tbody>
<tr>
<td>SFD3&lt;7:0&gt;</td>
</tr>
</tbody>
</table>

**Legend:**
- R = Readable bit
- W = Writable bit
- U = Unimplemented bit, read as ‘0’
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- x = Bit is unknown
- -n = Value at POR
- r = Reserved

**bit 7-0 SFD3<7:0>:** Start Frame Delimiter Pattern 3 Register Field bits
This octet is used as SFD pattern with 500 kbps rate.

When OPTIMAL = 0:
The hexadecimal digits must be different from 0x0 and different from the corresponding digits in SFD<k>, k = 1, 2, 4, 6, and the value 0xA7 is forbidden.

When OPTIMAL = 1:
The hexadecimal digits must be different from 0x0.

REGISTER 9-21: SFD4 (START FRAME DELIMITER PATTERN 4 CONFIGURATION REGISTER)
ADDRESS: 0x63

<table>
<thead>
<tr>
<th>RW-11100101</th>
</tr>
</thead>
<tbody>
<tr>
<td>SFD4&lt;7:0&gt;</td>
</tr>
</tbody>
</table>

**Legend:**
- R = Readable bit
- W = Writable bit
- U = Unimplemented bit, read as ‘0’
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- x = Bit is unknown
- -n = Value at POR
- r = Reserved

**bit 7-0 SFD4<7:0>:** Start Frame Delimiter Pattern 4 Register Field bits
This octet is used as SFD pattern with 250 kbps rate when proprietary MAC is in use, otherwise, the 0xA7 pattern defined in the standard <1> is used instead.

The hexadecimal digits must be different from 0x0 and from the corresponding digits in SFD<k>, where, k = 1, 2, 3, 6 when OPTIMAL = 0. The value 0xA7 is forbidden.
REGISTER 9-22: SFD5 (START FRAME DELIMITER PATTERN 5 CONFIGURATION REGISTER)

ADDRESS: 0x64

<table>
<thead>
<tr>
<th>RW-01001101</th>
</tr>
</thead>
<tbody>
<tr>
<td>SFD5&lt;7:0&gt;</td>
</tr>
</tbody>
</table>

bit 7-0  SFD5<7:0>: Start Frame Delimiter Pattern 5 Register Field bits

This octet is used as the MSB of the SFD pattern with 125 kbps rate.

Legend:  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as ‘0’  
‘1’ = Bit is set  
‘0’ = Bit is cleared  
x = Bit is unknown  
r = Reserved

REGISTER 9-23: SFD6 (START FRAME DELIMITER PATTERN 6 CONFIGURATION REGISTER)

ADDRESS: 0x65

<table>
<thead>
<tr>
<th>RW-10101000</th>
</tr>
</thead>
<tbody>
<tr>
<td>SFD6&lt;7:0&gt;</td>
</tr>
</tbody>
</table>

bit 7-0  SFD6<7:0>: Start Frame Delimiter Pattern 6 Register Field bits

This octet is used as the LSB of the SFD pattern with 125 kbps rate. When OPTIMAL = 1, this octet is used as the LSB of the SFD pattern with 2 Mbps rate.

The hexadecimal digits must be different from 0x0 and different from the corresponding digits in SFD<k>, k = 1, 2, 3, 4 when OPTIMAL = 0. The value 0xA7 is forbidden.

REGISTER 9-24: SFD7 (START FRAME DELIMITER PATTERN 7 CONFIGURATION REGISTER)

ADDRESS: 0x66

<table>
<thead>
<tr>
<th>RW-11001000</th>
</tr>
</thead>
<tbody>
<tr>
<td>SFD7&lt;7:0&gt;</td>
</tr>
</tbody>
</table>

bit 7-0  SFD7<7:0>: Start Frame Delimiter Pattern 7 Register Field bits

When OPTIMAL = 1, this octet is used as the LSB of the SFD pattern with 1 Mbps rate.
9.8 Start-of-Frame Delimiter (SFD) Detection

The following sections describe the SFD detection mechanism for the different data rates. Header processing is required to work at least as reliably as the demodulation. To meet this requirement, longer preamble and 16-bit SFD is defined for frames where the payload data rate is lower than the air-data rate of the preamble.

9.8.1 SFD DETECTION AT 125 kbps

The input contains a nibble of bits decoded from the received DSSS symbol. This input is updated on every new DSSS symbol received.

In each update, the latest four received nibbles are compared against the nibbles contained in the 16-bit SFD pattern that the host MC configures. At least three out of four nibbles must match to trigger an SFD_FOUND event.

SFD_TIMEOUT occurs if the latest five nibbles are different from “0000” (preamble lost) while SFD_FOUND is not triggered. Reception is reset on SFD_TIMEOUT.

For the 125 kbps data rate the last decoded four nibbles and the nibbles of pattern_125 must match in at least three nibble positions.

9.8.2 SFD DETECTION AT 1 Mbps

The input contains a byte, which is updated on every new byte received after the first preamble byte has been detected (this identifies the byte boundary). The two latest received bytes form a Word of 16 bits is denoted by W.

SFD_FOUND event is reported if W exactly matches the host configured 16-bit preamble pattern (SFD), or if an approximate match is found with the following error patterns:

• SFD XOR W = 110...0110…0 (two error bursts of length 2)
• SFD XOR W = 10...0110…01 (single error burst of length 2, and single error on either or both ends) SFD XOR W = 0...0110…0 (single error burst of length 2)
• SFD XOR W = 0...010…0 (single error)

The rationale behind selecting these patterns is that the maximum-likelihood demodulator tends to produce error bursts of length 2 due to the trellis of the MSK modulation (this particular tolerance scheme seems to be novel).

SFD_TIMEOUT event is reported if the latest three octets are different from 0x0F0F0F, while SFD_FOUND is not triggered. Reception is reset on SFD_TIMEOUT.

At 1 Mbps the match tolerates single bit or maximum 2 non-contiguous 2-bit burst differences in the comparison of the last received 16 bits and pattern_1000 (simultaneously isolated single bit mismatches at both ends of the pattern constitute a single 2-bit mismatch burst).
9.9 Physical Transmissions

MCU sets TXST and transmission starts when TXST is set to ‘1’. TXMAIF, TXSFDIF and RXSFDIF flags are handled. RXSFDIF is handled even with ACK. External PA/LNA is automatically handled.

For more information on this mode, see Section 9.13 “External Power Amplifier (PA)/Low-Noise Amplifier (LNA)”.

Channel, data rate and link adaptation is based on retransmission, and the information is from the receiver.

