



## TABLE OF CONTENTS

Introduction .....	1	Tx Automatic Unslotted CSMA-CA Operation.....	9
Revision History .....	2	Gain Control During CCA .....	10
Register Map Extension .....	3	Code Download Sequence .....	11
Frame Filtering Algorithm .....	7	Further Information.....	12
Frame Filtering Requirements.....	7	Download.....	12
More on Frame Filtering .....	7	References.....	12
Rx Automatic Acknowledgment .....	8		

## REVISION HISTORY

### 9/2017—Rev. A to Rev. B

Added ADF7241 .....	Throughout
Changed RCCM_IEEEEX_R1 to RCCM_IEEEEX_R6....	Throughout
Changes to Register Map Extension Section .....	3
Added Table 5 and Table 6; Renumbered Sequentially .....	6
Added Gain Control During CCA Section.....	9

## REGISTER MAP EXTENSION

The RCCM\_IIEEEX\_R6 module is configured with the additional registers listed in Table 1. To enable the code module and the associated register map extension, the `pkt_cfg.addon_en` configuration bit must be set.

All registers listed in Table 1 are in an undefined state after the PRAM code download. Therefore, it is important that all registers listed in Table 1 are initialized prior to setting the `pkt_cfg.addon_en` bit. Also, write 0x8D to the modem configuration register (MCR), Register 0x3FB and write 0x6B to the MCR, Register 0x3FC.

**Table 1. Automatic IEEE 802.15.4-2006 Mode Register Map Extension**

Register Address	Register Name	Bit Field	Bit Field Name	Access Mode	Reset Value	Description
0x112	pan_id0	[7:0]	pan_id0	R/W	X	Lower 8 bits of MAC PAN ID (pan_id) for frame filtering
0x113	pan_id1	[7:0]	pan_id1	R/W	X	Higher 8 bits of MAC PAN ID (pan_id) for frame filtering
0x114	short_addr_0	[7:0]	short_addr_0	R/W	X	Lower 8 bits of MAC short address (short_addr) for frame filtering
0x115	short_addr_1	[7:0]	short_addr_1	R/W	X	Higher 8 bits of MAC short address (short_addr) for frame filtering
0x116	ieee_addr_0	[7:0]	ieee_addr_0	R/W	X	Bits[7:0] of IEEE MAC address for frame filtering
0x117	ieee_addr_1	[7:0]	ieee_addr_1	R/W	X	Bits[15:8] of IEEE MAC address for frame filtering
0x118	ieee_addr_2	[7:0]	ieee_addr_2	R/W	X	Bits[23:16] of IEEE MAC address for frame filtering
0x119	ieee_addr_3	[7:0]	ieee_addr_3	R/W	X	Bits[31:24] of IEEE MAC address for frame filtering
0x11A	ieee_addr_4	[7:0]	ieee_addr_4	R/W	X	Bits[39:32] of IEEE MAC address for frame filtering
0x11B	ieee_addr_5	[7:0]	ieee_addr_5	R/W	X	Bits[47:40] of IEEE MAC address for frame filtering
0x11C	ieee_addr_6	[7:0]	ieee_addr_6	R/W	X	Bits[55:48] of IEEE MAC address for frame filtering
0x11D	ieee_addr_7	[7:0]	ieee_addr_7	R/W	X	Bits[63:56] of IEEE MAC address for frame filtering
0x11E	ffilt_cfg	0	accept_beacon_frames	R/W	X	Frame filter 0: discards beacon frames 1: accepts beacon frames
		1	accept_data_frames	R/W	X	Frame filter 0: discards data frames 1: accepts data frames
		2	accept_ack_frames	R/W	X	Frame filter 0: discards ACK frames 1: accepts ACK frames
		3	accept_maccmd_frames	R/W	X	Frame filter 0: discards MAC command frames 1: accepts MAC command frames
		4	accept_reserved_frames	R/W	X	Frame filter 0: discards reserved frames (FCF[2:0] = reserved) 1: accepts reserved frames
		5	accept_all_address	R/W	X	Address filter 0: enabled 1: accept all addresses
		[7:6]	Reserved	R/W	X	Reserved; set to 0
0x11F	auto_cfg	0	auto_ack_framepend	R/W	X	Controls Bit FCF[5] (frame pending) in ACK frame transmitted during automatic Rx
		1	is_pancoord	R/W	X	0: device is not PAN coordinator 1: device is PAN coordinator
		2	Reserved	R/W	X	Reserved, set to 0
		3	rx_auto_ack_en	R/W	X	0: disable 1: enable automatic ACK on Rx
		4	csma_ca_turnaround	R/W	X	0: disable 1: enable automatic turnaround
	[7:5]	Reserved	R/W	X	Reserved; set to 0	

