This integrated circuit can be damaged by ESD. Texas Instruments recommends that all integrated circuits be handled with appropriate precautions. Failure to observe proper handling and installation procedures can cause damage.

ESD damage can range from subtle performance degradation to complete device failure. Precision integrated circuits may be more susceptible to damage because very small parametric changes could cause the device not to meet its published specifications.

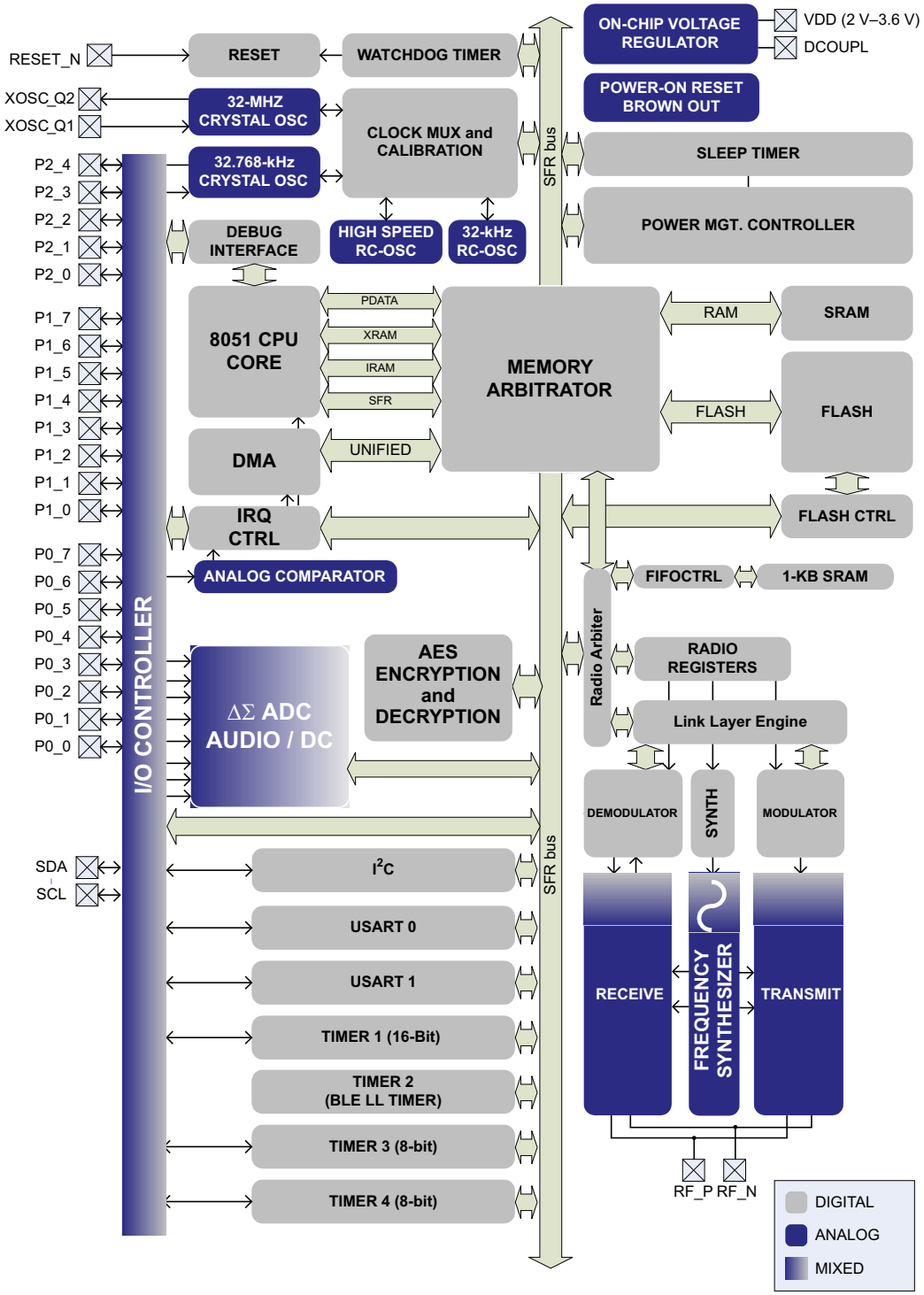


Figure 1. Block Diagram

CC2541 Proprietary Mode Radio

In proprietary mode, the CC2541 radio supports data rates up to 2 Mbps, and has extensive baseband automation, including auto-acknowledgment and address decoding. The **RF Core** controls the analog radio module and the RF transceiver state. In addition, it provides an interface between the MCU and the radio which makes it possible to issue commands, read status, and automate and sequence radio events.

It has 1 KB of dedicated RAM, which holds the 128-byte transmit and receive FIFO.

This chapter describes the proprietary mode operation of the CC2541 devices and features in the LLE program. For *Bluetooth* low energy operation, see Chapter 24.

Topic **Page**

25.1 RF Core

The **RF core** contains several submodules that support and control the analog radio modules. In addition, it provides an interface between the MCU and the radio which makes it possible to issue commands, read status, and automate and sequence radio events.

The **link-layer engine** (LLE) controls the RF transceiver state and most of the dynamically controlled analog signals such as power up and power down of analog modules. The LLE is used to provide the correct sequencing of events (such as performing an FS calibration before enabling the receiver). It handles packet assembly and decoding, including automatic length field handling, address insertion and filtering, and CRC generation and checking.

The **radio data RAM** holds a FIFO for transmit data (TX FIFO) and a FIFO for receive data (RX FIFO). Both FIFOs are 128 bytes long and have hardware control of pointers when data is entered and removed from the FIFOs. In addition, the RAM contains six segments of 128 bytes, one of which is used for communication with the LLE.

The **bit-stream processor** is used for whitening and de-whitening transferred signals and CRC generation and check.

The **modulator** transforms raw data into I/Q signals to the transmitter DAC.

The **demodulator** is responsible for retrieving the over-the-air data from the received signal.

The **frequency synthesizer (FS)** generates the carrier wave for the RF signal.

25.2 Interrupts

The radio is associated with two **interrupt** vectors on the CPU. These are the RFERR interrupt (interrupt 0) and the RF interrupt (interrupt 12) with the following functions.

- RFERR: Error situations in the radio are signaled using this interrupt.
- RF: Interrupts coming from normal operation are signaled using this interrupt.

The RF interrupt vector combines the interrupts in `RFIF`. Note that these RF interrupts are rising-edge triggered. Thus, an interrupt is generated when, for example, the TASKDONE status flag in the RFIRQF1 register goes from 0 to 1. The `RFIF` interrupt flags are described in Section 25.2.1.

25.2.1 Interrupt Registers

Two main interrupt-control SFR registers are used to enable the RF and RFERR interrupts. These are the following:

- RFERR: `IEN0.RFERRIE`
- RF: `IEN2.RFIE`

Two main interrupt-flag SFR registers hold the RF and RFERR interrupt flags. These are the following:

- RFERR: `TCON.RFERRIF`
- RF: `S1CON.RFIF`

The two interrupts generated from the RF core are a combination of several sources within the RF core. Each of the individual sources has its own enable and interrupt flags in RF core. Flags can be found in `RFIRQF0`, `RFIRQF1,` and `RFERRF`. Interrupt enable masks can be found in `RFIRQM0`, `RFIRQM1,` and `RFERRM`.

The interrupt enable bits in the mask registers are used to enable individual interrupt sources. Note that masking an interrupt source does not affect the updating of the corresponding status in the flag registers.

Due to the use of individual interrupt masks in the RF core, the interrupts coming from the RF core have two-layered masking, and care must be taken when processing these interrupts. The procedure is described as follows.

To clear an interrupt from the RF core, one must clear two flags, both the flag set in the RF core and the one set in the main interrupt flag SFR registers, S1CON or TCON (depending on which interrupt is triggered). If a flag is cleared in the RF core and there are other unmasked flags standing, the main interrupt flag is set. Exiting the interrupt service routine with the main interrupt flag set causes the interrupt service routine to be executed again.

TIP: For proper handling of interrupts in ISRs, the following is advised:

- At the start of the ISR, read and store the RF core flags
- Process the interrupts
- Clear the main interrupt flag
- Clear the processed RF core flags. It is important that this is done in a single operation.

25.3 RF Core Data Memory

The radio core has 1024 bytes of data RAM divided into eight pages of 128 bytes each. The pages are to be used as shown in Table 25-1.

Table 25-1. Radio RAM Pages

Page Number	Assignment
0	RAM-based registers
1	For RX with auto ACK: ACK payload FIFO for addresses 2 and 3
2	For RX with auto ACK: ACK payload FIFO for addresses 4 and 5
3	For RX with auto ACK: ACK payload FIFO for addresses 6 and 7
4	Free for MCU use
5	Additional RAM-based registers. Reserved for LLE
6	RX FIFO
7	TX FIFO; for RX with auto ACK: ACK payload FIFO for addresses 0 and 1

The active memory page is selected in register RFRAMCFG.PRE. The selected page is accessible at XDATA addresses 0x6000–0x607F. The RX FIFO page (page 6) is also accessible at XDATA addresses 0x6080–0x60FF. The TX FIFO page (page 7) is also accessible at XDATA addresses 0x6100–0x617F.

A page is used for transferring parameters to the LLE; see Section 25.3.3.

There is no hardware protection to prevent the MCU from overwriting memory used by the LLE and the FIFO. Thus, the MCU should never write to page 5 (except for special dedicated registers). The MCU should write to pages 0, 1, 2, 3, and 7 only as specified in this chapter. Writes to the FIFO pages should only be done in ways compatible with FIFO operation, except for accessing the TX FIFO page while running an RX task with auto ACK.

Pages 0, 1, 6, and 7 have retention in all power modes, whereas the contents of pages 2–5 are lost in PM2 and PM3.

Radio core hardware registers are located at XDATA addresses 0x6180–0x61F7. Figure 25-1 shows the mapping of radio memory to MCU XDATA memory space.

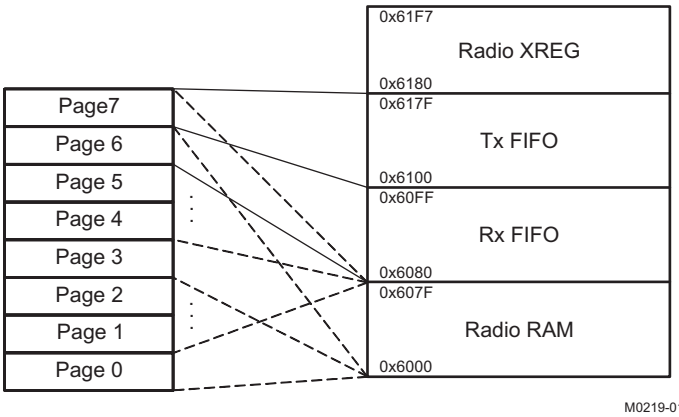


Figure 25-1. Mapping of Radio Memory to MCU XDATA Memory Space

25.3.1 FIFOs

The FIFOs are used for transporting data between the MCU and the radio. The FIFOs have hardware support for read and write pointer increment with circular buffering, overflow and underflow detection, and flushing of last entry or the entire FIFO.

The RX and TX FIFOs are fundamentally two similar modules. Each FIFO has four pointers: the write pointer (WP), the read pointer (RP), the start-of-packet write pointer (SWP), and the start-of-packet read pointer (SRP). WP and RP give the index in the FIFO where the next byte is to be written and read, respectively. SWP is used to indicate the start of the current packet being written, and SRP is used to indicate the start of the current packet being read. The use of the pointers is indicated in Figure 25-2.

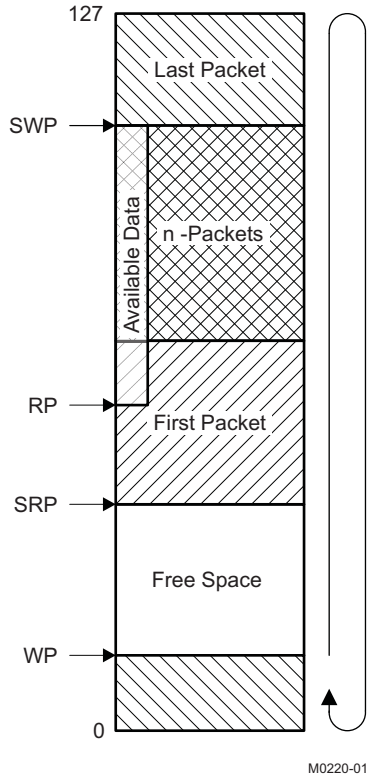


Figure 25-2. FIFO Pointers

The TX FIFO and RX FIFO may be accessed though the SFR register RFD (0xD9). Data is written to the TX FIFO when writing to the RFD register. Data is read from the RX FIFO when the RFD register is read. In addition, there are separate read and write registers for each FIFO (RFRXFRD, RFRXFWR, RFTXFRD, RFTXFWR).

The RX FIFO or TX FIFO can be cleared by issuing CMD_RXFIFO_RESET or CMD_TXFIFO_RESET (see Section 25.3.1.2), respectively. The contents of both FIFOs can be cleared by issuing CMD_FIFO_RESET.

Four operations are defined to handle the four pointers:

- *Deallocate* is setting SRP equal to RP. This should be done when the treatment of a packet that has been read from the FIFO is finished.
- *Retry* is setting RP equal to SRP. This is done to re-read a packet that has been read from the FIFO previously.
- *Discard* is setting WP equal to SWP. This is done to remove a packet that had been written to the FIFO.
- *Commit* is setting SWP equal to WP. This is done to confirm the writing of a packet to the FIFO and making it available to be read out.

Using the register RFFCFG, it is possible to set up auto-commit and auto-deallocate for each of the FIFOs. If auto-commit is enabled, SWP is set equal to WP each time a byte is written to the FIFO. If auto-deallocate is enabled, SRP is set equal to RP each time a byte is read from the FIFO. By default, auto-commit is enabled for the TX FIFO and auto-deallocate is enabled for the RX FIFO. This is also the recommended setting. However, if packets that exceed the FIFO size are to be supported, auto-commit must be enabled for the RX FIFO and auto-deallocate for the TX FIFO; see Section 25.8.1 and Section 25.8.2 for details. If auto-commit is disabled for the TX FIFO, the MCU must issue a commit command after writing a packet to the TX FIFO, and if auto-deallocate is disabled for the RX FIFO, the MCU must issue a deallocate command after reading a packet from the RX FIFO.

25.3.1.1 FIFO Status and Interrupts

The XREG registers RFRXFLEN and RFTXFLEN provide information on the amount of data in the FIFOs. This is the number of bytes between SRP and WP, that is, the number of bytes that is not free space in Figure 25-2. The register RFFSTATUS contains status bits for each of the FIFOs. FIFO empty is defined as the length being 0, and FIFO full is defined as the length being 128. The amount of data between RP and SWP is known as available data, and there is a status bit in the RFFSTATUS register telling whether there is available data for each of the FIFOs.

An attempt to write to a full FIFO results in a FIFO overflow. The data written is then ignored and the RX0VERF or TX0VERF flag is set in the RFERRF register, causing an RFERR interrupt. An attempt to read from a FIFO when no data is available results in a FIFO underflow. The value read is then zero, and the RXUNDERF or TXUNDERF flag is set in the RFERRF register, causing an RFERR interrupt.

Registers RFTXFTHRS and RFRXFTHRS are used to set threshold points for the TX and RX FIFOs, respectively. Each FIFO has one status flag and two interrupt flags; when the amount of data in the FIFO crosses the threshold, an interrupt flag is set. The FIFO status flags are available in RFFSTATUS, and the interrupt flags are available in RFIRQF0.

When the amount of data in the FIFO is above the threshold, that is, RFxXFLEN is greater than or equal to RFxXFTHRS, the status bit xXDTHEX of RFFSTATUS is 1, otherwise it is 0.

When data is written to the FIFO causing the FIFO threshold to be crossed, that is, xXDTHEX going from 0 to 1, the corresponding interrupt flag is set.

When data is read from the FIFO causing the FIFO threshold to be crossed, that is, xXDTHEX going from 1 to 0, the corresponding interrupt flag is set.

25.3.1.2 Command Register

The command register RFST can be used for sending commands to the FIFO. Commands in the range 0x80–0xFF are commands to the FIFO. Other commands are commands to the LLE; see Section 25.9.1.

The supported FIFO commands are listed in Table 25-2. A command in the range of 0x80–0xFF that does not match any of the listed commands is ignored.

Table 25-2. Commands to FIFO via RFST Register

Number	Command Name	Description
0x81	CMD_RXFIFO_RESET	Reset (empty) RX FIFO. Set RFRXF* := 0
0x82	CMD_RXFIFO_DEALLOC	Deallocate RX FIFO. This sets RFRXFSRP := RFRXFRP.
0x83	CMD_RXFIFO_RETRY	Retry RX FIFO. This sets RFRXFRP := RFRXFSRP
0x84	CMD_RXFIFO_DISCARD	Discard RX FIFO. This sets RFRXFWP := RFRXFSWP
0x85	CMD_RXFIFO_COMMIT	Commit RX FIFO. This sets RFRXFSWP := RFRXFWP
0x91	CMD_TXFIFO_RESET	Reset (empty) TX FIFO. Set RFRXF* := 0
0x92	CMD_TXFIFO_DEALLOC	Deallocate TX FIFO. This sets RFTXFSRP := RFTXFRP.
0x93	CMD_TXFIFO_RETRY	Retry TX FIFO. This sets RFTXFRP := RFTXFSRP
0x94	CMD_TXFIFO_DISCARD	Discard TX FIFO. This sets RFTXFWP := RFTXFSWP
0x95	CMD_TXFIFO_COMMIT	Commit TX FIFO. This sets RFTXFSWP := RFTXFWP
0xF1	CMD_FIFO_RESET	Reset both FIFOs
0xF2	CMD_FIFO_DEALLOC	Deallocate both FIFOs
0xF3	CMD_FIFO_RETRY	Retry both FIFOs
0xF4	CMD_FIFO_DISCARD	Discard both FIFOs
0xF5	CMD_FIFO_COMMIT	Commit both FIFOs

25.3.1.3 FIFO Pointer Operations

The FIFO pointers can be accessed directly through registers RFFRXFWP, RFFRXFRP, RFFRXFSWP, RFFRXFSRP, RFFTXFWP, RFFTXFRP, RFFTXFSWP, and RFFTXFSRP.

Because the placement of the pointers may be the same for an empty and a full FIFO, there are internal states distinguishing between these situations. This means that although any value can be written to the pointer registers, certain rules must be observed for the FIFO to function reliably after the pointer write.

Any writes to a pointer must be considered to move that pointer up. Hence, writing N to a pointer already holding N is considered equivalent to moving that pointer up 128 places, writing N-1 is equivalent to moving the pointer up 127 places, and so on.

The pointers must maintain a specific ordering: (Going from lowest position to highest) SRP, RP, SWP, WP.

A lower pointer may be moved up to but not past a higher pointer, whereas the highest pointer (WP) may be moved down to, but not past the lower.

25.3.1.4 Cooperation With LLE

The LLE performs FIFO operations as part of its operation. In order to avoid conflicts between the LLE and the MCU, access to FIFO registers should be done according to Table 25-3. Read accesses can always be made, except for the data-read registers, which causes the read pointers to be modified. If the MCU reads a register, one must take into account that the value may change at any time due to accesses from the LLE. The reset FIFO commands should only be run by the MCU between LLE tasks. They are marked with an asterisk in Table 25-3.

Table 25-3. Access to FIFO Registers

Register	Read Access	Write Access
RFD	MCU	MCU

Table 25-3. Access to FIFO Registers (continued)

Register	Read Access	Write Access
RFST (FIFO commands)	N/A	Depends on command: 0x81 : Reset RX FIFO: MCU* 0x82 : Deallocate RX FIFO: MCU 0x83 : Retry RX FIFO: MCU 0x84: Discard RX FIFO: LLE 0x85: Commit RX FIFO: LLE 0x91 : Reset TX FIFO: MCU* 0x92 : Deallocate TX FIFO: LLE(1) 0x93 : Retry TX FIFO: LLE(1) 0x94: Discard TX FIFO: MCU 0x95: Commit TX FIFO: MCU 0xF1: Reset both FIFOs: MCU* 0xF2: Deallocate both FIFOs: none 0xF3: Retry both FIFOs: none 0xF4: Discard both FIFOs: none 0xF5: Commit both FIFOs: none
RFFDMA0	Both	MCU
RFFDMA1	Both	MCU
RFFSTATUS	Both	N/A
RFFCFG	Both	MCU
RFRXFLEN	Both	N/A
RFRXFTHRS	Both	MCU
RFRXFWR	N/A	LLE
RFRXFRD	MCU	N/A
RFRXFWP	Both	LLE
RFRXFRP	Both	MCU
RFRXFSWP	Both	LLE
RFRXFSRP	Both	MCU
RFTXFLEN	Both	N/A
RFTXFTHRS	Both	MCU
RFTXFWR	N/A	MCU
RFTXFRD	LLE	N/A
RFTXFWP	Both	MCU
RFTXFRP	Both	LLE
RFTXFSWP	Both	MCU
RFTXFSRP	Both	LLE

(1) MCU if PRF_ADDR_ENTRY*n*.CONF.RETRY is 1

25.3.2 DMA

It is possible to use direct memory access (DMA) to move data between memory and the radio. See Chapter 8 for a detailed description on how to set up and use DMA transfers.

There are two DMA triggers associated with the radio: the RADIO DMA triggers 0 and 1 (DMA triggers 19 and 11).

The radio DMA trigger source is selected in registers RFFDMA0 and RFFDMA1. See the register descriptions in Section 25.12 for details.

25.3.3 RAM-Based Registers

A list of the memory entries of the general radio RAM area used for parameter transfer is shown in Table 25-4. All these registers are in page 0 of the radio RAM. Each memory entry is considered a RAM-based register and has a name. Numeric values that are two bytes long are represented in little-endian format.

The radio RAM registers have no defined reset value and must therefore be initialized by the MCU.

The registers SEMAPHORE0 and SEMAPHORE1 can be used to verify data integrity. These registers are changed to 0 when they are read. If a semaphore register is read and the value was 1, the semaphore has been successfully taken, and subsequent reads of the register return 0 until the semaphore is released. If a semaphore register is read as 0, the semaphore was not free. A semaphore can be released by writing 1 to the semaphore register; this should only be done if the semaphore has previously been taken by the MCU. The LLE takes SEMAPHORE0 when a task starts and SEMAPHORE1 when the radio has been set up. Both semaphores are released by the LLE at the end of the task. If the LLE is not granted the semaphore, it generates an error. If SEMAPHORE0 and SEMAPHORE1 are taken by the MCU before registers protected by these semaphores are modified by the MCU, data integrity is ensured, and an error occurs if the LLE is accidentally started while such an access is going on. SEMAPHORE2 is not used by the LLE.

Where bit numbering is used, bit 0 is the LSB and bit 7 is the MSB. Multi-byte fields are little-endian.

The detailed breakdown of the address entries PRF_ADDR_ENTRY0–PRF_ADDR_ENTRY7 is shown in Table 25-5 or Table 25-6, depending on the operational mode.

The **Prot** columns of Table 25-4, Table 25-5, and Table 25-6 list the type of protection for each entry:

Sem0: Entries protected by SEMAPHORE0. Should only be written by the MCU while the LLE does not have SEMAPHORE0. Is not modified by the LLE.

Sem1: Entries protected by SEMAPHORE1. Should only be written by the MCU while the LLE does not have SEMAPHORE1. Is not modified by the LLE.

Sem1/R: Entries containing state variables and accumulative counters that are updated by the LLE. They may be read by the MCU after a receive or transmit interrupt to see how many packets have been received or transmitted. The MCU must take into account that at the time these values are read, some of them may have been updated for the next interrupt and some not. When the LLE does not have SEMAPHORE1, the MCU may write to them to initialize. The counters are not initialized by the LLE.

None: No semaphore protection; special rules apply for access.

Table 25-4. RAM-Based Registers

Name	Addr	Prot	Description
PRF_CHAN	0x6000	Sem0	**Bits 0–6**: FREQ Frequency to use. 0: 2379 MHz ... 1-MHz steps 116: 2495 MHz 117–126: Reserved 127: The LLE does not program frequency; it is to be set up by the MCU through the FREQCTRL and MDMTEST1 registers. **Bit 7**: SYNTH_ON 0: Turn off synthesizer when task is done. 1: Leave synthesizer running after task is done.