REGISTER 9-25: TXPOW (TRANSMIT POWER CONFIGURATION REGISTER) ADDRESS 0x3A

<table>
<thead>
<tr>
<th>R/W-000</th>
<th>RW-11111</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHIPBOOST&lt;2:0&gt;</td>
<td>TXPOW&lt;4:0&gt;</td>
</tr>
</tbody>
</table>

bit 7-5 CHIPBOOST<2:0>: TX Chip Boosting Field bits

This field modifies the spectrum of the OQPSK transmission.

bit 4-0 TXPOW<4:0>: TX Power Register Field bits

This field enables configuring the TX power ranging from -19 to 1 dBm. Encoding:

11111 = +1 dBm

•

00001 = -19 dBm

00000 = PA OFF

FIGURE 9-5: OUTPUT POWER IN FUNCTION OF TXPOW

FIGURE 9-6: TX CURRENT IN FUNCTION OF TXPOW

Note: Some variance is expected due to a different matching network.
9.10 Signal Detection (Power-Save Listen Mode)

In Power-Save Listen Mode, only the RX front end circuit is powered, the baseband is switched off. In this mode, approximately 3 mA receive current is saved. To use this mode, set the PSAV bit to ‘1’.

**Note:** In this mode, MRF24XA consumes less current that causes sensitivity degradation.

**TABLE 9-8: RECOMMENDED SETTINGS FOR POWER-SAVE LISTEN MODE**

<table>
<thead>
<tr>
<th></th>
<th>125/250 Legacy</th>
<th>500 kbps</th>
<th>1 Mbps</th>
<th>2 Mbps</th>
</tr>
</thead>
<tbody>
<tr>
<td>DesensThr</td>
<td>0x3</td>
<td>0x4</td>
<td>0x5</td>
<td>0x5</td>
</tr>
<tr>
<td>PsavThr</td>
<td>0x9</td>
<td>0xC</td>
<td>0xF</td>
<td>0xF</td>
</tr>
</tbody>
</table>

**REGISTER 9-26: RATECON (RATE CONFIGURATION REGISTER) ADDRESS: 0x36**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>PSAV</th>
</tr>
</thead>
<tbody>
<tr>
<td>RW-0</td>
<td>RW-0</td>
<td>RW-0</td>
<td>RW-0</td>
<td>RW-0</td>
<td>RW-0</td>
<td>RW-1</td>
</tr>
<tr>
<td>DIS2000</td>
<td>DIS1000</td>
<td>DIS500</td>
<td>DIS250</td>
<td>DISSTD</td>
<td>DIS125</td>
<td>OPTIMAL</td>
</tr>
</tbody>
</table>

Legend:

- **R** = Readable bit
- **W** = Writable bit
- **U** = Unimplemented bit, read as ‘0’
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- **x** = Bit is unknown
- **r** = Reserved

bit 7-1 Out of scope.

bit 0 **PSAV**: Power-Save Mode Selection bit

When this bit is set, frame detection is dependent on the RSSI signal, and the receive signal processor is turned on when a sudden and significant increase (PSAVTHR<3:0>) is detected in the signal strength or the signal strength is above an absolute level (DESENSTHR<3:0>).

1 = Power-Save mode
0 = Hi-Sensitivity mode
REGISTER 9-27:  POWSAVE (POWER-SAVE CONFIGURATION REGISTER)  

ADDRESS 0x37

<table>
<thead>
<tr>
<th>bit 7-4</th>
<th>DESENSTHR&lt;3:0&gt;: Desensitization Threshold Field bits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>This field defines an absolute level on the RSSI signal to activate the receive signal processor if PSAV = 1.</td>
</tr>
<tr>
<td></td>
<td>Unit is: 4 dB/LSB. Unsigned encoding is used.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 3-0</th>
<th>PSAVTHR&lt;3:0&gt;: Frame Detection Threshold Register Field bits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>This field defines a relative (relative to the last 4 μs RSSI value) threshold level on the RSSI signal to activate the receive signal processor if PSAV = 1.</td>
</tr>
<tr>
<td></td>
<td>Unit is 0.5 dB/LSB. Unsigned encoding is used.</td>
</tr>
</tbody>
</table>

TABLE 9-9:  REGISTERS ASSOCIATED WITH POWER-SAVE LISTEN MODE

<table>
<thead>
<tr>
<th>Name</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>RATECON</td>
<td>DIS2000</td>
<td>DIS1000</td>
<td>DIS500</td>
<td>DIS250</td>
<td>DISSTD</td>
<td>DIS125</td>
<td>OPTIMAL</td>
<td>PSAV</td>
</tr>
<tr>
<td>POWSAVE</td>
<td>DESENSTHR&lt;3:0&gt;</td>
<td>PSAVTHR&lt;3:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend:  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as ‘0’  
-n = Value at POR  
‘1’ = Bit is set  
‘0’ = Bit is cleared  
x = Bit is unknown  
r = Reserved
9.11 AFC

AFC circuit of MRF24XA measures Carrier Frequency Offset (CFO), for all the received packets. The measured value is interpreted as the frequency offset between the two communicating nodes.

**Note:** AFC circuit stores CFO value in CFO-MEAS field after the SFD is detected and clears the field as the frame processing is finished and RXIF interrupt is generated. CFOTX is used as digital CFO compensation for transmitting. CFORX is used as digital CFO compensation for receiving.

**REGISTER 9-28: CFOCON (CFO PRE COMPENSATION REGISTER) ADDRESS: 0x34**

<table>
<thead>
<tr>
<th>RW-0000</th>
<th>RW-0000</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFOTX&lt;3:0&gt;</td>
<td>CFORX&lt;3:0&gt;</td>
</tr>
</tbody>
</table>

**Legend:**
- R = Readable bit
- W = Writable bit
- U = Unimplemented bit, read as ‘0’
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- x = Bit is unknown
- r = Reserved

**bit 7-4** CFOTX<3:0>: TX Carrier Frequency Offset Field bits

The host writes this value to compensate for the Carrier Frequency Offset of the node during transmission. Pre-compensation enables using crystals with wider tolerances.

Frequency Offset Unit is: 13 ppm/LSB. Two’s complement encoding.

**bit 3-0** CFORX<3:0>: RX Carrier Frequency Offset Field bits

The host writes this value to pre-compensate the Carrier Frequency Offset estimation window (±55 ppm).

Frequency Offset Unit is: 13 ppm/LSB. Two’s complement encoding.
## REGISTER 9-29: CFOMEAS (CFO MEASUREMENT INDICATION REGISTER)  
ADDRESS: 0x35

<table>
<thead>
<tr>
<th>bit 7-0</th>
<th>CFOMEAS&lt;7:0&gt;: CFO Measurement Field bits</th>
</tr>
</thead>
</table>
| bit 7-0 | If AFCOVR bit is cleared, then this register is written and valid when RXSFDIF is set with the value of the Carrier Frequency Offset that was estimated during the acquisition of the packet. The host may use this value together with the LQI as a preamble quality indication (the LQI is measured over the CFO compensated payload).
| bit 7-0 | If AFCOVR bit is set, this receiver compensates the Carrier Frequency Offset. Note that in this case, the CFO estimation algorithm is disabled, thus ±13 ppm CFO is tolerated. CFORX has no effect when AFCOVR is set.
| bit 7-0 | Frequency Offset Unit is: ~1.62 ppm/LSB of the 2.4 GHz carrier. Two’s complement encoding is used. |

### 9.12 Receive Status Vector (RSV)\(^{(1,2)}\)

The Receive Status Vector (RSV) can extend the received packet that gives extra information about the link. RSV bits are individually enabled.