Register Address	Register Name	Bit Field	Bit Field Name	Access Mode	Reset Value	Description
0x120	auto_tx1	[3:0]	max_frame_retries	R/W	X	Specifies number of attempts to retransmit unacknowledged frames while in automatic CSMA-CA Tx mode (auto_csma_tx_en = 1)
		[6:4]	max_cca_retries	R/W	X	Specifies number of attempts to repeat CSMA-CA algorithm prior to cancellation of RC_TX command (tx_auto_csma_en = 1); valid range is 0 to 5; 7: CSMA-CA algorithm is off
		7	Reserved	R/W	X	Reserved; set to 0
0x121	auto_tx2	[3:0]	csma_max_be	R/W	X	Specifies the maximum back-off exponent used in the CSMA-CA algorithm; valid range is 3 to 8
		[7:4]	csma_min_be	R/W	X	Specifies the minimum back-off exponent used in the CSMA-CA algorithm; valid range is 0 to csma_max_be
0x122	auto_status	[2:0]	auto_status	R	X	0: SUCCESS 1: SUCCESS_DATPEND 2: FAILURE_CSMACA 3: FAILURE_NOACK 4: ERROR_CFG 5...7: Reserved

The RCCM\_IEEEX\_R6 code module provides an additional interrupt source named address\_match to signal the detection of an address match in a received frame. The address\_match interrupt is asserted when frame filtering has been enabled (pkt\_cfg.addon\_en = 1 and ffilt\_cfg.accept\_all\_address\_en = 0) and the received frame meets all requirements of the frame filtering algorithm described in the Frame Filtering Algorithm section.

Table 2, Table 3, and Table 4 list the locations of the control bits for the address\_match interrupt in the irq1\_en1, irq2\_en1 and irq\_src1 registers, respectively. The conventions outlined in the Interrupt Controller section in the [ADF7241](#) and [ADF7242](#) data sheets fully apply.

**Table 2. RCCM\_IEEEX\_R6 Register irq1\_en1 Update**

Register Address	Register Name	Bit Field	Bit Field Name	Access Mode	Reset Value	Description
0x3C8	irq1_en1	0	cca_complete	R/W	0	CCA result in status word valid
		1	rx_sfd	R/W	0	SFD detected during Rx operation
		2	tx_sfd	R/W	0	Start of SFD transmit
		3	rx_pkt_rcvd	R/W	0	Packet received in RX_BUFFER
		4	tx_pkt_sent	R/W	0	Packet in TX_BUFFER sent
		5	frame_valid	R/W	0	Allowed frame detected during Rx
		6	address_valid	R/W	0	Address match detected during Rx
		7	csma_ca_complete	R/W	0	csma_ca operation has finished

**Table 3. RCCM\_IEEEX\_R6 Register irq2\_en1 Update**

Register Address	Register Name	Bit Field	Bit Field Name	Access Mode	Reset Value	Description
0x3CA	irq2_en1	0	cca_complete	R/W	0	CCA result in status word valid
		1	rx_sfd	R/W	0	SFD detected during Rx operation
		2	tx_sfd	R/W	0	Start of SFD transmit
		3	rx_pkt_rcvd	R/W	0	Packet received in RX_BUFFER
		4	tx_pkt_sent	R/W	0	Packet in TX_BUFFER sent
		5	frame_valid	R/W	0	Allowed frame detected during Rx
		6	address_valid	R/W	0	Address match detected during Rx
		7	csma_ca_complete	R/W	0	csma_ca operation has finished