Table 25-4. RAM-Based Registers (continued)

Name	Addr	Prot	Description
PRF_TASK_CONF	0x6001	Sem0	Configuration of task control **Bits 0–1**: MODE (operation mode) 00: Basic mode, fixed length 01: Basic mode, variable length 10: Auto mode, 9-bit header 11: Auto mode, 10-bit header **Bit 2**: REPEAT (repeated operation) 0: Single operation 1: Repeated operation **Bit 3**: START_CONF (start configuration) 0: Start each receive or transmit immediately 1: Start each receive or transmit on Timer 2 event 1 **Bits 4–5**: STOP_CONF (stop configuration) 00: No stop based on Timer 2. 01: End task after current packet is done on Timer 2 event 2 (end immediately in sync search or wait) 10: Stop transmit or receive immediately on Timer 2 event 2 11: End task on Timer 2 event 2 in first sync search or clear-channel assessment. No stop after first sync search or clear-channel assessment. **Bit 6**: TX_ON_CC_CONF 0: Listen until RSSI drops below given level, then start TX. 1: End task if RSSI is above given level **Bit 7**: REPEAT_CONF For TX_ON_CC with REPEAT = 1: 0: Listen again on repeated operation and retransmissions 1: Listen only before the first transmission, then transmit every time For RX with REPEAT = 1: 0: Recalibrate the synthesizer before listening for new packets 1: Recalibrate the synthesizer only when the task starts

Table 25-4. RAM-Based Registers (continued)

Name	Addr	Prot	Description
PRF_FIFO_CONF	0x6002	Sem1	Configure FIFO use **Bit 0**: AUTOFLUSH_IGN Keep received packets with unexpected sequence number in the RX FIFO. 0: Keep 1: Auto-flush **Bit 1**: AUTOFLUSH_CRC Keep received packets with CRC error in the RX FIFO. 0: Keep 1: Auto-flush **Bit 2**: AUTOFLUSH_EMPTY Keep packets with no payload in the RX FIFO. 0: Keep 1: Auto-flush **Bit 3**: RX_STATUS_CONF RX FIFO channel information 0: Do not append RSSI and RES 1: Append RSSI and RES **Bits 4–5**: RX_ADDR_CONF RX FIFO address and config byte configuration 00: Do not include address or config byte in RX FIFO 01: Include received address in RX FIFO (1-byte addresses only), but no config byte 10: Include config byte in RX FIFO, but no address byte 11: Include received address (1-byte addresses only) and config byte in RX FIFO **Bits 6–7**: TX_ADDR_CONF TX FIFO address and config byte configuration 00: No address or config byte; read address from PRF_ADDR_ENTRY0 01: Include address byte in TX FIFO, no config byte 10: Include config byte and use address index in that byte to find address from PRF_ADDR_ENTRYn 11: Read address from TX FIFO followed by config byte (where address information is ignored). Not allowed for PRF_TASK_CONF.MODE = 00 or 01.
PRF_PKT_CONF	0x6003	Sem0	Packet configuration **Bit 0**: ADDR_LEN. Number of address bytes (0 or 1). **Bit 1**: AGC_EN 0: Do not use AGC 1: Use AGC (Section 25.9.2.1) **Bit 2**: START_TONE 0: Ordinary transmission 1: Override extra preamble bytes with tone and reduce synthesizer calibration time accordingly (Section 25.9.2.2) **Bits 3-7**: Reserved, always write 0.
PRF_CRC_LEN	0x6004	Sem1	Number of CRC bytes. Permitted values: 0–4
PRF_RSSI_LIMIT	0x6005	Sem1	For transmit on clear channel. Start a transmit task by listening to the channel; start transmitting if the RSSI drops below the level (signed) given in this register.
PRF_RSSI_COUNT	0x6006–0x6007	Sem1	For transmit on clear channel. Number of additional RSSI measurements that must be below the RSSI limit before transmission takes place.

Table 25-4. RAM-Based Registers (continued)

Name	Addr	Prot	Description
PRF_CRC_INIT	0x6008–0x600B	Sem1	Initialization value for CRC. For less than a 4-byte CRC, the first bytes shall be 0 and the last bytes the desired value.
PRF_W_INIT	0x600C	Sem1	Byte to write to register BSP_W before a packet; initializes the PN7 whitener if that is used. If PN9 whitener is used, bit 7 should be 1.
PRF_RETRANS_CNT	0x600D	Sem1	Maximum number of retransmissions in automatic retransmit
PRF_TX_DELAY	0x600E–0x600F	Sem1	Time from end of transmission to new transmission of different payload, given in units of 62.5 ns
PRF_RETRANS_DELAY	0x6010–0x6011	Sem1	Time from end of transmission to retransmission in auto retransmit mode, given in units of 62.5 ns
PRF_SEARCH_TIME	0x6012–0x6013	Sem1	Time to perform search before giving up or retransmitting, given in 31.25-ns units. 0: Never give up. Must be at least 256 if not 0.
PRF_RX_TX_TIME	0x6014–0x6015	Sem1	Time to add to RX-TX turnaround time in RX with auto ACK, given in 31.25-ns units
PRF_TX_RX_TIME	0x6016–0x6017	Sem1	Time to add to TX-RX turnaround time in TX with auto retransmission, given in 31.25-ns units
PRF_ADDR_ENTRY0	0x6018–0x6023		Address structure for address number 0. See Table 25-5 and Table 25-6 for details.
PRF_ADDR_ENTRY1	0x6024–0x602F		Address structure for address number 1. See Table 25-5 and Table 25-6 for details.
PRF_ADDR_ENTRY2	0x6030–0x603B		Address structure for address number 2. See Table 25-5 and Table 25-6 for details.
PRF_ADDR_ENTRY3	0x603C–0x6047		Address structure for address number 3. See Table 25-5 and Table 25-6 for details.
PRF_ADDR_ENTRY4	0x6048–0x6053		Address structure for address number 4. See Table 25-5 and Table 25-6 for details.
PRF_ADDR_ENTRY5	0x6054–0x605F		Address structure for address number 5. See Table 25-5 and Table 25-6 for details.
PRF_ADDR_ENTRY6	0x6060–0x606B		Address structure for address number 6. See Table 25-5 and Table 25-6 for details.
PRF_ADDR_ENTRY7	0x606C–0x6077		Address structure for address number 7. See Table 25-5 and Table 25-6 for details.
PRF_N_TX	0x6078	Sem1/R	Total number of packets transmitted
PRF_LAST_RSSI	0x6079	Sem1/R	RSSI of last received packet
PRF_LAST_DCOFF	0x607A–0x607D	Sem1/R	DC offset of last received packet

Table 25-4. RAM-Based Registers (continued)

Name	Addr	Prot	Description
PRF_RADIO_CONF	0x607E	Sem0	Configure radio hardware **Bits 0–1**: RXCAP 00: Do not capture on RX packets 01: Capture start of every RX packet 10: Capture end of every RX packet 11: Capture start of first RX packet only **Bits 2–3**: TXCAP 00: Do not capture on TX packets 01: Capture start of every TX packet 10: Capture end of every TX packet 11: Capture start of first TX packet only **Bits 4–5**: TXIF: TX IF configuration (for 2 Mbps only) 00: Zero IF 01: ±1 MHz IF 10: ±2 MHz IF 11: ±3 MHz IF **Bit 6**: DCOFF: Special dc offset handling 0: Standard dc offset 1: Use special dc offset routine measuring dc offset right after RX start **Bit 7**: DCWB: Write back dc offset estimate to override registers 0: Do not write back 1: Write back after each received packet with CRC OK
PRF_ENDCAUSE	0x607F	None	Reason why LLE ended task

Table 25-5. Address Structure for Auto Mode

Name	Index	Prot	Description
CONF	0x00	Sem1	**Bit 0**: ENA0 (Enable for primary sync word – RX task only) 0: Disable address entry for primary sync word 1: Enable address entry for primary sync word **Bit 1**: ENA1 (Enable for secondary sync word – RX task only) 0: Disable address entry for secondary sync word 1: Enable address entry for secondary sync word **Bit 2**: REUSE (Allow reuse of transmitted packet) 0: LLE deallocates packet after it has been acknowledged 1: LLE does not deallocate packet after it has been acknowledged (this is up to the MCU) **Bit 3**: AA (Enable auto acknowledgement or auto retranmsmission) 0: Disable auto ack (RX) or auto retransmission (TX) for this address 1: Enable auto ack (RX) or auto retransmission (TX) for this address **Bit 4**: VARLEN (variable length support) 0: Use fixed length given by RXLENGTH in receiver when receiving packets or ACKs 1: Use variable length up to RXLENGTH in receiver when receiving packets or ACKs **Bit 5**: FIXEDSEQ (fixed sequence number – TX task only) 0: Insert sequence number from SEQSTAT.SEQ 1: Read sequence number from TX FIFO **Bit 6**: TXLEN 0: Insert packet length in header when transmitting 1: Used fixed-length word when transmitting Note: Must not be set to 1 unless the peer uses fixed length
RXLENGTH	0x01	Sem1	Maximum length of received packet (0–127)
ADDRESS	0x02	Sem1	Address of packet

Table 25-5. Address Structure for Auto Mode (continued)

Name	Index	Prot	Description
SEQSTAT	0x03	Sem1/R	**Bit 0**: VALID (RX task only) 0: The status is not valid. Any packet is viewed as new. On successful reception of a packet, the LLE sets this bit. 1: The status is valid. Only packets with a sequence number and CRC different from the previous one are accepted. **Bits 1–2**: SEQ (sequence number). For RX, the sequence number of the last successfully received packet. For TX, the sequence number of the next or current packet to be transmitted **Bits 3–4**: ACKSEQ (ACK sequence number – RX task only) For RX with auto ACK, the sequence number of the next or current ACK to be transmitted **Bit 5**: ACK_PAYLOAD_SENT (RX task only) 0: The last received packet was not acknowledged with payload. 1: The last received packet was acknowledged with payload. **Bit 6**: NEXTACK (next ACK buffer to use – RX task only) 0: Use ACK buffer 0. 1: Use ACK buffer 1.
ACKLENGTH0	0x04	None	For RX with auto ACK: Length of payload to be transmitted from buffer 0. When 0, the buffer is free. After the payload has been transmitted and a packet with a new sequence number is received, the value is set to 0 by the LLE. The MCU only writes to the register when it is zero; the LLE only writes it to zero when it is non-zero.
ACKLENGTH1	0x05	None	For RX with auto ACK: Length of payload to be transmitted from buffer 1. When 0, the buffer is free. After the payload has been transmitted and a packet with a new sequence number is received, the value is set to 0 by the LLE. The MCU only writes to the register when it is zero; the LLE only writes it to zero when it is non-zero.
CRCVAL	0x06–0x07	Sem1/R	CRC value (last two bytes if more than 2 CRC bytes) of last successfully received packet
N_TXDONE	0x08	Sem1/R	Number of packets transmitted. For auto retransmission, only acknowledged packets with new sequence number are counted. For auto ACK, only packets with new payload are counted when the payload has been confirmed.
N_RXIGNORED	0x09	Sem1/R	Number of retransmitted packets received with CRC OK
N_RXOK	0x0A	Sem1/R	Number of new packets received with CRC OK or ACK packets without payload received
N_RXNOK	0x0B	Sem1/R	Number of packets received with CRC error

Table 25-6. Address Structure for Basic Mode

Name	Index	Prot	Description
CONF	0x00	Sem1	**Bit 0**: ENA0 (enable for primary sync word – RX task only) 0: Disable address entry for primary sync word 1: Enable address entry for primary sync word **Bit 1**: ENA1 (enable for secondary sync word – RX task only) 0: Disable address entry for secondary sync word 1: Enable address entry for secondary sync word **Bit 2**: REUSE (allow reuse of transmitted packet) 0: LLE deallocates packet after it has been transmitted 1: LLE does not deallocate packet after it has been transmitted (this is up to the MCU)
RXLENGTH	0x01	Sem1	Maximum length of received packet (0–255)
ADDRESS	0x02	Sem1	Address of packet
	0x03–0x09		Reserved
N_RXOK	0x0A	Sem1/R	Number of packets received with CRC OK
N_RXNOK	0x0B	Sem1/R	Number of packets received with CRC error

25.3.4 Variables in RAM Page 5

Some additional RAM registers are placed in page 5 of the RFCORE RAM. These variables have the prefix PRFX and are listed in Table 25-7. The addresses overlap other RAM registers, and to access them page 5 must be selected using the RFRAMCFG register; see Section 25.3. Some of the registers have a reset value. This value is written by the LLE shortly after it has been taken out of reset by LLECTRL.LLE_EN being set to 1. If the MCU must modify these registers, the modification must be done each time the LLE is reset. After taking the LLE out of reset, the MCU may modify the registers after LLASTAT.LLE_IDLE has gone high.

Table 25-7. RAM-Based Registers in RAM Page 5(1)

Name	Addr	Prot	Reset Val	Description
PRFX_LAST_FREQEST	0x6006	Sem1/R	–	Last frequency offset estimate, read from the FREQEST register at the end of receiving each packet
PRFX_RSSI_LIM_LOWER	0x6008	Sem1	0x20	Lower RSSI limit for use in AGC algorithm
PRFX_RSSI_LIM_UPPER	0x6009	Sem1	0x3C	Upper RSSI limit for use in AGC algorithm
PRFX_RSSI_DIFF	0x600A	Sem1	0x14	Difference between high and low RSSI gain
PRFX_LNAGAIN_SAT	0x600B	Sem1	0x4A	LNAGAIN setting to use while close to saturation
PRFX_TONE_DURATION	0x600C–0x600D	Sem1	0x064A	Duration of tone in start of packet if PRF_PKT_CONF.START_TONE = 1, given in 31.25-ns units
PRFX_TONE_OFFSET	0x600E–0x600F	Sem0	0x0600	Time to subtract from TX synthesizer calibration time if PRF_PKT_CONF.START_TONE = 1, given in 31.25-ns units

(1) Note that the LLE is reset when the device enters PM2 or PM3. This means that the PRFX registers must be re-initialized after coming up from one of these power modes.

The parts of RAM page 5 that are not listed in Table 25-7 are reserved for use by the LLE and should not be written by the MCU.

25.4 Bit-Stream Processor

The **bit-stream processor** (BSP) supports automatic insertion of CRC and detection of CRC error with a programmable polynomial of 8, 16, 24, or 32 bits.

The **bit-stream processor** also supports whitening and de-whitening. The whitening sequences supported are a PN7 sequence and a PN 9 sequence compatible with CC2500 and CC2510.

The bit-stream processor is used by the LLE to do the whitening and CRC generation and checking. This operation is based on the configuration set up by the MCU. The BSP can also be run in a coprocessor mode to calculate whitened sequences and CRCs. This must only be done while the LLE is not running.

25.4.1 Whitening

The BSP supports two whiteners, a PN7 and a PN9 whitener. The register BSP_MODE is used to enable or disable each whitener. When no whitener is enabled, it outputs zero. The whitener sequence is XORed with the transmitted or received signal.

It is possible to enable both whiteners. This is useful, for example, in conjunction with the test command CMD_TX_TEST (#IMPLIED) to transmit a white test signal.

25.4.1.1 PN7 Whitening

The PN7 whitener is shown in Figure 25-3. It has a 7-bit whitening shift register **w** used for calculating the PN sequence given by the polynomial $x^7 + x^4 + 1$. The output is the same as the shift register feedback.

The **w** register must be initialized by writing **w** into register BSP_W.W before starting receiving or transmitting a packet. Doing this sets w_6 to BSP_W[0], w_5 to BSP_W[1] and so on up to w_1 to BSP_W[5]; w_0 is set to 1.

When running normal receive or transmit tasks, writing to BSP_W is done by the LLE, which writes the value in PRF_W_INIT to this register, but for test commands and co-processor mode, the BSP_W register must be written by the MCU.

The PN7 whitener is enabled by the bit W_PN7_EN of the BSP_MODE register.

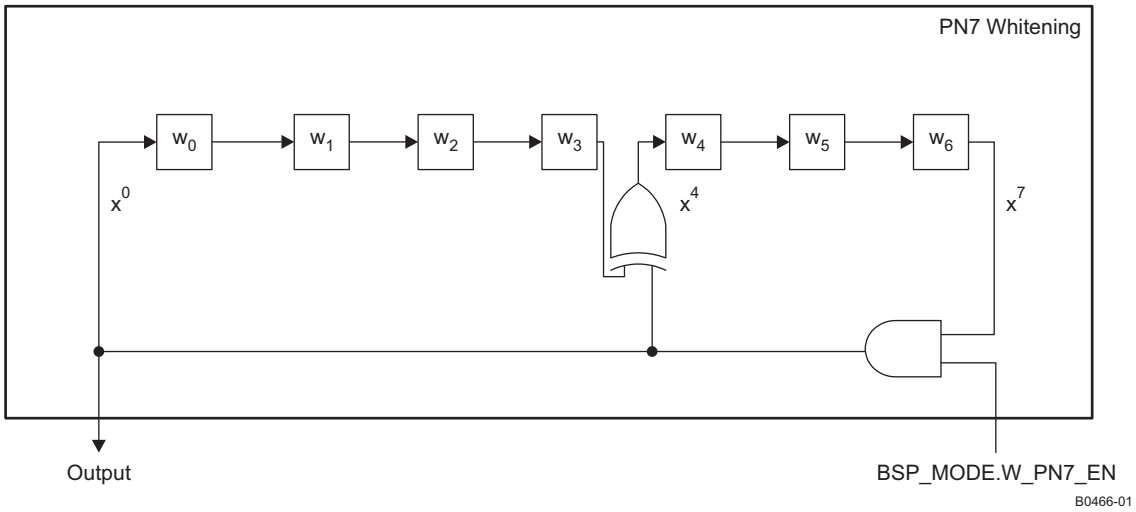


Figure 25-3. PN7 Whitening

25.4.2 CC2500-Compatible PN9 Whitening

The CC2500-compatible PN9 whitener is shown in Figure 25-4. It has a 9-bit whitening shift register s and an 8-bit output register b. It produces a whitening sequence compatible with CC2500, CC2510 and other TI devices. These devices use the polynomial $x^9 + x^4 + 1$. The whitening sequence is produced one byte at a time, and the byte is bit-reversed before being XORed with a received or transmitted byte. Before starting reception or transmission of a packet, the s and b registers must be initialized to all ones by writing a 1 to register BSP_W.W_PN9_RESET. As for the PN7 whitener, this is done by the LLE for normal receive and transmit tasks, provided that bit 7 of PRF_W_INIT is 1.

In Figure 25-4, the dashed arrows going from the s blocks to the b blocks denote a copy that takes place after the whitening of one byte is done. This means that the first byte is whitened by the 8 bits that are in the b register after initialization (all ones). As this byte is being whitened, the s register is updated. After the first byte is whitened, the value of the s register is copied into the b register and used for whitening the second byte.

The CC2500-compatible whitener is enabled by bit W_PN9_EN of the BSP_MODE register.

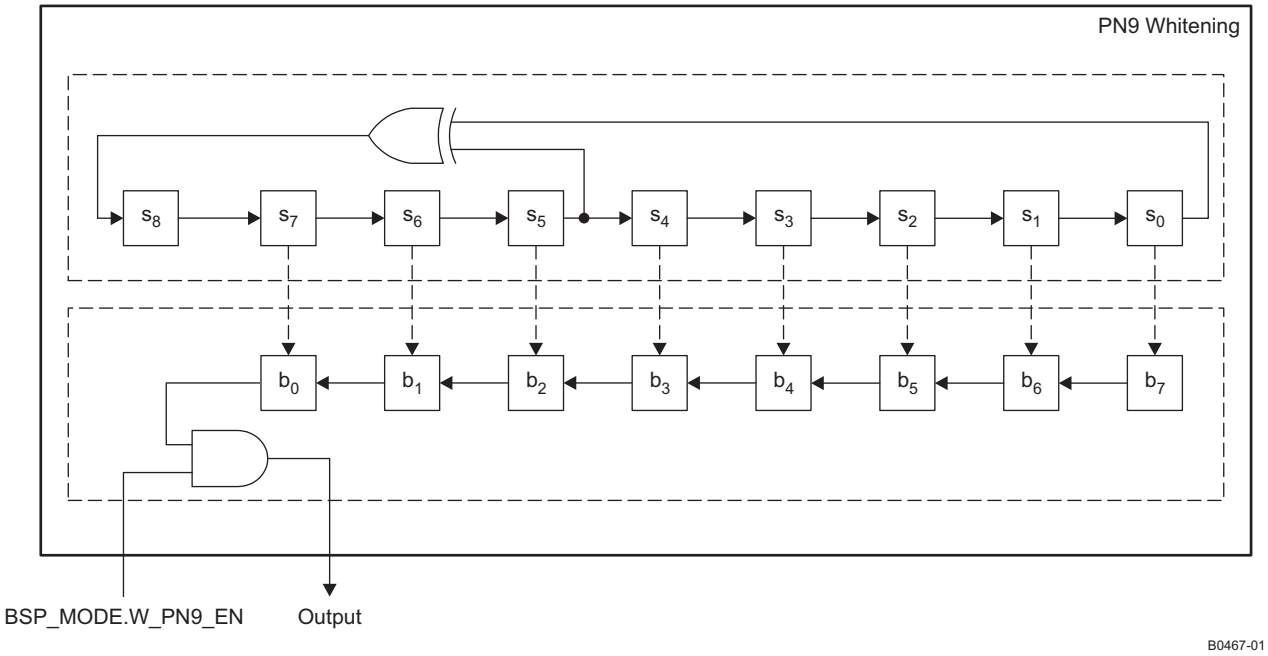


Figure 25-4. CC2500-Compatible Whitening

25.4.3 CRC

A block diagram showing the operation of the CRC module is given in Figure 25-5. The CRC sub-module has two registers:

- A 32-bit data shift register **d**
- A 32-bit register **p** for holding the polynomial

The **p** register defines the shift register used for calculating CRC. There is a feedback tap in the locations where the corresponding bit of **p** is set to 1. The module input is XORed by the output of the shift register, and this becomes the feedback of the shift register.

The current value of the data shift register **d** is the CRC value. Prior to the start of CRC calculation, the **d** and **p** registers should be initialized by writing **d** to registers `BSP_D[0–3]` and **p** to registers `BSP_P[0–3]`. The `BSP_P[0–3]` registers only must be set once, whereas the `BSP_D[0–3]` registers should be set again for each packet. In normal transmit and receive modes, this is handled by the LLE, which writes the value of `PRF_CRC_INIT[0–3]` to `BSP_D[0–3]`. At the end of CRC calculation, the value of the register is serially shifted out on the output. When performing CRC checking, all the `BSP_D[0–3]` registers should be 0 for the CRC to be OK after the received CRC has been fed through the shift register.

If whitening is enabled, calculated CRC bytes are whitened before transmission, and received CRC bytes are de-whitened before CRC checking.

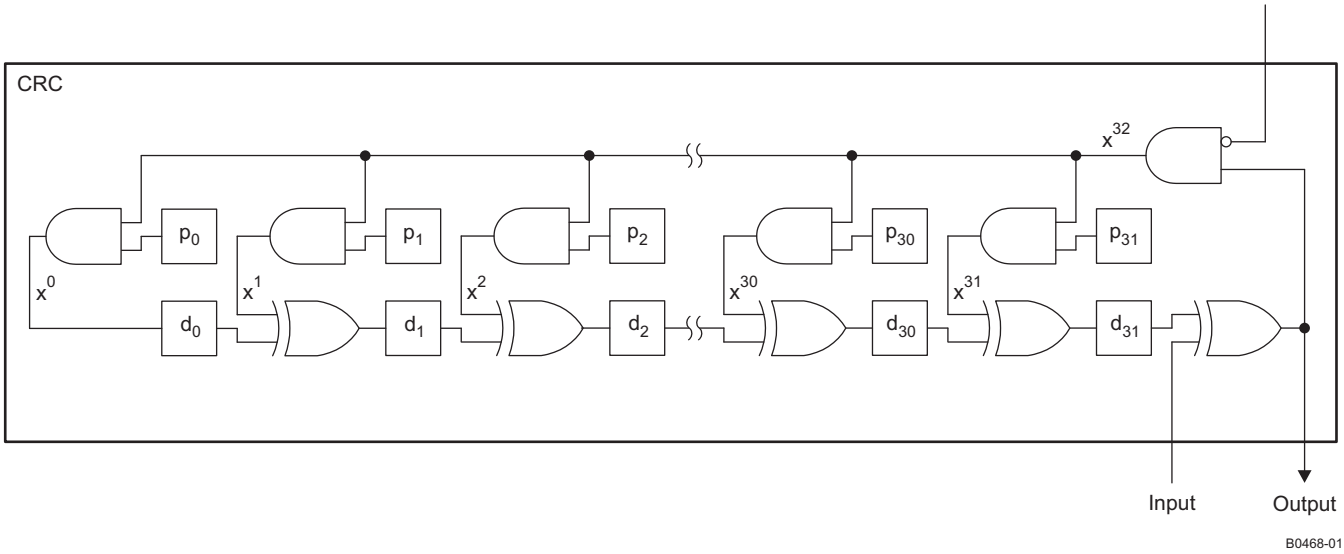


Figure 25-5. CRC Module

A 32-bit CRC polynomial can be described by the equation $x^{32} + a_{31}x^{31} + \ldots + a_1x^1 + 1$, where all a_n are 0 or 1. To represent this, each P[n] bit in the BSP_P0–BSP_P3 registers should be set to a_n , and P[0] should be set to 1. To reduce the size of the polynomial to k, set the bits P[33 – k:0] to 0 and P[32 – k] to 1. In this case, the initialization value must have zeros at D[33 – k:0]. In practice, only polynomials of order 8, 16, 24, and 32 are supported, as the number of CRC bits produced in the transmitter and checked in the receiver is always a multiple of 8. The number of CRC bytes produced in normal transmit tasks is given by RAM register PRF_CRC_LEN.

