ANALOG DEVICES

AN-1082
APPLICATION NOTE

One Technology Way • P.O. Box 9106 • Norwood, MA 02062-9106, U.S.A. • Tel: 781.329.4700 • Fax: 781.461.3113 • www.analog.com

Automatic IEEE 802.15.4 Operating Modes

by Mary O'Keefe

INTRODUCTION

This application note describes automatic IEEE 802.15.4 operating modes for the ADF7242 transceiver IC, which are enabled by the radio controller code module RCCM_IEEEX_R1. The RCCM_IEEEX_R1 code module adds the following features to the ADF7242:

- Automatic IEEE 802.15.4 frame filtering
- Automatic acknowledgement of received valid IEEE 802.15.4 frames
- Automatic frame transmission using unslotted CSMA_CA with automatic retries

Figure 1 shows a block diagram of the ADF7242 low power 2.4 GHz transceiver IC. The transceiver contains a custom MCU core with a mask ROM, which implements packet handling functions and translates radio commands into internal control sequences. An additional 2 kB of program RAM (PRAM) is available and serves as reconfigurable program code memory. It enables the addition of radio controller commands to provide modified or extended functionality. The radio controller code module described in this application note is based on program code downloads into the PRAM program code memory.

The PRAM block is volatile memory and must be reloaded each time the transceiver is waking up from sleep state. The PRAM locations can be accessed sequentially through the SPI interface. The length of the RCCM_IEEEX_R1 module is less than 2 kB. At an SCLK clock speed of 10 Mbps, the code module can thus be downloaded into the PRAM in less than 1650 μs. The details of the code download mechanism are explained in the Code Download Sequence section.

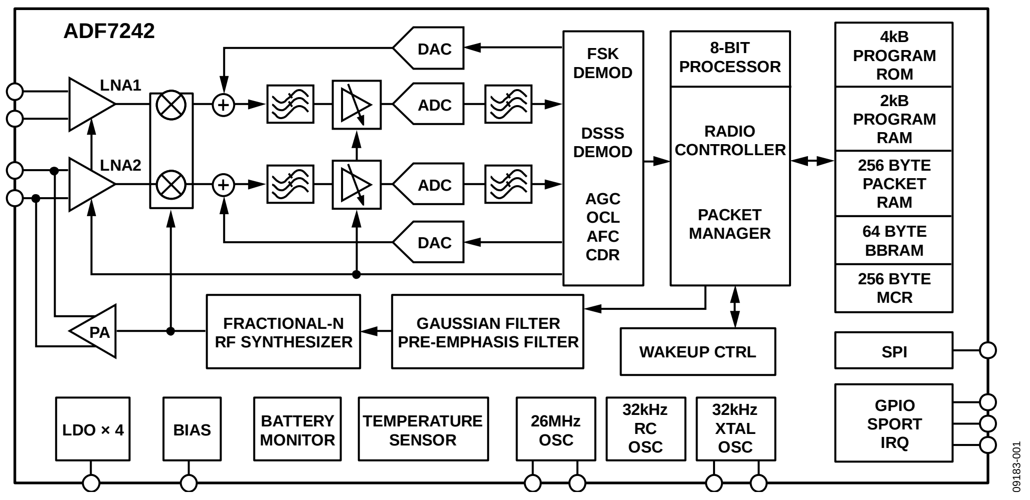

Figure 1. ADF7242 Block Diagram

TABLE OF CONTENTS

REGISTER MAP EXTENSION

The RCCM_IEEEX_R1 module is configured with the additional registers listed in Table 1. In order 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 MCR Register 0x3FB and write 0xCA to MCR Register 0x3FC.

Table 1. Automatic IEEE 802.15.4 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_R1 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, 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 ADF7242 data sheet fully apply.

Table 2. RCCM_IEEEX_R1 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_R1 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_R1 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

FRAME FILTERING ALGORITHM

Frame filtering is only available when the ADF7242 operates in IEEE 802.15.4 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 FCF field.

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 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 the 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 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 source PAN identifier in the FCF must be equal to pan_id.

Frame Type = Acknowledgement

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

Frame Type = 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 destintation 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 allowed to be 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 the address filtering, Bit 7 of the length byte is cleared. The address_match interrupt is asserted even if the FCS field of received frame is invalid. When enabled, the rx_pkt_rcvd interrupt is only asserted when the received frame has been 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 SFD and the source address field, the frame filtering function is undefined because the ADF7242 uses either the old or updated values.