**Note 1:** RSV does not affect the LENGTH field of the packet.

**Note 2:** LQI, RSSI, CHDR and CFO are the order of appending the CRC.
REGISTER 9-30: RXCON1 (MAC RECEIVE CONTROL 1 REGISTER)

ADDRESS: 0x15

<table>
<thead>
<tr>
<th>RW/HC/HS-0</th>
<th>R/W-0</th>
<th>RW/HC-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>R/W-0</th>
<th>R-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>RXEN</td>
<td>NOPA</td>
<td>RXDEC</td>
<td>RSVLQIEN</td>
<td>RSVRSSIEN</td>
<td>RSVCHDREN</td>
<td>RSVCFOEN</td>
<td>r</td>
<td></td>
</tr>
</tbody>
</table>

Legend:  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'

-n = Value at POR  
'1' = Bit is set  
'0' = Bit is cleared  
x = Bit is unknown

bit 7  
**RXEN:** Receive Enable Field bit

This bit enables/disables the packet reception. If an RX packet is currently being received, clearing this bit causes that packet to be discarded.

1 = RX enabled  
0 = RX disabled

Hardware clear/set when:

- Cleared when TRXMODE is set to TX-Streaming mode
- Set when TRXMODE is set to RX-Streaming mode

Clearing this bit aborts the current operation in the following cases:

- Receiving a packet in Packet mode or in RX-Streaming mode

Changes to most RX related settings must be only done while this bit is cleared.

The clear channel assessment (CSMAEN) and ACK-frame reception does not require RXEN = 1 as the device turns the radio into RX when needed, irrespective of the status of the RXEN bit.

bit 6  
**NOPA:** No Parsing bit

This bit disables packet parsing. Only CRC is checked, if it is enabled. This feature is useful in Sniffer mode.

1 = Disable packet parsing  
0 = Enable packet parsing

bit 5  
**RXDEC:** RX Decryption bit

Setting this bit starts RX security processing (authentication or decryption, or both) on the last received packet.

1 = RX security processing started/in process. RXDECIF or RXTAGIF is set.  
0 = RX security processing inactive or complete

This bit clears itself after RX decryption has completed.

bit 4  
**RSVLQIEN:** Receive Status Vector LQI Enable bit

If bit is set, the measured Link Quality is appended after the received frame in the packet buffer.

1 = Append LQI field  
0 = Do not append LQI field

bit 3  
**RSVRSSIEN:** Receive Status Vector RSSI Enable bit

If bit is set, the measured RSSI is appended after the received frame in the packet buffer.

1 = Append RSSI field  
0 = Do not append RSSI field
MRF24XA

9.13 External Power Amplifier (PA)/Low-Noise Amplifier (LNA)

MRF24XA has a PA control pin (pin 20) and an LNA control pin (pin 21) to handle external PAs and LNAs or external antenna switch circuits. MRF24XA can also tolerate different start-up times of different external circuits by sending or accepting data if the external circuits complete their ramp up. MRF24XA can handle both active-high or active-low control signal sensitive circuits.

9.13.1 EXTERNAL PA HANDLING

MRF24XA can automatically switch ON and OFF external PA circuits as the internal functionalities require to transmit any signal. PA pin is automatically set to its preset active state as external PA is needed and set back to its inactive state if PA is not needed.

To enable external PA handling, set PAEN bit of EXTPA register to ‘1’. EXTPAP bit sets the active state of PA control line. The current value of EXTPAP bit is the active state of the PA line. RFOP<2:0>, field of the Radio Operation Register shows the status of the radio and external PA.

9.13.1.1 PA Switch Time Management

MRF24XA is used with various external PA circuits. Different PA circuits may have different start-up time constraints to reach the steady state. MRF24XA can manage to start transmitting if both the internal and external PA circuits are ready to operate.

TX2TXMA<4:0>, Transmit Power-Up to Medium Access Configuration, defines the time delay that MRF24XA waits after powering on the external PA before sending any data to transmit. Figure 9-7 illustrates the method of PA time management of MRF24XA.

PA2TXMA<4:0>, External Power Amplifier Power-Up to Medium Access Configuration, defines the time delay that MRF24XA waits after powering on the external PA before sending any data to transmit.

REGISTER 9-30: RXCON1 (MAC RECEIVE CONTROL 1 REGISTER) (CONTINUED)

bit 2  RSVCHDREN: Receive Status Vector Channel/MAC Type/Data Rate Enable bit

If bit is set, Channel, MAC type and Data Rate configurations used with the received frame are appended after the received frame in the packet buffer, using the encoding specified for CH<3:0>, FRMFMT and DR<2:0> (concatenated in this order when MSb is first).

1 = Append Channel, MAC type and Data Rate fields
0 = Do not append Channel, MAC type and Data Rate fields

bit 1  RSVCFQFOEN: Receive Status Vector CFO Enable bit

If bit is set, the estimated Carrier Frequency Offset of the received frame is appended after the received frame in the packet buffer, using the same encoding as CFOMEAS register.

1 = Append CFO estimation
0 = Do not append estimated CFO

bit 0  Reserved: Maintain as ‘0’
### FIGURE 9-7: PA ACCESS TIME MANAGEMENT

<table>
<thead>
<tr>
<th>Switching ON</th>
<th>Switching ON</th>
<th>Start Transmitting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal PA</td>
<td>External PA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T0 - TX2TXMA</td>
<td>T0 - PA2TXMA</td>
<td>T0</td>
</tr>
</tbody>
</table>

Note: The diagram illustrates the timing for switching on the internal and external PAs and the start of transmitting.
9.13.2 EXTERNAL LNA HANDLING

MRF24XA can automatically switch ON and OFF external LNA circuit as the internal functionalities require receiving. LNA pin is automatically set to its predefined active state as external LNA circuit is needed, and set back to its inactive state if LNA is not needed.

To enable external LNA handling, set LNAEN bit of the EXTLNA register to ‘1’. EXTLNAP bit sets the active state of LNA line. The actual value of EXTLNAP bit is the active state of the LNA line.

MRF24XA is programmed to delay signal receiving after powering on the external LNA circuit. It enables to optimize the power consumption to the startup time of the external LNA circuit. LNADLY<4:0> defines the time delay between LNA power-up and the start of signal reception. The time base is 1 µs. Higher LNADLY value means longer Wait before starting reception. RFOP<2:0> field of the Radio Operation register shows the status of the radio and external LNA.