This is summarized in Table 25-8 for the four CRC polynomial orders supported. In the BSP_Px column, the numbers are binary, with the most significant bit at the left. In the PRF_CRC_INIT column, an X indicates the initialization value to use (each X does not have to be the same). Some examples are shown in Table 25-9.

Table 25-8. Register Settings for Different CRCs

Order	PRF_CRC_LEN	Polynomial	BSP_Px	PRF_CRC_INIT
8	1	$x^8 + a_7x^7 + \ldots + a_1x^1 + 1$	BSP_P0 = 0000 0000 BSP_P1 = 0000 0000 BSP_P2 = 0000 0000 BSP_P3 = $a_7\ a_6\ a_5\ a_4\ a_3\ a_2\ a_1 1$	PRF_CRC_INIT[0] = 0 PRF_CRC_INIT[1] = 0 PRF_CRC_INIT[2] = 0 PRF_CRC_INIT[3] = X
16	2	$x^{16} + a_{15}x^{15} + \ldots + a_1x^1 + 1$	BSP_P0 = 0000 0000 BSP_P1 = 0000 0000 BSP_P2 = $a_7\ a_6\ a_5\ a_4\ a_3\ a_2\ a_1 1$ BSP_P3 = $a_{15}\ a_{14}\ a_{13}\ a_{12}\ a_{11}\ a_{10}\ a_9\ a_8$	PRF_CRC_INIT[0] = 0 PRF_CRC_INIT[1] = 0 PRF_CRC_INIT[2] = X PRF_CRC_INIT[3] = X
24	3	$x^{24} + a_{23}x^{23} + \ldots + a_1x^1 + 1$	BSP_P0 = 0000 0000 BSP_P1 = $a_7\ a_6\ a_5\ a_4\ a_3\ a_2\ a_1 1$ BSP_P2 = $a_{15}\ a_{14}\ a_{13}\ a_{12}\ a_{11}\ a_{10}\ a_9\ a_8$ BSP_P3 = $a_{23}\ a_{22}\ a_{21}\ a_{20}\ a_{19}\ a_{18}\ a_{17}\ a_{16}$	PRF_CRC_INIT[0] = 0 PRF_CRC_INIT[1] = X PRF_CRC_INIT[2] = X PRF_CRC_INIT[3] = X
32	4	$x^{32} + a_{31}x^{31} + \ldots + a_1x^1 + 1$	BSP_P0 = $a_7\ a_6\ a_5\ a_4\ a_3\ a_2\ a_1 1$ BSP_P1 = $a_{15}\ a_{14}\ a_{13}\ a_{12}\ a_{11}\ a_{10}\ a_9\ a_8$ BSP_P2 = $a_{23}\ a_{22}\ a_{21}\ a_{20}\ a_{19}\ a_{18}\ a_{17}\ a_{16}$ BSP_P3 = $a_{31}\ a_{30}\ a_{29}\ a_{28}\ a_{27}\ a_{26}\ a_{25}\ a_{24}$	PRF_CRC_INIT[0] = X PRF_CRC_INIT[1] = X PRF_CRC_INIT[2] = X PRF_CRC_INIT[3] = X

Table 25-9. Register Settings for Some Commonly Used CRCs, Assuming Initialization With All 1s

Order	PRF_CRC_LEN	CRC	BSP_Px	PRF_CRC_INIT
8	1	CRC-8-ATM $x^8 + x^2 + x + 1$	BSP_P0 = 0x00 BSP_P1 = 0x00 BSP_P2 = 0x00 BSP_P3 = 0x07	PRF_CRC_INIT[0] = 0x00 PRF_CRC_INIT[1] = 0x00 PRF_CRC_INIT[2] = 0x00 PRF_CRC_INIT[3] = 0xFF
8	1	CRC-8 $x^8 + x^7 + x^6 + x^4 + x^2 + 1$	BSP_P0 = 0x00 BSP_P1 = 0x00 BSP_P2 = 0x00 BSP_P3 = 0xD3	PRF_CRC_INIT[0] = 0x00 PRF_CRC_INIT[1] = 0x00 PRF_CRC_INIT[2] = 0x00 PRF_CRC_INIT[3] = 0xFF
16	2	CRC-16 (used in CC2500) $x^{16} + x^{15} + x^2 + 1$	BSP_P0 = 0x00 BSP_P1 = 0x00 BSP_P2 = 0x05 BSP_P3 = 0x80	PRF_CRC_INIT[0] = 0x00 PRF_CRC_INIT[1] = 0x00 PRF_CRC_INIT[2] = 0xFF PRF_CRC_INIT[3] = 0xFF
16	2	CRC-16-CCITT $x^{16} + x^{12} + x^5 + 1$	BSP_P0 = 0x00 BSP_P1 = 0x00 BSP_P2 = 0x21 BSP_P3 = 0x10	PRF_CRC_INIT[0] = 0x00 PRF_CRC_INIT[1] = 0x00 PRF_CRC_INIT[2] = 0xFF PRF_CRC_INIT[3] = 0xFF
24	3	CRC-24 $x^{24} + x^{22} + x^{20} + x^{19} + x^{18} + x^{16} + x^{14} + x^{13} + x^{11} + x^{10} + x^8 + x^7 + x^6 + x^3 + x + 1$	BSP_P0 = 0x00 BSP_P1 = 0xCB BSP_P2 = 0x6D BSP_P3 = 0x5D	PRF_CRC_INIT[0] = 0x00 PRF_CRC_INIT[1] = 0xFF PRF_CRC_INIT[2] = 0xFF PRF_CRC_INIT[3] = 0xFF
32	4	CRC-32-IEEE 802.3 $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$	BSP_P0 = 0xB7 BSP_P1 = 0x1D BSP_P2 = 0xC1 BSP_P3 = 0x04	PRF_CRC_INIT[0] = 0xFF PRF_CRC_INIT[1] = 0xFF PRF_CRC_INIT[2] = 0xFF PRF_CRC_INIT[3] = 0xFF

25.4.4 Coprocessor Mode

The coprocessor mode is used to run the BSP as a stand-alone and not part of the signal path. It must not be used while the LLE is running. Coprocessor mode is selected by setting BSP_MODE.CP_MODE to 01 or 11. In these modes, one byte to be processed is written to the BSP_DATA register, and the result of processing this byte can later be read back from the same register. When BSP_MODE.CP_MODE is 01, the coprocessor is in receive mode, where the whitener is applied before the CRC. When BSP_MODE.CP_MODE is 11, the coprocessor is in transmit mode, where the whitener is applied after the CRC.

To apply the BSP operations to a byte, write it to the BSP_DATA register. When this register is written to, the BSP_MODE.CP_BUSY bit goes high.

If CP_MODE.CP_END is 0, the first bit provided is the LSB and the last bit is the MSB. If CP_MODE.CP_END is 1, the first bit provided is the MSB and the last bit is the LSB.

When BSP_MODE.CP_BUSY goes low, the processed data can be read from the BSP_DATA register. If one or both whiteners are enabled, this byte is whitened or de-whitened. Otherwise, it is the same as the byte written, except if the CRC is being read as described in the following text.

To read out a CRC in transmit mode, set BSP_MODE.CP_READOUT to 1. A zero must be written to the BSP_DATA register, and when BSP_MODE.CP_BUSY goes low, a CRC byte can be read from BSP_DATA. This should be repeated for each CRC byte. If whitening is enabled, the read back CRC bytes are whitened.

The BSP must not be set in coprocessor mode while the LLE is processing a packet.

25.5 Frequency and Channel Programming

For normal transmit and receive tasks, the carrier frequency is set by using the PRF_CHAN.FREQ register. The carrier frequency is 2379 + *n* MHz, where *n* is the value of this register, and *n* can be from 0 to 116. This gives a frequency range from 2379 MHz to 2495 MHz. Note that this frequency range extends beyond the ISM band.

If PRF_CHAN.FREQ is set to 127, and for the RX and TX test commands, the frequency must be programmed directly in hardware registers. In this case, the synthesizer frequency is set by programming the 7-bit frequency word located in FREQCTRL.FREQ[6:0]. The synthesizer frequency is given by 2379 + *n* MHz, where n is the value of FREQCTRL.FREQ[6:0], and is programmable in 1-MHz steps. The device supports synthesizer frequencies in the range from 2379 MHz to 2495 MHz. The usable settings for FREQ[6:0] is consequently 0 to 116.

In RX, the system operates on a low intermediate frequency (IF) of 1 MHz for data rates up to 1 Mbps, and on a zero IF for 2 Mbps. In TX, the system supports operating on low IF or zero IF. The IF to be used for TX can be programmed in the register MDMTEST1.TX_TONE. The receiver may operate on a positive or negative IF when the data rate is 1 Mbps and lower; this is controlled with MDMTEST1.RX_IF.

When the symbol rate is 1 Mbps or lower and the LLE programs the frequency, it uses a ±1 MHz IF on TX. For both RX and TX, a negative IF is used when PRF_CHAN.FREQ < 62, and a positive IF is used when PRF_CHAN.FREQ ≥ 62

When the symbol rate is 2 Mbps and the LLE programs the frequency, it uses an IF on TX as specified in PRF_RADIO_CONF.TXIF. This IF may be zero, or ±1 MHz, ±2 MHz, or ±3 MHz. The recommended setting is ±1 MHz. A negative IF is used when PRF_CHAN.FREQ < 62, and a positive IF is used when PRF_CHAN.FREQ ≥ 62.

For all data rates, the setting of MDMCTRL1.PHASE_INVERT is taken into account by the LLE when finding the setting for MDMTEST1.TX_TONE. The FREQCTRL register is programmed corresponding to the programmed IF in order to operate on the channel specified by PRF_CHAN.FREQ.

25.6 Modulation Formats

The CC2541 supports GFSK and MSK modulation formats. For GFSK modulation, the deviation can be set to 160 kHz or 250 kHz (320 kHz or 500 kHz for 2 Mbps). The data rate can be set to 250 kbps, 500 kbps, 1 Mbps, or 2 Mbps. The desired modulation scheme is set in the MDMCTRL0.MODULATION register.

Not all combinations of modulation format, data rate and deviation are supported. Table 25-10 gives an overview of supported combinations.

Table 25-10. Supported Modulation Formats, Data Rates, and Deviations

Modulation Format	Data Rate	Deviation	MDMCTRL0.MODULATION
GFSK	2 Mbps	500 kHz	0011
GFSK	2 Mbps	320 kHz	0111
GFSK	1 Mbps	250 kHz	0010
GFSK	1 Mbps	160 kHz	0110
GFSK	250 kbps	160 kHz	0100
MSK	500 kbps	–	1001
MSK	250 kbps	–	1000

25.7 Receiver

When the receiver is started, it searches for the preamble and the sync word. These are used for frequency offset compensation and bit and byte synchronization. The sync word can be programmed to be from 16 to 32 bits.

Checking the sync word is done in a two-stage process. First, a correlation value is calculated. If this correlation is above a programmable threshold, a data decision of the received sync word is done. It can be programmed in MDMCTRL3.SYNC_MODE whether this data decision is to be ignored, no bit errors are to be accepted, or one bit error is to be accepted. The correlation threshold value is programmed in MDMCTRL1.CORR_THR. This threshold value should depend on the sync word length. As a rule of thumb, a value of 0.25 times the number of bits (rounded down) can be used.

For the bit synchronization to work well, some guidelines should be followed for the sync word. It should have enough transitions, but not long runs of 10 1010... or other short, repeated patterns. Generally, a longer sync word gives better performance.

The CC2541 devices have support for two independent sync words. The primary and secondary sync words are specified in two sets of registers. The secondary sync word can be enabled by the SW_CONF.DUAL_RX bit, and if enabled, the received signal is correlated against both sync words. If the correlation with one of the sync words is above the threshold, data decision is done against that sync word.

While the receiver is running, a received signal strength indicator (RSSI) is updated. The RSSI is available some time after the receiver is started, regardless of whether sync is found. It can be read from the RSSI register, which is 0x80 when no RSSI is available. The value given is in the range 0 to approximately 64, with a change of 1 corresponding to a 1-dB change. The offset from a true dBm value depends on the receiver mode and can be found in the device data sheet. For high received signal levels, the reported RSSI saturates at one of the highest possible reported values. The accuracy and update time of the RSSI can be traded off using MDMTEST0.RSSI_ACC. The RSSI can be calculated over a window of 5.33 µs or 21.3 µs, and 1, 2, or 4 such windows can be averaged to give the result. Using a longer average time gives higher accuracy, but it takes longer before a result is ready, and doing the average over a longer time means that the result may be wrong for short packets. An average of n windows of length t_{RSSI} should only be used for packets lasting longer than $(n + 1)\ t_{RSSI}$ (including preamble, sync word, and CRC).

The receiver must run dc offset estimation and removal. The dc offset estimation mode can be controlled with MDMTEST0.DC_BLOCK_MODE. For data rates of 1 Mbps and lower, where the receiver runs on a low IF, it is recommended to use the default setting for this register (continuous estimation). For 2 Mbps, where the receiver runs on zero IF, delayed dc offset estimation should normally be used. This causes the dc offset estimation to be done in front of the packet. The delay can be controlled through MDMTEST0.DC_BLOCK_LENGTH and MDMTEST1.DC_DELAY. The recommendation is to set MDMTEST0.DC_BLOCK_LENGTH to 11 (128 samples) and MDMTEST1.DC_DELAY to 00 (5 delays), which allows for up to approximately 105 µs of energy in front of the packet payload, including the preamble and sync word. As an alternative for 2 Mbps, dc offset estimation can be turned off, and a previously found value can be used, written into the DC_I_L, DC_I_H, DC_Q_L, and DC_Q_H registers. Values can be found in advance, but differ for each frequency. For auto acknowledgments and other packets that are received at a known time, the LLE can perform a special dc offset algorithm as described in Section 25.9.2.

25.8 Packet Format

The packet format is configurable. There are two operation modes for radio packet control, basic mode and auto mode. Of these, only auto mode supports automatic acknowledgment and retransmissions. The LLE-controlled part of the packet format is also different for the two modes. In basic mode, there is an optional length field followed by an optional address of 1 byte, as shown in Figure 25-6. In auto mode, there is a 9-bit or 10-bit header field containing length and sequence number information. This format is shown in Figure 25-7. The figures show the packet formats with their configurability. The fields with a header in gray are controlled directly by the modem and are used in the acquisition of received packets. The fields with header in white are controlled by the LLE.

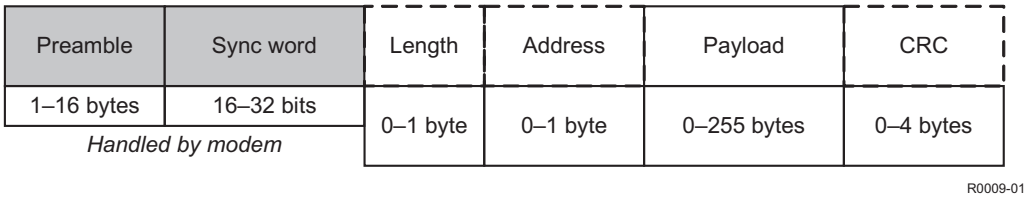
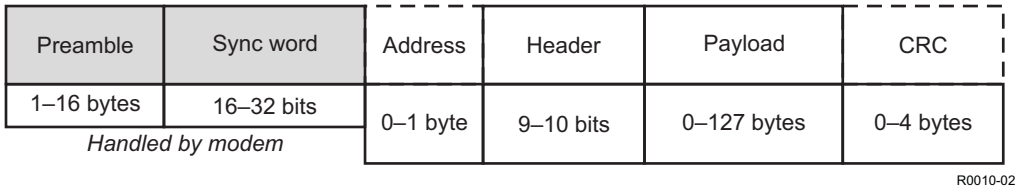


Figure 25-6. Air Interface Packet Format for Basic Mode



When using a 9 bit header, the payload length is limited to the range 0-63 bytes. Note that the LLE must be reset when the device enters PM2 or PM3. This means that the PRFX registers must be re-initialized after the LLE has been re-enabled after coming up from one of these power modes.

Figure 25-7. Air Interface Packet Format for Auto Mode

The preamble is a sequence of 1010 1010 or 0101 0101. It can be from 1 to 16 bytes. The type of preamble and the number of bytes can be set up in the MDMCTRL2 register.

The sync-word field is a synchronization word that can have any length from 16 to 32 bits. The length is programmed in the SW_CONF.SW_LEN register. The sync word itself is programmed in the SW0, SW1, SW2, and SW3 registers for the primary sync word, and SW4, SW5, SW6, and SW7 for the secondary sync word. The bit ordering of the sync word is set up with MDMCTRL2.SW_BIT_ORDER. If SW_BIT_ORDER is 0, the LSB of SW0 (SW4) is transmitted first and the MSB of SW3 (SW7) is transmitted last. If SW_BIT_ORDER is 1, the MSB of SW3 (SW7) is transmitted first and the LSB of SW0 (SW4) is transmitted last. The first bit transmitted is always the same regardless of the sync word length; the unused bits for sync word length of less than 32 bits are the ones that would have been transmitted last.

The optional length byte in basic mode (see Figure 25-6) is present if PRF_TASK_CONF.MODE = 01. It indicates the number of address and payload bytes following the length byte. If the length field is not present, the length is fixed as described in Section 25.9.2.

The optional address is 1 byte if present; the length is configured with the PRF_PKT_CONF.ADDR_LEN register. In the transmitter, the address can be used for identification or to direct the message to a particular receiver, and in the receiver, the address can be used to filter out messages from unknown or unwanted transmitters and to distinguish between messages from different transmitters. See Section 25.9.2 for details on how the address is used. Note that for the packet format in Figure 25-7 or if a length field is not used, the address field immediately follows the sync word, and can thus be seen as an extension of it.

The 9-bit or 10-bit header shown in Figure 25-7 is shown in more detail in Figure 25-8 and Figure 25-9. This field consists of a 6-bit or 7-bit length followed by a 2-bit sequence number and a flag called NO_ACK (NOA in Figure 25-8 and Figure 25-9) to inform that acknowledgment of the packet is not expected. If the configuration is to use a fixed length, the value of the length field is ignored in the receiver. It can be configured always to set the length field to 11 0011 in the transmitter for fixed-length packets.

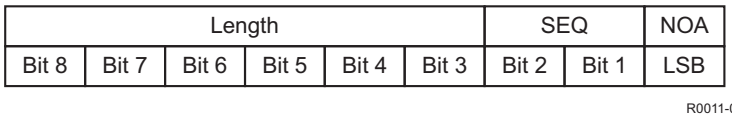


Figure 25-8. Bits of 9-Bit Header

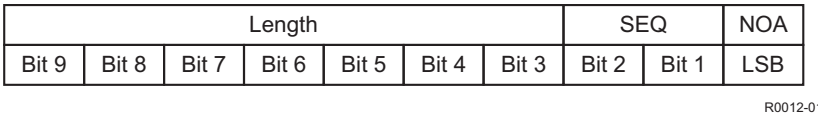


Figure 25-9. Bits of 10-Bit Header

The payload can be from zero to 255 bytes in basic mode, but the sum of the number of address and payload bytes must not exceed 255. In auto mode, the payload can be from 0 to 63 bytes with a 9-bit header or 0 to 127 bytes with a 10-bit header. The maximum packet length can be limited, see Section 25.9.2.3.1 and Section 25.9.2.3.2

The bit ordering when transmitting the length, address, payload, and CRC bytes is set up with the ENDIANNESS bit of the FRMCTRL0 register; if 0, the LSB of each byte is transmitted first and if 1, the MSB is transmitted first. Normally, FRMCTRL0.ENDIANNESS and MDMCTRL2.SW_BIT_ORDER should have the same value. Note that for correct operation in auto mode, FRMCTRL0.ENDIANNESS must be set to 1 so that MSB is transmitted first.

The CRC field contains 0 to 4 bytes and is used to check the packet for errors if present. See Section 25.4.3 on how to set up the CRC generation and checking.

25.8.1 RX FIFO Packet Organization

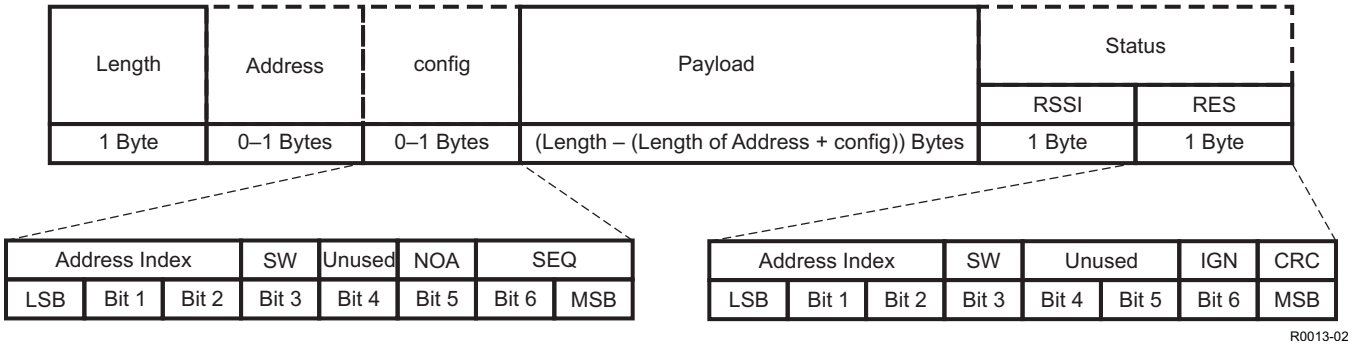


Figure 25-10. Structure of Packets in the RX FIFO

The structure of a packet in the RX FIFO is shown in Figure 25-10. All packets start with a length byte, regardless of whether a length byte is present on the air. The length is the number of bytes in the address, config, and payload fields following the length byte, and it may be modified compared to the length received on the air or configured as fixed-length. If packets are longer than what can fit in the FIFO, packets must be read from the FIFO while reception takes place, either by DMA or directly by the MCU. The auto-flush options in PRF_FIFO_CONF cannot be used in this case, and auto-commit and auto-deallocate must be enabled for the RX FIFO in RFFCFG.

The address byte is placed after the length byte and is present if configured in PRF_FIFO_CONF.RX_ADDR_CONF. The address is written in the FIFO as it was received on the air.

The config byte following the length byte and address byte is present if configured in PRF_FIFO_CONF.RX_ADDR_CONF. In this case, the index *n* to the PRF_ADDR_ENTRYn containing the received address is present in bits 0–2, and bit 3 is 0 if the primary sync word was received and 1 if the secondary sync word was received. In auto mode, the 3 MSBs of the config byte are set to the 3 LSBs of the received header.

The payload is as received on the air. In case of an empty packet, there is no payload.

The status field consists of 2 bytes appended to the FIFO entry if configured in PRF_FIFO_CONF.RX_STATUS_CONF. The presence of a status field is not reflected in the value of the length byte, so if a status field is present, the MCU must read 2 extra bytes. It is possible to configure this even using DMA with automatic length extraction. The status bytes are:

- RSSI is the received signal-strength indication from the demodulator.
- RES contains information on the address and CRC result.
  - The 3 LSBs contain the address index as in the config byte.
  - Bit 3 is 0 if the primary sync word was received and 1 if the secondary sync word was received.
  - IGN is 1 for packets that may be ignored by the MCU due to repeated sequence number and 0 otherwise.
  - CRC is 1 if there was a CRC error and 0 otherwise.