RX AUTOMATIC ACKNOWLEDGMENT

The ADF7242 offers a feature that enables the automatic transmission of acknowledgement frames after successfully receiving a frame. The RX automatic acknowledge feature may only be used in conjunction with the address filtering feature. It is enabled when Bit auto_cfg.rx_auto_ack_en is set and Bit ffilt_cfg.accept_all_address = 0 is reset. For correct operation, the pkt_cfg.auto_fcs_off bit and Register buffercfg.rx_buffer_mode must both be set to 0.

When enabled, an acknowledgement 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 acknowledgement frame.
- The acknowledgement request bit is set in the FCF of the received frame.

Figure 2 shows the format of the acknowledgement frame assembled by the ADF7242. The sequence number (SEQ NUM) is copied from the frame stored in the RX_BUFFER. The RX automatic acknowledgement feature utilizes the TX_BUFFER to store the constructed acknowledgement frame prior to its transmission. Any data present in the TX_BUFFER is overwritten by the acknowledgement frame prior to its transmission.

When the acknowledgement 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.

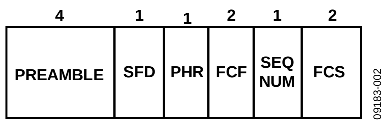

Figure 2. ACK Frame Format

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 μs, which suits networks using unslotted CSMA-CA.

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

When RX automatic acknowledgement mode is enabled, the ADF7242 remains in an RX state until a valid frame has been received. When enabled, an rx_pkt_rcvd interrupt is generated when a valid frame has been received. The ADF7242 then automatically enters TX state until the transmission of the acknowledgement frame is complete. When enabled, a tx_pkt_sent interrupt is generated to signal the end of the transmission phase. Subsequently, the ADF7242 returns to state PHY_RDY.

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 CCA retries with random back off, frame transmission, reception of the acknowledgement frame, and automatic retries in the case of transmission failure. Partial support is provided for slotted CSMA-CA operation.

Table 5. Additional Command Provided by Firmware Module

Command	Code	Description
RC_CSMACA	0xC1	Available when RCCM_IEEEX_R1 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 5 lists the command details. The RC_CSMACA command is available when the firmware module has been successfully loaded into the PRAM. The command is valid when the ADF7242 is in state PHY_RDY. It is used like other SPI commands. Command access and status word details can be found in the ADF7242 data sheet.

Command RC_CSMACA is similar to the 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 has expired.

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, then an acknowledgement frame is expected following the transmission. Otherwise, the transaction is complete after the frame has been transmitted. The Acknowledgement 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 has been requested (auto_tx1.max_cca_retries = 0 ... 5). It is initiated with the RC_CSMACA command. Prior to the 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 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.csma_min_be.

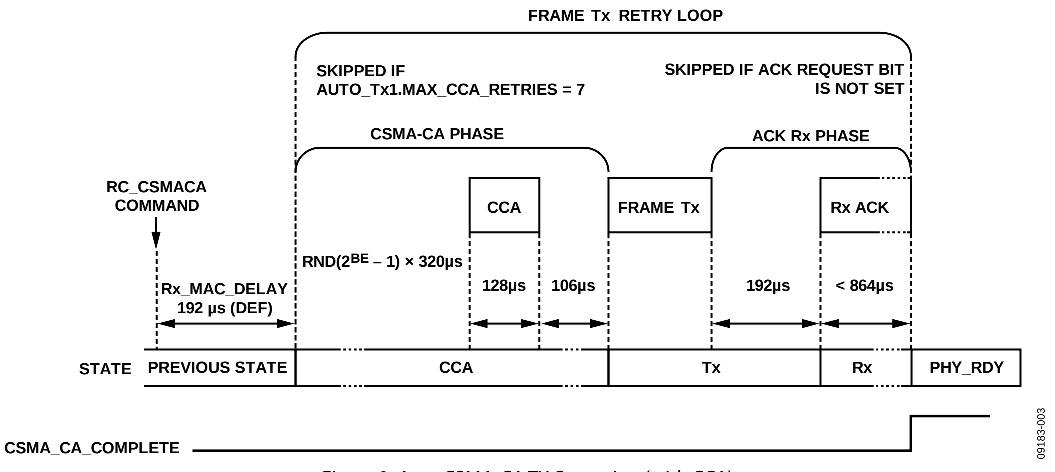

Figure 3. Auto CSMA-CA TX Operation (with CCA)

The random number is derived from a PRBS generator. If a busy channel is detected during the CCA, the back off, the exponent BE is incremented until it reaches the value configured in Register auto_tx2.csma_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 has been reached, the operation is aborted with the status auto_status.auto_status = FAILURE_CSMACA.