<table>
<thead>
<tr>
<th>REGISTER 9-31: OPSTATUS (OPERATION STATUS)</th>
<th>Address: 0x02</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-0</td>
<td>R/HS/HC-0</td>
</tr>
<tr>
<td>r</td>
<td>MACOP&lt;3:0&gt;</td>
</tr>
</tbody>
</table>

Legend:  
R = Readable bit     W = Writable bit     U = Unimplemented bit, read as '0'  
-n = Value at POR     ‘1’ = Bit is set     ‘0’ = Bit is cleared     x = Bit is unknown  
r = Reserved     HC = Hardware Clear     HS = Hardware Set

bit 7 Reserved: Maintain as ‘0’
bit 6-3 Out of scope
bit 2-0 RFOP<2:0>: Radio Operation Register Field bits  
Provides status information on the current Radio state. Encoding on RFOP<2:0>:  
111 = TX with external PA is turned on (TX+PA)  
110 = RX with external LNA is turned on (RX+LNA)  
101 = Synthesizer and external PA or LNA is turned on (SYNTH+PA/LNA)
REGISTER 9-32: TX2TXMA (TRANSMIT POWER-UP TO MEDIUM ACCESS CONFIGURATION REGISTER) ADDRESS 0x3C

<table>
<thead>
<tr>
<th>bit 7-5</th>
<th>bit 4-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reserved: Maintain as ‘0’</td>
<td>TX2TXMA&lt;4:0&gt;: Transmit Power-Up to Medium Access Configuration Field bits</td>
</tr>
</tbody>
</table>

Defines the time interval between turning on the transmitter of the device and the start time of medium access (start of the PHY-layer frame).

TX_TO_TXMA = The transient time of the transmitter, in the following scenarios:

- PAEN = 0
- PAEN = 1, but the PA is turned on first. PA_TO_TXMA = TX_TO_TXMA + PA transient time.
- PAEN = 1, but the TX and PA transients are NOT sequenced.

TX_TO_TXMA = The transient time of the transmitter + PA_TO_TXMA:

PAEN = 1, and the transmitter is turned on first (transients are sequenced).

Representation: 1 \( \mu s \)/1 LSB. No offset.
### REGISTER 9-33: EXTPA (EXTERNAL POWER AMPLIFIER CONFIGURATION REGISTER)

**ADDRESS 0x3D**

<table>
<thead>
<tr>
<th>Bit</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Reserved</td>
<td>Maintain as '0'</td>
</tr>
<tr>
<td>6</td>
<td>EXTPAP</td>
<td>External Power Amplifier Polarity bit</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>= 3.3V turns Power Amplifier ON</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>= GND turns Power Amplifier ON</td>
</tr>
<tr>
<td>5</td>
<td>PAEN</td>
<td>External Power Amplifier Enable bit</td>
</tr>
<tr>
<td></td>
<td>This bit enables the PA pin to output the control signal for external Power Amplifier.</td>
<td></td>
</tr>
<tr>
<td>4-0</td>
<td>PA2TXMA&lt;4:0&gt;</td>
<td>External Power Amplifier Power-Up to Medium Access Configuration Field bits</td>
</tr>
<tr>
<td></td>
<td>Defines the time interval between turning on the external PA of the device and the start time of medium access (start of the PHY-layer frame).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PA_TO_TXMA = The transient time of the external PA in the following scenarios:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PAEN = 1, and the transmitter is turned on first. TX_TO_TXMA = PA_TO_TXMA + TX transient time.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PAEN = 1, but the TX and PA transients are NOT sequenced.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PA_TO_TXMA = The transient time of the PA + TX_TO_TXMA:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PAEN = 1, and the external power amplifier is turned on first (Transients are sequenced).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Representation: 1 μs/1 LSB. No offset</td>
<td></td>
</tr>
</tbody>
</table>

**Legend:**
- **R** = Readable bit
- **W** = Writable bit
- **U** = Unimplemented bit, read as ‘0’
- **-n** = Value at POR
- ‘1’ = Bit is set
- ‘0’ = Bit is cleared
- **x** = Bit is unknown
- **r** = Reserved
REGISTER 9-34: EXTLNA (EXTERNAL LOW-NOISE AMPLIFIER CONFIGURATION REGISTER)

ADDRESS 0x3E

<table>
<thead>
<tr>
<th>R-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-00100</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>EXTLNAP</td>
<td>LNAEN</td>
<td>LNADLY&lt;4:0&gt;</td>
</tr>
</tbody>
</table>

bit 7

Legend:  
R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as ‘0’  
-n = Value at POR  
‘1’ = Bit is set  
‘0’ = Bit is cleared  
x = Bit is unknown  
r = Reserved

bit 6  
**EXTLNAP**: External Low Noise Amplifier Polarity bit  
1 = 3.3V turns Low-Noise Amplifier ON  
0 = GND turns Low-Noise Amplifier ON

bit 5  
**LNAEN**: External Low-Noise Power Amplifier Enable bit  
This bit enables the LNA pin to output the control signal for external Low-Noise Amplifier.

bit 4-0  
**LNADLY<4:0>**: External Low-Noise Amplifier Power-Up Transient Delay Field bits  
Defines the duration between turning on the LNA and the time when the reception is valid.  
LNA and receiver are turned on together. The longer transient is awaited before input signal is accepted as valid.  
Representation: 1 µs/1 LSB. No offset.

**TABLE 9-10: REGISTERS ASSOCIATED WITH EXTERNAL PA AND LNA**

<table>
<thead>
<tr>
<th></th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPSTATUS</td>
<td>r</td>
<td></td>
<td>MACOP&lt;3:0&gt;</td>
<td></td>
<td>RFO&lt;2:0&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TX2TXMA</td>
<td>r</td>
<td></td>
<td></td>
<td>TX2TXMA&lt;4:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXTPA</td>
<td>r</td>
<td>EXTPAP</td>
<td>PAEN</td>
<td></td>
<td>PA2TXMA&lt;4:0&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXTLNA</td>
<td>r</td>
<td>EXTLNAP</td>
<td>LNAEN</td>
<td></td>
<td>LNADLY&lt;4:0&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend:  
r = Reserved, read as ‘0’.  
© 2015 Microchip Technology Inc.  
Preliminary  
DS70005023C-page 235
10.0 BATTERY LIFE OPTIMIZATION

In a battery-operated application, the device only wakes up when it must transmit or requires to poll for data. Polling is used for data reception as a means to synchronize the remotely transmitting node to the wake-up event in the receiver. Between transmission and reception the device must be held in Deep Sleep mode drawing less current than the battery self-discharge, which is about 1 µA. Register contents and internal Calibration state are maintained in Deep Sleep mode for efficient power mode changes. Long battery life is achieved through low currents in each state of the device and a series of system features that contribute to minimize the duration required for transmit or receive.