25.8.2 TX FIFO Packet Organization

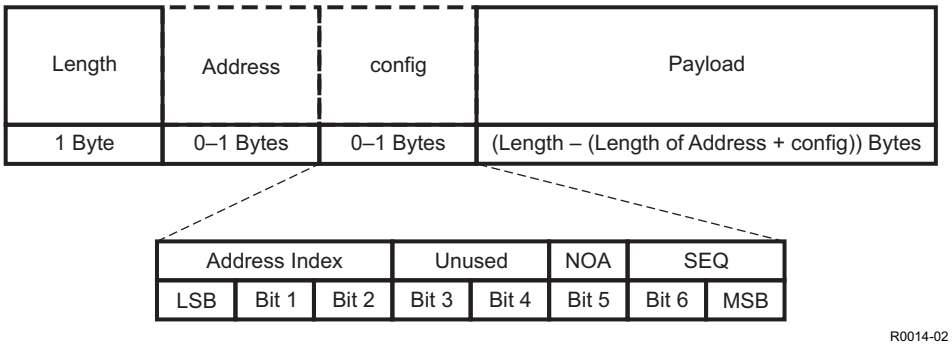


Figure 25-11. Structure of Packets in the TX FIFO

The structure of a packet in the TX FIFO is shown in Figure 25-11. All packets start with a length byte, regardless of whether a length byte is present on the air. The length is the number of bytes in the address, config, and payload fields following the length byte, and it may be modified before being transmitted on the air. If a fixed length is used, it is up to the MCU to ensure that the length is correct given the fixed length expected by the receiver. If packets are longer than what can fit in the FIFO, packets must be written to the FIFO while transmission takes place, either by DMA or directly by the MCU. Auto-commit and auto-deallocate must then be enabled for the TX FIFO in RFFCFG.

The address byte is placed after the length byte and is present if configured in PRF_FIFO_CONF.TX_ADDR_CONF. If it is included, the address is transmitted on the air as it is read from the FIFO. If it is not included, but a config byte is included, the three LSBs of the config byte tell the index *n* of PRF_ADDR_ENTRYn from which the address is inserted. If neither an address nor a config byte is included, the address is inserted from PRF_ADDR_ENTRY0.ADDRESS.

The config byte following the length byte and optional address byte is present if configured in PRF_FIFO_CONF.TX_ADDR_CONF. This byte contains an address index which is used to determine the address if no address byte is included as explained previously. If an address byte is included, the address index is used to determine which address entry to read the configuration from, but the ADDRESS field in that address entry is ignored. In auto mode, the NO_ACK bit (LSB) of the transmitted header is set to bit 5 of the config byte. If PRF_ADDR_ENTRYn.CONF.FIXEDSEQ, where *n* is the index of the address used, is 1, the SEQ field of the transmitted header is taken from the SEQ field (bits 6–7) of the config byte; otherwise, the sequence number on the air is inserted from PRF_ADDR_ENTRYn.SEQSTAT.SEQ. If the config byte is not included, the NO_ACK bit is always sent as 0 and PRF_ADDR_ENTRYn.CONF.FIXEDSEQ should be 0 (otherwise the SEQ field always remains 0).The payload is transmitted as present in the FIFO.

25.8.3 TX Buffers for ACK Payload

The hardware TX FIFO is not used for ACK payload in RX tasks in auto mode. Instead, an acknowledgment packet for each address can be placed in one of two dedicated buffers for that address. These two buffers constitute a FIFO capable of holding two packets. The buffers for the first two addresses are placed in the RAM page normally used for the hardware TX FIFO. These four buffers can either be accessed from the TX FIFO space at 0x6100 or by selecting page 7 through RFRAMCFG, but the TX FIFO registers should not be used. The other twelve buffers must be addressed from the configurable radio memory bank through the RFRAMCFG register. The mapping of each buffer is shown in Table 25-11.

Table 25-11. Segments for Holding ACK Payload for Each Address Entry

Address Entry Number	Buffer Number	Setting of RFRAMCFG	Start Address
0	0	7 *or* X	0x6000 *or* 0x6100
0	1	7 *or* X	0x6020 *or* 0x6120
1	0	7 *or* X	0x6040 *or* 0x6140
1	1	7 *or* X	0x6060 *or* 0x6160
2	0	1	0x6000

Table 25-11. Segments for Holding ACK Payload for Each Address Entry (continued)

Address Entry Number	Buffer Number	Setting of RFRAMCFG	Start Address
2	1	1	0x6020
3	0	1	0x6040
3	1	1	0x6060
4	0	2	0x6000
4	1	2	0x6020
5	0	2	0x6040
5	1	2	0x6060
6	0	3	0x6000
6	1	3	0x6020
7	0	3	0x6040
7	1	3	0x6060

The status of buffer *k* for address *n* is contained in the PRF_ADDR_ENTRYn.ACKLENGTHk register. If the value is 0, the buffer is free.

In order to enter a payload for address *n*, the MCU must follow the following procedure:

1. Read PRF_ADDR_ENTRYn.ACKLENGTH0 and PRF_ADDR_ENTRYn.ACKLENGTH1. Call the values len_0 and len_1, respectively.
2. Read PRF_ADDR_ENTRYn.SEQSTAT.NEXTACK and call this value *k*. Let *m* be NOT *k* (that is, 1 – *k*).
3. Check if len_*k* is 0. If so, write the payload to buffer *k* for address entry *n* (see Table 25-11), then write the payload length to PRF_ADDR_ENTRYn.ACKLENGTHk. End the procedure.
4. Otherwise, check whether len_ *m* is 0. If so, write the payload to buffer *m* for address entry *n* (see Table 25-11), then write the payload length to PRF_ADDR_ENTRYn.ACKLENGTHm. End the procedure.
5. Otherwise, no ACK payload buffer for that address is free, and no payload can be entered at this time.

The ACK payload length can be 1–32. When a buffer becomes free, the LLE writes the PRF_ADDR_ENTRYn.ACKLENGTHk to 0 and raises a TXDONE interrupt.

A buffer contains only the payload to be transmitted. The length is given by PRF_ADDR_ENTRYn.ACKLENGTHk, and the address and sequence number are as described in Section 25.9.2.3.2.

In order to flush the buffers for address *n*, issue the command CMD_FLUSH_ACK *n* (see Table 25-12). This causes the LLE to write PRF_ADDR_ENTRYn.ACKLENGTH0 and PRF_ADDR_ENTRYn.ACKLENGTH1 to 0 and clear PRF_ADDR_ENTRYn.SEQSTAT.ACK_PAYLOAD_SENT. If no task is running, the LLE takes SEMAPHORE1; if it fails, it does not write to PRF_ADDR_ENTRYn.SEQSTAT.ACK_PAYLOAD_SENT. If the transmission of an acknowledgment with payload had started on that address, flushing happens after the transmission is finished. After the flushing is done, the LLE raises a TXFLUSHED interrupt.

25.9 Link Layer Engine

The link layer engine controls radio operation. It is started by setting the LLECTRL.LLE_EN bit to 1. The LLE must be started before the radio can be operated.

The LLE can be reset by clearing and setting LLECTRL.LLE_EN. The LLE should not be reset while the radio is active. The MCU should not enter PM1, PM2, or PM3 while the LLE is running a task. Before entering PM2 or PM3, LLECTRL.LLE_EN must be set to 0, otherwise the behavior of the RF core may be unpredictable after waking up. The mode of the LLE is selected with LLECTRL.LLE_MODE_SEL. For the proprietary-mode operation described in this chapter, this field must be written to 00. For BLE operation, this field is 01; that value should only be written by the TI BLE stack. In order to switch modes, the LLE must be reset; writing to LLECTRL.LLE_MODE_SEL while LLECTRL.LLE_EN is 1 has no effect.

25.9.1 Command Register

The command register RFST can be used for sending commands to the LLE and the FIFOs. Commands in the range 0x80–0xFF are commands to the FIFOs, see Section 25.3.1. Other commands are commands to the LLE.

The commands are listed in Table 25-12. There are commands for starting receive and transmit modes. In addition, there is a command CMD_SHUTDOWN to stop the radio operation and end the task directly. The commands CMD_SEND_EVENT1 and CMD_SEND_EVENT2 do the same action as receiving a Timer 2 event 1 or event 2.

If an unknown command is entered, the LLE responds by generating an LLEERR interrupt. If a task is running, it stops.

When sending a command to the LLE, the RFST register retains its value until the LLE has received the command (but not necessarily executed it) and then is set to 0. Commands should not be sent to the LLE unless RFST is 0. FIFO commands may be sent at any time.

Table 25-12. Commands From MCU to LL Engine via RFST Register

Number	Command Name	Description
0x01	CMD_SHUTDOWN	Stop operation immediately
0x02	CMD_DEMOD_TEST	Start demodulator without sync search
0x03	CMD_RX_TEST	Start demodulator and sync search
0x04	CMD_TX_TEST	Start transmitter and transmit zeros
0x05	CMD_TX_FIFO_TEST	Start transmitter and transmit from TX FIFO
0x06	CMD_PING	Respond with a PINGRSP interrupt
0x08	CMD_RX	Start receive operation
0x09	CMD_TX	Start transmit operation
0x0A	CMD_TX_ON_CC	Start transmit operation on clear channel
0x0B	CMD_STOP	Gracefully stop radio task
0x21	CMD_SEND_EVENT1	Do the same action as if Timer 2 event 1 was observed
0x22	CMD_SEND_EVENT2	Do the same action as if Timer 2 event 2 was observed
0x30	CMD_FLUSH_ACK0	Flush the ACK payload buffers for address 0
0x31	CMD_FLUSH_ACK1	Flush the ACK payload buffers for address 1
0x32	CMD_FLUSH_ACK2	Flush the ACK payload buffers for address 2
0x33	CMD_FLUSH_ACK3	Flush the ACK payload buffers for address 3
0x34	CMD_FLUSH_ACK4	Flush the ACK payload buffers for address 4
0x35	CMD_FLUSH_ACK5	Flush the ACK payload buffers for address 5
0x36	CMD_FLUSH_ACK6	Flush the ACK payload buffers for address 6
0x37	CMD_FLUSH_ACK7	Flush the ACK payload buffers for address 7

25.9.2 Radio Tasks

Before starting a task, radio registers should be set up with the desired packet format, and the desired input sensitivity and output power should be programmed. Furthermore, the sync word in use must be programmed in the SW0, SW1, SW2, and SW3 registers. If a secondary sync word is used, it must be programmed in the SW4, SW5, SW6, and SW7 registers. The RAM registers must be programmed to configure the task. The way the task runs depends on the PRF_TASK_CONF register. The operation mode is set up by the MODE bits of this register. A value of 00 or 01 gives basic mode and thus disables auto ACK or auto retransmission. A value of 10 or 11 gives auto mode where auto acknowledgment or auto retransmission can be enabled per the address in PRF_ADDR_ENTRYn.CONF.AA.

All tasks start with a start-of-task command from the MCU. The LLE takes SEMAPHORE0 at this time; if the semaphore is not available, the task ends with an error. Depending on the configuration in PRF_TASK_CONF.START_CONF, the LLE either starts the task immediately or waits for a Timer 2 event 1 before starting. Note that a Timer 2 event 1 may be pending from before the LLE starts waiting; in that case, the task starts immediately. To clear a pending Timer 2 event 1, reset the LLE. To prevent

unwanted events from reaching the LLE, Timer 2 event 1 can be disabled in the T2EVTCFG register; see Chapter 22. The frequency word is programmed according to the setting of PRF_CHAN.FREQ, except if it is 127, in which case no frequency programming is done and any value written by the MCU is retained. When using auto mode on 2 Mbps, the frequency must be programmed through the PRF_CHAN.FREQ register. Then the LLE changes the IF frequency automatically (for 2 Mbps, the recommended settings use different IF for transmission and reception) when changing from receive operation to transmit operation (for sending an acknowledgment packet) and vice versa. The LLE starts configuring the transmitter or receiver, depending on the type of task. After the transmitter or receiver has been set up, the LLE takes SEMAPHORE1 to gain access to the remaining RAM-based registers, read the parameters, and start transmission or reception.

Programming of frequency is done as described in Section 25.5. For symbol rates of 1 Mbps and lower, RX and TX are done on the same synthesizer frequency, whereas for a symbol rate of 2 Mbps, the synthesizer frequency changes between RX and TX. This change is done without a recalibration of the synthesizer.

At the end of a packet, the LLE reads the RSSI register and writes the value to the PRF_LAST_RSSI register and, if so configured, to the RSSI byte of the RX FIFO. This read is done after the next-to-last byte has been obtained from the demodulator. Note that for a bit rate of 2 Mbps and for sync words shorter than 32 bits, MDMCTRL3.RSSI_MODE should be set to 11 to ensure a correct reading. Before turning off the demodulator, the LLE reads the dc offset from the DC_I_L, DC_I_H, DC_Q_L, and DC_Q_H registers and writes the result to PRF_LAST_DCOFF (in the byte order listed for the register read). The LLE also reads the frequency offset from the FREQEST register and writes the result to PRFX_LAST_FREQEST (see Table 25-7).

If PRF_RADIO_CONF.DCOFF is 1, the LLE runs a procedure that estimates the dc offset right after receiver startup. This mode is suitable for packets that are known to be received at a certain time, such as acknowledgment packets. In this mode, the LLE starts the receiver with normal dc cancellation mode and forces the LNA gain to minimum. After a short time, the LLE reads out the value of the dc offset estimate, writes it into the override registers, and selects manual override mode for dc offset estimation. It sets the LNA gain back to the programmed value and after a waiting time to allow the LNA to stabilize, starts sync search. The time to start RX with this mode is the same as for ordinary start of RX.

If PRF_RADIO_CONF.DCWB is 1, the LLE writes the dc offset estimate read out at the end of the packet into the dc offset override register, provided that the received packet did not have a CRC error. This is suited for the delayed dc offset mode, where the override value for dc offset is used before a delayed dc offset is available.

Some of the RAM registers are checked by the LLE to verify that their values are permitted. This applies to PRF_CHAN.FREQ, PRF_FIFO_CONF.TX_ADDR_CONF, and PRF_CRC_LEN. If any of these registers has values that are not permitted, the task ends with an error.

A CMD_SHUTDOWN command, undefined command, or any command starting a new task, ends the task immediately. If a packet was being transmitted or received, an RXTXAB0 interrupt to the MCU is raised. This means that to avoid unwanted abort of commands, the CPU should wait for a TASKDONE interrupt or check that LLESTAT.LLE_IDLE is 1 before starting another command.

If a CMD_STOP command is received, the task ends after the current reception or transmission is done. Timer 2 event 2 can be configured to end a task: If PRF_TASK_CONF.STOP_CONF is 01, Timer 2 event 2 behaves as a CMD_STOP, and if PRF_TASK_CONF.STOP_CONF is 10, Timer 2 event 2 behaves as a CMD_SHUTDOWN. Setting PRF_TASK_CONF.STOP_CONF to 00 disables Timer 2 event 2 as a stop event. With the 11 setting, Timer 2 event 2 only applies to sync search or listen right after a CMD_RX or CMD_TX_ON_CC (this setting is not meaningful for a CMD_TX task) or a start by Timer 2 event 1. This is explained in later subsections.

Timer 2 may capture the time of a packet based on the setting in PRF_RADIO_CONF. The fields TXCAP and RXCAP decide how capture is configured for TX and RX, respectively; see Table 25-13. The captured value can be read from the registers T2M0, T2M1, T2MOVF0, T2MOVF1, and T2MOVF2 when t2_cap and t2ovf_cap are selected using the T2MSEL register; see Chapter 22.

Table 25-13. Timer 2 Capture Settings

TXCAP	Description
00	Capture of transmitted packets off
01	Capture the start (after the sync word) of every transmitted packet
10	Capture the end of every transmitted packet
11	Capture the start of the first transmitted packet, that is, capture of transmitted packets is turned off after a packet has been transmitted.
RXCAP	**Description**
00	Capture of received packets off
01	Capture the start (after the sync word) of every received packet
10	Capture the end of every received packet
11	Capture the start of the first received packet, that is, capture of received packets is turned off after a packet has been fully received.

When capture is done at the beginning of a packet, the time captured is the time right after the sync word has been received or transmitted. Setting TXCAP or RXCAP to 11 enables capture at the start of a packet, but the capture is turned off after a packet has been transmitted or fully received in a task, so it is the start of the first packet in the task that is captured. The MCU should normally only read the captured value after a task is done; otherwise, the captured value may be overwritten with a new value. The user must take into account that a timer value may be captured on a received packet that does not match the address or that has a length which is not permitted, and that is thus not reported. It is possible to turn on capture for both received and transmitted packets in the same task. If so, it is up to the user to determine if the captured value was from a received or transmitted packet.

When a task is finished, the LLE writes an end-of-task cause in PRF_ENDCAUSE, frees the semaphores, raises a TASKDONE interrupt, and halts its operation. The possible values of PRF_ENDCAUSE are listed in Table 25-14.

If PRF_CHAN.SYNTH_ON is 1, the synthesizer is not turned off after the task ends. This can be used to start a new task immediately on the same channel and get faster start of RX or TX. To do so, the next task should be started with PRF_CHAN.FREQ set to 127. Note that the synthesizer should not be allowed to run for a long time after a task has ended, as this causes excessive power consumption. The synthesizer can be stopped by sending a CMD_SHUTDOWN command.

Table 25-14. End-of-Task Causes

Number	Name	Description
Normal task ending		
0	TASK_ENDOK	Task ended normally
1	TASK_RXTIMEOUT	Timer 2 event 2 or CMD_STOP observed while waiting for RX sync
2	TASK_NOSYNC	Sync was not obtained in the specified time
3	TASK_NOCC	TX_ON_CC ended because channel was not clear
4	TASK_MAXRT	Task ended because maximum number of retransmissions was reached
5	TASK_STOP	Task ended after transmission or reception by Timer 2 event 2 or CMD_STOP while transmitting or receiving or with ACK or retransmission in progress
6	TASK_ABORT	Task aborted by command
MCU interface error		
255	TASKERR_INTERNAL	Internal program error
254	TASKERR_CMD	Unknown command
253	TASKERR_SEM	Unable to obtain semaphore
252	TASKERR_PAR	Unpermitted parameter
251	TASKERR_TXFIFO	TX FIFO without available data when not permitted
250	TASKERR_RXFIFO	Overfull RX FIFO in TX task

Table 25-14. End-of-Task Causes (continued)

Number	Name	Description
249	TASKERR_MODUNF	Modulator underflow observed

25.9.2.1 AGC Algorithm

If PRF_PKT_CONF.AGC_EN is 1, an automatic gain control (AGC) algorithm is run while the receiver is looking for sync. The AGC algorithm switches between two different front-end gain settings in the LNAGAIN register. It is recommended to use AGC when running on 2 Mbps to improve the saturation performance.

Parameters for control of the AGC algorithm are found in page 5 of the radio RAM; see Table 25-7. This table lists reset values that the LLE sets for these parameters.

The LLE polls the RSSI value at every update and compares it to the values of PRFX_RSSI_LIM_LOWER and PRFX_RSSI_LIM_UPPER. If the observed RSSI is below PRFX_RSSI_LIM_LOWER, the LNA gain is set to the high gain setting. If the observed RSSI is above PRFX_RSSI_LIM_UPPER, the LNA gain is set to the low gain setting. If the observed RSSI is between these limits, the LNA gain is not changed.

The high gain to use is the value found in the LNAGAIN register when the task is started. The low gain to use is the value found in the PRFX_LNAGAIN_SAT RAM register.

The PRFX_RSSI_LIM_LOWER and PRFX_RSSI_LIM_UPPER values must differ in order to account for the difference that is observed from the RSSI register when the LNA gain is changed and to have hysteresis to avoid too-frequent gain changes.

When sync is obtained on the receiver, the AGC algorithm stops updating the LNA gain, which remains at the value last set. When the receiver is switched off, the LNAGAIN register is set back to the value it had when the task started, that is, the high gain setting.

When the gain is reduced during the reception of a packet, the value found in the PRF_LAST_RSSI register and (if configured) in the RSSI byte of the RX FIFO is updated to reflect this. This update is done by adding the value of the register PRFX_RSSI_DIFF to the value found in the RSSI register. PRFX_RSSI_DIFF should therefore contain the difference between the RSSI offset for the two LNA gain settings in use, available from the device data sheet. Note that the hardware RSSI register is not updated this way.

For the AGC algorithm to operate correctly, it requires some signal, having the same power as the packet, transmitted in the band in front of the packet. That signal can be extra preamble bytes or tone. The length required for this signal depends on the RSSI accuracy setting in MDMTEST0.RSSI_ACC, see Section 25.7. An average of n windows of length t_{RSSI} requires the extra signal to last at least $(n + 1)\ t_{RSSI}$. Extra preamble bytes can be set up using MDMCTRL2.NUM_PREAM_BYTES. Note that the extra signal required comes in addition to the 1 preamble byte always used in a packet. When adding extra preamble bytes, this must be accounted for in PRF_TX_DELAY, PRF_RETRANS_DELAY, and PRF_RX_TX_TIME. The RX requires $n \times t_{RSSI}$ extra time to start when using the AGC. In the dc offset estimation, the extra signal must be accounted for when setting the delay.

25.9.2.2 Tone in Front of Packet

In order to get the transmission format to resemble that of other vendors, a tone may be sent in front of the preamble. This tone can be used by the AGC algorithm on the receiver side. If PRF_PKT_CONF.START_TONE is 1, such a tone is transmitted as a replacement of the first preamble bytes. This means that this feature must only be used in combination with increasing the number of preamble bytes. The tone lasts for a time given by the RAM register PRFX_TONE_DURATION. In order to get a smooth transition from tone to preamble, it is recommended to set PRFX_TONE_DURATION as given in Table 25-15.

Tone transmission is allowed to coincide with the synthesizer stabilizing (this may, however, cause the start of the tone to have larger frequency variations than the packet). For this reason, when PRF_PKT_CONF.START_TONE is 1, the synthesizer startup time is reduced by the value of the register PRFX_TONE_OFFSET. This should normally correspond to the time of the extra preambles, but it must not be larger than 4096 (corresponding to 128 µs). PRFX_TONE_OFFSET can thus be used to compensate for the extra time added by the extra preamble bytes used for tone generation. However, the duration of the extra preamble bytes configured must be accounted for in PRF_TX_DELAY, PRF_RETRANS_DELAY, and PRF_RX_TX_TIME.

The default values of PRFX_TONE_DURATION and PRFX_TONE_OFFSET correspond to 48 µs and are tuned for using 12 extra preamble bytes (13 in total) on 2 Mbps. When using the reset values, MDMCTRL2.NUM_PREAM_BYTES should thus be set to 0x0C.

If PRFX_TONE_DURATION is set too large compared to the number of preamble bytes configured, the modulator underflows. If this happens, the task ends with TASKERR_MODUNF as end cause.

Table 25-15. Recommended RAM Register Settings for Start Tone

Data Rate	PRFX_TONE_DURATION	PRFX_TONE_OFFSET
2 Mbps	MDMCTRL2.NUM_PREAM_BYTES × 0x80 + 0x4A	MDMCTRL2.NUM_PREAM_BYTES × 0x80
1 Mbps	MDMCTRL2.NUM_PREAM_BYTES × 0x100 + 0x52	MDMCTRL2.NUM_PREAM_BYTES × 0x100
500 kbps	MDMCTRL2.NUM_PREAM_BYTES × 0x200 + 0x62	min(MDMCTRL2.NUM_PREAM_BYTES × 0x200, 0x1000)
250 kbps	MDMCTRL2.NUM_PREAM_BYTES × 0x400 + 0x82	min(MDMCTRL2.NUM_PREAM_BYTES × 0x400, 0x1000)

25.9.2.3 Receive Task

When a CMD_RX command is received, the LLE configures the radio on the channel given by PRF_CHAN.FREQ and starts listening for a sync word.

The LLE can set up an internal time-out for the sync search in the PRF_SEARCH_TIME register. If this register is non-zero and no sync has been obtained in the number of 32 MHz cycles given by this register, the task ends with a TASK_NOSYNC end cause. Note that the value of this register must be large enough to have time for the duration of the sync word and 1 preamble byte, in addition to some margin, in order to get sync. The task can also be set up to end on Timer 2 event 2, based on PRF_TASK_CONF.STOP_CONF. If this bit field is 11, the Timer 2 event 2 time-out applies only during the first sync search after a CMD_RX command has been issued if PRF_TASK_CONF.START_CONF is 0. In this case, the time-out in PRF_SEARCH_TIME does not apply to the first sync search, but it still applies to subsequent sync searches in the same task. If PRF_TASK_CONF.STOP_CONF is 11 and PRF_TASK_CONF.START_CONF is 1, the time-out applies to every sync search and PRF_SEARCH_TIME never applies, but the Timer 2 event 2 timeout does not apply after sync is obtained or while waiting for Timer 2 event 1 to restart listening. If sync is obtained, the LLE starts reading the packet.

If sync is found on a packet, the time of sync is captured by the Timer 2 capture function (see Section 22.1.10).