**Table 4. RCCM\_IEEEX\_R6 Register irq\_src1 Update**

Register Address	Register Name	Bit Field	Bit Field Name	Access Mode	Reset Value	Description
0x3CC	irq_src1	0	cca_complete	R/W	0	CCA result in status word valid
		1	rx_sfd	R/W	0	SFD detected during Rx operation
		2	tx_sfd	R/W	0	Start of SFD transmit
		3	rx_pkt_rcvd	R/W	0	Packet received in RX_BUFFER
		4	tx_pkt_sent	R/W	0	Packet in TX_BUFFER sent
		5	frame_valid	R/W	0	Allowed frame detected during Rx
		6	address_valid	R/W	0	Address match detected during Rx
		7	csma_ca_complete	R/W	0	csma_ca operation has finished

Table 5 and Table 6 support the option when using the RCCM\_IIIEEX\_R6 code module to set the gain control to manual during the CCA.

**Table 5. RCCM\_IIIEEX\_R6 Automatic Gain Control (AGC) Register Readback**

Register Address	Register Name	Bit Field	Bit Field Name	Access Mode	Reset Value	Description
0x3B1	agc_readback	[1:0]	lna_out	R	0	LNA gain readback 0: 22 dB 1: 10 dB 2, 3: -2 dB
		[5:2]	pga_out	R	0	PGA readback PGA gain (dB) = 30 dB - 3 dB × pga_out

**Table 6. RCCM\_IIIEEX\_R6 Manual Gain Control Configuration Register**

Register Address	Register Name	Bit Field	Bit Field Name	Access Mode	Reset Value	Description
0x3B3	manual_gain_control	0	man_lna_gain	R/W	0	0: auto LNA gain control 1: manual LNA gain control
		[2:1]	lna_gain	R/W	0	User set LNA gain value when man_lna_gain = 1
		3	man_pga_gain	R/W	0	0: auto PGA gain control 1: manual PGA gain control
		[7:4]	pga_gain	R/W	0	User set PGA gain value when man_pga_gain = 1

## FRAME FILTERING ALGORITHM

Frame filtering is only available when the [ADF7241](#) and [ADF7242](#) operate in IEEE 802.15.4-2006 packet mode (`rc_cfg.rc_mode = 0`). The frame filtering function rejects received frames not intended for the wireless node. The filtering procedure is a superset of the procedure described in Section 7.5.6.2 (third filtering level) of the IEEE 802.15.4-2006 standard.

The `pkt_cfg.addon_en` bit controls whether frame filtering is enabled or not. If `ffilt_cfg.accept_all_address = 1`, the address information contained in the received frame is ignored, while the configuration bits in the `ffilt_cfg` register still control filtering of the frame control field (FCF).

### FRAME FILTERING REQUIREMENTS

If `ffilt_cfg.accept_all_address = 0`, only frames fulfilling all of the requirements noted in the following sections are accepted.

#### Frame Integrity

- The Frame Type subfield of the FCF must contain a nonreserved frame type.
- The Frame Version subfield of the FCF must contain a nonreserved value.
- All source and destination address modes in the FCF are nonreserved values.

#### Destination Address

- If a destination personal area network (PAN) identifier is present, it must match `pan_id` or the broadcast PAN identifier (0xFFFF).
- If a short destination address is included in the frame, it must either match `short_addr` or Broadcast Address 0xFFFF.
- If an extended destination address is included in the frame, it must match `ieee_addr_0` to `ieee_addr_7`.

#### Frame Type—Beacon

- Configuration Bit `ffilt_cfg.accept_beacon_frames` must be set.
- The destination address mode is 0 (no destination address).
- A source address is included (a Source Address Mode 2 or Source Address Mode 3).
- The source PAN identifier must match `pan_id` or `pan_id = 0xFFFF`. If `pan_id = 0xFFFF`, the beacon frame is accepted regardless of the source PAN identifier.