The following enhanced features are used to minimize radio ON time:

- High air-data-rates to minimize the packet duration
- Automatic, on-the-fly, per frame, air-data-rate adaptation in the receiver, allowing the transmitter to select the highest data rate that fits the quality of the link
- Minimized framing overheads in both the PHY and the MAC layers
- Minimized ramp-up and turnaround times
- Short, still reliable channel assessment
- Automatically handled TX and RX signal paths
- Inferred destination addressing

On-the-fly, per frame air-data-rate detection is the capability of the receiver to synchronize to the transmitter data rate without knowing the sender of the frame and the expected data rate in advance. On-the-fly, per frame, air-data-rate detection gives the following advantages:

- Each low-power node can use the highest data rate allowed by its link quality to save its battery charge. The evaluation of the link quality requires MCU interaction.
- Multiple data rates are used within the same network.

As opposed to conventional protocols supporting the simultaneous use of multiple air-data-rates in the network traffic, the frame header, which encodes the payload data rate, does not use the lowest data rate. Without this feature either the worst link defines the air-data-rate that all nodes must use, or each node must use the lowest data rate for the frame header, which can severely compromise the throughput and battery efficiency of the highest rates.

Passive listening, channel assessment and the duration of the turnaround between transmit and receive contribute to the power consumption.

In this regard, MRF24XA excels by minimized TX-to-RX turnaround durations, fast but reliable channel assessment and short PLL and AGC ramp-up durations. Power modes are automatically sequenced during CSMA sending by the internal state machines of the device without interaction from the MCU. These mechanisms can optionally control external PA and LNA.

The Message Chart in Figure 10-1 illustrates a typical wake-up cycle:

1. While the low-power device is in Deep Sleep mode, the coordinator listens to the channel and buffers any messages addressed to the low-power node.
2. The low-power node wakes up when must transmit, or periodically to poll the coordinator for any pending data.
3. First, the low-power node sends a poll command to the coordinator and any data it must send.
4. Low-power node can go back to Deep Sleep mode as soon as it gets an ACK unless the coordinator has buffered pending data. This condition is indicated in a specific bit field of the acknowledge frame that the coordinator is sending.
5. In the case of pending data, the low-power node may want to turn off the radio for a predetermined duration allowing the coordinator to retrieve the pending data for sending.
6. Finally, the coordinator turns to Receive mode to get the pending data. On successful reception, it turns to transmit to send an ACK and returns to Deep Sleep mode.
7. Time-outs ensure that the low-power node does not stay powered-up forever in the case when the coordinator fails to respond in any of the transactions above.

Figure 10-1 shows that all the radio activities are minimized to immediately return to Deep Sleep mode.

As a result, the average current consumption is reduced by multiple factors in comparison to the standard IEEE 802.15.4 operation. The comparison is done for three corner cases as follows:

- Table 10-1 for polling without pending data
- Table 10-2 for polling with 80 octets pending data
- Table 10-3 for the transmission of 80 octets

A combination of the three cases allow evaluating the energy budget of complex scenarios. A yearly 10 mAh is to be added for battery self-discharge and Deep Sleep mode. Equation 10-1 shows the self discharge current calculation. The consumption of the MCU and any sensors, displays must be added.
The enhanced MAC and PHY feature set also compresses the frame header time to achieve the shortest possible radio ON time.

To reduce leakage current, ensure that GPIO weak pull up is turned off in case it is not needed in Deep Sleep mode.

**EQUATION 10-1: BATTERY SELF-DISCHARGE**

The discharge caused by 1 µA average current over one year: 1µA x 1 year = 8.76 mAh

**TABLE 10-1: POLLING FOR PENDING DATA – NO PENDING DATA IS AVAILABLE\(^{(1)}\)**

<table>
<thead>
<tr>
<th>Wake-up</th>
<th>Consumed Battery Charge</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mode</td>
<td>Period</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>Single wake-up</td>
<td>per wake-up</td>
<td>4480</td>
</tr>
<tr>
<td>Yearly average</td>
<td></td>
<td>39.3</td>
</tr>
<tr>
<td>while waking up</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>regularly in every</td>
<td></td>
<td>0.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.1</td>
</tr>
</tbody>
</table>

**Note 1:** The calculations are strongly depended on the used protocol. It may happen that a given protocol cannot produce the listed battery life values.

**TABLE 10-2: POLLING FOR PENDING DATA – 80 OCTETS OF PENDING DATA RECEIVED\(^{(1)}\)**

<table>
<thead>
<tr>
<th>Wake-up</th>
<th>Consumed Battery Charge</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mode</td>
<td>Period</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>Single wake-up</td>
<td>per wake-up</td>
<td>12030</td>
</tr>
<tr>
<td>Yearly average</td>
<td></td>
<td>163.7</td>
</tr>
<tr>
<td>while waking up</td>
<td></td>
<td>8.2</td>
</tr>
<tr>
<td>regularly in every</td>
<td></td>
<td>0.5</td>
</tr>
</tbody>
</table>

**Note 1:** The calculations are strongly depended on the used protocol. It may happen that a given protocol cannot produce the listed battery life values.

**TABLE 10-3: POLLING FOR PENDING DATA – TRANSMITTING 80 OCTETS TO COORDINATOR (AS PIGGYBACK DATA – NO PENDING RECEIVED DATA)\(^{(1)}\)**

<table>
<thead>
<tr>
<th>Wake-up</th>
<th>Consumed Battery Charge</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mode</td>
<td>Period</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>Single wake-up</td>
<td>per wake-up</td>
<td>10560</td>
</tr>
<tr>
<td>Yearly average</td>
<td></td>
<td>92.5</td>
</tr>
<tr>
<td>while waking up</td>
<td></td>
<td>4.6</td>
</tr>
<tr>
<td>regularly in every</td>
<td></td>
<td>1.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.3</td>
</tr>
</tbody>
</table>

**Note 1:** The calculations are strongly depended on the used protocol. It may happen that a given protocol cannot produce the listed battery life values.
FIGURE 10-1: MRF24XA POWER MODES DURING DATA POLLING (MESSAGE SEQUENCE CHART)

Coordinator (mains powered)

Listen/Receive

POLL for pending data (+ optional piggyback data)

Acknowledge + indicate if data is pending

Listening/Receive

TX/RX OFF

MCU retrieves the data to send and loads it for transmission

Waits for data retrieval (fixed duration)

Pending DATA

Listen/Receive

Low-power (battery-powered)

Duration is...