25.9.2.3.1 Basic Mode

This section describes the receive operation if PRF_TASK_CONF.MODE is 00 or 01.

If PRF_TASK_CONF.MODE is 01, the length byte is read first. It gives the number of bytes between the length byte and the CRC, including the address. If the length is too small to contain the address, the reception of the packet is stopped and the device goes back to sync search (regardless of the setting in PRF_TASK_CONF.REPEAT).

Next, the address byte is read if PRF_PKT_CONF.ADDR_LEN is 1. It is compared against PRF_ADDR_ENTRYn.ADDRESS for the values of *n* where the entry is enabled for the received sync word. If there is a matching entry, this entry is used when receiving the packet; otherwise, reception is stopped and the device goes back to sync search. If PRF_PKT_CONF.ADDR_LEN is 0, the first entry that is enabled for the received sync word is used. If PRF_TASK_CONF.MODE is 00, the packet length is then read from PRF_ADDR_ENTRYn.RXLENGTH. This length includes the address, so it must be greater than or equal to the number of address bytes. If PRF_TASK_CONF.MODE is 01, the received length byte is compared against PRF_ADDR_ENTRYn.RXLENGTH. If it is greater than that value, reception is stopped and the device goes back to sync search.

If reception is stopped due to address mismatch or invalid length, the time-out given by PRF_SEARCH_TIME or Timer 2 event 2 still applies. If the first packet of the task is being received and PRF_TASK_CONF.STOP_CONF is 11, the next packet still counts as the first packet.

If a CRC field is present, it is checked using the polynomial configured in the BSP and the initialization value from PRF_CRC_INIT. The result of the CRC check is written in the MSB of the RES byte in the status field if a status field is configured. If the CRC is not correct and PRF_FIFO_CONF.AUTOFLUSH_CRC is 1, the LLE sends a discard RX FIFO command to remove the packet from the RX FIFO.

A packet where the length is equal to the address size contains no payload. Such a packet is known as an empty packet. If PRF_FIFO_CONF.AUTOFLUSH_EMPTY is 1 and an empty packet is received, the LLE sends a discard RX FIFO command to remove the empty packet from the RX FIFO.

Note that if the CRC length is 1 byte, the CRC check is not correct for empty packets if fixed length is configured or no address bytes are used.

If the RX FIFO becomes full while receiving a packet, the packet is discarded from the FIFO and no more bytes are stored in the RX FIFO, but the packet is received to its end. After that, it is checked whether the packet would be discarded from the RX FIFO anyway due to the setting of PRF_FIFO_CONF. If so, the task proceeds as normally. Otherwise, an RXFIFOFULL error interrupt is raised in lieu of the normal interrupt for received packets.

After receiving a packet, the LLE raises an interrupt to the MCU. Depending on CRC result and whether the packet was empty, the interrupts are generated as shown in Table 25-16, provided an RXFIFOFULL interrupt is not raised as described previously. The table also shows which of the counters among the RAM registers are to be updated.

Table 25-16. Interrupt and Counter Operation for Received Messages

CRC Result	Payload Length > Address Length	Counter Incremented	Interrupt Raised
OK	No	PRF_ADDR_ENTRYn.N_RXOK	RXEMPTY
OK	Yes	PRF_ADDR_ENTRYn.N_RXOK	RXOK
NOK	No	PRF_ADDR_ENTRYn.N_RXNOK	RXNOK
NOK	Yes	PRF_ADDR_ENTRYn.N_RXNOK	RXNOK

An address entry should not be modified while the receiver is running. In order to modify, stop the receiver, modify the entry or entries, and restart the receiver.

25.9.2.3.2 Auto Mode

This section describes the receive operation if PRF_TASK_CONF.MODE is 10 or 11.

If PRF_PKT_CONF.ADDR_LEN is 1, the address byte is compared against PRF_ADDR_ENTRYn.ADDRESS, where *n* ranges from 0 to 7. It is compared against PRF_ADDR_ENTRYn.ADDRESS for the values of *n* where the entry is enabled for the received sync word. If there is a matching entry, this entry is used when receiving the packet, otherwise reception is stopped and the device goes back to sync search. If PRF_PKT_CONF.ADDR_LEN is 0, the first entry that is enabled for the received sync word is used.

Next, the 9-bit or 10-bit header is read. If PRF_ADDR_ENTRYn.CONF.VARLEN is 1, the length is fetched from the header and compared against PRF_ADDR_ENTRYn.RXLENGTH. If it is greater than that value, reception is stopped and the device goes back to sync search. If PRF_ADDR_ENTRYn.CONF.VARLEN is 0, the length field in the received header is ignored and the packet length is read from PRF_ADDR_ENTRYn.RXLENGTH. In both cases, the length is the number of bytes after the header and before the CRC. The length must be less than or equal to 63 for a 9-bit header and 127 for a 10-bit header. When a 10-bit header is used, the MCU must ensure that an entire packet can fit in the RX FIFO for auto ACK to be possible. This limits the maximum packet size based on the settings in PRF_FIFO_CONF.

If reception is stopped due to address mismatch or invalid length, the time-out given by PRF_SEARCH_TIME or Timer 2 event 2 still applies. If the first packet of the task is being received and PRF_TASK_CONF.STOP_CONF is 11, this still counts as the first packet.

If a CRC field is present, it is checked using the polynomial configured in the BSP and the initialization value from PRF_CRC_INIT. The result of the CRC is written in the MSB of the RES byte in the status field if a status field is configured. If the CRC is not correct and PRF_FIFO_CONF.AUTOFLUSH_CRC is 1, the LLE sends a discard RX FIFO command to remove the packet from the RX FIFO.

If the CRC is correct, the sequence number is checked against the sequence number stored in PRF_ADDR_ENTRYn.SEQSTAT.SEQ. If the sequence numbers are equal and PRF_ADDR_ENTRYn.SEQSTAT.VALID is 1, the two last received CRC bytes are compared against the 2 bytes in PRF_ADDR_ENTRYn.LASTCRC. If they are equal, the packet is determined to be a retransmission which can be ignored. If the CRC is 1 byte only, the received CRC byte is compared to PRF_ADDR_ENTRYn.LASTCRC[0] only, and if there is no CRC, the comparison is always viewed as equal. If the packet was a retransmission, the IGN bit of the RES byte in the status field is set if a status field is configured. After reception of a packet with CRC OK and which fits in the RX FIFO, PRF_ADDR_ENTRYn.SEQSTAT.VALID is set to 1, PRF_ADDR_ENTRYn.SEQSTAT.SEQ is set to the sequence number of the header of the received packet, and PRF_ADDR_ENTRYn.LASTCRC is set to the value of the last two received CRC bytes.

If the RX FIFO becomes full while receiving a packet, the packet is discarded from the FIFO and no more bytes are stored in the RX FIFO, but the packet is received to its end. After that, it is checked whether the packet would be discarded from the RX FIFO anyway due to the setting of PRF_FIFO_CONF. If so, the task proceeds as normally. Otherwise, an RXFIFOFULL error interrupt is raised, and no acknowledgment is transmitted. The sequence number is not updated so that a retransmission of the packet is not ignored.

If the received packet was not a retransmission and PRF_ADDR_ENTRYn.SEQSTAT.ACK_PAYLOAD_SENT is 1, the packet is seen as a confirmation of the last transmitted acknowledgment payload. If so, PRF_ADDR_ENTRYn.SEQSTAT.ACK_PAYLOAD_SENT is set to 0, a TXDONE interrupt is raised, and the PRF_ADDR_ENTRYn.NTXDONE counter is incremented. PRF_ADDR_ENTRYn.ACKLENGTHk is set to 0 for the *k* found in PRF_ADDR_ENTRYn.SEQSTAT.NEXTACK, and PRF_ADDR_ENTRYn.SEQSTAT.NEXTACK is inverted.

After receiving a packet, the LLE raises an interrupt to the MCU. Depending on the CRC result, the payload length, and whether the received packet is a retransmission to be ignored, the interrupts are generated as shown in Table 25-17. The table also shows which of the counters among the RAM registers are to be updated.

Table 25-17. Interrupt and Counter Operation for Received Messages

CRC Result	Ignore	Length	Counter Incremented	Interrupt Raised
OK	No	> 0	PRF_ADDR_ENTRYn.N_RXOK	RXOK
OK	No	= 0	PRF_ADDR_ENTRYn.N_RXOK	RXEMPTY
OK	Yes	X	PRF_ADDR_ENTRYn.N_RXIGNORED	RXIGNORED
NOK	X	X	PRF_ADDR_ENTRYn.N_RXNOK	RXNOK

After reception of a packet, the next action is determined as follows:

- If the CRC of the received packet was not correct, the treatment of the packet is finished and the next action is as described in Section 25.9.2.3.3.
- If PRF_ADDR_ENTRYn.CONF.AA is 0, the treatment of the packet is finished and the next action is as described in Section 25.9.2.3.3.
- If the NO_ACK bit of the received header is 1 and the CRC was correct, the treatment of the packet is finished and the next action is as described in Section 25.9.2.3.3.
- If the packet did not fit in the RX FIFO and was not otherwise to be discarded, the treatment of the packet is finished and the next action is as described in Section 25.9.2.3.3.
- Otherwise, an acknowledgment is transmitted as described in the following text.

After receiving a packet where the CRC is correct and where an acknowledgment is supposed to be sent, the transmitter is configured. The transmission starts at a time given by the PRF_RX_TX_TIME register. Synthesizer recalibration is performed only if there is time. The LLE checks PRF_ADDR_ENTRYn.SEQSTAT.NEXTACK to find *k*. If PRF_ADDR_ENTRYn.ACKLENGTHk is nonzero, payload is included the packet. In this case, PRF_ADDR_ENTRYn.SEQSTAT.ACK_PAYLOAD_SENT is set to 1 by the LLE. The transmitted packet has the same sync word and address as the received packet. If PRF_ADDR_ENTRYn.CONF.TXLEN is 0, the length field in the header is set equal to PRF_ADDR_ENTRYn.ACKLENGTHk. If PRF_ADDR_ENTRYn.CONF.TXLEN is 1, the length field is set to 11 0011 for a 9-bit header and to 011 0011 for a 10-bit header. Note that a value of 0 for PRF_ADDR_ENTRYn.CONF.TXLEN may be used regardless of the VARLEN setting in the peer device, as the length field is ignored for fixed length. A value of 1 must only be used if the peer is configured to use fixed length for the ACK payload, and should only be used with ACKs without payload. The sequence number is set to the value of PRF_ADDR_ENTRYn.SEQSTAT.ACKSEQ, and NO_ACK is set to 0. If there is payload, it is read from the buffer as described in Section 25.8.3.

After the acknowledgement has been transmitted, PRF_ADDR_ENTRYn.SEQSTAT.ACKSEQ is incremented modulo 4, the PRF_N_TX counter is incremented, and the next action is as described in Section 25.9.2.3.3.

25.9.2.3.3 Continuation and Ending of Receive Tasks

When a task ends, a TASKDONE interrupt is raised and an end cause is then available in PRF_ENDCAUSE.

After a packet has been received and potentially an acknowledgment has been transmitted, the next action depends on PRF_TASK_CONF.REPEAT. If this value is 0, the task ends. In this case, the PRF_ENDCAUSE register is set to TASK_ENDOK.

If PRF_TASK_CONF.REPEAT is 1, reception restarts. If PRF_TASK_CONF.START_CONF is 1, the LLE behaves as if the task was started again, with the LLE waiting for Timer 2 event 1; then starting to listen. If PRF_TASK_CONF.START_CONF is 0, the receiver restarts as soon as possible, as starting a new task (except for the behavior of Timer 2 event 2 if PRF_TASK_CONF.STOP_CONF is 11). In both cases, synthesizer recalibration is done if PRF_TASK_CONF.REPEAT_CONF is 0, otherwise not. Skipping synthesizer recalibration reduces the time before listening is restarted.

If a CMD_SHUTDOWN or a command starting a new task is observed while the task is running, it ends immediately with TASK_ABORT as the end cause. If the receiver or transmitter was running, an RXTXABO interrupt is also raised.

If CMD_STOP is received while in sync search, the task ends immediately with TASK_RXTIMEOUT as the end cause. If CMD_STOP is received while receiving or while transmitting an ACK or in the transition between those, the task ends with TASK_STOP as the end cause after the packet is fully received and (if ACK is to be sent) the ACK is sent. If CMD_STOP is received while waiting for Timer 2 event 1 to restart reception, the task ends immediately with TASK_STOP as the end cause

If Timer 2 event 2 (either from Timer 2 or from CMD_SEND_EVENT2) is observed during the task, the behavior depends on PRF_TASK_CONF.STOP_CONF:

- 00: Nothing happens
- 01: Behaves as if a CMD_STOP was received
- 10: Behaves as if a CMD_SHUTDOWN was received

- 11: If received while in sync search for the first packet after the task was started, or if PRF_TASK_CONF.START_CONF is 1 while in sync search for any packet, the task ends immediately with TASK_RXTIMEOUT as the end cause. Otherwise, nothing happens.

In addition, the task can end due to an internal time-out as described in the beginning of Section 25.9.2.1, or it can end due to an error condition. The full list of possible end causes is summarized in Table 25-18.

Table 25-18. End-of-Receive Tasks

Condition	End-of-Task Cause	Comment
Received packet (and potentially sent ACK) with PRF_TASK_CONF.REPEAT = 0	TASK_ENDOK	
Received packet (and potentially sent ACK) with PRF_TASK_CONF.REPEAT = 1 after having observed CMD_STOP or Timer 2 event 2 with PRF_TASK_CONF.STOP_CONF = 01 and PRF_TASK_CONF.REPEAT = 1	TASK_STOP	
While in sync search, observed CMD_STOP or Timer 2 event 2 with PRF_TASK_CONF.STOP_CONF = 01	TASK_RXTIMEOUT	
Observed Timer 2 event 2 while in sync search of the first packet with PRF_TASK_CONF.STOP_CONF = 11	TASK_RXTIMEOUT	
Did not get sync in the time specified by PRF_SEARCH_TIME	TASK_NOSYNC	
Received command for starting new task or CMD_SHUTDOWN or observed Timer 2 event 2 with PRF_TASK_CONF.STOP_CONF = 10	TASK_ABORT	If transmitter was running or receiver was running and had obtained sync, an RXTXABO interrupt is also raised.
Received unknown command	TASKERR_CMD	LLEERR interrupt is also raised. If transmitter was running or receiver was running and had obtained sync, an RXTXABO interrupt is also raised.
Semaphore is not free when expected	TASKERR_SEM	Task ends without any radio operation. LLEERR interrupt is also raised.
Unpermitted value of RAM register	TASKERR_PAR	LLEERR interrupt is also raised.
For PRF_TASK_CONF.MODE = 00: PRF_ADDR_ENTRYn.RXLENGTH corresponding to the received address is smaller than address length	TASKERR_PAR	LLEERR interrupt is also raised.
For auto mode: PRF_ADDR_ENTRYn.ACKLENGTGHm for the ACK payload to be transmitted exceeded 32	TASKERR_PAR	LLEERR interrupt is also raised.

25.9.2.4 Transmit Task

When a CMD_TX command is received, the LLE configures the radio on the channel given by PRF_CHAN.FREQ and starts transmitting the packet from the TX FIFO.

If the TX FIFO has no available data, the task ends with TASKERR_TXFIFO as the end cause. Otherwise, the number of bytes given by the length byte in the TX FIFO is read from the TX FIFO and transmitted or otherwise handled as described in following sections. No check of data availability is done after the length byte is read, so if the FIFO contains fewer bytes than indicated in the length field, a TX FIFO underflow interrupt is raised by the FIFO hardware.

25.9.2.4.1 Basic Mode

This section describes the transmit operation if PRF_TASK_CONF.MODE is 00 or 01.

If PRF_TASK_CONF.MODE is 01, the length field is calculated from the length field in the FIFO and transmitted. It is up to the MCU to ensure that the calculated length field does not exceed 255. If PRF_TASK_CONF.MODE is 00, no length field is transmitted.

If an address is configured, it is found based on the setting in PRF_FIFO_CONF.TX_ADDR_CONF. It can be set to take the address from PRF_ADDR_ENTRY0, to read it from the TX FIFO (which for the transmitter is equivalent to not having an address configured), or to read an index *n* from the config byte in the FIFO and read the address from PRF_ADDR_ENTRYn.ADDRESS. The values of ENA0 and ENA1 in PRF_ADDR_ENTRYn.CONF are ignored for the transmitter; the primary sync word is always transmitted.

The payload (if any) is transmitted as given in the FIFO.

If configured, a CRC with the number of bytes given by PRF_CRC_LEN is transmitted at the end.

When a packet has been transmitted, the LLE sends a deallocate TX FIFO command if PRF_ADDR_ENTRYn.REUSE is 0. Otherwise, the MCU must issue either a deallocate TX FIFO (to send a new packet) or a retry TX FIFO (to reuse) before sending again. The PRF_N_TX counter is incremented. A TXDONE interrupt to the MCU is raised when the packet has been completely read out of the TX FIFO by the LLE. Note that due to modulator delay, CRC transmission and ramp-down, this will happen before the packet transmission is finished. The next action is as given in Section 25.9.2.4.3.

25.9.2.4.2 Auto Mode

This section describes the transmit operation if PRF_TASK_CONF.MODE is 10 or 11.

When a 10-bit header is used, the MCU must ensure that an entire packet can fit in the TX FIFO for auto retransmission to be possible. This limits the maximum packet size based on the settings in PRF_FIFO_CONF.

If an address is configured, it is the first byte transmitted. It is found based on the setting in PRF_FIFO_CONF.TX_ADDR_CONF. It can be set to take the address from PRF_ADDR_ENTRY0, to read it from the TX FIFO, or to read an index *n* from the config byte in the FIFO and read the address from PRF_ADDR_ENTRYn.ADDRESS. In other cases, *n* is always assumed to be 0 in the following text. The values of ENA0 and ENA1 in PRF_ADDR_ENTRYn.CONF are ignored for the transmitter; the primary sync word is always transmitted.

The 9-bit or 10-bit header is transmitted next. If PRF_ADDR_ENTRYn.CONF.TXLEN is 0, the length field is set to the number of payload bytes after the header, which is calculated from the length byte in the TX FIFO. If PRF_ADDR_ENTRYn.CONF.TXLEN is 1, the length field is set to 11 0011 for a 9-bit header and to 011 0011 for a 10-bit header. Note that a value of 0 for PRF_ADDR_ENTRYn.CONF.TXLEN may be used regardless of the VARLEN setting in the receiver, as a receiver configured to use fixed length ignores the length field. A value of 1 must only be used if the receiver is configured to use fixed length. The NO_ACK bit transmitted is set according to bit 5 of the config byte read from the TX FIFO if present, otherwise to 0. If PRF_ADDR_ENTRYn.CONF.FIXEDSEQ is 1, the SEQ bits transmitted are set equal to bits 6 and 7 of the config byte read from the FIFO. Otherwise, the SEQ bits are set to the value of PRF_ADDR_ENTRYn.SEQSTAT.SEQ.

The payload (if any) is transmitted as given in the FIFO.

If configured, a CRC with the number of bytes given by PRF_CRC_LEN is transmitted at the end.

When a packet has been transmitted, the N_TX counter is incremented.

After transmission of a packet, the action depends on PRF_ADDR_ENTRYn.CONF.AA and the NO_ACK bit in the transmitted header. If PRF_ADDR_ENTRYn.CONF.AA = 0 or NO_ACK = 1, no acknowledgment is expected, and the action is as if a valid acknowledgment had been received.

If PRF_ADDR_ENTRYn.CONF.AA is 1 and the transmitted NO_ACK bit was 0, the LLE configures RX to listen for an acknowledgment. To listen for acknowledgment, the receiver is configured at a time given by the PRF_TX_RX_TIME register. Synthesizer recalibration is performed only if there is time. The unit looks for sync. The sync search times out at the time given by PRF_SEARCH_TIME. If sync is found, the packet is received into the RX FIFO. If PRF_PKT_CONF.ADDR_LEN is 1, the address byte is compared against PRF_ADDR_ENTRYn.ADDRESS for the *n* that was used in transmission. If not matching, reception is stopped.

Next, the 9-bit or 10-bit header is read. If PRF_ADDR_ENTRYn.CONF.VARLEN is 1, the length is fetched from the header and compared against PRF_ADDR_ENTRYn.RXLENGTH. The maximum allowed value of this register is 32. If the received length is greater than PRF_ADDR_ENTRYn.RXLENGTH, reception is stopped and the device goes back to sync search. If PRF_ADDR_ENTRYn.CONF.VARLEN is 0, the length field in the received header is ignored and the packet length is read from PRF_ADDR_ENTRYn.RXLENGTH, which should normally be 0 in this case. The length is the number of bytes after the header and before the CRC.

If a CRC field is present, it is checked using the polynomial configured in the BSP and the initialization value from PRF_CRC_INIT. The result of the CRC is written in the MSB of the RES byte in the status field if a status field is configured. If the CRC is not correct and PRF_FIFO_CONF.AUTOFLUSH_CRC is set, the LLE sends a discard RX FIFO command to remove the packet from the RX FIFO.

The received sequence number is written to the config byte of the RX FIFO if configured, but is otherwise ignored.

If the RX FIFO becomes full while receiving an acknowledgment packet, the packet is discarded from the FIFO and no more bytes are stored in the RX FIFO, but the packet is received to its end. After that, it is checked to see whether the packet would be discarded from the RX FIFO anyway due to the setting of PRF_FIFO_CONF. If so, the task proceeds as normally. Otherwise, the task ends after the packet is received and an RXFIFOFULL error interrupt is raised. In this case, the treatment of the packet is as if the acknowledgment were not successfully received. This means that the next time a transmit task is started, the packet is retransmitted so that the receiver retransmits the ACK payload.

After receiving an acknowledgment, the LLE raises an interrupt to the MCU. Depending on the CRC result, the payload length, and whether the received packet had the same sequence number as the transmitted one, the interrupts are generated as shown in Table 25-19. It also shows which of the counters among the RAM registers are to be updated.

Table 25-19. Interrupt and Counter Operation for Received ACK Packets

CRC Result	Length	Counter Incremented	Interrupt Raised
OK	> 0	PRF_ADDR_ENTRYn.N_RXOK	RXOK
OK	= 0	PRF_ADDR_ENTRYn.N_RXIGNORED	RXEMPTY
NOK	X	PRF_ADDR_ENTRYn.N_RXNOK	RXNOK

If an acknowledgment was not received (because no sync was obtained in time, the address did not match, the sequence number was wrong, the CRC check failed, or the ACK did not fit in the RX FIFO and was not otherwise to be discarded) the LLE sends a retry TX FIFO command. If the number of retransmissions already performed (not including the original transmission) is equal to PRF_RETRANS_CNT, the task ends. Otherwise, the packet is retransmitted. The time from the end of the previous transmission to the start of the retransmission is given in units of 62.5 ns by PRF_RETRANS_DELAY.

If the received packet was a valid acknowledgment, or if a packet was completely read out of the TX FIFO and no acknowledgment was expected, the LLE sends a deallocate TX FIFO command if PRF_ADDR_ENTRYn.REUSE is 0. Otherwise, the MCU must issue either a deallocate TX FIFO (to send a new packet) or a retry TX FIFO (to reuse) before sending again. The PRF_ADDR_ENTRYn.NTXDONE counter is incremented. A TXDONE interrupt to the MCU is raised. If PRF_ADDR_ENTRYn.CONF.FIXEDSEQ = 0, PRF_ADDR_ENTRYn.SEQSTAT.SEQ is incremented by 1 modulo 4. The next action is as given in Section 25.9.2.3.3.

If the task ends because of a maximum number of retransmissions, a retry TX FIFO command is sent before the task ends, and PRF_ADDR_ENTRYn.SEQSTAT.SEQ is not incremented. This means that by default, the packet retransmission is attempted in the next task. If this is not desired, the packet must be removed from the FIFO. This can be done either by issuing a CMD_TXFIFO_RESET (this also removes any subsequent packets in the TX FIFO), by reading out the packet using the RFTXFRD register and issuing a CMD_TX_FIFO_DEALLOC command, or by TX FIFO pointer manipulation (Section 25.3.1.3). PRF_ADDR_ENTRYn.SEQSTAT.SEQ should then be incremented by one. These operations should only take place between tasks (that is, while the LLE does not have SEMAPHORE1).

25.9.2.4.3 Continuation and Ending of Transmit Tasks

When a task ends, a TASKDONE interrupt is raised and an end cause is then available in PRF_ENDCAUSE.

After a packet has been transmitted and potentially a valid acknowledgment has been received, the next action depends on PRF_TASK_CONF.REPEAT. If this value is 0, the task ends. In this case, the PRF_ENDCAUSE register is set to TASK_ENDOK.