#### Frame Type—Data Frame

- The `ffilt_cfg.accept_data_frames` configuration bit must be set.
- A destination address and/or source address must be included in the frame. If no destination address is included in the frame, the `auto_cfg.is_pancoord` bit must be set and the source PAN identifier in the FCF must be equal to `pan_id`.

#### Frame Type—Acknowledgment (ACK)

- The `ffilt_cfg.accept_ack_frames` configuration bit must be set.
- Length byte must equal 5.

#### Frame Type—Media Access Control (MAC) Command Frame

- The `ffilt_cfg.accept_maccmd_frames` configuration bit must be set.
- A destination address and/or source address must be included in the frame. If no destination address is included in the frame, the `auto_cfg.is_pancoord` bit must be set and the source PAN identifier in the FCF must be equal to `pan_id`.

#### Frame Type—Reserved Frame

- Configuration Bit `ffilt_cfg.accept_reserved_frames` must be set.
- A destination address and/or source address must be included in the frame. If no destination address is included in the frame, the `auto_cfg.is_pancoord` bit must be set and the source PAN identifier in the FCF must be equal to `pan_id`.

### MORE ON FRAME FILTERING

If a frame is detected and its type is received, a `frame_valid` interrupt is generated. If enabled, an `address_valid` interrupt is also generated when a received frame is accepted by the frame filtering algorithm. While the transceiver is in the Rx state, the `RX_BUFFER` is overwritten each time a frame is received, even if the frame is subsequently rejected during the frame filtering procedure.

Prior to performing address filtering, Bit 7 of the length byte is cleared. The `address_match` interrupt is asserted even if the frame check sequence (FCS) field of received frame is invalid. When enabled, the `rx_pkt_rcvd` interrupt is only asserted when the received frame is accepted by the frame filtering algorithm and its FCS is correct.

It is not a requirement to exit the Rx state when changing the values of the `pan_id`, `short_addr`, `ieee_addr` registers or the configuration bits contained in Register `ffilt_cfg`. However, if the changes are applied between the reception of a valid start of frame delimiter (SFD) and the source address field, the frame filtering function is undefined because the [ADF7241](#) and the [ADF7242](#) use either the old or updated values.

## Rx AUTOMATIC ACKNOWLEDGMENT

The [ADF7241](#) and [ADF7242](#) offer a feature that enables the automatic transmission of acknowledgment frames after successfully receiving a frame. The Rx automatic acknowledgment feature can only be used in conjunction with the address filtering feature. It is enabled when the `auto_cfg.rx_auto_ack_en` bit field is set, and the `ffilt_cfg.accept_all_address = 0` bit field is reset. For correct operation, the `pkt_cfg.auto_fcs_off` bit field and the `buffercfg.rx_buffer_mode` bit field must both be set to 0.

When enabled, an acknowledgment frame is automatically transmitted when the following conditions are met:

- The received frame is accepted by the frame filtering procedure.
- The received frame is not a beacon or acknowledgment frame.
- The acknowledgment request bit is set in the FCF of the received frame.

Figure 2 shows the format of the acknowledgment frame assembled by the [ADF7241](#) and [ADF7242](#). The sequence number (SEQ NUM) is copied from the frame stored in the `RX_BUFFER`. The Rx automatic acknowledgment feature utilizes the `TX_BUFFER` to store the constructed acknowledgment frame prior to its transmission. Any data present in the `TX_BUFFER` is overwritten by the acknowledgment frame prior to its transmission.

When the acknowledgment frame is sent in response to a data request MAC command frame, the content of the frame pending bit in the FCF of the constructed acknowledgment

frame is defined by the `auto_cfg.auto_ack_framepend` bit. Otherwise, the frame pending bit is set to 0.