Deep Sleep maximized

Wake-up minimized

Clear channel assessment minimized by sensitive detector

RX-to-TX minimized by zero-IF architecture

Transmit minimized by high data rate and low framing overhead

TX-to-RX minimized by zero-IF architecture

Listen/Receive minimized by high data rate and low framing overhead

If data is NOT pending then Deep Sleep else

TX/RX OFF

TRXOFF-to-RX (start PLL)

RX-to-TX minimized by zero-IF architecture

Transmit minimized by high data rate and low framing overhead
11.0 ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings

- Ambient temperature under bias: -40°C to +85°C
- Storage temperature: -40°C to +125°C
- Voltage on any digital or analog pin with respect to VSS (except VDD): -0.3V to (VDD + 0.3V)
- Voltage on VDD with respect to VSS: -0.3V to 6V
- Maximum output current sunk by GPIO0-GPIO2 pins: 2mA at 0.3xVDD
- Maximum output current sourced by GPIO0-GPIO2 pins: 2mA at 0.3xVDD

Note: Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

TABLE 11-1: RECOMMENDED OPERATING CONDITIONS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Min.</th>
<th>Typ</th>
<th>Max.</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ambient Operating Temperature</td>
<td>-40</td>
<td>+25</td>
<td>+85</td>
<td>°C</td>
</tr>
<tr>
<td>Supply Voltage for RF, analog (AVDD) and digital circuits (DVDD)</td>
<td>1.08</td>
<td>1.2</td>
<td>1.32</td>
<td>V</td>
</tr>
<tr>
<td>Supply Voltage for LDO Input (pin 30) and digital I/O (pin 23)</td>
<td>1.5</td>
<td>3.3</td>
<td>3.6</td>
<td>V</td>
</tr>
<tr>
<td>Input High Voltage (VIH)</td>
<td>0.65 x VDD</td>
<td>—</td>
<td>VDD +0.3</td>
<td>V</td>
</tr>
<tr>
<td>Input Low Voltage (VIL)</td>
<td>-0.3</td>
<td>—</td>
<td>0.35 x VDD</td>
<td>V</td>
</tr>
</tbody>
</table>

TABLE 11-2: CURRENT CONSUMPTION

Typical Values: TA = 25°C, VDD = 3.3V

<table>
<thead>
<tr>
<th>Operating mode</th>
<th>Condition</th>
<th>Min.</th>
<th>Typ</th>
<th>Max.</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deep Sleep</td>
<td>All GPIO pins are grounded</td>
<td>—</td>
<td>40</td>
<td>—</td>
<td>nA</td>
</tr>
<tr>
<td>Sleep</td>
<td>—</td>
<td>—</td>
<td>0.3</td>
<td>—</td>
<td>mA</td>
</tr>
<tr>
<td>Crystal ON</td>
<td>—</td>
<td>—</td>
<td>1.1</td>
<td>—</td>
<td>mA</td>
</tr>
<tr>
<td>Synthesizer ON</td>
<td>—</td>
<td>—</td>
<td>7</td>
<td>—</td>
<td>mA</td>
</tr>
<tr>
<td>RX Listen Power-Save</td>
<td>All data rates</td>
<td>—</td>
<td>13.5</td>
<td>—</td>
<td>mA</td>
</tr>
<tr>
<td>RX Listen</td>
<td>All data rates</td>
<td>—</td>
<td>16.5</td>
<td>—</td>
<td>mA</td>
</tr>
<tr>
<td>RX Packet Demodulation</td>
<td>1 Mbps or 2 Mbps</td>
<td>—</td>
<td>15.5</td>
<td>—</td>
<td>mA</td>
</tr>
<tr>
<td>RX Packet Demodulation</td>
<td>500 kbps, 250 kbps or 125 kbps</td>
<td>—</td>
<td>16.5</td>
<td>—</td>
<td>mA</td>
</tr>
<tr>
<td>TX</td>
<td>at maximum power(1)</td>
<td>—</td>
<td>30</td>
<td>—</td>
<td>mA</td>
</tr>
</tbody>
</table>

Note 1: For further details, see Figure 9-5 and Figure 9-6.
### TABLE 11-3: RECEIVER CHARACTERISTICS

Typical Values: $T_A = 25^\circ C$, $V_{DD} = 3.3V$

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Condition</th>
<th>Min.</th>
<th>Typ</th>
<th>Max.</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF Input Frequency</td>
<td>—</td>
<td>2.405</td>
<td></td>
<td>2.480</td>
<td>GHz</td>
</tr>
<tr>
<td>RF Sensitivity</td>
<td>Data Rate: 250 kbps, PER 1%</td>
<td>—</td>
<td>-95</td>
<td>—</td>
<td>dBm</td>
</tr>
<tr>
<td></td>
<td>125 kbps at 0 ppm CFO</td>
<td>—</td>
<td>-103</td>
<td>—</td>
<td>dBm</td>
</tr>
<tr>
<td></td>
<td>at +/- 110 ppm CFO</td>
<td>—</td>
<td>-100</td>
<td>—</td>
<td>dBm</td>
</tr>
<tr>
<td></td>
<td>250 kbps at 0 ppm CFO</td>
<td>—</td>
<td>-100</td>
<td>—</td>
<td>dBm</td>
</tr>
<tr>
<td></td>
<td>at +/- 110 ppm CFO</td>
<td>—</td>
<td>-99</td>
<td>—</td>
<td>dBm</td>
</tr>
<tr>
<td></td>
<td>500 kbps: Legacy and Optimal framing</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>dBm</td>
</tr>
<tr>
<td></td>
<td>at 0 ppm CFO</td>
<td>—</td>
<td>-97</td>
<td>—</td>
<td>dBm</td>
</tr>
<tr>
<td></td>
<td>at +/- 110 ppm CFO</td>
<td>—</td>
<td>-96</td>
<td>—</td>
<td>dBm</td>
</tr>
<tr>
<td></td>
<td>1 Mbps: Legacy framing</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>dBm</td>
</tr>
<tr>
<td></td>
<td>at 0 ppm CFO</td>
<td>—</td>
<td>-92</td>
<td>—</td>
<td>dBm</td>
</tr>
<tr>
<td></td>
<td>at +/- 110 ppm CFO</td>
<td>—</td>
<td>-92</td>
<td>—</td>
<td>dBm</td>
</tr>
<tr>
<td></td>
<td>Optimal framing</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>dBm</td>
</tr>
<tr>
<td></td>
<td>at 0 ppm CFO</td>
<td>—</td>
<td>-91</td>
<td>—</td>
<td>dBm</td>
</tr>
<tr>
<td></td>
<td>at +/- 85 ppm CFO</td>
<td>—</td>
<td>-89</td>
<td>—</td>
<td>dBm</td>
</tr>
<tr>
<td></td>
<td>2 Mbps: Legacy framing</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>dBm</td>
</tr>
<tr>
<td></td>
<td>at 0 ppm CFO</td>
<td>—</td>
<td>-88</td>
<td>—</td>
<td>dBm</td>
</tr>
<tr>
<td></td>
<td>at +/- 110 ppm CFO</td>
<td>—</td>
<td>-88</td>
<td>—</td>
<td>dBm</td>
</tr>
<tr>
<td></td>
<td>Optimal framing</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>dBm</td>
</tr>
<tr>
<td></td>
<td>at 0 ppm CFO</td>
<td>—</td>
<td>-87</td>
<td>—</td>
<td>dBm</td>
</tr>
<tr>
<td></td>
<td>at +/- 85 ppm CFO</td>
<td>—</td>
<td>-86</td>
<td>—</td>
<td>dBm</td>
</tr>
<tr>
<td>Maximum RF Input</td>
<td>—</td>
<td>—</td>
<td>-10</td>
<td>—</td>
<td>dBm</td>
</tr>
<tr>
<td>LO Leakage</td>
<td>Maximal leakage when radio is in RX,</td>
<td>—</td>
<td>-51.8</td>
<td>—</td>
<td>dBm</td>
</tr>
<tr>
<td></td>
<td>measured after balun</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adjacent Channel Rejection</td>
<td>Offset frequency &gt;3.5 MHz, at 0 dBm</td>
<td>—</td>
<td>-35</td>
<td>—</td>
<td>dBm</td>
</tr>
<tr>
<td></td>
<td>output power</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RSSI Range</td>
<td>—</td>
<td>—</td>
<td>75</td>
<td>—</td>
<td>dB</td>
</tr>
<tr>
<td>RSSI Error</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>± 5</td>
<td>dB</td>
</tr>
</tbody>
</table>