If PRF_TASK_CONF.REPEAT is 1, the TX FIFO status is checked. If the TX FIFO has no available data, the task ends with TASK_ENDOK as the end cause. Otherwise, transmission restarts. If PRF_TASK_CONF.START_CONF is 1, it behaves as if the task was started again with the LLE waiting for Timer 2 event 1, then performing a synthesizer calibration and starting to transmit. If PRF_TASK_CONF.START_CONF is 0, the transmitter restarts PRF_TX_DELAY after the end of the previously transmitted packet, with synthesizer recalibration only if there is enough time, but in other respects as starting a new task. The PRF_TX_DELAY register gives the wait time in units of 62.5 ns. If the value is too small to fulfill, the transmission starts as soon as possible.

If a CMD_SHUTDOWN or a command starting a new task is observed while the task is running, it ends immediately with TASK_ABORT as the end cause. If the transmitter or receiver was running, an RXTXABO interrupt is also raised.

If CMD_STOP is received while transmitting a packet, waiting for transmission of another packet, waiting for an ACK, receiving an ACK, or in the transition between those, the task ends with TASK_STOP as the end cause after the packet is fully transmitted and (if ACK is expected) the ACK is received or given up. If CMD_STOP is received while waiting for Timer 2 event 1 to restart reception, the task ends immediately with TASK_STOP as the end cause.

If Timer 2 event 2 (either from Timer 2 or from CMD_SEND_EVENT2) is observed during the task, the behavior depends on PRF_TASK_CONF.STOP_CONF.

- 00: Nothing happens.
- 01: Behaves as if a CMD_STOP was received
- 10: Behaves as if a CMD_SHUTDOWN was received
- 11: Nothing happens.

In addition, the task can end for reasons described earlier, or it can end due to an error condition. The full list of possible end causes is summarized in Table 25-20.

Table 25-20. End-of-Transmit Tasks

Condition	End-of-Task Cause	Comment
Transmitted packet (and potentially received ACK) with PRF_TASK_CONF.REPEAT = 0	TASK_ENDOK	
Transmitted packet (and potentially received ACK) and observed TX FIFO with no available data	TASK_ENDOK	
Transmitted packet (and potentially received ACK) after having observed CMD_STOP or Timer 2 event 2 with PRF_TASK_CONF.STOP_CONF = 01	TASK_STOP	
Did not get valid acknowledgment after having retransmitted the number of times given by PRF_RETRANS_CNT	TASK_MAXRT	
Observed empty TX FIFO when packet transmission is supposed to start, or TX FIFO is in an invalid state	TASKERR_TXFIFO	LLEERR and RXTXABO interrupts are also raised.
RX FIFO went overfull while receiving an ACK that was not otherwise to be discarded	TASKERR_RXFIFO	RXFIFOFULL interrupt is also raised.
Received command for starting new task or CMD_SHUTDOWN or observed Timer 2 event 2 with PRF_TASK_CONF.STOP_CONF = 10	TASK_ABORT	If transmitter was running, an RXTXABO interrupt is also raised.
Received unknown command	TASKERR_CMD	LLEERR interrupt is also raised. If transmitter or receiver was running and had obtained sync, an RXTXABO interrupt is also raised.

Table 25-20. End-of-Transmit Tasks (continued)

Condition	End-of-Task Cause	Comment
Semaphore is not free when expected	TASKERR_SEM	Task ends without any radio operation. LLEERR interrupt is also raised.
Invalid value of RAM register	TASKERR_PAR	LLEERR interrupt is also raised.
Length field to be transmitted exceeded maximum allowed value (255 in basic mode, 127 or 63 in auto mode with variable length)	TASKERR_PAR	LLEERR interrupt is also raised.

25.9.2.5 Transmit on Clear-Channel Task

When a CMD_TX_ON_CC command is received, the LLE configures the receiver on the channel given by PRF_CHAN.FREQ, but sync search is not enabled. The LLE polls the RSSI register every 5.33 µs and compares it to the value of PRF_RSSI_LIMIT. If a valid RSSI below the value of PRF_RSSI_LIMIT is observed more than PRF_RSSI_COUNT times in a row, the system starts transmitting. From there, the operation is as a normal transmit task, see Section 25.9.2.4, except for the operation after a packet has been transmitted and potentially acknowledged, which is described in Section 25.9.2.5.1.

25.9.2.5.1 Continuation and Ending of Transmit on Clear-Channel Tasks

If PRF_TASK_CONF.TX_ON_CC_CONF is 1, the task ends if a valid RSSI value is not below the limit. If PRF_TASK_CONF.TX_ON_CC_CONF is 0, the device keeps listening until an RSSI below the given value is found.

If PRF_TASK_CONF.STOP_CONF is 11, Timer 2 event 2 may give a time-out while listening for a clear channel the first time, but not after the first transmission has been started.

If PRF_TASK_CONF.TX_ON_CC_CONF is 0, the clear-channel assessment must not be used as the only medium access control scheme in a multiuser environment, as this may cause all the devices to start transmission at the same time.

If retransmission is enabled, the LLE listens for acknowledgment and retransmits if needed as for normal TX tasks. However, if PRF_TASK_CONF.REPEAT_CONF is 0 after applying the retransmit delay, the device returns to listening, performing the same operation as when the task started, before possibly retransmitting.

If PRF_TASK_CONF.REPEAT is 0, the task ends after transmission as for normal TX tasks.

If PRF_TASK_CONF.REPEAT is 1, the TX FIFO status is checked. If the TX FIFO has no available data, the task ends with TASK_ENDOK as the end cause. Otherwise, transmission restarts. If PRF_TASK_CONF.REPEAT_CONF is 0, the task returns to listening, whereas if it is 1, the task restarts as if it were a standard transmit task. If PRF_TASK_CONF.START_CONF is 1, it behaves as if the task was started again with the LLE waiting for Timer 2 event 1, then performing a synthesizer calibration and starting to listen or transmit. If PRF_TASK_CONF.START_CONF is 0, the listening or transmission restarts PRF_TX_DELAY after the end of the previously transmitted packet, with synthesizer recalibration only if there is enough time, but in other respects as starting a new task.

If Timer 2 event 2 (either from Timer 2 or from CMD_SEND_EVENT2) is observed during the task, the behavior depends on PRF_TASK_CONF.STOP_CONF.

- 00: Nothing happens.
- 01: If received while transmitting a packet or waiting for or receiving an ACK or in the transition between those, the task ends with TASK_STOP as the end cause after the packet is fully transmitted and (if ACK is expected) the ACK is received or given up. If received while waiting for Timer 2 event 1 to restart reception, the task ends immediately with TASK_STOP as the end cause.
- 10: Behaves as if a CMD_SHUTDOWN was received
- 11: If received while listening for RSSI below the level before the first packet after the task was started, or if PRF_TASK_CONF.START_CONF is 1 while listening before any packet, the task ends immediately with TASK_NOCC as the end cause. Otherwise, nothing happens.

The task can end for all the same reasons as a normal transmit task summarized in Table 25-20. In addition it can end for the reasons listed in Table 25-21.

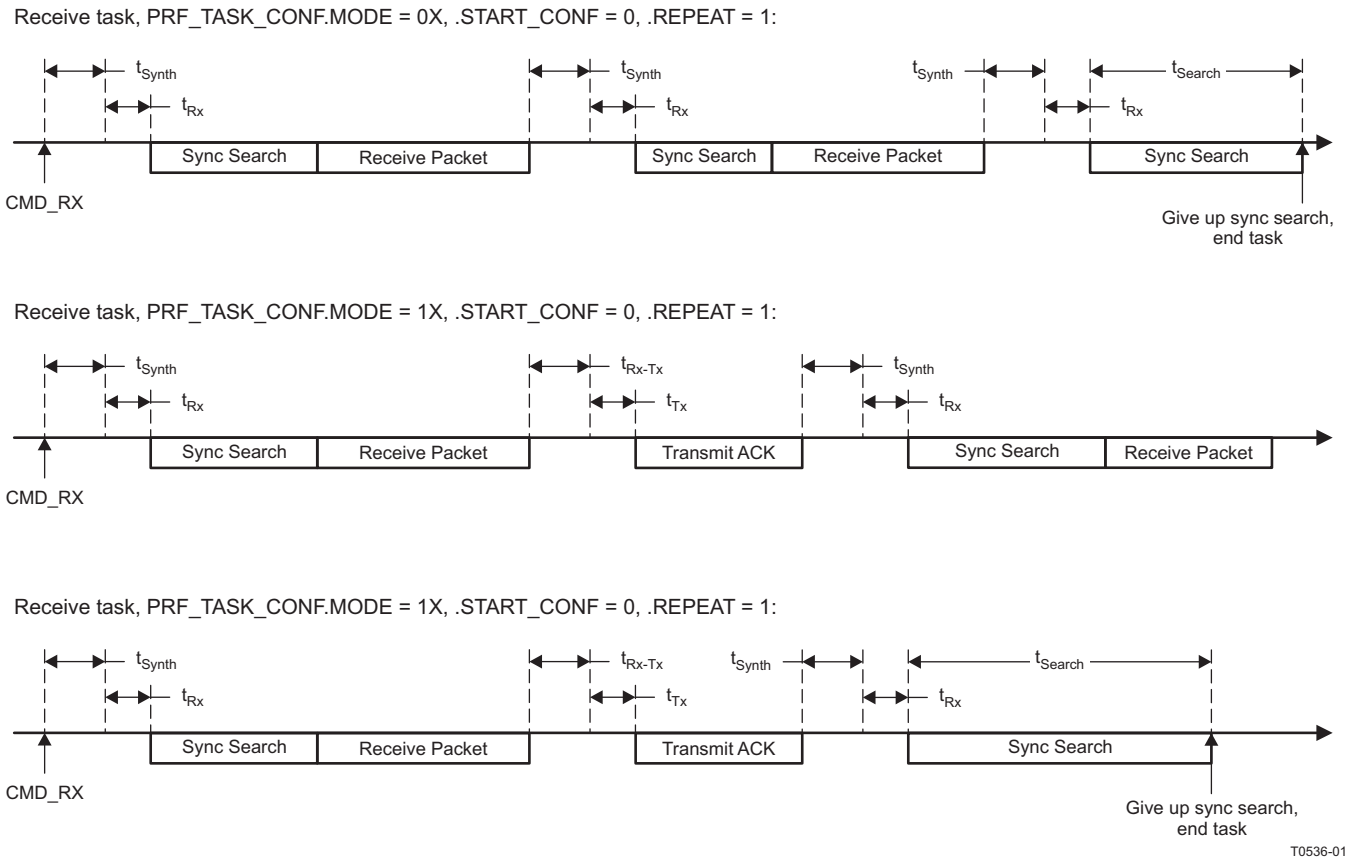
Table 25-21. Additional Reasons for End-of-Transmit on Clear-Channel Tasks

Condition	End-of-Task Cause	Comment
Observed valid RSSI above the threshold with PRF_TASK_CONF.TX_ON_CC_CONF = 1	TASK_NOCC	
Observed CMD_STOP or Timer 2 event 2 with PRF_TASK_CONF.STOP_CONF = 01 or 11 while listening for RSSI below the threshold	TASK_NOCC	

25.9.2.6 Timing

The timing in tasks is given in registers PRF_TX_DELAY, PRF_RETRANS_DELAY, PRF_SEARCH_TIME, PRF_RX_TX_TIME, and PRF_TX_RX_TIME. The first two of these registers are multiplied by 2 and then represent the number of 32-MHz samples, while the rest directly give the number of 32-MHz samples. PRF_TX_DELAY gives the time from the end of the previous transmission in the task to the start of the next transmission. Some examples of these delays are shown in Figure 25-12 and Figure 25-13 for RX and TX tasks, respectively. The time from the end of a received packet to the beginning of a transmitted packet is 130 µs in an RX task with auto ACK.

When sync search takes place, either for receiving a normal packet or for receiving ACK, a time-out can be set up for when to give up the search. This time-out, given in 32-MHz cycles, is set up in the PRF_SEARCH_TIME register. Setting this register to 0 disables the time-out. In case of a time-out, the task ends for a normal sync search, or a packet is retransmitted in case of an ACK sync search.

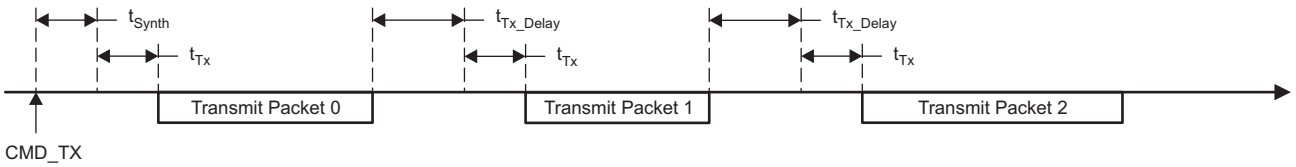


NOTE: The time given by PRF_SEARCH_TIME is denoted t_{Search} and the time given by PRF_RX_TX is denoted t_{RX-TX}. The setup and wait time for the synthesizer, receiver, and transmitter are denoted t_{Synth}, t_{TX}, and t_{RX}, respectively.

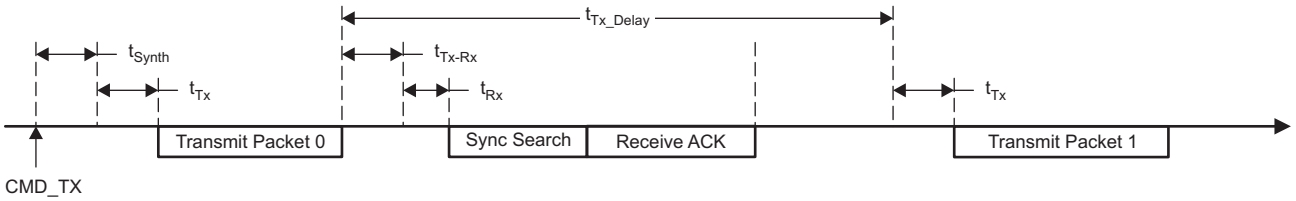
Figure 25-12. Timing of Packets in RX Tasks

For auto retransmit tasks, the time PRF_RETRANS_DELAY is the time from the end of a transmission to the retransmission of the packet in case an ACK is not found or there is a CRC error; see Figure 25-13. The values of PRF_SEARCH_TIME and the maximum packet length in PRF_PAYLOAD_LEN should be set such that this time can always be achieved. If it is not possible to achieve the retransmission time, the packet is retransmitted as early as possible.

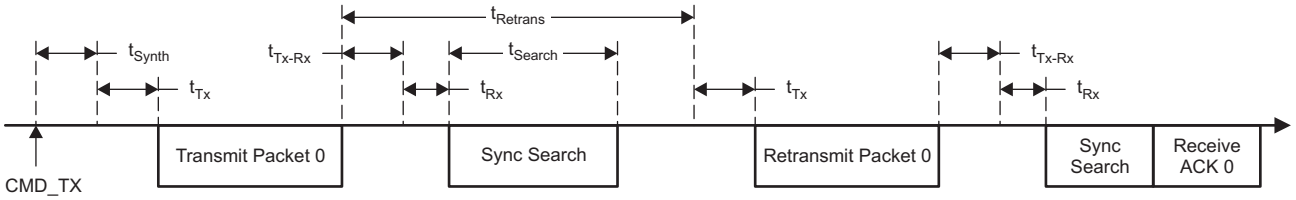
Transmit task, PRF_TASK_CONF.MODE = 0X, .START_CONF = 0, .REPEAT = 1:



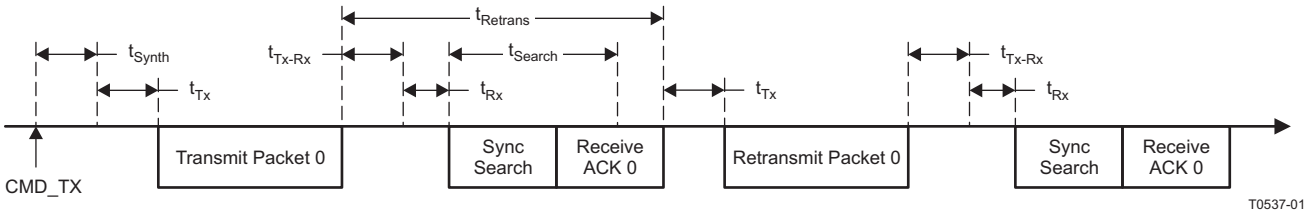
Transmit task, PRF_TASK_CONF.MODE = 0X, .START_CONF = 0, .REPEAT = 1:



Transmit task, PRF_TASK_CONF.MODE = 1X, .START_CONF = 0, .REPEAT = 1,
no ACK found first time:



Transmit task, PRF_TASK_CONF.MODE = 1X, .START_CONF = 0, .REPEAT = 1,
CRC error on ACK:



NOTE: The time given by PRF_TX_DELAY is denoted t_{TX_Delay}, the time given by PRF_SEARCH_TIME is denoted t_{Search}, the time given by PRF_RETRANS_DELAY is denoted $t_{Retrans}$, and the time given by PRF_TX_RX is denoted $t_{TX\text{-}RX}$. The setup and wait times for the synthesizer, receiver, and transmitter are denoted t_{Synth}, t_{TX}, and t_{RX}, respectively.

Figure 25-13. Timing of Packets in TX Tasks

25.9.3 RF Test Commands

In Table 25-12, there are listed some commands for modem test purposes. No registers are programmed by the LLE based on these commands, so all registers must be set up by the MCU. This includes the frequency word, which is otherwise written by the LLE; see Section 25.5 on how to program this. These commands take no parameters, and do not cause the LLE to create any interrupts or to write any end cause.

When receiving CMD_DEMOD_TEST, the LLE starts the receiver, but does not start sync search. The receiver runs until a CMD_SHUTDOWN command (or a command starting another task) is sent.

When receiving CMD_RX_TEST, the LLE starts the receiver and starts sync search. Any received data is discarded. The receiver runs until a CMD_SHUTDOWN command (or a command starting another task) is sent.

When receiving a CMD_TX_TEST, the LLE starts the transmitter and starts sending all zeros. The TX test command is normally combined with configuration of the modem to send a tone or by the BSP to send a whitening sequence. The transmitter runs until a CMD_SHUTDOWN command (or a command starting another task) is sent. In order to send random modulated data for test purposes, it is recommended to set BSP_MODE to 0x03 to enable both whiteners.

In order to send a continuous wave (CW), MDMCTRL0.TX_IF can be set to 1 before the CMD_TX_TEST command is issued. In this case, the radio outputs a tone with the frequency given in MDMTEST1.TX_TONE. In most cases, a tone at the synthesizer frequency is wanted (for example, to measure phase noise), in which case MDMTEST.TX_TONE should be set to 0x0A. The frequency synthesizer must be programmed to the carrier frequency with no offset in this case; see Section 25.5.

When receiving a CMD_TX_FIFO_TEST, the LLE starts the transmitter and starts sending from the TX FIFO; otherwise, the command is as CMD_TX_TEST. The MCU must feed data into the TX FIFO to avoid underflow, and the TX FIFO must be set up with auto commit and auto deallocate.

When receiving a CMD_PING command, the LLE responds with a PINGRSP. This can be used for checking that the LLE code is running.

25.10 Random Number Generation

The CC2541 has a hardware pseudo-random register, as explained in Section 14.1. The RF core register bank provides a second interface to this register. Reading the RFPSRND register is equivalent to reading RNDL, then writing 01 to ADCCON1.RCTL.

For seeding the pseudo-random number generator and for tasks where higher entropy of the random numbers is needed, the radio can be used as a true-random generator. The register RFRND provides access to the least-significant bits of the radio ADC, which is random when noise is received. In order to get values on this register, the receiver must be turned on. The value in RFRND is updated every 0.17 µs, so the sampling of that register must be slower than that in order to get a new value with every sample.

To get true random numbers, the following procedure can be followed:

1. Program FREQCTRL to a channel that is not likely to contain a narrow-band signal. A frequency outside the ISM band, such as a setting of 0, is recommended.
2. Program LNAGAIN to 0 to have minimum reception of a signal on the air.
3. Start the receiver in test mode by issuing a CMD_DEMOD_TEST command.
4. Wait until ADC data are ready. This can be seen by the RFRND register having a value different from 0.
5. Read the number of values needed from RFRND. Make sure that there is at least 0.17 µs between each read (that is, at least 6 cycles if running on 32 MHz). For instance, to seed the pseudo-random generator, two values are needed.
6. Shut down the receiver by issuing a CMD_SHUTDOWN command.

The values read from the RFRND register do not have a perfectly uniform distribution. In order to improve this, several random numbers can be combined to produce one random number. One way of doing this is to use the pseudo-random generator in CRC mode and combine eight numbers into one. An example of how this can be done is given in the C code below:

```
// Store LNA gain setting and set minimum LNA gain
lnagain_stored = LNAGAIN;
LNAGAIN = 0x00;

// Set lowest possible frequency to avoid signals in ISM band
FREQCTRL = 0x00;

// Enable radio in RX without sync search
while (RFST != 0);
```

```
RFST = CMD_DEMOD_TEST;

// Wait for modem to be running
while (RFRND == 0);

// Seed RNG
RNDL = RFRND;
RNDL = RFRND;

for (j=0; j<ARRAY_SZ; j++) {
    // Read 8 random bytes into CRC generation
    RNDH = RFRND;
    RNDH = RFRND;
    RNDH = RFRND;
    RNDH = RFRND;
    RNDH = RFRND;
    RNDH = RFRND;
    RNDH = RFRND;
    RNDH = RFRND;
    // Read out LSB of CRC state
    rndarray[j] = RNDL;
}
            // Shut down radio
while (RFST != 0);
RFST = CMD_SHUTDOWN;
// Restore LNA gain setting
LNAGAIN = lnagain_stored;
```

25.11 Packet Sniffing

Packet sniffing is a nonintrusive way of observing data that is either being transmitted or received. The packet sniffer outputs a clock and a data signal, which should be sampled on the rising edges of the clock. The two packet-sniffer signals are observable as GPIO outputs. For accurate time stamping, the SFD signal should also be output. The packet sniffer does not work for the 2 Mbps data rate.

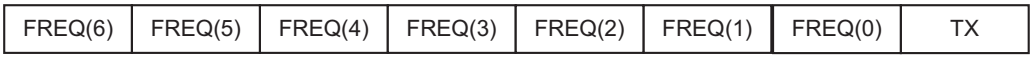
The packet-sniffer mode is selected in register MDMCTRL3.RFC_SNIFF_CTRL; see Table 25-22 for a description of the different modes of operation.

Table 25-22. Packet-Sniffer Modes of Operation

MDMCTRL3.RFC_SNIFF_CTRL	Description
00	Packet sniffer disabled
01	Data output from BSP. TX data, including CRC, is whitened if the whitener is enabled. RX data, including CRC, is always de-whitened.
10	Data output from modulator. Only TX data before whitening is output. Any CRC bytes are 0.
11	Data output from the demodulator. Only RX data before de-whitening is output.

The packet-sniffer clock frequency is equal to the RF data rate. The data is output serially, in the received or transmitted order. It is possible to use a SPI slave to receive the data stream.

When a complete packet has been transferred, the packet sniffer appends a status byte that tells which value of the FREQCTRL register was used and if it was a TX or RX packet (bit 0 is high if it was a TX packet). The appended byte is formatted as follows (first transmitted bit to the left):



R0015-01

Figure 25-14. Complete Appended Packet

This allows for the external receiver to differentiate between RX and TX packets.

To set up the packet-sniffer signals or some of the other RF core observation outputs (in total maximum 3; rfc_obs_sig0, rfc_obs_sig1, and rfc_obs_sig2), the user must perform the following steps:

Step1: Determine which signal (RFC_OBS_CTRL[0–2]) to output on which GPIO pin (P1[0:5]). This is done using the OBSSELx control registers (OBSSEL0–OBSSEL5) that control the observation output to the pins P1[0:5] (overriding the standard GPIO behavior for those pins).

Step2: Set the (RFC_OBS_CTRL[0–2]) control registers to select the correct signals (rfc_obs_sig); for example, for packet sniffing one needs the rfc_sniff_data for the packet-sniffer data signal.and rfc_sniff_clk for the corresponding clock signal.

Step3: Enable the packet-sniffer module in the MDMCTRL3 register.