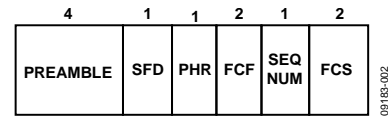


Figure 2. ACK Frame Format

The transmission of the ACK frame starts after the combined delay given by the sum of the delays specified in Register `delaycfg1.tx_mac_delay` and Register `delaycfg2.mac_delay_ext` has elapsed.

The default settings of `delaycfg1.tx_mac_delay = 192` and `delaycfg2.mac_delay_ext = 0` result in a delay of 192  $\mu$ s, which suits networks using unslotted CSMA-CA.

Optionally, Register `delaycfg2.mac_delay_ext` can be updated dynamically while the delay specified in Register `delaycfg1.tx_mac_delay` elapses. This option enables accurate alignment of the acknowledgment frame with the back off slot boundaries in networks using slotted CSMA-CA.

When Rx automatic acknowledgment mode is enabled, the [ADF7241](#) and [ADF7242](#) remain in an Rx state until a valid frame is received. When enabled, an `rx_pkt_rcvd` interrupt is generated when a valid frame is received. The [ADF7241](#) and [ADF7242](#) then automatically enter Tx state until the transmission of the acknowledgment frame is complete. When enabled, a `tx_pkt_sent` interrupt is generated to signal the end of the transmission phase. Subsequently, the [ADF7241](#) and [ADF7242](#) return to the `PHY_RDY` state.



## Tx AUTOMATIC UNSLOTTED CSMA-CA OPERATION

The automatic CSMA-CA transmit operation automatically performs all necessary steps to transmit frames in accordance with the IEEE 802.15.4-2006 standard for unslotted CSMA-CA network operation. It includes automatic clear channel assessment (CCA) retries with random back off, frame transmission, reception of the acknowledgment frame, and automatic retries in the case of transmission failure. Partial support is provided for slotted CSMA-CA operation.

**Table 7. Additional Command Provided by Firmware Module**

Command	Code	Description
RC_CSMACA	0xC1	Available when the RCCM_IEEEX_R6 module is loaded into PRAM; initiates CSMA-CA channel access sequence and frame transmission

The automatic CSMA-CA transmit sequence is initiated when command RC\_CSMACA is issued. Table 7 lists the command details. The RC\_CSMACA command is available when the firmware module is successfully loaded into the PRAM. The command is valid when the ADF7241 and ADF7242 are in the PHY\_RDY state. It is used like other SPI commands. Command access and status word details can be found in the ADF7241 and ADF7242 data sheets.

Command RC\_CSMACA is similar to command RC\_TX with the addition of an automatic CSMA-CA channel access sequence and automatic ACK validation. The number of CSMA-CA CCA retries is specified with Register auto\_tx1.max\_cca\_retries, which must be between 0 and 5 in accordance with the IEEE 802.15.4-2006 standard for unslotted CSMA/CA network operation. Setting auto\_tx1.max\_cca\_retries = 7 disables the CSMA-CA procedure and causes the transmission of the frame to commence immediately after the MAC delay expires. This configuration facilitates the implementation of the transmit

procedure in networks using slotted CSMA-CA. In this case, the timing of the CCA operation must be controlled by the user MCU, and the value of Register auto\_tx1.max\_frame\_retries must be set to 1.

The number of frame transmission retries is configured with Register auto\_tx1.max\_frame\_retries. It specifies the number of unacknowledged or incorrectly acknowledged frame transmissions before the radio controller cancels the operation and signals a failure condition.

Prior to the transmission of the frame stored in the TX\_BUFFER, the radio controller checks if the acknowledge request bit in the FCF of that frame is set. If set, an acknowledgment frame is expected following the transmission. Otherwise, the transaction is complete after the frame is transmitted. The acknowledge request bit is Bit 5 of the byte located at the address contained in txpb.tx\_packet\_base + 1.

Figure 3 depicts the automatic CSMA-CA Tx procedure when a CSMA-CA phase is requested (auto\_tx1.max\_cca\_retries = 0 ... 5). It is initiated with the RC\_CSMACA command. Prior to the ADF7241 and ADF7242 starting the CSMA-CA phase, the total Rx MAC delay comprised of the delay defined by Register delaycfg0.rx\_mac\_delay and Register delaycfg2.mac\_delay\_ext elapses.