If the CCA was successful, the radio controller changes from CCA state to TX state and transmits the frame stored in the TX_BUFFER. If neither the acknowledge request bit in the transmitted frame nor the csma_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 csma_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 acknowledgement frame. If an acknowledgement frame is not received within 864 μs, the radio controller checks if the maximum number of frame retries specified with Register auto_tx1.max_frame_retries has been 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 has been reached, the procedure exits to PHY_RDY with auto_status.auto_status = FAILURE_NOACK.

If an acknowledgement frame is received while the RX state is active, the radio controller compares the sequence number contained in that acknowledgement 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 has not yet been reached. If the sequence numbers match, the radio controller checks the Frame Pending bit in the received acknowledgement 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 csma_ca_turnaround bit is set, the radio controller enters RX, otherwise it enters the PHY_RDY state.

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

The automatic CSMA-CA TX operation may 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.

CODE DOWNLOAD SEQUENCE

The PRAM is organized into eight pages with a length of 256 bytes each. The RCCM_IEEEX_R1 code module must be stored in the PRAM starting from Address 0x0000 or Address 0x00in Page 0. The current PRAM page is selected with Register prampg.pram_page.

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 6 shows the command relevant for the PRAM code upload. Figure 4 illustrates 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. 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 is used to read from the program RAM. This may be performed to verify that a firmware module has been 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, Field pram_page. Following this, the host MCU may 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 6. Command for PRAM Code Upload

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

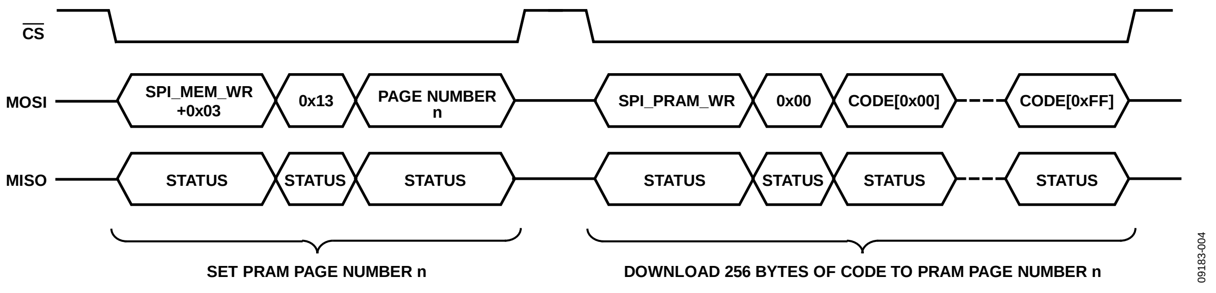

Figure 4.Download Sequence for a PRAM Page

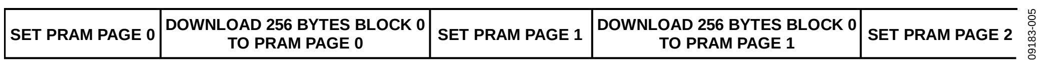
SET PRAM PAGE 0
DOWNLOAD 256 BYTES BLOCK 0 TO PRAM PAGE 0
SET PRAM PAGE 1
DOWNLOAD 256 BYTES BLOCK 0 TO PRAM PAGE 1
SET PRAM PAGE 2
09183-005

Figure 5. Download Sequence for the Code Module

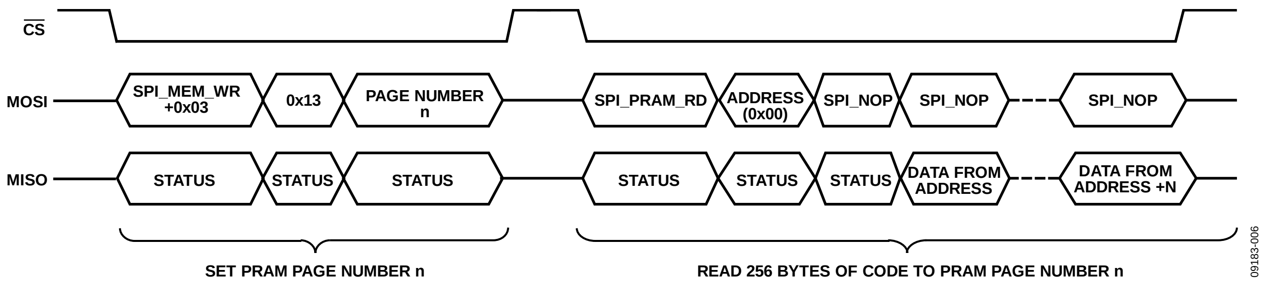

Figure 6. Read Sequence for a PRAM Page

FURTHER INFORMATION

DOWNLOAD

The AN-1082 RCCM_IEEEX_R1 Code.zip file, which contains the code referenced here, is available online.

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.

NOTES