25.12 Registers

25.12.1 Register Overview

25.12.1.1 SFR Registers

1 - RFIRQF0 (0xE9) RF interrupt flags

2 - RFIRQF1 (0x91) RF interrupt flags

3 - RFERRF (0xBF) RF error interrupt flags

4 - RFD (0xD9) RF data

5 - RFST (0x6189) LLE and FIFO commands

25.12.1.2 XREG Registers

Table 25-23. XREG Register Overview

Address (Hex)	+ 0x0000	+ 0x001	+ 0x002	+ 0x003
0x6180	FRMCTRL0	RFIRQM0	RFIRQM1	RFERRM
0x6184	FREQCTRL	FREQTUNE	TXPOWER	TXCTRL
0x6188	LLESTAT		SEMAPHORE0	SEMAPHORE1
0x618C	SEMAPHORE2	RFSTAT	RSSI	RFPSRND
0x6190	MDMCTRL0	MDMCTRL1	MDMCTRL2	MDMCTRL3
0x6194	SW_CONF	SW0	SW1	SW2
0x6198	SW3	FREQEST	RXCTRL	FSCTRL
0x619C				
0x61A0	LNAGAIN	AAFGAIN	ADCTEST0	
0x61A4		MDMTEST0	MDMTEST1	
0x61A8		ATEST		
0x61AC			RFC_OBS_CTRL0	RFC_OBS_CTRL1
0x61B0	RFC_OBS_CTRL2	LLECTRL		
0x61BC	TXFILTCFG			RFRND
0x61C0	RFRAMCFG			RFFDMA0
0x61C4	RFFDMA1	RFFSTATUS	RFFCFG	
0x61C8	RFRXFLEN	RFRXFTHRS	RFRXFWR	RFRXFRD
0x61CC	RFRXFWP	RFRXFRP	RFRXFSWP	RFRXFSRP
0x61D0	RFTXFLEN	RFTXFTHRS	RFTXFWR	RFTXFRD
0x61D4	RFTXFWP	RFTXFRP	RFTXFSWP	RFTXFSRP

Table 25-23. XREG Register Overview (continued)

Address (Hex)	+ 0x0000	+ 0x001	+ 0x002	+ 0x003
0x61E0	BSP_P0	BSP_P1	BSP_P2	BSP_P3
0x61E4	BSP_D0	BSP_D1	BSP_D2	BSP_D3
0x61E8	BSP_W	BSP_MODE	BSP_DATA	
0x61F8	SW4	SW5	SW6	SW7
0x61FC	DC_I_L	DC_I_H	DC_Q_L	DC_Q_H

25.12.2 Register Settings Update

This section contains a summary of the register settings that should be updated from their default value to have optimal performance. For some of the registers, the setting depends on the required gain in the receiver chain for bit rates of 1 Mbps and lower. For 2 Mbps, other values are needed, and different values should be used for RX and TX tasks. Note that registers that are listed only in the 2-Mbps table should have their default values when operating at 1 Mbps and lower, and vice versa.

Table 25-24. Registers That Should Be Updated From Their Default Value, Bit Rates 1 Mbps and Lower

Register Name	Address (Hex)	Standard Gain: New Value (Hex)(1)	High Gain: New Value (Hex)	Description
FRMCTRL0	6180	43	43	Amplitude weight in frequency offset compensation (assuming sync word included in CRC and MSB first)
MDMCTRL1	6191	48	48	Sync word correlation threshold (32-bit sync word)
MDMCTRL2	6192	C0	C0	Use inverse of preamble for frequency offset estimation (assuming MSB first)
MDMCTRL3	6193	63	63	Set RSSI mode to peak detect after sync
RXCTRL	619A	33	3F	Receiver currents
FSCTRL	619B	55	5A	Prescaler and mixer currents
LNAGAIN	61A0	3A	7F	LNA gain
TXFILTCFG	61BC	03	03	Sets TX anti-aliasing filter to appropriate bandwidth
TXPOWER	6186	E1	E1	TX power (0 dBm)
TXCTRL	6187	19	19	DAC current
IVCTRL	6265	1B	1B	PA, mixer, and DAC bias

(1) Not all modulation types are characterized for the standard gain setting; see the *CC2541 2.4-GHz* Bluetooth *low energy and Proprietary System-on-Chip* data sheet (SWRS110).

Table 25-25. Registers That Should Be Updated From Their Default Value, Bit Rate 2 Mbps

Register Name	Address (Hex)	RX Tasks	TX Tasks	Description
FRMCTRL0	6180	43	43	Amplitude weight in frequency offset compensation (assuming sync word is included in CRC and MSB first)
MDMCTRL1	6191	48	48	Sync word correlation threshold (32-bit sync word)
MDMCTRL2	6192	CC	CC	Use inverse of preamble for frequency offset estimation (assuming MSB first); set extra preamble bytes
MDMCTRL3	6193	63	63	Set RSSI mode to peak detect after sync
RXCTRL	619A	29	29	Receiver currents

Table 25-25. Registers That Should Be Updated From Their Default Value, Bit Rate 2 Mbps (continued)

Register Name	Address (Hex)	RX Tasks	TX Tasks	Description
FSCTRL	619B	5A	5A	Prescaler and mixer currents
ADCTEST0	61A2	66	66	Reduce ADC gain
MDMTEST0	61A5	0F	01	Select dc offset compensation method; change RSSI averaging
TXFILTCFG	61BC	03	03	Transmit filter bandwidth
PRF_PKT_CONF	6003	06	06	Enable AGC and start tone
PRF_RADIO_CONF	607E	90	D0	Set 1-MHz TX IF and dc write-back; for TX tasks also special dc offset compensation
TXPOWER	6186	E1	E1	TX power (0 dBm)
TXCTRL	6187	19	19	DAC current
IVCTRL	6265	1B	1B	PA, mixer, and DAC bias

The values for FRMCTRL0, MDMCTRL2, and PRF_PKT_CONF may require further adjustment based on the frame format, and the correlation threshold in MDMCTRL1 should be adjusted according to the sync word length, see Section 25.7.

In addition to these modifications, registers must be set in order to set up the modulation format, packet handling, and so forth, as explained throughout this chapter.

25.12.3 SFR Register Descriptions

RFIRQF0 (0xE9) – RF Interrupt Flags

Bit No.	Name	Reset	R/W	Description
7:4	–	0000	R0	Reserved
3	RXTHSHUP	0	R/W0	RX FIFO goes above its upper threshold. 0: No interrupt pending 1: Interrupt pending
2	TXTHSHUP	0	R/W0	TX FIFO goes above its upper threshold. 0: No interrupt pending 1: Interrupt pending
1	RXTHSHDN	0	R/W0	RX FIFO goes below its lower threshold. 0: No interrupt pending 1: Interrupt pending
0	TXTHSHDN	0	R/W0	TX FIFO goes below its lower threshold. 0: No interrupt pending 1: Interrupt pending

RFIRQF1 (0x91) – RF Interrupt Flags

Bit No.	Name	Reset	R/W	Description
7	PINGRSP	0	R/W0	When receiving a CMD_PING command, the LLE responds with a PINGRSP. This can be used for checking that the LLE is running. 0: No interrupt pending 1: Interrupt pending
6	TASKDONE	0	R/W0	TX FIFO packet completed 0: No interrupt pending 1: Interrupt pending
5	TXDONE	0	R/W0	Task ended 0: No interrupt pending 1: Interrupt pending
4	RXEMPTY	0	R/W0	Empty packet received 0: No interrupt pending 1: Interrupt pending
3	RXIGNORED	0	R/W0	Packet received with unexpected sequence number 0: No interrupt pending 1: Interrupt pending
2	RXNOK	0	R/W0	Packet received with CRC error 0: No interrupt pending 1: Interrupt pending
1	TXFLUSHED	0	R/W0	TX ACK buffer flushed 0: No interrupt pending 1: Interrupt pending
0	RXOK	0	R/W0	Packet received correctly 0: No interrupt pending 1: Interrupt pending

RFERRF (0xBF) – RF Error Interrupt Flags

Bit No.	Name	Reset	R/W	Description
7	–	0	R/W0	Reserved
6	RXFIFOFULL	0	R/W0	RX FIFO is full when trying to store received data 0: No interrupt pending 1: Interrupt pending
5	LLEERR	0	R/W0	LLE command or parameter error 0: No interrupt pending 1: Interrupt pending
4	RXTXABO	0	R/W0	Receive or transmit operation aborted 0: No interrupt pending 1: Interrupt pending
3	RXOVERF	0	R/W0	RX FIFO overflow 0: No interrupt pending 1: Interrupt pending
2	TXOVERF	0	R/W0	TX FIFO overflow 0: No interrupt pending 1: Interrupt pending
1	RXUNDERF	0	R/W0	RX FIFO underflow 0: No interrupt pending 1: Interrupt pending
0	TXUNDERF	0	R/W0	TX FIFO underflow 0: No interrupt pending 1: Interrupt pending

RFD (0xD9) – RF data

Bit No.	Name	Reset	R/W	Description
7:0	RFD[7:0]	0x00	R/W	Data written to the register is written to the TX FIFO. When reading this register, data from the RX FIFO is read.

RFST (0xE1) – LLE and FIFO commands

Bit No.	Name	Reset	R/W	Description
7:0	RFST[7:0]	0x00	R/WH0	Commands to radio are written to this register. The register is cleared (set to 0x00) when the radio is ready for a new command.

25.12.3.1 XREG Register Descriptions

FRMCTRL0 (0x6180) – Frame Control

Bit No.	Name	Reset	R/W	Description
7:5	FOC_MAGN_CONT[2:0]	000	R/W	Controls how signal amplitude is weighted into the frequency offset compensation scheme. 000: Magnitude has no effect 001 to 111: Low-to-high weighting of the magnitude
4:2	–	000	R/W	Reserved always write 0.
1	SW_CRC_MODE	0	R/W	0: The sync word is not included in the CRC calculation. 1: The sync word is included in the CRC calculation. Only to be used with whitening disabled.
0	ENDIANNESS	0	R/W	0: The data goes LSB-first over the air. 1: The data goes MSB-first over the air.

RFIRQM0 (0x6181) – RF Interrupt Masks

Bit No.	Name	Reset	R/W	Description
7:4	–	0000	R0	Reserved
3	RXTHSHUP	0	R/W	RX FIFO goes above its upper threshold. 0: Interrupt disabled 1: Interrupt enabled
2	TXTHSHUP	0	R/W	TX FIFO goes above its upper threshold. 0: Interrupt disabled 1: Interrupt enabled
1	RXTHSHDN	0	R/W	RX FIFO goes below its lower threshold. 0: Interrupt disabled 1: Interrupt enabled
0	TXTHSHDN	0	R/W	TX FIFO goes below its lower threshold. 0: Interrupt disabled 1: Interrupt enabled

RFIRQM1 (0x6182) – RF Interrupt Masks

Bit No.	Name	Reset	R/W	Description
7	PINGRSP	0	R/W	When receiving a CMD_PING command, the LLE responds with a PINGRSP. This can be used for checking that the LLE is running. 0: Interrupt disabled 1: Interrupt enabled
6	TASKDONE	0	R/W	TX FIFO packet completed 0: Interrupt disabled 1: Interrupt enabled
5	TXDONE	0	R/W	Task ended 0: Interrupt disabled 1: Interrupt enabled
4	RXEMPTY	0	R/W	Empty packet received 0: Interrupt disabled 1: Interrupt enabled
3	RXIGNORED	0	R/W	Packet received with unexpected sequence number 0: Interrupt disabled 1: Interrupt enabled
2	RXNOK	0	R/W	Packet received with CRC error 0: Interrupt disabled 1: Interrupt enabled
1	TXFLUSHED	0	R/W	TX ACK buffer flushed 0: Interrupt disabled 1: Interrupt enabled
0	RXOK	0	R/W	Packet received correctly 0: Interrupt disabled 1: Interrupt enabled

RFERRM (0x6183) – RF Error Interrupt Masks

Bit No.	Name	Reset	R/W	Description
7	–	0	R/W	Reserved
6	RXFIFOFULL	0	R/W	RX FIFO is full when trying to store received data. 0: Interrupt disabled 1: Interrupt enabled
5	LLEERR	0	R/W	LLE command or parameter error 0: Interrupt disabled 1: Interrupt enabled
4	RXTXABO	0	R/W	Receive or transmit operation aborted 0: Interrupt disabled 1: Interrupt enabled
3	RXOVERF	0	R/W	RX FIFO overflow 0: Interrupt disabled 1: Interrupt enabled
2	TXOVERF	0	R/W	TX FIFO overflow 0: Interrupt disabled 1: Interrupt enabled
1	RXUNDERF	0	R/W	RX FIFO underflow 0: Interrupt disabled 1: Interrupt enabled
0	TXUNDERF	0	R/W	TX FIFO underflow 0: Interrupt disabled 1: Interrupt enabled

FREQCTRL (0x6184) – Synthesizer Frequency Control

Bit No.	Name	Reset	R/W	Description
7	–	0	R0	Reserved. Read as zero
6:0	FREQ[6:0]	0x16	R/W	Frequency control word. Controls frequency of local oscillator. See Section 25.5 for the relation between the LO frequency and the RF frequency. f_{LO} = (2379 + FREQ[6:0]) MHz The frequency word in FREQ[6:0] is an offset value from 2379 MHz. The device supports the frequency range from 2379 MHz to 2495 MHz. The usable settings for FREQ[6:0] are consequently 0 to 116. Settings outside this range (117–127) give a frequency of 2495 MHz.

FREQTUNE (0x6185) – Crystal Oscillator Frequency Tuning

Bit No.	Name	Reset	R/W	Description
7:4	–	0x0	R0	Reserved. Read as zero
3:0	XOSC32M_TUNE[3:0]	0xF	R/W	Tune crystal oscillator. The default setting of 1111 leaves the XOSC not tuned. Changing setting from default switches in extra capacitance to the oscillator, effectively lowering the XOSC frequency. Hence, a higher setting gives a higher frequency.

TXPOWER (0x6186) – Output Power Control

Bit No.	Name	Reset	R/W	Description
7:0	PA_POWER[7:0]	0xF5	R/W	PA power control. NOTE: This value should be updated. See the device data sheet (Appendix C) for recommended values.

TXCTRL (0x6187) – TX Settings

Bit No.	Name	Reset	R/W	Description
7	–	0	R0	Reserved
6	–	1	R/W	Reserved
5:4	DAC_CURR[1:0]	10	R/W	Change the current in the DAC to change the current steps
3:2	DAC_DC[1:0]	10	R/W	Adjusts the dc level to the TX mixer.
1:0	TXMIX_CURRENT[1:0]	01	R/W	Transmit mixers core current: Current increases with increasing setting.

LLESTAT (0x6188) – LLE Status

Bit No.	Name	Reset	R/W	Description
7:5	–	000	R0	Reserved
4	AGC_LOWGAIN	0	R	1 if the AGC algorithm has reduced the front-end gain; 0 otherwise
3	WAIT_T2E1	0	R	Indication on the LLE waiting for Timer 2 event 1 to start a task 0: Not waiting for Timer 2 event 1 1: Command processed, event 1 not yet received
2	LLE_IDLE	0	R	Link-layer engine idle 0: The LLE is busy processing or finishing a command, or in reset. 1: The LLE is idle waiting for a command to start a new task.
1	SYNC_SEARCH	0	R	RX search for sync 0: The modem is not ready to receive a packet 1: The modem is in search for a sync word or is receiving a packet
0	VCO_ON	0	R	VCO on 0: The VCO is powered down, so the next RX or TX operation must start and calibrate the synthesizer before transmitting or receiving 1: The VCO is powered up. If the LLE is idle, it means the next task starts quickly if frequency programming is disabled (PRF_CHAN.FREQ = 127)

SEMAPHORE0 (0x618A) – Semaphore for Accessing RF Data Memory

Bit No.	Name	Reset	R/W	Description
7:1	–	0000 000	R0	Reserved, read as 0
0	SEMAPHORE	1	R/W1	When SEMAPHORE = 1 and SEMAPHORE0 is read, SEMAPHORE is set to 0. SEMAPHORE can only be set to 1 by a reset or by writing 1 to it.

SEMAPHORE1 (0x618B) – Semaphore for Accessing RF Data Memory

Bit No.	Name	Reset	R/W	Description
7:1	–	0000 000	R0	Reserved, read as 0
0	SEMAPHORE	1	R/W1	When SEMAPHORE = 1 and SEMAPHORE1 is read, SEMAPHORE is set to 0. SEMAPHORE can only be set to 1 by a reset or by writing 1 to it.

SEMAPHORE2 (0x618C) – Semaphore

Bit No.	Name	Reset	R/W	Description
7:1	–	0000 000	R0	Reserved, read as 0
0	SEMAPHORE	1	R/W1	When SEMAPHORE = 1 and SEMAPHORE2 is read, SEMAPHORE is set to 0. SEMAPHORE can only be set to 1 by a reset or by writing 1 to it.

RFSTAT (0x618D) – RF Core Status

Bit No.	Name	Reset	R/W	Description
7	MOD_UNDERFLOW	0	R/W0	Modulator has underflowed. Must be cleared by software
6:5	DEM_STATUS	00	R	Demodulator status 00: Idle 01: Active 10: Finishing 11: Error
4	SFD	0	R	High when the sync word has been sent in TX or when sync has been obtained in RX
3	CAL_RUNNING	0	R	Frequency synthesizer calibration status. 0: Calibration done or not started 1: Calibration in progress
2	LOCK_STATUS	0	R	1 when PLL is in lock; 0 otherwise
1	TX_ACTIVE	0	R	Status signal, active when the LLE is in one of the transmit states
0	RX_ACTIVE	0	R	Status signal, active when the LLE is in one of the receive states

RSSI (0x618E) – Received Signal Strength Indicator

Bit No.	Name	Reset	R/W	Description
7:0	RSSI_VAL[7:0]	0x80	R	RSSI estimate on a logarithmic scale. Unit is 1 dB; offset depends on the gain of the RX chain, including external components; see the device data sheet. The reset value of 0x80 also indicates that the RSSI value is invalid or the measurement is not yet complete.

RFPSRND (0x618F) – Pseudorandom Number Generator

Bit No.	Name	Reset	R/W	Description
7:0	RNG_DOUT[7:0]	0x00	R	The value read from the pseudorandom number generator, see Chapter 14. Reading this register causes the shift register to be updated with 13 times rollout.

MDMCTRL0 (0x6190) – Modem Configuration

Bit No.	Name	Reset	R/W	Description
7:6	FOC_DECAY[1:0]	00	R/W	Controls decay ratio of frequency offset compensation mechanism. Value by which to increment preamble cost function at each decay 00: 8 01: 16 10: 32 11: 64
5	TX_IF	0	R/W	0: Modulation is done at an IF set by rfr_tx_tone. 1: Modulator outputs tone set by rfr_tx_tone.
4:1	MODULATION[3:0]	0010	R/W	Modulation scheme 0010: GFSK 250-kHz deviation, 1-Mbps data rate 0011: GFSK 500-kHz deviation, 2-Mbps data rate 0100: GFSK 160-kHz deviation, 250-kbps data rate 0110: GFSK 160-kHz deviation, 1-Mbps data rate 0111: GFSK 320-kHz deviation, 2-Mbps data rate 1000: MSK, 250-kbps data rate 1001: MSK, 500-kbps data rate Others: Reserved
0	PHASE_INVERT	0	R/W	Set one of two RF modulation modes for RX or TX 0: Normal (binary 0 represented with negative frequency deviation, binary 1 represented with positive frequency deviation) 1: Inverted phase (binary 0 represented with positive frequency deviation, binary 1 represented with negative frequency deviation)

MDMCTRL1 (0x6191) – Modem Configuration

Bit No.	Name	Reset	R/W	Description
7:6	FOC_MODE	01	R/W	Frequency-offset average filter behavior 00: No frequency-offset compensation done 01: Freeze frequency-offset estimate after sync 10: Continuously estimate and remove frequency offset 11: Freeze the frequency-offset estimate after sync, use double decay rate
5	–	0	R0	Reserved
4:0	CORR_THR[4:0]	0 1111	R/W	Demodulator correlator threshold value, used in sync search. Optimal threshold value depends on SW_CONF.SW_LEN. CORR_THR adjusts how the receiver synchronizes to data from the radio. If threshold is set too low, sync can more easily be found on noise. If set too high, the sensitivity is reduced but sync is not likely to be found on noise.

MDMCTRL2 (0x6192) – Modem Configuration

Bit No.	Name	Reset	R/W	Description
7	SW_BIT_ORDER	0	R/W	0: The sync word is transmitted LSB to MSB (from SYNC_WORD[0] to SYNC_WORD[31]), and in receive the correlator expects this bit ordering. 1: The sync word is transmitted MSB to LSB (from SYNC_WORD[31] to SYNC_WORD[0]), and in receive the correlator expects this bit ordering.
6	DEM_PREAM_MODE	0	R/W	Use PREAM_SEL[1:0] or 1s complement of PREAM_SEL[1:0] for frequency offset estimation. 0: PREAM_SEL[1:0] 1: 1s complement of PREAM_SEL[1:0]
5:4	PREAM_SEL[1:0]	00	R/W	00: Select preamble based on first bit of sync word; last bit of preamble is inverse of first bit of sync word. 01: Select preamble based on first bit of sync word; last bit of preamble is same as first bit of sync word. 10: Use preamble 0101 0101 11: Use preamble 1010 1010
3:0	NUM_PREAM_BYTES[3:0]	0000	R/W	The number of preamble bytes to be sent in TX mode prior to the sync word 0000: 1 leading preamble byte 0001: 2 leading preamble bytes 0010: 3 leading preamble bytes 0011: 4 leading preamble bytes … 1111: 16 leading preamble bytes

MDMCTRL3 (0x6193) – Modem Configuration

Bit No.	Name	Reset	R/W	Description
7:6	SYNC_MODE[1:0]	01	R/W	00: Correlation value above threshold is sufficient as sync criterion. 01: Correlation value above threshold and data decision on all symbols of sync word is used as sync criterion. 10: Correlation value above threshold and data decision on all symbols of sync word is used as sync criterion. Accept one bit error in sync word 11: Reserved
5	RAMP_AMP	1	R/W	1: Enable ramping of DAC output amplitude during start-up and finish. 0: Disable ramping of DAC output amplitude.
4:3	RFC_SNIFF_CTRL[1:0]	00	R/W	Enable and disable *rfc_sniff*. 00: Sniffer disabled 01: Output data out of the BSP 10: Output data out of the modulator before the BSP 11: Output data out of the demodulator before the BSP
2	–	0	R0	Reserved. Read as 0.
1:0	RSSI_MODE[1:0]	00	R/W	Controls mode of RSSI 00 : Continuous mode 01 : Freeze estimate at sync 10 : Peak detect 11 : Continuous before sync, peak detect after sync

SW_CONF (0x6194) – Sync Word Configuration

Bit No.	Name	Reset	R/W	Description
7	DUAL_RX	0	R/W	0: Only search for primary SW 1: Search for both primary and secondary SW
6	–	0	R/W	Reserved. Always write 0.
5	SW_RX	0	R	0: Primary SW received 1: Secondary SW received Valid only when RFSTAT.SFD is 1
4:0	SW_LEN[4:0]	0 0000	R/W	Determines how many of the bits in SYNC_WORD are to be used. This allows for arbitrary sync word lengths. 0 0000: 32-bit SW 0 0001 to 0 11111: Reserved 1 0000: 16-bit SW 1 0001: 17-bit SW 1 0010: 18-bit SW 1 0011: 19-bit SW … 1 1111: 31-bit SW

SW0 (0x6195) – Primary Sync Word Byte 0

Bit No.	Name	Reset	R/W	Description
7:0	SYNC_WORD[7:0]	0x00	R/W	Contains bits 7:0 of the primary synchronization word

SW1 (0x6196) – Primary Sync Word Byte 1

Bit No.	Name	Reset	R/W	Description
7:0	SYNC_WORD[15:8]	0x00	R/W	Contains bits 15:8 of the primary synchronization word

SW2 (0x6197) – Primary Sync Word Byte 2

Bit No.	Name	Reset	R/W	Description
7:0	SYNC_WORD[23:16]	0x00	R/W	Contains bits 23:16 of the primary synchronization word

SW3 (0x6198) – Primary Sync Word Byte 3

Bit No.	Name	Reset	R/W	Description
7:0	SYNC_WORD[31:24]	0x00	R/W	Contains bits 31:24 of the primary synchronization word

SW4 (0x61F8) – Secondary Sync Word Byte 0

Bit No.	Name	Reset	R/W	Description
7:0	SYNC_WORD2[7:0]	0x00	R/W	Contains bits 7:0 of the secondary synchronization word

SW5 (0x61F9) – Secondary Sync Word Byte 1

Bit No.	Name	Reset	R/W	Description
7:0	SYNC_WORD2[15:8]	0x00	R/W	Contains bits 15:8 of the secondary synchronization word

SW6 (0x61FA) – Secondary Sync Word Byte 2

Bit No.	Name	Reset	R/W	Description
7:0	SYNC_WORD2[23:16]	0x00	R/W	Contains bits 23:16 of the secondary synchronization word

SW7 (0x61FB) – Secondary Sync Word Byte 3

Bit No.	Name	Reset	R/W	Description
7:0	SYNC_WORD2[31:24]	0x00	R/W	Contains bits 31:24 of the secondary synchronization word

FREQEST (0x6199) – Estimated RF Frequency Offset

Bit No.	Name	Reset	R/W	Description
7:0	FREQEST[7:0]	0x00	R	Signed value. Contains an estimate of the frequency offset between carrier and the receiver frequency. FOC_MODE controls when this estimate is updated.