During the CSMA-CA phase, the algorithm outlined in Section 7.5.1.4 in the IEEE 802.15.4-2006 standard for unslotted CSMA-CA is executed. It begins with the ADF7241 and ADF7242 delaying the first CCA by a random number of back off periods with a nominal length of 320 μs each. The maximum number of back off periods is determined by the back off exponent, BE, according to  $2^{BE} - 1$ . The back off exponent is initialized with the value contained in Register auto\_tx2.csm\_min\_be.

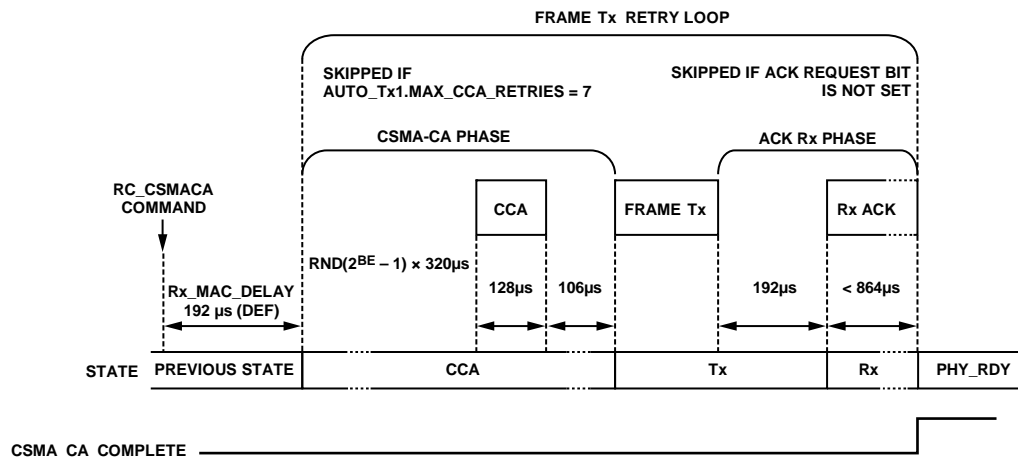


Figure 3. Automatic CSMA-CA Tx Operation (with CCA)

The random number is derived from a pseudorandom binary sequence (PRBS) generator. If a busy channel is detected during the CCA, the back off exponent, BE, is incremented until it reaches the value configured in Register `auto_tx2.csm_max_be`. The radio controller performs the next delay/CCA cycle until the maximum number of CCA retries specified with Register `auto_tx1.max_cca_retries` is reached. If the maximum number of allowed CCA retries is reached, the operation is aborted with the status `auto_status.auto_status = FAILURE_CSMACA`.

If the CCA is successful, the radio controller changes from a CCA state to a Tx state and transmits the frame stored in the `TX_BUFFER`. If neither the acknowledge request bit in the transmitted frame nor the `csm_ca_turnaround` bit are set, the radio controller falls back to the `PHY_RDY` state immediately upon completion of the frame transmission.

The procedure exits with `auto_status.auto_status = SUCCESS` and triggers the `csm_ca_complete` interrupt. Otherwise, it enters Rx.

If the acknowledge request bit is set in the transmitted frame, the radio controller enters the Rx state upon completion of the frame transmission. While in the Rx state, it is waiting for a valid acknowledgment frame. If an acknowledgment frame is not received within 864  $\mu$ s, the radio controller checks if the maximum number of frame retries specified with Register `auto_tx1.max_frame_retries` is reached. If not, the radio controller remains inside the frame transmit retry loop and starts the next CSMA-CA cycle. If the maximum number of frame transmission retries is reached, the procedure exits to `PHY_RDY` with `auto_status.auto_status = FAILURE_NOACK`.