### TABLE 11-4: TRANSMITTER CHARACTERISTICS

Typical Values: $T_A = 25^\circ C$, $V_{DD} = 3.3V$

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Condition</th>
<th>Min.</th>
<th>Typ</th>
<th>Max.</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF Carrier Frequency</td>
<td>—</td>
<td>2.405</td>
<td></td>
<td>2.480</td>
<td>GHz</td>
</tr>
<tr>
<td>Maximum RF Output Power</td>
<td>—</td>
<td>—</td>
<td>1</td>
<td>—</td>
<td>dBm</td>
</tr>
<tr>
<td>RF Output Power Control Range</td>
<td>—</td>
<td>—</td>
<td>20</td>
<td>—</td>
<td>dB</td>
</tr>
<tr>
<td>TX Spectrum Mask for O-QPSK Signal</td>
<td>Offset frequency &gt;3.5 MHz, at 0 dBm</td>
<td>—</td>
<td>-35</td>
<td>—</td>
<td>dBm</td>
</tr>
<tr>
<td></td>
<td>output power</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TX EVM</td>
<td>—</td>
<td>17</td>
<td>23</td>
<td>30</td>
<td>%</td>
</tr>
</tbody>
</table>
11.1 Matching Network

Due to some innovative RF solutions built into the chip, the conventional way of matching network design does not work with this radio. For further information, contact support or your sales representative.
12.0 PACKAGING INFORMATION

12.1 Package Marking Information

32-Lead Very Thin Plastic Quad Flat, No Lead Package (MQ) - 5x5x0.9 mm Body [VQFN]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging
32-Lead Very Thin Plastic Quad Flat, No Lead Package (MQ) - 5x5x0.9 mm Body [VQFN]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging

### Table: Dimension Limits

<table>
<thead>
<tr>
<th>Units</th>
<th>MILLIMETERS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MIN</td>
</tr>
<tr>
<td>Number of Pins</td>
<td>N</td>
</tr>
<tr>
<td>Pitch</td>
<td>e</td>
</tr>
<tr>
<td>Overall Height</td>
<td>A</td>
</tr>
<tr>
<td>Standoff</td>
<td>A1</td>
</tr>
<tr>
<td>Terminal Thickness</td>
<td>A3</td>
</tr>
<tr>
<td>Overall Width</td>
<td>E</td>
</tr>
<tr>
<td>Exposed Pad Width</td>
<td>E2</td>
</tr>
<tr>
<td>Overall Length</td>
<td>D</td>
</tr>
<tr>
<td>Exposed Pad Length</td>
<td>D2</td>
</tr>
<tr>
<td>Terminal Width</td>
<td>b</td>
</tr>
<tr>
<td>Terminal Length</td>
<td>L</td>
</tr>
<tr>
<td>Terminal-to-Exposed-Pad</td>
<td>K</td>
</tr>
</tbody>
</table>

Notes:
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated.
3. Dimensioning and tolerancing per ASME Y14.5M
   - BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   - REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-160A Sheet 2 of 2
RECOMMENDED LAND PATTERN

<table>
<thead>
<tr>
<th>Units</th>
<th>Dimension Limits</th>
<th>MIN</th>
<th>NOM</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact Pitch</td>
<td>E</td>
<td>0.50 BSC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optional Center Pad Width</td>
<td>X2</td>
<td></td>
<td>3.70</td>
<td></td>
</tr>
<tr>
<td>Optional Center Pad Length</td>
<td>Y2</td>
<td></td>
<td>3.70</td>
<td></td>
</tr>
<tr>
<td>Contact Pad Spacing</td>
<td>C1</td>
<td>5.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Contact Pad Spacing</td>
<td>C2</td>
<td>5.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Contact Pad Width (X32)</td>
<td>X1</td>
<td>0.30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Contact Pad Length (X32)</td>
<td>Y1</td>
<td>0.80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Contact Pad to Center Pad (X32)</td>
<td>G1</td>
<td>0.20</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Notes:
1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
APPENDIX A: REVISION HISTORY

Revision A (August 2011)
This is the initial released version of the document.

Revision B (March 2013)
Incorporated major formatting and text updates throughout the document.

Revision C (March 2015)
The following lists the modifications:
• Incorporated major formatting
• Revised a few sentences throughout the document
• Added Chapter 12 “Packaging Information”
• Updated PULLDIRGPIOx bit description to add 75 kOhm information
• Updated Product Identification System section to add package information
• Revised Table 2-3 and Figure 2-3.
THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

• **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user’s guides and hardware support documents, latest software releases and archived software
• **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
• **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip’s customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com. Under “Support”, click on “Customer Change Notification” and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

• Distributor or Representative
• Local Sales Office
• Field Application Engineer (FAE)
• Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://microchip.com/support
**PRODUCT IDENTIFICATION SYSTEM**

To order or obtain information, for example, on pricing or delivery, refer to the factory or the listed sales office.

<table>
<thead>
<tr>
<th>PART NO.</th>
<th>M.</th>
<th>X</th>
<th>T</th>
<th>-X</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device</td>
<td>Module</td>
<td>Tape and Reel</td>
<td>Temperature Range</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MRF24XA; VDD range 1.5V to 3.6V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I = -40º C to +85º C (Industrial)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MQ = Very Thin Plastic Quad Flat (5X5 mm Body), 32-lead</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MQ = Ultra Thin Plastic Quad Flat, No-Lead (5X5X0.9 mm Body), 32-lead</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example:

a) MRF24XA -I/ = Industrial temp. tray.
INDEX

A
Absolute Maximum Ratings .............................................. 241
Agility .................................................................................. 189

C
Channel Agility .................................................................. 191
Customer Change Notification Service ............................. 251
Customer Notification Service ....................................... 251
Customer Support ............................................................. 251

D
Deep Sleep mode ............................................................. 237

E
Electrical Characteristics ................................................... 241
Errata .................................................................................... 3

I
Internet Address ................................................................ 251

L
Low Dropout (LDO)............................................................... 7

M
Microchip Internet Web Site .............................................. 251
MRF24XA ............................................................................. 5