RXCTRL (0x619A) – Receive Section Tuning

Bit No.	Name	Reset	R/W	Description
7:6	–	00	R0	Reserved
5:4	GBIAS_LNA2_REF[1:0]	10	R/W	Adjusts front-end LNA2/mixer PTAT current output (M = GBIAS_LNA2_REF[1:0] + 3), default: M = 5.
3:2	GBIAS_LNA_REF[1:0]	10	R/W	Adjusts front-end LNA PTAT current output (M = GBIAS_LNA_REF[1:0] + 3), default: M = 5.
1:0	MIX_CURRENT[1:0]	01	R/W	Control of the output-current consumption of the receiver mixers. The current increases with increasing setting.

FSCTRL (0x619B) – Frequency Synthesizer Tuning

Bit No.	Name	Reset	R/W	Description
7:6	PRE_CURRENT [1:0]	01	R/W	Prescaler current setting
5:4	LODIV_BUF_CURRENT_TX [1:0]	01	R/W	Adjusts current in mixer and PA buffers (lodiv_buf_current). Used when lle_tx_active = 1
3:2	LODIV_BUF_CURRENT_RX [1:0]	01	R/W	Adjusts current in mixer and PA buffers (lodiv_buf_current). Used when lle_tx_active = 0
1:0	LODIV_CURRENT [1:0]	01	R/W	Adjusts divider currents, except mixer and PA buffers.

LNAGAIN (0x61A0) – LNA Gain Setting

Bit No.	Name	Reset	R/W	Description
7	–	0	R0	Reserved, read as 0
6:5	LNA1_CURRENT[1:0]	11	R/W	Gain setting, LNA1 00: 0-dB gain (reference level) 01: 3-dB gain 10: Reserved 11: 6-dB gain
4:2	LNA2_CURRENT[2:0]	111	R/W	Gain setting, LNA2 000: 0-dB gain (reference level) 001: 3-dB gain 010: 6-dB gain 011: 9-dB gain 100: 12-dB gain 101: 15-dB gain 110: 18-dB gain 111: 21-dB gain
1:0	LNA3_CURRENT[1:0]	11	R/W	Gain setting, LNA3 00: 0-dB gain (reference level) 01: 3-dB gain 10: 6-dB gain 11: 9-dB gain

AAFGAIN (0x61A1) – AAF Gain Setting

Bit No.	Name	Reset	R/W	Description
7:2	–	0000 00	R0	Reserved. Read as zero
1:0	AAF_GAIN[1:0]	11	R/W	Controls attenuation in AAF 00: 9-dB attenuation in AAF 01: 6-dB attenuation in AAF 10: 3-dB attenuation in AAF 11: 0-dB attenuation in AAF (reference level)

ADCTEST0 (0x61A2) – ADC Tuning

Bit No.	Name	Reset	R/W	Description
7:0	ADC_ADJ[7:0]	0x10	R/W	Adjust ADC gain

MDMTEST0 (0x61A5) – Modem Configuration

Bit No.	Name	Reset	R/W	Description
7:5	RSSI_ACC[2:0]	101	R/W	RSSI accuracy 000: 5.33-µs average window 001: Mean of two 5.33-µs average windows 010: Reserved 011: Mean of four 5.33-µs average windows 100: 21.3-µs average window 101: Mean of two 21.3-µs average windows 110: Reserved 111: Mean of four 21.3-µs average windows
4	–	0	R/W	Reserved, always write 0.
3:2	DC_BLOCK_LENGTH[1:0]	00	R/W	Controls the number of samples to be accumulated between each dump of the accumulate-and-dump filter used in dc removal. 00: 16 samples 01: 32 samples 10: 64 samples 11: 128 samples
1:0	DC_BLOCK_MODE[1:0]	01	R/W	Selects the mode of operation: 00 : Manual override mode 01 : Enable dc cancellation. Normal operation 10 : Freeze estimates of dc when sync is found. Start estimating dc again when searching for the next frame. 11 : Delayed dc offset estimate used. Delay set by MDMTEST1.DC_DELAY. Until the first estimate is ready, the manual override value is used.

MDMTEST1 (0x61A6) – Modem Configuration

Bit No.	Name	Reset	R/W	Description
7:6	DC_DELAY	00	R/W	Controls delay of dc estimate delayed dc block mode. Delay unit is set by MDMTEST0.DC_BLOCK_LENGTH 00: 5 delays 01: 6 delays 10: 7 delays 11: 8 delays
5	RX_IF	0	R/W	Controls mixer frequency in demodulator (not 2 Mbps) 0: 1 MHz 1: –1 MHz For 2 Mbps, always write 0. The receiver then operates at zero IF.
4:0	TX_TONE[4:0]	0 0000	R/W	Controls baseband frequency of transmission Note: If MDMCTRL0.PHASE_INVERT is 1, the sign of the frequency is inverted 0: –8 MHz 1: –6 MHz 2: –4 MHz 3: –3 MHz 4: –2 MHz 5: –1 MHz 6: –500 kHz 7: –250 kHz 8: –125 kHz 9: –4 kHz 10: 0 Hz 11: 4 kHz 12: 125 kHz 13: 250 kHz 14: 500 kHz 15: 1 MHz 16: 2 MHz 17: 3 MHz 18: 4 MHz 19: 6 MHz 20: 8 MHz

ATEST (0x61A9) – Analog Test Control

Bit No.	Name	Reset	R/W	Description
7:6	–	00	R0	Reserved. Read as zero
5:0	ATEST_CTRL[5:0]	00 0000	R/W	Controls the analog test mode: 00 0000: Disabled 00 0001: Enables the temperature sensor (see also the TR0 register description in #IMPLIED).Other values reserved.

RFC_OBS_CTRL0 (0x61AE) – RF Observation Mux Control 0

Bit No.	Name	Reset	R/W	Description
7	–	0	R0	Reserved. Read as 0
6	RFC_OBS_POL0	0	R/W	The signal chosen by RFC_OBS_MUX0 is XORed with this bit
5:0	RFC_OBS_MUX0	00 0000	R/W	Controls which observable signal from rf_core is to be muxed out to rfc_obs_sigs(0); see Section 7.9. 00 0111: rfc_sniff_data – Data from packet sniffer, see Section 25.11 00 1000: rfc_sniff_clk – Clock for packet sniffer data, see Section 25.11 00 1001: tx_active 00 1010: rx_active 00 1011: vco_on – VCO on Low: The VCO is powered down, so the next RX or TX operation must start and calibrate the synthesizer before transmitting or receiving. High: The VCO is powered up. If the LLE is idle, it means the next task starts quickly if frequency programming is disabled (PRF_CHAN.FREQ = 127). 00 1100: sync_search – RX search for sync Low: The modem is not ready to receive a packet. High: The modem is in search for a sync word or receiving a packet. 00 1101: lle_idle – Link-layer engine idle Low: The LLE is busy processing or finishing a command, or in reset. High: The LLE is idle waiting for a command to start a new task. 00 1110: wait_t2e1 – Indication on the LLE waiting for Timer 2 event 1 to start a task Low: Not waiting for Timer 2 event 1 High: Command processed, event 1 not yet received 00 1111: agc_lowgain – High if the AGC algorithm has reduced the front-end gain; low otherwise 01 0011: fsc_lock – High when PLL is in lock; low otherwise 01 1011: pa_pd - Power amplifier power-down signal 10 1100: lnamix_pd - Low-noise amplifier power-down signal 11 0000: dem_sync_found - High when demodulator has detected a sync word. Stays high until end of packet. 11 0001: mod_sync_sent - High when modulator has sent a sync word. Stays high until end of packet. Others: Reserved

RFC_OBS_CTRL1 (0x61AF) – RF Observation Mux Control 1

Bit No.	Name	Reset	R/W	Description
7	–	0	R0	Reserved. Read as 0
6	RFC_OBS_POL1	0	R/W	The signal chosen by RFC_OBS_MUX1 is XORed with this bit.
5:0	RFC_OBS_MUX1	00 0000	R/W	Controls which observable signal from rf_core is to be muxed out to rfc_obs_sigs(1). See description of RFC_OBS_CTRL0.

RFC_OBS_CTRL2 (0x61B0) – RF Observation Mux Control 2

Bit No.	Name	Reset	R/W	Description
7	–	0	R0	Reseved. Read as 0
6	RFC_OBS_POL2	0	R/W	The signal chosen by RFC_OBS_MUX2 is XORed with this bit.
5:0	RFC_OBS_MUX2	00 0000	R/W	Controls which observable signal from rf_core is to be muxed out to rfc_obs_sigs(2). See description of RFC_OBS_CTRL0.

LLECTRL (0x61B1) – LLE Control

Bit No.	Name	Reset	R/W	Description
7:3	–	0000 0	R0	Reseved. Read as 0
2:1	LLE_MODE_SEL	00	R/W	LLE mode. Changing this field has no effect unless LLE_EN is changed from 0 to 1. 00: Proprietary mode (described in this chapter) 01: BLE mode (only for use by the BLE stack) Others: Reserved
0	LLE_EN	0	R/W	Must be set to 0 before entering PM2 or PM3, otherwise the behavior of the RF core after waking up may be unpredictable. 0: LLE held in reset 1: LLE enabled

TXFILTCFG (0x61BC) – TX Filter Configuration

Bit No.	Name	Reset	R/W	Description
7:4	–	0000	R0	Reserved
3:2	–	11	R/W	Reserved
1:0	FC	11	R/W	Sets TX anti-aliasing filter to appropriate bandwidth. Reduces spurious emissions close to signal. For the best value to use, see Table 25-24 and Table 25-25.

RFRND (0x61BF) – Random Data

Bit No.	Name	Reset	R/W	Description
7:0	RND	0x00	R	Random bits, provided analog part is in random number generation mode (receiver running without sync)

RFRAMCFG (0x61C0) – Radio RAM Configuration

Bit No.	Name	Reset	R/W	Description
7:3	–	0000 1	R	Reserved
2:0	PRE	000	R/W	Selects active memory page for RF core data memory

RFFDMA0, (0x61C3) – Radio DMA Trigger 0 Control

Bit No.	Name	Reset	R/W	Description
7:5	–	000	R	Reserved
4:0	DMA0	0 0000	R/W	Generate a pulse on radio DMA trigger 0 (DMA trigger 19) when: 0 0000: Never 0 0001: A byte is read from RX FIFO and more bytes remain or when a byte arrives in RX FIFO and it was previously empty. 0 0010: A byte is written to RX FIFO and there is available space left or when there becomes available space when the RX FIFO was full. 0 0011: RX FIFO is empty. 0 0100: RX FIFO is full. 0 0101: RX FIFO length equals RFRXFTHRS after a write to RX FIFO. 0 0110: RX FIFO is read when its size equals RFRXFTHRS. 0 0111: RX FIFO is reset (see Table 25-2). 0 1000: RX FIFO is deallocated (see Table 25-2). 0 1001: RX FIFO is retried (see Table 25-2). 0 1010: RX FIFO is discarded (see Table 25-2). 0 1011: RX FIFO is committed (see Table 25-2). 0 1100–0 1111: Reserved (never) 1 0000: Never 1 0001: A byte is read from TX FIFO and more bytes remain or when a byte arrives in TX FIFO and it was previously empty. 1 0010: A byte is written to TX FIFO and there is available space left or when there becomes available space when the TX FIFO was full. 1 0011: TX FIFO is empty. 1 0100: TX FIFO is full. 1 0101: TX FIFO length equals RFTXFTHRS after a write to TX FIFO. 1 0110: TX FIFO is read when its size equals RFTXFTHRS. 1 0111: TX FIFO is reset (see Table 25-2). 1 1000: TX FIFO is deallocated (see Table 25-2). 1 1001: TX FIFO is retried (see Table 25-2). 1 1010: TX FIFO is discarded (see Table 25-2). 1 1011: TX FIFO is committed (see Table 25-2). 1 1100–1 1111: Reserved (never)

RFFDMA1, (0x61C4) – Radio DMA Trigger 1 Control

Bit No.	Name	Reset	R/W	Description
7:5	–	000	R	Reserved
4:0	DMA1	0 0000	R/W	Condition for generating a pulse on radio DMA trigger 1 (DMA trigger 11). See RFFDMA0 for the list of conditions.

RFFSTATUS (0x61C5) – FIFO Status

Bit No.	Name	Reset	R/W	Description
7	TXAVAIL	0	R	0: No readable data in TX FIFO 1: Readable data present in TX FIFO
6	TXFEMPTY	1	R	0: Data present in TX FIFO 1: TX FIFO is empty
5	TXDTHEX	1	R	0: There is less data in TX FIFO than the threshold amount given by RFTXFTHRS. 1: There is more than or equal amount of data in TX FIFO than the threshold amount given by RFTXFTHRS
4	TXFFULL	0	R	0: TX FIFO has available space 1: TX FIFO is full
3	RXAVAIL	0	R	0: No readable data in RX FIFO 1: Readable data present in RX FIFO
2	RXFEMPTY	1	R	0: Data present in RX FIFO 1: RX FIFO is empty
1	RXDTHEX	1	R	0: There is less data in RX FIFO than the threshold amount given by RFRXFTHRS. 1: There is more than or equal amount of data in RX FIFO than the threshold amount given by RFRXFTHRS
0	RXFFULL	0	R	0: RX FIFO has available space 1: RX FIFO is full

RFFCFG (0x61C6) – FIFO Configuration

Bit No.	Name	Reset	R/W	Description
7:6	–	00	R	Reserved
5	TXAUTOCOMMIT	1	R/W	0: Commit TX FIFO only on command 0x95 1: Always set RFTXSWP = RFTXWP
4	TXFAUTODEALLOC	0	R/W	0: Deallocate TX FIFO only on command 0x92 1: Always set RFTXFSRP = RFTXFRP.
3:2	–	00	R	Reserved
1	RXAUTOCOMMIT	0	R/W	0: Commit RX FIFO only on command 0x85 1: Always set RFRXSWP = RFRXWP
0	RXFAUTODEALLOC	1	R/W	0: Deallocate RX FIFO only on command 0x82 1: Always set RFRXFSRP = RFRXFRP.

RFRXFLEN (0x61C8) – RX FIFO Length

Bit No.	Name	Reset	R/W	Description
7:0	D	0x00	R	Amount of data present in RX FIFO

RFRXFTHRS (0x61C9) – RX FIFO Threshold

Bit No.	Name	Reset	R/W	Description
7	–	0	R	Reserved
6:0	D	000 0000	R/W	Threshold value for RX FIFO

RFRXFWR (0x61CA) – RX FIFO Write Register

Bit No.	Name	Reset	R/W	Description
7:0	D	0x00	W	Data written to this register is written to the RX FIFO address at offset RFRXFWP from the start of the RX FIFO area (see Figure 25-1). RFRXFWP (and RFRXFSWP if RFFCFG.RXAUTODEALLOC = 1) is incremented by 1 modulo 0x80 unless the write fails.

RFRXFRD (0x61CB) – RX FIFO Read Register

Bit No.	Name	Reset	R/W	Description
7:0	D	0x00	R	When this register is read, the data in RX FIFO address offset RFRXFRP from the start of the RX FIFO area is returned (see Figure 25-1). RFRXFRP (and RFRXFSRP if RFFCFG.RXAUTOCOMMIT = 1) is incremented by 1 modulo 0x80 unless the read fails.

RFRXFWP (0x61CC) – RX FIFO Write Pointer

Bit No.	Name	Reset	R/W	Description
7	–	0	R	Reserved
6:0	D	000 0000	R/W	RX FIFO write pointer. This is the offset into the RX FIFO to which the next write operation writes.

RFRXFRP (0x61CD) – RX FIFO Read Pointer

Bit No.	Name	Reset	R/W	Description
7	–	0	R	Reserved
6:0	D	000 0000	R/W	RX FIFO read pointer. This is the offset into the RX FIFO from which the next read operation reads.

RFRXFSWP (0x61CE) – RX FIFO Start-of-Frame Write Pointer

Bit No.	Name	Reset	R/W	Description
7	–	0	R	Reserved
6:0	D	000 0000	R/W	RX FIFO start of written package. This is the point to which the write pointer can be reset if a discard command is issued.

RFRXFSRP (0x61CF) – RX FIFO Start-of-Frame Read Pointer

Bit No.	Name	Reset	R/W	Description
7	–	0	R	Reserved
6:0	D	000 0000	R/W	RX FIFO start of read package. This is the start of the allocated part of the RX FIFO.

RFTXFLEN (0x61D0) – TX FIFO Length

Bit No.	Name	Reset	R/W	Description
7:0	D	0x00	R	Amount of data present in TX FIFO

RFTXFTHRS (0x61D1) – TX FIFO Threshold

Bit No.	Name	Reset	R/W	Description
7	–	0	R	Reserved
6:0	D	000 0000	R/W	Threshold value for TX FIFO

RFTXFWR (0x61D2) – TX FIFO Write Register

Bit No.	Name	Reset	R/W	Description
7:0	D	0x00	W	Data written to this register is written to the TX FIFO address at offset RFTXFWP from the start of the TX FIFO area (see Figure 25-1) is returned. RFTXFWP (and RFTXFSWP if RFFCFG.TXAUTODEALLOC = 1) is incremented by 1 modulo 0x80 unless the write fails.

RFTXFRD (0x61D3) – TX FIFO Read Register

Bit No.	Name	Reset	R/W	Description
7:0	D	0x00	R	When this register is read, the data in TX FIFO address offset RFTXFRP from the start of the TX FIFO area is returned (see Figure 25-1). RFTXFRP (and RFTXFSRP if RFFCFG.TXAUTOCOMMIT = 1) is incremented by 1 modulo 0x80 unless the read fails.

RFTXFWP (0x61D4) – TX FIFO Write Pointer

Bit No.	Name	Reset	R/W	Description
7	–	0	R	Reserved
6:0	D	000 0000	R/W	TX FIFO write pointer. This is the offset into TX FIFO the next write operation writes to.

RFTXFRP (0x61D5) – TX FIFO Read Pointer

Bit No.	Name	Reset	R/W	Description
7	–	0	R	Reserved
6:0	D	000 0000	R/W	TX FIFO read pointer. This is the offset into TX FIFO the next read operation reads from.

RFTXFSWP (0x61D6) – TX FIFO Start-of-Frame Write Pointer

Bit No.	Name	Reset	R/W	Description
7	–	0	R	Reserved
6:0	D	000 0000	R/W	TX FIFO start of written package. This is the point to which the write pointer can be reset if a discard command is issued.

RFTXFSRP (0x61D7) – TX FIFO Start-of-Frame Read Pointer

Bit No.	Name	Reset	R/W	Description
7	–	0	R	Reserved
6:0	D	0x00	R/W	TX FIFO start-of-read package. This is the start of the allocated part of the TX FIFO.

BSP_P0 (0x61E0) – CRC Polynomial Byte 0

Bit No.	Name	Reset	R/W	Description
7:0	P[7:0]	0x00	R/W	Bits 7:0 of p register in CRC sub-module

BSP_P1 (0x61E1) – CRC Polynomial Byte 1

Bit No.	Name	Reset	R/W	Description
7:0	P[15:8]	0x5B	R/W	Bits 15:8 of p register in CRC sub-module

BSP_P2 (0x61E2) – CRC Polynomial Byte 2

Bit No.	Name	Reset	R/W	Description
7:0	P[23:16]	0x06	R/W	Bits 23:16 of p register in CRC sub-module

BSP_P3 (0x61E3) – CRC Polynomial Byte 3

Bit No.	Name	Reset	R/W	Description
7:0	P[31:24]	0x00	R/W	Bits 31:24 of p register in CRC sub-module

BSP_D0 (0x61E4) – CRC Value Byte 0

Bit No.	Name	Reset	R/W	Description
7:0	D[7:0]	0x00	R/W	Bits 7:0 of d register in CRC sub-module

BSP_D1 (0x61E5) – CRC Value Byte 1

Bit No.	Name	Reset	R/W	Description
7:0	D[15:8]	0x5B	R/W	Bits 15:8 of d register in CRC sub-module

BSP_D2 (0x61E6) – CRC Value Byte 2

Bit No.	Name	Reset	R/W	Description
7:0	D[23:16]	0x06	R/W	Bits 23:16 of d register in CRC sub-module

BSP_D3 (0x61E7) – CRC Value Byte 3

Bit No.	Name	Reset	R/W	Description
7:0	D[31:24]	0x00	R/W	Bits 31:24 of d register in CRC sub-module

BSP_W (0x61E8) – Whitener Value

Bit No.	Name	Reset	R/W	Description
7	W_PN9_RESET	0	R0	When a 1 is written to this bit, the CC2500-compatible whitener is reset, and all bits in the s and b registers are set to 1.
6:0	W[6:0]	110 0101	R/W	Write: Writes all whitening registers. w_6 is set to BSP_W[0], w_5 is set to BSP_W[1] and so on up to w_1 is set to BSP_W[5]. w_0 is set to 1. Read: Reads back w register. BSP_W[0] is set to w_6, BSP_W[1] is set to w_5 and so on up to BSP_W[6] is set to w_0.

BSP_MODE (0x61E9) – Bit Stream Processor Configuration

Bit No.	Name	Reset	R/W	Description
7	–	0	R0	Reserved. Read as zero
6	CP_BUSY	0	R	Coprocessor mode busy. Goes to 1 after a byte has been written to BSP_DATA. Goes to 0 when a byte is ready to be read back from BSP_DATA
5	CP_READOUT	0	R/W	Coprocessor mode readout
4	CP_END	0	R/W	Endianness of data in coprosessor mode. 0: LSB processed first 1: MSB processed first
3:2	CP_MODE[1:0]	00	R/W	Coprocessor mode 00: Coprocessor disabled 01: Coprocessor receive mode 10: Reserved 11: Coprocessor transmit mode
1	W_PN9_EN	0	R/W	Enable CC2500-compatible PN9 whitener
0	W_PN7_EN	1	R/W	Enable PN7 whitener

BSP_DATA (0x61EA) – Bit Stream Processor Coprocessor Data

Bit No.	Name	Reset	R/W	Description
7:0	BSP_DATA[7:0]	0x00	R/W	When BSP_MODE.CP_BUSY = 0: Write: Provide byte to be processed in coprocessor mode Read: Read processed byte

DC_I_L (0x61FC) – In-Phase DC Offset Estimate, Low Byte

Bit No.	Name	Reset	R/W	Description
7:0	DC_I[7:0]	0x00	R*/W	When running dc estimation, this register reflects the 8 LSBs of the dc estimate in the I channel. When manual dc override is selected, the override value is written to this register.

DC_I_H (0x61FD) – In-Phase DC Offset Estimate, High Byte

Bit No.	Name	Reset	R/W	Description
7:0	DC_I[15:8]	0x00	R*/W	When running dc estimation, this register reflects the 8 MSBs of the dc estimate in the I channel. When manual dc override is selected, the override value is written to this register.

DC_Q_L (0x61FE) – Quadrature-Phase DC Offset Estimate Low Byte

Bit No.	Name	Reset	R/W	Description
7:0	DC_Q[7:0]	0x00	R*/W	When running dc estimation. this register reflects the 8 LSBs of the dc estimate in the Q channel. When manual dc override is selected, the override value is written to this register.

DC_Q_H (0x61FF) – Quadrature-Phase DC Offset Estimate High Byte

Bit No.	Name	Reset	R/W	Description
7:0	DC_Q[15:8]	0x00	R*/W	When running dc estimation, this register reflects the 8 MSBs of the dc estimate in the Q channel. When manual dc override is selected, the override value is written to this register.

IVCTRL (0x6265) – Analog Control Register

Bit No.	Name	Reset	R/W	Description
7:6	–	00	R0	Reserved
5:4	TX_MIX_LOAD	01	R/W	Controls load capacitor in TX mixer 00: Minimum load ... (Intermediate loads) 11: Maximum load
3	LODIV_BIAS_CTRL	0	R/W	Controls bias current to LODIV 0: IVREF bias 1: PTAT bias
2	–	0	R/W	Reserved
1:0	PA_BIAS_CTRL	11	R/W	Controls bias current to PA 00: IREF bias 01: IREF and IVREF bias 10: PTAT bias 11: Increased PTAT slope bias