If an acknowledgment frame is received while the Rx state is active, the radio controller compares the sequence number contained in that acknowledgment frame with the sequence number of the frame stored in the `TX_BUFFER`. The sequence number is expected at location `txpb.tx_packet_base + 3` in the `TX_BUFFER`. If the sequence numbers do not match, the radio controllers start a further iteration of the transmit retry loop if the maximum number of retries is not yet reached. If the sequence numbers match, the radio controller checks the frame pending bit in the received acknowledgment frame. If the frame pending bit is set, the procedure exits to Rx with status `auto_status.auto_status = SUCCESS_DATPEND`. If the frame pending bit is not set, the procedure exits with status `auto_status.auto_status = SUCCESS`. If the `csm_ca_turnaround` bit is set, the radio controller enters Rx; otherwise, it enters the `PHY_RDY` state.

Note that the acknowledgment frame overwrites the contents of the `RX_BUFFER`. Irrespective of a successful or unsuccessful outcome of the transmit procedure, a `csm_ca_complete` interrupt is generated when enabled. The status word contained in `auto_status.auto_status` is only valid after the `csm_ca_complete` interrupt is asserted or the radio controller status word indicates a `PHY_RDY` state.

The automatic CSMA-CA Tx operation can be aborted at any time by means of an appropriate SPI command. In this case, no `tx_pkt_sent` interrupt is generated and the status register `auto_status.auto_status` is undefined.

### GAIN CONTROL DURING CCA

The `RCCM_IEEEX_R6` module adds the option to use manual gain control during CCA. Manually setting the gain enables improved repeatability for CCA measurements. The firmware checks the Register `0x3B3` setting to determine whether manual gain control is selected for use during CCA and, if so, what the manual gain setting is. The range of received signal strength indication (RSSI) readings, which may be obtained with a fixed gain manually set, is typically 16 dB.

To optimize accuracy when the channel energy is close to the target CCA threshold, it is recommended to set the manual gain so that the threshold is centered within this range. If the energy is much higher or lower than the threshold, a valid busy or clear assessment can be made, though the result clamps to the higher or lower end of the RSSI readback range, respectively. For a CCA threshold of  $-68$  dBm, a Register `0x3B3` setting of `0x4B` is recommended.

If a modification to the target threshold is desired, the center of the range can be increased by 3 dB by setting Register `0x3B3` to `0x5B`, while setting it to `0x3B` decreases the center of the range by 3 dB. There is also an option to read Register `0x3B1` with a signal applied at the target threshold level and the default AGC enabled. The readback `pga_out` and `lna_out` gain settings from Register `0x3B1` can then be applied in Register `0x3B3`. This readback process is useful if a very different CCA threshold is desired.

If manual gain is selected, the firmware reverts the device to automatic gain control when receiving an acknowledge by updating the Register `0x3B3` setting back to `0x00`. If manual gain control for CCA is selected and an expected acknowledge is not received, the device automatically returns to manual gain control during the next CCA period.

## CODE DOWNLOAD SEQUENCE

The PRAM is organized into eight pages with a length of 256 bytes each. The RCCM\_IEEEX\_R6 code module must be stored in the PRAM starting from Address 0x0000 or Address 0x00 in Page 0. The current PRAM page is selected with the `prampg.pram_page` register.

Prior to downloading the PRAM, the radio controller code module must be divided into blocks of 256 bytes commensurate with the size of the PRAM pages. Each 256-byte block is downloaded into the currently selected PRAM page using the `SPI_PRAM_WR` command.

Table 8 shows the command relevant for the PRAM code upload. Figure 4 shows the sequence required for downloading a code block of 256 bytes to a PRAM page. In the first step, the current PRAM page number is configured. The page number must increase for every 256-byte block written. Subsequently,

the module code block is downloaded into the selected page using the `SPI_PRAM_WR` command. The `SPI_PRAM_WR` command code is followed by Address Byte 0x00 to align the code block with the base address of the PRAM page.

Figure 5 shows the overall download sequence for the code module. With the exception of the last page written to the PRAM, all pages must be filled with 256 bytes of module code.