P
Packaging Information ...................................................... 245
PIE2 .................................................................................... 25
PIE3 .................................................................................... 26
Power-Saving Features .................................................... 210
proprietary MAC ................................................................ 169

R
Received Signal Strength Indicator (RSSI) ....................... 7
Recommended Operating Conditions ......................... 241
RXLISTEN ........................................................................ 204

S
Synthesizer ....................................................................... 199

W
WWW Address ................................................................. 251
WWW, On-Line Support ..................................................... 3
Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip’s Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip’s code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, flexPWR, JukeBlox, KEELOQ, KEELOQ logo, Kleer, LANCheck, MediaLB, MOST, MOST logo, MPLAB, OptoLyzer, PIC, PICSTART, PIC® logo, RightTouch, SpyNIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

The Embedded Control Solutions Company and mTouch are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, ECAN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, KleerNet, KleerNet logo, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, RightTouch logo, REAL ICE, SQI, Serial Quad I/O, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademarks of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2015, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-63277-137-7
# Worldwide Sales and Service

## AMERICAS

**Corporate Office**
- 2335 West Chandler Blvd.
- Chandler, AZ 85224-6199
- Tel: 480-792-7200
- Fax: 480-792-7277

Technical Support:
- [http://www.microchip.com/support](http://www.microchip.com/support)

Web Address:
- [www.microchip.com](http://www.microchip.com)

**Atlanta**
- Duluth, GA
- Tel: 678-957-9614
- Fax: 678-957-1455

**Austin, TX**
- Tel: 512-227-3370

**Boston**
- Westborough, MA
- Tel: 774-760-0087
- Fax: 774-760-0088

**Chicago**
- Itasca, IL
- Tel: 630-285-0071
- Fax: 630-285-0075

**Cleveland**
- Independence, OH
- Tel: 216-447-0464
- Fax: 216-447-0643

**Dallas**
- Addison, TX
- Tel: 972-818-7423
- Fax: 972-818-2924

**Detroit**
- Novi, MI
- Tel: 248-848-4000

**Houston, TX**
- Tel: 281-894-5983

**Indianapolis**
- Noblesville, IN
- Tel: 317-773-8323
- Fax: 317-773-5453

**Los Angeles**
- Mission Viejo, CA
- Tel: 949-462-9523
- Fax: 949-462-9608

**New York, NY**
- Tel: 631-435-6000

**San Jose, CA**
- Tel: 408-735-9110

**Canada - Toronto**
- Tel: 905-673-0699
- Fax: 905-673-6509

## ASIA/PACIFIC

**Asia Pacific Office**
- Suites 3707-14, 37th Floor
- Tower 6, The Gateway
- Harbour City, Kowloon
- Hong Kong
- Tel: 852-2943-5100
- Fax: 852-2401-3431

**Australia - Sydney**
- Tel: 61-2-9868-6733
- Fax: 61-2-9868-6755

**China - Beijing**
- Tel: 86-10-8569-7000
- Fax: 86-10-8528-2104

**China - Chengdu**
- Tel: 86-28-8865-5511
- Fax: 86-28-8865-7889

**China - Chongqing**
- Tel: 86-23-8980-9588
- Fax: 86-23-8980-9500

**China - Dongguan**
- Tel: 86-769-8702-9880

**China - Hangzhou**
- Tel: 86-571-8792-8115
- Fax: 86-571-8792-8116

**China - Hong Kong SAR**
- Tel: 852-2943-5100
- Fax: 852-2401-3431

**China - Nanjing**
- Tel: 86-25-6407-5533
- Fax: 86-25-6407-5066

**China - Shenyang**
- Tel: 86-24-2334-2829
- Fax: 86-24-2334-2393

**China - Shenzhen**
- Tel: 86-755-8684-2200
- Fax: 86-755-8203-1760

**China - Wuhan**
- Tel: 86-27-5980-5300
- Fax: 86-27-5980-5118

**China - Xian**
- Tel: 86-29-8833-7252
- Fax: 86-29-8833-7256

**ASIA/PACIFIC**

**China - Xiamen**
- Tel: 86-592-2388138
- Fax: 86-592-2388130

**China - Zhuhai**
- Tel: 86-756-3210040
- Fax: 86-756-3210049

**India - Bangalore**
- Tel: 91-80-3090-4444
- Fax: 91-80-3090-4123

**India - New Delhi**
- Tel: 91-11-4160-8631
- Fax: 91-11-4160-8632

**India - Pune**
- Tel: 91-20-5319-1500

**Japan - Osaka**
- Tel: 81-6-6152-7160
- Fax: 81-6-6152-9310

**Japan - Tokyo**
- Tel: 81-3-6880-3770
- Fax: 81-3-6880-3771

**Korea - Daegu**
- Tel: 82-53-744-4301
- Fax: 82-53-744-4302

**Korea - Seoul**
- Tel: 82-2-554-7200
- Fax: 82-2-554-5934

**Malaysia - Kuala Lumpur**
- Tel: 60-3-6201-9857
- Fax: 60-3-6201-9859

**Malaysia - Penang**
- Tel: 60-4-227-8870
- Fax: 60-4-227-4068

**Philippines - Manila**
- Tel: 63-2-634-9065
- Fax: 63-2-634-9069

**Singapore**
- Tel: 65-6334-8870
- Fax: 65-6334-8850

**Taiwan - Hsin Chu**
- Tel: 886-3-5778-366
- Fax: 886-3-5770-955

**Taiwan - Kaohsiung**
- Tel: 886-7-213-7828

**Taiwan - Taipei**
- Tel: 886-2-2508-8600
- Fax: 886-2-2508-0102

**Thailand - Bangkok**
- Tel: 66-2-694-1351
- Fax: 66-2-694-1350

## EUROPE

**Austria - Wels**
- Tel: 43-7242-2244-39
- Fax: 43-7242-2244-393

**Denmark - Copenhagen**
- Tel: 45-4450-2828
- Fax: 45-4485-2829

**France - Paris**
- Tel: 33-1-69-53-63-20
- Fax: 33-1-69-30-90-79

**Germany - Dusseldorf**
- Tel: 49-2129-3766400
- Fax: 49-2129-3766400

**Germany - Munich**
- Tel: 49-89-627-144-0
- Fax: 49-89-627-144-44

**Germany - Pforzheim**
- Tel: 49-7231-424750

**Italy - Milan**
- Tel: 39-0331-742611
- Fax: 39-0331-667681

**Italy - Venice**
- Tel: 39-049-7625286

**Netherlands - Drunen**
- Tel: 31-416-690399
- Fax: 31-416-690340

**Poland - Warsaw**
- Tel: 48-22-3325737

**Spain - Madrid**
- Tel: 34-91-708-08-90
- Fax: 34-91-708-08-91

**Sweden - Stockholm**
- Tel: 46-8-5090-4654

**UK - Wokingham**
- Tel: 44-118-921-5800
- Fax: 44-118-921-5820

1/27/15