The `SPI_PRAM_RD` command reads from the program RAM, which can verify that a firmware module is correctly written to the program RAM. Like the `SPI_PRAM_WR` command, the host MCU must select the program RAM page to read via Register `prampg.pram_page`. Following this, the host MCU can use the `SPI_PRAM_RD` command (0x3E) to block read the selected program RAM page. The structure of this command is shown in Figure 6.

**Table 8. Command for PRAM Code Upload**

Command	Code	Description
<code>SPI_PRAM_WR</code>	0x1E	Write data sequentially to current PRAM page selected with <code>prampg.pram_page</code> , starting from address following command code.

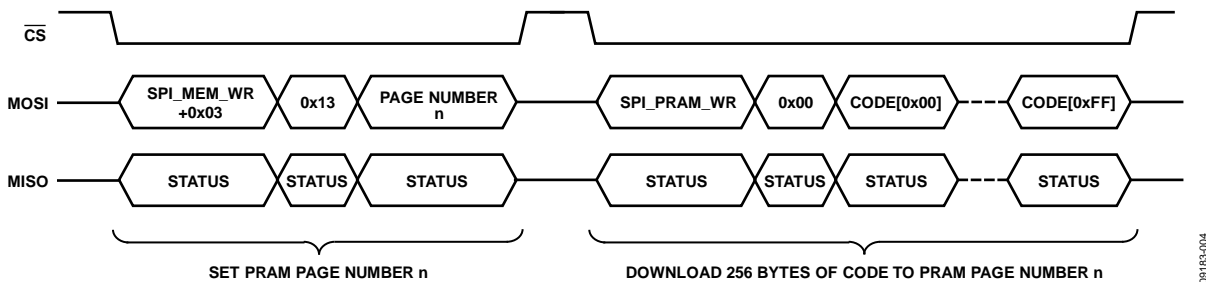


Figure 4. Download Sequence for a PRAM Page



Figure 5. Download Sequence for the Code Module

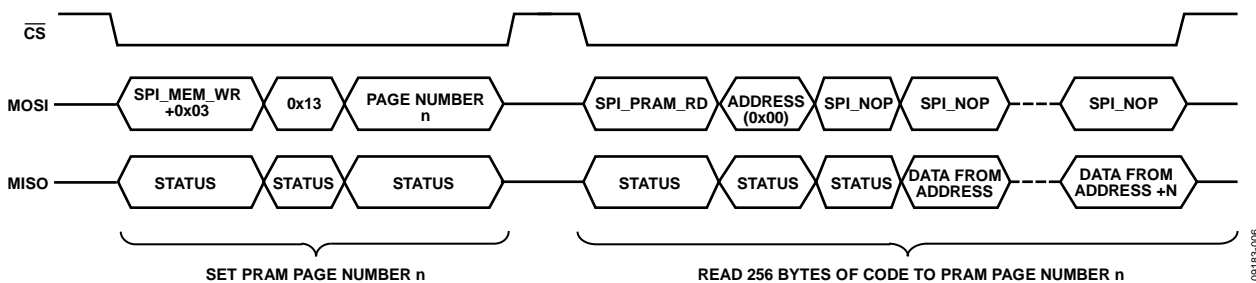


Figure 6. Read Sequence for a PRAM Page

## FURTHER INFORMATION

### DOWNLOAD

The [AN-1082 RCCM\\_IEEEX\\_R6 Code.zip file](#), which contains the code referenced in this application note, is available online.

Copyright (c) Analog Devices, Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of Analog Devices, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
- The use of this software may or may not infringe the patent rights of one or more patent holders. This license does not release you from the requirement that you obtain separate licenses from these patent holders to use this software.

THIS SOFTWARE IS PROVIDED BY ANALOG DEVICES "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, NON-INFRINGEMENT, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ANALOG DEVICES BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, INTELLECTUAL PROPERTY RIGHTS, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE

### REFERENCES

IEEE Standard for Information Technology, Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) for Low-Rate Wireless Personal Area Networks (WPANs), IEEE Standard 802.15.4-2006.